SECURITY ISSUES AND THEIR RESOLUTIONS IN CLOUD BASED SERVICES

Thesis

Submitted to the Punjab Agricultural University
in partial fulfillment of the requirements
for the degree of

MASTER OF TECHNOLOGY
in
COMPUTER SCIENCE AND ENGINEERING
(Minor Subject: Information Technology)

By

Simranjit Kaur
(L-2018-AE-206-M)

Department of Electrical Engineering and Information Technology
College of Agricultural Engineering and Technology

LUDHIANA-141004

2020

CERTIFICATE-I

This is to certify that the thesis entitled, **"SECURITY ISSUES AND THEIR RESOLUTIONS IN CLOUD BASED SERVICES"** submitted for the degree of **M. Tech,** in the subject of **Computer Science and Engineering (Minor subject: Information Technology)** of the Punjab Agricultural University, Ludhiana, is a bonafide research work carried out by **Simranjit Kaur (L-2018-AE-206-M)** under my supervision and no part of this thesis has been submitted for any other degree.

The assistance and help received during the course of investigations have been fully acknowledged.

_____

(Dr. Lokesh Jain)
Major Advisor
Associate Professor
Department of Electrical Engineering and Information Technology,
Punjab Agricultural University,
Ludhiana- 141004

CERTIFICATE-II

This is to certify that the thesis entitled, **"SECURITY ISSUES AND THEIR RESOLUTIONS IN CLOUD BASED SERVICES"** submitted by **Simranjit Kaur (L-2018-AE-206-M)** to the Punjab Agricultural University, Ludhiana in partial fulfillment of the requirements for the degree of **M. Tech.,** in the subject of **Computer Science and Engineering (Minor subject: Information Technology)** has been approved by the Student's Advisory Committee along with Head of the Department after an oral examination on the same.

_____
(Dr. Lokesh Jain)
Major Advisor

_____
(Dr. Mukesh Kumar)
External Examiner
Associate Professor
UIET, PU, Chandigarh

_____
(Dr. Derminder Singh)
Professor and Head
Department of Electrical Engineering and Information Technology,
Punjab Agricultural University,
Ludhiana- 141004

_____
(Dr. Gurinder Kaur Sangha)
Dean, Postgraduate Studies,
Punjab Agricultural University,
Ludhiana- 141004

ACKNOWLEDGEMENT

First and foremost, I would like to offer my gratitude to the ***God Almighty*** *for the wisdom, strength, physical and mental well-being He bestowed upon me. Feeling gratitude and not expressing it is like wrapping a present and not giving it. Hence, I solemnly believe it to be my duty to appreciate all the help and support I received in the meaningful formation of this research.*

I would like to thank my advisor, ***Dr. Lokesh Jain****, Associate Professor, Department of Electrical Engineering and Technology, PAU, Ludhiana for being exceptionally supportive and providing me with all the guidance I needed right from the start.*

I express my profound thanks to my Advisory Committee, ***Dr. O P Gupta,*** *Professor, IT Section, College of Agriculture;* ***Dr. Derminder Singh,*** *(Dean PGS nominee), Professor cum Head, Department of Electrical Engineering and Technology and* ***Dr. Sourabh Ratra,*** *Assistant Professor, Department of Electrical Engineering and Technology, for their valuable criticism and elucidation during the research.*

I wish to convey my greatest acknowledgement to my parents ***Sh. Gajinder Singh Chohan*** *and* ***Smt. Paramjit Kaur****, my siblings for the unconditional love and support they have been showering me with. I am greatly indebted to them and I am obliged to have their guidance and support with me.*

At last, I would like to offer my regards and wishes to all those who helped or supported in any respect during the fruition of the project.

(Simranjit Kaur)

Title of the Thesis	:	Security issues and their resolutions in cloud based services
Name of the Student and Admission No.	:	Simranjit Kaur (L-2018-AE-206-M)
Major Subject	:	Computer Science and Engineering
Minor Subject	:	Information Technology
Name and Designation of Major Advisor	:	Dr. Lokesh Jain Associate Professor
Degree to be awarded	:	Master of Technology
Year of award of degree	:	2020
Total pages in thesis	:	45 + Annexures (xxxv) + VITA
Name of University	:	Punjab Agricultural University, Ludhiana-141004, Punjab, India

ABSTRACT

The use of cloud technology is surging high these days and it is capable of handling tremendous amount of business records and processes. It not only cuts the cost incurred by storing and handling data at a very minimal price, but also enables the users to stay connected to their data at any given moment just by connecting to the Internet. Although this added advantage encourages a lot of people to store their data in the cloud, the cloud still requires a deep analysis to eradicate the many issues it has. This research focuses on going through cloud security papers and surveys to find problems. A hybrid cryptographic scheme, Hybrid Elliptic curve Cryptography-Triple Data encryption Standard (ECC-TDES), has been proposed so as to increase the data security over cloud. Through these hybrid algorithms, the data not only gets added security, but it also gains immunity to certain malicious attacks that the cloud might face. For the test runs, audio, video, image and text files of size 10-100 KB were uploaded to a cloud-based web application encrypted with the proposed algorithm in order to record performance metrics like encryption time, decryption time and accuracy. The results were then put in comparison to both ECC and TDES algorithms, individually. The outcome of the comparison showed that even though ECC-TDES takes more encryption/decryption time, it manages to get the highest accuracy at 0.1 per cent error rate.

Keywords: Cloud Computing; Data security; Challenges; Cryptography; Decryption; Encryption; Hybrid; Elliptic curve; Triple DES

_____ _____

Signature of Major Advisor **Signature of the Student**

ਖੋਜ ਦਾ ਸਿਰਲੇਖ	:	ਕਲਾਉਡ ਅਧਾਰਤ ਸੇਵਾਵਾਂ ਵਿੱਚ ਸੁਰੱਖਿਆ ਦੇ ਮੁੱਦੇ ਅਤੇ ਉਨ੍ਹਾਂ ਦੇ ਮਤੇ
ਵਿਦਿਆਰਥੀ ਦਾ ਨਾਂ	:	ਸਿਮਰਨਜੀਤ ਕੌਰ
ਅਤੇ ਦਾਖ਼ਲਾ ਨੰਬਰ		(ਐੱਲ-2018-ਏਈ-206-ਐੱਮ)
ਪ੍ਰਮੁੱਖ ਵਿਸ਼ਾ	:	ਕੰਪਿਊਟਰ ਸਾਇੰਸ ਐਂਡ ਇੰਜੀਨੀਅਰਿੰਗ
ਸਹਿਯੋਗੀ ਵਿਸ਼ਾ	:	ਇੰਨਫਰਮੇਸ਼ਨ ਟੈਕਨਾਲੋਜੀ
ਮੁੱਖ ਸਲਾਹਕਾਰ ਦਾ ਨਾਂ	:	ਡਾ. ਲੋਕੇਸ਼ ਜੈਨ
ਅਤੇ ਅਹੁੱਦਾ		ਐਸੋਸੀਏਟ ਪ੍ਰੋਫੈਸਰ
ਡਿਗਰੀ	:	ਐੱਮ.ਐੱਸਸੀ.
ਡਿਗਰੀ ਨਾਲ ਸਨਮਾਨਿਤ ਕਰਨ ਦਾ ਸਾਲ	:	2020
ਖੋਜ ਪੱਤਰ ਵਿੱਚ ਕੁੱਲ ਪੰਨੇ	:	45 + ਅੰਤਿਕਾਵਾਂ (xxxv) + ਵੀਟਾ
ਯੂਨੀਵਰਸਿਟੀ ਦਾ ਨਾਮ	:	ਪੰਜਾਬ ਖੇਤੀਬਾੜੀ ਯੂਨੀਵਰਸਿਟੀ, ਲੁਧਿਆਣਾ-141 004 ਪੰਜਾਬ, ਭਾਰਤ ।

ਸਾਰ-ਅੰਸ਼

ਕਲਾਉਡ ਟੈਕਨੋਲੋਜੀ ਦੀ ਵਰਤੋਂ ਇਨ੍ਹਾਂ ਦਿਨਾਂ ਵਿੱਚ ਬਹੁਤ ਜ਼ਿਆਦਾ ਹੈ ਅਤੇ ਇਹ ਬਹੁਤ ਸਾਰੇ ਕਾਰੋਬਾਰ ਦੇ ਰਿਕਾਰਡਾਂ ਅਤੇ ਪ੍ਰਕਿਰਿਆਵਾਂ ਨੂੰ ਸੰਭਾਲਣ ਦੇ ਸਮਰੱਥ ਹੈ। ਇਹ ਨਾ ਸਿਰਫ ਬਹੁਤ ਘੱਟ ਕੀਮਤ 'ਤੇ ਡੇਟਾ ਨੂੰ ਸਟੋਰ ਕਰਨ ਅਤੇ ਸੰਭਾਲਣ ਨਾਲ ਆਉਣ ਵਾਲੀ ਲਾਗਤ ਨੂੰ ਘਟਾਉਂਦਾ ਹੈ, ਬਲਕਿ ਉਪਭੋਗਤਾਵਾਂ ਨੂੰ ਸਿਰਫ ਇੰਟਰਨੈਟ ਨਾਲ ਜੁੜ ਕੇ ਕਿਸੇ ਵੀ ਪਲ ਆਪਣੇ ਡੇਟਾ ਨਾਲ ਜੁੜੇ ਰਹਿਣ ਦੇ ਯੋਗ ਕਰਦਾ ਹੈ। ਹਾਲਾਂਕਿ ਇਸ ਨਾਲ ਜੋੜਿਆ ਗਿਆ ਫਾਇਦਾ ਬਹੁਤ ਸਾਰੇ ਲੋਕਾਂ ਨੂੰ ਆਪਣੇ ਡੇਟਾ ਨੂੰ ਕਲਾਉਡ ਵਿੱਚ ਸਟੋਰ ਕਰਨ ਲਈ ਉਤਸ਼ਾਹਤ ਕਰਦਾ ਹੈ, ਪਰ ਕਲਾਉਡ ਅਜੇ ਵੀ ਬਹੁਤ ਸਾਰੇ ਮੁੱਦਿਆਂ ਨੂੰ ਮਿਟਾਉਣ ਲਈ ਡੂੰਘੇ ਵਿਸ਼ਲੇਸ਼ਣ ਦੀ ਜ਼ਰੂਰਤ ਹੈ। ਇਹ ਖੋਜ ਸਮੱਸਿਆਵਾਂ ਦਾ ਪਤਾ ਲਗਾਉਣ ਲਈ ਕਲਾਉਡ ਸੁਰੱਖਿਆ ਕਾਗਜ਼ਾਂ ਅਤੇ ਸਰਵੇਖਣਾਂ ਵਿਚੋਂ ਲੰਘ ਰਹੀ ਹੈ। ਇਕ ਹਾਈਬ੍ਰਿਡ ਕ੍ਰਿਪਟੋਗ੍ਰਾਫਿਕ ਯੋਜਨਾ, ਹਾਈਬ੍ਰਿਡ ਐਲਪਟਿਕ ਕਰਵ ਕ੍ਰਿਪਟੋਗ੍ਰਾਫੀ-ਟ੍ਰਿਪਲ ਡਾਟਾ ਐਨਕ੍ਰਿਪਸ਼ਨ ਸਟੈਂਡਰਡ (ਈ.ਸੀ.ਸੀ.-ਟੀ.ਡੀ.ਈ.ਐੱਸ.) ਨੂੰ ਪ੍ਰਸਤਾਵਿਤ ਕੀਤਾ ਗਿਆ ਹੈ ਤਾਂ ਜੋ ਕਲਾਉਡ ਉੱਤੇ ਡਾਟਾ ਸੁਰੱਖਿਆ ਵਧਾਈ ਜਾ ਸਕੇ। ਇਹ ਹਾਈਬ੍ਰਿਡ ਐਲਗੋਰਿਦਮ ਨਾ ਸਿਰਫ ਡਾਟਾ ਸੁਰੱਖਿਆ ਨੂੰ ਵਧਾਉਂਦੇ ਹਨ ਬਲਕਿ ਕੁਝ ਖਤਰਨਾਕ ਹਮਲਿਆਂ ਤੋਂ ਕਲਾਉਡ ਨੂੰ ਛੋਟ ਵੀ ਪ੍ਰਦਾਨ ਕਰਦੇ ਹਨ। ਟੈਸਟ ਰਨਜ਼ ਲਈ, ਆਡੀਓ, ਵੀਡੀਓ, ਚਿੱਤਰ ਅਤੇ ਟੈਕਸਟ ਫਾਈਲਾਂ 10-100 KB ਆਕਾਰ ਦੀਆਂ ਕਲਾਉਡ-ਅਧਾਰਿਤ ਵੈੱਬ ਐਪਲੀਕੇਸ਼ਨ ਤੇ ਅਪਲੋਡ ਕੀਤੀਆਂ ਗਈਆਂ ਸਨ, ਜਿਸ ਨਾਲ ਇਨਕ੍ਰਿਪਸ਼ਨ ਸਮਾਂ, ਡਿਕ੍ਰਿਪਸ਼ਨ ਸਮਾਂ ਅਤੇ ਸ਼ੁੱਧਤਾ ਵਰਗੇ ਪ੍ਰਦਰਸ਼ਨ ਮੈਟ੍ਰਿਕਸ ਨੂੰ ਰਿਕਾਰਡ ਕਰਨ ਲਈ ਪ੍ਰਸਤਾਵਿਤ ਐਲਗੋਰਿਦਮ ਨਾਲ ਐਨਕ੍ਰਿਪਟ ਕੀਤਾ ਗਿਆ ਸੀ। ਫਿਰ ਨਤੀਜੇ ਈਸੀਸੀ ਅਤੇ ਟੀਡੀਈਐਸ ਐਲਗੋਰਿਦਮ ਦੋਵਾਂ ਦੀ ਤੁਲਨਾ ਵਿਚ ਵੱਖਰੇ ਤੌਰ 'ਤੇ ਪਾਏ ਗਏ। ਤੁਲਨਾ ਦੇ ਨਤੀਜੇ ਨੇ ਦਿਖਾਇਆ ਕਿ ਹਾਲਾਂਕਿ ਈ.ਸੀ.ਸੀ.-ਟੀ.ਡੀ.ਈ.ਐੱਸ. ਵਧੇਰੇ ਐਨਕ੍ਰਿਪਸ਼ਨ / ਡਿਕ੍ਰਿਪਸ਼ਨ ਸਮਾਂ ਲੈਂਦਾ ਹੈ, ਪਰ ਇਹ 0.01 ਪ੍ਰਤੀਸ਼ਤ ਗਲਤੀ ਦਰ ਤੇ ਸਭ ਤੋਂ ਵੱਧ ਸ਼ੁੱਧਤਾ ਪ੍ਰਾਪਤ ਕਰਦਾ ਹੈ।

ਮੁੱਖ ਸ਼ਬਦ: ਕਲਾਉਡ ਕੰਪਿਊਟਿੰਗ, ਡਾਟਾ ਸੁਰੱਖਿਆ, ਚੁਣੌਤੀਆਂ, ਕ੍ਰਿਪਟੋਗ੍ਰਾਫੀ, ਡਿਕ੍ਰਿਪਸ਼ਨ, ਐਨਕ੍ਰਿਪਸ਼ਨ, ਹਾਈਬ੍ਰਿਡ, ਐਲਪਟਿਕ ਕਰਵ, ਟ੍ਰਿਪਲ ਡੀ.ਈ.ਐੱਸ. ।

ਮੁੱਖ ਸਲਾਹਕਾਰ ਦੇ ਹਸਤਾਖਰ **ਵਿੱਦਿਆਰਥੀ ਦੇ ਹਸਤਾਖਰ**

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AES	:	Advanced Encryption Standard
AI	:	Artificial Intelligence
API	:	Application Programming Interface
ASP.NET	:	Active Server Pages dot Network Enabled Technology
CERT	:	Computer emergency response team
CSA	:	Cloud Security Alliance
CSS3	:	Cascading Style Sheets 3
CSP	:	Cloud Service Provider
DES	:	Data Encryption Standard
DKGA-CS	:	Dynamic Keys Generator Algorithm in Cloud System
DNSSEC	:	Domain Name System Security
DoS	:	Denial-of-service
DSA	:	Digital Signature Algorithm
ECC	:	Elliptic Curve Cryptography
ECDH	:	Elliptic-curve Diffie-Hellman
ECDSA	:	Elliptic Curve Digital Signature Algorithm
GUI	:	Graphical User Interface
HTML	:	Hypertext Markup Language
HTTPS	:	Hypertext Transfer Protocol Secure
IAM	:	Inadequate Access Management
IBM	:	International Business Machines
IDE	:	Integrated Development Environment
IDEA	:	International Data Encryption Algorithm
IEEE	:	Institute of Electrical and Electronics Engineers
IoT	:	Internet of Things
IT	:	Information Technology
MD5	:	Message Digest 5
MIP	:	Million instructions per second
.NET	:	Network Enabled Technology

NIST	:	National Institute of Standards and Technology
PHP	:	Hypertext Preprocessor
RSA	:	Rivest Shamir Adleman
SASL	:	Simple Authentication and Security Layer
SSL	:	Secure Socket Layer
SHA	:	Secure Hash Algorithm
SQL	:	Structured Query Language
3DES/TDES	:	Triple Data Encryption Standard
TLS	:	Transport Layer Security
VB.NET	:	Visual Basic dot Network Enabled Technology
VPN	:	Virtual Private Network
XML	:	Extensible Markup Language

CHAPTER-I

INTRODUCTION

From the past few years, Cloud Computing (or simply *Cloud)* has become a popular argot in the IT world. Cloud computing can be defined simply in the terms of a facility provided by a service provider. The consumers of the facility do not need to be aware of or worried about how the delivery or maintenance of services (for example storage, network, infrastructure, application) is done. Instead, the consumer is only concerned that the service is available whenever it is required.

The United States of America's National Institute of Standards and Technology (NIST) defines cloud computing as: "…a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" (Grance and Mell 2011).

The founding idea for development of cloud is separation of operating system from overlying physical hardware, with the help of virtualization and automation techniques (Gandhi and Kaushik 2016). Based on usage and consumer requirements, the services provided by a cloud can be categorized as shown in the figure below;

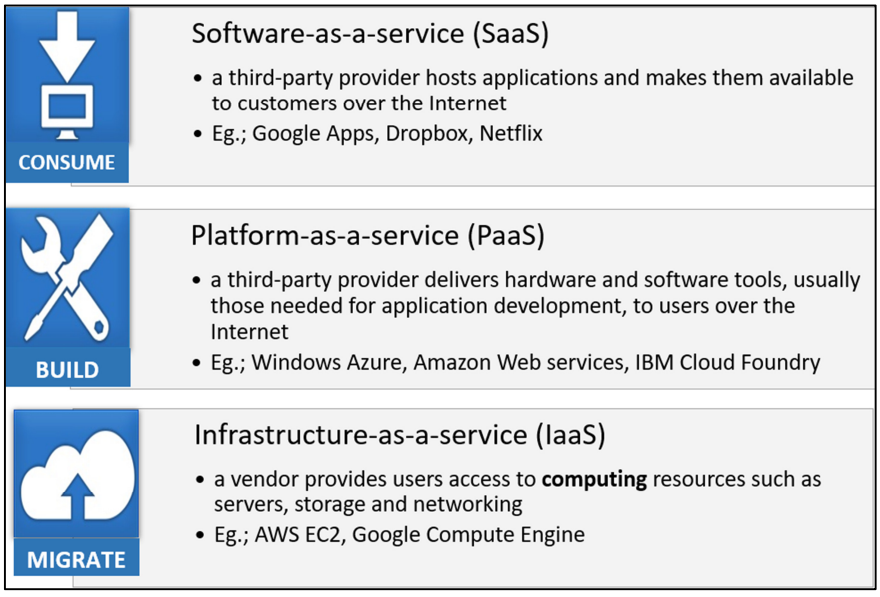

Fig. 1.1 Cloud Service Models

Now that the service models have been discussed, clouds are also categorized on the basis of deployment models as listed below (Rizvi an d Tandra 2014*)*:

Public cloud

Hosted at the service provider's site, public cloud aids to multiple consumers by sharing infrastructure, through internet connectivity. Public cloud can only be deployed for insensitive data but is less expensive as compared to private cloud. The drawback of public cloud is that all of the data exists beyond the organisation's firewall.

- **Private cloud**

Dedicated to solo consumers, the private cloud is best suitable for the confidential data, which requires high level of security. The organisation using a private cloud may or may not choose to deploy the cloud on its premises. Based on this the private cloud has two implementations;

i. In-house private cloud: requires buying, developing and maintaining the cloud on organisation's own infrastructure.

ii. Hosted private cloud: hosted on cloud service provider's own site. It doesn't utilize a shared infrastructure and managed for a single customer. Also, the network connection between the client and service provider is established over a private network or tunnel.

- **Hybrid cloud**

  Developed by combining features of private and public cloud, hybrid cloud aims at catering to the consumers with most fitting solutions. The business-critical data can be secured over a private cloud while general information can be moved to public cloud.

- **Community cloud**

  This type of cloud aids multiple consumer organisations that share common interests or computing concerns. A consumer using community cloud can experience the essence of a public cloud, while their data being secured by privacy features of a private cloud

The term cloud computing on being split up, gives two words; "Cloud" denoting internet with enormous databases built up of applications and other virtual resources; and "Computing" taking into account the use of computer-based operations (Gupta *et al* 2019).The cloud computing approach alleviates the needs of the companies to have their own data centres by circumventing the purchase and management costs for storage, hardware, power supplies or other infrastructure. As a result, the application of cloud computing supports companies to quickly build a new business and work efficiently. "Cloud" can communicate everything (as administrator) over the web in view of the client request, for example operating system, organizing equipment, storage, assets, and software (Ilayaraja *et al* 2017). This can be held up both as a strength and a weakness, as explained further.

Technology, growing at a fast pace, has made numerous online services feasible including e-mail, e-banking, e-billing etc. For these services to work, the consumer's data also needs to be online, this data may comprise of confidential information (for example bank

account details, health insurance, loan details etc.) which, if divulged, can prompt a serious loss to the consumer. With such a large amount of consumer's workload moving to the cloud, security and privacy are the testing factors for data sharing efficacy of the cloud and a major concern for the cloud service providers (Alharthi *et al* 2015). To improve the data sharing efficacy, the cloud service providers need to deploy diverse strategies and mechanisms conforming to form and size of data (Alassafi *et al* 2016).

It is apparent that cloud service supplier is dealing with the client data. As with any remote storage system, there are three principal aspects of data security in cloud storage, namely, Confidentiality, Integrity and Availability (CIA). Ensuring confidentiality means protecting consumer's data from unauthorized access i.e. no one can read the data other than the authorized consumer; integrity ensures that no changes to the data can be done without the modifications being detected; and availability means that the authorized consumer can access his/her data at any time. Cloud computing, just as any other service, also faces security risk/threats. Some of the potential security attacks on cloud include authentication attacks, denial of service (DoS) attacks, wrapping attacks, flooding attacks, browser attacks, malware-injection attacks and man-in-the-middle (Rizvi and Tandra 2014*)*. These attacks can destabilize any of the three principle aspects of cloud security. Furthermore, data also needs to be protected while transferring and storing phases in cloud storage servers.

The term "cloud storage" primarily refers to an online space i.e. a remotely located storage. This storage can be in the form of magnetic tape drives, USB flash drives, compact discs etc. In the cloud environment, the user can back up data on a server hosted by cloud service provider. This kind of online storage facility uses a network of virtual storage servers and includes maintenance tools for the virtual storage space. The online storage facility makes it easy for users to reach the data using an internet connection and a user interface facilitated by the cloud storage vendor. Cloud storage spread over multiple locations, offers a data abstraction which makes it safer as compared to data stored in single physical place (which has a risk of being damaged or lost).

Reliability and security have been two major concerns regarding data. The storage provider needs to be trustworthy and reliable while also protecting the data. Different techniques have been developed for secure data while it is in transit or at rest, encryption being one among them. The technique applied to achieve encryption of the data is called cryptography. It is also referred to as code making/generation. It involves using a key to convert the original data into a form incomprehensible to an outsider. In order to access the information, the end user requires the appropriate key to decrypt the file. The reliability is increased by using authentication, achieved by demanding appropriate credentials from the

user every time they wish to access the services. The building blocks for a cryptographic scheme are;

- Key generation: producing keys to encode the original data.
- Encryption unit: using keys to convert plaintext to ciphertext.
- Decryption unit: extracting original data from ciphertext using keys.

The key generation performs the most vital task, generating key for cryptography using random bits and a key derived from passwords, thus defining the strength of encryption scheme (Kumar and Ragupathy 2016). The original message is denoted as plaintext while the encrypted message is ciphertext. Key selection being an important part is used to categorize cryptographic techniques as listed below (Stallings 2005);

1. **Symmetric Key Cryptography:** private key cryptography, uses same key to perform encryption and decryption. Symmetric encryption transforms plaintext into ciphertext using a secret key and an encryption algorithm. Using the same key and a decryption algorithm, the plaintext is recovered from the ciphertext. DES, AES, Carlisle Adams and Stafford Tavares (CAST) algorithm, Blowfish, Two-fish, International Data Encryption Algorithm (IDEA), and Secure and Fast Encryption Routine (SAFER) are some examples of symmetric encryption.
2. **Asymmetric Key Cryptography:** encryption and decryption use separate keys, a public key and a private key respectively. Also identified by public-key encryption, it ensures confidentiality, authentication, or both for the data. Some examples are RSA, DSA, El-gama and ECC.
3. **Hybrid Cryptography Scheme:** developed by incorporating strengths of private key like fast and easy processing, and advantages of public-key encryption like security as they do not require the sender and receiver to share a common secret in order to communicate securely. A hybrid cryptosystem can be constructed using any two separate cryptographic algorithms:
    - a public-key cryptosystem based key encapsulation scheme.
    - a data encapsulation scheme based on a symmetric-key cryptosystem.

As a result of the incessant extension of cloud computing, more and more consumption will be introduced. In the meantime, the challenge from security issues is unparalleled, which requires the continual exploration of IT and information security experts. Hence, CSA and other organisations that support the security of cloud computing were established to aid in the best utilisation of cloud computing resources and provide compact protection to it (Feng *et al* 2011). It is related to technical as well as to the facets connected with standardization, supervision model, law and regulations. So, solving cloud computing

security from technical view is not sufficient, it also requires the support from academia in information security, industries and departments of government (Ou 2015).

In this research, the emphasis has been laid on identifying different security issues found in cloud services and techniques deployed for eliminating them. Efforts have been made to apply some of these techniques to a web application running in cloud environment. The web application namely "Secure Cloud Computing" has been planned to be delivered under Software-as-a-service (SaaS) model of cloud. The major goal is to provide data security. Network security, physical security, legal compliance, disaster or risk management are not in the scope of this thesis. The motivation for the recommended security measure has been derived from previously done research work by other researchers.

1.1 Knowledge Gap

Cloud security issues have evolved immensely in the last few years and require an up to-date insight into its impact on the performance of the cloud so that proper utilisation of solutions conforming to particular application can be done.

1.2 Objectives

1. To study the security issues and techniques in cloud-based services.
2. Analysis of these security techniques for computational applications.

CHAPTER-II

REVIEW OF LITERATURE

The features held by cloud computing attracts researchers to develop optimized frameworks by taking data security, virtualizing, concurrency control etc. into consideration. Researchers all over the world have made and are making efforts to improve and introduce more comprehensive cloud computing models.

For planning a secure cloud environment, the vulnerabilities of cloud need to be deliberated. So, the recent research works have been audited concentrating on security issues, solutions, and difficulties in cloud computing infrastructure.

Over the past few years, a rapid progress has been observed in cloud computing. With the increasing number of companies opting for cloud-based services, it is vital to protect the data of various consumers of centralized resources. Some major challenges being faced by cloud computing experts include securing, protecting and processing the data which is the property of the consumer. For every new outcome of research, the previous research has to be reviewed as follows:

Maddineni and Ragi (2011) describe two instances of how data is being stored over cloud and their vulnerabilities, these scenarios are;

1. When a local network is moved to cloud, a part of its data is placed in the cloud, while critical data still resides in the local network. The cloud provider doesn't have any authority over this data but it may require access to some information from it. This event of access holds high chances of instigating unauthorised access to the local network of resources. The stream of information travelling from the client's network to the cloud network can face passive and active attacks. The passive attacks include traffic analysis while active attacks are masquerading, replay attack, message alteration and DoS.
2. When whole of the data is shifted from local network to the cloud, so that local network and approved users can physically access data in the cloud. For this purpose, users of the cloud are provided with virtual machines accessible with valid logins. Since, these logins are easy to be cracked, the possibility for unauthorized to enter the cloud and access the data is high.

The authors also conducted surveys to find major security challenges expected for the future of cloud computing. The results based on experts' opinions reveal following challenges: Eavesdropping; Hypervisor viruses; Legal Interception point; Abuse and nefarious use of Cloud Computing; Insecure application programming interfaces; Trusted

transaction; Risk of multiple Cloud tenants; Smart phone data slinging; Malicious insiders; Virtual machine security; Shared technology vulnerabilities; and Service and traffic hijacking. These security issues stall the three pillars of cloud confidentiality, integrity, and availability plus compromise security. The countermeasures identified from the survey to ensure security are identity-based authentication; RSA algorithm and other encryption techniques; Multitenancy-based access control model; Dynamic intrusion detection system; TLS Handshake; Diffie-Hellman key exchange; Data coloring and software water marking techniques; proxy re-encryption; time bound ticket based mutual authentication scheme; and identity management.

Sandha *et al* (2011) conducted their research with the goal to provide fair performance comparison of various encryption techniques with different text file sizes. The algorithms selected for evaluation were DES, 3DES, AES and Blowfish. The presented simulation results exhibit Blowfish's best throughput followed by AES. 3DES was found to be least efficient of all the algorithms.

Dhiman *et al* (2014) quote that the rapid development of telecom technology and the limitation for transmission of data, require an efficient cryptographic algorithm with more speed, less time and increased throughput. They evaluated DES, TDES, AES, Blowfish and Twofish encryption algorithms on the basis of space complexity. The results suggest that TDES is most efficient among other algorithms. DES may need less space than TDES, but it can be cracked with 2^{56} combinations of brute force attack.

The research paper written by Kaul and Shaikh (2014) credits AES and ECC as the best two symmetric and asymmetric encryption algorithms. Based on this the authors have devised a hybrid cryptography model with combination of AES and Blowfish. With this they have used Message Digest 5 for data integrity, ECDH for key exchange and ECDSA for digital signature. The performance of the system was evaluated on the basis of encryption/decryption time, throughput and memory usage for text, image, audio and video file formats. The experimental results show that Blowfish gave highest throughput compared to other individual algorithms. The hybridised system of AES and Blowfish has characteristics of both algorithms and is strong against the same vulnerabilities the parent algorithms are immune to. Also, they stated that ECC is the best asymmetric encryption technique and provides the highest strength-per-bit of any cryptosystem, resulting in faster computations, lower power consumption and memory. It also provides a method for obtaining high-speed, effective and scalable application of protocols for authentication and key agreement.

Rizvi and Tandra (2014) developed a cloud architecture for "ifoodbag" web service, taking into consideration various aspects of cloud security. In the thesis, identified areas with security issues and recommendations to resolve them have been summarised as below;

1.	Load Balancer	:	Use SQUID proxy server on a dedicated machine and avoid running it as a root user. Instead create a separate sandbox.
2.	Browsing web application	:	HTTPS for secure browsing
3.	Client-server authentication	:	Publicized data in the website can be browsed using server authentication. For logging-in registered users and allowing financial transactions, use strong two-factor authentication.
4.	Communication between nodes	:	Deploy VPN tunnels between nodes in the same or different tiers to secure their communication.
5.	Name resolving	:	Use DNSSEC and authoritative-only name server software.
6.	Distributed database / Storage data protection	:	Use AES algorithm for encrypting/decrypting the stored data.
7.	Memcached	:	Use firewall or Memcached with SASL

Further laying emphasis on threats to cloud storage, the authors expressed that reliability and security are biggest concerns. The complex data security challenges as enlisted are;

- Protect confidential business or regulatory data.
- Shared cloud service among multiple clients
- Compliance with data movement and legal issues as per privacy norms set by government.
- Lack of standards instructing CSPs on how to erase existing data and recycle disk space.
- Auditing, reporting and agreement concerns.
- Inclusion of third party who isn't part of the company but has visibility and access to the data.

A combination of Encryption (using different cryptographic algorithms), Authentication (providing user name and password) and Authorization techniques has been suggested in order to maintain and secure unprotected data. They used and compared TDES and AES cryptographic algorithm to encrypt data for "ifoodbag". The observations made acclaim AES as the better algorithm.

Ou (2015) has discussed the typical utilization of cloud, primary security issues involved and schemes to solve them. The utilizations as acknowledged by the researcher are IoT, cloud storage, cloud computing platform, cloud gaming, security as a service and Big Data. And the main security issues identified are;

- Privacy management:

  With the multi-tenancy feature of cloud, it is inevitable to have privacy issues. Most instances of privacy management found in cloud, emphasize the use of cloud server equipped with management component. The new type of privacy manager involved, is based on the users' terms and trust models. The cloud service provider assists users to control their own sensitive information by the means of public key encryption.

- Data security and confidentiality:

  Since users can't have full control over their data once it is uploaded to cloud, it is crucial that the cloud service provider offers effective security guidelines. Cloud service models like IaaS and PaaS are provided through and accessed by web browser. The protocols linked with network application layer in data transmission are carried by XML. Evidences have been found that XML has a connection with attacks carried on XML signature and other security issues in web browsers. Thus, XML encryption in the core code of the web browser should be enforced with transmission layer security technology.

- Data audit:

  As said before users' data moving to cloud means its responsibility is with the cloud service provider, so it's important that the data is processed and stored properly. This integrity verification requires a scheme to audit data remotely. Some examples are; RSA-based homomorphic tag to implement verifiable data control; homomorphic authenticator equipped with random masking to protect privacy etc.

- Authentication and access control policy:

  The clients using the cloud services should be authenticated by the service providers. Additionally, different CSPs should also be able to verify each other and use access control policies to control accessibility of cloud data and services. One certain policy suggested by the author is Proxy re-encryption (PRE).

- Virtual machine security and automated management:

  Virtualization is the building block of cloud computing as it helps in separation of physical hardware from underlying operating system. This means the virtual machine surveillance should be trustable and by no means should refer to the users' private information. This brings forward the concept of VMware Cloud (VMC). In addition to

automated control and management of virtual machines in huge cloud data centres, VMC also demonstrates a path for executing virtual network access control, disaster recovery and virtual machine detection.

Stallings (2017) has discussed cloud security risks recorded from CSA's the top cloud-specific security threats report from 2010. These risks include; Abuse and nefarious use of cloud computing, Insecure interfaces and APIs; Malicious insiders; Shared technology issues; Data loss or leakage; Account or service hijacking; and Unknown risk profile. Focusing on data protection element of cloud security the author quotes that, "Data must be secured while at rest, in transit, and in use, and access to the data must be controlled". Mediating over different ways to apply encryption, Stallings (2017) writes that if the client encrypts data in transit, the CSP will be involved through key management responsibilities. Another alternative is to enforce access control, but again CSP may be involved to some extent depending on the cloud service model. So, the only viable option for the client is to encrypt the data at rest i.e. encrypted database. This way CSP will have no access to the encryption key and eventually no access to the data. Only a few algorithms including RSA, ECC and Diffie-Hellman have received widespread acceptance in the several decades since the concept of public key cryptography was proposed. The author has analysed several approaches to encryption/decryption using elliptic curve cryptography and compared it with various algorithms in terms of key sizes and computational effort for cryptanalysis. RSA is widely preferred public key cryptography for encryption and digital signatures, increasing its key length over time. This has put a significantly heavier processing overhead on the applications. This load has consequences for e-commerce sites conducting large number of secure transactions. Here, ECC beats RSA by offering equal security for a far smaller key size. Also, ECC is showing up in standardization efforts, including the IEEE P1363 Standard for Public-Key Cryptography.

Anonymous (2018) observed that some threats are encountered due to inaccuracies that occur because of human error (which cannot be avoided). Data, if not backed up properly, can be lost even when actions are not malicious. As analysed above, security risks are inevitable, but fortunately effective solutions have been and can be implemented and tested easily with satisfactory results. Key management and data encryption (cryptography) has been observed to significantly cut the prospect of a data breach to 40% on average, in the past few years.

Brandão (2018) has identified main challenges to cloud security as follows;

- Authentication: The multi-tenancy feature makes the data stored by a user over a cloud available to all other users.

- Access Control: Cloud computing needs to have strong access control policies. This needs to be supervised to ensure the access is given to verified users only.
- Policy Integration: The cloud service providers and the consumers need to have similar outlooks, establishing safe work flow in cloud.
- Services Administration: Google and Amazon like eminent cloud service providers, sign agreements subjected to information sharing. This makes their businesses grow. Users must be provided with easy and safe localized services, allowing them to handle each service independently, keeping administrations of clouds separated.
- Data Security: The security of data uploaded by cloud user is the most crucial challenge and stands on three pillars of confidentiality, authentication and integrity.

Organisations in every industry have been impacted by data loss. As per the reports compiled by Carr (2018), 55% of the general business sector was affected; attacked areas accounted for 23% of the health care industry; 10% of education sector was breached; while 6% of both financial services industry and military/government sectors faced security threats. Once data thieves gain access to a system, not even the most cutting-edge cloud security can safe-guard the system from data theft. Unfortunately, unauthorized access or IAM has become a significant issue. Industries of every scale demonstrate "a lack of scalable identity, access management systems, weak password use, failure to use multi-factor authentication and a lack of ongoing automated rotation of cryptographic keys, password and certificates," making IAM among the top five cloud security issues identified in 2018.

Hameed *et al* (2018) analysed that transmission and reception of data hold high risks of data loss, hacking and data modification. As a solution, the research team designed and implemented Dynamic Keys Generator Algorithm in Cloud System (DKGA-CS) using interconnected languages of HTML5, JavaScript, PHP, CSS3 and MySQL database. The proposed algorithm produces dynamic keys by the means of coding and permutations. The polynomial equations used in algorithm expand the original key. The algorithm holds the ability to manage clients creating secret keys of variable length and tamper the key subsequently over a period of use. The proposed block cipher algorithm involves use of cipher algorithms, artificial intelligence and logical circuits to protect the data exchanged between clients, hence increasing complexity and security of framework.

Kumar and Singh (2018) did analysis of cryptographic algorithms based on their key size, block size, rounds and many more parameters. The purpose of cryptography is explained

in four parameters of confidentiality, integrity, authentication and access control. The comparison results are as recorded in table 2.1 below. The conclusion made at end is that TDES (or 3DES) algorithm is best for security of data as it uses three keys to encrypt and decrypt. Moreover, the key size of TDES is much stronger to crack and so is recommended for use in better security of data.

Table 2.1 Comparison of cryptographic algorithms.

Algorithm / **Parameter**	**DES**	**TDES**	**AES**	**Blowfish**	**RSA**	**ECC**
Key type	Symm	Symm	Symm	Symm	Asymm	Asymm
Key size (bits)	56	168	128, 192, 256	32-448	1024	168
Block Size (bits)	64	64	128	64	512	64
Scalability	Scalable	Not scalable	Not scalable	Scalable	Not scalable	Scalable
Speed	Moderate	Slower	Faster	Faster	Slower	Faster
Security Levels	Not secure enough	Excellent	Excellent	Least	Least	Average
Vulnerabilities	Brute Force, Linear and differential Cryptanalysis-sis attack	Meet-in-the-middle-attack	Brute Force Attack	Birthday attack	Brute force and Oracle attack	Brute Force attack

In a report sponsored by IBM Security, 507 companies that experienced breach in 2019 were recruited and more than 3211 individuals, who are knowledgeable about the incidents, were interviewed (Anonymous 2019). The survey was done based on two factors:

- Number of customer records lost or stolen.
- Percentage of customer base lost following the data breach incident.

The key findings were that the difference from 2014 to 2019 represents a 31 percent growth rate in the likelihood of experiencing a breach within two years. Malicious attacks comprising of Distributed DoS, hijacking attacks, malware injections, phishing/ social engineering and SQL injection, contributed to 51 percent of root causes of data breach. Human error contributed to 24% and system glitches caused 25% of the data breaches. The

calculated average total cost of a data breach was $3.92 million which can be mitigated by $360,000, with an extensive use of encryption techniques.

Babu *et al* (2019) have developed a hybrid encryption algorithm to enhance cloud security. Addressing the cloud security issues they have stated that privacy and security are major issues in cloud adoption. Cloud storage and security have an interdependent relation thus enhancement of security requires an effective storage technique. The CSP must assure that its infrastructure is secure enough to safeguard clients' data. Encryption plays a major role in cloud security field. The data must be encrypted before its transmitted to the cloud. For this purpose, several encryption algorithms like AES, DES, RSA and MD5 are available. The author has used homomorphic algorithms and Blowfish algorithm to perform multi-level encryption on the data.

In the security model proposed by Gupta *et al* (2019), multi-layered encryption is brought into play to enhance data privacy in cloud computing. The algorithms used are AES and RSA for first and second level of encryption respectively. Separate keys are required for encryption and decryption in order to view data, which ultimately makes it impossible for an unauthorized user to access data without valid keys. They also did comparisons between AES and DES, which revealed that AES works 6 times faster than DES.

Kaushik and Patel (2019), have concluded from their research that using hybrid symmetric encryption produces more complex cipher text, which makes the data immune to brute force attack. In the proposed model, the authors have combined transposition and substitution, as primary and secondary cryptographic techniques respectively. The data is encrypted before being stored over the cloud. Both of the cryptographic algorithms mentioned use symmetric key cryptography to encrypt/decrypt plain text. Further recording encryption and decryption times, they found that hybrid of AES and Two-fish cryptography algorithms works more efficiently than hybrid of AES and Blowfish algorithms. So, as concluded hybrid of more advanced symmetric or asymmetric cryptographic techniques can be used to build high security frameworks for safeguarding cloud against malicious activities.

Anonymous (2020) offer a review of 2019's major data breach events. They detected that among the top four public cloud vulnerabilities, unauthorized access accounts for highest 42 percent. Also, the cloud data security issue inhibits cloud adoption by 25 percent. The study specified that cryptominers remained prevalent malware type in 2019. As reported, on June 6, 2019 American Medical Collection Agency (AMCA) suffered huge data breach in which attackers infiltrated their web payment portal. The healthcare and billing details of over 20 million patients was exposed. This was a huge setback as AMCA had to file bankruptcy because of financial and legal consequences faced following the breach. As of 2020 the study

predicts that increasing reliance on public cloud, increases enterprises' exposure to the risk of outages. This will drive them to reconsider their existing cloud approach and adopt hybrid environments comprising both private and public clouds.

Cloud Security Alliance (CSA) releases the Top Threats report annually to raise awareness about threats, risks and vulnerabilities in the cloud. As per the 2019 comprehensive report compiled by the CSA, the top threats to cloud security can be broadly identified as (Brook 2019);

1. Data Breaches
2. Insecure interfaces and APIs
3. Lack of cloud security architecture and strategy
4. Insufficient Identity, Credential, Access and Key Management
5. Weak control panel
6. Metastructure and Applistructure failures
7. Misconfiguration and Inadequate change control
8. Insider threat
9. Limited Cloud Usage Visibility
10. Account hijacking
11. Abuse and Nefarious Use of Cloud Services

Overall, cloud computing is a new pattern based on the expansion, utilization and interaction of the Internet. As observed, there is a need for a suitable set of security guidelines for each web application or service to be implemented in the cloud. From the studies reviewed, the most prevalent cloud security issues over the past decade are:

1. **Insufficient Identity, Credential, Access and Key Management:** These management systems include tools and guidelines for organisations to manage, monitor and secure access to their valuable resources (like electronic documents, physical resources, server rooms).
2. **Data Breaches and Data loss:** Because of its high business value and the impact of its loss, data is becoming the major target of cyberattacks. Encryption techniques may add overhead to cloud's performance but effectively protect the data.
3. **Insecure API's:** Web and cloud services use API keys to identify the third-party applications using the services. In some cases, these APIs without any authentication applied, leave the system and data open for random access. Attackers all over the internet keep surfing around finding these doors. If service providers are not careful, an attacker with access to the key can cause a denial-of-service or rack up fees on behalf of the victim.

4. **Insider threat:** CERT's definition of an insider threat is, "the potential for an individual who has or had authorized access to an organization's assets to use their access, either maliciously or unintentionally, to act in a way that could negatively affect the organization".
5. **Account hijacking:** Phishing, exploitation of cloud-systems, or stealing credentials of highly privileged accounts are some of the attacks used to disrupt cloud environment. CSPs should vigorously promote Defence-in-depth and IAM controls play key role in mitigating account hijacking.
6. **Lack of cloud security architecture and strategy:** Enterprises of every scale are prone to suffer financial loss, legal consequences and fines; if a proper security architecture for secure moving, deploying and operating of cloud elements is not implemented.

These security issues should be considered while designing any type of cloud model.

CHAPTER-III

METHODOLOGY

3.1 Overview

Storing personal data on a cloud, even if it's a private cloud, means trusting it with third party servers. Even though CSPs claim to provide appropriate security, it's dicey to believe them. Data traveling through network or stored in the cloud storage in plain text is a serious security threat. Additionally, cloud is a shared network used by multiple organisations or individual clients, which may or may not be interlinked. This multi-tenancy makes the cloud data easily accessible through any of the connected or distributed resources. With the growing databases over time, cloud is becoming more vulnerable to attacks. This demands for better security mechanism to assure discretion of consumer's data.

Cryptography is one of the means to protect and secure the confidentiality of consumer's data from any kind of malicious attacks and unauthorized access. Unauthorized access to user's data also holds risk of hampering its integrity. But owing to the lack of key, breaking the encrypted data becomes a monotonous task for the attacker. Only the user possessing the key has the authority and capability to decrypt the received data into its original form. The users revoked raises cloud efficacy issue, deterring the private key encryption process. Rizvi and Tandra (2014) designing the cloud architecture for "ifoodbag" web service, have also suggested using combination of encryption, authorization and access control techniques in order to maintain and secure cloud data.

Traditional cryptographic algorithms are easy to break once key or logic is revealed. Thus, new encryption techniques are being discovered every day. There are three types of commonly used methods namely; symmetric cryptographic technique, asymmetric cryptography technique and hash function. Symmetric and asymmetric cryptographic schemes have been discussed in introductory chapter of this thesis. The hash function is a one-way, infeasible-to-invert function, that uses a mathematical transformation to irreversibly "encrypt" information. Examples include MD5, Whirlpool, SHA-1, SHA-2 and SHA-3.

Every cryptographic algorithm has its strengths and weaknesses. Hybrid cryptographic scheme is one way to combine and exploit the strengths of different algorithms to overcome their weaknesses (mostly the convenience of a public-key cryptosystem with the efficiency of a symmetric-key cryptosystem). This produces better security techniques in result. Lots of work have been done in hybrid technique such as AES-ECC (Gupta *et al* 2016), Fibonacci series-XOR bitwise-PN sequence (Jayakumari *et al* 2015), IDEA-RSA (Ming and Wu 2010), and DES-RSA (Akinlolu and Kazeem 2014). In this concept, one

algorithm is used to encrypt the secret key and this secret key is used with another technique to encrypt the plaintext. The hybridised algorithm inherits properties of the parent algorithms; thus, it is secure against the same attacks the parent algorithms are immune to (Cramer and Shoup 2019).

3.2 Hybrid ECC-TDES Scheme

This research proposes a new hybrid cryptosystem constructed using Elliptic Curve cryptography (ECC) for key generation and Triple DES (TDES) algorithm for data encryption (asymmetric and symmetric algorithms respectively). Secure Hash Algorithm 3 (SHA-3) used generates a unique fingerprint for each file, hence ensuring authentication. The proposed algorithm has been tested with text, audio, image and video files to find the security enhancements and performance on different file systems. The performance metrics have been recorded in the form of encryption and decryption time; throughput; and accuracy of reconstructed files.

3.2.1 ECDH Key exchange

ECC finds its major application in the cryptosystems in two forms as ECDSA and ECDH. ECDSA (Elliptic Curve Digital Signature Algorithm) is DSA (Digital Signature Algorithm) digital signature standard equivalent for elliptic curves used to provide verifiable digital signatures for data messages. Elliptic Curve Diffie-Hellman (ECDH) is a variant of Diffie-Hellman algorithm which uses elliptic curve's properties to generate key pairs for two communication participants A and B and determine the method how to exchange public keys over an insecure channel. It is standardised and free to use cryptosystem. The ECDH key exchange mechanism can be paired with symmetric encryption methods and can be adapted to different data security solutions (Magons 2016).

To understand how the proposed algorithm works, ECC key generation and the arithmetic behind it needs to be explained. As defined by Stallings (2017), a finite field $n(2^m)$ consists of 2^m elements, together with addition and multiplication operations that can be defined over polynomials.

For elliptic curves over $n(2^m)$, a cubic equation (Eq. 1) is used in which the variables and coefficients all take on values in $n(2^m)$ for some number m and in which calculations are performed using the rules of arithmetic in $n(2^m)$. Fig.3.1 shows a general representation of elliptic curve

$$y^2 = x^3 + ax + b \qquad \text{(Eq. 1)}$$

The prerequisites for an elliptic curve key generation are;

- an elliptic curve equation defined over a finite field
- a large integer q which is either a prime number or integer of the form 2^m
- elliptic curve parameters a and b satisfying Equation (1)

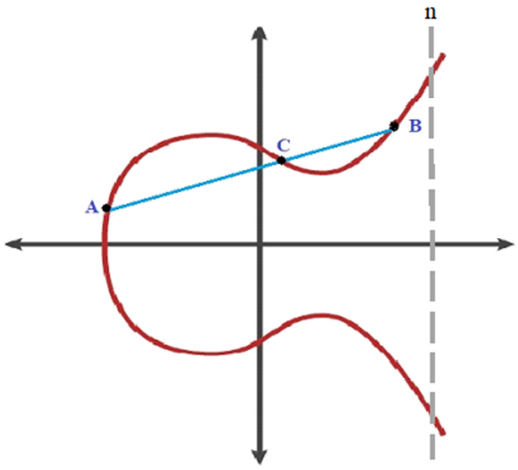

Fig. 3.1 General plot of an elliptic curve

Based in ECC key generation, ECDH key exchange algorithm can be summed up as below;

Step 1. Two global public elements are selected as:

- $E_q(a,b)$, where a and b are elliptic curve parameters for equation (1), and q is a prime number or integer of the form 2^m.
- G, the key generator point, is a point in $E_q(a,b)$ available to every participant in the communication medium.

Step 2. User A's Key Generation

a) Select a private key $n_a < n$
b) Calculate public key, $P_a = n_a \times G$

Step 3. User B's Key Generation

a) Select a private key $n_b < n$
b) Calculate public key, $P_b = n_b \times G$

Now that public keys have been generated for both the users, the process moves on to key exchange. For this both the users calculate their secret keys as:

- User A's secret, $K_a = n_a \times P_b$
- User B's secret, $K_b = n_b \times P_a$

The resulting keys will be such that K_a equals K_b (let resultant be K), hence completing the key exchange process. To break this scheme, an attacker would need to be able to compute K given G and KG, which is assumed to be hard, making the function irreversible. This trapdoor feature is called discrete logarithm problem for elliptic curves.

3.2.2 TDES Encryption

As stated before, ECDH key exchange can be paired with any symmetric encryption technique. Based on analysis carried out by Kumar and Singh (2018), TDES is the best algorithm for security of data as it uses three keys to encrypt and decrypt and the key size of TDES is much stronger to crack. So, TDES is the chosen method for encryption of the data. Pairing it with ECC, may also enhance its performance. Moreover, frequently used cryptographic techniques like AES, DSA, RSA etc. have already been paired and tested with ECC in other research works.

Triple Data Encryption Standard (TDES) is a symmetric block cipher and successor of DES. As the name suggests, TDES encrypts each block of data three times. The block size used in TDES algorithm is 64 bits. It was developed by IBM in 1978 to overcome the problems of DES (Kumar and Singh 2018). Unlike use of single key in DES, TDES can run with two or three 56-bit keys. The use of three key TDES requires a key length of $56 \times 3 = 168$ bits, which can be a bulky task. An alternative to this is to use only two keys for the encrypt-decrypt-encrypt (EDE) sequence of TDES. Moreover, ECDH gave two keys in output hence in the proposed algorithm, the public keys P_a and P_b, (say T1 and T2respectively) generated by ECC are forwarded to TDES for the encryption process following the EDE procedure as below:

1. T1 performs DES and encrypts the original file (plaintext).
2. T2 performs reverse DES on the encrypted file produced by T1.
3. T1 again performs DES to encrypt the file that was obtained from encryption done by T2.

A schematic diagram explaining DES algorithm is presented in Fig. 3.2. An input of 64 bits as plaintext produces a 64-bit output as ciphertext. Each 64-bit data block goes through 16 rounds of encryption (Kumar and Singh 2018).

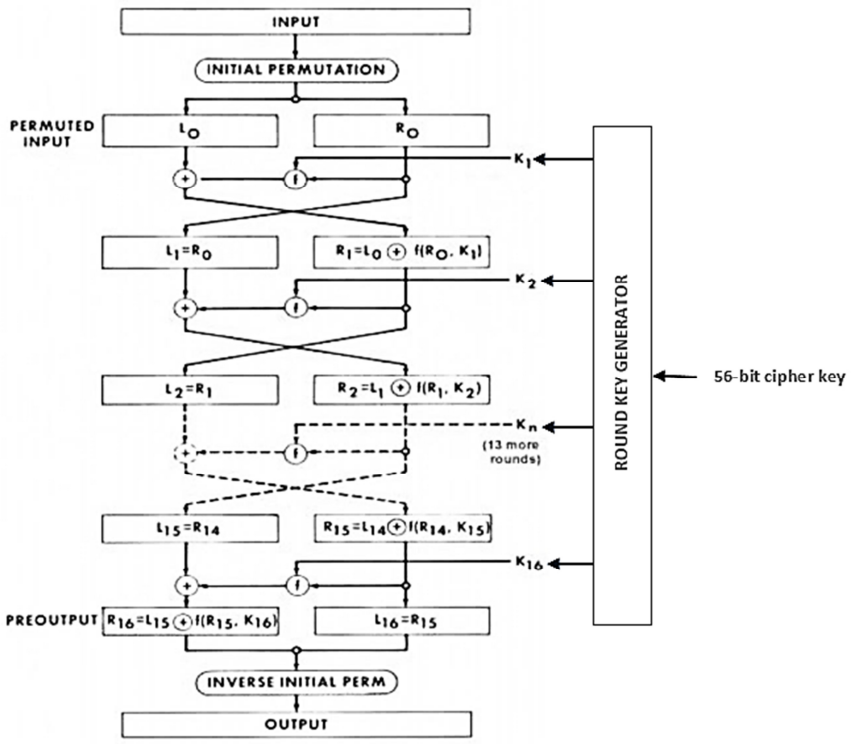

Fig. 3.2 DES algorithm

TDES has two salient features that make it widely acceptable. First feature is it's 168-bit key length which makes it invulnerable to Brute-force attack. Second feature is that the underlying encryption algorithm of TDES is same as DES. Moreover, TDES has been under scrutiny for a long period of time, and no effective cryptanalytic attack other than Brute-force has been found. As far as security is the concern, there is a high level of confidence that TDES is an apposite choice for decades to come.

Cryptography working on small key sizes is a need of the hour in a world where millions of people use individual less powerful devices like mobile phones, forming huge data reservoirs. Avoiding the cost of slower cryptography performance due to longer key lengths, ECC appears to offer a better trade-off: high security with short, fast keys. Cloudflare like organisations have been using ECC deployed with RSA, because their SSL certificate complies with RSA key pair. As predicted by cloud security experts in past, the mathematical algorithms supporting RSA and Diffie-Hellman have been broken, leaving ECC as the only reasonable alternative.

Symmetric algorithms show high performance and take less time for encryption and decryption. Comparatively, asymmetric algorithms use large-sized keys and greater computational time. This is another reason that in the implemented hybrid scheme, ECC (the

asymmetric algorithm) has been used for the process of key distribution and TDES (the symmetric algorithm) has been used to encrypt and decrypt the data.

3.3 Work flow diagram

All the keys bind with users' profiles to give high security factor while communicating in the cloud network. The key exchange mechanism has been deployed to form a hybrid form of both symmetric and asymmetric algorithms with enhanced hashing module. Fig. 3.2 represents the work flow of the proposed scheme**.**

1. File Encryption

Step 1: Upload the file and forward it for key generation.

Step 2: Perform the ECC key generation which would produce a set of public and private keys binding to user's profile and the files uploaded.

Step 3: The next step in the encryption process is forwarding the public key generated in the previous step for encrypting the file using TDES.

Step 4: The final step in encryption process of the file is to store the cipher data generated by executing the above listed steps in a database.

2. File Decryption

The decryption of a file is similar to the process of encryption. Decryption is encryption performed in reverse chronological order. The procedure followed for decryption is as mentioned below.

Step 1: Load the encrypted file bytes from the database and forward to decryption module.

Step 2: Implement the TDES algorithm using private key generated by ECC, to decrypt the file.

Step 3: The decrypted file is downloaded to the user's device. Also, it gives encryption time, decryption time and other parameters in output.

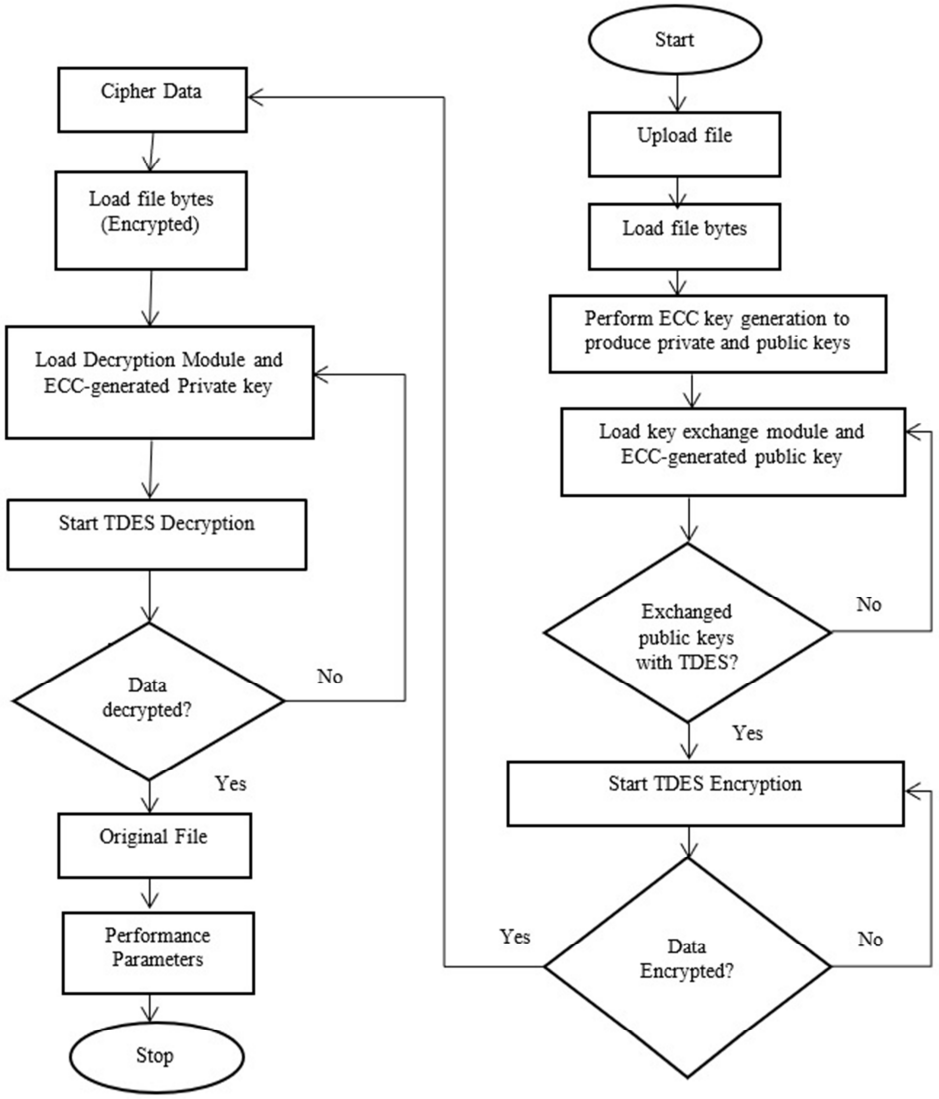

Fig. 3.3 Work flow diagram for proposed scheme

3.4 Implementation

To realise the potential of the proposed algorithm, it is mandatory to implement it and observe its performance empirically. For this purpose, the algorithm has been applied to the database of a web application titled "Secure Cloud Computing" running in cloud environment. The application has been developed using ASP.NET framework available in Microsoft Visual Studio. The data uploaded to cloud is stored in a database supported by SQL Server 2019.

Storing plain text files (files in original form) in the database can lead to serious data security threats. Since the data is important to the cloud user, the CSP needs to be sure that data stored is safe. The web application "Secure Cloud Computing" encrypts the files before storing them, ensuring data confidentiality and integrity. The cloud can be used to store

image, audio, video and text files of any format. A comparison between the hybrid ECC-TDES algorithm, ECC algorithm and TDES algorithm has been carried to evaluate the performance of the proposed technique. For this purpose, the experiment will be run both on a local machine and on hosted application, the configuration used for the same has been listed in table 3.1 and table 3.2 respectively. It is anticipated that using ECC-TDES hybrid scheme will deliver more safety to data for cloud storage as well as more accuracy of reconstructed files after decryption.

Table 3.1 Configuration specification used for running experiment on localhost

Network	The client and server both resided in the same local machine, eliminating any restriction or delay possible due to a network connection.
Server Machine	Operating system: Windows 10 Enterprise. System type: (64bit). RAM: 4.00GB. Processor: Intel(R) Core (TM) i3-4160T CPU @ 3.10 GHz 3.10 GHz
Client Machine	Same as server machine
Tools	A tool was written in order to send and receive files from server, encrypt and decrypt data in database using the proposed hybrid ECC-TDES algorithm, as well as measure the encryption/decryption time and accuracy obtained in decrypting data. The tool displays the parameters in the web application itself. The tool was written with C# programming language. Microsoft Visual Studio 2019 was used to run the code and integrate the environment. The source code for this tool is provided in Appendix I and II.

Table 3.2 Configuration specification used for running experiment over a network connection

Network	Client and Server both reside in the same wireless Local Area Network (LAN) with connection speed 150 MBPs. The full bandwidth was left free for the experiment, any other network operations were avoided.
Client machine	Operating system: Windows 10 Home. System type: (64bit). RAM: 8.00GB. Processor: Intel(R) Core (TM) i5-7200U CPU @ 2.50 GHz 2.71 GHz
Server machine	Operating system: Windows 10 Enterprise. System type: (64bit). RAM: 4.00GB. Processor: Intel(R) Core (TM) i3-4160T CPU @ 3.10 GHz 3.10 GHz
Tools	A tool was written in order to send and receive files from server, encrypt and decrypt data in database using the proposed hybrid ECC-TDES algorithm, as well as measure the encryption/decryption time and accuracy obtained in decrypting data. The tool displays the parameters in the web application itself. The tool was written with C# programming language. Microsoft Visual Studio 2019 was used to run the code and integrate the environment. The source code for this tool is provided in Appendix I and II.

Files of varying format and sizes will be uploaded and downloaded multiple times to record encryption time, decryption tine and accuracy of each algorithm.

3.5 Framework Design

This section represents the design of “Secure Cloud Computing” web application.

3.5.1 Home page

The home page holds the title “Secure Cloud computing” with subtitle “research to secure the storage and transmission over a cloud network” indicative of the research idea. The tabs to different sections of the page can also be seen in top right section of the page in Fig. 3.3(a). Clicking on the arrow button below the title takes the user directly to the “Sign-up section” of the page.

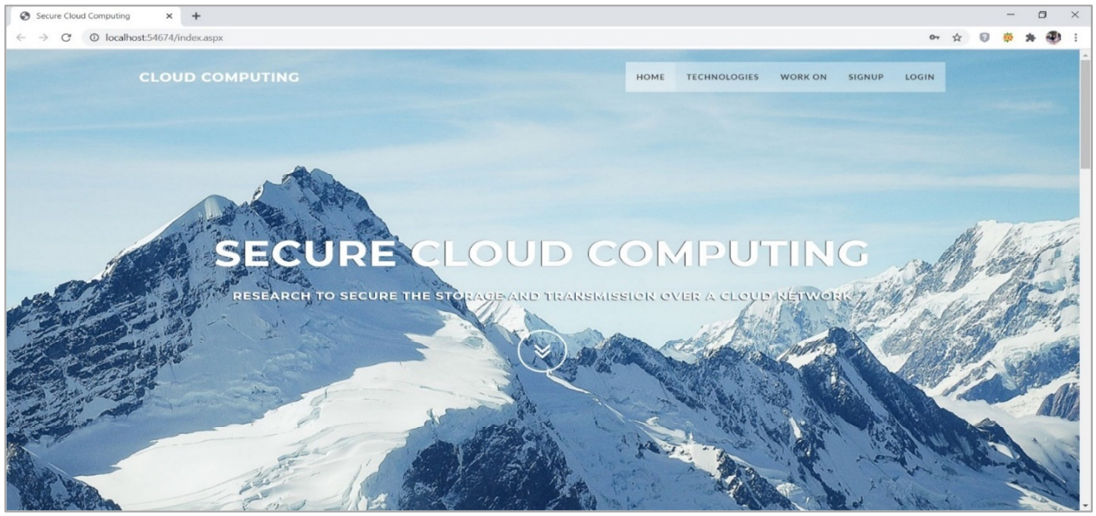

Fig. 3.3(a) Home page

Scrolling further the “Technologies” and “Work on” sections can be seen as in the figures 3.3(b) and 3.3(c) below.

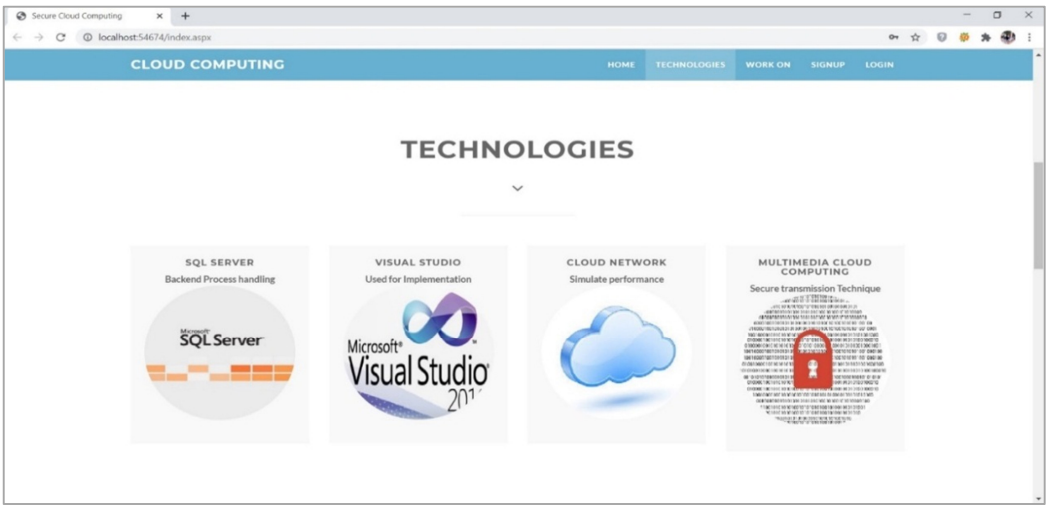

Fig. 3.3(b) Technologies section on Home page

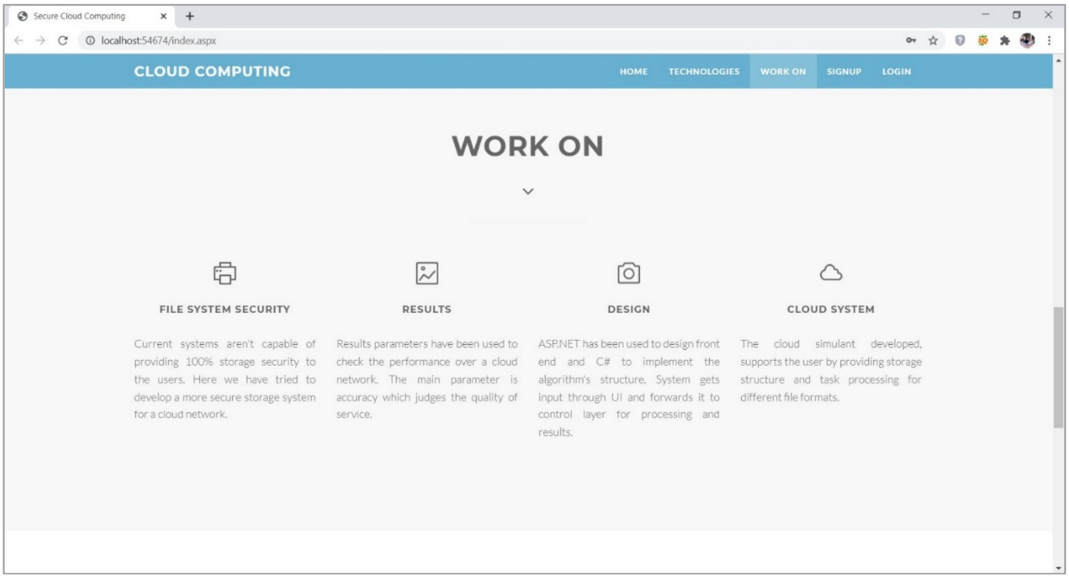

Fig. 3.3(c) Work On section of Home page

3.5.2 Main authorization section

The main authorization section is titled "Create your account" on Home page as shown in Fig. 3.4(a). It has two modules;

- Sign up for creating new user account
- Log in for authorized users to access their existing accounts

The log in credentials and data structures, of each user are unique and confidential, hence the uploaded data is secure within the cloud environment.

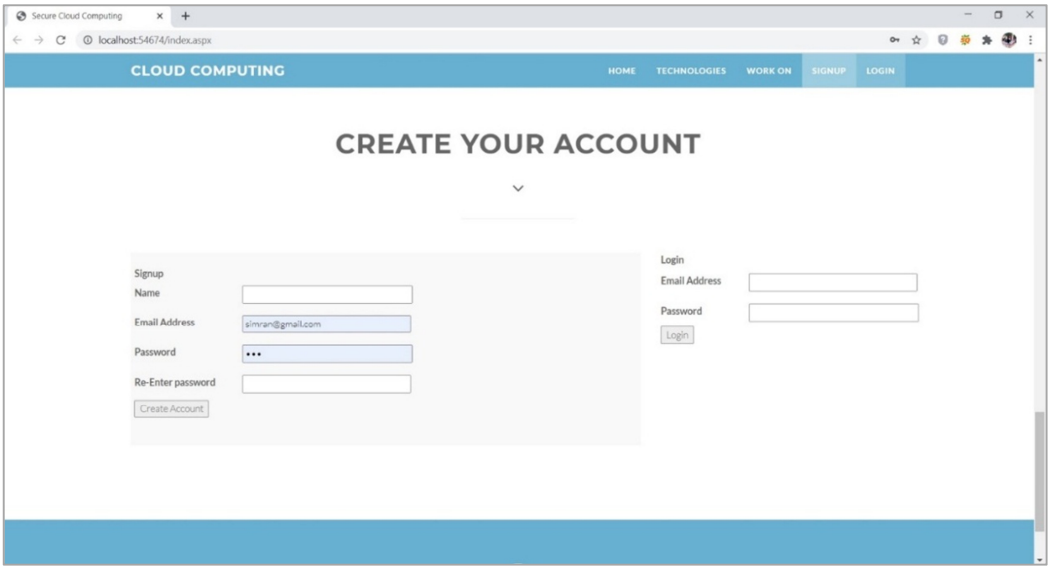

Fig. 3.4(a) Main authorization window

The “Email Address” field in Sign up section has validation applied to it, so the user needs to enter a valid email address (i.e. of the form name@example.com) otherwise the message “Must enter valid email.” will be displayed (Fig. 3.4(b)). Also, in case the values for “Password” and “Re-enter password” don’t match, the account will not be created (Fig. 3.4(c)).

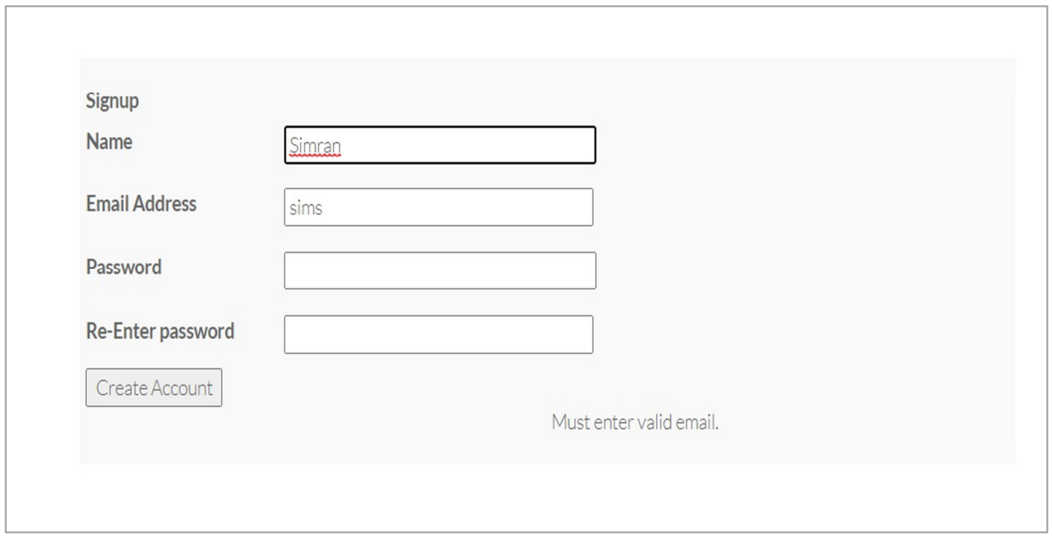

Fig. 3.4(b) Email validation in Sign up module

Fig. 3.4(c) Passwords differed; user account not created

Once all the valid credentials are filled in, user account is successfully created (Fig. 3.4(d)). The user can fill in the login module and start using the service.

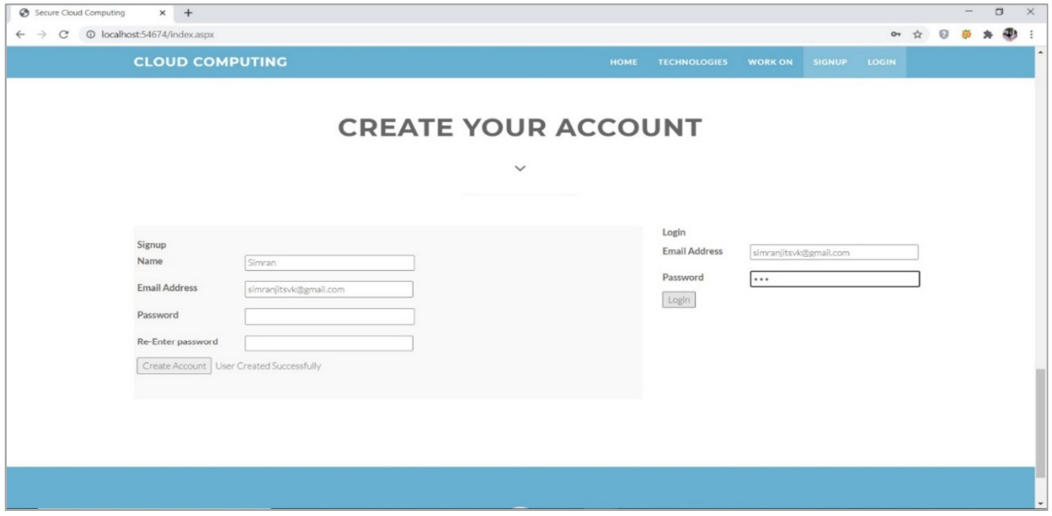

Fig. 3.4(d) User account ready for login

3.5.3 User account

Once logged in the user is welcomed with the page shown in Fig. 3.5. The page has same sections as described in fig. 3.4(b) and 3.4(c). The changed tabs have been discussed further.

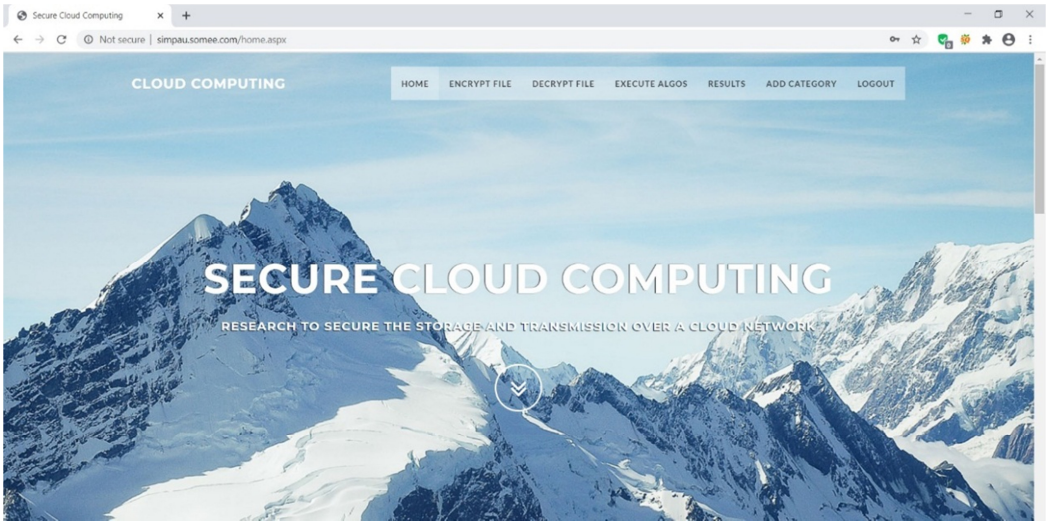

Fig. 3.5 Welcome page for logged in user

3.5.4 Category module

Clicking on "Add category" takes the logged in the user to category specification window. Here the user is required to add categories representing the files to be uploaded (Fig. 3.6). These user-defined categories are stored in SQL database with character format and displayed at the time of uploading/downloading of files. The purpose of category module is to easily locate file in database at the time of downloading from decryption panel.

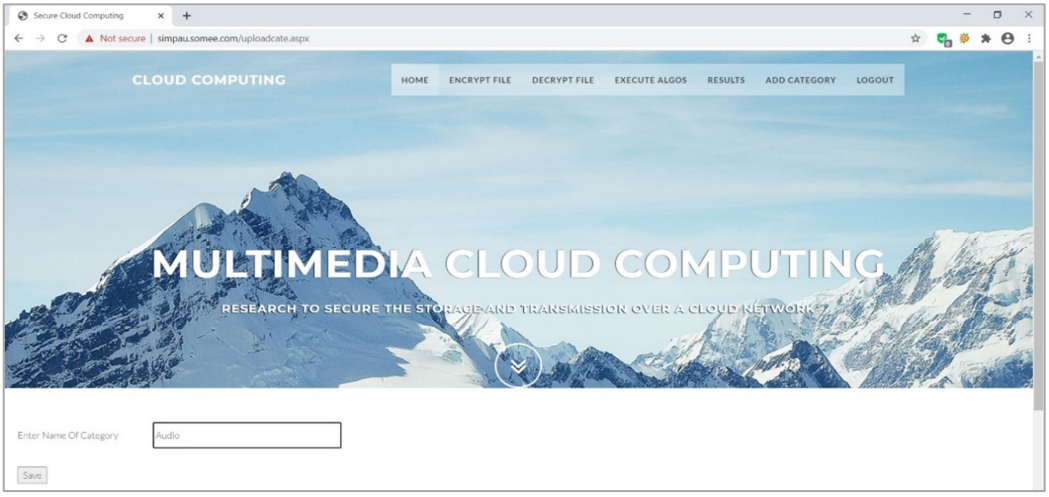

Fig. 3.6 Category specification window

3.5.5 Encryption module

The encryption window is the first module of framework that deploys the proposed ECC-TDES cryptosystem. This module has some important fields which are required to store and retrieve files from database. This window allows user to add multimedia content file including text, image, video and audio. Before uploading a file, the user needs to fill in every field displayed on the window in Fig. 3.7. The category field lists the categories added by the user itself. The server collects the data from the user end and binds columns to extract from the cloud database.

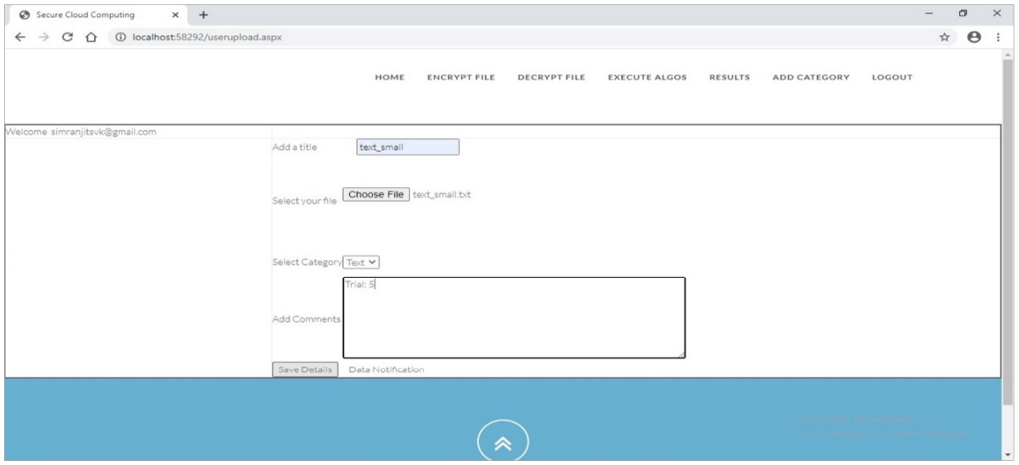

Fig. 3.7 Encryption window

Clicking on “save details” button executes encryption, producing cipher code for the submitted content. The keys and cipher codes are stored on database for further processing as well. Once the encrypted file is uploaded to the cloud server, user can go to the decryption module for downloads.

3.5.6 Decryption module

Download module is attached with file decryption functionality. With the selection of file and entering valid ID as done in Fig. 3.8, users can retrieve the encrypted files in original format from cloud storage. Clicking on “Download” button will save the file to the user’s device.

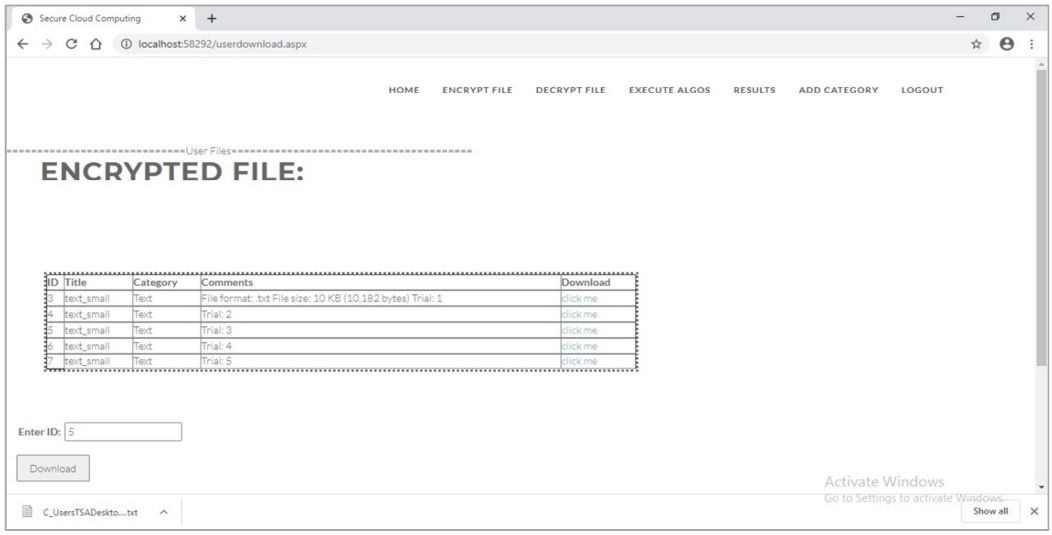

Fig. 3.8 Decrypted file download

3.5.7 Execute algorithms

The “Execute algos” module performs ECC and TDES cryptography on the data separately. This addition has been done for the purpose of comparing individual ECC and TDES algorithms with the proposed hybrid ECC-TDES scheme. Clicking on the “ECC” and “TDES” buttons displayed in Fig. 3.9 downloads decrypted file chosen in previous decryption module.

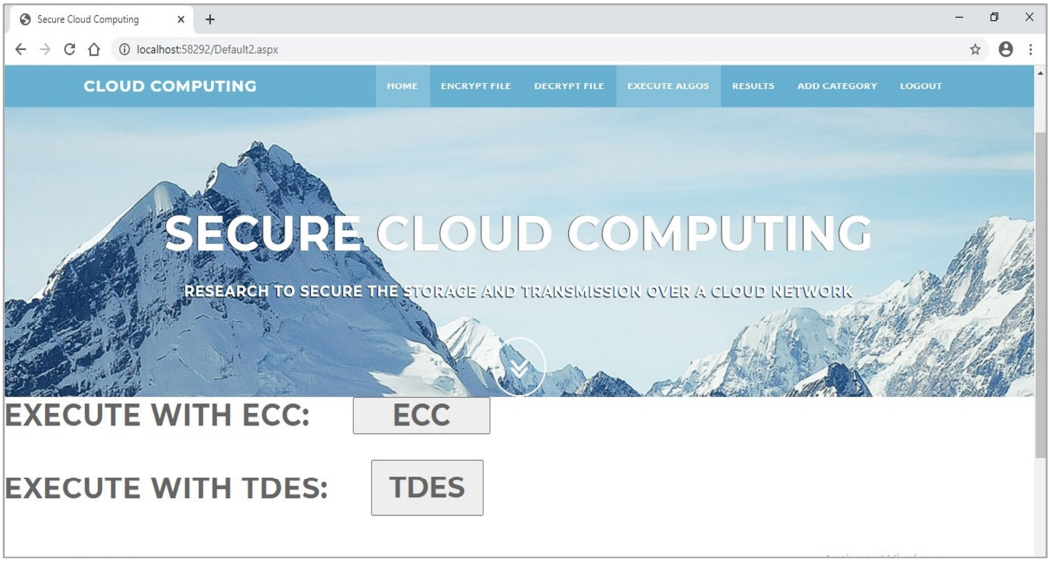

Fig. 3.9 Execute algorithms

3.5.8 Results module

For analysing the performance of the proposed hybrid ECC-TDES algorithm, encryption time, decryption time and accuracy have been presented and compared with individual ECC and TDES algorithms in "Results" module of the developed cloud framework (Fig. 3.10). The attributes of this section have been further discussed in the results chapter.

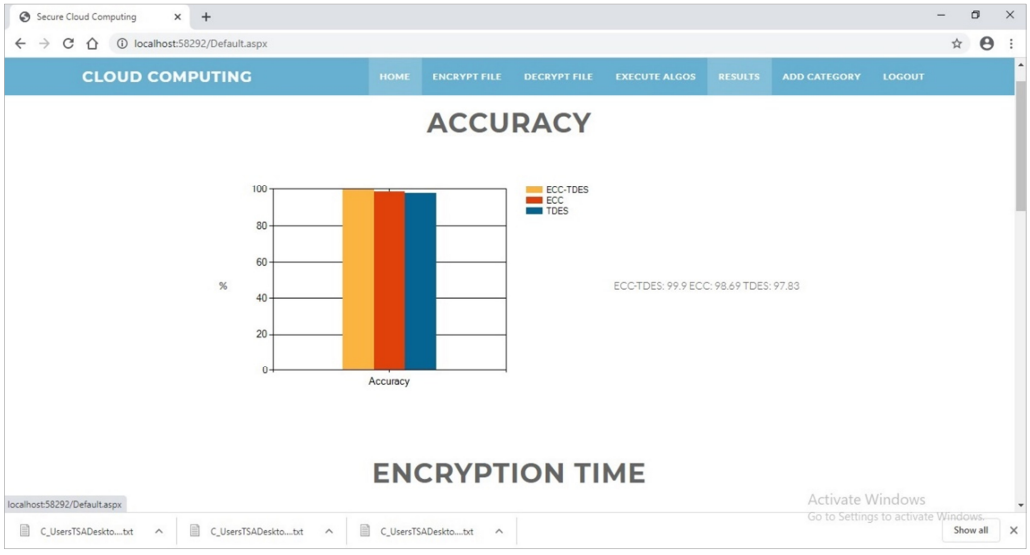

Fig. 3.10 Results module

CHAPTER-IV

RESULTS AND DISCUSSION

The first objective of this research was to study the security issues in cloud-based services. For this purpose, the research works and reports compiled over the past decade were reviewed. The security issues observed to be most widespread are:

1. Insufficient Identity, Credential, Access and Key Management
2. Data Breaches and Data loss
3. Insecure APIs
4. Insider threat
5. Account hijacking
6. Lack of cloud security architecture and strategy

In present time, all the major industries are adopting the trend to store their data in the cloud, as it provides easy and quick access to data anywhere and anytime. The information stored in cloud database ranges from highly sensitive to non-sensitive data. Since data is one of the precious assets for any organisation it is important to maintain data security. Using cryptography to encrypt data before storing it to the database is one of the most widely techniques used for securing the cloud data. In this thesis, efforts have been made to do the same using hybrid cryptography scheme.

As discussed before, a cloud-based web application “Secure Cloud Computing” was designed to be delivered in SaaS service model. A hybrid cryptography algorithm comprising of ECC (for key generation) and TDES (for encryption/decryption of data) was developed to encrypt data to be stored in the database. To further analyse the proposed technique’s performance, it was compared with individual ECC and TDES algorithms. These comparisons were made on the basis of encryption time, decryption time, accuracy and throughput. The experiments were run using four file types: text, image, audio and video; each with two different file sizes ranging from 10 KB to 100 KB. Also, the experiment was run 10 times each on localhost and a remote machine by hosting the application on cloud. The raw data of the experiments has been presented in Appendix III. The observations for the raw data were recorded in the form of accuracy, encryption time, and decryption time from “Results” page of the web application as shown in fig 4.1 (a), 4.1(b) and 4.1(c). Encryption time is the main parameter which shows the time taken by any algorithm to encrypt uploaded content. Similarly, decryption time shows time taken to decrypt and retrieve the original file from the database. The reconstructed file obtained after decryption should have nominal error rate. The accuracy of the decrypted file depends on the recovery rate of data frame.

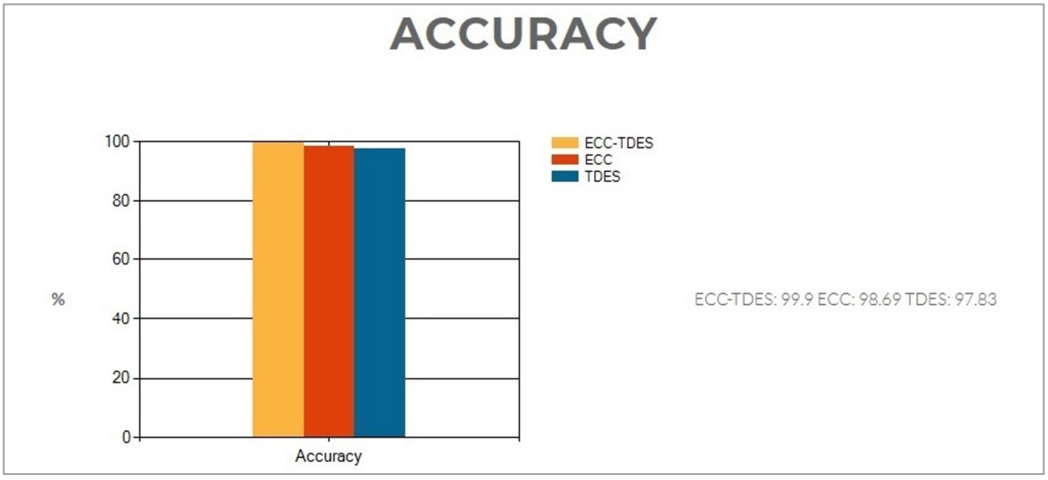

Fig. 4.1(a) Accuracy as displayed on Results page

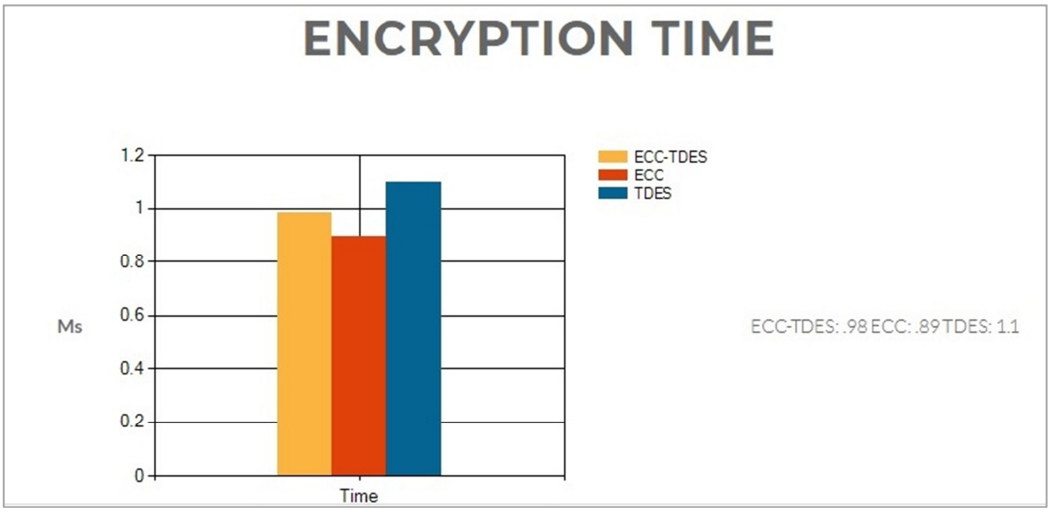

Fig. 4.1(b) Encryption time as displayed on Results page

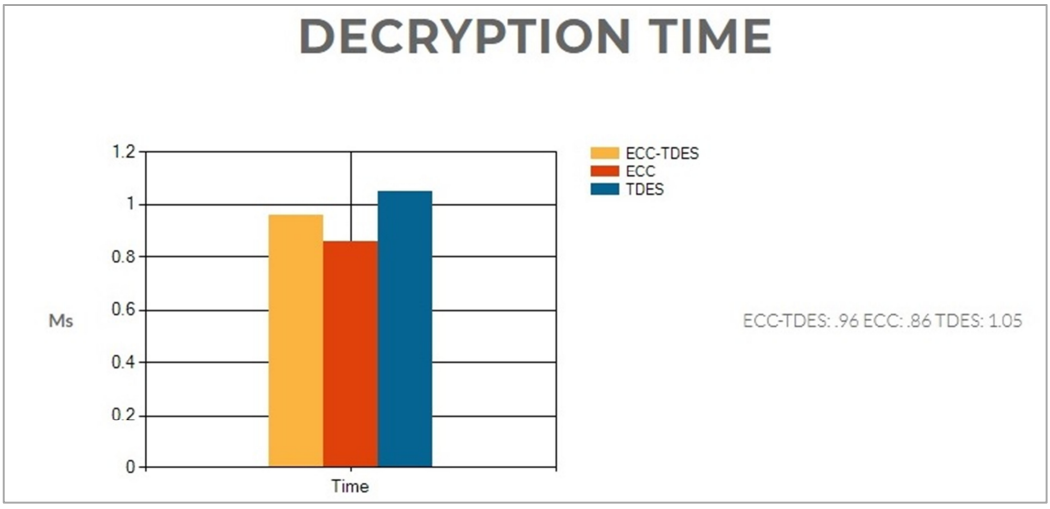

Fig. 4.1(c) Decryption time as displayed on Results page

Taking reference from Sandha *et* al (2011), the throughput of encryption as well as decryption schemes was calculated separately as the file sizes in kilo bytes divided by the average encryption time (in milliseconds), and in the case of decryption scheme throughput is calculated as the file size in kilo bytes divided by the average decryption time (in milliseconds). Further presented are empirical observations of the four file types.

1. Text File

For experimenting with text files, two Notepad (.txt) files of sizes 10 KB and 100 KB were used, named "text_small" and "text_big" respectively. The comparisons have been done in tables below.

File name: text_small.txt

File size: 9.94 KB (10,182 bytes) ~ 10 KB

No. of trials: 10

Table 4.1(a) Metrices for experiment run on localhost for 10 KB text file

Algorithm	Average Encryption time (ms)	Average Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	0.863	0.886	99.90	11.52	11.22
ECC	0.813	0.782	98.58	12.23	12.71
TDES	0.802	0.766	97.48	12.39	12.98

Table 4.1(b) Metrices for experiment run over network connection for 10 KB text file

Algorithm	Average Encryption time (ms)	Average Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	0.743	0.901	99.90	13.38	11.03
ECC	0.770	0.738	98.53	12.91	13.49
TDES	0.801	0.766	97.49	12.41	12.98

Table 4.1(c) Comparative accuracy and throughput for 10 KB text file

Algorithm	Average Encryption Throughput (KB/ms)	Average Decryption Throughput (KB/ms)	Average Accuracy (%)
ECC-TDES	12.45	11.13	99.9
ECC	12.57	13.10	98.55
TDES	12.40	12.98	97.48

File name: text_big.txt

File size: 100 KB (1,02,719 bytes)

No. of trials: 10

Table 4.2(a) Metrices for experiment run on localhost for 100 KB text file

Algo	Encryption time (ms)	Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	8.428	8.967	99.90	11.87	11.15
ECC	6.948	6.672	98.43	14.39	14.99
TDES	8.612	8.237	97.55	11.61	12.14

Table 4.2(b) Metrices for experiment run over network connection for 100 KB text file

Algo	Encryption time (ms)	Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	9.367	9.071	99.90	10.68	11.02
ECC	6.977	6.701	98.45	14.33	14.92
TDES	7.679	7.281	97.44	13.02	13.73

Table 4.2(c) Comparative accuracy and throughput for 100 KB text file

Algorithm	Average Encryption Throughput (KB/ms)	Average Decryption Throughput (KB/ms)	Average Accuracy (%)
ECC-TDES	11.28	11.13	99.9
ECC	14.36	13.10	98.55
TDES	12.32	12.98	97.48

Table 4.1(c) and 4.2(c), clearly suggest that ECC algorithm has highest encryption and decryption throughput while ECC-TDES gives maximum accuracy.

2. Image File

For experimenting with image files, two JPG files of sizes 10 KB and 100 KB were used, named “img_small” and “img_big” respectively. The comparisons have been done in tables below.

File name: img_small.jpg

File size: 10 KB (10,261 bytes)

No. of trials: 10

Table 4.3(a) Metrices for experiment run on localhost for 10 KB image file

Algorithm	Encryption time (ms)	Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	0.988	0.899	99.90	10.21	11.12
ECC	0.696	0.668	98.43	14.37	14.97
TDES	0.839	0.802	97.52	11.92	12.47

Table 4.3(b) Metrices for experiment run over network connection for 10 KB image file

Algorithm	Encryption time (ms)	Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	0.884	0.947	99.90	11.31	10.56
ECC	0.796	0.764	98.26	12.56	13.09
TDES	0.931	0.897	97.26	10.74	11.15

Table 4.3(c) Comparative accuracy and throughput for 10 KB image file

Algorithm	Average Encryption Throughput (KB/ms)	Average Decryption Throughput (KB/ms)	Average Accuracy (%)
ECC-TDES	10.76	10.84	99.9
ECC	13.47	14.03	98.35
TDES	11.33	11.81	97.39

File name: img_big.jpg

File size: 100 KB (1,02,858 bytes)

No. of trials: 10

Table 4.4(a) Metrices for experiment run on localhost for 100 KB image file

Algorithm	Encryption time (ms)	Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	9.802	9.404	99.9	10.21	10.63
ECC	7.357	7.062	98.477	13.59	14.16
TDES	8.713	8.334	97.589	11.48	11.99

Table 4.4(b) Metrices for experiment run over network connection for 100 KB image file

Algorithm	Encryption time (ms)	Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	8.599	8.571	99.9	11.63	11.67
ECC	7.691	7.382	98.519	13.01	13.55
TDES	8.249	7.89	97.502	12.12	12.67

Table 4.4(c) Comparative accuracy and throughput for 100 KB image file

Algorithm	Average Encryption Throughput (KB/ms)	Average Decryption Throughput (KB/ms)	Average Accuracy (%)
ECC-TDES	10.92	11.15	99.9
ECC	13.3	13.86	98.03
TDES	11.80	12.33	97.55

Table 4.3(c) and 4.4(c), show that ECC algorithm has highest encryption and decryption throughput while ECC-TDES gives maximum accuracy.

3. Audio File

For experimenting with audio files, two MPEG files of sizes 10.7 KB and 99.3 KB were used, named "aud_small" and "aud_big" respectively. The comparisons have been done in tables below.

File name: aud_small.mpeg

File size: 10.7 KB (11,059 bytes)

No. of trials: 10

Table 4.5(a) Metrices for experiment run on localhost for 10.7 KB audio file

Algorithm	Encryption time (ms)	Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	0.935	1.152	99.9	11.44	9.29
ECC	0.807	0.776	98.496	13.26	13.79
TDES	0.908	0.87	97.524	11.78	12.29

Table 4.5(b) Metrices for experiment run over network connection for 10.7 KB audio file

Algorithm	Encryption time (ms)	Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	1.036	1.101	99.9	10.33	9.72
ECC	0.78	0.747	98.463	13.72	14.32
TDES	0.863	0.827	97.478	12.39	12.94

Table 4.5(c) Comparative accuracy and throughput for 10.7 KB audio file

Algorithm	Average Encryption Throughput (KB/ms)	Average Decryption Throughput (KB/ms)	Average Accuracy (%)
ECC-TDES	10.89	9.51	99.9
ECC	13.49	14.06	98.48
TDES	12.09	12.62	97.50

File name: aud_big.mpeg

File size: 99.3 KB (1,01,761 bytes)

No. of trials: 10

Table 4.6(a) Metrices for experiment run on localhost for 99.3 KB audio file

Algorithm	Encryption time (ms)	Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	8.098	7.946	99.9	12.26	12.49
ECC	7.818	7.505	98.553	12.70	13.23
TDES	8.637	8.262	97.558	11.49	12.02

Table 4.6(b) Metrices for experiment run over network connection for 99.3 KB audio file

Algorithm	Encryption time (ms)	Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	8.55	9.549	99.9	11.61	10.39
ECC	7.695	7.388	98.529	12.91	13.44
TDES	7.859	7.516	97.468	12.64	13.21

Table 4.6(c) Comparative accuracy and throughput for 99.3 KB audio file

Algorithm	Average Encryption Throughput (KB/ms)	Average Decryption Throughput (KB/ms)	Average Accuracy (%)
ECC-TDES	11.94	11.44	99.9
ECC	12.81	13.35	98.54
TDES	12.07	12.62	97.51

Table 4.5(c) and 4.6(c), show that ECC algorithm has highest encryption and decryption throughput while ECC-TDES gives maximum accuracy.

4. Video File

As a video file of minimum size was required, a 3gp file of 24.6 KB, namely "vid_small" was used. For bigger video file, MP4 file of 111 KB size was used. The comparisons have been done in tables below.

File name: vid_small.3gp

File size: 24.6 KB (25,201 bytes)

No. of trials: 10

Table 4.7(a) Metrices for experiment run on localhost for 24.6 KB video file

Algorithm	Encryption time (ms)	Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	2.262	2.179	99.9	10.88	11.29
ECC	1.999	1.92	98.611	12.31	12.81
TDES	2.037	1.949	97.509	12.08	12.62

Table 4.7(b) Metrices for experiment run over network connection for 24.6 KB video file

Algorithm	Encryption time (ms)	Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	2.191	2.173	99.9	11.23	11.32
ECC	2.06	1.978	98.606	11.94	12.44
TDES	1.868	1.787	97.431	13.17	13.77

Table 4.7(c) Comparative accuracy and throughput for 24.6 KB video file

Algorithm	Average Encryption Throughput (KB/ms)	Average Decryption Throughput (KB/ms)	Average Accuracy (%)
ECC-TDES	11.06	11.31	99.9
ECC	12.13	12.63	98.61
TDES	12.63	13.19	97.47

File name: vid_big.mp4

File size: 111 KB (1,14,659 bytes)

No. of trials: 10

Table 4.8(a) Metrices for experiment run on localhost for 111 KB video file

Algorithm	Encryption time (ms)	Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	9.78	9.722	99.9	11.35	11.42
ECC	8.489	8.151	98.507	13.08	13.62
TDES	9.977	9.542	97.583	11.13	11.63

Table 4.8(b) Metrices for experiment run over network connection for 111 KB video file

Algorithm	Encryption time (ms)	Decryption time (ms)	Accuracy (%)	Encryption Throughput (KB/ms)	Decryption Throughput (KB/ms)
ECC-TDES	10.752	10.158	99.9	10.32	10.93
ECC	7.993	7.673	98.453	13.89	14.47
TDES	9.571	9.155	97.568	11.59	12.13

Table 4.8(c) Comparative accuracy and throughput for 111 KB video file

Algorithm	Average Encryption Throughput (KB/ms)	Average Decryption Throughput (KB/ms)	Average Accuracy (%)
ECC-TDES	10.84	11.18	99.9
ECC	13.49	14.05	98.48
TDES	11.36	11.88	97.58

Table 4.7(c) shows TDES has highest throughput for 3gp file of 24.6 KB, while table 4.8(c) supports ECC for highest throughput. This difference in observations can be investigated in future. ECC-TDES again gave maximum accuracy for video file.

4.1 Conclusion

In the experiment performed above, data encryption and decryption were done separately using the proposed hybrid ECC-TDES scheme, ECC and TDES. The observations demonstrate that ECC gives best performance with highest throughput, which is also advocated by other research workers. This observation also supports the choice of ECC for key generation in the proposed hybrid cryptography scheme. Another observation is that ECC-TDES gives highest accuracy in the decrypted files.

Even though the experiment proves hybrid ECC-TDES to be low in performance as compared to ECC and TDES, the hybrid structure of enhanced TDES provides more security by increasing the complexity. ECC is the best algorithm of asymmetric encryption technology as with 160-bit key size it takes 9.6x 1011 MIP years with the best-known attack to be broken down (ElTaweel *et al* 2020). If analysed individually, the results of this work showed ECC-TDES is less time-consuming and highly accurate cryptographic system, and provides high security to the cloud storage. ECC-TDES is a highly accurate algorithm means it offers reliability and integrity for users' data. Since the foundation of cloud computing is based on fast networks that are growing exponentially, quick and light-weight key generation algorithms like ECC-TDES are crucial for expanding cloud performance. The only drawback of the proposed cryptographic system is the decreased throughput that can be due to the increased complexity from combination of two algorithms.

CHAPTER-V

SUMMARY

Each year cloud adoption is hitting a new growth spurt as organisations around the globe are discovering the power of cloud services to accommodate almost any business need. Cloud computing is no more limited to an expanse of servers or services, paid by consumers for usage over the internet. Cloud has become a symbol for modern day computing – where mix and match of services can meet several application needs. Adding to the load, each consumer company requires cloud services in separate structures to continue their routine business activities. To meet this growing requirement, efforts are being made by cloud developers and experts to produce more comprehensive and optimized cloud computing models. These dynamic cloud computing trends demand a thorough elaboration of structure, concept, deployment, and security mechanism for cloud environment. And that's what propelled this research towards studying present the security issues and techniques in cloud-based services, along with analysis of these security techniques for computational applications.

Pertaining to the first objective, the recent research works, articles and reports concentrating on security issues, solutions, and difficulties in cloud services were audited. The cloud security issues most prevalent over the past decade are; Insufficient Identity, Credential, Access and Key Management; Data Breaches and Data loss; Insecure API's; Insider threat; Account hijacking; and Lack of cloud security architecture and strategy. In this thesis project, different security issues found in cloud services and techniques deployed for eliminating them were also studied. Efforts have been made to apply some of these techniques to a web application running in cloud environment. The web application namely "Secure Cloud Computing" has been delivered under Software-as-a-service (SaaS) model of cloud. The application has been developed using ASP.NET framework available in Microsoft Visual Studio. The data uploaded to cloud is stored in a database supported by SQL Server 2019. The major goal was to provide data security. Network security, physical security, legal compliance, disaster or risk management were not in the scope of this thesis. The motivation for the recommended security measure was derived from previously done research work by other researchers.

Major tasks for cloud development are securing, protecting and processing the data uploaded to the cloud by consumers. This can be done while data is in transit or is stored in cloud storage servers. Most cloud service providers target data accessibility, claims, and data and process isolation to ensure integrity of cloud users' data. But since the cloud services are interconnected, users need to be aware of the network used to communicate their confidential

data. But then again securing a network spread over the globe is unfeasible, fortunately an alternative to protect confidential data is converting it to a form that is uncomprehensive for an unauthorized person i.e. encryption. Prior research works suggest that data encryption (cryptography) and access control are two primal techniques being used to implement cloud security in present times. Cryptographic security mechanisms have known to reduce the prospect of data breach by 60% (Anonymous 2018).

This research proposes a new hybrid cryptosystem constructed using Elliptic Curve cryptography (ECC) for key generation and Triple DES (TDES) algorithm for data encryption (asymmetric and symmetric algorithms respectively). A file uploaded to the cloud database is forwarded to encryption module. The ECC algorithm performs key generation to produce a set of public and private keys binding to user's profile and the file uploaded. Using SHA3, the ECC-generated keys are exchanged with keys generated by TDES, producing final set of keys. TDES using the final set of keys performs encryption and the encrypted file is stored in the cloud's database. This algorithm was developed using C# programming language.

To realise it's potential, the proposed algorithm was implemented on the database of the web application titled "Secure Cloud Computing" running in cloud environment. The web application "Secure Cloud Computing" encrypts the files before storing them, ensuring data confidentiality and integrity. A comparison between the hybrid ECC-TDES algorithm, ECC algorithm and TDES algorithm was also done to evaluate the performance of the proposed technique. For this purpose, the experiments were run using four file types: text, image, audio and video; each with two different file sizes ranging from 10 KB to 100 KB. Also, the experiment was run 10 times each on localhost and a remote machine by hosting the application on cloud. The observations were recorded in the form of accuracy, encryption time, and decryption time from "Results" page of the web application. This raw data collected from the experiments has been presented in Appendix III.

Implications

In the experiment, ECC gave best performance with highest throughput as advocated by other research workers. This observation supports the choice of ECC for key generation in the proposed hybrid cryptography scheme. Another observation was that ECC-TDES gives highest accuracy in the decrypted files.

Even though the experiment proves hybrid ECC-TDES to be low in performance as compared to ECC and TDES, the hybrid structure of enhanced TDES provides more security by increasing the complexity. If analysed individually, the results of this work showed ECC-TDES is less time-consuming and highly accurate cryptographic system, and provides high security to the cloud storage. ECC-TDES is a highly accurate algorithm means it offers

reliability and integrity for users' data. Since the foundation of cloud computing is based on fast networks that are growing exponentially, quick and light-weight key generation algorithms like ECC-TDES are crucial for expanding cloud performance. The only drawback of the proposed cryptographic system is the decreased throughput that can be due to the increased complexity from combination of two algorithms.

Future scope

The proposed algorithm can be tested against different attacks like Brute-force attack, Man-in-the-middle, phishing, eavesdropping etc. Efforts can be made to increase the throughput of the algorithm. Also, the proposed cryptographic scheme can be applied and tested for data in transmission. More advanced cryptographic algorithms can be designed by hybridizing different symmetric and asymmetric cryptography schemes. The complexity of algorithms can be increased by incorporating artificial intelligence or deep learning. The cloud security issues identified other than data security can also be resolved in future research work.

REFERENCES

Akinlolu P and Kazeem A (2014) A New Hybrid Data Encryption and Decryption Technique to Enhance Data Security in Communication Networks: Algorithm Development. *Int J Sci Engg Res* **5**:804-11.

Alassafi O, Albugmi A, Walters R and Wills G (2016) Data Security in Cloud Computing. *Proc 5th Int Conf Future Gen Comm Tech.* pp 55-59. University of Bedfordshire, Luton. doi: 10.1109/FGCT.2016.7605062.

Alharthi A, Walters R J, Wills G B and Yahya F (2015) An Overview of Cloud Services Adoption Challenges in Higher Education Institutions. *Proc 2nd Int Workshop Emerging Software as a Service and Analytics.* Vol 1, pp 102-109. Lisbon, Portugal.

Anonymous (2018) *The 2018 Global Cloud Data Security Study.* Conducted by Ponemon Institute LLC, Michigan, United States.

Anonymous (2019) *Cost of a Data Breach Report 2019.* Conducted by Ponemon Institute LLC, Michigan, United States.

Anonymous (2020) *The 2020 Cyber Security Report.* Complied by Check Point Software Technologies Ltd., San Carlos, United States.

Babu S, Sajay K R and Vijayalakshmi Y (2019) Enhancing the security of cloud data using hybrid encryption algorithm. *J Ambient Intell Human Comp* https://doi.org/10.1007/s12652-019-01403-1

Brandão P R (2018) The Importance of Authentication and Encryption in Cloud Computing Framework Security. *Int J Data Sci Tech* **4:**1-5.

Brook J (2019) *Top Threats to Cloud Computing: The Egregious 11.* Conducted by Cloud Security Alliance, Washington, United States

Carr R (2018) *Top challenges in cloud security for 2018.* Retrieved January 20, 2019, from https://www.zettaset.com/blog/top-challenges-cloud-security-2018/

Cramer R and Shoup V (2019) Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack. *SIAM J Comp* **33:**167–226.

Dhiman H, Hussain S, Maheshwari S and Mushtaque M D A (2014) Evaluation of DES, TDES, AES, Blowfish and Twofish encryption algorithm based on space complexity. *Int J Engg Res Tech* **3:**1922–33.

ElTaweel G S, Hassan H E and Tahoun M (2020) A robust computational DRM framework for protecting multimedia contents using AES and ECC. *Alexandria Engg J* **59:**1275-1286.

Feng, D, Zhang M, Zhang Y and Xu Z (2011) Study on Cloud Computing Security. *J Software* **22**:71-83.

Gandhi C and Kaushik S (2016) Cloud data security with hybrid symmetric encryption. *Int Conf on Techs in Info and Comm Tech.* pp 636-640, GGS Indraprastha University, New Delhi. doi: 10.1109/ICCTICT.2016.7514656

Grance T and Mell P (2011) The NIST Definition of Cloud Computing. NIST Special Publication 800-145.

Gupta H, Khatri S K and Sharma Y (2019) A security model for the enhancement of data privacy in cloud computing. *Amity Int Conf Artif Intell.* pp 898-902, Amity University, Dubai. doi:10.1109/AICAI.2019.8701398

Gupta S, Iyera S C and Sedamkar R R (2016) A Novel Idea on Multimedia Encryption using Hybrid Crypto Approach. *7th Int Conf Com Comp Virt*, Vol 79, pp 293-98, Thakur College of Engineering and Technology, Mumbai, India.

Hameed R T, Hamid O T, Hussain A A, Mohamad O A, Zidan K A and Salman S A (2018) Improved cloud computing security. *1st Annual Int Conf on Info and Sci.* pp 170-175, Fallujah, Iraq. doi:10.1109/AICIS.2018.00041

Ilayaraja M, Kumar K S, Rajesh M and Kumar K S (2017) Sensitive data security in cloud computing aid of different encryption techniques. *J Adv Res Dynam Cont Sys* **9** (supple 18).

Jayakumari J, Khan M A, Mishra K K and Santhi N (2015) A New Hybrid Technique for Data Encryption. *Proc 2015 Global Conf on Comm Tech.* pp 925-929, doi: 10.1109/GCCT.2015.7342801.

Kaul V and Shaikh A P (2014) Enhanced Security Algorithm using Hybrid Encryption and ECC. *IOSR J Comp Engg* **16**:80-85.

Kaushik S and Patel A (2019) Secure cloud data using hybrid cryptographic scheme. *4th Int Conf IoT: Smart Innov Usages.* pp 1-6, Ghaziabad, India. doi:10.1109/IoT-SIU.2019.8777592.

Kumar M G V and Ragupathy U S (2016) A survey on current key issues and status in cryptography. *Int Conf on Wireless Comm Signal Proc Net.* pp 205-210, Chennai, India. doi: 10.1109/WiSPNET.2016.7566121.

Kumar S and Singh S (2018) Analysis of Various Cryptographic Algorithms. *Int J Adv Res Sci Engg Tech.* **5**:8.

Maddineni V S K and Ragi S (2011) *Security Techniques for Protecting Data in Cloud Computing.* M. Tech. Thesis, Blekinge Institute of Technology, Karlskrona, Sweden

Magons K (2016) Applications and Benefits of Elliptic Curve Cryptography. *Proc SOFSEM co-located with 42nd Int Conf Current Trends Theory Prac Comp Sci.* Vol 1548, pp 32-42, Harrachov, Czech Republic.

Ming X and Wu X (2010) Research of the Database Encryption Technique Based on Hybrid Cryptography. *Proc 2010 Int Symp Comp Intell Design.* pp 68-71, Hangzhou. doi: 10.1109/ISCID.2010.105. doi: 10.1109/ISCID.2010.105.

Ou Y (2015) *The concept of cloud computing and the main security issues in it.* Bachelor's thesis, Turku University of Applied Sciences Turku, Southwest Finland

Rizvi S I and Tandra S A (2014) *Security for Cloud Based Services.* Master's thesis, School of Information and Communication Technology (ICT), KTH Royal Institute of Technology Stockholm, Sweden

Sandha K S, Singh G and Singla A K (2011) Through Put Analysis of Various Encryption Algorithms. *Int J Comp Sci Engg* **2:**527-29.

Stallings W (ed) (2005) *Cryptography and Network Security: Principles and Practices.* pp 310-313. Prentice Hall, New Jersey.

Stallings W (ed) (2017) *Cryptography and Network Security: Principles and Practices.* pp 330-334. Pearson Education Ltd., Essex.

Appendix I

Code to perform proposed Hybrid ECC-TDES encryption on data and measure encryption time in C# programming language.

```
namespace EncryptUsingMR
{
 class encrypt_class
  {
    public byte[] encrypt_PublicKey;
    public byte[] data_encrypt_class()
    {
      using (ECDiffieHellmanCng encrypt_ = new ECDiffieHellmanCng())
      {
        encrypt_.KeyDerivationFunction = ECDiffieHellmanKeyDerivationFunction.Hash;
        encrypt_.HashAlgorithm = CngAlgorithm.Sha256;
        encrypt_PublicKey = encrypt_.PublicKey.ToByteArray();
      }
      return encrypt_PublicKey;
    }
    public void Send(byte[] key, byte[] secretMessage, out byte[] encryptedMessage, out
byte[] tdesIV)
    {
      byte[] MyEncryptedArray = new byte[24];
      Array.Copy(key,MyEncryptedArray, 23);
      MD5CryptoServiceProvider MyMD5CryptoService = new
        MD5CryptoServiceProvider();
      byte[] data_bytes = secretMessage;
```

```
            byte[] MysecurityKeyArray = MyEncryptedArray;
            MyMD5CryptoService.Clear();
            var MyTripleDESCryptoService = new
              TripleDESCryptoServiceProvider();
            tdesIV = MyTripleDESCryptoService.IV;
            MyTripleDESCryptoService.IV = tdesIV;
            MyTripleDESCryptoService.Key = MyEncryptedArray;
            MyTripleDESCryptoService.Padding = PaddingMode.Zeros;
            MyTripleDESCryptoService.Mode = CipherMode.ECB;
            MyTripleDESCryptoService.Padding = PaddingMode.PKCS7;
            var MyCrytpoTransform = MyTripleDESCryptoService
              .CreateEncryptor();
            byte[] MyresultArray = MyCrytpoTransform.TransformFinalBlock(data_bytes, 0,
data_bytes.Length);
            MyTripleDESCryptoService.Clear();
            encryptedMessage = MyresultArray;
        }
    }
    public class Bob
    {
        public byte[] bobPublicKey;
        private byte[] bobKey;
        public void get_Bob(byte[] encrypt_PublicKey)
        {
            using (ECDiffieHellmanCng bob = new ECDiffieHellmanCng())
            {
```

```
            bob.KeyDerivationFunction = ECDiffieHellmanKeyDerivationFunction.Hash;
            bob.HashAlgorithm = CngAlgorithm.Sha256;
            bobPublicKey = bob.PublicKey.ToByteArray();
            bobKey = bob.DeriveKeyMaterial(CngKey.Import(encrypt_PublicKey,
CngKeyBlobFormat.EccPublicBlob));
          }
        }
    }
}
```

Appendix II

Code to perform proposed Hybrid ECC-TDES decryption on data and measure decryption time and accuracy of reconstructed file in C# programming language.

```
using System;

using System.Configuration;

using System.Data;

using System.Linq;

using System.Web;

using System.Web.Security;

using System.Web.UI;

using System.Web.UI.HtmlControls;

using System.Web.UI.WebControls;

using System.Web.UI.WebControls.WebParts;

using System.Xml.Linq;

using System.Security.Cryptography;

using System.IO;

using System.Data.SqlClient;

using System.Net;

using System.Net.Mail;

using System.Text;

using MicroSoft_CryptoVerification;

using System.Security.Cryptography;

using System.Text;

using System.Threading.Tasks;

using System.time;

using MicroSoft_CryptoVerification;
```

```
using MicrosoftParametersHandler;
using System.Threading;
using System.FileProcessing.Verfication;
public partial class Default3 : System.Web.UI.Page
{
    SqlConnection con = new SqlConnection();
    SqlConnection conaz = new SqlConnection();
    protected void Page_Load(object sender, EventArgs e)
    {
        try
        {
            con.ConnectionString =
ConfigurationManager.ConnectionStrings["con"].ConnectionString;
            if (con.State == ConnectionState.Closed)
            {
                con.Open();
            }
        }
        catch
        {
            Response.Write("Problem Connecting to the local server , Check sql server settings ");
        }
        if (Page.IsPostBack == false)
        {
            filldltexxt();
        }
```

```
    }
    private void filldltexxt()
    {//where id=@id
        SqlDataAdapter adp = new SqlDataAdapter("select * from tbtext where id=@id ",
ConfigurationManager.ConnectionStrings["con"].ConnectionString);
        adp.SelectCommand.Parameters.Add("@id", SqlDbType.VarChar, 500).Value =
Session["email"].ToString();
        DataSet ds = new DataSet();
        adp.Fill(ds);
        DataList2.DataSource = ds;
        DataList2.DataBind();
    }
    protected void DataList1_EditCommand(object source, DataListCommandEventArgs e)
    {
        Button1.Visible = true;
        TextBox1.Visible = true;
    }
    protected void DataList1_SelectedIndexChanged(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender, EventArgs e)
    {
        int id = Convert.ToInt32(TextBox1.Text);
        String query1 = "select * from keytable inner join tbtext on keytable.id=tbtext.file_id
where keytable.id=@id";
        String name = "";
        SqlCommand cmd1 = new SqlCommand(query1, con);
```

```
        cmd1.Parameters.Add("@id", SqlDbType.Int).Value = id;
        byte[] key = null;
        byte[] decryptedMessage = null;
        byte[] data = null;
        byte[] ivparam = null;
        SqlDataReader rd = cmd1.ExecuteReader();
        if (rd.Read())
        {
            key = (byte[])rd["encryptionKey"];
            data = (byte[])rd["text"];
            ivparam = (byte[])rd["ivparam"];
            name = BitConverter.ToString(key)+rd["name"].ToString() ;
        }
        rd.Close();
        cmd1.Dispose();
        Datetime.start(data);
        new EncryptUsingMR_download.encrypt_class().Send(key, data,ivparam, out
decryptedMessage);
        String path = Server.MapPath(name);
        File.WriteAllBytes(path, decryptedMessage);
        double al = Datetime.stop();
        double accu = ProcessError.errorRate(data, decryptedMessage);
        Session["dec_time"] = al;
        Session["accuracy"] = 100-accu;
        double hybridvals = AnalyticReport.CustomAnalysis(decryptedMessage, data,
"hybrid");
```

```
        Session["hybridanalysisReport"] = hybridvals;
        CaptureAnalytics objs = (CaptureAnalytics)Session["analysisReport"];
        objs.setAnalyticsPropertyCounter12(al);
        objs.setAnalyticsPropertyCounter6(100 - accu);
        Session["analysisReport"] = objs;
        try
        {
            string strURL = "Downloads\\"+name;
            WebClient req = new WebClient();
            HttpResponse response = HttpContext.Current.Response;
            response.Clear();
            response.ClearContent();
            response.ClearHeaders();
            response.Buffer = true;
            response.AddHeader("Content-Disposition", "attachment;filename=\"" +
Server.MapPath(strURL) + "\"");
            response.BinaryWrite(decryptedMessage);
            response.End();
        }
        catch (Exception ex)
        {
        }
    }
    protected void Button2_Click(object sender, EventArgs e)
    {
    }
```

```
    protected void DataList1_SelectedIndexChanged(object source,
DataListCommandEventArgs e)
    {
     Button1.Visible = true;
        TextBox1.Visible = true;
    }
}
namespace EncryptUsingMR_download
{
    class encrypt_class
    {
        public byte[] encrypt_PublicKey;
        public byte[] data_encrypt_class()
        {
            using (ECDiffieHellmanCng encrypt_ = new ECDiffieHellmanCng())
            {
                encrypt_.KeyDerivationFunction = ECDiffieHellmanKeyDerivationFunction.Hash;
                encrypt_.HashAlgorithm = CngAlgorithm.Sha256;
                encrypt_PublicKey = encrypt_.PublicKey.ToByteArray();
            }
            return encrypt_PublicKey;
        }
        public void Send(byte[] key, byte[] secretMessage,byte[] tdesIV, out byte[]
decryptedMessage)
        {
            byte[] MyEncryptedArray = new byte[24];
```

```
            Array.Copy(key, MyEncryptedArray, 23);
            MD5CryptoServiceProvider MyMD5CryptoService = new
              MD5CryptoServiceProvider();
            byte[] data_bytes = secretMessage;
            byte[] MysecurityKeyArray = MyEncryptedArray;
            MyMD5CryptoService.Clear();
            var MyTripleDESCryptoService = new
              TripleDESCryptoServiceProvider();
            tdesIV = MyTripleDESCryptoService.IV;
            MyTripleDESCryptoService.IV = tdesIV;
            MyTripleDESCryptoService.Key = MyEncryptedArray;
            MyTripleDESCryptoService.Padding = PaddingMode.Zeros;
            MyTripleDESCryptoService.Mode = CipherMode.ECB;
            MyTripleDESCryptoService.Padding = PaddingMode.PKCS7;
            var MyCrytpoTransform = MyTripleDESCryptoService
              .CreateDecryptor();
            byte[] MyresultArray = MyCrytpoTransform.TransformFinalBlock(data_bytes, 0,
data_bytes.Length);
            MyTripleDESCryptoService.Clear();
            decryptedMessage = MyresultArray;
        }
    }
    public class Bob
    {
        public byte[] bobPublicKey;
        private byte[] bobKey;
```

```
        public void get_Bob(byte[] encrypt_PublicKey)
        {
            using (ECDiffieHellmanCng bob = new ECDiffieHellmanCng())
            {
                bob.KeyDerivationFunction = ECDiffieHellmanKeyDerivationFunction.Hash;
                bob.HashAlgorithm = CngAlgorithm.Sha256;
                bobPublicKey = bob.PublicKey.ToByteArray();
                bobKey = bob.DeriveKeyMaterial(CngKey.Import(encrypt_PublicKey,
CngKeyBlobFormat.EccPublicBlob));
            }
        }
    }
}
```

Appendix III

Experimental observations

1. ECC-TDES encryption/decryption time and accuracy for 9.94 KB text file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	0.87	1.22
2	99.9	0.68	0.57
3	99.9	1.09	0.64
4	99.9	0.64	1.25
5	99.9	0.98	0.96
6	99.9	0.81	0.59
7	99.9	0.95	1.04
8	99.9	1.01	0.94
9	99.9	0.82	0.96
10	99.9	0.78	0.69
AVERAGE	99.9	0.863	0.886

2. ECC encryption/decryption time and accuracy for 9.94 KB text file on localhost

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.43	0.69	0.67
2	98.87	1.04	1.00
3	98.15	0.47	0.45
4	98.73	0.93	0.90
5	98.69	0.89	0.86
6	98.68	0.89	0.85
7	98.80	0.98	0.94
8	98.72	0.93	0.89
9	98.16	0.48	0.46
10	98.60	0.83	0.80
AVERAGE	98.583	0.813	0.782

3. TDES encryption/decryption time and accuracy for 9.94 KB text file on localhost

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.22	0.57	0.54
2	97.44	0.76	0.73
3	97.20	0.56	0.53
4	97.78	1.06	1.01
5	97.83	1.10	1.05
6	97.57	0.88	0.84
7	97.22	0.57	0.55
8	97.40	0.72	0.69
9	97.54	0.85	0.81
10	97.60	0.95	0.91
AVERAGE	97.48	0.802	0.766

4. ECC-TDES encryption/decryption time and accuracy for 100 KB text file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	7.00	6.30
2	99.9	9.32	12.14
3	99.9	9.64	8.24
4	99.9	7.07	6.78
5	99.9	5.61	12.13
6	99.9	11.89	12.13
7	99.9	7.43	6.81
8	99.9	11.44	6.98
9	99.9	6.52	6.98
10	99.9	8.36	11.18
AVERAGE	99.9	8.428	8.967

5. ECC encryption/decryption time and accuracy for 100 KB text file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.80	9.98	9.58
2	98.14	4.65	4.46
3	98.50	7.57	7.27
4	98.77	9.68	9.30
5	98.86	10.38	9.96
6	98.30	5.91	5.68
7	98.22	5.28	5.07
8	98.21	5.16	4.96
9	98.37	6.45	6.19
10	98.11	4.42	4.25
AVERAGE	98.428	6.948	6.672

6. TDES encryption/decryption time and accuracy for 100 KB text file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.70	9.98	9.55
2	97.31	6.54	6.25
3	97.29	6.38	6.10
4	97.82	11.05	10.57
5	97.48	8.08	7.73
6	97.67	9.68	9.25
7	97.14	5.10	4.88
8	97.82	10.98	10.50
9	97.58	8.89	8.51
10	97.64	9.44	9.03
AVERAGE	97.545	8.612	8.237

7. ECC-TDES encryption/decryption time and accuracy for 10 KB image file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	0.68	1.21
2	99.9	1.24	1.25
3	99.9	1.00	0.57
4	99.9	0.96	0.93
5	99.9	1.20	0.76
6	99.9	0.78	0.80
7	99.9	1.23	0.61
8	99.9	1.02	1.06
9	99.9	1.17	0.80
10	99.9	0.60	1.00
AVERAGE	99.9	0.988	0.899

8. ECC encryption/decryption time and accuracy for 10 KB image file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.26	0.56	0.54
2	98.19	0.50	0.48
3	98.23	0.53	0.51
4	98.88	1.06	1.02
5	98.81	1.00	0.96
6	98.19	0.51	0.49
7	98.38	0.66	0.63
8	98.38	0.66	0.63
9	98.79	0.99	0.95
10	98.17	0.49	0.47
AVERAGE	98.428	0.696	0.668

9. TDES encryption/decryption time and accuracy for 10 KB image file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.58	0.89	0.86
2	97.60	0.91	0.87
3	97.45	0.78	0.74
4	97.27	0.62	0.59
5	97.31	0.65	0.62
6	97.74	1.03	0.98
7	97.55	0.86	0.83
8	97.4	0.74	0.70
9	97.43	0.76	0.73
10	97.87	1.15	1.10
AVERAGE	97.52	0.839	0.802

10. ECC-TDES encryption/decryption time and accuracy for 10 KB image file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	10.84	11.66
2	99.9	7.97	12.82
3	99.9	10.21	5.98
4	99.9	1034	10.56
5	99.9	12.36	10.34
6	99.9	12.28	7.47
7	99.9	7.65	7.58
8	99.9	12.45	9.84
9	99.9	5.54	11.24
10	99.9	8.38	6.55
AVERAGE	99.9	9.802	9.404

11. ECC encryption/decryption time and accuracy for 100 KB image file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.76	9.66	9.27
2	98.49	7.49	7.19
3	98.30	5.91	5.67
4	98.20	5.15	4.95
5	98.68	8.96	8.6
6	98.45	7.11	6.83
7	98.76	9.66	9.27
8	98.33	6.17	5.92
9	98.23	5.37	5.15
10	98.57	8.09	7.77
AVERAGE	98.477	7.357	7.062

12. TDES encryption/decryption time and accuracy for 100 KB image file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.82	11.0	10.53
2	97.23	5.90	5.64
3	97.21	5.69	5.44
4	97.54	8.61	8.23
5	97.41	7.48	7.15
6	97.87	11.47	10.97
7	97.75	10.39	9.94
8	97.60	9.10	8.71
9	97.72	10.18	9.74
10	97.40	7.31	6.99
AVERAGE	97.589	8.713	8.334

13. ECC-TDES encryption/decryption time and accuracy for 10.7 KB audio file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	1.08	1.11
2	99.9	0.71	1.02
3	99.9	1.03	1.05
4	99.9	0.56	0.81
5	99.9	0.71	1.23
6	99.9	0.93	1.15
7	99.9	1.25	1.24
8	99.9	1.14	1.08
9	99.9	0.89	1.27
10	99.9	1.05	0.78
AVERAGE	99.9	0.935	1.152

14. ECC encryption/decryption time and accuracy for 10.7 KB audio file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.61	0.91	0.87
2	98.11	0.48	0.46
3	98.57	0.87	0.84
4	98.32	0.65	0.63
5	98.48	0.79	0.76
6	98.87	1.13	1.08
7	98.6	0.90	0.87
8	98.14	0.50	0.48
9	98.65	0.94	0.90
10	98.61	0.90	0.87
AVERAGE	98.496	0.807	0.776

15. TDES encryption/decryption time and accuracy for 10.7 KB audio file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.38	0.77	0.74
2	97.28	0.68	0.65
3	97.18	0.58	0.56
4	97.59	0.97	0.93
5	97.62	1.00	0.96
6	97.75	1.12	1.07
7	97.87	1.24	1.18
8	97.66	1.03	0.99
9	97.54	0.93	0.89
10	97.37	0.76	0.73
AVERAGE	97.524	0.908	0.870

16. ECC-TDES encryption/decryption time and accuracy for 99.3 KB audio file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	8.59	9.55
2	99.9	5.63	11.91
3	99.9	6.82	5.73
4	99.9	11.63	8.08
5	99.9	5.31	5.72
6	99.9	11.07	8.22
7	99.9	10.12	7.11
8	99.9	6.32	8.03
9	99.9	8.83	7.79
10	99.9	6.66	7.32
AVERAGE	99.9	8.098	7.946

17. ECC encryption/decryption time and accuracy for 99.3 KB audio file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.79	9.80	9.41
2	98.76	9.50	9.12
3	98.69	9.01	8.65
4	98.29	5.76	5.53
5	98.21	5.15	4.94
6	98.23	5.29	5.07
7	98.74	9.39	9.02
8	98.79	9.80	9.41
9	98.59	8.19	7.86
10	98.35	6.29	6.04
AVERAGE	98.553	7.818	7.505

18. TDES encryption/decryption time and accuracy for 99.3 KB audio file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.74	10.19	9.74
2	97.18	5.36	5.12
3	97.25	5.94	5.68
4	97.74	10.21	9.76
5	97.79	10.65	10.19
6	97.56	8.69	8.32
7	97.14	5.00	4.79
8	97.73	10.16	9.72
9	97.82	10.89	10.42
10	97.63	9.28	8.88
AVERAGE	97.558	8.637	8.262

19. ECC-TDES encryption/decryption time and accuracy for 24.6 KB video file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	2.77	1.34
2	99.9	2.01	1.40
3	99.9	2.09	2.53
4	99.9	1.73	2.51
5	99.9	1.81	1.70
6	99.9	3.04	2.52
7	99.9	2.27	3.05
8	99.9	1.84	2.31
9	99.9	2.23	2.20
10	99.9	2.83	2.23
AVERAGE	99.9	2.262	2.179

20. ECC encryption/decryption time and accuracy for 24.6 KB video file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.34	1.53	1.47
2	98.80	2.44	2.34
3	98.89	2.62	2.51
4	98.64	2.12	2.03
5	98.66	2.16	2.08
6	98.63	2.11	2.03
7	98.47	1.80	1.73
8	98.35	1.56	1.50
9	98.48	1.81	1.74
10	98.49	1.84	1.77
AVERAGE	98.611	1.999	1.92

21. TDES encryption/decryption time and accuracy for 24.6 KB video file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.85	2.77	2.65
2	97.63	2.30	2.20
3	97.16	1.28	1.23
4	97.37	1.74	1.67
5	97.44	1.89	1.80
6	97.50	2.02	1.94
7	97.77	2.60	2.49
8	97.69	2.43	2.32
9	97.31	1.61	1.54
10	97.37	1.73	1.65
AVERAGE	97.509	2.037	1.949

22. ECC-TDES encryption/decryption time and accuracy for 111 KB video file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	7.69	6.22
2	99.9	11.39	14.04
3	99.9	9.63	8.71
4	99.9	6.65	12.21
5	99.9	12.25	12.02
6	99.9	8.13	9.71
7	99.9	11.19	8.05
8	99.9	12.15	7.53
9	99.9	7.63	11.58
10	99.9	11.09	7.15
AVERAGE	99.9	9.78	9.722

23. ECC encryption/decryption time and accuracy for 111 KB video file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.84	11.49	11.03
2	98.71	10.33	9.92
3	98.83	11.39	10.93
4	98.22	5.91	5.68
5	98.60	9.30	8.93
6	98.84	11.47	11.01
7	98.40	7.53	7.23
8	98.22	5.92	5.69
9	98.31	6.73	6.46
10	98.10	4.82	4.63
AVERAGE	98.507	8.489	8.151

24. TDES encryption/decryption time and accuracy for 111 KB video file on localhost.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.79	11.96	11.44
2	97.70	11.11	10.63
3	97.19	6.13	5.87
4	97.24	6.67	6.38
5	97.37	7.89	7.54
6	97.74	11.50	11.00
7	97.80	12.11	11.58
8	97.79	12.00	11.47
9	97.66	10.70	10.23
10	97.55	9.70	9.28
AVERAGE	97.583	9.977	9.542

25. ECC-TDES encryption/decryption time and accuracy for 9.94 KB text file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	0.69	0.98
2	99.9	0.55	0.53
3	99.9	0.93	0.66
4	99.9	0.77	1.02
5	99.9	0.72	0.82
6	99.9	0.68	1.06
7	99.9	0.97	1.19
8	99.9	0.73	0.97
9	99.9	0.71	1.19
10	99.9	0.68	0.59
AVERAGE	99.9	0.743	0.901

26. ECC encryption/decryption time and accuracy for 9.94 KB text file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.58	0.81	0.78
2	98.49	0.74	0.71
3	98.54	0.78	0.75
4	98.65	0.87	0.83
5	98.86	1.04	0.99
6	98.69	0.90	0.86
7	98.35	0.63	0.60
8	98.12	0.44	0.43
9	98.85	1.02	0.98
10	98.15	0.47	0.45
AVERAGE	98.528	0.77	0.738

27. TDES encryption/decryption time and accuracy for 9.94 KB text file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.54	0.85	0.81
2	97.12	0.48	0.46
3	97.85	1.12	1.07
4	97.36	0.69	0.66
5	97.52	0.83	0.79
6	97.29	0.63	0.61
7	97.34	0.67	0.64
8	97.40	0.73	0.70
9	97.76	1.04	0.99
10	97.68	0.97	0.93
AVERAGE	97.486	0.801	0.766

28. ECC-TDES encryption/decryption time and accuracy for 100 KB text file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	10.53	9.35
2	99.9	11.26	6.95
3	99.9	7.12	8.08
4	99.9	7.66	10.13
5	99.9	10.98	9.72
6	99.9	8.34	9.83
7	99.9	5.76	8.82
8	99.9	11.81	8.82
9	99.9	7.52	12.78
10	99.9	12.69	6.23
AVERAGE	99.9	9.367	9.071

29. ECC encryption/decryption time and accuracy for 100 KB text file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.23	5.36	5.15
2	98.64	8.66	8.32
3	98.25	5.48	5.27
4	98.53	7.76	7.45
5	98.44	7.04	6.76
6	98.40	6.70	6.43
7	98.19	5.04	4.84
8	98.41	6.78	6.51
9	98.70	9.16	8.80
10	98.53	7.79	7.48
AVERAGE	98.452	6.977	6.701

30. TDES encryption/decryption time and accuracy for 100 KB text file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.31	6.59	6.30
2	97.84	11.19	10.07
3	97.75	10.43	9.98
4	97.17	5.30	5.07
5	97.20	7.63	5.39
6	97.46	7.89	7.55
7	97.50	8.19	7.83
8	97.30	6.48	6.19
9	97.34	6.85	6.55
10	97.50	8.24	7.88
AVERAGE	97.437	7.679	7.281

31. ECC-TDES encryption/decryption time and accuracy for 10 KB image file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	1.19	1.17
2	99.9	0.97	1.04
3	99.9	1.08	0.83
4	99.9	0.85	1.24
5	99.9	1.24	1.13
6	99.9	0.96	0.78
7	99.9	0.54	0.89
8	99.9	0.66	1.10
9	99.9	0.68	0.51
10	99.9	0.67	0.78
AVERAGE	99.9	0.884	0.947

32. ECC encryption/decryption time and accuracy for 10 KB image file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.82	1.01	0.97
2	98.34	0.62	0.60
3	98.57	0.81	0.77
4	98.75	0.95	0.91
5	98.81	1.00	0.96
6	98.68	0.89	0.86
7	98.77	0.97	0.93
8	98.13	0.46	0.44
9	98.58	0.81	0.78
10	98.11	0.44	0.42
AVERAGE	98.256	0.796	0.764

33. TDES encryption/decryption time and accuracy for 10 KB image file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.70	1.00	0.95
2	97.69	0.99	0.95
3	97.55	0.86	0.83
4	97.79	1.07	1.03
5	97.52	0.83	0.80
6	97.37	0.71	0.68
7	97.70	0.99	0.95
8	97.78	1.07	1.02
9	97.88	1.16	1.11
10	97.28	0.63	0.60
AVERAGE	97.626	0.931	0.897

34. ECC-TDES encryption/decryption time and accuracy for 10 KB image file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	10.71	9.84
2	99.9	7.16	5.99
3	99.9	11.62	5.88
4	99.9	10.04	10.67
5	99.9	12.01	707
6	99.9	5.74	10.47
7	99.9	5.37	8.40
8	99.9	5.37	10.91
9	99.9	8.17	7.49
10	99.9	9.80	8.99
AVERAGE	99.9	8.599	8.571

35. ECC encryption/decryption time and accuracy for 100 KB image file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.55	7.94	7.62
2	98.88	10.60	10.18
3	98.25	5.56	5.34
4	98.64	8.64	8.29
5	98.58	8.17	7.84
6	98.67	8.90	8.54
7	98.46	7.23	6.94
8	98.34	6.23	5.98
9	98.18	4.97	4.77
10	98.64	8.67	8.32
AVERAGE	98.519	7.691	7.382

36. TDES encryption/decryption time and accuracy for 100 KB image file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.49	8.10	7.75
2	97.66	9.62	9.20
3	97.41	7.47	7.14
4	97.87	11.50	11.00
5	97.68	9.82	9.39
6	97.46	7.90	7.56
7	97.30	6.45	6.17
8	97.10	4.74	4.54
9	97.60	9.14	8.74
10	97.45	7.75	7.41
AVERAGE	97.502	8.249	7.89

37. ECC-TDES encryption/decryption time and accuracy for 10.7 KB audio file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	1.14	1.23
2	99.9	0.94	0.99
3	99.9	1.32	1.32
4	99.9	1.14	1.15
5	99.9	0.87	1.35
6	99.9	0.82	1.11
7	99.9	0.51	0.78
8	99.9	1.35	1.19
9	99.9	1.13	0.78
10	99.9	1.14	1.11
AVERAGE	99.9	1.036	1.101

38. ECC encryption/decryption time and accuracy for 10.7 KB audio file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.28	0.62	0.60
2	98.73	1.01	0.97
3	98.71	0.99	095
4	98.11	0.47	0.45
5	98.31	0.65	0.62
6	98.29	0.63	0.60
7	98.62	0.92	0.88
8	98.29	0.63	0.60
9	98.88	1.14	1.09
10	98.41	0.74	0.71
AVERAGE	98.463	0.78	0.747

39. TDES encryption/decryption time and accuracy for 10.7 KB audio file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.79	1.16	1.11
2	97.32	0.72	0.69
3	97.18	0.58	0.56
4	97.85	1.21	1.16
5	97.46	0.85	0.81
6	97.88	1.25	1.19
7	97.11	0.51	0.49
8	97.34	0.73	0.70
9	97.18	0.58	0.56
10	97.67	1.04	1.00
AVERAGE	97.478	0.863	0.827

40. ECC-TDES encryption/decryption time and accuracy for 99.3 KB audio file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	6.70	7.31
2	99.9	9.24	9.99
3	99.9	7.40	12.29
4	99.9	8.37	7.10
5	99.9	8.91	10.88
6	99.9	8.87	8.04
7	99.9	7.18	12.46
8	99.9	10.31	8.77
9	99.9	8.67	11.58
10	99.9	9.85	7.07
AVERAGE	99.9	8.55	9.549

41. ECC encryption/decryption time and accuracy for 99.3 KB audio file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.27	5.62	5.39
2	98.71	9.14	8.77
3	98.69	9.00	8.64
4	98.48	7.29	7.00
5	98.70	9.08	8.72
6	98.87	10.39	9.98
7	98.18	4.90	4.71
8	98.30	5.90	5.67
9	98.41	6.73	6.46
10	98.68	8.90	8.54
AVERAGE	98.529	7.695	7.388

42. TDES encryption/decryption time and accuracy for 99.3 KB audio file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.75	10.27	9.82
2	97.49	8.06	7.71
3	97.10	4.68	4.47
4	97.48	7.93	7.58
5	97.14	5.00	4.78
6	97.39	7.20	6.89
7	97.56	8.64	8.27
8	97.14	5.04	4.82
9	97.87	11.36	10.87
10	97.76	10.41	9.95
AVERAGE	97.468	7.859	7.516

43. ECC-TDES encryption/decryption time and accuracy for 24.6 KB video file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	2.70	2.66
2	99.9	1.37	2.49
3	99.9	3.00	2.45
4	99.9	1.32	1.42
5	99.9	2.04	2.39
6	99.9	2.33	1.47
7	99.9	2.44	2.36
8	99.9	2.63	2.46
9	99.9	2.70	1.67
10	99.9	1.38	2.36
AVERAGE	99.9	2.191	2.173

44. ECC encryption/decryption time and accuracy for 24.6 KB video file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.15	1.17	1.12
2	98.66	2.17	2.08
3	98.64	2.13	2.04
4	98.66	2.15	2.07
5	98.33	1.51	1.45
6	98.80	2.44	2.35
7	98.84	2.53	2.43
8	98.71	2.27	2.18
9	98.75	2.35	2.25
10	98.52	1.88	1.81
AVERAGE	98.606	2.06	1.978

45. TDES encryption/decryption time and accuracy for 24.6 KB video file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.18	1.32	1.26
2	97.62	2.26	2.17
3	97.64	2.31	2.21
4	97.32	1.63	1.56
5	97.49	2.00	1.92
6	97.30	1.58	1.51
7	97.79	2.65	2.53
8	97.49	2.00	1.9
9	97.30	1.59	1.52
10	97.18	134	1.28
AVERAGE	97.431	1.868	1.787

46. ECC-TDES encryption/decryption time and accuracy for 111 KB video file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	99.9	10.37	8.56
2	99.9	11.81	12.38
3	99.9	12.75	14.21
4	99.9	12.10	12.13
5	99.9	8.86	9.97
6	99.9	9.15	9.02
7	99.9	11.79	11.52
8	99.9	11.36	6.08
9	99.9	10.89	10.79
10	99.9	8.44	6.92
AVERAGE	99.9	10.752	10.158

47. ECC encryption/decryption time and accuracy for 111 KB video file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	98.77	10.79	10.36
2	98.67	9.94	9.54
3	98.69	10.13	9.72
4	98.33	6.93	6.65
5	98.55	8.87	8.52
6	98.28	6.43	6.18
7	98.29	6.52	6.26
8	98.54	8.73	8.38
9	98.17	5.48	5.26
10	98.24	6.11	5.86
AVERAGE	98.453	7.993	7.673

48. TDES encryption/decryption time and accuracy for 111 KB video file over network connection.

Trial No.	Accuracy (%)	Encryption time (ms)	Decryption time (ms)
1	97.10	5.27	5.04
2	97.54	9.54	9.12
3	97.13	5.55	5.31
4	97.70	11.15	10.67
5	97.79	11.99	11.47
6	97.25	6.69	6.40
7	97.81	12.23	11.70
8	97.62	10.34	9.89
9	97.89	13..02	12.45
10	97.58	9.93	9.50
AVERAGE	97.568	9.571	9.155

VITA

Name of the student	:	Simranjit Kaur
Father's name	:	S. Gajinder Singh Chohan
Mother's name	:	Smt. Paramjit Kaur
Nationality	:	Indian
Date of Birth	:	June 6, 1996
Permanent home address	:	29-BX, Model Town Extension-II, Ludhiana, Punjab, 141001
Email	:	simranjitsvk@gmail.com

EDUCATIIONAL QUALIFICATION

Bachelor's degree	:	B.Tech. (Computer Science and Engineering)
University	:	I.K.G. Punjab Technical University
Year of award	:	2018
% Marks	:	72.2
Master's degree	:	M.Tech. (Computer Science and Engineering)
University and year of award	:	Punjab Agricultural University, Ludhiana
Year of award	:	2020
OCPA	:	8.12
Title of Master's Thesis	:	Security issues and their resolutions in cloud based services