# SECURITY MODEL FOR DATA ON TRANSIT IN MOBILE BANKING APPLICATIONS

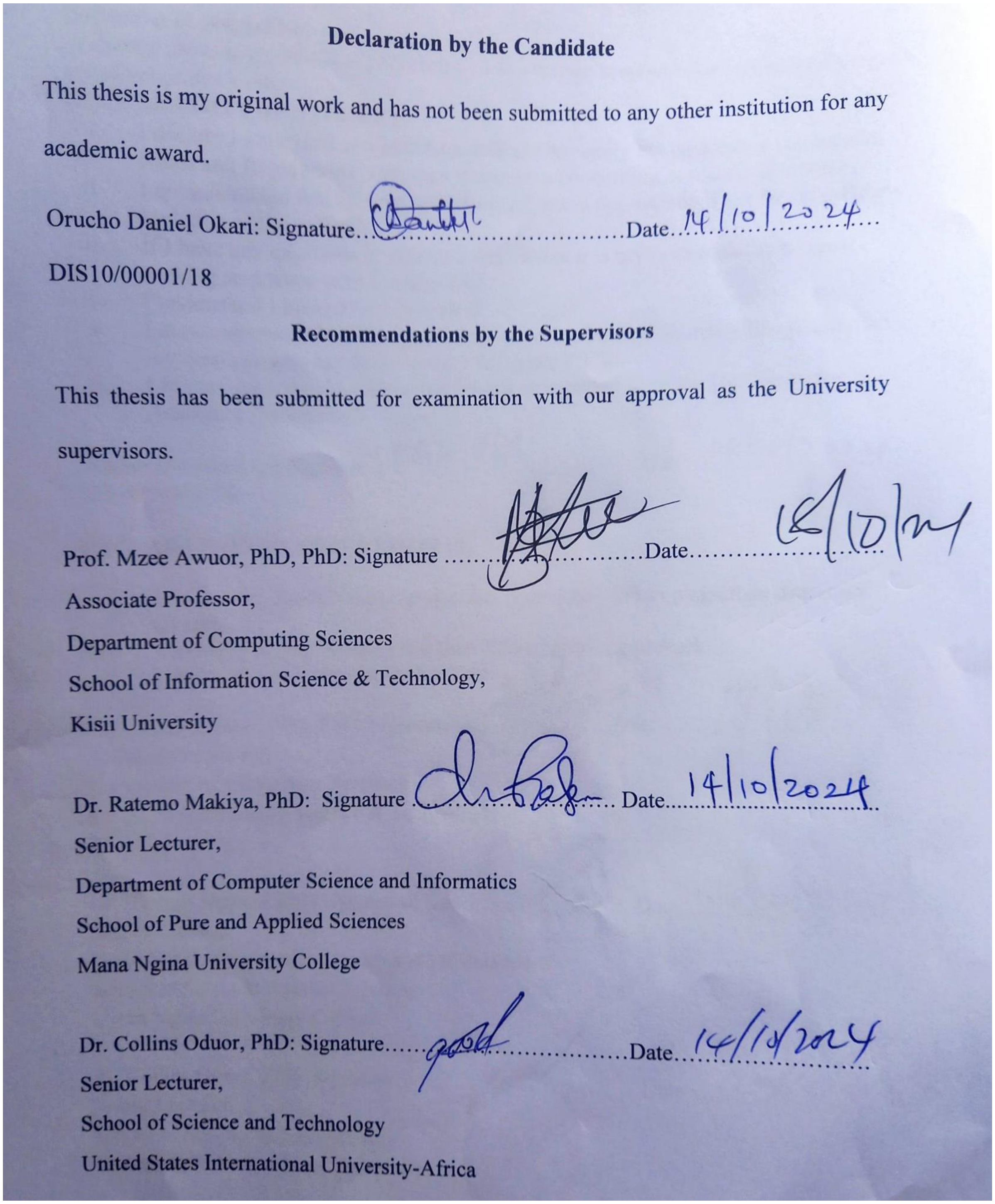
ORUCHO DANIEL OKARI

MBA, MIS (KISII UNIVERSITY), B.ED. (SAINT MARY'S UNIVERSITY OF MINNESOTA),

A THESIS SUBMITTED TO THE BOARD OF POSTGRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF DOCTOR OF PHILOSOPHY IN INFORMATION SYSTEMS, DEPARTMENT OF COMPUTER SCIENCE AND INFORMATICS IN THE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY, KISII UNIVERSITY

2024

# DECLARATION AND RECOMMENDATIONS



## Declaration by the Candidate

This thesis is my original work and has not been submitted to any other institution for any academic award.

Orucho Daniel Okari: Signature……………………………Date 14/10/2024…

DIS10/00001/18

## Recommendations by the Supervisors

This thesis has been submitted for examination with our approval as the University supervisors.

Prof. Mzee Awuor, PhD, PhD: Signature ……………………………Date 18/10/24……

Associate Professor,

Department of Computing Sciences

School of Information Science & Technology,

Kisii University

Dr. Ratemo Makiya, PhD: Signature ……………………… Date 14/10/2024

Senior Lecturer,

Department of Computer Science and Informatics

School of Pure and Applied Sciences

Mana Ngina University College

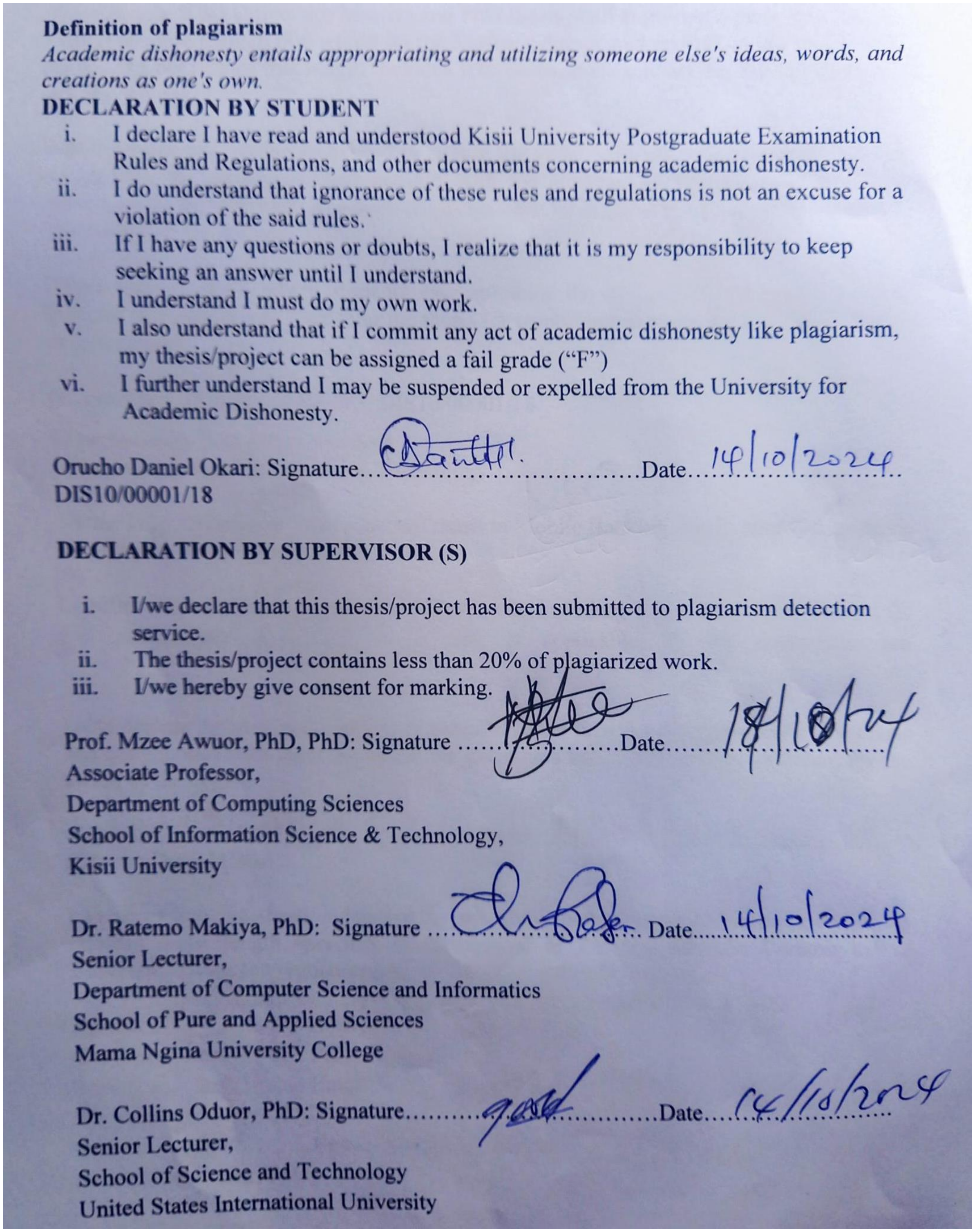
Dr. Collins Oduor, PhD: Signature……………………………Date 14/10/2024

Senior Lecturer,

School of Science and Technology

United States International University-Africa

# PLAGIARISM DECLARATION



**Definition of plagiarism**

*Academic dishonesty entails appropriating and utilizing someone else's ideas, words, and creations as one's own.*

**DECLARATION BY STUDENT**

i. I declare I have read and understood Kisii University Postgraduate Examination Rules and Regulations, and other documents concerning academic dishonesty.

ii. I do understand that ignorance of these rules and regulations is not an excuse for a violation of the said rules.

iii. If I have any questions or doubts, I realize that it is my responsibility to keep seeking an answer until I understand.

iv. I understand I must do my own work.

v. I also understand that if I commit any act of academic dishonesty like plagiarism, my thesis/project can be assigned a fail grade ("F")

vi. I further understand I may be suspended or expelled from the University for Academic Dishonesty.

Orucho Daniel Okari: Signature……………………………Date 14/10/2024
DIS10/00001/18

**DECLARATION BY SUPERVISOR (S)**

i. I/we declare that this thesis/project has been submitted to plagiarism detection service.

ii. The thesis/project contains less than 20% of plagiarized work.

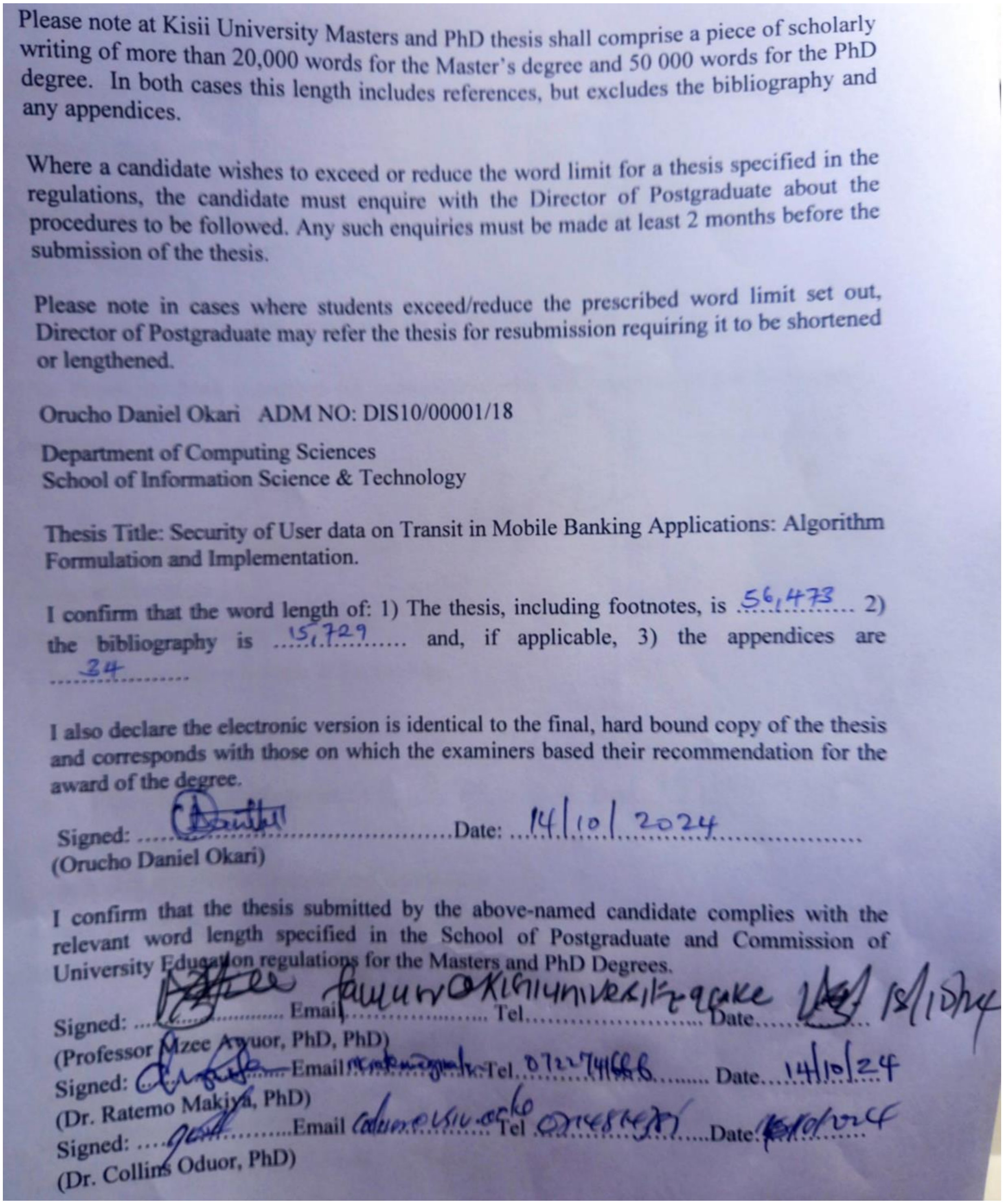
iii. I/we hereby give consent for marking.

Prof. Mzee Awuor, PhD, PhD: Signature ……………………Date 18/10/24
Associate Professor,
Department of Computing Sciences
School of Information Science & Technology,
Kisii University

Dr. Ratemo Makiya, PhD: Signature ……………………Date 14/10/2024
Senior Lecturer,
Department of Computer Science and Informatics
School of Pure and Applied Sciences
Mama Ngina University College

Dr. Collins Oduor, PhD: Signature……………………Date 14/10/2024
Senior Lecturer,
School of Science and Technology
United States International University

# DECLARATION OF NUMBER OF WORDS FOR PHD THESIS



Please note at Kisii University Masters and PhD thesis shall comprise a piece of scholarly writing of more than 20,000 words for the Master's degree and 50 000 words for the PhD degree. In both cases this length includes references, but excludes the bibliography and any appendices.

Where a candidate wishes to exceed or reduce the word limit for a thesis specified in the regulations, the candidate must enquire with the Director of Postgraduate about the procedures to be followed. Any such enquiries must be made at least 2 months before the submission of the thesis.

Please note in cases where students exceed/reduce the prescribed word limit set out, Director of Postgraduate may refer the thesis for resubmission requiring it to be shortened or lengthened.

Orucho Daniel Okari ADM NO: DIS10/00001/18

Department of Computing Sciences
School of Information Science & Technology

Thesis Title: Security of User data on Transit in Mobile Banking Applications: Algorithm Formulation and Implementation.

I confirm that the word length of: 1) The thesis, including footnotes, is 56,473 2) the bibliography is 15,729 and, if applicable, 3) the appendices are 34

I also declare the electronic version is identical to the final, hard bound copy of the thesis and corresponds with those on which the examiners based their recommendation for the award of the degree.

Signed: ........................ Date: 14/10/2024
(Orucho Daniel Okari)

I confirm that the thesis submitted by the above-named candidate complies with the relevant word length specified in the School of Postgraduate and Commission of University Education regulations for the Masters and PhD Degrees.

Signed: ........ Email fawuor@kisiiuniversity.ac.ke Tel........ Date 15/10/24
(Professor Mzee Awuor, PhD, PhD)
Signed: ........ Email [illegible] Tel 0722741666 Date 14/10/24
(Dr. Ratemo Makiya, PhD)
Signed: ........ Email coduor@kisiiu.ac.ke Tel [illegible] Date [illegible]
(Dr. Collins Oduor, PhD)

# COPYRIGHT

# DEDICATION

I dedicate this work to my mother, Sabina Sarange, and my father, Andrew Orucho, whose unwavering examples have inspired me to pursue the goals I have always set for myself. To my sister Everline Kemunto, my brothers David Orucho and Julius Orucho, Abbot Leonard Imai Oese, and my dear friends Fr. Erick Ondieki, Veronicah Hoster, Isaack Okaya Andanje, and my cousin Joakim Okenye – I am deeply grateful for your constant support, encouragement, and belief in me throughout this Ph.D. journey. This thesis is dedicated to you, with my heartfelt appreciation for being part of my life.

# ACKNOWLEDGEMENT

I express my gratitude to God, the source of all wisdom and understanding, for His protection and for granting me physical and mental strength. I extend my deepest thanks to Prof. Mzee Awuor, Dr. Cyprian Ratemo, and Dr. Collins Oduor for dedicating your time and effort to review this thesis and providing timely feedback to enhance its quality. To my closest friends, Fr. Erick, Brother Isaac, and Veronicah, I am forever grateful for your steadfast support.

# ABSTRACT

Mobile banking applications is an advanced technology within mobile banking that exploits the use of wireless and cellular networks to deliver banking services to users with increased convenience round the clock. However, vulnerabilities on this banking channel are utilized by cybercriminals to gain unauthorized access and illegally acquire confidential information of customers to steal money from their accounts. Notably, it is essential to preserve customer data from online cyber thieves. In order to achieve this goal, this thesis sought to assess: operation of mobile banking applications; security threats affecting user-data on transit in mobile banking applications; techniques used to secure user-data on transit in mobile banking applications; develop a hybrid algorithm that ensures secure user-data on transit in mobile banking applications, and evaluate the developed hybrid algorithm for secure user-data on transit in mobile banking applications. Most state-of-the-art techniques have focused mainly on web-based systems rather than mobile banking applications. This study adopted a positivist research paradigm. This study embraced two research designs, that is, descriptive and data science methodologies. Secondary data was gathered from peer reviewed published journal articles, conference proceedings and books. Image processing dataset from The University of Southern California Signal and Image Processing Institute database was utilized to obtain high quality pictures for the hybrid algorithm test simulations. Data analysis was performed through several methods, including content analysis, gap analysis, visual quality analysis, entropy analysis, and statistical analysis. Inferences from the research were illustrated objective-wise, figures and tables. Simulations were carried out on Matlab (R2021a) software using six color pictures. Inferences from this study revealed that operation of mobile banking applications require registration and configuration of mobile banking applications while Transport Layer Security initiates a safe interconnection that links client application and the banks' server. Threats to user data in mobile banking applications are mobile malware, packet sniffing attacks, Man-In-The-the-Middle attacks, Domain Name System poisoning attacks, Secure Sockets Layer strip session hijacking, eavesdropping attacks, Denial of Service attacks, and social engineering attacks. These security threats can be alleviated by utilization of cryptography, steganography, strong authentication, strong antivirus software, and user education. Development of the proposed hybrid algorithm assimilated Least Significant Bit steganography and Advanced Encryption Standard algorithm. Evaluation of the proposed hybrid algorithm was done using Mean Squared Error, Peak Signal-to-Noise Ratio, Histograms, and entropy. The hybrid algorithm exhibited low MSE values between 0.0001297 to 0.0005646 and high PSNR values of 80.65 to 87.04 decibel. Entropy values were between 6.295 and 7.762 inferring the developed hybrid algorithm was robust against MITM attacks for user-data on transit in mobile banking applications. Histograms analysis showed no conceivable differences between cover and Stego-pictures. This study recommended utilization of LSB-AES on transit user-data protection algorithm to fortify mobile banking applications.

# TABLE OF CONTENTS

**CHAPTER TWO**

**CHAPTER THREE**

**CHAPTER FOUR**

CHAPTER FIVE

CHAPTER SIX

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF APPENDICES

# LIST OF ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| **AES:** | Advanced Encryption Standard |
| **AI:** | Artificial Intelligence |
| **API:** | Application Programming Interface |
| **ARP:** | Address Resolution Protocol |
| **AU:** | Audio Units |
| **BHIM:** | Bharat Interface for Money |
| **CA:** | Certification Authority |
| **CIA:** | Confidentiality, Integrity & Availability |
| **COBIT:** | Control Objectives for Information and Related Technologies |
| **COSO:** | Committee of Sponsoring Organizations |
| **COVID-19:** | Coronavirus Desease-19 |
| **CPU:** | Central Processing Unit |
| **CRT:** | Chinese Reminder Theorem |
| **DDoS:** | Distributed Denial of Service |
| **DES:** | Data Encryption Standard |
| **DHCP:** | Dynamic Host Configuration Protocol |
| **DLP:** | Discrete Logaritm Problem |
| **DMZ:** | Demilitized Zone |
| **DNA:** | Deoxyribonucleic acid |
| **DNS:** | Domain Name System |
| **DSA:** | Digital Signature Algorithm |
| **DSRM:** | Design Science Research Methodology |
| **DSS:** | Digital Signature Standard |
| **DTMF:** | Dual Tone Multifrequency |
| **DoS:** | Denial of Service |
| **ECC:** | Elliptic Curve Cryptography |
| **ECDH:** | Elliptic Curve Diffie Hellman |
| **ECDSA:** | Elliptic Curve Digital Signature Algorithm |
| **EHMM:** | Enhanced Hidden Markov Model |
| **EMBASB:** | European Mobile Banking Application Security Benchmark |
| **EV-SSL:** | Extended Verification Secure Socket Layer |
| **FTP:** | File Transfer Protocol |
| **GF:** | Galois Field |
| **GSM:** | Global System for Mobile Communication |
| **HOIC:** | High Orbit Ion Cannon |
| **HTTP:** | Hyper Text Transfer Protocol |

| | |
|---|---|
| **HTTPS:** | Hyper Text Transfer Protocol Secure |
| **HSTS:** | Hypertext Transfer Protocol Strict Transport Security |
| **ID:** | Identification Document |
| **IEC:** | International Electro technical Commission |
| **IMEI:** | International Mobile Equipment Identity |
| **ISDN:** | Integrated Services Digital Network |
| **IPSEC:** | Internet Protocol Security |
| **ISAKMP:** | Internet Security Association and Key Management Protocol |
| **ISDN:** | Integrated Services Digital Network |
| **ISO:** | International Organization for Standardization |
| **IoT:** | Internet of Things |
| **J2ME:** | Java 2 Micro Edition |
| **KLT:** | Kanade-Lucas-Tomasi |
| **LAN:** | Local Area Network |
| **LDAP:** | Lightweight Directory Access Protocol |
| **LSB:** | Least Significant Bit |
| **MAC:** | Media Access Control |
| **MATLAB:** | Matrix Laboratory |
| **MD:** | Message Digest |
| **MFA** | Multi-Factor Authentication |
| **MNO:** | Mobile Network Operator |
| **M-PIN:** | Mobile Banking Personal Identification Number |
| **MP3:** | Moving Pictures Group 3 |
| **MSB:** | Most Significant Bit |
| **MSE:** | Mean Squared Error |
| **NTP:** | Network Time Protocol |
| **OSI:** | Open Systems Interconnection |
| **OTP:** | One Time Password |
| **OWASP:** | Open Web Application Security Project |
| **PIN:** | Personal Identification Number |
| **PKI:** | Public Key Infrastructure |
| **POP3** | Post Office Protocol (Version 3) |
| **PSNR:** | Peak Signal to Noise Ratio |
| **PVD:** | Pixel Value Differencing |
| **QR:** | Quick Response Code |
| **RAM:** | Random Access Memory |
| **RC:** | Rivest Cipher |
| **RGB:** | Red Green Blue |
| **ROI:** | Region of Interest |
| **RNS:** | Residual Number System |

| | |
|---|---|
| **RSA:** | Rivest Shamir Adleman |
| **SAT:** | Sim Based Applications Toolkit |
| **S-Box:** | Substitution Box |
| **SDK:** | Software Development Kit |
| **SFA:** | Single Factor Authentication |
| **SHA:** | Secure Hash Algorithm |
| **SIP:** | Session Initiation Protocol |
| **SMS:** | Short Message Service |
| **SMSC:** | Short Message Service Center |
| **SMTP:** | Simple Mail Transfer Protocol |
| **SNR:** | Signal-to-Noise Ratio |
| **SQL:** | Structured Query Language |
| **SQLI:** | Structured Query Language Injection |
| **SSL:** | Secure Socket Layer |
| **TCP:** | Transport Control Protocol |
| **TFA:** | Two-Factor Authentication |
| **TIFF:** | Tag Image File Format |
| **TN:** | Transaction Number |
| **TTL:** | Time to Live |
| **UDP:** | User Datagram Protocol |
| **USB:** | Universal Serial Bus |
| **USSD:** | Unstructured Supplementary Service Data |
| **UPI:** | Unified Payment Interface |
| **USC-SIPI:** | University of Southern California Signal and Image Processing Institute |
| **VLSI:** | Very Large-Scale Integration |
| **WAP:** | Wireless Application Protocol |
| **WAV:** | Waveform Audio File |
| **WEP:** | Wired Equivalent Privacy |
| **WIM:** | Wireless Application Identity Module |
| **WIM:** | Wireless Markup Language |
| **WPA:** | Wi-Fi Protected Access |
| **WPKI:** | Wireless Application Protocol Public Key Infrastructure |
| **WTLS:** | Wireless Transaction Layer Security |
| **XSS:** | Cross Site Scripting |

# CHAPTER ONE

# INTRODUCTION

This chapter introduces the study and the problem solved. Specifically, it captures the background of the study, statement of the problem, objectives of the study, research questions, significance of the study, assumptions of the study, scope of the study, limitations of the study, and justification of the study.

## 1.1 Background of the Study

Mobile banking is the process of carrying out financial transactions pertaining to a client's bank account using portable devices like smartphones. Among these services are balance checks, bill payments, money transfers, and cheque requests (Raharja & Tresna, 2019). Other services offered by mobile banking are reservations, loan repayments, and airtime top-ups (Malaquias & Silva, 2020).

Banks develop mobile banking applications as client applications to make it easier for customers to access their accounts and make remote service requests to the bank server. Customers can easily access their bank accounts and conduct financial transactions while on the go with the help of mobile banking applications (Mohammadi, 2015). Furthermore, the introduction of mobile banking applications has stimulated the market for mobile banking by improving the delivery of financial services online (Shahid, Islam, Malik, & Hasan, 2022).

Customers download and install mobile banking applications on their smartphones from application stores like Apple Store, Google Play, and Amazon Store to mention a few

(Farooqi, Feal, Lauinger, McCoy, Shafiq, & Vallina, 2020: Venkatakrishnan, Kaushik, & Verma, 2020). Once registered successfully, a customer can utilize the client application installed on their smartphone to utilize mobile banking services.

If a customer wishes to log in or request for financial services, then a variety of authentication methods are available, including usernames, passwords, and biometric techniques like fingerprint, iris, and facial recognition among others. Furthermore, these apps often demand a One-Time Password (OTP) for crucial operations like money transfers as an additional degree of authentication (Abuhamad, Abusnaina, Nyang & Mohaisen, 2020). If the bank server successfully verifies login credentials, customers can use mobile banking applications to access banking services.

Applications utilized in mobile banking are in demand lately because of their widespread adoption. For example, in the United States, applications utilized for remote banking clinched at 57 million users in 2019, with 86% of US banks providing remote banking services. Similarly, there were 25 million users of mobile banking applications in 2019 in the United Kingdom (Gbo, 2021). In developing countries such as Turkey, approximately one-third of all mobile banking clients are registered for mobile banking services (Ecer, 2018). In Kenya, Mobile Money company (M-Pesa) saw an increase of registered active M-Pesa users to over 23 million in the year 2020 (OECD, 2020).

Mobile banking adoption has seen an increase of people who utilize this technology. However, a segment of the population still do not adopt the technology for lack of trust and security concerns (Hanif & Lallie, 2021). Due to growth in online fraud, the current authentication methods used in mobile banking are becoming more vulnerable to numerous threats and attacks (Cavus, Mohammed, & Isah, 2023).

Mobile banking applications fall under the category of applications that are most sensitive in regards to data security. An analysis of numerous real-world banking applications shows new kinds of threats and vulnerabilities that are challenging for the current commercial and open-source mobile application security tools such as QiHoo360 (QARK, 2017), AndroBugs (AndroBugs, 2015), MobSF (MobSF, 2017). Challenges to mobile banking applications include mobile malware, fake banking applications, and Man-in-the-Middle (MITM) attacks. Additionally, many banking application-specific threats and vulnerabilities have been witnessed globally (Panda, 2015; Panda, 2016). In developing countries (Africa and South America), the most common threats and attacks are Short Message Service (SMS) interceptions, MITM attacks, unauthorized access, and mobile malware (Castle, Pervaiz, Weld, Roesner, & Anderson, 2016; Kourouma, Warren, Atkins-Ball, JacksonSundhir, Trivedi, & Breaux, 2022). These attacks are designed to capture customers' confidential data, which attackers can then use to infiltrate their bank accounts and steal funds.

Cybercriminals obtain private financial data through a range of attack vectors that exploit vulnerabilities in mobile banking, despite the fact that these applications allow users to conveniently access financial services at any time and from any location. Cyberattacks can take the form of malware (Souppaya & Scarfone, 2013), Phishing (Suzuki & Monroy, 2022; Gupta, Tewari, Jain, & Agrawal, 2017; Conti, Dragoni, & Lesyk, 2016), and MITM (Conti, Dragoni, & Lesyk, 2016).

Mobile banking is susceptible to threats and attacks and as such, banks have experienced large financial losses. Globally, cybercriminals pilfer over USD 114 billion annually through mobile banking. Similarly, the banking industry spends USD 274 billion worldwide to thwart cyberattacks (Acharya & Joshi, 2020). Successful intrusions result in loss of integrity and confidentiality (Prakash, 2023).

User-data on transit in mobile banking refers to information that is being transferred from the mobile banking application to the banks' server and vice versa. Data on transit is more susceptible to threats and attacks than data at rest because of the channel of communication used such as wireless and cellular networks in which software can be used to acquire system privileges while data is being transmitted (Sealpath, 2020).

Data on transit is vulnerable to a number of threats, including packet sniffing, denial of service (DoS), and MITM attacks (Bhattacharya & Reddy, 2022; Kaka, Sastry, & Maiti, 2017; Javeed, Badamesi, Ndubuisi, Soomro, & Asif, 2020). Authentication and encryption algorithms are two methods used in mobile banking applications to secure data while being transferred from the bank's server to the mobile banking application.

### 1.2 Statement of the Problem

Mobile banking applications have become a strategic technological resource for both banks and customers, enhancing customer satisfaction while reducing operational costs. Customers can conveniently access banking services anytime and anywhere via their smartphones, eliminating the need to physically visit the bank and waste time in long queues. Banks benefit from mobile banking by providing remote access to services, improving after-sales services, and optimizing resource management, such as staff, which contributes to lower operational expenses including the cost of renting large space to accommodate many customers physically present at the bank.

Today, rather than using the traditional banking hall, most banking services are offered to clients via their mobile devices. Apparently, the available security techniques used to protect confidential data in mobile banking include utilization of username and passwords, personal identification numbers, two factor authentication, multifactor authentication, and encryption. These techniques support bank customers to login to the banking system in order to access banking services remotely and securely.

Notably, these techniques are vulnerable to various attacks. For example, utilization of username and passwords is susceptible to phishing attacks which involve tricking legitimate users into providing sensitive information related to their accounts by using various methods such as sending emails, SMS, fake notifications on websites, or even creating fake banking applications to gather user login credentials.

The available authentication mechanisms suffer from various attacks. For example, while two factor authentication is susceptible to MITM attacks, forgeries and eavesdropping attacks, multi-factor authentication suffers from shoulder surfing attacks. Finally, while cryptographic algorithms are susceptible to cryptanalysis attacks, steganographic algorithms are susceptible to steganalysis attacks.

Although there are still measures in place to protect consumers' private information when using mobile banking, they are not robust to stop new cybercrime strategies that employ wireless and cellular networks to gain unauthorized access to customers’ bank accounts.

For example, inadequate cryptographic protection was discovered in 90% of evaluated mobile banking applications (EMBASB, 2022).

A study by NerdWallet (2022) established that users of mobile banking applications get worried that their bank accounts can get hacked when accessing remote banking services using mobile banking applications because of lack of robust security mechanisms. Therefore, it is crucial to enhance the current mechanisms for securing access to mobile banking services. This thesis aims to enhance two established techniques, cryptographic encryption algorithm and steganographic algorithm, by developing a hybrid algorithm.

### 1.3 Objectives of the Study

This thesis aimed to assess how mobile banking applications function to deliver services to customers remotely. It also assessed threats that compromise user data during transmission in these applications, as well as techniques that mitigate the threats. Additionally, the focus was on developing and evaluating a hybrid algorithm aimed at securing user data during transmission within mobile banking applications. The specific objectives of this thesis were:

i. To assess operation of mobile banking applications that allow users to access banking services remotely

ii. To assess security threats affecting user-data on transit in mobile banking applications

iii. To assess techniques used to secure user-data on transit in mobile banking applications

iv. To develop a hybrid algorithm that ensures secure user-data on transit in mobile banking applications

v. To evaluate the developed hybrid algorithm for secure user-data on transit in mobile banking applications

## 1.4 Research Questions

i. How do mobile banking applications function to provide consumers with remote access to banking services?

ii. Which security threats affect user-data on transit in mobile banking applications?

iii. What techniques are employed to secure user-data on transit in mobile banking applications?

iv. How can a hybrid algorithm be developed to secure user-data on transit in mobile banking applications?

v. How is the developed hybrid algorithm for secure user-data in transit in mobile banking applications evaluated?

## 1.5 Significance of the Study

This thesis is paramount to bank customers who utilize mobile banking applications to access their funds remotely because user-data on transit and those transactions through mobile banking applications will be secure. Additionally, mobile banking applications provides ease of accessibility on anytime-anywhere availability with the help of mobile telecommunication services.

Findings from this study will be important for mobile application developers because it enriches them with skills on how to develop secure mobile applications with enhanced security features. Knowledge from this thesis is crucial for decision making when developers design applications used for critical services such as mobile banking because security is a fundamental module in those applications.

This study is also important to policy makers such as legislators and policy analysts. Findings from this study will make a substantial contribution in terms of security of mobile banking which has attracted cyber criminals in the recent past in which customers, banks and other government agencies have fallen victims to cyber criminals and money lost. Legislation can therefore evaluate and amend mobile banking policies in support for strong techniques for mobile banking application channel.

This study will be important to policy analysts because it will enrich them with an awareness of secure techniques that can be utilized for remote banking and mitigate incidences of cyber thieves. This will then help policy analysts to raise public awareness and propose solutions in regard to mobile banking and influence the government to take action.

The study's findings will be important to scholars and researchers. Scholars particularly academicians in information security will benefit from the pool of knowledge in this thesis. This knowledge will aid to prepare in-depth content that can be disseminated to leaners and interested parties in the society. To students and other researchers, this study provides a pool of knowledge that can be theoretically and practically implemented to the field of cybersecurity.

## 1.6 Assumptions of the Study

The foundation of this investigation was grounded on the following assumptions:

i. Mobile banking applications are vulnerable to security flaws that adversaries can exploit to unlawfully intercept user data on transit. However, a hybrid algorithm can be created to secure user data during transmission in these applications.

ii. When two algorithms are combined to create a hybrid algorithm, then the security features from the algorithms produce a robust system that can be tested and implemented in the available testing environment.

iii. Hybrid algorithms can be utilized to secure user-data on transit in mobile banking applications. In order to evaluate their robustness, there are available evaluation metrics that can be utilized.

## 1.7 Scope of the Study

This research focused on mobile banking applications, primarily on how they function to enable users to access banking services remotely, security threats that impact user data while on transit, and methods for securing user data while on transit. Additionally, the study developed and evaluated a hybrid algorithm that guarantees secure user-data on transit.

Both a descriptive and data-driven research design were employed in this study. Descriptive research design was used to accurately describe how mobile banking applications that allow users to connect to banking services remotely operate, evaluate security risks affecting user data in transit inside these applications, and ultimately evaluate methods used to secure user data in transit within these applications. Data science research design which is primarily concerned with design of artifacts and algorithms, was applied in this study, in order to develop and evaluate a hybrid algorithm for secure user-data on transit in mobile banking applications.

Secondary data was gathered via desktop research from reliable academic sources such as books, conference proceedings, and referenced journal articles retrieved from databases such as Association for Computing Machinery (ACM), Springer link, Emerald insight, Institute of Electrical and Electronics Engineers (IEEE), and Google scholar.

The proposed hybrid algorithm's results were centered on statistical analysis, including histogram analysis, and visual analysis, including Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and entropy analysis. These metrics were chosen for analysis because they are the best metrics that can be utilized for Least Significant Bit (LSB) steganography technique when embedding text messages to cover images. Steganalysis on the developed hybrid algorithm focused on visual analysis and simulated MITM attack to test if the developed algorithm is robust.

## 1.8 Limitations of the Study

This thesis did not focus on other data states, such as data at rest, and was limited to creating and assessing a hybrid algorithm for the safe transfer of user data in mobile banking applications. This is because data on transit can be intercepted and used by an adversary to access banks server remotely and siphon customers' money.

Two algorithms used in the study were Advanced Encryption Standard (AES) and LSB. These two algorithms when combined offer an extra layer of security than when any one of them is utilized individually (Biswas, Gupta, & Haque, 2019). This study utilized University of Southern California-Signal Image and Processing Institute (USC-SIPI) dataset and did not concentrate on other datasets such as ImageNet because SIPI dataset is widely used for quality analysis of images and provides standard images for image processing.

## 1.9 Justification of the Study

In order to retrieve users' banking requests, mobile banking applications frequently contact the bank server over wireless and cellular networks. Because of this, users of mobile banking applications place a high importance on the security of user-data on transit. Cybercriminals profit from mobile banking applications' security weaknesses to intercept and steal confidential information because these applications are vulnerable to social engineering attacks, DoS attacks, MITM attacks, and eavesdropping among others. Since users must have constant and secure access to their bank accounts, it is crucial to safeguard customer confidential information while on transit in mobile banking applications.

The Business Daily (2021) reported that KES 24.4 million was stolen from NCBA bank and a conspiracy to defraud another KES 190 million was envisaged by two adversaries through hacking NCBA banking system. Similarly, BBC news in 2017 reported that Kenya Revenue Authority (KRA) system was hacked resulting to loss of $39 million in which the adversary interfered with KRAs computer systems. According to the cybercrime unit, Kenya lost $165 million through hacking in 2016 (Standard Newspaper, 2017).

Globally, numerous financial institutions have faced cyber-attacks. In 2012, banks such as Bank of America, JPMorgan Chase, U.S. Bank, and Citigroup were targeted with the intent to disrupt services and cause financial losses. In 2014, Bank of America experienced a Distributed Denial of Service (DDoS) attack, allowing hackers to gain remote control over the bank’s computers and servers. These incidents led to client frustration, financial losses, and data theft (Tariq, 2018).

# CHAPTER TWO

# LITERATURE REVIEW

## 2.1 Overview

This chapter presents relevant literature related to mobile banking. Specifically, it captures platforms utilized for mobile banking, security of mobile banking, data security in mobile banking applications, and headings derived from the study's objectives such as operation of mobile banking applications, threats affecting user-data on transit in mobile banking applications, securing user-data on transit in mobile banking applications, evaluation of hybrid algorithms, information theory, and summary of research gaps.

## 2.2 Mobile Banking

Utilization of mobile applications to conduct banking is growing exponentially. For example, in the US, applications for mobile banking are among the three most popular utilized applications. Further, more than 2 billion individuals across the world utilized their mobile phones to conduct mobile banking by the year 2021 (Citi, 2018). Additionally, reports indicate that more bank customers are using mobile banking instead of electronic banking (Juniper Research, 2019). While checking account balance is the most frequent activity, users frequently use mobile banking applications to pay bills and for fund transfer.

In the late 1990s, PayBox international company cooperated with Deutsche bank to institute mobile banking in Europe (Shaikh & Karjaluoto, 2015). This method was initially created in European nations to provide basic utilities like viewing account balances and locating the closest Automated Teller Machines (ATMs) on their bank's websites or mobile applications. This was because bank websites could only be viewed via unfriendly web browsers with limited functionality, sluggish screen refresh rates, and limited features. Due to this, adoption rates were low (Cleveland, 2016).

Over time, mobile banking quickly became a great avenue for providing a range of interactive mobile banking services (Foroughi, Iranmanesh, & Hyun, 2019). These services include real-time bill payment via smart devices, cash transfer, and balance inquiries (Shaikh & Karjaluoto, 2015). Cheque requests, loan repayment details, insurance services, and bookings are some other services provided by mobile banking. With increased mobile banking services, the number of people using remote banking increased dramatically.

The Chinese industrial and commercial bank begun offering mobile banking with over 68 million users adopting the technology in the year 2020 (Statista, 2020). The amount of United Kingdom residents who used mobile banking increased from 73% in 2019 to 76% in 2020 (Statista, 2020). The constant increase in remote banking has been caused by the expansion of mobile banking services, increased smart phone usage, and availability of internet in these areas (Statista, 2020). Moreover, 24/7 ease of mobile banking services and improved service quality have been linked to higher adoption rates in these locations (Owusu, Kwateng, Atiemo & Appiah, 2019).

Garanti Bankasi bank first offered mobile banking to developing countries like Turkey in 2004. At the time, a third of all mobile clients had registered for mobile banking (Ecer, 2018). Mobile banking was introduced in Kenya in 2004 by Co-operative Bank (Co-Op

Bank), which launched an innovative system of banking at the time (Weetracker, 2021). This was followed by the launch of M-PESA by Safaricom in March 2007, which significantly expanded mobile banking services in the country (Munga, 2010).

According to KCB Capital Company records in 2020, there were over 23 million active M-Pesa users. The service enables users to make deposits into accounts, send money to other users, pay service providers and suppliers on their mobile devices, and redeem deposits for cash (OECD, 2020). From this period, the technology expanded beyond Kenyan borders.

Banks and customers both benefit from mobile banking. Banks gain from cost reduction, which lowers the number of branches and employees and boosts profitability. Additionally, mobile banking enables users to access financial data and utilize services like balance checking, money transfers, bill payment, and financial management anytime, anywhere using their mobile devices. Therefore, mobile banking helps clients save time and money by minimizing the amount of time spent traveling to the physical bank and waiting to be served by bank tellers (Zhou, 2018). Notably, a variety of platforms can be used to deploy mobile banking services as is detailed in the following section.

### 2.2.1 Interactive Voice Response

A system known as Interactive Voice Response (IVR), enables automated computers to communicate with customers of a business using vocalized tones. Customers can get solutions to frequent questions by using IVR. Customers can engage with the use of an IVR using both dynamically generated and pre-recorded audio. Most client questions can be resolved via IVR systems. IVR systems fall into one of two categories: content-based

IVR, which retrieves data direct access from a database and providing it to users. Thirumaran, Soni, and Gayathry (2015) point out that the program employs an internal Application Programming Interface (API) to get and show database results. The biggest drawback of this technique is its inability to generate dynamic IVR invocations. The next version utilized interlacing or webbing amenities to pick up data from databases. Web amenities do not rely on built-in APIs because they are machine independent. Following the receipt of an IVR request, the system constructs a web service using the parameters and calls it. The IVR system receives the information that the web service has obtained from the database.

The main benefit of IVR services is that it gives the system the composition of web services based on user requests. IVR programs are frequently used in financial institutions to collect consumer individual banking information. The primary drawbacks of IVR systems include their static nature, lack of customer knowledge, inability to display menus or alternatives tailored to the user's needs and circumstances (Thirumaran, Soni, & Brendha, 2015).

### 2.2.2 Short Message Service

Short Message Service (SMS) communication is the motivation behind the earliest innovation of mobile banking. To send an SMS to a client's cell phone or answer a client's demand, a basic number of SMS APIs were generated. As an illustration, a client balance query produces and sends SMS to the bank. The required data is subsequently communicated back through an SMS response, which can hold up to 160 characters (Bojjagani & Sastry, 2017).

The sent SMS is kept in the customer's cell phone inbox and can be accessed by unauthorized users to reveal their mobile banking PINs. Prior to being forwarded to the bank, a client's SMS center where SMS is stored may allow the mobile operator to follow customer activity and expose private information to unauthorized employees. Numerous safe measures such as one-time password (OTP) generation, username and password, have been put in place to protect SMS banking from assaults. However, it has been discovered that these mechanisms pose significant security risks such as interception of SMS since it is sent in plaintext (Bojjagani & Sastry, 2017).

### 2.2.3 Unstructured Supplementary Service Data

This is a menu-driven variant of SMS whereby customers receive text menu in a list. This technology is actually a channel in which 160-character long messages can be sent between mobile devices and the network. Unstructured Supplementary Service Data (USSD) is based on sessions and allows interactive communication between the user and a particular group of applications, in contrast to SMS, which is a store-and-forward method (Sanganagouda, 2011; Date, Waghmare, Sharma, & Chavan, 2017).

To safeguard consumer data, USSD creates a solitary USSD session between the device and software at the network operator, processor, or bank. To put it in another way, the transaction is completed immediately and is not kept for later utilization. The risk of internal data misuse can be reduced by encrypting the data as soon as it reaches the bank or network operator's USSD gateway. The main danger is that the communication layer's data is not encrypted itself. Data can be accessible to someone who could defeat the GSM encryption (Sanganagouda, 2011; Date, Waghmare, Sharma, & Chavan, 2017).

### 2.2.4 Wireless Application Protocol

Wireless Application Protocol (WAP) banking first arose following SMS banking. Bank clients could use Internet connectivity on their mobile devices to access their bank accounts. A mobile phone with WAP functionality is required for the user to access services via WAP for any bank. The user only has to reimburse the traffic that the cell phone service provider produces as a result. Therefore, using WAP banking involves either directly accessing banking services via mobile internet or indirectly accessing them via custom-installed mobile applications that mobile device is linked via the internet (FinMark Trust, 2007).

Protocols and standards are used to make up a WAP security system (FinMark Trust, 2007). Wireless Transport Layer Security (WTLS) protocol was created specifically for portable devices with constrained hardware capabilities. It utilizes Transport Layer Security (TLS) technology. WTLS's primary objective is authentication, data integrity, and privacy between two communicating applications. It offers new capabilities such datagram support, dynamic key updating, and an improved handshake in addition to functionality that is identical to TLS 1.0. It bears networks, which have comparatively high latency and minimal bandwidth, and therefore designed for the WTLS protocol.

To transport data via a wired network, the gateway binary data received from a mobile device must first be decrypted, and then it must be re-encrypted using SSL. As a result, end-to-end security cannot be established while converting WTLS to Hyper Text Transfer Protocol (HTTP) (FinMark Trust, 2007). WAP gateway is vulnerable to attacks from both insiders and outside hackers.

Three models are applied to guarantee complete security and prevent exposure of translating issues: Internet service providers are the realm of providing the gateway network security. Firstly, the service provider is in charge of keeping the gateway secure. By specifying the gateway's Internet Protocol (IP) address, the client can establish a direct connection. However, this approach is not yet enabled. Secondly, using the SSL protocol throughout the entire communication process ensures end-to-end security. Thirdly, Internet Protocol version 6 (IPv6) is employed. However, it is efficient at integrating wireless technologies and direct connections between end users (Hayikader, Hadi, & Ibrahim, 2016)

### 2.2.5 Mobile Banking Applications

Mobile banking applications need a certain level of security while sending data over a network. This degree of security is used to describe utilization of hardware and software resources to protect programs so attackers cannot take over these programs and change them in a way that is advantageous to them. Attacks may be initiated by posing as a trusted user, and if the system views them as such, it may provide the attacking party complete access, resulting in victimization. This may be brought on by outdated network-level security regulations that restrict access to a given IP address to authorized users. These security measures are no longer relevant due to technological advances (Bhadauria & Sanyal, 2012).

Each bank offers a particular type of banking protection to safeguard its customers from unauthorized access to data. The most popular technique is to provide an additional layer of defense by combining the user identification number (ID), default password, and transaction number (TN) such as OTP. There are various ways to produce TN, such as SMS, lists of access codes, and security tokens (Hayikader, Hadi, & Ibrahim, 2016).

## 2.3 Security of Mobile Banking Applications

Customers of banks can use their smartphones or tablets to remotely access their banks via mobile banking applications. Bill payment, money transfers, transaction and balance checks, security alerts and reminders, and bill payment are some of the most popular features offered by mobile banking applications (Kavitha, 2015). Bank customers who utilize applications on their phone to conduct banking services remotely are assured that it is a safe technology and would be more likely embrace the system and have trust in mobile banking, both of which, as some statistical data indicates, are difficult tasks.

Studies have indicated that 79% of consumers in Germany utilize mobile banking. However, 77% of the consumers are still worried about mobile banking systems because of the fear of unauthorized access (C-Insights, 2015). This is so because of the following reasons: first, since SMS messages are sent in plaintext, there is no end-to-end encryption technique available in the SMS-based mobile banking channel. At the SMS bank server and Base Transceiver Station (BTS), only the transmission is encrypted, while A5, a family of symmetric stream ciphers is well known for being the encryption method used in Global System for Mobile Communications (GSM) and later technologies (Athidas & Alamelu, 2018).

A5 is known to be a weak encryption method because it utilizes linear feedback shift registers with irregular clocking which makes it susceptible to cryptographic attacks. Other weaknesses of A5 include shorter key length of 54 bits, which can be brute-forced with modern computing power, and passive and active attacks. All these weaknesses led to A5 being considered insecure for protecting sensitive communications (Athidas & Alamelu, 2018).

Additionally, SMS-banking is vulnerable to spoofing attacks, in which attackers use other people's devices to send messages on the network by altering the sender's phone number. Thus, most banks do not use SMS banking because of SMS spoofing (Athidas & Alamelu, 2018).

## 2.4 Data Security in Mobile Banking Applications

Information security encompasses confidentiality, integrity and availability. Other crucial characteristics related to those who access the data include authentication, authorization, and non-repudiation (Pesante, 2017). Implementing a robust set of controls can effectively ensure data security in mobile banking. These controls may take many different forms, including organizational structures, software and hardware functionalities, rules, processes, and procedures. The following are important data security constructs.

### 2.4.1 Confidentiality

Confidentiality is a quality that prohibits user details from being made accessible or revealed to unapproved people, organizations and procedures (Lundgren & Möller, 2017). In mobile banking, it’s crucial to prevent the disclosure of information to unauthorized individuals or systems. For instance, to conduct a transaction while using a credit card online, both the buyer and the merchant must transmit the cardholder's credit number to a network. The system strives to maintain confidentiality by restricting where the card number can be displayed. There is a confidentiality breach if the card number is acquired by an unauthorized party in any way (Ambhire & Telemde, 2011). In mobile banking, there are a number of attacks on confidentiality, including eavesdropping, brute force, and shoulder surfing.

Interception of information such as user PINs and passwords can be used for user impersonation or to engage in unauthorized business attacks in mobile banking using wireless and cellular networks in which plaintext data can be exposed (Talom & Tengeh, 2019). The adversary is capable of guessing the password or PIN needed to access the mobile banking account via brute force attacks. Despite mobile banking being fairly straightforward, these attacks have a higher success rate (Castle, Pervaiz, Weld, Roesner, & Anderson, 2016). Assaults by shoulder-surfing are attacks on confidentiality in which adversaries merely glance over the victim's shoulder as they do business to obtain PINs and confidential information (Gwahula, 2016). Encrypting data during transmission is one of the efficient modern methods that can be utilized to safeguard data security (Hayikader, Hadi, & Ibrahim, 2016).

### 2.4.2 Integrity

The concept of data integrity states that information must be correct, complete, and legitimate and cannot be changed or tampered with during transaction or the transmission period (Sarfraz, Alsoraya, AlBathali, & Al-Mayyas, 2021). Inability of data to be arbitrarily altered and replaced is a component of data integrity (Mollah, Azad & Vasilakos, 2017). For instance, if a consumer uses the mobile banking application to pay bills, someone could hijack the account and adjust the bill without consumer consent. Considering dependability of mobile banking activities and correctness of financial dealings, data integrity improves trust in mobile banking (Shayganmehr & Montazer, 2021).

During mobile banking transactions, data integrity protects all the unique confidential information secured. In order to be sure that the data and devices have not been altered or compromised, valid customer credentials should be acknowledged. When the correctness, dependability, and system are guaranteed and unauthorized change is avoided, the integrity of the data is safeguarded (Khidzir, Daud, Ismail, Ghani, & Ibrahim, 2018). The integrity of consumer information is jeopardized when it is accessed and changed in mobile banking.

Integrity is subject to three different forms of attacks; Salami strikes, insider attacks, and MITM. Salami attacks, or salami slicing attacks, are a form of cybercrime where perpetrators steal tiny, seemingly trivial amounts of money or data incrementally. These small thefts often go undetected individually but can add up to substantial losses or significant data breaches over time (Altwairqi, Alzain, Soh, Masud, & Al-Amri, 2019).

Attacks by MITM is when adversaries place themselves in between mobile banking application and the bank server to intercept and change traffic before it reaches the intended recipient. Malicious software can be installed by MITM on mobile banking applications which allow them to take money out of users' accounts and deposit it into their own accounts (Altwairqi, Alzain, Soh, Masud, & Al-Amri, 2019).

Insider attacks in mobile banking occur when individuals within an organization, such as employees or contractors, misuse their access to steal sensitive information or commit fraud. These insiders have legitimate access to the organization’s systems and data, making it easier for them to bypass security measures and carry out malicious activities. Such attacks can lead to significant financial losses, data breaches, and damage to the organization’s reputation (SentinelOne, 2019).

The two common hashing algorithms, secure hash algorithm (256 and 512 bits) can be used to ensure data integrity (Akinyede & Esese, 1017). A hash function uses a noninvertible compression function to translate any length of input into a certain length of output. Hash functions are used to provide adequate security services like data integrity since they have the attribute of being difficult to reverse. They are utilized in applications like e-commerce, digital signatures, and online transactions. Hashing operations are more effective when compared to cryptographic ciphers such as symmetric and asymmetric ciphers (Al-Riyami, Zhang, & Keane, 2016).

Passwords can also be utilized to help prevent unauthorized access to smartphones, ensuring the integrity and accuracy of the stored data. Passwords should not be kept in your device in order to avoid interception by adversaries. A smart phone should also have anti-malware software installed to prevent attacks and data from being accessed (Nikam, Priyanka, Vakhariya, Mohite & Magdum, 2020). Data integrity in mobile banking applications can also be strengthened by the use of encryption and digital signatures (Jibril, Kwarteng, Chovancova, & Denanyoh, 2020).

### 2.4.3 Availability

Availability states that owners of data should have unrestricted access to the information they require (Shukla, 2021). Additionally, availability demonstrates the system is dependable by guaranteeing only permitted users to log on to the system at any moment and from anywhere (Chaimaa, Najib, & Rachid, 2021). So, availability is when a customer can conveniently execute mobile banking transactions whenever they need (Kang, 2018). Data availability has consequences on dissemination of mobile banking facilities (Carranza, Dáz, Sánchez-Camacho, & Martn-Consuegra, 2021) because of the following scenario.

When an adversary intentionally suspends the mobile banking users' bank server or application server, it is considered availability attack. In this form of attack, adversaries use a variety of strategies, including mobile theft, among other assaults to make services inaccessible. When adversaries transmit bogus traffic to overwhelm servers and obstruct user requests, Distributed Denial of Service (DDoS) and Denial of Service (DoS) attacks are initiated. These assaults aim to overwhelm the bank server with several fictitious requests with an aim of rendering it unable to recognize and differentiate distorted from genuine user demands rendering unavailability of mobile banking services (Ali, Dida, & Sam, 2020).

Attacks involving stolen mobile phones are a type of availability attack whereby a victim's mobile device is taken, Subscriber Identity Module (SIM) card's wallet account rendered inaccessible so the attacker can switch it. Attacks involving phone theft can result in data access since attacker will take control of the account of the victim's e-wallet, rendering it unavailable to legitimate account user. Backup servers are utilized to guarantee that mobile banking services are accessible to consumers (Ali, Dida, & Sam, 2020).

### 2.4.4 Authentication

Authentication is a method of verifying a device's identification before allowing usage of a product or service (Belas, Koraus, Kombo, & Koraus, 2016). When communicating across a network, authentication also refers to the verification of the sender and the recipient (Guo, Campbell, Kapadia, Reiter, & Caine, 2021). Entity authentication and data authentication are the two principal types (Chaimaa, Najib, & Rachid, 2021). These forms of authentication are important in mobile banking because it provides secure access to authorized users, builds trust between the bank and its customers, helps financial institutions comply with regulatory requirements, and helps prevent identity theft. Whereas data verification concerns the accuracy of data used in entity interactions, entity authentication concerns the verification of the entities engaged in a communication (Al Farawn, Rjeib, Ali, & Al-Sadawi, 2020).

### 2.4.5 Authorization

When information is accessed, authorization guarantees that the user has the right to see, modify, and make use of the information (Kizza, 2020). Additionally, it indicates the authority to address user requests within a system. Permission is a major criterion impacting consumers' confidence in mobile banking facilities (Ghali, 2021). Policies governing system access are viewed as authorisation security guidelines in the context of mobile banking (Mogos & Jamail, 2021).

### 2.4.6 Non-repudiation

Non-repudiation in mobile banking is a security measure that ensures a party involved in a transaction cannot deny the authenticity of their actions or the content of the transaction. This means that once a transaction is completed, neither the user nor the bank can dispute the validity of the transaction. Non-repudiation is typically achieved through mechanisms like digital signatures or unique transaction IDs, which provide cryptographic proof of the transaction's authenticity (Fang, Chen, Zhang, Pei, Gao & Wang, 2020; Mohamed, 2020; Wang & Long, 2020).

## 2.5 Operation of Mobile Banking Applications

Mobile banking makes use of client-side and server-side technologies. Applications created using technologies used on servers are installed on a bank server rather than on a customer's Subscriber Identity Module (SIM) card or mobile device. In the server-side applications, client data necessary for transaction processing (account or card details) is kept in a bank's server or designated service company in a highly secure environment. IVR, USSD, WAP, SMS, and other server-side technologies are a few examples. Client-side technologies include programs, services, and other tools created for or integrated into a customer's SIM card or mobile device.

Thus, client-server computer architecture utilizes interconnection between mobile client application and bank servers. The server has more computing power and resources compared to a mobile device, therefore as a result of resource restrictions, the smart phone application serves as the client. The client-server connection is a result of the disparity in processing capability. On the network, the application server keeps an eye on new requests. An appeal is subsequently interpreted by the server into an understandable format. The received request is then examined for security flaws using a specified protocol before being carried out. The bank or a third-party application vendor can find and manage the application server (technopedia, 2022). To maintain data integrity and confidentiality, hash algorithms and message digest are used in the application layer. As a result, they protect applications' interaction and the server (Salihu, Jimoh, Salihu, & Modi, 2024).

For added security, banking systems require users to utilize a second factor of authentication in addition to their login and password to access their accounts remotely. A history of a customer's behavior allows financial institutions to keep track of how customers behave when using their mobile banking applications and build a profile of their interactions. The ability to erase or reset customer data on majority of smart phones and tablets from remotely is another benefit of using mobile applications. Thus, if a mobile device is stolen, the owner can use a computer or any device with an Internet connection to delete all of the device's data and applications, preventing anybody else from accessing the user's account through mobile devices (Symantec, 2012).

The requisite functioning of mobile banking needs user registration and validation in order to access banking services remotely. After registration, an OTP is sent to the registered phone number and is used to set up the administration login details. In addition to usernames and passwords, mobile banking system also supports biometric-based identification using audio, facial, or fingerprint recognition (Kieseberg, Fruhwirt, Schrittwieser, & Weippl, 2015).

Transport Layer Security (TLS) is used to create connections by linking a client and server for safeguarding data sent in the network. It offers security services for all application-based protocols, such as Hyper Text Transfer Protocol (HTTP), File Transfer Protocol (FTP), and Post Office Protocol version 3 (POP3), among others. It features client/server architecture and provides the following communication properties that is, integrity, confidentiality and authentication (Curguz, 2016).

At the heart of TLS is the Handshake protocol, which negotiates the security frameworks registered for transmission of encrypted data as well as authentication of each communication participant. Both parties in the discussion are informed by means of the ChangeCipherSpec protocol that the session's state has been upgraded to negotiated parameters and has switched to secure communication. The Alert protocol is used to notify the parties involved when communication problems occur, such as when a link is lost or a message cannot be encrypted. During the handshake step, all cryptographic primitives required for connection security are created. During the handshake phase, messages with predetermined forms are used to communicate between the client and the server (Curguz, 2016).

### 2.6 Threats Affecting User-Data in Transit in Mobile Banking Applications

Network attacks represent a significant risk to data security because the network is essentially required for data transmission. Adversaries can exploit technological weaknesses and network vulnerabilities to intercept data, alter access rights and obtain or modify data while it is in transit. Both physical layer encapsulation of electromagnetic waves and network connectivity might cause data to leak or be stolen.

The following factors ought to be considered in order to build secure network that can be free from attacks: routine backup data, storage of data on dependable hardware, keep system software up-to-date, install security certificates to protect against attacks, regularly upgrade firewall with Access Control List (ACL), proxy and routers. Any threats to network security that could affect a computer system are considered a threat to network security. Threats to network security might be either active or passive (Patil, 2020).

Active threats include data alterations or modifications during communication and an attacker's attempt to access a computer system without authorization. The user may discover that the attacker is personally involved in these nefarious operations. Examples of active threats include mobile banking Trojans, MITM attacks, phishing attacks, ransomware, data theft and overlay attacks. In passive attacks, the adversary does not alter or adjust the communication during attacks. Examples of passive threats are eavesdropping, traffic analysis and shoulder surfing. File sharing and messaging are two types of communication that an attacker can monitor (Patil, 2020).

Network attacks consume majority of the network bandwidth at the application layer, which subsequently brings down network equipment. User data is lost or deleted during transmission when services fail to respond to user requests, which affects the data's availability and integrity (Zhe, Qinhong, Naizheng, & Yuhan, 2017). The following subsection covers recent threats to data on transit in mobile banking applications.

### 2.6.1 Mobile Malware

Malware refers to malicious code or software, essentially a program secretly embedded within a system with the intent to compromise data by disrupting its integrity, confidentiality or accessibility (Souppaya & Scarfone, 2013). Because malware can interfere with a system's ability to function, it is considered a serious external danger (Abomhara & Koien, 2015; Al-Marghilani, 2021; Souppaya & Scarfone, 2013).

### 2.6.2 Phishing

Phishing involves sending deceptive messages through seemingly trustworthy emails or SMS (Sizuki & Monroy, 2022; Conti, Dragoni, & Lesyk, 2016). These messages often appear legitimate but are designed to lead recipients to harmful scripts or files (Sizuki & Monroy, 2022). Phishing attacks can result in the theft of personal information, such as login credentials and passwords for online accounts. Users may unknowingly disclose this data by clicking on malicious links or responding to fraudulent emails, potentially leading to identity theft (Almomani, Gupta, Atawneh, Meulenberg, & Almomani, 2013).

Phishing attacks have become one of the most serious threats to online security, casing businesses significant financial losses each year (Aleroud & Zhou, 2017). These attacks account for more than half of all cybercrimes targeting legitimate users. Often, phishing involves installing malicious software such as ransomware or Trojan horses, on a user’s system (Phishingpro, 2016). Phishing is particularly effective because many users are unaware of Internet security risks and even experienced users often lack sufficient training on cyberattacks (Arachchilage, Love, & Beznosov, 2016). On smartphones, phishing can involve installation of malicious applications that collect and transmit user's mobile data to cybercriminals (Felt & Wagner, 2011).

### 2.6.3 Man-in-the-Middle

The MITM attack is one of the oldest types of cyber-attacks, where a malicious individual secretly intercepts and manipulates communication between two parties. By creating a secret, pseudo-fake link between the victim's and their computers, this attack allows the adversary to read and change the data that the victim transmits. MITM attacks frequently aim to either impersonate one of the communication parties or steal bank account information, passwords, and personal information. This can lead to unauthorized changes in login details or fraudulent money transfers. Public WI-FI areas are particularly vulnerable to such attacks because unencrypted packets can be easily intercepted. Attackers on these networks redirect traffic through themselves, allowing them to capture and exploit personal data or passwords from unsuspecting users (Conti, Dragoni, & Lesyk, 2016).

### 2.6.4 DNS Poisoning

An exploit that takes advantage of domain name system weaknesses is reflected as a Domain Name System (DNS). These types of threats involve faking DNS or poisoning DNS caches. DNS is widely used to convert domain names into IP addresses and vice versa, as is well known. Establishing trust for TLS requires DNS. Nowadays, it is common to obtain TLS certificates by only demonstrating domain ownership. Therefore, a breach in DNS record integrity can lead to security errors such as issuing of fake certificates, which endanger the basis of public key cryptography (Brandt, Dai, Klein, Shulman, & Waidner, 2018).

Attacks on any of the caches that make up the various layers of the modern DNS infrastructure occur at the application layer and operating system, which are most frequently targeted. The flaws also affect almost all widely used DNS software stacks, such as Bind (Internet Systems Consortium, 2020), Unbound (NLnet Labs, 2020) and dnsmasq (Kelley, 2020), which are built on top of Linux operating systems. A key requirement is that the victim operating system software be able to send outbound Internet control message protocol signals. It is interesting that faults in the user datagram protocol standards or the various implementation details that result in side channels based on global restriction of Internet control message protocol error messages allow for highly definite de-randomization of source port.

Diverse safeguards have been advocated to reduce the risks associated with DNS cache poisoning. They essentially render the initial attack unfeasible. The most popular and used defences against DNS poisoning assaults is described as follows. System administrators must implement domain name system security (DNSSEC). However, only 12% of resolvers that support DNSSEC really make an effort to validate the incoming records (Kelley, 2020).

### 2.6.5 SSL Strip Session Hijacking

A session can be described as involvement of two or more devices sharing information. Each session must have unique session identification in order to be identified. Typically, a server will give its clients session identities. When a session is hijacked, someone who knows the session identity can claim to be an authorized user. The attacker has the same level of getting inside the system as legitimate administrator once session identities are known. The session identities are normally stored in a cookie which is a little piece of personal data. Every time a client visits a website, the web server sends this data. To keep track of a client's behaviour, the browser sends this private data to the server each time a client reloads this webpage. Each time a person visits a website, the browser sends the cookie back to it (Hossain, Paul, & Islam, 2018).

This cookie can readily be sniffed using plain text tools like Wireshark and Ettercap, among others, if a web page uses HTTP (dsniff, 2022; Wireshark, 2022; Ettercap, 2022). HTTPS offers website authentication, privacy protection, and information integrity assurance for sent data (Sarkar & Fitzgerald, 2016). HTTPS guards against MITM attacks and information eavesdropping between the server and the client (Hossain, Paul, & Islam, 2018).

Whenever sensitive data is transmitted over the Internet, TLS ensures its security. Encrypting the data sent between a web server and a web, the client application ensures data security and privacy (e.g. browser). Web URLs protected by SSL start with HTTPS prefix rather than HTTP prefix. Therefore, an SSL-enabled website's web address utilizes *https* rather than the attacker-vulnerable *http*. Detaching this SSL, a session hijacking assault can be initiated.

SSL strip attacks allow hackers to stealthily utilize victims' login credentials and other sensitive financial and personal details. Therefore, hackers can steal a sizable sum of money from their targets. Attacks using SSL strip are created such that the target is unaware of their presence. The following two actions are taken in order to launch a successful SSL stripping attack (Hossain, Paul, & Islam, 2018).

The first action is ARP spoofing, in which the MAC address associated with an IP address is bound by the ARP table. If the attacker discloses his MAC address tied to the gateway IP, then all traffic intended for the network gateway is transmitted through him. Similarly, the attacker can choose to communicate to the victim's IP address and his MAC address to the gateway (Hossain, Paul, & Islam, 2018). As a result, any traffic intended for the victim will pass via the attacker. The victim and the attacker need to be on the same local area network. However, a single networked workstation's weak password or a remote vulnerability could potentially allow an attacker to access a LAN. One popular tool for ARP spoofing is Arpspoof (Hossain, Paul, & Islam, 2018).

The second action is known as DNS spoofing, which involves binding a human-readable web URL to its matching IP address in DNS. The URL binds with the IP address of an attacker's malicious website through DNS spoofing. The attacker must be connected to the same LAN or employ a remote vulnerability, just like ARP spoofing. Dnsspoof is a common utility for DNS spoofing. Additionally, setting up a Wi-Fi hotspot requires the attacker to do so (using a fake hotspot) and then allow victims to connect to it. As a result, all traffic pertaining to the victim will go via the attacker's hotspot. This makes the attacker's hotspot acts as a proxy server. The available tools for this action are Easy-creds, Airodump-ng, and Airbase-ng (Avinash, 2015).

Another action that can be utilized to launch SSL stripping attack is to switch HTTP with HTTPS. It involves redirecting and replacing HTTPS traffic with HTTP. This can be achieved by using Sslstrip tool which was developed by Marlinspike (Prandini, Ramili, Cerroni, & Callegati, 2010).

Session hijacking based on SSL stripping typically utilize two user flaws. Firstly, without specifically going to HTTPS websites for example, instead of using *https://securebank.com* or *https://www.securebank.com*, users frequently go to *securebank.com* or *www.securebank.com*. If the attacker is successful in rerouting traffic, an SSL stripping attack can be carried out with ease. Secondly, people often choose to disregard any browser warnings and visit a website regardless of whether it is susceptible due to their propensity to accept phony certificates. The attacker can easily launch a session hijacking attack by accepting false certificates (Hossain, Paul, & Islam, 2018).

The existing approaches against SSL stripping can be grouped into client-side and server-side (Hossain, Paul, & Islam, 2018). A client-side approach eliminates the need for server involvement. In this category, only the client is responsible to protect itself against SSL stripping attacks. Different client-side approaches include: ARP-related, restricting Internet Control Message Protocol (ICMP) packets and history proxy. Server-side approach includes server involvement. Some of the approaches in this category can further be grouped into HTTP Strict Transport Security (HSTS), two-way authentication and Extended Verification Secure Socket Layer (EV-SSL) certificate.

### 2.6.6 Eavesdropping

Interception of network traffic, whether active or passive, is known as eavesdropping. Without the trusted parties' consent, the attacker listens in on data being sent over the network between the bank server and the mobile banking app. Insecure network communication allows attackers to access sensitive data. When unencrypted data is sent via a communication channel, this usually happens (Talom & Tengeh, 2019).

Unlike passive eavesdropping, where a hacker passively finds the information by listening to the message being transmitted over the network, active eavesdropping requires a hacker to access information by posing as helpful and asking probing questions to transmitters (Talom & Tengeh, 2019).

Adversaries use network sniffer software, formerly known as Ethereal, to intercept data while it is in transit (Mhato, 2015). Ethereal shows all network communication, both wired and wireless. This protocol analyzer is compatible with numerous platforms and protocols. It supports several well-known wireless security protocols for decryption such as Bluetooth and 802.11 (Ndatinya & Xiao, 2015).

The Wireshark application presents acquired data in a logical and readable fashion. The Wireshark program also allows users to develop custom filters in addition to having a large number of built-in filters. These filters can be used to only gather specific data, like port numbers and Internet protocol addresses. Programs that use sniffers can effectively process data that is sent in cleartext. Encrypted data requires an encrypted key cracker. Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA) are two older encryption technologies that may be cracked using a variety of tools, including AirSnort for WEP and AirCrack (Ndatinya & Xiao, 2015).

It is difficult to detect and stop passive eavesdropping because there are no network disruptions or changes. Even so, once network alterations are noticed, data has already been collected.  However, active attacks are simpler to identify. By employing security technologies such as network segmentation, network monitoring, and authentication, as well as by being aware of suggested security procedures like virtual private networks, firewalls, and antimalware software, active eavesdropping attacks can be avoided (Salim, Sagheer, & Yaseen, 2020; Okpara & Bekaroo, 2017).

### 2.6.7 Denial of Service

DoS attacks is defined by Prasad, Reddy, and Rao (2014) as attacks that involve flooding servers, systems, and networks with excessive traffic, overflowing the bandwidth resources available, and preventing the system from handling genuine requests. Additionally, DDoS attacks occurs when multiple computers initiate DoS simultaneously (Gordon & Ford, 2006; Prasad, Reddy, & Rao, 2014). Network management may find it very challenging to pinpoint the attack's origin because DDoS attackers can come from any IP address in the world (Al-Khater, Al-Maadeed, Ahmed, Saliq, & Khan, 2020; Prasad, Reddy, & Rao, 2014).

### 2.6.8 Brute-force and Password Spraying

Brute-force attacks usually target individual accounts, with the attacker methodically trying different passwords to gain access (Goel, Sharma, & Gupta, 2022). This approach often leads to numerous failed login attempts (Goel, Sharma, & Gupta, 2022; Popoutsakis, Fysarakis, Spanoudakis, Ioannidis, & Koloutsou, 2021). However, modern cybersecurity measures are typically equipped to detect such activities and will lock an account after several failed login attempts within a short period (Popoutsakis, Fysarakis, Spanoudakis, Ioannidis, & Koloutsou, 2021).

However, attackers can circumvent cybersecurity standard procedures by using password spraying (Broonkrong, 2021). As a result, the attacker may attempt to log in to multiple user accounts using a range of commonly used passwords. By cycling through one password across several accounts before switching to another, the attacker can avoid triggering standard lockout protocols. This approach allows the attacker to continue testing various passwords on the target accounts (Goel, Sharma, & Gupta, 2022; Popoutsakis, Fysarakis, Spanoudakis, Ioannidis, & Koloutsou, 2021; Broonkrong, 2021).

Password spraying attacks are frequently successful because many users fail to follow recommended password usage standards. Commonly used passwords include recognizable number sequences such as “12345”, popular female names like Jennifer, and the word "password."(Kanta, Coray, Coisel, & Scanlon, 2021; Beno & Poet, 2020).

The use of these and an additional 200 easily guessed passwords has contributed to numerous data breaches (Kanta, Coray, Coisel, & Scanlon, 2021; Beno & Poet, 2020). Consequently, attackers who target a large number of usernames and employ a wide array of common passwords are likely to gain access to some accounts (Abomhara & Koien, 2015; Al-Marghilani, 2021; Goel, Sharma, & Gupta, 2022; Kanta, Coray, Coisel, & Scanlon, 2021; Beno & Poet, 2020).

### 2.6.9 Social Engineering

Social engineering is a serious risk to both clients and banking institutions. Although these risks can be avoided, they might not be recognized. There is a comparable pattern to social engineering despite their distinctions. According to Mouton, Leenen, and Venter (2016), the typical pattern consists of four steps: studying the target, getting to know the target, using what you have learnt to carry out an assault, and leaving no trace behind. Rather than employing technology system attack techniques, opponents conduct research on human behavior to get beyond a bank's security measures (Nerwal, Mohapatra, & Usmani, 2019).

Phishing, spear phishing, scareware, pretexting, and baiting are all common techniques used in social engineering (Nerwal, Mohapatra, & Usmani, 2019). The primary goal of phishing attacks is to fraudulently acquire the targets' private banking information via email or phone calls.

Attackers use a variety of tactics, including websites, emails, advertisements, scareware, antivirus software, PayPal websites, rewards, and freebies, to lure bank customers into divulging private and sensitive information. For example, clicking on email links or getting calls from a fictitious bank department requesting personal information are examples of social engineering behavior attacks. With the use of mobile banking, these attempts are intended to mine crucial, private bank information that may be utilized to access personal accounts (Salahdine & Kaabouch, 2019).

Phishing assaults deceive people into clicking on links to get freebies. Similar to Trojan horses, they start attacks using unprotected computer resources such USB drives or storage media that are hidden away at coffee shops and carry malware. Similar to a real-life Trojan horse, the USB drive infects the victims' PCs when they plug it into system ports. The victims of this attack are unaware of the background operations that are taking place (Costantino, La Marra, Martinelli, & Matteucci, 2018).

Pretexting attacks involve creating fictitious but plausible situations in order to obtain personal information from the victim. They are predicated on explanations that give both the perpetrator and the victim reason to believe in one another. Attackers use physical media, emails, or phone calls to carry out their attacks. Attackers obtain the knowledge they need to carry out their attack from phone directories, public websites, and conferences attended by professionals in the same industry. A pretext might be anything, such as an offer of employment or services, a request for private information, assisting a friend in gaining access to something or even winning the lottery (Ghafir, 2016).

Organizations should plan the most effective ways to prevent these assaults because they pose serious dangers. Banking institutions ought to implement a policy that instructs both employees and clients on the various ways social engineering is spread and how to stop it. These can be in the form of training for staff or by way of short and simplified infographic materials and video to customers.

### 2.6.10 Cross-site Scripting

Cross-site scripting (XSS) is a web security vulnerability that allows attackers to exploit user interactions with inadequately secured applications (Shalini & Usha, 2011; Nithya, Pandian, & Malarvizhi, 2015). This vulnerability in the user's system enables attackers to bypass the same-origin policy that usually isolates different websites (Alhawamleh, 2012). As a result, attackers can impersonate the user, perform actions on their behalf, and access all their data. If the user has privileged access within the application, the attacker could potentially take complete control over its functionality (Shalini & Usha, 2011; Nithya, Pandian, & Malarvizhi, 2015).

### 2.6.11 SQL Injection

Web and other web-based applications have generally been widely used since data is easily accessible. But with more people using the internet, concerns about security vulnerabilities are growing. Everyone who uses the internet wants to protect their personal data from unauthorized access. Data is considered the "new oil" that runs and powers the world (Shachi, Shourav, Ahmed, Brishty, & Sakib, 2021). Web security vulnerabilities erode the quality of data security principles by giving hackers access to any data on the network (Marashdeh, Suwais & Alia, 2021).

Structured Query Language (SQL) injection is a technique for inserting SQL code into dynamic web applications (Tasevski, & Jakimoski, 2020). SQL Injection attacks involve adding code and testing SQL queries in order to navigate a website. Websites can be compromised using this technique. In the event that an unauthorized person gains access to the website, they may be able to read data, delete tables, and make other potentially dangerous acts such as table alterations (Abirami, Devakunchari, & Valliyammai, 2015). The process by which an adversary alters the logic, semantics, or syntax of a SQL query by introducing new SQL keywords or operators is known as Structural Query Language Injection (SQLI) (Tajpour, Heydari, Masrom, & Ibrahim, 2010).

An input string is inserted into the program in this type of attack in order to update or change the SQL query to the attacker's benefit. This attack poses a risk since it has the potential to compromise functionality and confidentiality by causing data loss or unauthorized use.

Additionally, as part of this type of attack, system-level commands are carried out, denying approved users access to the required data. MITM attacks and session hijacking are types of attacks where an outsider could silently seize control of a communication route connecting a number of endpoints. The MITM attacker has the ability to interfere with, alter, or even replace the intended victims' communication flow (Kiat, Obaja, Wei, & Hui, 2017).

By utilizing SQLI, an attacker or unscrupulous person can gain unauthorized access to a website and gather data for malicious purposes. Attackers can create harmful code as a character unit to embed in web applications by using simple SQL queries like SELECT, INSTALL, UPDATE, and MODIFY. SQLI technique allows attackers to get unauthorized access to a website by circumventing the authentication and authorization system for online applications (Bhateja, Sikka, & Malhotra, 2021). Analysis and machine learning techniques can be used to reduce the impact of SQLI attacks (Marashdih, Zaaba, Suwais, & Mohd, 2019; Doukas, Stavroulakis, & Bardis, 2021). This can be achieved through classification algorithms, anomaly detection, behavioural detection, real-time monitoring, feature engineering and hybrid techniques (Demilie & Deriba, 2022).

## 2.7 Securing Data on Transit in Mobile Applications

Cryptographic algorithms are one of the many technologies used in mobile banking to protect user data while on transit (Yadav & Dhankhar, 2015). Other approaches include steganographic algorithms and authentication. Cryptography is a fundamental tenet of information security. It is the study of conveying and acquiring information such that only those who a message is intended may read and process it (Ahmad & Garko, 2019).

The main goal of cryptography is to transform data into an unreadable format so that only authorized users can access it and prevent attackers from intercepting data being transferred (Purnama & Rohayani, 2015; Patil, Narayankar, Narayan, & Meena, 2016).

Cryptographic algorithms are administered in two parts which are encryption and decryption. Encryption involves transforming plain text into non-readable format while decryption reshapes non-readable text into readable format using a cipher. A cipher is a paired algorithm, which formulates enciphering and deciphering procedures. Substantial procedures of encipher are managed by a sequence of steps and a key (Omar, Elsadd, & Guirguis, 2017).

The Caesar cipher, one of the oldest ciphers, was named after Julius Caesar who employed it to communicate with army officials during the Gallic Wars. According to a report from the University of Wisconsin, Caesar was the first individual to use encryption to protect sensitive data. Because his soldiers lacked the education necessary to comprehend the intricate coding scheme, Caesar made the decision to create the simplest substitution cipher that made it possible to secure messages (Shallal & Bokhari, 2016; Sohal & Sharma, 2022).

Transposition and replacement techniques are frequently employed in ciphers to transform data into unreadable form. Caesar cipher replaces every character in the plaintext with a fixed character using a fixed character replacement technique based on changing the initial character's position (Chandra, 2014; Mohammed, Argabi, & Alam, 2019). There are different types of symmetric and asymmetric encryption that are utilized to secure data communication. This classification has been developed based on how each algorithm's key's function.

### 2.7.1 Symmetric Encryption Algorithms

Symmetric encryption utilizes the same private key for both encryption and decryption. A word, a number or an arbitrary collection of letters can serve as the key. Both sender and recipient must have the key in order for a message to be encrypted and decoded successfully (Ahmed & Naeem, 2022).

### 2.7.1.1 Advanced Encryption Standard

The Federal Processing Standards Publications-approved AES (Advanced Encryption Standard) cryptographic technique can be used to secure electronic data (FIPS). Digital data can be encrypted and decrypted using AES. Keys of bit sizes of 128, 192, and 256 can be used to encrypt and decrypt information blocks of 128 bits (NIST, 2022).

AES's state structure, which is used to run complex AES block cipher designs, contains two dimensions of 4 by 4 matrix arrangement bytes. The state array's s is a single byte that comprises indicators: a row of tokens r, which ranges $0 \le r < 4$, and a column of tokens c, which ranges $0 \le c < 4$. Likewise, the representation of a state-specific byte is s_(r,c) or s [r, c]. AES's requirements require that the input byte arrangement be replicated to the 4 by 4 square values of variables shown in (2.1).

$$s\,[r, c] = in\,[r + 4\,c] \quad \text{for} \quad 0 \le r < 4 \text{ and } 0 \le c < 4 \tag{2.1}$$

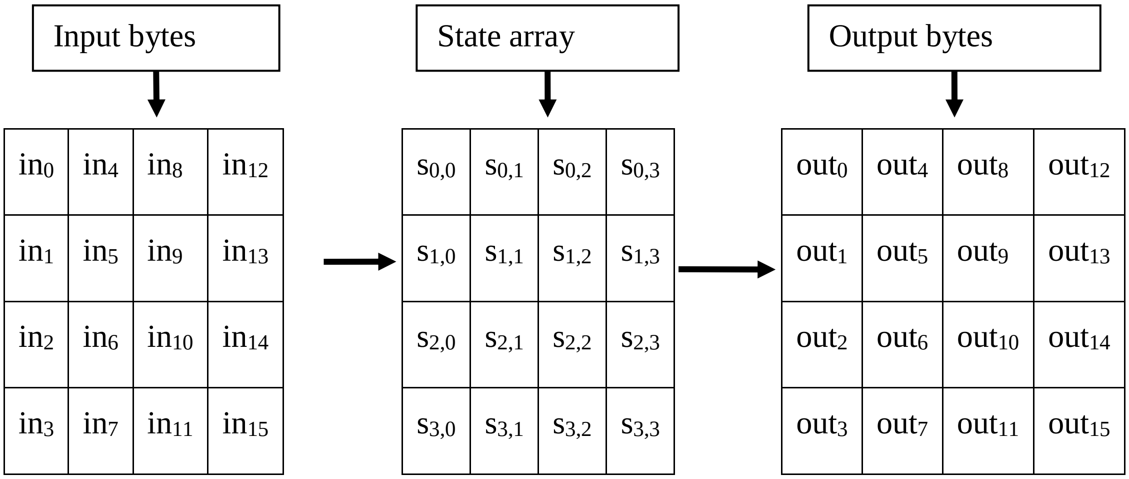
(2.1) illustrates the first step in AES block cipher is to duplicate the supplied byte array, *in*, to the state array *s*. (2.2) illustrates the final product.

$$out\,[r + 4c] = s\,[r, c] \quad \text{for} \quad 0 \le r < 4 \text{ and } 0 \le c < 4 \qquad (2.2)$$

Afterward, the state array is subjected to a series of changes, with the sequence $out_0$, $out_1$, ……., $out_{15}$ as demonstrated in (2.2). The correspondence between the indices of the input and output with the indices of the state array is demonstrated in Figure 2.1.

Figure 2.1

AES State Array Input Output (FIPS 197, p. 7)



Input bytes

|  |  |  |  |
|---|---|---|---|
| $in_0$ | $in_4$ | $in_8$ | $in_{12}$ |
| $in_1$ | $in_5$ | $in_9$ | $in_{13}$ |
| $in_2$ | $in_6$ | $in_{10}$ | $in_{14}$ |
| $in_3$ | $in_7$ | $in_{11}$ | $in_{15}$ |

State array

|  |  |  |  |
|---|---|---|---|
| $s_{0,0}$ | $s_{0,1}$ | $s_{0,2}$ | $s_{0,3}$ |
| $s_{1,0}$ | $s_{1,1}$ | $s_{1,2}$ | $s_{1,3}$ |
| $s_{2,0}$ | $s_{2,1}$ | $s_{2,2}$ | $s_{2,3}$ |
| $s_{3,0}$ | $s_{3,1}$ | $s_{3,2}$ | $s_{3,3}$ |

Output bytes

|  |  |  |  |
|---|---|---|---|
| $out_0$ | $out_4$ | $out_8$ | $out_{12}$ |
| $out_1$ | $out_5$ | $out_9$ | $out_{13}$ |
| $out_2$ | $out_6$ | $out_{10}$ | $out_{14}$ |
| $out_3$ | $out_7$ | $out_{11}$ | $out_{15}$ |

The relationship between the indices of the state array and the input and output indices is shown in Figure 2.1. A block is made up of four words, and a word is made up of four bytes. (2.3) shows how the four columns of the state array s are read as an array v of four words.

$$v_0 = \begin{pmatrix} s_{0,0} \\ s_{1,0} \\ s_{2,0} \\ s_{3,0} \end{pmatrix}, v_1 = \begin{pmatrix} s_{0,1} \\ s_{1,1} \\ s_{2,1} \\ s_{3,1} \end{pmatrix}, v_2 = \begin{pmatrix} s_{0,2} \\ s_{1,2} \\ s_{2,2} \\ s_{3,2} \end{pmatrix}, v_3 = \begin{pmatrix} s_{0,3} \\ s_{1,3} \\ s_{2,3} \\ s_{3,3} \end{pmatrix} \tag{2.3}$$

Thus, the column index *c* of *s* becomes the index for *v*, and the row index *r* of *s* becomes the index for the four bytes in each word. Given a one-dimensional array *u* of words, *u*[*i*] denotes the word that is indexed by *i*, and the sequence of four words *u* [*i*], u [*i* + 1], *u* [*i* + 2], u [*i* + 3] is signified by *u* [*i*…*i* +3]

The central component Cipher () and Inverse Cipher function InvCipher () algorithms is a *Round*, which is a series of fixed state modifications. *Round key*, a block that is typically expressed in series of 4 words, or 16 bytes, is an additional input needed for every *Round* (NIST, 2022).

The block cipher key is fed through an expansion algorithm, designated using key expansion function KeyExpansion(), which outputs the round keys. The input of KeyExpansion() is specifically a word array signaled *key*, and output is an enlarged word array signaled *w*, known as *key schedule* (NIST, 2022).

AES can be explained using the following parameters: *Nr* signifies number of rounds, *Nk* signifies word count in the key, and *Nb* signifies word count in the state and in this case Nb=4. Additionally, there are no additional Rijndael configurations that adhere to this Standard but only specific values of *Nk*, *Nb,* and *Nr* are provided in Table 2.1 (NIST, 2022).

Table 2.1

AES Key-Block-Round Combinations (FIPS 197, p. 11)

| | Key | length | Block size | | Number of rounds |
|---|---|---|---|---|---|
| | *Nk* | in bits | *Nb* | ( in bits) | *Nr* |
| AES-128 | 4 | 128 | 4 | 128 | 10 |
| AES-192 | 6 | 192 | 4 | 128 | 12 |
| AES-256 | 8 | 256 | 4 | 128 | 14 |

Table 2.1 illustrates AES ciphers in which *Nk* denotes word count, *Nb* shows the word count in the state and *Nr* denotes rounds count. In AES standard, *Nb* is 4 and size are 128 bits.

The three sources of Cipher () are the data, which is a block encoded as a linear array of 16 bytes, the instance's round count Nr, and the round keys as illustrated (2.4) (NIST, 2022).

AES-128 (in, key) = Cipher (in, 10, KeyExpansion (key))
AES-128 (in, key) = Cipher (in, 12, KeyExpansion (key)) (2.4)
AES-128 (in, key) = Cipher (in, 14, KeyExpansion (key))

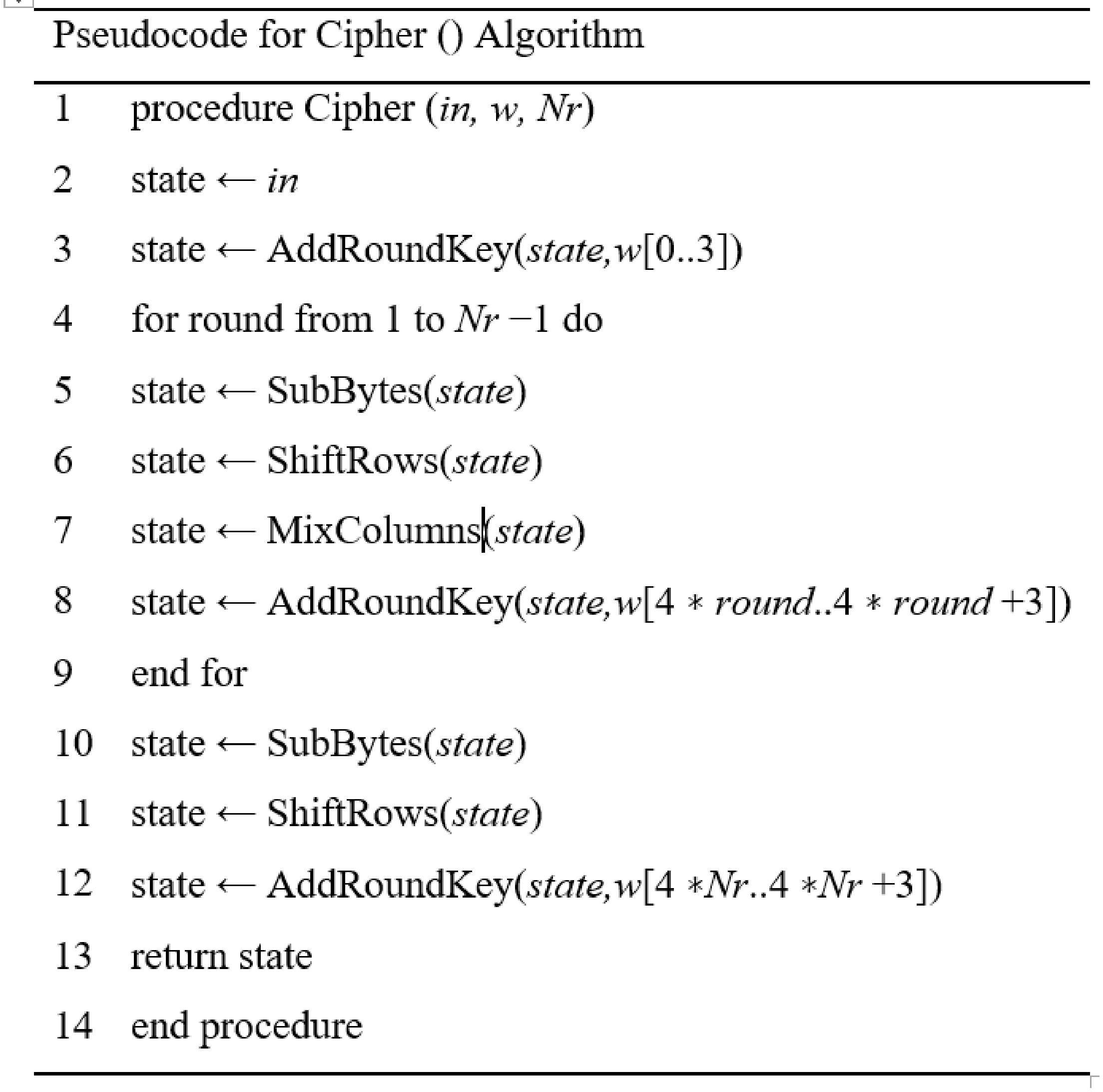
(2.4) illustrates encryption process used to generate ciphertext. Line one indicates that the AES-128 encryption process involves taking the plaintext and the encryption key, then expanding the key into multiple round keys, and finally performing 10 rounds of encryption to produce ciphertext. Line two indicates that the AES-128 encryption process involves taking the plaintext and the encryption key, expanding the key into multiple round keys, then performing 12 rounds of encryption to produce the ciphertext. Line three indicates that the AES-128 encryption process involves taking the plaintext and the encryption key, expanding the key into multiple round keys, then performing 14 rounds of encryption to produce the ciphertext (NIST, 2022).

The commands SubBytes() and ShiftRows() apply substitution tables (S-boxes) to each byte. The substitution table is a crucial component used during the SubBytes step of the encryption process. It is designed in order to provide non-linearity in the cipher, which helps to protect against cryptographic attacks (NIST, 2022).

After each byte is translated into two hexadecimal numbers, every byte is put back with another using a table in the non-linear substitution stage known as SubBytes while ShiftRows is a transposition step that rotates each row in a circular motion by 0, 1, 2, and 3 positions for the appropriate rows. MixColumns() and AddRoundKey() combine round keys and the state array, respectively. A mixing procedure called MixColumns works with the state's column data. Every column in the state is changed into a new column and finally one column is introduced using AddRoundKey at a time using bitwise XOR action, Add Round Key appends a string of characters to each state column (Sakr, Omara, & Nomir, 2013). The pseudocode in Table 2.2 contains information about the Cipher() (NIST, 2022).

Table 2.2

Pseudocode for AES Cipher (FIPS 197, p. 12)



Pseudocode for Cipher () Algorithm

```
1   procedure Cipher (in, w, Nr)
2       state ← in
3       state ← AddRoundKey(state,w[0..3])
4       for round from 1 to Nr −1 do
5           state ← SubBytes(state)
6           state ← ShiftRows(state)
7           state ← MixColumns(state)
8           state ← AddRoundKey(state,w[4 ∗ round..4 ∗ round +3])
9       end for
10      state ← SubBytes(state)
11      state ← ShiftRows(state)
12      state ← AddRoundKey(state,w[4 ∗Nr..4 ∗Nr +3])
13      return state
14  end procedure
```

Table 2.2 illustrates the pseudocode for AES cipher algorithm. The procedure Cipher (*in*, *w*, *Nr*) refers to the core encryption process with *in* representing the plaintext that needs to be encrypted, *w* is the expanded key schedule derived from the original encryption key, and *Nr* stands for the number of rounds. The AddRoundKey in line 3 integrates the encryption key into the plaintext, ensuring subsequent transformations (SubBytes, ShiftRows, MixColumns) are dependent on the key (NIST, 2022).

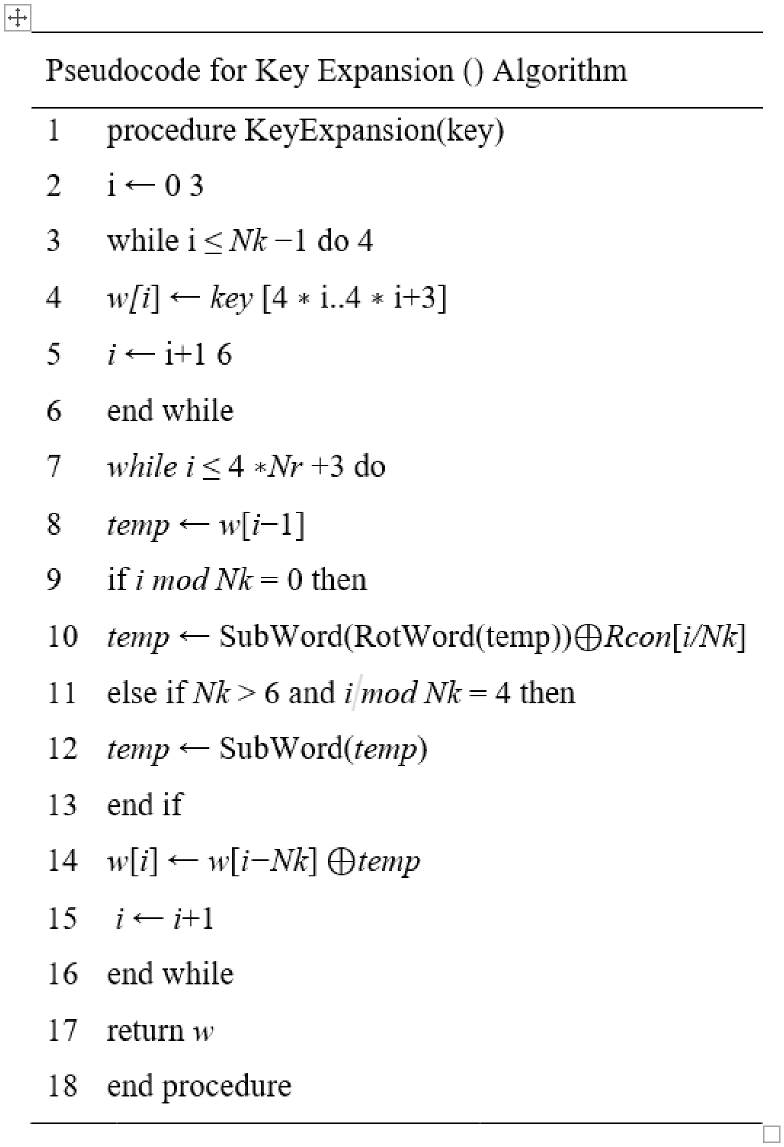
Lines 4 to 12 apply the round function $Nr$ times, changing the state array following an initial round key addition in line 3. Lines 10 to 12's last round differs by skipping the MixColumns() transformation. The output in line 13 then contains the final state, which is returned in accordance with the state conventions.

The state is transformed using ShiftRows(), which involves cyclically shifting the bytes in the state's final three rows. Each of the four columns in the state known as MixColumns() is multiplied by a single fixed matrix during the transformation. Through a bitwise XOR operation, a round key and the state are connected in the transformation of the state known as AddRoundKey(). Each round key, in particular, is composed of four essential schedule words (NIST, 2022).

To generate $4 \times (Nr + 1)$ words from a key, the KeyExpansion() routine is used. For instance, four words are produced for each of the Nr +1 AddRoundKey() applications found in the Cipher specification(). The routine's output is a list of words in order designated $w[i]$, where $i$ falls in the vicinity of $0 \leq i < 4 \times (Nr + 1)$. Then fixed phrases are invoked through KeyExpansion(), which is indicated by $Rcon[j]$ for$1 \leq j \leq 10$. The round constants are ten words. Each of ten round keys for AES-128 is generated by calling a unique round constant. The first six and eight of these constants are referred to by the key expansion algorithm for AES-192 and AES-256, respectively. Table 2.3 demonstrates pseudocode for KeyExpansion () (NIST, 2022).

Table 2.3

Pseudocode for AES Key Expansion (FIPS 197, p. 18)



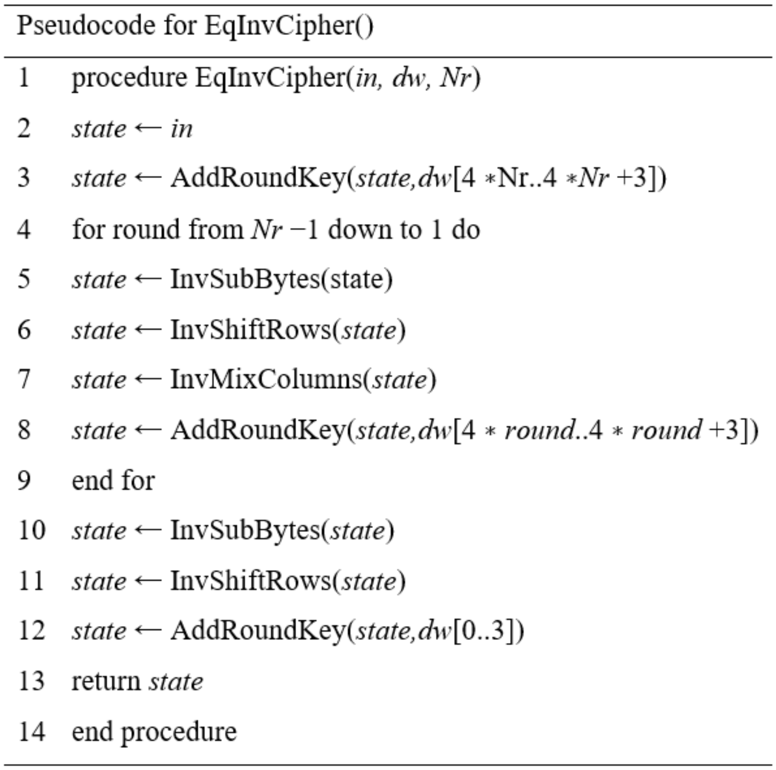
Pseudocode for Key Expansion () Algorithm

```
procedure KeyExpansion(key)
    i ← 0 3
    while i ≤ $Nk - 1$ do 4
        $w[i]$ ← $key$ [4 ∗ i..4 ∗ i+3]
        $i$ ← i+1 6
    end while
    $while$ $i \leq 4 * Nr + 3$ do
        $temp$ ← $w[i-1]$
        if $i$ $mod$ $Nk = 0$ then
            $temp$ ← SubWord(RotWord(temp))⨁$Rcon[i/Nk]$
        else if $Nk > 6$ and $i$ $mod$ $Nk = 4$ then
            $temp$ ← SubWord($temp$)
        end if
        $w[i]$ ← $w[i-Nk]$ ⨁$temp$
        $i$ ← $i$+1
    end while
    return $w$
end procedure
```

Table 2.3 illustrates the pseudocode for key expansion algorithm. The whole process generates a series of round keys from the initial encryption key. The key expansion ensures that each round key of AES encryption uses a unique round key and thus enhancing the security of the encryption process. Table 2.4 illustrates the pseudocode for equivalent inverse cipher.

Table 2.4

Pseudocode for AES EqInCipher (FIPS 197, pp. 24)



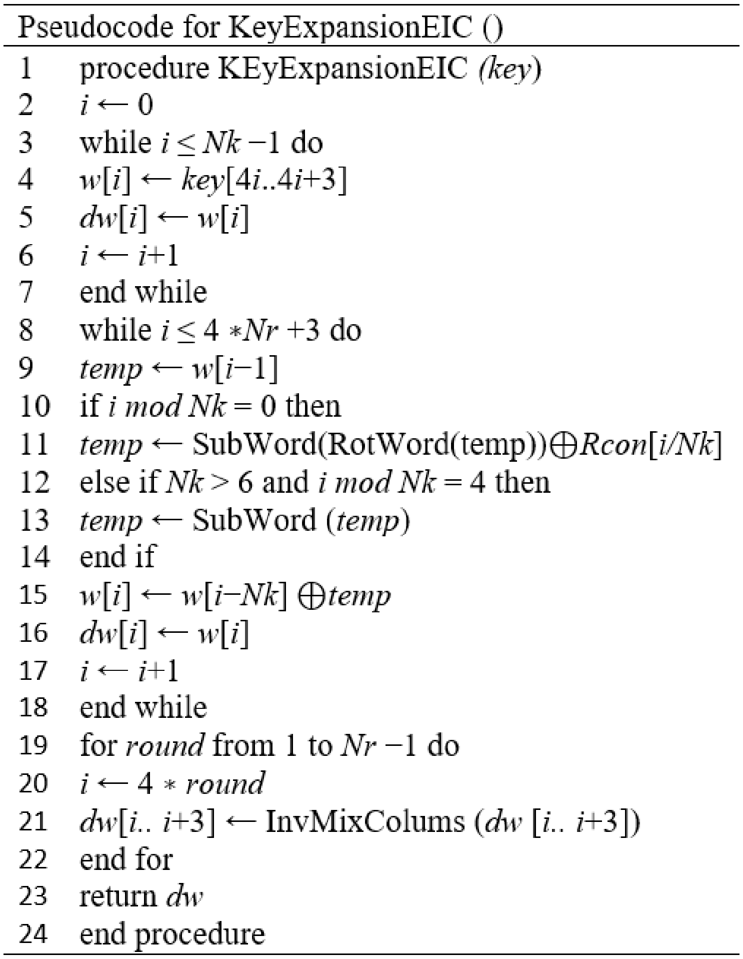
Pseudocode for EqInvCipher()

```
procedure EqInvCipher(in, dw, Nr)
    state ← in
    state ← AddRoundKey(state,dw[4 *Nr..4 *Nr +3])
    for round from Nr −1 down to 1 do
        state ← InvSubBytes(state)
        state ← InvShiftRows(state)
        state ← InvMixColumns(state)
        state ← AddRoundKey(state,dw[4 * round..4 * round +3])
    end for
    state ← InvSubBytes(state)
    state ← InvShiftRows(state)
    state ← AddRoundKey(state,dw[0..3])
    return state
end procedure
```

Table 2.4 illustrates the pseudocode for equivalent inverse cipher. The pseudocode outlines the procedure for decrypting data using AES algorithm. It starts with initializing the state with the input ciphertext, then followed by the initial round in which XOR with the state with the last round key from the expanded key schedule. Next follows main rounds and finally a final round. Table 2.5 contains a pseudocode for AES key expansion

Table 2.5

Pseudocode for AES KeyExpansion (FIPS 197, p. 25)



```
Pseudocode for KeyExpansionEIC ()
1   procedure KEyExpansionEIC $(key)$
2     $i \leftarrow 0$
3     while $i \le Nk - 1$ do
4       $w[i] \leftarrow key[4i..4i+3]$
5       $dw[i] \leftarrow w[i]$
6       $i \leftarrow i+1$
7     end while
8     while $i \le 4 * Nr + 3$ do
9       $temp \leftarrow w[i-1]$
10      if $i \bmod Nk = 0$ then
11        $temp \leftarrow$ SubWord(RotWord(temp))$\oplus Rcon[i/Nk]$
12      else if $Nk > 6$ and $i \bmod Nk = 4$ then
13        $temp \leftarrow$ SubWord $(temp)$
14      end if
15      $w[i] \leftarrow w[i-Nk] \oplus temp$
16      $dw[i] \leftarrow w[i]$
17      $i \leftarrow i+1$
18    end while
19    for $round$ from 1 to $Nr - 1$ do
20      $i \leftarrow 4 * round$
21      $dw[i..\ i+3] \leftarrow$ InvMixColums $(dw\ [i..\ i+3])$
22    end for
23    return $dw$
24  end procedure
```

The first and last round keys are identical in *dw* and *w*. Lines 18 to 21 show how the remaining round keys have been modified. The input to InvMixColumns () is described in line 21's comment; a single-dimensional word list is transformed into a byte-only two-dimensional array as seen in Figure 2.2 (NIST, 2022).

AES algorithm implementation must support at least one of the three key lengths such as 128, 192, or 256 bits (for instance, *Nk* is 4, 6 or 8 respectively). Two or three key lengths may be supported by implementations on an optional basis, which could improve the compatibility of algorithm implementations (NIST, 2022).

Internationally, AES is utilized in high security systems, banking systems, and government systems to protect online and mobile banking (Khelifi, 2013). This is due to the fact that it might take longer than the universe's age to crack a 128-bit AES key. When security is a concern, banks, governments, and wireless communications all use the encryption standard AES because there have been no specific attacks against it so far (JScape, 2022).

Current cryptanalysis assaults on AES algorithm are based on work that is progressing against AES. Amrita, Gupta, and Mishra (2018) claim that AES is currently vulnerable to a wide range of side channel attacks. These attacks make use of the descriptive information obtained from the implementation of the protocols and cryptographic primitives. This characteristic information can be obtained by considering aspects such as timing, power consumption, or electromagnetic radiation. Errors in computation, changes in temperature or frequency, and defects in hardware or software can all result in different kinds of information. Side channel attacks take use of the characteristics of the software and hardware components as well as the implementation structure of the cryptographic primitive (Jani, 2015).

Additional cryptanalysis attacks against AES include collision attacks, eXtended Linearization (XL), eXtended Sparse Linearization (XSL), algebraic attacks, and cube attacks (Anwar, Hasan, Hasan, Loren, & Hossain, 2015). These techniques are developing gradually, but no significant advancements have been made as of yet. Due to these advancements, AES will not last as long as the traditional algorithm suite approved for classified applications. Nevertheless, there are workable solutions that can eliminate these flaws at the equipment level (Amrita, Gupta, & Mishra, 2018).

#### 2.7.1.2 Blowfish Algorithm

A popular symmetric cipher algorithm for data encryption and security is called Blowfish. Because it can accommodate key lengths ranging from 32 to 448 bits, it is a useful tool for data security. Blowfish is an effective tool for data security because of its variable key length, it is known for fast encryption and decryption speeds and therefore making it suitable for applications that require quick data processing. Additionally, the algorithm is adaptable and has the ability to work with variable block sizes and enables efficient data transfers with minimal resource utilization (Encryption Consulting, 2024). Despite having a known vulnerability related to weak keys, it has not been subject to any successful attacks (Schneier, 1994; Schneier, 1996).

Blowfish is a 64-bit symmetric block cipher. Data encryption and key expansion are the two phases in which the technique works. Several subkey arrays totaling 4168 bytes are created from a key of up to 448 bits during the key expansion stage. Feistel network with 16 rounds is used in the data encryption phase (Schneier, 1994; Schneier, 1996).

Blowfish algorithm is suitable for applications such as automatic file encryption systems or communication links where the key is constant over time. This is because the algorithm is efficient with robust security from its variable key length. Additionally, the algorithm is simple with low resource utilization (Encryption Consulting, 2024). Table 2.6 demonstrates the basic algorithm for Blowfish.

Table 2.6

Basic Algorithm for Blowfish (Schneier, 1994)

Basic Algorithm for Blowfish

```
1   Divide X into two 32-bit halves XL and XR
2   For i=1 to 16:
    XL = XL Pi
    XR = F (XL) XR
    Swap XL and XR
    End for
3   Swap XL and XR
4   XR = XR P17
5   XL = XL P18
6   Recombine XL and XR
7   Output X (64-bit data block: cipher text)
```

Table 2.6 provides a pseudocode which outlines the encryption process of Blowfish algorithm. The 64-bit input data block X is split into two 32-bit halves: XL for left half and XR for the right half. The process is followed by initial key mixing as shown in step two. After initial key mixing, a final swap is initiated. This is followed by recombining the data which are the two 32-bit halves XL and XR bank into a single 64-bit block. Finally, the output is a 64-bit ciphertext block.

### 2.7.1.3 Twofish Algorithm

Twofish algorithm is an improvement of the prior model of Blowfish algorithm. Twofish employs identical keys for data encipherment and decipherment. Precomputed key-dependent S-boxes are utilized. Any block cipher algorithm must include an S-box, commonly referred to as a substitution box. Twofish comprises key length of 128 to 256 bits. Any encryption method that uses 128 bits or more for encryption is theoretically shown to be secure from brute force assault. Products like 96Crypt by eRightSoft, KeePass, GnuPG, PGP, among others, utilize Twofish (Mandal & Singh, 2021).

Performance has been a priority throughout the development of Twofish. It performs well on a range of hardware but more specifically support a variety of performance tradeoffs at different levels, depending on rate, key structure, memory utilization, hardware gate count and other implementation characteristics. The outcome is a very adaptable algorithm that can be successfully used in a number of cryptographic applications (Mandal & Singh, 2021).

Applications running on miniature devices and are integrated in hardware supports effectiveness of Twofish algorithm. This enables the implementer to balance performance by adjusting the encryption's speed, timing and size. The cypher is a 16-round Feistel network with a bijective F function made up of four key-dependent eight-by-eight-bit S-boxes, bitwise transformations, a fixed four-by-four maximum distance separable matrix over GF (28), and a carefully thought-out key schedule (Nagaraj, 2023). The Twofish algorithm is demonstrated in Table 2.6.

Table 2.7

Twofish Algorithm Procedure (Nagaraj, 2023)

| | Twofish Algorithm Procedure |
|---|---|
| 1 | Plaintext divided into four 32-bits. |
| 2 | This undergoes an XOR operation with four keywords in the whitening input stage. |
| 3 | A sixteen-round procedure is used. The function g receives the two words on the left as input in each round. |
| 4 | Subsequent to the linear mixing step based on the MDS matrix, the g function is composed of S-boxes that rely on the given key bytes. |
| 5 | Every S-box generates an 8-bit output and needs an 8-bit input. |
| 6 | In order to multiply the four results by 4X4 MDS (maximum distance that can be replaced), they are transmitted as vectors of length 4. multilayer |
| 7 | Using the Pseudo-Hadamard Transform (IPM), the outputs of the two functions g are merged, and two keywords are appended. |
| 8 | The initial 1 bit is played after that, opening one of the phrases XOR on the right, which is comprised of these two results. |
| 9 | Next, during the following round, the left and right are swapped. |
| 10 | The last round switch is reversed after all rounds, and empathy is XORed with four additional keywords to create the encrypted text. |

Table 2.6 demonstrates how Twofish encryption algorithm works. The algorithm starts with dividing plaintext into four 32-bit words which makes up a total of 128 bits. In the whitening stage, each of the 32-bit words undergo an XOR operation with four key words. This stage is followed by a sixteen-round procedure in which the function g processes the two leftmost words. The function g involves a linear mixing step based on the maximum distance separable matrix and S-boxes. Each of the S-boxes takes an 8-bit input and produces an 8-bit output. The output from the S-boxes is treated as vectors of length 4 and multiplied by a four-by-four matrix. The output of the two g functions is then combined using Pseudo-Hadamard Transform (PHT), and two more key words are added. The result of the PHT is XORed with the rightmost two words from the previous step, and the right halves are swapped for the next round. Steps 3 to 8 are repeated for a

total of 16 rounds, with the left and right halves being swapped at the end of each round. After all rounds are completed, the final output undergoes another XOR operation with four additional key words, producing the encrypted ciphertext.

### 2.7.2 Asymmetric Encryption Algorithms

Asymmetric encryption utilizes both a public key (known to sender and recipient) and a private key (known to a recipient). The public key can be used by anybody who requires to send or receive a message while information is decoded using the private key and thus increasing security (Devi, 2015). Asymmetric encryption performs better in terms of security than symmetric encryption because of high computer resources required in security mechanisms for enforcing digital certificates, hashing, digital signatures and encryption.

#### 2.7.2.1 Rivest-Shamir-Adelman Algorithm

Two distinct keys are used to encrypt and decode data in an asymmetric cryptosystem (Bhanot & Hans, 2015). In networks, Rivest-Shamir-Adleman (RSA) is frequently used to protect data. The two primary unresolved RSA puzzles where the outcome of two prime numbers is N are the integer factorization problem and the RSA problem, such as finding the Nth root.

Number theory says that the product for two prime integers is easy to calculate, but factorization is difficult. An essential component of RSA security is the enormous numbers principle. The RSA algorithm's key size range, which is between 2048 and

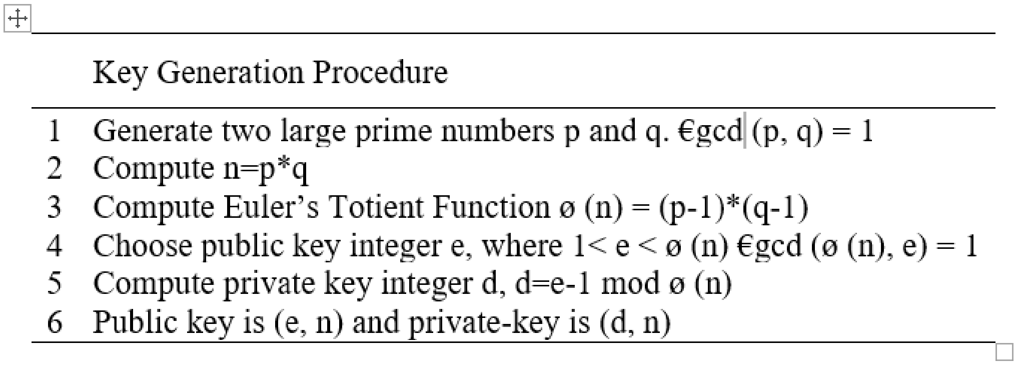
4096, makes factorization challenging. The RSA method's decryption is based on the variables d and N, where d is the decryption key.

RSA has a time complexity of $O(n^2)$ (Bhanot & Hans, 2015). This is due to the fact that both encryption and decryption involve modular exponentiation, which is computationally intensive. Additionally, the complexity is directly related to the bit length of keys. As such, when the key size increases, the number of operations required for encryption and decryption grows quadratically.

This cryptosystem incorporates: key generation, encryption and decryption (Saini & Vandana, 2022). Table 2.7 illustrates RSA key generation procedure, while Table 2.8 details RSA encryption and decryption process.

Table 2.8

Rivest-Shamir-Adelman Algorithm Key Generation (Tahir, 2015)



| | Key Generation Procedure |
|---|---|
| 1 | Generate two large prime numbers p and q. €gcd (p, q) = 1 |
| 2 | Compute n=p*q |
| 3 | Compute Euler’s Totient Function ø (n) = (p-1)*(q-1) |
| 4 | Choose public key integer e, where 1< e < ø (n) €gcd (ø (n), e) = 1 |
| 5 | Compute private key integer d, d=e-1 mod ø (n) |
| 6 | Public key is (e, n) and private-key is (d, n) |

Table 2.8 demonstrates RSA algorithm key generation. Step one involves choosing two large prime numbers p and q (Meng & Zeng, 2015). This is followed by computing n as a product of p and q. The next step involves computing Euler’s totient function. The next step involves choosing a public key integer e followed by computing a private key integer

d. The public key is the pair (e, n )which is distributed and used for encrypting messages and the private key is the pair (d, n) which is kept secret and used decrypting messages. Table 2.9 demonstrates the RSA encryption and decryption process.

Table 2.9

RSA Algorithm Encryption Decryption process (Lin, Sun, & Qu, 2018, p. 4)

| Encryption Process | Decryption Process |
|---|---|
| *C* = Pe mod *n* | *P* = *Cd* mod *n* |

The following parameter denotations are used in Table 2.9 to illustrate the encryption process algorithm: e is the public key, n is the product of two prime numbers, C is the ciphertext, P is the plaintext, and so on. The decoding technique uses the same settings. Both sender and recipient need to understand the value of n. The sender is aware of the value of e, whereas only the recipient knows the value of d. Accordingly, the public key (e, n) can be considered the key of this algorithm, whereas the private key (d, n) can be considered the key of this algorithm (Anada, Yasuda, Kawamoto, Weng, & Sakurai, 2019; Sharma, Agrawal, Pandey, Khan, & Dinkar, 2019).

The primary security services provided by RSA are confidentiality, secrecy, authentication, integrity, and non-repudiation (Sharma & Bohra, 2017). The algorithm's strong security public-key cryptosystem makes it difficult for hackers to decipher (Gaur, Mehra, & Kumar, 2018).

According to Sharma and Bohra (2017), RSA's main security services include confidentiality, secrecy, authentication, integrity, and non-repudiation. According to

Gaur, Mehra, and Kumar (2018), the algorithm's robust public-key cryptosystem makes it challenging for hackers to understand.

### 2.7.2.2 Elliptic Curve Cryptography

Elliptic Curve Cryptography (ECC) is a fundamental component of modern cryptographic systems. It is built upon the sophisticated mathematics of elliptic curves, which serve as a rich framework for cryptographic processes (Singh, 2014). The security of ECC relies on the difficulty of solving the Elliptic Curve Discreate Logarithm Problem (ECDLP). Unlike traditional discrete logarithm problems in finite fields, the ECDLP is considered harder to solve, especially for large key sizes. This difficulty allows ECC to achieve comparable security with smaller key sizes, making it efficient in terms of computational resources and bandwidth (Singh, Nayyar, & Garg, 2023). ECC demonstrates built-in robustness against quantum algorithms like Shor's algorithm (Singh, 2016).

ECC is widely utilized in several cryptographic areas, including key exchange, encryption, and digital signatures. By creating shared secret keys, protocols such as Elliptic Curve Diffie-Hellman (ECDH) allow parties to exchange keys securely over insecure channels.

### 2.7.2.3 Digital Signatures

Digital Signature Algorithm (DSA) is a federal information processing standard that was standardized in 1994 (NIST, 2022). DSA functions with discrete logarithm problems and modular exponentiation which are difficult to calculate using brute-force attacks. A combination of the hardness of the discrete logarithm problem and the computational intensity of modular exponentiation ensures that DSA is secure against brute-force attacks (Simplilearn, 2022).

A hashing algorithm, which determines the result of another hash or digest based on even minor changes to the data, makes this technique conceivable. A modified piece of data creates a new hash, informing the recipient that the data they recently received was not transmitted by the original sender and was changed or corrupted during transit (Thapar & Sarangal, 2018). Digital signatures further reinforce security during data transmissions. To achieve non-repudiation, integrity, and authentication, digital signatures are used (Mishra, 2017). The DSA procedure consists of three steps: creating a key, creating a signature, and verifying a signature (Simplilearn, 2022). A complete DSA design technique is shown in Figure 2.2.

Figure 2.2

Digital Signature Algorithm Design (Simplilearn, 2022)

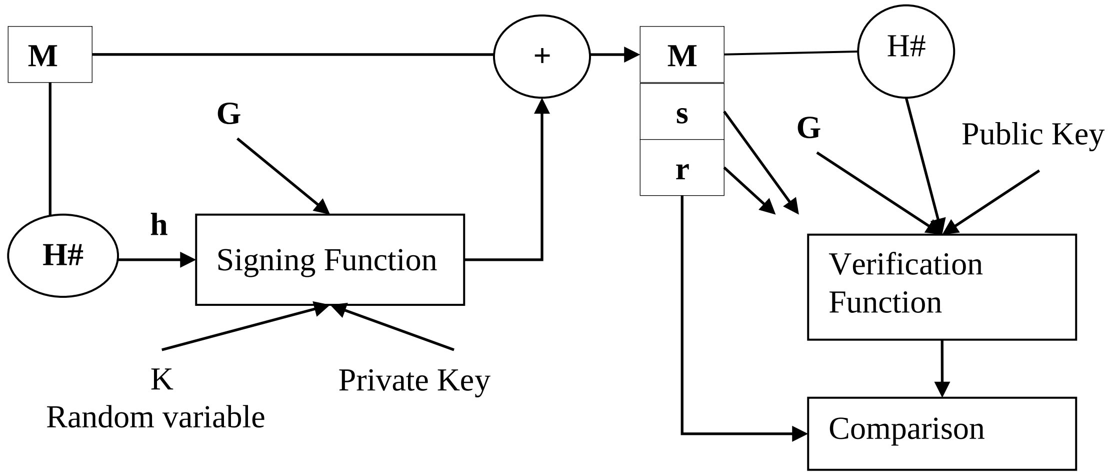
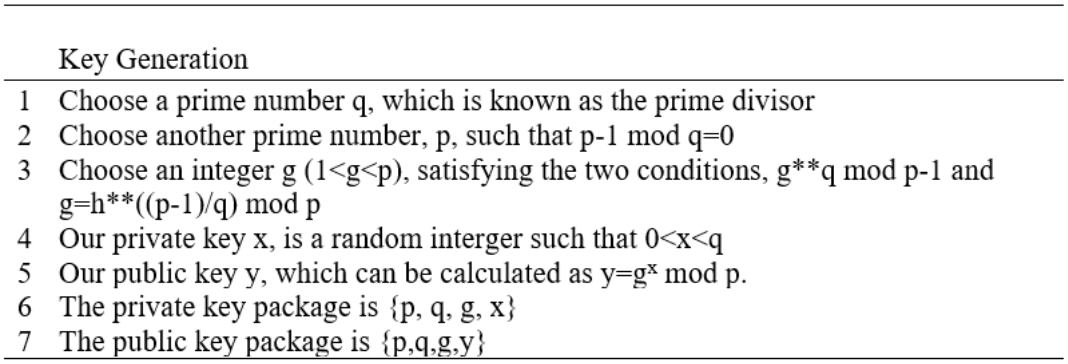


Figure 2.2 demonstrates DSA workflow where M is the message, H is the hash function, h is the hash digest, E is the encryption, + is the bundle both plaintext and digest. To produce a digest, the original message M is first run through a hash function. The message is then encrypted using the sender's private key after being combined with the hash digest h. The recipient receives the encrypted bundle and uses the sender's public key to decrypt it. The same hash function is used to produce a comparable digest after the message has been decoded. The freshly created hash is then contrasted with the hash value that was sent with the message. If they match, then data integrity is verified. Figure 2.2 demonstrates DSA algorithm design while Table 2.10 illustrates DSA algorithm key generation procedure.

Table 2.10

DSA Algorithm Key Generation Procedure (Simplilearn, 2022)



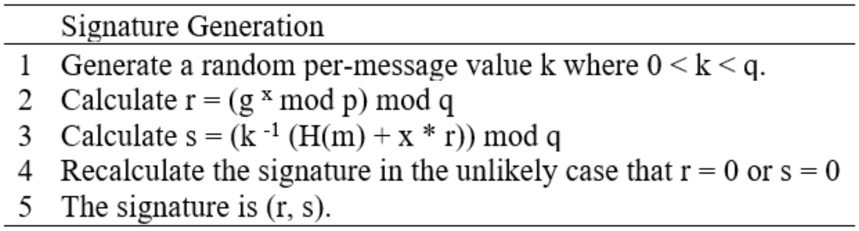
|   | Key Generation |
|---|---|
| 1 | Choose a prime number q, which is known as the prime divisor |
| 2 | Choose another prime number, p, such that p-1 mod q=0 |
| 3 | Choose an integer g (1<g<p), satisfying the two conditions, g**q mod p-1 and g=h**((p-1)/q) mod p |
| 4 | Our private key x, is a random interger such that 0<x<q |
| 5 | Our public key y, which can be calculated as $y=g^x$ mod p. |
| 6 | The private key package is {p, q, g, x} |
| 7 | The public key package is {p,q,g,y} |

Table 2.10 illustrates the DSA key generation procedure which begins by choosing a prime number q which is known as the prime divisor. The next step involves choosing another prime number p such that p-1 is divisible by q. The third step involves selecting an integer g such that 1< g < p. The fourth step involves choosing a private key x which is a random integer such that 0 < x < q. This is followed by calculating the public key y

which is calculated as $y=g^x \bmod p$. The private key package consists of the values p, q, g, x while the public key consists of the values p, q, g, y. Table 2.11 illustrates the DSA generation process.

Table 2.11

Digital Signature Algorithm Generation Process (Simplilearn, 2022).

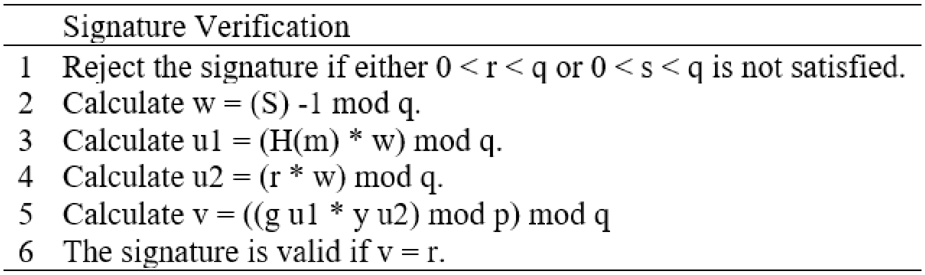


| | Signature Generation |
|---|---|
| 1 | Generate a random per-message value k where $0 < k < q$. |
| 2 | Calculate $r = (g^{x} \bmod p) \bmod q$ |
| 3 | Calculate $s = (k^{-1} (H(m) + x * r)) \bmod q$ |
| 4 | Recalculate the signature in the unlikely case that $r = 0$ or $s = 0$ |
| 5 | The signature is (r, s). |

Table 2.11 demonstrates the DSA algorithm generation process. It involves generating a random per-message value k such that $0 < k < q$. This value must be kept secret and should be different for each message to ensure security. In line 2, calculate r using the formula $r = (g^{x} \bmod p) \bmod q$. Here, g is the base from the public key k which is the random value generated in step 1 and p and q are the prime numbers chosen during key generation. In line 3, s is calculated using the formula $s = (k^{-1} (H(m) + x * r)) \bmod q$. the value $k^{-1}$ is the modular inverse of k modulo q while (H(m)) is the hash of the message m and x is the private key while r is the value calculated in step 2. In line 4 the signature can be recalculated, if necessary, in case either r is 0 or s is 0 then the process is repeated from step 1to generate a new k and recalculate r and s. Finally, the signature for the message is the pair r, s. Table 2.12 illustrates DSA algorithm verification process.

Table 2.12

Digital Signature Algorithm Verification Process (Simplilearn, 2022)



|   | Signature Verification |
|---|---|
| 1 | Reject the signature if either 0 < r < q or 0 < s < q is not satisfied. |
| 2 | Calculate w = (S) -1 mod q. |
| 3 | Calculate u1 = (H(m) * w) mod q. |
| 4 | Calculate u2 = (r * w) mod q. |
| 5 | Calculate v = ((g u1 * y u2) mod p) mod q |
| 6 | The signature is valid if v = r. |

Table 2.12 illustrates DSA verification process in which from line 1 a signature can be rejected as invalid if either $0 < r < q$ or $0 < s < q$ is not satisfied. This involves checking the values for r and s from the signature. In line 2, computation of modular inverse of s modulo q is done to find the value of w which is used in subsequent calculations. In line 3, the value of u1 is computed by multiplying the hash of the message (H(m)) with the value of w and then take the result modulo q. in line 4, the value of u2 is computed by multiplying r from the signature with w and take the result modulo q. In line 5, the value of v is computed using the formula $v = ((g^{u1} * y^{u2}) \bmod p) \bmod q$. Here g is the base from the public key, y is the public key, u1 and u2 are the values calculated in line 3 and 4 and p and q are the prime numbers chosen during key generation. In line 6, the signature is valid if $v = r$.

DSA technology is utilized in smart cards, Integrated Services Digital Network (ISDN), web applications and email verification among other systems. DSA commonly provides message authentication, message integrity, non-repudiation, and secrecy as communication security services (Ramya, 2013).

Because the DSA technique generates keys faster and is more stable and secure than the RSA method, it is advantageous because it requires less storage space over its entire lifecycle., because it utilizes smaller key sizes, efficient signature representation and fixed-length components such as r and s (Simplilearn, 2022). A safe digital signature system is one that requires the solution of the discrete logarithm problem before the algorithm can be cracked (Stinson, 2006).

Forgeries, known-message attacks, chosen-message attacks, and key-only attacks are the most common DSA attacks. The public verification key is the only weapon available to the attacker in key-only attacks. A valid signature for several messages that the attacker is aware of but hasn't specifically selected is given to them in a known-message attack. According to Kumar (2016), in a chosen-message attack, the attacker first determines the signatures on any randomly picked messages. There are two types of forgery attacks: selective forgeries and potential forgeries. Existential forgery happens when someone else creates a message/signature pair m that the authorized signer did not create. A message/signature pair m is created by the adversary during selective forging, with m predetermined by the adversary (Kumar, 2016).

Additionally, a variety of information enciphered and concealment techniques have been developed in order to obtain security in data channeling (Sachin & Dhawan, 2021). Steganography and watermarking are two main methods for concealing information (Sachin & Dhawan, 2021; Kadhim, Premaratne, Vial, & Halloran, 2019). A message on a media surface is concealed using both the two domains. The two approaches are also closely related, although each has its own goals (Taher, Ahmad, Hameed & Mokri, 2022).

Protecting the confidentiality of the secret communication from adversaries is the primary objective of watermarking (Peng, Lao, & Li, 2021). Steganography, on the other hand, aims to conceal the secret message in a public medium without providing evidence that the message has been located (Sachin & Dhawan, 2021). The secret message is disordered using the information encryption method called as cryptography so that no enemy can read it or comprehend it. This strategy may draw attention to or even make opponents aware of hidden message, even though it is impossible to understand and makes no sense (Kadhim, Premaratne, Vial, & Halloran, 2019).

The Greeks coined the words steganos and graphy to get the origin of the word steganography (Mushenko, Zolkin, & Yatsumira, 2021). While graphy is viewed as writing, steganos means covered or veiled. Greek historian Herodotus provided a written description of steganography in the year 440 BC. They penned their confidential information on wood carvings and covered them with wax to keep them secret and out of sight. The Greeks also developed a technique for writing a message on a messenger's shaved head that would be sent once the hair grew back (Mushenko, Zolkin, & Yatsumira, 2021).

Similarly, in order to disguise their communications, the Germans developed the microdot technique during World War I by using cover materials that were not threatening (Ravi, Joshi & Nandal, 2021). The Germans utilized various tools during World War II, like indestructible paints and open-coded messages, to script messages and evade scrutiny (Alsaawy, Sen, Alkhodre, Bahbouh, Baghanim, & Alharbi, 2021). It is possible to use audio, text, Deoxyribonucleic Acid (DNA), video, network and image steganography.

### 2.7.3 Steganography Algorithms

Steganography is a clever technique utilized to conceal data by lodging it onto a cover media such that an opponent cannot identify or notice the media carrying the secret message or data (Kadhim, Premaratne, Vial, & Halloran, 2019). Both steganography and cryptography focus on hiding a secret message to keep it safe from intruders or hackers (Hussaina, Wahaba, Idris, Antony, Jung, Ki, 2018). The two strategies can be combined effectively to increase security for data transmission via wireless networks.

Important factors including imperceptibility, embedding ability, security, robustness, and computational complexity are used to evaluate steganography techniques (Kadhim, Premaratne, Vial, & Halloran, 2019; Hussaina, Wahaba, Idris, Antony, Jung, 2018). Imperceptibility in steganography refers to the property that the object used to camouflage confidential message cannot be discerned using human eyes (Kadhim, Premaratne, Vial, & Halloran, 2019; Hussain, Wahaba, Idris, Antony, Jung, 2018).

Embedding capacity refers to the maximum amount of secret information that can be hidden within a cover object (such as an image, audio file or video) without noticeably altering its appearance or quality. This concept is crucial because it balances the need to conceal data effectively while maintaining the cover object’s imperceptibility (Hussain, Wahaba, Idris, Antony, Jung, 2018).

Majority of steganography methods are susceptible to a variety of attacks since adversaries are always improving their systems to recognize buried secret messages in stego images. Steganalysis attacks come in various ways, including visual, conventional, and non-conventional (Al-Rikabi & Hazim, 2021).

One important feature that is assessed in the transform field is the ability of stego images (images on which a secret message is already encoded) to keep the hidden message even after undergoing various image tests, like cropping, blurring, and adding noise, among others (Sachin & Gupta, 2021; Pirandola *et al*., 2020). The ability of algorithms to operate efficiently and quickly following the embedding and extraction stages is known as computational complexity. As a result, because of how they operate, algorithms with minimal computational complexity are frequently used (AbdelWahab, Hussein, Hamed, Kelash, Khalaf, 2021).

Various steganographic algorithms have been implemented to guarantee the security of data. Though not all steganography systems require secret keys to function, the utilization of the Kerckhoff principle can improve steganographic system security. The idea suggests that in order to successfully attack a steganographic system, an attacker must possess the secret key, even if he is aware of the system's architecture and operation. Thus, while implementing steganographic systems, it is advisable to include the secret keys which can be public or private (Morkel, 2012).

Through the incorporation of information into another medium, steganographic algorithms guarantee confidentiality and information security. Only a steganographic key can unlock such concealed data (Al-Shaaby, 2017). On the other hand, the method and strategy employed to hide the data can potentially function as identification evidence. If the information is embedded incorrectly, the method of identification and authentication could turn into a shared secret (Morkel, 2012). The embedded data is unable to be submitted to an integrity check since it may have been altered inadvertently or intentionally, and the changes made to the retrieved data may not have been observed (Domain, 2018).

It is important to note that visual analysis and statistical analysis are two main steganalysis approaches that target steganographic techniques. Visual analysis seeks to identify hidden information through inspection using a naked eye or with the use of a computer. Small changes in the carrier objects are sought for by statistical analysis (Hussain, Wahab, Idris, Ho, & Jung, 2018). Additionally, email firewalls have the ability to erase confidential data while filtering photographs, which poses a threat to image steganography. Different steganography methods are divided into different message carriers as illustrated in the following section.

#### 2.7.3.1 Text Steganography

This form of steganography utilizes text files as cover media where data is camouflaged. Format-based strategies, random and statistical creation and semantic techniques are some of the several methods that can be used to immerse texts. Using the ASCII representation of characters, the LSB algorithm in text steganography modifies the characters' least significant bits to subtly encode secret messages inside the text (Harinath, Raja, Suneel, Muzafer, & Rajesh, 2024).

According to Savia, Gutub, and Ghanmdi (2019), a method called Kashida includes inserting extension characters. If the concealed bit is 1, the mechanism can unite Kashida between characters. If the confidential bit is 1, two white spaces are started between words before moving on to the next word. Nothing is added between words if confidential bit is set to 0. A significant flaw with text steganography is the usage of the two white spaces, which could raise suspicion.

#### 2.7.3.2 Image Steganography

Images are used as a means of concealing information. Sensitive information can be concealed in images using a number of techniques, such as adaptive territory, special territory and frequency territory. Manikandan and Masilamani (2018) employed an inverse data hiding technique to extract a message from a stego image using encryption theory. For instance, it is unnecessary to forward patient details into another portfolio because in their trial, they preserved patient data onto medical photographs of the same patient. Their suggested method's main goal was to achieve strong data embedding capacity while having a low bit error rate. To share data between a sender and a recipient, the authors used three keys. This technique demonstrated good embedding capability and required little processing time.

Using the LSB of each color channel (Red, Green, and Blue) in pixels, the LSB algorithm in picture steganography allows information to be hidden within an image without compromising its aesthetics (Harinath, Raja, Suneel, Muzafer, & Rajesh, 2024).

The use of the LSB technique to experiment with concealing hidden data in human faces was investigated by Marella, Straub, and Benard (2019). The mouth, thumb, eyes, nose and other visible facial characteristics were among the surface elements that held these covert messages. The writers searched through the many image surface features to find the most useful location that was available hid a message there. Simulations of the studies suggested the technique could store hidden message in the human face's features and that it was challenging to find hidden information inside the encrypted image.

### 2.7.3.3 Audio Steganography

Audio steganography use audio as cover medium to obfuscate private information. This kind of steganography can be decrypted using a variety of techniques, including spread spectrum, phase concealment, parity concealment and LSB bit concealment. It is used to conceal data in Waveform Audio File (WAV), Audio Units (AU), and Moving Pictures Layer 3 (MP3) sound files.

It is not recommended to hide sensitive information in the same LSB position, as adversaries can easily retrieve the concealed data (Biswajita, Pal, & Bandyopadhyay, 2016). Hiding three bits in specific locations achieves data embedding at non-continuous sample regions, which reduces the stego image's capacity. Binny and Koilakuntla (2014) created a steganography algorithm in which an audio sample was transformed into bits before concealing the text data using LSB technology. Before applying the LSB technique for concealing, the cryptic data characters must first be converted into the matching binary digits. The evaluation metric applied to the various audio inputs was the Signal-to-Noise Ratio (SNR).

### 2.7.3.4 Video Steganography

Digital videos are utilized as cover media to camouflage confidential data. This method is useful because it allows for the concealment of enormous amounts of data in video files, which are moving streams of sounds and images. Video steganography is therefore seen as a synergy that incorporates both audio and image steganography. The two most popular methods of video steganography are direct data concealment into compressed data stream and concealment of raw video followed by compression of the data (Kunhoth, Subramarican, Al-maadeed, & Bouridane, 2023).

A video steganography technique that makes use of an Enhanced Hidden Markov Model (EHMM) to speed up the rate of retrieving hidden data was created by (Mritha & Isa, 2015). The algorithm has two phases. The first phase is to identify conditional states in the chosen video frames, and its second phase is to compute transitions between conditional states in series. Furthermore, the LSB approach in video steganography works on individual frames, enabling data to be hidden throughout a number of video frames (Harinath, Raja, Suneel, Muzafer, & Rajesh, 2024).

#### 2.7.3.5 Network Steganography

Network steganography entails concealing data within the header or payload fields of network protocols. This technique establishes covert channels for communication between a covert sender and receiver. These covert channels can be categorized into storage-based, timing-based, and hybrid-based channels (Jozef, Mazurczyk, & Szczpiorski, 2014).

For the secret communication space, Huang and Tang (2016) presented a new network voice-based steganography model. To solve the issue of packet loss, this method utilized a geographic design and a quick-start retransmission methodology. Two participants in communication can swap the obviously used vector across a secure channel by utilizing time and space negotiation framework.

Only a portion of the media packets in the sender's media stream are used by the method to mask data. In this case, the receiver must be able to distinguish both the covert vector being used to disguise the secret data and the streaming media containing it (Yang & Peng, 2017). Once the covert vector employed in the media data stream has been identified, the receiver can promptly recover the hidden message. Without compromising channel concealment, this method increased payload complexity, imperceptibility, and synchronization efficiency.

Zhijun (2021) introduced an approach that uses an information embedding technique based on Internet Protocol (IP) phone transcoding to compress publicly available information and create more room for data concealing. The initial phase of this technique was to either encode the original voice or analyze the payload packet. The voice stream is then produced using public encoding, bearing a resemblance to the original sound quality but with low payload. The primary payload field ultimately received the voice stream after transcoding, and the leftover space was made accessible for data concealment.

The primary objective of steganography is secure communication, which makes it impossible for anyone who is not permitted to access the actual message. Three factors should be considered when concealing information: robustness (the amount of modification the stego image can withstand before an adversary can erase concealed information), security (the ability to find concealed information in a cover medium), and the amount of information that can be concealed in a cover medium (Yadav & Dhankhar, 2015).

Steganography offers confidentiality as a security service, which is accomplished by integrating data into medium. Only a steganographic key can be used to expose the concealed information (Al-Shaaby, 2017). However, the method by which the information was concealed might also be used to prove identity. If steganography is not done correctly, the embedding method could be used as a form of identification and authentication and turn into a shared secret (Morkel, 2012). According to Domain (2018), embedded data cannot be subjected to an integrity check since it could have been altered unintentionally or on purpose, and the modifications made to the extracted data could not be apparent.

Steganography is vulnerable to a number of assaults, including structural and statistical attacks as well as visual ones. In visual attacks, the image is split into bit planes for further analysis with the use of computer or visual inspection to uncover the presence of hidden information. Structural assaults entail modifying the file's format while the concealed data is embedded. Finding these distinctive architectural changes can aid in determining the presence of data in an image (Hussain, Wahab, Idris, Ho, & Jung, 2018).

### 2.7.4 Authentication

Authentication is the process of identifying and authorizing an end user application, allowing the user to access the system or application (Basavala, Kumar, & Agarrwal, 2012; Omar, Hammod, Muamer, Muhammed, & Waleed, 2017). The goal of authentication is to verify that the provided information is a legitimate request from a specific individual. There are various methods of authentication, including those that use fingerprint, smart card, passwords, digital signatures, and biometrics, among others (Sepehri-Rad, Sadjadi, & Sadi-Nezhad, 2019). Authentication can be divided into the following three categories:

#### 2.7.4.1 Single-Factor Authentication Mechanism

Using a single attribute linked to their identity, such a PIN to unlock a phone, a person seeking access requests an authenticating party to confirm their identity with Single-Factor Authentication (SFA) (Rouse, 2017; Rahav, 2018). Numerous firms have adopted SFA due to its ease of use and simplicity. However, because it is susceptible to assaults like shoulder surfing, brute force, social engineering, and impersonation attacks, this kind of authentication cannot be used with financial institutions or other crucial transactions (Rahav, 2018). Using two-factor authentication is one potential mitigating technique for single-factor authentication against unauthorized access in mobile banking.

#### 2.7.4.2 Two-Factor Authentication Mechanism

When a customer requests access through the Two-Factor Authentication system (TFA), they ask the authenticating party to confirm their identity by providing two attributes, such as something they are or know that is connected to their identity (Rahav, 2018).

The following two tests must be passed by a TFA end-user protection system that requires authentication. In order to use this strategy, the user must provide two forms of identification: something that one has and the other must be something the user can recall. In order to be categorized as a TFA, the user should possess both a device (what you have) and knowledge (such as a PIN or password).

In this method, the user needs something (such a mobile phone) from which to obtain a token, and they also need something (the PIN or code after receiving it) in order to know it. Users must add their phone numbers to their accounts in order to achieve this (Lupu, Gaitan, & Lupu, 2015; Arshah, Hammood, & Kamaludin, 2018). Due to its dynamic nature, OTP is a disposable password that cannot be tracked. This means that after the password has been entered, it can no longer be used because its longevity is constrained.

As such, one of the possible mitigation strategies for TFA authentication against unauthorized access by MITM attacks, forgeries, and eavesdropping in mobile banking is utilization of Multi-Factor Authentication (MFA) (Ometov, Bezzateev, MäkitaloAndreev, Mikkonen, & Koucheryavy, 2018).

#### 2.7.4.3 Multi-Factor Authentication Mechanism

The technique of integrating various facets in a system's user is known as multi-factor authentication. Typically, these factors for authentication are something a person has, something they are (such biometric like a fingerprint), and something they know (like a text password, smart card, authentication token among others). While MFA can comprise many elements from the same category, the commonly used MFA normally consists of two factors from different categories (Papaspirou, Maglaras, Ferrag, Kantzavelou, Janicke, & Douligeris, 2021).

MFA authentication protocols are frequently used to verify users online. Client-server architecture is the essential component of online applications. When a client tries to connect to a remote server, it asks the web server for the required service. The majority of online users utilize web browsers as their primary Internet client software to interact with the internet. Many Application Programming Interfaces (APIs) and other system data are accessible to web servers via web browsers using the HTTP protocol, which is used by the client-server architecture to transfer information. Web browsers can use a variety of information vectors exposed by the rising complexity of modern web applications to give users the features and services they desire.

These information vectors might be specific and helpful in recognizing a certain system. The process of identifying a device or browser by using information vectors provided by a web browser is known as browser fingerprinting. A browser fingerprint may or may not work as a reliable authentication method, depending on how distinctive it is. There are two ways to figure out a browser's fingerprint. The first is the fingerprint, which may be obtained from publicly accessible browser data, including time zones, IP addresses and HTTP headers. The second method involves the web server actively gathering a browser's fingerprint.

Constructed queries are sent by a web server, which monitors the web browser's response. The behavioural differences across browsers are caused by variations in hardware and software configurations and these differences can be utilized as a browser fingerprint. Combining many information vectors can result in a more unique browser fingerprint (Alaca & Van Oorschot, 2016). Because browser fingerprints can be used to monitor users, research in this area is primarily concerned with privacy and fingerprinting prevention (Durey, Laperdrix, Rudametkin, & Rouvoy, 2021). Additionally, user security and device authentication can benefit from the use of browser fingerprints (Alaca & Van Oorschot, 2016; Durey, Laperdrix, Rudametkin, & Rouvoy, 2021).

During authentication, users can submit a knowledge factor in a different format with graphical passwords. Because people can recall visual representation better than text, these passwords may be easier for users to remember. Drawing a pattern on a grid, which has been made popular by mobile phones is the most popular way to construct a graphical password. Different graphical passwords with different levels of success have been used. However, they also create wider openings for attacks such as shoulder surfing (Kaka, Ishaq, & Ojeniyi, 2020). Attacks on MFA can be mitigated using a combination of techniques such as facial biometric authentication and utilization of OTPs.

### 2.8 Evaluation of Hybrid Algorithms

Hybrid algorithms combine two or more other algorithms to solve a problem (Stallings, 2014; Forouzan, 2007). Typically, this is done to combine desired characteristics of each algorithm leverage making the overall algorithm better than individual algorithms used solely. For example, in their study, Gupta and Sharma (2012) discussed about a new hybrid encryption scheme that integrates Diffie Hellman and RSA to get data security for services in the cloud. This technique aimed at leveraging the strengths of both the algorithms. Therefore, the idea of this integration is to exploit secret key cryptography and public key cryptography which then offers robust security for a communication over the internet.

#### 2.8.1 Complexity Evaluation

The main criterion for assessing complexity is the ciphering method's security. Security is impacted by the algorithm's complexity. Consequently, complexity might be considered an assessment criterion for developed encryption systems. Additionally, an algorithm is considered secure if it remains unbreakable even when executed on a machine with limited memory and processing power. However, we may say that a system has a measured level of security if we study this method on a computer with a large memory and a high processing speed (Krishnamurthy & Ramaswamy, 2009).

The complexity of an algorithm is measured in terms of time and space complexity. The sort of algorithm determines how the complexity of the algorithm is assessed. For instance, the complexity of a cipher method varies depending on whether it is monoalphabetic or polyalphabetic. While the private key used determines complexity of the transposition cipher and randomness determines complexity of stream cipher (Krishna & Babu, 2010).

### 2.8.2 Statistical Evaluation

This section determines the value of the secret keys that substitution ciphers utilize using a variety of statistical criteria, such as the index of coincidence and modified statistical tests that rely on the variances, skewness, kurtosis and measure of the expected value (Etienne, 2018).

#### 2.8.2.1 Histogram of Ciphered Text

Calculating the histograms of the ciphertexts produced by various cipher algorithms is necessary for the process to use the histogram concept as an evaluation parameter. By analyzing the features of the histogram generated by a multiplicative cipher method using different keys using a rectangular window of a predefined length, an attempt is made to ascertain these variables (Mali, Chakraborty & Roy, 2015). This database will function as a trustworthy source for information on the multiplicative secret and multiplicative cipher that are employed.

### 2.8.2.2 Modified Statistical Evaluation

Variance, skewness, and kurtosis are statistical factors that are utilized as expected values in evaluations. These values are used as evaluation parameters for algorithms of the monoalphabetic and polyalphabetic ciphers. The relationship can be utilized to compute the expected value of the ciphertext or plaintext mathematically (Neamah, 2015).

$$E(x) = \frac{1}{\mathrm{n}}\sum_{i=A}^{z}(\mathrm{p_i} \times \mathrm{F_i}) \qquad (2.4)$$

The anticipated value range for the cipher text produced by monoalphabetic algorithms is smaller than the expected value range for cipher texts produced by polyalphabetic algorithms. The encryption procedure is considered monoalphabetic if the predicted value falls within the range of 0.061 to 0.069 contingent upon the length of the ciphertext. If the predicted value is more than 0.069, the cipher algorithm will be polyalphabetic, depending on the length of the ciphertext and the secret key. The following equation provides a numerical representation of the plain text's variance.

$$\mathrm{var(x)} = \sum_{i=A}^{z}(\mathrm{P_i})^2 - 0.038 \qquad (2.5)$$

(2.5) illustrates variance in which the encryption procedure is considered monoalphabetic if the variance falls within the range of 0.96 to 0.97, contingent upon the ciphertext's length. When the variance exceeds 0.97, the cipher procedure becomes a polyalphabetic one, contingent on the length of the ciphertext and the secret key.

## 2.9 Information Theory

In a general communication system, the seminal work of Claude Shannon's information theory (Shannon, 1948) consists of multiple elements, such as a source that transmits a message for another entity to receive and interpret as its own. The transmitter transforms the message generated at the source into a signal suitable for transmission. The communication must then be secured so that the entity in charge of sending the data also performs the encoding operation. To transfer the signal from a transmitter to a receiver, a channel is also needed. The channel may be transmitted wirelessly, using fiber optics or via a cable. In the end, the message is received by the receiver after they have taken the signal and pieced it together (Zubair, Ali, & Anam, 2023).

The metrics of information theory can be used as assessment parameters. For example, the substitution monoalphabetic cipher system can be distinguished from the polyalphabetic encryption system using the unicity distance. The unicity distance can be defined as the minimal number of letters needed to decipher a cipher text, as determined by the following equations (Sattar & Sadkhan, 2017; Sattar, Sadkhan, Sabiha, & Jawad, 2020):

$$U_d = \frac{H(k)}{R} \tag{2.6}$$

Where:

$$H(k) = \sum_{A}^{z} \mathrm{P_c(k)Log_2} \frac{1}{\mathrm{P_c(k)}} \tag{2.7}$$

Where R is the redundancy and is the essential entropy (Dalkilic & Gungor, 2000). Redundancy is the quantity of letters that can be omitted from a word or sentence without altering its meaning. When the unicity distance for the cipher text is greater than or equal to 1.35, the encipher algorithm is polyalphabetic; however, when it is less than 1.35, the cipher algorithm is additive (Alibadi & Sadkhan, 2018). Important entropy parameters from information theory and unicity distance help to provide practical evaluation metrics for security measures.

Information theory was used in this study because of its fundamental metric of entropy, which quantifies the amount of uncertainty or surprise associated with outcomes (Shannon, 1948). Entropy is helpful in determining the typical number of bits needed to encode message components in an image in order to measure the degree of security of a system.

## 2.10 Summary of Research Gaps

Most mobile banking systems are at risk from various attacks especially when utilized to perform mobile banking over wireless networks. The methods employed to stop the different forms of attacks contribute to an increase in both time and expense of visiting physical banks. Fingerprints may be quickly and inexpensively be duplicated utilizing silicon artificial fingerprints to avoid detection methods in fingerprint systems. A paradigm for evaluating aspects impacting user interaction with touch interactions on mobile devices was presented by (Ellavarason, Guest, & Deravi, 2018).

Customers are occasionally warned by banks not to divulge one-time passwords to third parties, including bank staff or to click on phishing links from con artists. In this context, it is advised that organizations should be able to recognize and block fake links that are shown as advertisements, particularly on social media. Additionally, users need to be wary of social engineering attacks (Yildirim & Varol, 2019).

Currently, mobile banking applications use username and password security mechanisms, which are easily accessed by simple guesswork and password hacking. Combining user identity, password and fingerprint recognition system are some of the techniques which can significantly help to decrease potential security risks. The security and privacy features for mobile banking need to be improved (Sharma & Mathuria, 2018).

Putra, Sadikin, and Windarta (2017) created a secure online banking paradigm for protecting information sent between communicating parties, particularly as part of authentication process. The schema supports many layers of authentication. It starts with pair-based text authentication which entails adopting key pairs in every set and interchanging verification data with the server before administration login details are input. An OTP is processed by a contactless smart card after being sent to the recipient through SMS. However, this design is susceptible to security issues, such as when the user misplaces or has their phone stolen.

Majority of the mobile banking applications that were examined in 2016 had flaws, with access to private customer data being the greatest danger. In 2017, fraud, identity theft, and access to consumer banking data were all factors that affected mobile banking applications. Banks should focus more on a suitable architecture, thorough articulation of technology requirements, and secure development in order to prevent these dangers.

Consequently, testing of applications and security measures is required (Positive Technologies, 2018).

A mechanism for protecting mobile banking applications from MITM attacks was put forth by Luvanda (2014). In his study, majority of mobile applications were found to be exposed to MITM assaults because of poor administration of TLS conventions and poor blueprints of programming outlook and utilization of code libraries. The study found implementing public key cryptography and protocol encryption are the best available defense techniques against MITM attacks. The study points out that asymmetric cryptographic schemes singly are open to MITM assaults. Banks should conduct penetration testing in software design to uncover existing and new flaws for examination and improvement (Blyskal, 2015). Furthermore, mobile banking applications should incorporate and employ biometric systems like fingerprint reading, administration of hybrid cryptosystems that incorporate more than one algorithm to add additional layer of safety to stop assaults on mobile banking.

DoS attacks utilize potential weaknesses and defects in the application layer conventions (Singh, Singh, & Kumar, 2018). The flaws emanate because of insufficient effort on the part of designers and developers to create secure protocols. Functionality is sometimes given more weight than security when designing protocols (Tripathi & Hubballi, 2018). As a result, there is a significant attack surface left behind, which adversaries might take advantage of to execute application layer DoS operations. These attacks use very few resources to depose a system by significant computational power and network capacity. Application layer DoS attacks mostly affect client-side services (Gonzalez, Antoine Gosselin-Lavigne, Stakhanova, & Ghorbani, 2015).

Numerous steganographic approaches have been combined with a variety of algorithms, including AES cryptographic scheme, and random key generation. According to recent studies, combining both strategies can result in a framework that is more reliable, and robust than when individual algorithms or authentication are utilized solely (Taba, Rahim, Lasta, Hashim, & Alzuabidi, 2019).

A technique that merges DNA sequence and hyperelliptic curve cryptography was proposed by (Vijayakumar, Vijayalakshmi, & Zayaraz, 2016). From the sender's point of view, this algorithm is divided into three stages. Firstly,the cover photo and secret message's pixel values must be converted into matching DNA triplet values using characters. Secodnly, those triplet values must then be converted into binary values. The last step is to construct the stego photos by applying XOR login to the binary values of the cover and secret images.

To protect confidential information during transmission, encryption system and steganographic algorithm was suggested by Karthikeyan, Kosaraju and Gupta (2016). According to this method, confidential information would be encrypted using AES-128 key encryption technique before being encoded into a Quick Response (QR) code. The encrypted message is then translated from its UTF-8 format into base 64 in order to make sure it can be processed further. The secret information is then safely conveyed via an appropriate carrier that conceals the scrambled QR code. In order to achieve digital image steganography, LSB approach was used. The confidential information is recovered from the courier through a decipher procedure when the receiver receives the message.

A technique that performs a gray-level adjustment for true color photos and relies on secret key cryptography and image transposition was proposed by (Muhammad, Ahmad, Sajjad, & Zubair, 2015). Here, confidential information and secret key are first encrypted utilizing a variety of encryption algorithms prior to being included in a cover picture. Prior to embedding, the supplied picture should be modified. The suggested solution uses picture interchange, bit-XOR, restructuring bits, stego key-based encryption and grey-level adjustment to provide five degrees of protection. Because of this, data recovery is a challenging task for attackers.

An information-hiding system combining LSB steganography and RSA cryptography was proposed by Shailender, Ankur, and Bharat (2012) and tested using MATLAB 7.01. The LSB's lower efficiency is the system's weak point. Secret audio and image files were concealed using a steganography system by (Kolla, 2019) using LSB. The hashing algorithms used were Message Digest version 2 (MD2), MD4 and MD5 and the file formats were BMP for graphics and WAV for audio files. The encryption algorithms utilized were DES and RC2.

Adebayo, Ganiyu, Osang, Ajiboye, Olamilekan and Abdulazeez (2022) presented a solution for data privacy employing cryptography and steganography. To increase security and preserve data privacy, the Most Significant Bit (MSB) steganography technique was utilized. Python was used as the programming language to implement the system.

Steganography and cryptography were compared in a study by (Almuhammadi & Al-Shaaby, 2017). They examined different strategies for combining cryptography and steganographic techniques into one system. They also classify and compare these methods according to the steganographic approach, the file type utilized as a cover image and the encryption algorithms used. They concluded that techniques starting with cryptography are more common than those starting with steganography and provide better security with less exposure to encrypted data. The availability of additional capacity for less secret information was the only benefit of steganographic methods.

From the summary of research gaps, most studies concentrated on building secure systems that utilized user IDs and passwords, OTPs, various forms of authentication, various symmetric and asymmetric algorithms, steganography and a combination of cryptography and steganography. Surprisingly, most of these studies developed security frameworks for different systems other than mobile banking applications.

Three symmetric encryption algorithms: AES, Blowfish and Twofish were considered. Blowfish was found to have weak key and second order differential attacks. While Twofish was considered more secure than Blowfish, its major disadvantage was susceptibility to truncated differential attacks. Finally, AES is adaptable to various key lengths and provides flexible security, making it widely employed method for safeguarding sensitive information in digital communication. However, its major drawback is that it is susceptible to side channel attacks and known plaintext attacks (Shallal & Bokhari, 2016).

Similarly, three asymmetric encryption algorithms: RSA, ECC and DSA were considered and assessed their strengths and weaknesses. Despite being a reliable encryption method for many years, RSA can occasionally lose its effectiveness, especially when handling larger files or applications with limited processing power. The computational cost of RSA increases with data size, influencing the length of time needed for encryption and decryption (Boussif, 2022).

However, ECC can achieve the same level of security with lower key lengths, and thus a viable substitute that can improve performance and reduce computational overhead particularly in settings where resources are limited. For key exchange and digital signatures, ECC is more efficient than RSA since it provides the same level of security with much smaller key sizes (Mhato & Yadav, 2017).  However, ECC is susceptible to side channel attacks, power analysis attacks, electromagnetic analysis attacks, error message attacks, fault analysis and timing attacks.

Lastly, DSA offers crucial services including confidentiality, authentication, non-repudiation and data integrity. It should be noted that DSA does not use a private key to encrypt or a public key to decrypt message digests. Instead, it creates a digital signature consisting of two 160-bit values using scientific abilities (Technopedia, 2013). However, DSA are susceptible to key-only attacks, known-message attacks, chosen-message attacks and forgeries.

When comparing cryptographic encryption algorithms with steganographic algorithms, we find that cryptographic encryptions algorithms are classified into two categories depending on the number of keys they operate on the plaintext. They are symmetric and asymmetric encryption algorithms. Some of the symmetric encryption algorithms discussed in this study were AES, Blowfish and Twofish. Asymmetric encryption algorithms were RSA, ECC and DSA.

Steganographic algorithms, on the other hand, fall into two categories. Hex symbol method (uses hex code) and traditional methods (uses binary code). Conventional techniques use networks, text, images, audio and video as cover media. However, compared to old approaches, the Hex symbol method is more reliable, offers a greater level of protection and can embed more information.

Steganography provides authentication, confidentiality and identification security services, whereas cryptography provides non-repudiation, authentication, data integrity, confidentiality and identification security services. These differences can be seen when comparing steganography and cryptographic encryption algorithms. Furthermore, while cryptographic encryption techniques are vulnerable to cryptanalysis assaults, steganographic algorithms are subject to steganalysis attacks.

Table 2.13

Techniques used to Secure User Data on Transit in Mobile Banking

| Author | Technique | Drawback | Comment |
|---|---|---|---|
| Akinyede & Esese (2017) | SHA-512<br>AES<br>OTP | System not immune to future security threats and attacks | suggested a safe mobile electronic banking system that uses OTP for authentication, SHA-512 for encryption and decryption, and AES |
| Vincent,Okediran, Abayomi-Ali, & Adeniran (2020) | ECC | Phone data are not encrypted directly on the mobile phone, but on the payment gateway | Key size, security strength, computing power, memory capacity, encryption and decryption time, and mobile phone battery life were all taken into consideration when evaluating the suggested technique. The outcome demonstrates that the plan offers privacy, secrecy, and integrity. |
| Sankpal, Rathod, Kodre, Sayyed & Sayta (2017) | LDEA<br>AES &<br>Haversine | encryption of this protocol is limited to a static location and cannot be used in dynamic location | The suggested solution used virtualization to create and implement a location-based algorithm that only decrypts the ciphertext in a designated area. |
| Sodhi & Gaba (2018) | SHA-160 | Susceptible to collision attacks | The suggested system demonstrated how effective the SHA-160 approach was in terms of throughput and randomization. |
| Nwoye (2015) | RSA cryptosystem | Slow, more Central Processing Unit time and battery power, data compromised by man-in-the-middle | In order to solve the security and privacy concerns with credit card information in e-commerce transactions, the suggested system integrated an RSA e-commerce security system (RSA-ESS). In order to safeguard payment information and |

| | | | |
|---|---|---|---|
| | | | complete an e-commerce transaction quickly enough, this system uses RSA. |
| Zay (2019) | RSA Cryptosystem | The proposed system can be used only for the security and privacy of payment information | This method bypasses the online merchant and sends a customer's financial data, like as credit or debit card information, directly to a payment gateway, also known as TTP. |
| Joshi (2018) | Steganography | Once detected the message is known | suggested a method for concealing text within a picture. The technique offers a more secure methodology and is resistant to authentication assaults. |
| Khelifi et al., (2013) | AES using Open-Source Key encryption algorithm | Open source exposes the source code to examination by everyone, both the attackers and defenders | suggested a brand-new, open-source, AES-based symmetric key encryption method. By accelerating the processing of the method, it resolves the computational overhead problem and enhances the security of e-banking services. |
| Kumar (2015) | DSA | Since anyone with access to a private key can use it to transmit messages signed to public key holders, losing one could have major repercussions. The recipients will believe that the communication originated from the actual private key holder since the public key will accept these messages as authentic. | DSA does not provide secrecy of data. To provide secrecy, other methods such as encryption and decryption should be used |
| Chauhan, Jyotsna, Kumar, & Doegar, (2017). | LSB | Variable block size data encryption | Efficiency can be increased |

Table 2.13 is a summary of the different cryptosystems developed for different applications. It demonstrates different authors and particular years that the cryptosystems were developed, and a comment on each of the cryptosystems. The comment section discusses on the proposals made and the achievement made so far from the cryptosystem.

The drawback section highlighted known weaknesses of the proposed cryptosystems. It can be noted from the table that some of the cryptosystems developed were implemented on a single cryptosystem such as DSA, LSB, AES, RSA, SHA-160 and ECC. However, other authors developed cryptosystems that combine more than two algorithms to enhance data security. They include a combination of SHA-512, AES and OTP, and a combination of LDEA, AES and Haversine algorithms. Table 2.13 also demonstrates that most of the cryptosystems highlighted on different application areas. Other cryptosystems focused on electronic banking systems. Therefore, this study found an existing gap and focused to develop a secure and robust cryptosystem by combing AES and LSB algorithms for mobile banking applications.

# CHAPTER THREE

# RESEARCH METHODOLOGY

## 3.1 Overview

In this chapter, the following are presented and discussed; the research philosophy and designing, sampling techniques and dataset used, the proposed hybrid algorithm for protection of data on transit in mobile banking applications, and the results of the study. The ethical considerations during the study are also discussed.

## 3.2 Research Philosophy

Positivist methodology was used in this study. Positivism is compatible with the deductive logical reasoning processes used in the natural sciences in which a conclusion is arrived at from multiple premises that are generally assumed to be true. Positivism relies on research evidence that can be independently verified and that is gathered in an impartial manner in order to produce objective findings and develop general scientific rules (Zekauskas, Vveinhardt, & Andriukaitiene, 2017).

Positivism was used in this study because it deals with contemporary phenomenon within a real-life context that is problems affecting the society and seeks practical solutions towards those problems.  Therefore, positivism sought to respond to the problem of user data on transit in mobile banking applications which is apparently susceptible to security threats and propose a hybrid algorithm that is robust against weaknesses inherent in mobile banking applications.

### 3.3 Research Design

The broad framework of research design defines the overall strategy for carrying out research work. It specifies goals, data collection and analysis techniques, results, and conclusions. This study utilized both descriptive research design and Data Science (DS). One method for precisely describing an existing occurrence is descriptive study. As a result, this approach was used to evaluate how well mobile banking apps that let users access banking services from a distance operate, evaluate security risks that could compromise user data while it's in transit, and evaluate methods for protecting user data while it's in transit within mobile banking apps.

DS was also adopted. It is used to improve the execution of system design and its outcomes as an IT artifact (Dresch, Lacerda, & Antunes, 2015). DS enable researchers to expand their method library by including a solution-based methodology. Theorizing and theory building take place prior to, throughout, after, and as a result of Design Science Methodology (DSR). The general goal of DSR is to generate insights on how to develop new solutions that effectively tackle pressing problems (Kruse, Seidel, & Purao, 2016). In its most basic form, this is a problem-solving methodology used in the development, design, and construction of application systems (Geerts, 2011). In this study DSR was used to develop and evaluate the proposed hybrid algorithm in this thesis. DS is thus used when coming up with artifacts such as algorithms, interfaces and system design approaches (Peffers, Tuunanen, Gengler, Rossi, Hui, Virtanen, & Bragge, 2020).

On the other hand, desktop research was used to gather secondary data, which came from peer reviewed and refereed academic publications including books, conference proceedings and journal articles. Publications were collected from numerous academic research databases, including Association for Computing Machinery (ACM), Emerald Insight, IEEE, Springer and Google Scholar.

### 3.4 Sampling and Sampling Techniques

Sampling is an approach of selecting a suitable sample or a representative portion of a population with the intention of identifying the parameters or traits of the entire population. A sampling procedure is the process of choosing representative elements from a population (Sharma, 2023).

This study utilized purposive sampling. Critical case sampling was utilized to select six color pictures from USCI-SIPI database for test simulations. Critical case sampling permits logical generalization and maximum application of information to other cases. The six pictures were chosen because of the following reasons; firstly, fewer images reduce the risk of detection. This is because it becomes harder for steganalysis tools to identify patterns or anomalies that indicate hidden data. Secondly, the six images were chosen for quality preservation. Altering a large number of images can degrade their quality and thus making the changes more noticeable. Thirdly, managing and processing a smaller number of images is more efficient in terms of computational resources and time. This is particularly important for real-time applications where speed is crucial. Lastly fewer images mean fewer point of failure. This implies that if one image is compromised, the entire hidden message is not exposed. This adds an extra layer of security to the steganographic process.

Therefore, critical case sampling involves identifying and selectively choosing specific samples. Data is collected from these samples, and the findings are then generalized to other samples that share similar characteristics (Nyimbili & Nyimbili, 2024). The dataset from which six color pictures were selected is described in the next section.

### 3.4.1 Dataset

USC-SIPI image dataset was utilized for image processing. USC-SIPI image database is a collection of digitized images containing 306 photos in four volumes. The collection was obtained from SIPI Image Database – Miscellaneous (usc.edu)[1]. USC-SIPI image dataset is a well-known benchmark sample created by the University of Southern California's Signal and Image Processing Institute (USC-SIPI Image Database, 2023). USC-SIPI dataset is used for quality analysis of images in steganography (Das, 2022). Furthermore, USC-SIPI database has been used in digital image steganography for theoretical foundation and image processing (Panwar, Damani, & Kumar, 2018).

Six color pictures were selected from USC-SIPI dataset namely airplane, female, house, couple, peppers, and sailboat. These images were selected because they are currently stored in Tagged Image File Format (TIFF). TIFF allows for lossless compression (Unit 4 Lab 4, 2023) as well as high-quality images with large image size which is good for data hiding (Majjed, 2023). These images were selected from the dataset to evaluate image quality during simulation tests. Images used for test simulations are shown in Figure 3.1.

The Red Green Blue (RGB) color Model was used in this study. RGB cover images were used because color images have large space for hiding information than grey images.

[1] https://sipi.usc.edu/database/

Moreover, color pictures make hardly noticeable changes to the RGB color model. (Ayyoub, Nader, & Al-Qadi, 2019).

Figure 3.1

Images Utilized for Simulations (USC-SIPI Image Database, 2023).

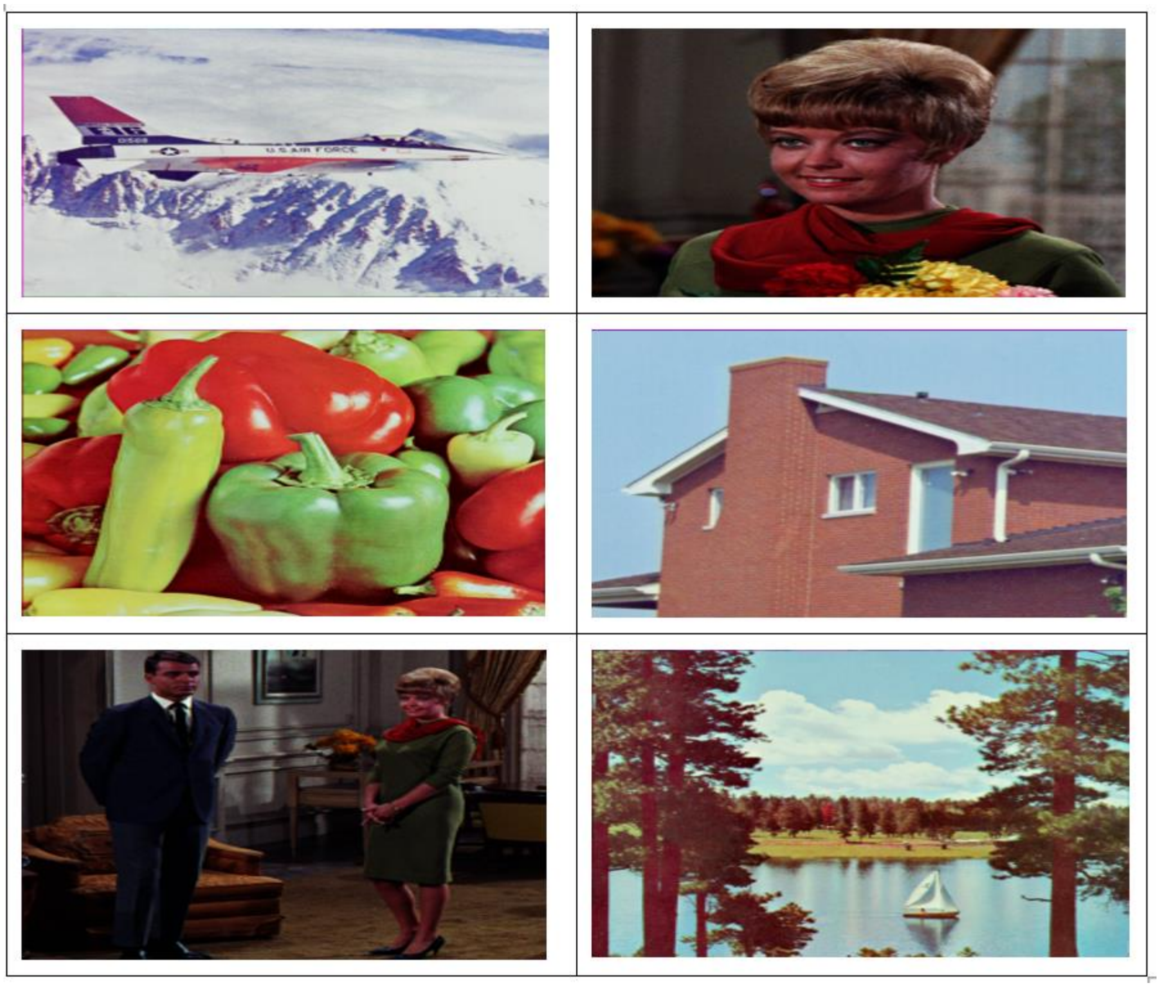


Figure 3.1 illustrates pictures utilized for test simulations. The most common hue models are RGB, Cyan, Magenta, Yellow, and Black (CMYK), Luminance Component, Chroma blue and Chroma, red (YCbCr), which are derived from RGB colour space that represents colours the way the human eyes perceive and interpret colours (Muhammad, Ahmad, Farman, & Zubair, 2014).

## 3.5 LSB Algorithm

LSB is a well-known steganography technique used to embed secret messages within a cover image (Yigit & Karabatak, 2019; Gutub & Al-Shaarani, 2020). This technique includes two sub-methods: the insertion-based method and the substitution-based method (Arora, Singh, Thakral, & Jarwal, 2016). The insertion-based method embeds secret data by increasing the size of the image, while the substitution-based method replaces the image's bits with secret data without altering the image size (Molato & Gerardo, 2018). This study adopted LSB substitution method because it is easy to implement with minimal impact on image quality as changes made to LSB are imperceptible to the human eye. Moreover, LSB allows for a relatively high amount of data to be embedded within an image.

### 3.5.1 Embedding Algorithm

This algorithm illustrates a step-by-step on how a secret message is embedded into a cover picture to produce a stego-picture. The stego-picture can then be sent to a recipient.

Table 3.1

Embedding Algorithm (Aye, 2018).

| Embedding Algorithm | |
|---|---|
| Step 1 | Open the cover photo and private message. |
| Step 2 | Use the AES technique to encrypt the secret message. |
| Step 3 | Transform the message into binary code. |
| Step 4 | Start the sequential encoding process to recognize the cover picture's pixels. |
| Step 5 | Sequentially positioned pixels' LSB will change based on message bit values. |
| Step 6 | The altered pixel value is returned to its original location. |
| | The LSBs of the image pixels are changed based on the message data size. |
| Step 7 | Save the stego-picture, then send it. |

It starts with loading a cover picture into a system followed by uploading a secret message. The message is then encrypted using AES algorithm and converted into its binary bits. Sequential encoding is initialized and the message bits are substituted into the image pixels. The image is finally saved as a stego-image and sent to the recipient. Figure 3.2 illustrates LSB embedding process.

Figure 3.2

LSB Embedding Process

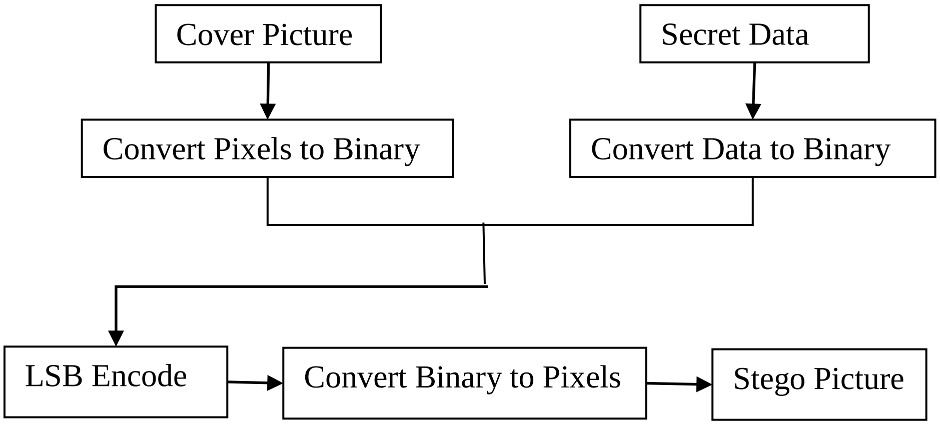


Figure 3.1 illustrates the embedding process. It involves uploading cover picture onto the system and uploading secret data. The, apply LSB encoding. The outcome of this process is a stego picture that contains hidden secret data.

### 3.5.2 Decoding Algorithm

Decoding algorithm provides a step-by-step process which is used to retrieve encrypted secret message hidden in the stego-picture. Table 3.2 illustrates the process of retrieving hidden message from a stego-picture.

Table 3.2

Decoding Algorithm (Aye, 2018).

| Decoding Steps | |
|---|---|
| Step 1 | Open the stego-picture file. |
| Step 2 | Read the LSB of every recognized pixel in the stego-picture. |
| Step 3 | Apply AES decryption algorithm |
| Step 4 | To extract the message from the stego-picture, use sequential decoding. |

This process is used to load the stego-picture, followed by reading LSB of each pixel, after which AES decryption algorithm is applied. Sequential decoding is initiated in order to regain the secret message from the stego-picture.

Figure 3.3

LSB Encoding Process

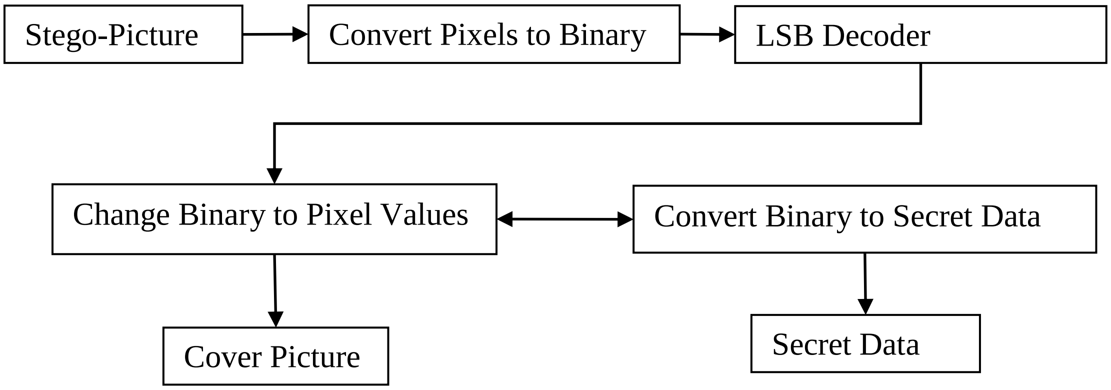


Figure 3.2 illustrates the decoding process in which a stego-picture is input into the system. Then pixel values of stego-picture are converted to binary representation.  When LSB decoding is applied, secret data is retrieved.

In the proposed algorithm, in section 3.6, six cover pictures were used to conceal encrypted secret messages onto cover picture pixel values using LSB substitution technique. It involves repeatedly iterating through an image's pixels and substituting secret message binary digits with LSB values of picture pixels (Astuti, Setiadi, Rachmawato, & Sari, 2018). The basic notion underlying this technique can be demonstrated using the hidden letter "S," which has the ASCII code 83 and the binary value 1010011. The LSB picture elements contain binary bits encoded within it. Consider the following example in Table 3.3.

Table 3.3

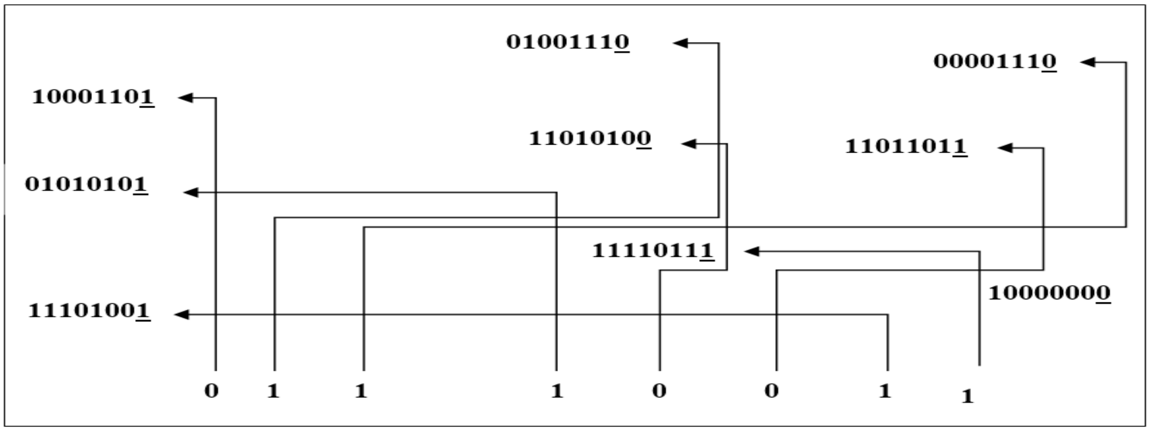
Image Pixel Values Prior to Substitution of Message Binary Digits (Astuti, Setiadi, Rachmawato, & Sari, 2018).

| | | |
|---|---|---|
| 10001101 | 01001110 | 00001110 |
| 01010101 | 11010100 | 11011011 |
| 11101001 | 11110111 | 10000000 |

Image pixels are represented in binary form in Figure 3.3. Figure 3.4 demonstrates how binary digits 1010011 of letter S are substituted in the binary digits of a picture element.

Figure 3.4

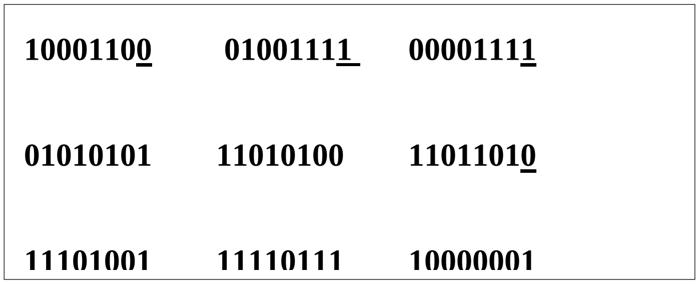
Letter S Binary Digits Substituted in an Image Pixel (Astuti, Setiadi, Rachmawato, & Sari, 2018).



The letter S is seen in Figure 3.4 with binary digits1010011 replaced on the picture elements’ binary digits. The new picture pixels occur when all elements of S have been consecutively substituted on the picture pixels as shown in Figure 3.5. Be aware that the picture pixels stay intact when value 1 of a message is swapped for picture pixel with value 1. The same is true when a zero is replaced with a zero of the picture pixel value. The picture pixel LSB, however, transforms to 1 the message binary value of 1 is replaced with picture pixel with value 0, and vice versa. After replacing letter S binary values with picture pixel values, the new modified picture pixel values are demonstrated in Figure 3.5

Figure 3.5

Changed Image Pixel Values after LSB Substitution (Astuti, Setiadi, Rachmawato, & Sari, 2018).



| | | |
|---|---|---|
| **10001100** | **01001111** | **00001111** |
| **01010101** | **11010100** | **11011010** |
| **11101001** | **11110111** | **10000001** |

The appearance of the original picture is unaffected by changing the picture pixel values, hence the resultant picture resembles the cover picture almost exactly. Even though the binary form of the alphabet S replaced into the first 7 bytes of the picture pixels, only the emphasized values were altered. Every basic color has 256 possible shades, so altering the LSB of a pixel only slightly alters the intensity of the colors. The message is effectively hidden by these progressions since they are invisible to the naked eye (Farham & Alwan, 2018).

By utilizing LSB substitution steganography approach, the LSB image pixel values are replaced by message values to produce a new image with invincible message that cannot be easily conceived by human vision mechanism. The new image is saved and forwarded to a recipient via internet medium. LSB has been used successfully to improve payload capacity, security, and integrity tests (Mustafa & Wisam, 2018).

## 3.6 AES Algorithm

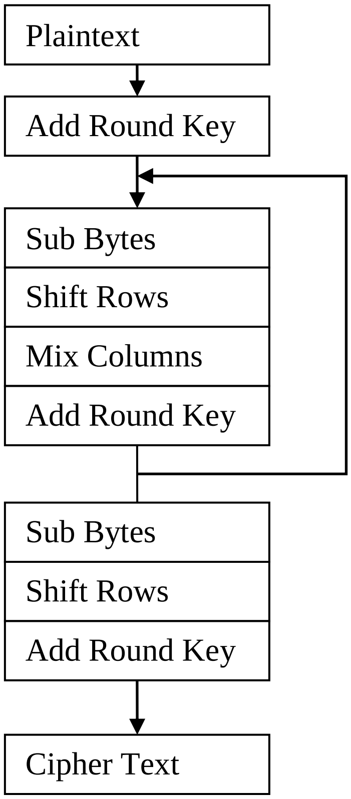
AES is one of the most widely used symmetric encryption algorithms. Each round of the encryption process includes four operations: SubBytes, ShiftRows, MixColumns, and AddRoundKey (James, Kumar, & Scholar, 2016; Awad, 2018). The algorithm supports key lengths of 128, 192, and 256 bits, and the number of rounds varies accordingly: 10, 12, and 14 (James, Kumar, & Scholar, 2016; Arya, 2016; Kak, 2019).

Each round of AES involves four operations and is repetitive in nature. The output from the first round serves as the input for the second round, with the process repeating with a new set of keys for each round. This continues until the final round, which omits the MixColumns operation. The state array produced in the last round becomes the ciphertext ready for transmission.

AES employs four stages for encrypting and decrypting messages: SubBytes, ShiftRows, MixColumns, and AddRoundKey. Among these stages, ShiftRows contributes the least to the security of AES, as it involves a straightforward, linear operation (James, Kumar, & Scholar, 2016; Kak, 2019). Consequently, this study aims to propose a more secure technique by modifying the ShiftRows step of AES. AES is widely used for information and data security within symmetric ciphers. In AES, each round of the encryption process involves four operations: SubBytes, ShiftRows, MixColumns, and AddRoundKey (Chowdhury, Mahmud, Kamal, Hamid, & Member, 2018; Verma, Kaur, & Tech, 2016). Figure 3.6 illustrates AES encryption algorithm diagram.

Figure 3.6

AES Encryption Block Diagram (Rajaiah, Basha, Hidayathulla, Reddy, Kusuma, & Ahmmad, 2023).



### 3.6.1 AES Encryption

All operations in AES are byte-oriented. A 128-bit plaintext is organized into a 4x4 matrix consisting of 16 bytes. The primary steps in AES encryption are SubBytes, ShiftRows, MixColumns, and AddRoundKey (Deshmukh, 2016):

i. In the Sub Bytes Transformation, or SubBytes, is a stage in which every byte of the encrypted text is changed using a common lookup table known as the S-box. The S-box uses a byte's initial hexadecimal value to find the row index and its second hexadecimal value to finds the column index.

ii. In the ShiftRows transformation, the rows of the matrix are circularly shifted to the left. Row 0 remains unchanged, Row 1 is shifted left by 1 byte, Row 2 is shifted left by 2 bytes, and Row 3 is shifted left by 3 bytes.

iii. In the MixColumns transformation, each column of the matrix is multiplied by a corresponding column of a predefined matrix.

iv. In the AddRoundKey step, the resulting matrix is XORed with the expanded key produced from the initial key.

### 3.6.2 AES Decryption

Decryption is the reverse of encryption, where the ciphertext is converted back to plaintext. This process involves the inverse operations of SubBytes, MixColumns, and ShiftRows, as well as the use of the inverse S-box.

i. Inverse SubBytes Transformation involves substituting each byte in the matrix using inverse S-box table to obtain a new matrix.

ii. Inverse ShiftRows Transformation involves circularly shifting each row of the matrix to the right.

iii. Inverse MixColumns Transformation is the reverse of the MixColumns transformation.

iv. Inverse AddRoundKey Transformation involves selecting the round keys in reverse order and XORing them with the state matrix.

The security properties of the AES encryption technique and the LSB substitution algorithm were merged in the suggested hybrid algorithm. An essential security feature made possible by the LSB replacement approach is the ability to conceal a secret message on a cover image in order to prevent detection of its presence. However, the AES method added a crucial security element by encrypting a secret message to prevent enemies from decrypting it. Consequently, an extra degree of security was offered by combining the two techniques.

### 3.7 Performance Analysis of the Proposed LSB-AES Hybrid Algorithm

The complexity of the proposed LSB-AES hybrid algorithm was analysed in terms of time and space complexity. Time complexity is the amount of time it takes to embed secret data into the cover image while space complexity is the amount of memory required during the embedding process including the memory required to store the cover image and secret data (Alanzy, Alomrani, Alqarni, & Almutairi, 2023).

The time complexity for embedding data using LSB steganography is (O(n)), where (n) is the number of pixels in the image and O stands for “order of” in Big O notation to describe the upper bound of an algorithm’s time and space complexity. The dimensions of the six images that were utilized for simulation tests were: airplane, peppers and sailboat with 515 by 512 has 262,144 pixels while female, house and couple with 256 by 256 has 65,536 pixels. Large number of pixels allows for a significant amount of data to be hidden and increases imperceptibility. Similarly, the time complexity for extracting data from the stego-image is (O(n)), as each pixel needs to be checked to retrieve the hidden data.

The space complexity for AES encryption is (O(n)) because it uses a fixed amount of memory for the key and state arrays regardless of the input size. The proposed LSB-AES algorithm's space complexity remains O(n)), as both LSB steganography and AES encryption do not require additional space proportional to the input size. Thus LSB-AES hybrid algorithm has a time complexity of (O (k\ cdot n)) and space complexity of (O(n)) which makes it efficient in terms of space but dependent on the size of the unput data for time complexity where k is a constant and in the case of AES encryption it is the number of rounds which is typically 10, 12 or 14 rounds depending of the key size (Alanzy, Alomrani, Alqarni, & Almutairi, 2023; Roy & Islam, 2022; Tiwari & Gangurde, 2020).

Performance analysis of the proposed LSB-AES hybrid algorithm involved evaluating security using PSNR, MSE, entropy and robustness and resilience using attack resistance to MITM. Higher PSNR values indicates better quality of the stego-image, meaning that the changes introduced by embedding secret data are less noticeable while lower PSNR values indicate more noticeable distortion which makes the stego-image easier to detect. PSNR values above 40 dB are excellent (Setiadi, 2021). PSNR from the images of the proposed LSB-AES hybrid algorithm were between 80.65 to 87.14 dB inferring that the images were of good quality and surpassed the minimum threshold of 40 dB.

Higher MSE values indicates more significant differences between the original and stego images, suggesting that the embedding process has introduced noticeable changes while low MSE values indicates that the stego-image is similar to the original image, meaning that the embedding process has introduced minimal distortion. The values of the images used in the proposed LSB-AES hybrid algorithm ranged from 0.0001297 to 0.0005646.

These results infer that the embedding process introduced minimal distortion to the stego-images.

Entropy is used to evaluate the security and quality of the stego image. Higher entropy values indicate better security and less detectable hidden data (Kumar, Pathak & Badal, 2022). For an image to be secure and detectable, entropy should be close to 8 (Hari, Moses, Syaiful, & Atika, 2017). Entropy values from the images used in the proposed hybrid algorithm were between 6.295 to 7.67. These values indicate better security and the embedded data was less detectable.

Histogram analysis was used as a tool for detecting hidden data in an image. It involves comparing histograms of the cover image and the stego-image to detect anomalies that may indicate the presence of hidden data. Robustness and resilience of the proposed LSB-AES algorithm tested the algorithm's resistance to MITM attacks. Simulation experiment conducted six images revealed that the histograms of the cover and stego images were identical and therefore less suspicious to MITM attacks. Finally, a combination of LSB and AES enhances security by encrypting data before embedding it. This makes it difficult for MITM to decrypt embedded messages from the stego an image since a decryption key is required (Talasila, Vijaya, Vijaya, Nainika, Veda, & Mohan, 2024).

### 3.8 Simulation Tests Setup

From the USC-SIPI dataset, six color images, airplane, female, house, couple, peppers, and sailboat were chosen as cover images. The pixels of the airplane, peppers, and sailboat were 512 by 512, whereas the pixels of the female, house, and couple were 256 by 256.

These images were selected for simulation tests from the dataset to assess the quality of image analysis.

Modern steganographic systems are shown to use at least two cover images for test simulations and a minimum of ten for main images (Zhang & Zhao, 2013; Alabaichi, Ali, Al-Dabbas, & Salih, 2020). For this reason, six cover images in 24-bit with TIFF format were used in this investigation. This format was selected because, once messages are embedded, TIFF format images remain identical to the original cover images. Furthermore, TIFF format files are lossless for archiving (Chikouche & Chikouche, 2017).

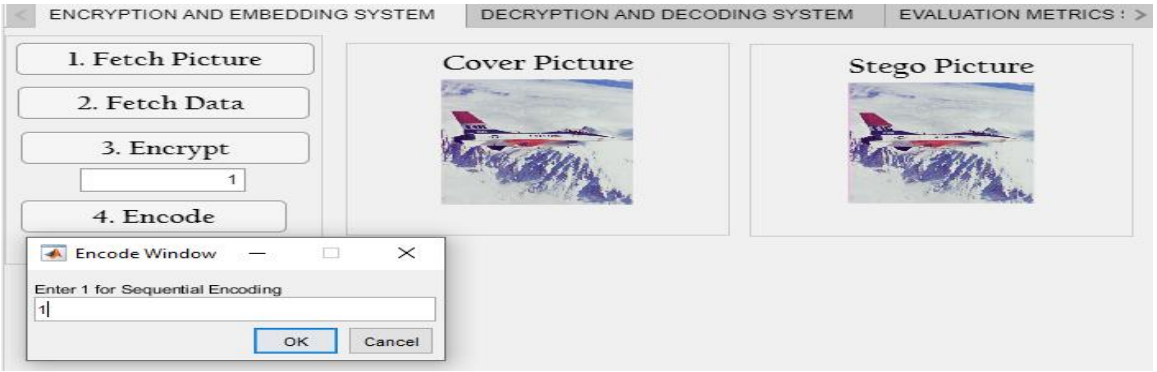
Test one simulation utilized an OTP in which six characters were encrypted and embedded in airplane cover image to yield a stego image. In test simulation two, a password with twenty-nine characters was encrypted and embedded into female cover picture and finally in simulation test three, a sixteen-character credit card number was encrypted and embedded into peppers cover image to produce a stego picture. Simulation tests comprised of two main interfaces; encryption and embedding component and decryption and decoding component. A third interface was used for evaluation metrics of the proposed algorithm.

### 3.8.1 Encryption and Embedding Interface

This component comprises four tabs which are; fetch picture, fetch data, encrypt, encode and an interface. The interface contains User Interface Axes (UIAXes) which are used to create axes for plots within applications. They are useful when developing Graphical User Interface (GUIs) with App designer in MATLAB. UIAXes were used to display the cover and stego-picture as displayed in Figure 3.7.

Figure 3.7

Encryption and Embedding System (Researcher, 2023).



1

Enter 1 for Sequential Encoding

1

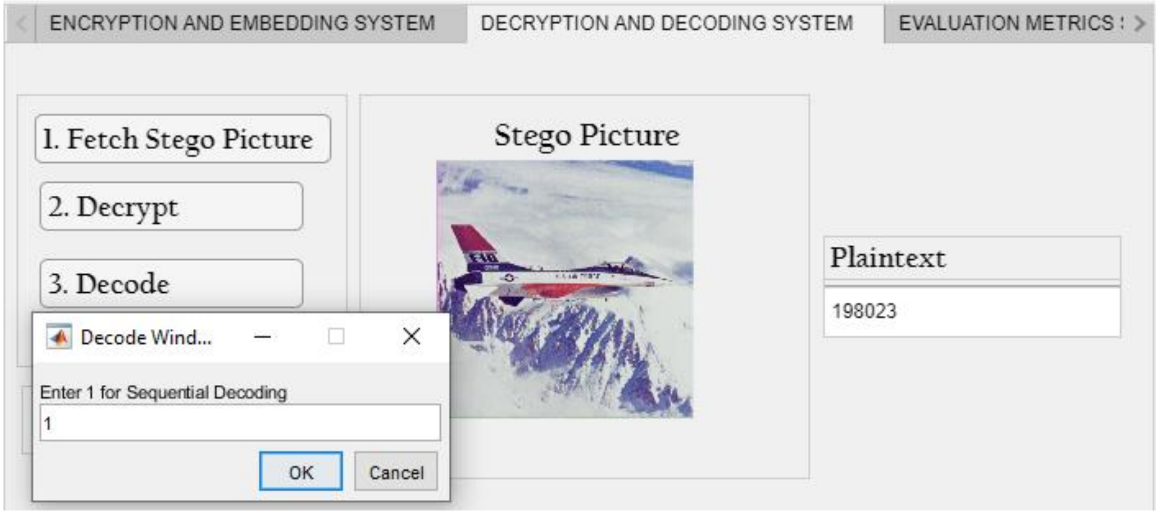
The fetch picture tab is utilized for uploading cover pictures into the system. The fetch data tab is used for uploading data for encryption. The encrypt tab was used to apply AES encryption on the uploaded data and finally encode tab was utilized to sequentially insert encrypted data onto the cover picture to yield a stego picture.

### 3.8.2 Decryption and Decoding Interface

Decryption and decoding component comprise of three tabs; fetch stego picture, decrypt, decode and additional two UIAXes where cover picture and stego picture is displayed. Additionally, this component contained a text edit field where retrieved secret data is displayed after the decryption and decoding process.

Figure 3.8

Decryption and Decoding System (Researcher, 2023).



Enter 1 for Sequential Decoding

1

Plaintext

198023

The fetch stego picture tab is used to upload stego pictures which contain encrypted and embedded confidential data. The AES decryption algorithm key was applied using the decrypt tab to reorganize the ciphertext data and the embedded information from the Stego picture was successively retrieved using the decode tab. The component for decryption and decoding is shown in Figure 3.8.

### 3.9 Results Analysis and Presentation

This study used two approaches to analyze and present findings. To assess the performance of mobile banking applications, threats in mobile banking applications, and state-of-the-art techniques for safeguarding user data in mobile banking, content review was utilized. The results of gap analysis were presented objective wise to determine opportunities for further inquiry.

### 3.10 Evaluation of the Proposed LSB-AES Hybrid Algorithm

Analysis of the proposed algorithm was done using visual analysis, entropy analysis, and statistical analysis. The effectiveness of a technique is demonstrated by the results of the analyzed scheme using various methods when their values fall within anticipated threshold. Results from the three analyses were displayed in tables and figures. The following subsection reports on how the proposed LSB-AES hybrid algorithm was evaluated.

### 3.10.1 Visual Quality Analysis

Hiding information inside an image's pixels is a steganographic technique which modifies an image's appearance such that the changes made to a picture should not be visible to adversaries. The strength of the image quality can be verified using a variety of common evaluation techniques, such as Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSI) and Normalized Absolute Error (NAE) (Jan, Parah, Hussan, & Malik, 2022).

MSE and PSNR were chosen for the study because they are the best metrics that can be utilized for secret writing, encipherment, data interpretation, and determination (El-Abbadi, Al-Zubaidi, & Razzaq, 2020). Additionally, these two metrics were used because they were crucial components in the proposed algorithms' analysis and evaluation. Results from the metrics were displayed in figures and tables.

#### 3.10.1.1 Mean Squared Error

The average squared error between two images is measured by Mean Squared Error (MSE). The dissimilarity between the original and modified pictures is linked by their averaged squared value.  It therefore expresses the mean flaw between original and stego picture (GU, He, Liu, & Ye, 2021). (3.1) demonstrates how MSE is calculated.

$$MSE = \frac{\sum M, N\left[l_1(m,n) - l_2(m,n)\right]^2}{M \times N} \tag{3.1}$$

The letters M and N stands for measurements of two images and *$l_1$* and *$l_2$* is the cover and stego pictures respectively. MSE is a full reference metric in which values closer to zero are better (Sara, Akter, & Uddin, 2019). Results from the proposed algorithm yielded low MSE values ranging from 0.0001297 to 0.0005646. These results infer that an adversary cannot be able to distinguish between a cover picture and a stego picture.

### 3.10.1.2 Peak Signal-to-Noise Ratio

PSNR measures the ratio of a signal's maximum potential power to the corrupting noise that degrades the quality of the image that is displayed. The greatest image quality is indicated by PSNR values of more than 40 decibels (dB) (Lakshmi, Srinives, Kumar, & Chandra, 2016). According to (Sukumar, Subramaniyaswamy, Vijayakumar, & Ravi, 2020), the PSNR value should be greater than 39 (dB) for improved image quality. (3.2) demonstrates how PSNR is calculated.

$$PSNR = 10Log_{10}\left(\frac{R^2}{MSE}\right) \tag{3.2}$$

*Where $R^2$* is highest number of pixels and MSE is the mean squared error of cover and stego image. Results from the proposed algorithm yielded higher PSNR values ranging from 80.65 to 87.14 dB. These results infer higher PSNR values indicate reconstructed stego picture had high standards.

### 3.10.1.3 Entropy Analysis

In this study, entropy was used to determine how secure the proposed algorithm was. An image's entropy can be used as a statistic to quantify its information content. Finding the average bit count required to encode message components is useful. Entropy levels ought to be near the value 8 (Hari, Moses, Syaiful, & Atika, 2017). (3.3) demonstrates how to calculate entropy.

$$Entropy = \sum{}^{n}{}_{i} = 1\pi Log_2\left(\pi\right) \qquad (3.3)$$

*where* the likelihood that the data value i will occur, and n is the number of distinct data values. According to Sneha, Sankar, and Kumar (2020), the optimal entropy value for an 8-bit system is close to 8, with values outside of this range often falling between 0 and 8. Results from the developed algorithm yielded entropy values ranging from 6.295 to 7.762. These values indicate that the proposed algorithm has good entropy capacity.

### 3.10.2 Statistical Analysis

An image’s resistance to assaults is evaluated using statistical analysis. For determining a system’s resistance to assaults, statistical tests such as histogram analysis and correlation coefficient are frequently utilized. Histogram analysis was used to analyze robustness of the developed algorithm against steganalysis attacks. Furthermore, histogram analysis is the most widely used technique that evaluates an encrypted pictures’ resistance to attacks (Jan, Parah, Hussan, & Malik, 2022).

#### 3.10.2.1 Histograms Analysis

A histogram represents regularity with which a pixel comes about graphically. Alteration of picture elements is impacted during insertion of secret message on cover picture. Steganography can be identified using these changes. An element of every one of the RGB color model describes a pixel and these element values reflect intensities of the RGB pixel model (Jan, Parah, Hussan, & Malik, 2022). Histograms are used to analyze two pictures (cover and stego) such that if no discernible change exists in the histogram of the two different pictures, then adversaries cannot obtain any clue about the data embedded in the stego-picture.

To prevent statistical attacks, histogram analysis of stego picture should match histogram of cover picture. An encrypted picture histogram, on the other hand, should be consistent to demonstrate its unpredictable behavior. The robustness of the encryption technique is claimed by the uniformity of the histogram among pixels. To guard against statistical attacks, the histogram of an encrypted picture should be consistent (Jan, Parah, Hussan, & Malik, 2022).

The histograms generated from the cover and stego pictures are displayed in Table 4.10. Findings reveal that the cover and stego pictures have little distortion based on the six simulation experiments. The histograms' RGB color distribution shows that there is no difference between the images. This indicates that the proposed algorithm is robust against statistical attacks.

## 3.11 Ethical Considerations

Researcher acquired introductory letter from Kisii University, and utilized it to apply for authority license for research from relevant authorities. As part of desktop research, the researcher downloaded peer reviewed academic publications including books, conference proceedings and journal articles. Publications were collected from numerous academic research databases, including ACM, Emerald, IEEE, Springer and Google Scholar. Scholarly interests were served by the utilization of these resources, and the research community will be informed of the study's conclusions.

The USC-SIPI image database's image processing dataset was used in this study which is accessible and free. The researcher did not violate the database's privacy in any manner by downloading and using cover images from the collection, hence there was no harm done to the database. Only for study purposes were the database's photographs used. Finally, the research community will be given access to simulation test findings for the used images.

# CHAPTER FOUR

# IMPLEMENTATION AND RESULTS

## 4.1 Overview

In this chapter, the following are presented and discussed; development of LSE-AES On-transit user-data protection algorithm, simulation tests and evaluation of LSB-AES on-transit user-data protection algorithm. Additionally, simulation tests, performance metrics, and steganalysis of the developed algorithm have been discussed.

## 4.2 Development of LSB-AES On-Transit User-Data Protection Algorithm

LSB-AES hybrid algorithm was developed by merging two established algorithms LSB and AES encryption algorithm. LSB substitution technique contributes a security element of embedding messages in cover images while AES algorithm encrypts the secret messages and makes it difficult for adversaries to decode the encrypted message without a decoding key. It has nine steps as demonstrated in Table 4.1.

Table 4.1

LSB-AES On-Transit User-Data Protection Algorithm (Researcher, 2023)

| LSB-AES On-Transit User-Data Protection Algorithm | |
|---|---|
| Step 1: | Read the Input Cover Image |
| Step 2: | Read the Confidential Information |
| Step 3: | Apply AES Encryption |
| Step 4: | Apply LSB Encoding |
| Step 5: | Save the Stego-Image |
| Step 6: | Read the Stego-Image for Decoding |
| Step 7: | Apply LSB Decoding |
| Step 8: | Apply AES Decryption |
| Step 9: | Save or Display the Decrypted Data |

The LSB-AES on-transit user-data protection algorithm starts with loading the cover image that is used to hide confidential information. This step is followed by loading data that need to be hidden in the cover image. For this study, the data that is used is text. AES is then applied to encrypt confidential data to secure it before embedding. LSB embedding is then applied to embed encrypted data into the LSB bits of the cover image. This is followed by saving the image that contains embedded data. The stego-image is read to the system in order to start the process of extracting hidden data. LSB decoding is applied in order to extract encrypted data from the LSB of the stego-image. AES decryption is then applied in order to decrypt extracted data to retrieve the original confidential information. Lastly, the decrypted data can be saved or displayed as needed. Figure 4.1 demonstrates the steps involved in embedding and decrypting confidential messages using LSB-AES on-transit user-data protection algorithm.

Figure 4.1

LSB-AES On-Transit User-Data Protection Algorithm (Researcher, 2023)

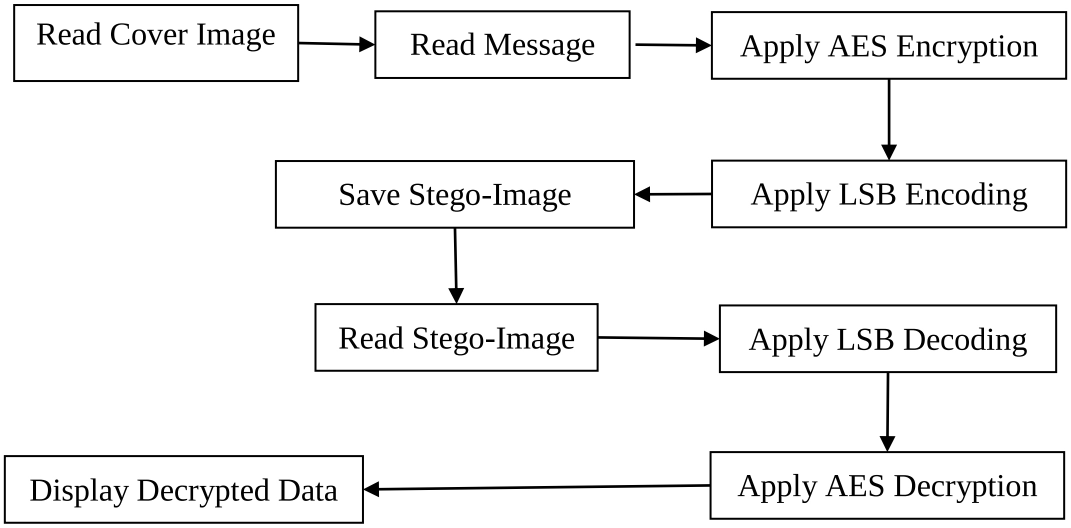
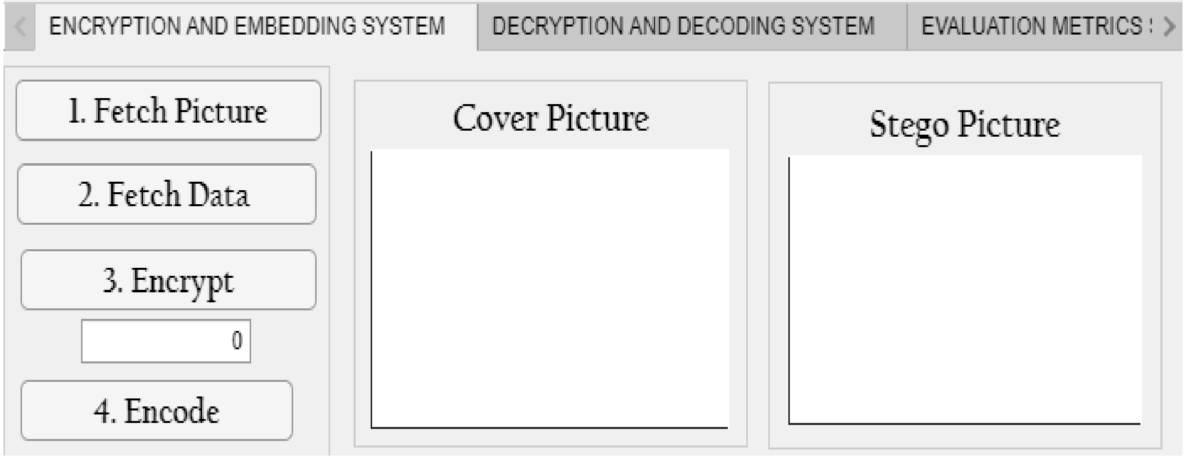


Figure 4.1 shows a block diagram of the steps used to develop LSB-AES on-transit user-data protection algorithm. The algorithm was implemented using MATLAB with three components. The first component with the title ENCRYPTION AND EMBEDDING SYSTEM is responsible for fetching cover images, fetching confidential data, applying encryption key using AES and embedding encrypted data into a cover image using LSB substitution technique. This component was designed with the following tabs: Fetch Picture, Fetch Data, Encrypt, and Encode tabs. This system also contains two UIAXes where the cover and stego pictures are displayed upon provoked by respective callback as well as an edit textfield where encryption key can be displayed. Figure 4.2 displays encryption and embedding system.

Figure 4.2

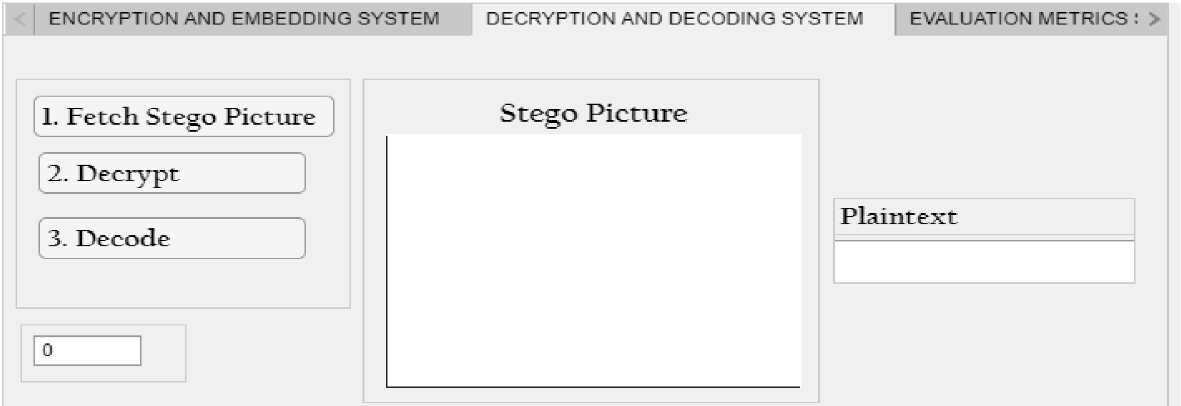
Encryption and Embedding System (Researcher, 2023).



The second component with the title DECRYPTION AND DECODING SYSTEM is responsible for fetching the stego image from the computer, and decrypting ciphertext to recover plaintext message. This component was designed with the following three tabs in mind: Fetch Stego Image, Decrypt and Decode with UIAXes where a stego picture can be displayed. Figure 4.3 illustrates decryption and decoding system.

Figure 4.3

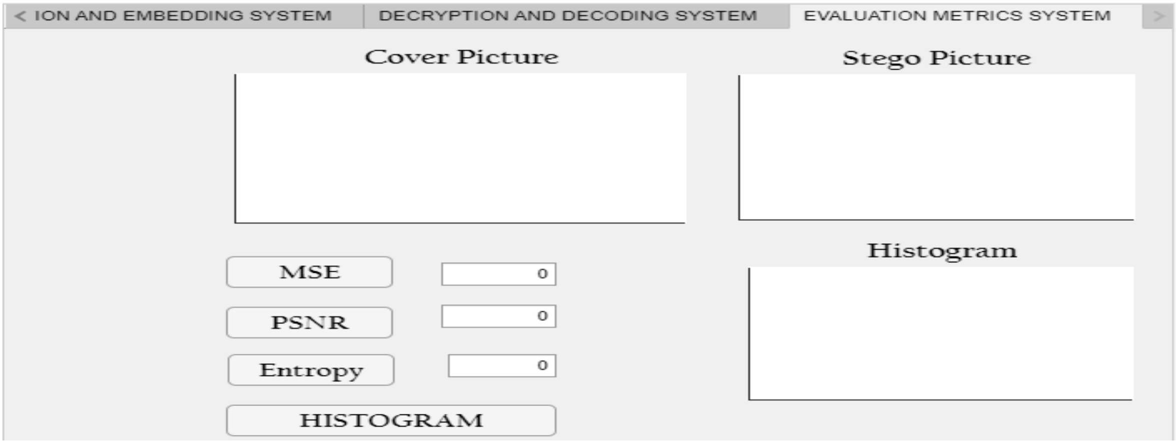
Decryption and Decoding System (Researcher, 2023).



The third component with the title EVALUATION METRICS SYSTEM is responsible for calculating and displaying MSE, PSNR, Entropy and Histograms upon provoked by the respective tabs. This section also contains three UIAXes, one for cover picture, the second for stego picture and the third for histograms. Finally, the component contains three edit textfields where MSE, PSNR and Entropy can be displayed. Figure 4.4 displays the evaluation metrics component.

Figure 4.4

Evaluation Metrics (Researcher, 2023).


< ION AND EMBEDDING SYSTEM
DECRYPTION AND DECODING SYSTEM
EVALUATION METRICS SYSTEM
>
Cover Picture
Stego Picture
Histogram
HISTOGRAM

MSE 0

PSNR 0

Entropy 0

LSB substitution technique and AES algorithms were utilized in three simulation tests to hide three different private messages inside three cover images. Three secret message samples were encrypted and embedded in the stego images in the encryption and embedding system and later on decrypted to recover plaintext in the decryption and decoding system. The message samples were: OPT for simulation test 1, password sample for simulation test 2, and credit card number for simulation test 3.

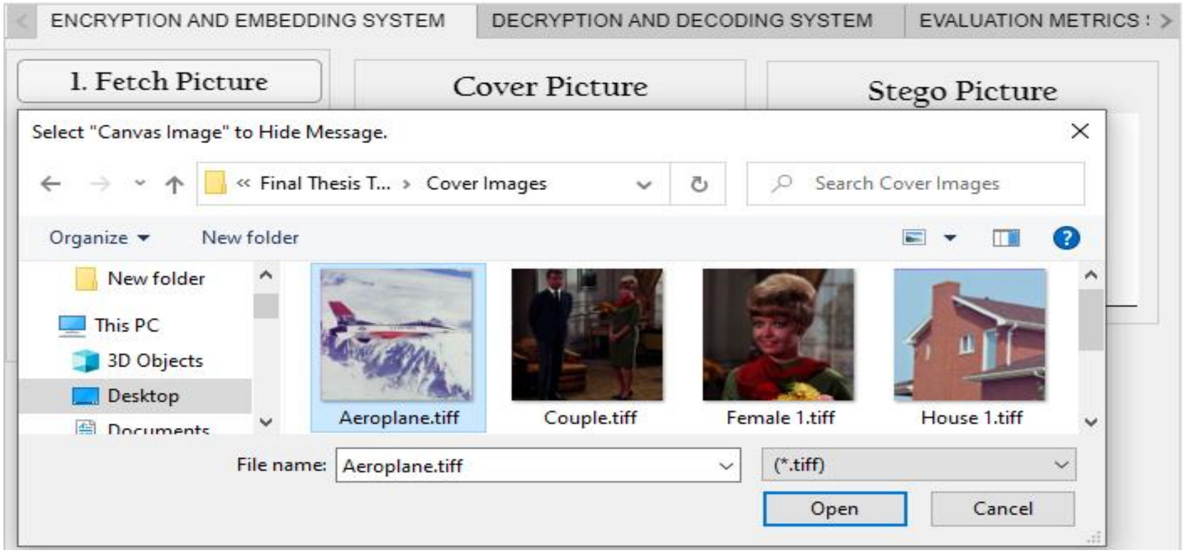
## 4.3 Implementation

The developed LSB-AES on user data protection algorithm was implemented using MATLAB, which is popularly used for developing and implementing algorithms and simulations. Implementation of the developed algorithm has three simulation tests. In simulation test one, an OTP test sample with six digits which are 198023 was used for test experiment one. An OTP is random string of letters or numbers utilized just once. OTPs reduces the possibility of unauthorized login attempts and serves as a second layer of authentication and thus protects mobile banking applications from online crimes like social engineering and MITM. OTPs, however, are sent in plain text raising the possibility that a third party would intercept it and access one’s bank account and make a withdrawal.

In this simulation test, the OTP is encrypted with AES encryption technique and embedded onto a cover picture to increase mobile banking security levels and prevent it from being intercepted by an attacker. The fetch picture tab is manifested in Figure 4.5.

Figure 4.5

Fetch Picture Tab (Researcher, 2023).



Select "Canvas Image" to Hide Message.

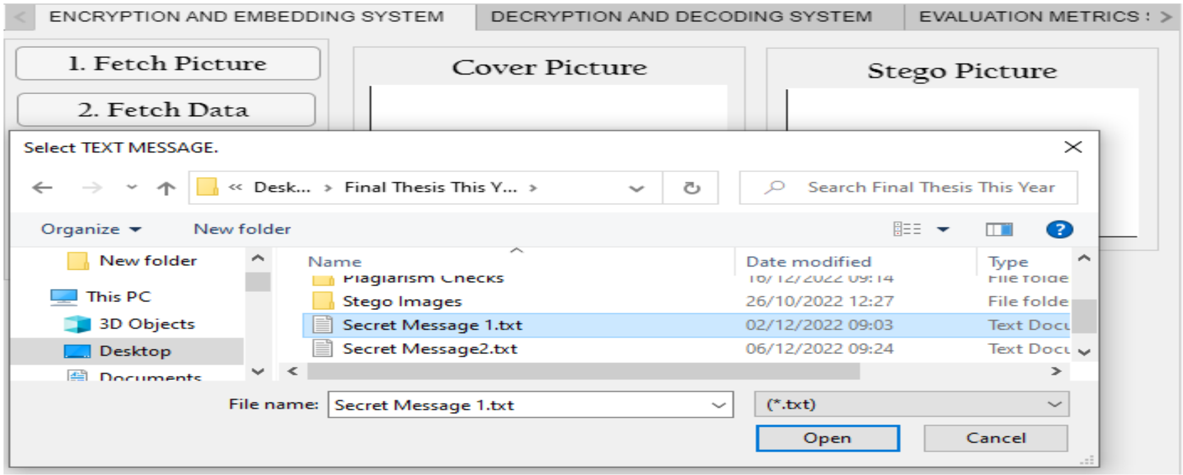
File name: Aeroplane.tiff

(*.tiff)

In Simulation test 1, the system's fetch picture tab is used to upload an image into the system. Then, an OTP is uploaded onto the system via fetch data tab. The OTP is first encrypted using AES on the encryption key tab, and then embedded onto the cover image using the encode tab to create a stego-image.

Figure 4.6

Fetch Data Tab (Researcher, 2023).



Select TEXT MESSAGE.

« Desk... › Final Thesis This Y... ›

Search Final Thesis This Year

| Name | Date modified | Type |
|---|---|---|
| Plagiarism Checks | 16/12/2022 09:14 | File folde |
| Stego Images | 26/10/2022 12:27 | File folde |
| Secret Message 1.txt | 02/12/2022 09:03 | Text Docu |
| Secret Message2.txt | 06/12/2022 09:24 | Text Docu |

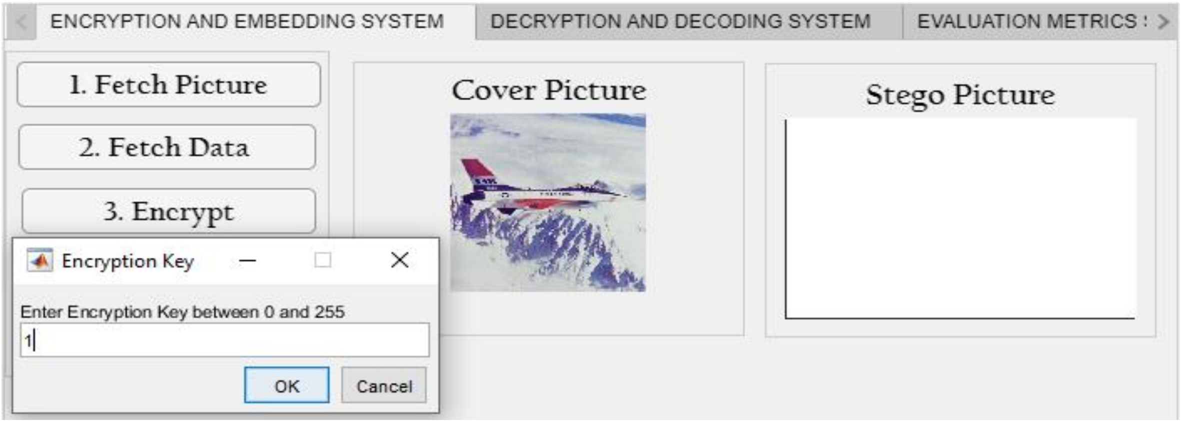
File name: Secret Message 1.txt

(*.txt)

The fetch data tab is used to locate secret messages on the computer and uploads them into the system. In this simulation, the system's fetch data tab is utilized to upload an OTP sample with six characters. The user uploads secret messages from any location on the computer utilizing the open dialog form as shown in Figure 4.6.

Figure 4.7

Encrypt Tab (Researcher, 2023).



Encryption Key

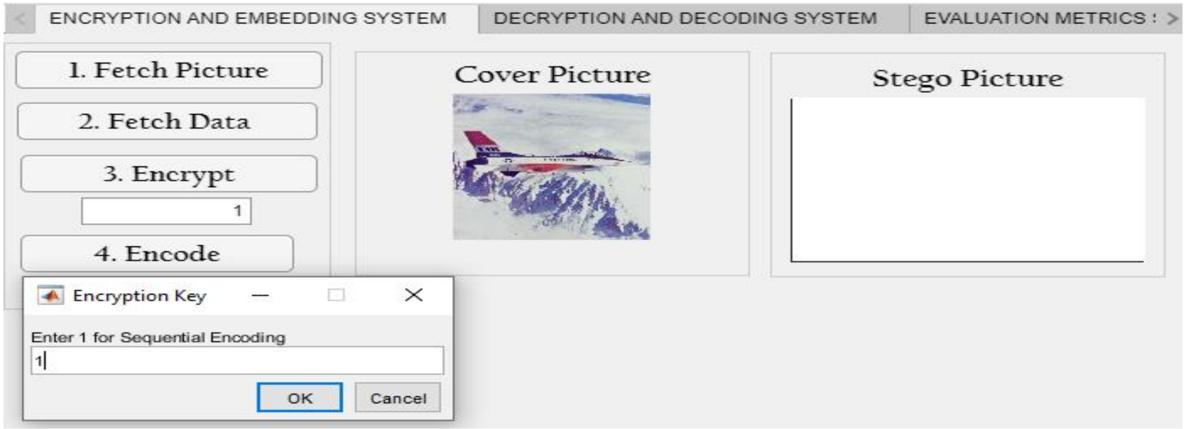
Enter Encryption Key between 0 and 255

1

The function of AES encryption is to transform the OTP that has already been uploaded into the system to ciphertext using a key. The same key that is used in this step is the same key that is used during decryption process. Encryption tab is exhibited in Figure 4.7 and encode tab in Figure 4.8.

Figure 4.8

Encode Tab (Researcher, 2023)



1

Encryption Key

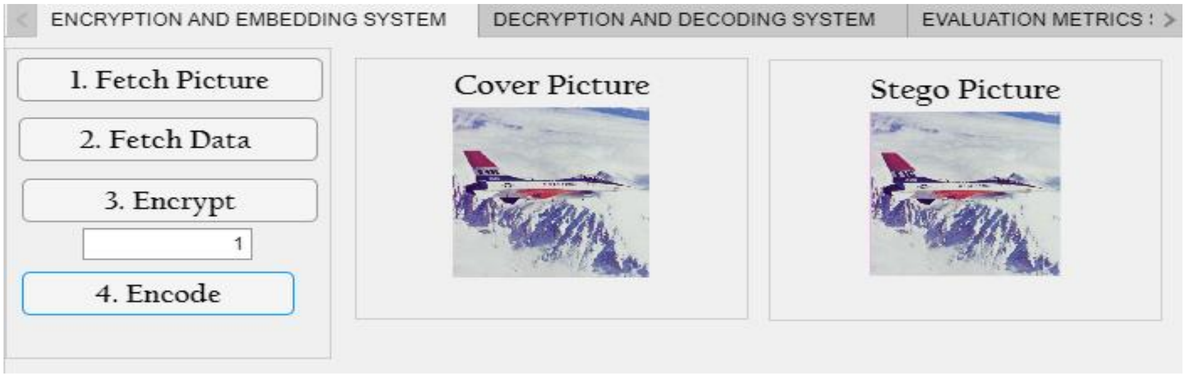
Enter 1 for Sequential Encoding

1

In this simulation test, sequential encoding is used to pick image pixels and embed the message sequentially.

Figure 4.9

Cover and Stego Picture Display Interface (Researcher, 2023).

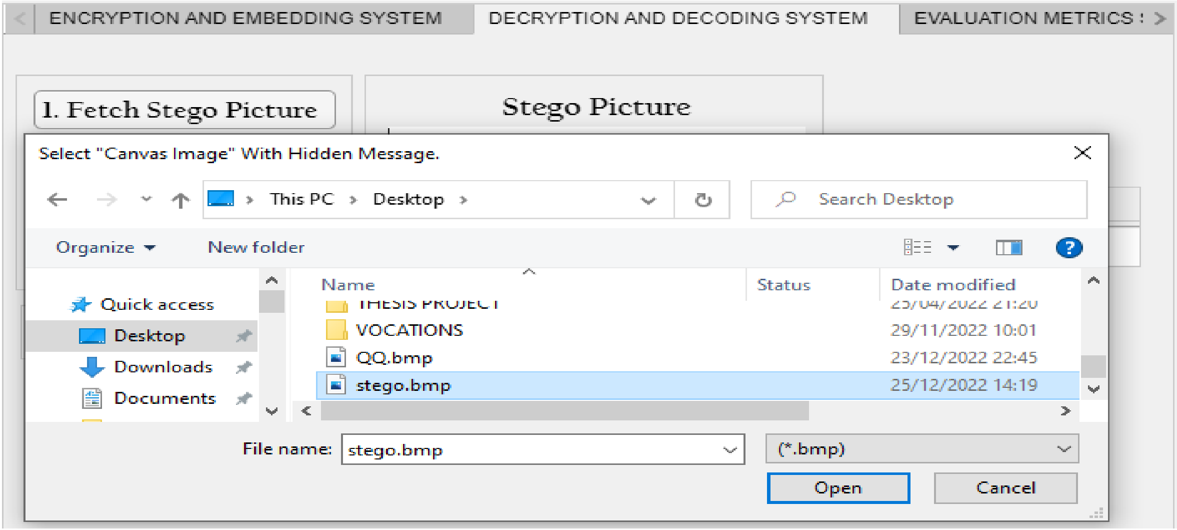


1

Simulation test 1 on the encryption and embedding system was successfully implemented and stego-image saved successfully. The next component is decryption and decoding phase in which three tabs are utilized. Figure 4.10 exhibits the fetch stego-image tab.

Figure 4.10

Fetch Stego Picture Tab (Researcher, 2023).



Select "Canvas Image" With Hidden Message.

| Name | Status | Date modified |
|---|---|---|
| THESIS PROJECT | | 25/04/2022 21:20 |
| VOCATIONS | | 29/11/2022 10:01 |
| QQ.bmp | | 23/12/2022 22:45 |
| stego.bmp | | 25/12/2022 14:19 |

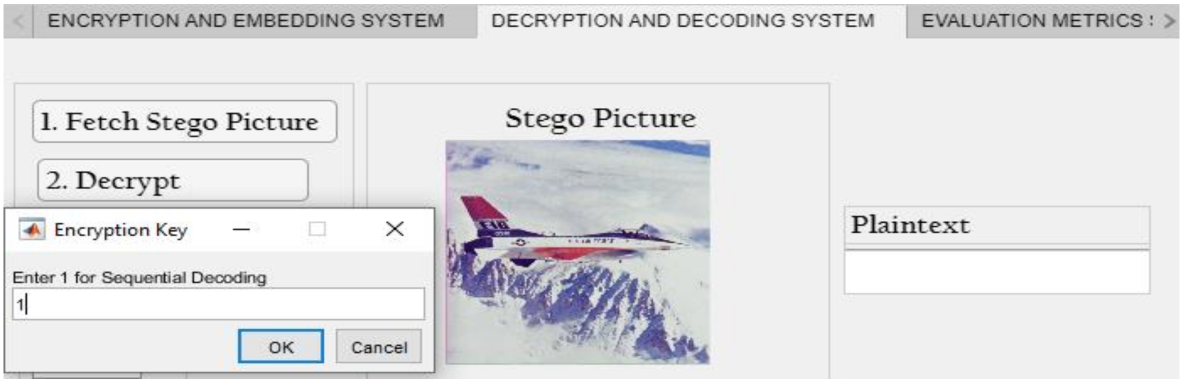
File name: stego.bmp

(*.bmp)

In this simulation, the system's fetch stego picture tab is used to upload a stego-image from the computer. Once the stego-image is uploaded successfully, it is exhibited in the respective UIAXes. Figure 4.11 puts on view an aeroplane stego-image.

Figure 4.11

Decrypt Tab (Researcher, 2023).



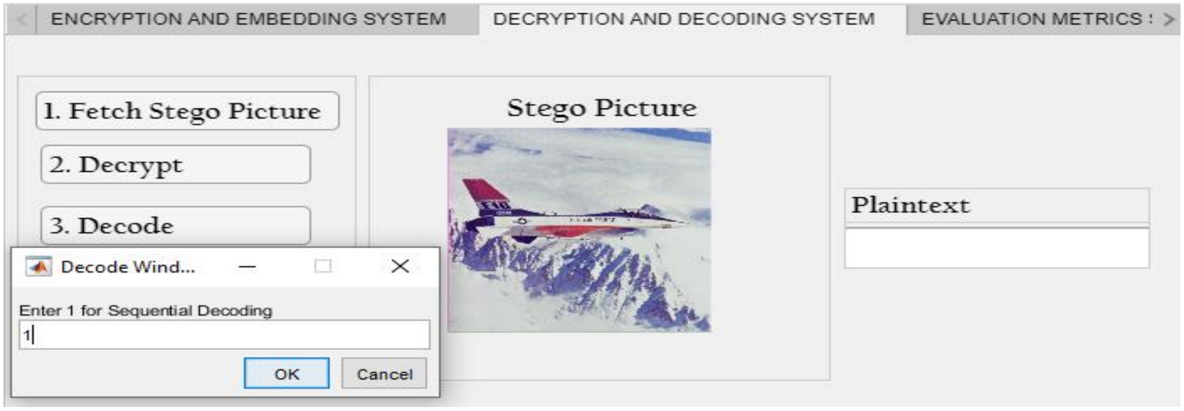
Enter 1 for Sequential Decoding

1

After successfully fetching a stego-image, the OTP is restructured from ciphertext to plaintext. The same key that is used in this step corresponds to one used during encryption process. At this stage, the stego-image is already fetched and presented on UIAXes. Figure 4.12 depicts decoding tab.

Figure 4.12

Decode Tab (Researcher, 2023).



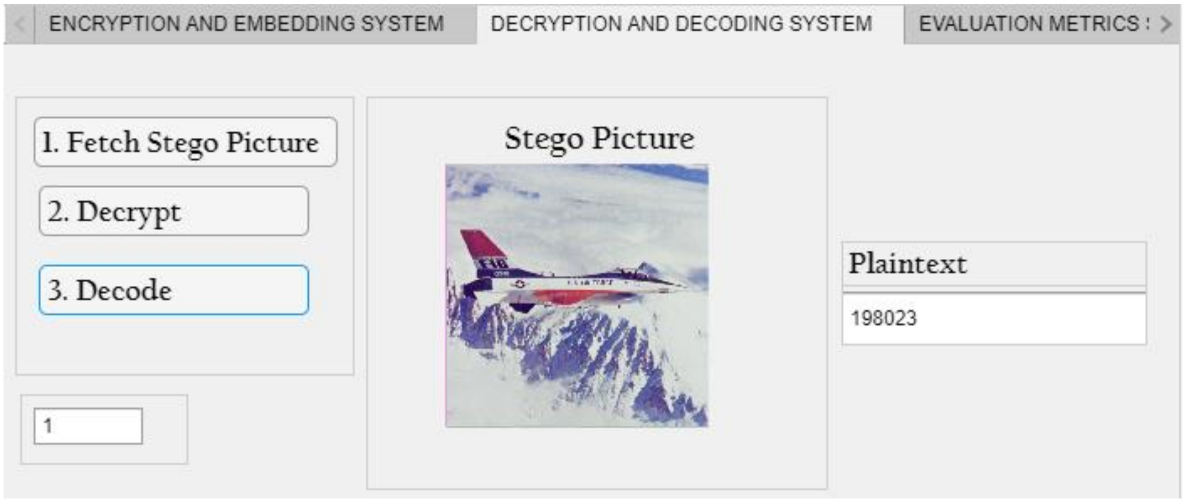
Enter 1 for Sequential Decoding

1

This tab provokes a popup window in which a user can enter a decoding style. Decoding can be sequential or random. In this simulation test, sequential decoding is used in which a key is used to pick picture pixels and decode embedded message bits sequentially. By employing LSB substitution approach, OTP bits are decoded and displayed on an edit textfield. Figure 4.13 exhibits successfully decoded OTP in plaintext form.

Figure 4.13

OTP Plaintext Display (Researcher, 2023).



1

Plaintext

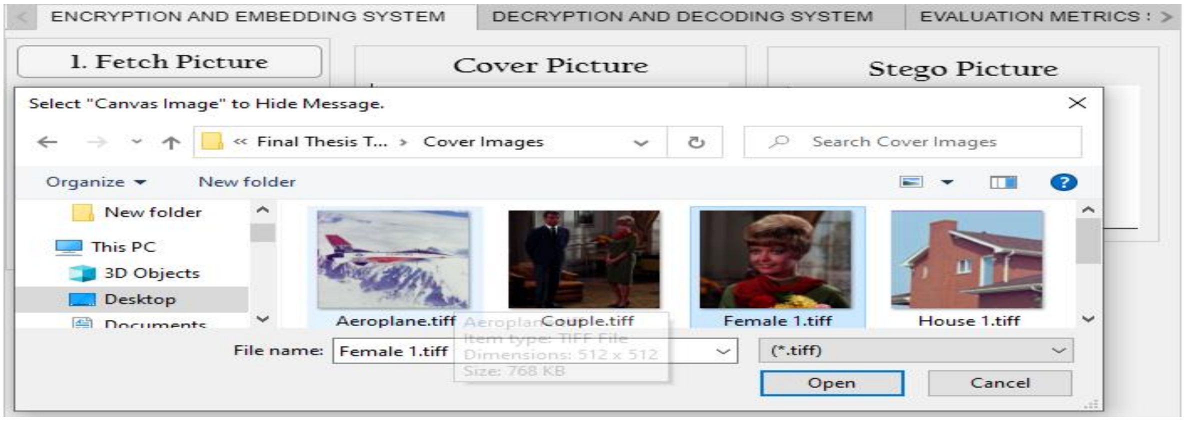
198023

Test simulation 1 one was successfully implemented by encrypting and embedding OTP into a cover image and decrypting and decoding OTP securely on the developed system.

In test simulation two, a sample password (ArtificialIntelligence@#$2029) was used for the test experiment. In this simulation test, a password was encrypted with AES method and embedded onto a female cover-image to increase mobile banking security and prevent it from being intercepted by MITM. The fetch picture tab is manifested in Figure 4.14.

Figure 4.14

Fetch Picture Tab (Researcher, 2023).



Select "Canvas Image" to Hide Message.

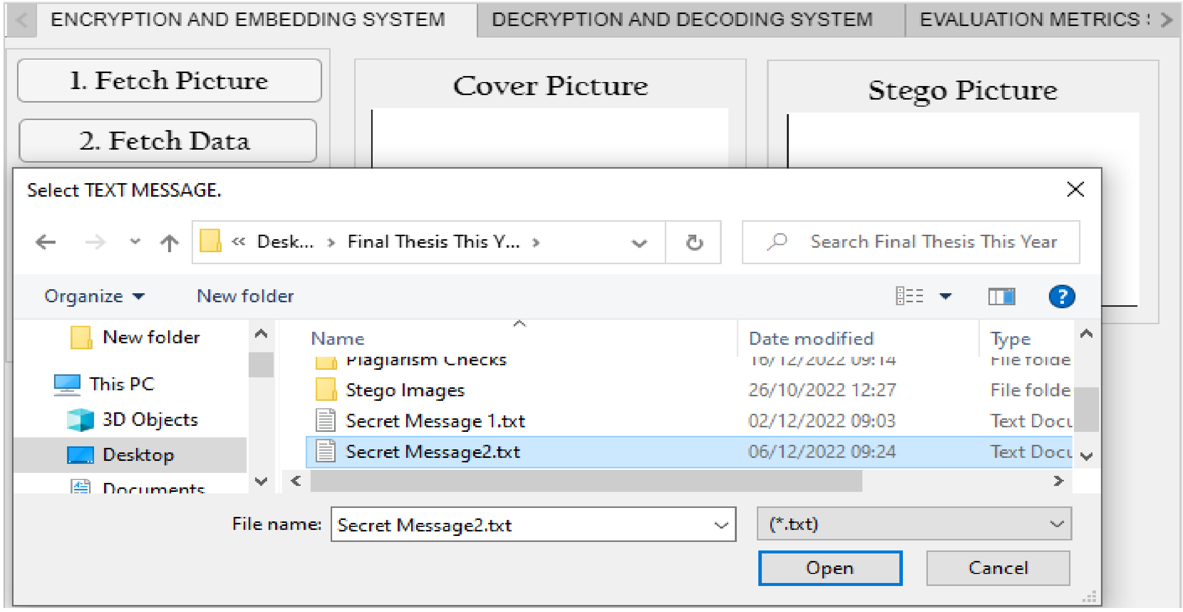
File name: Female 1.tiff

(*.tiff)

In Simulation test 2, the system's fetch picture tab is used to upload an RGB picture called female. Then, a password is uploaded onto the system via fetch data tab. The user uploads cover-images from any location on the computer utilizing the open dialog form as exhibited in Figure 4.15.

Figure 4.15

Fetch Data Tab (Researcher, 2023).



Select TEXT MESSAGE.

| Name | Date modified | Type |
|---|---|---|
| Plagiarism Checks | 10/12/2022 09:14 | File folde |
| Stego Images | 26/10/2022 12:27 | File folde |
| Secret Message 1.txt | 02/12/2022 09:03 | Text Docu |
| Secret Message2.txt | 06/12/2022 09:24 | Text Docu |

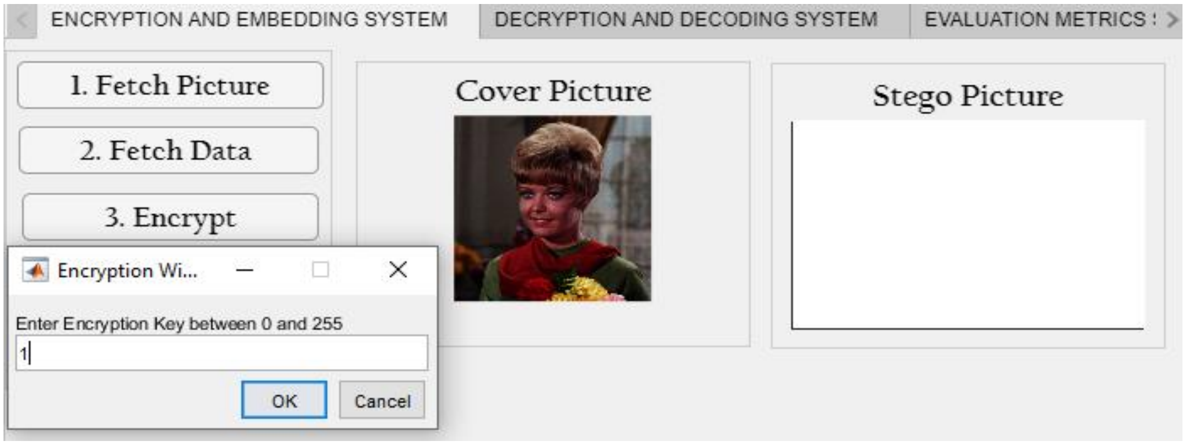
File name: Secret Message2.txt

(*.txt)

In this simulation, the system's fetch data tab is utilized to upload a password sample with twenty-nine characters. After successfully uploading the password, AES encryption is applied to convert it to ciphertext before it is embedded into the cover-image. The user uploads secret messages from any location on the computer utilizing the open dialog form as shown in Figure 4.16.

Figure 4.16

Encrypt Tab (Researcher, 2023).



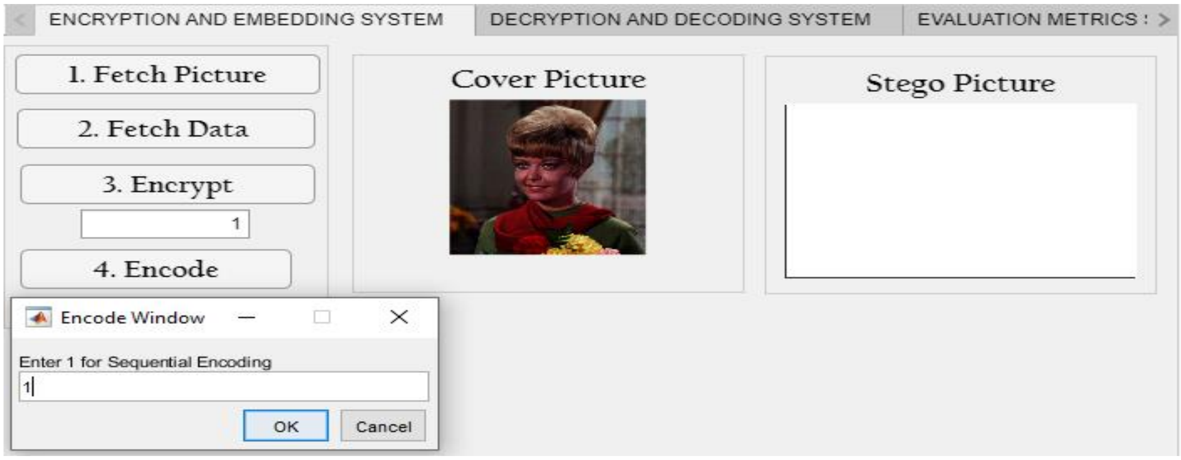
Enter Encryption Key between 0 and 255

1

After successfully fetching data using fetch data tab, the function of AES encryption is to transform the password to ciphertext. At this stage, the cover-image is already fetched and presented on UIAXes as shown in Figure 4.16.

Figure 4.17

Encode Tab (Researcher, 2023).



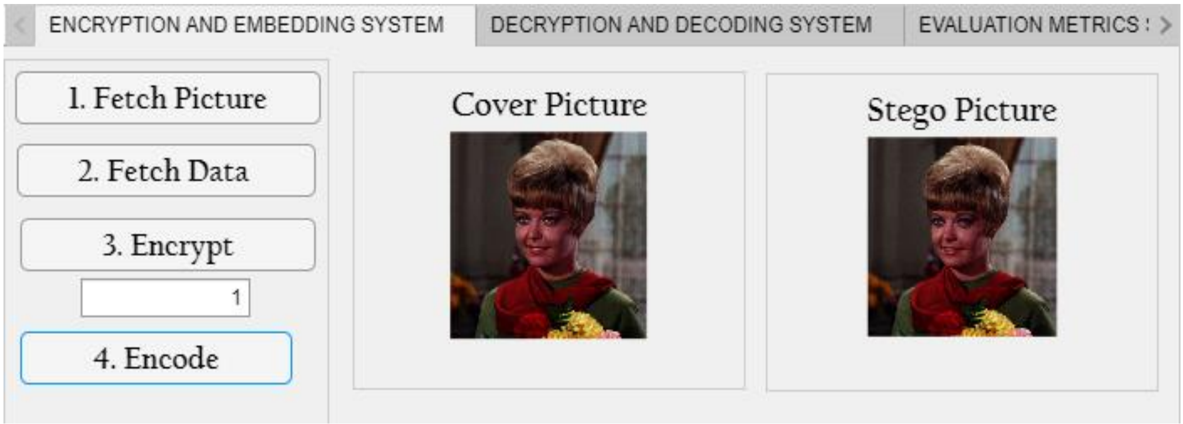
Enter 1 for Sequential Encoding

1

This tab provokes a popup window in which a user can enter an encoding style. This simulation test utilized sequential encoding to embed the message sequentially. By employing LSB substitution approach, password bits are encoded onto the cover-image as shown in Figure 4.17.

Figure 4.18

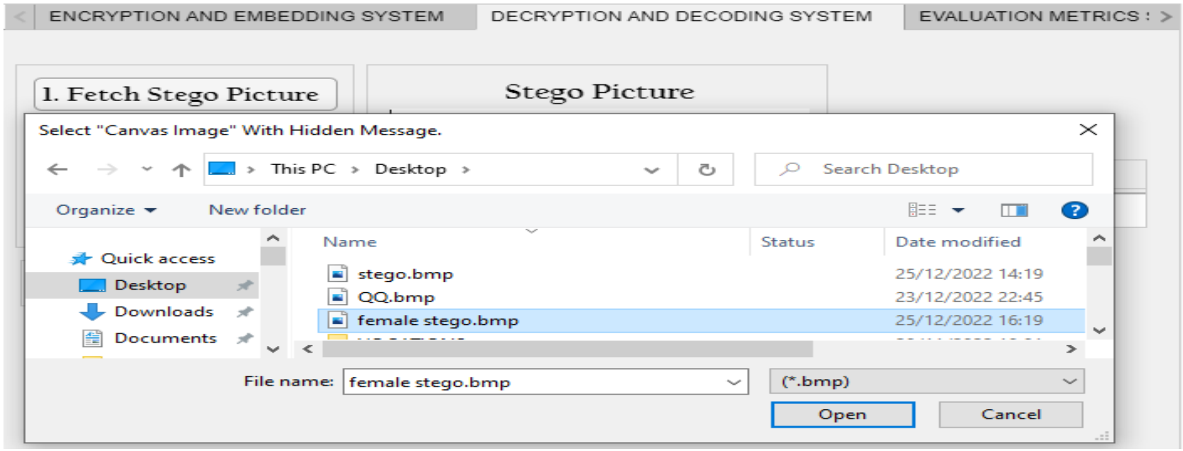
Cover and Stego Picture Display Interface (Researcher, 2023).



The cover picture and stego picture are exhibited in their respective UIAXes IN Figure 4.18. The stego-image is then saved in bmp format. Simulation test two on the encryption and embedding system was successfully implemented and stego-image saved successfully. The next phase is decryption and decoding phase in which three tabs are utilized. Figure 4.19 exhibits the fetch stego picture tab.

Figure 4.19

Fetch Stego Picture Tab (Researcher, 2023).



Select "Canvas Image" With Hidden Message.

| Name | Status | Date modified |
|---|---|---|
| stego.bmp | | 25/12/2022 14:19 |
| QQ.bmp | | 23/12/2022 22:45 |
| female stego.bmp | | 25/12/2022 16:19 |

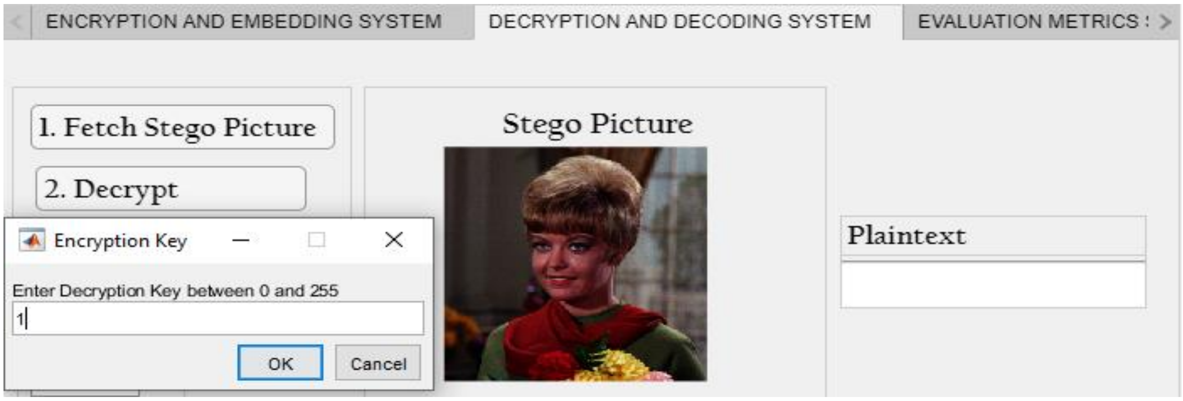
File name: female stego.bmp

(*.bmp)

In simulation test two, the system's fetch stego picture tab is utilized to upload a stego-image of female with bmp format. Then, decrypt tab is used to enter a key that is used to restructure the password. Figure 4.20 puts on view a female stego image.

Figure 4.20

Decrypt Tab (Researcher, 2023).



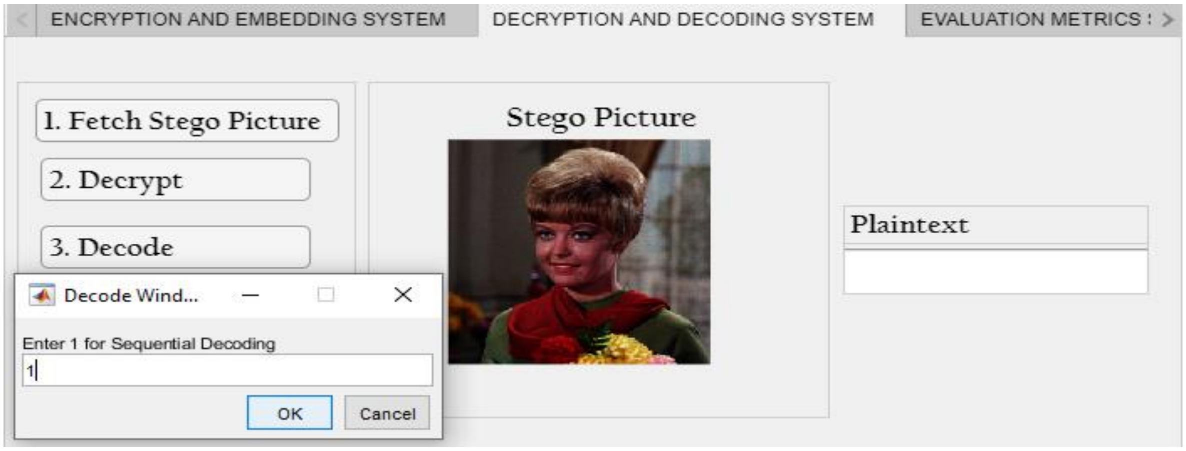
Enter Decryption Key between 0 and 255

1

After successfully fetching the stego-image, the password is restructured from ciphertext to plaintext. The same key that is used in this step corresponds to one used during encryption process. Figure 4.20 depicts the decrypt tab.

Figure 4.21

Decode Tab (Researcher, 2023).



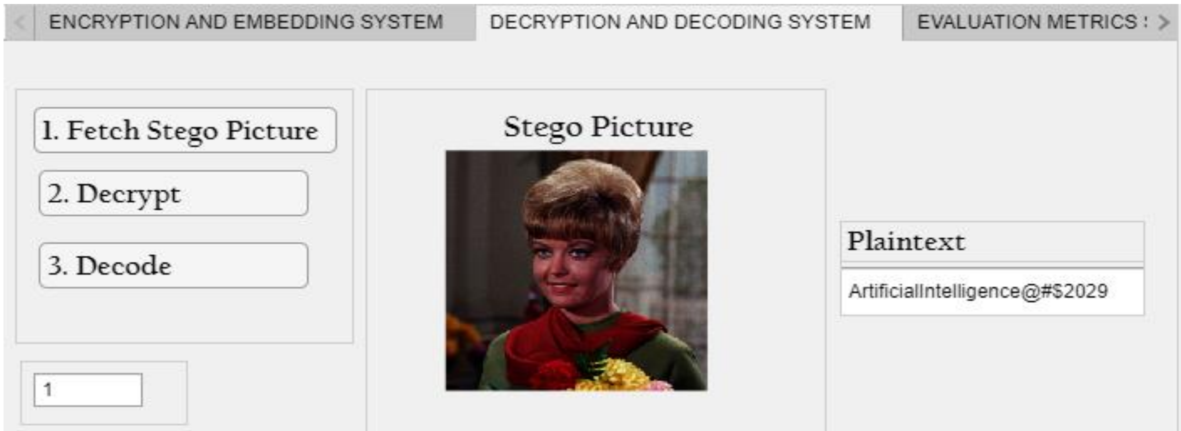
Enter 1 for Sequential Decoding

1

In this simulation test, sequential decoding is used to decode embedded message bits sequentially. By employing the LSB substitution approach, password bits are decoded and displayed on an edit textfield. Figure 4.22 exhibits successfully decoded password in plaintext form.

Figure 4.22

Plaintext Display of a Password (Researcher, 2023).



1

Plaintext

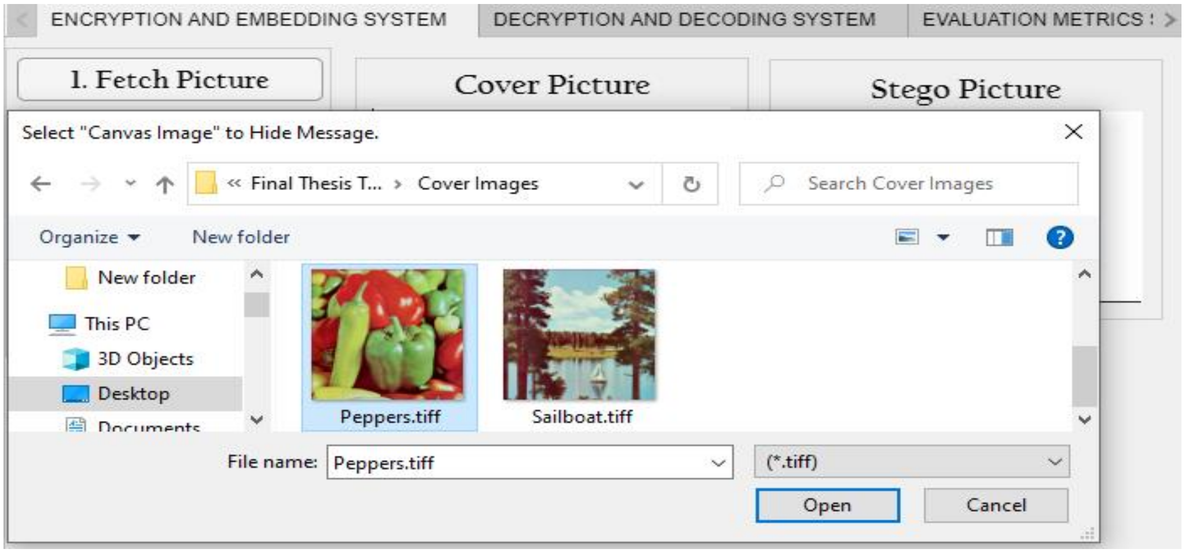
ArtificialIntelligence@#$2029

The decoded plaintext password is used to login to mobile banking applications in order to access financial services remotely. Test simulation two was successfully implemented by encrypting and embedding password onto a cover picture and decrypting and decoding password securely on the developed system.

In test simulation three, a sample of credit card number with sixteen characters (1234 4321 5678 8765) is used. A credit card number is string of numbers utilized when doing online shopping. If a credit card gets lost, it can be used to conduct illegal transactions such as withdrawal of money and paying bills. In this simulation test, the credit card number was encrypted with AES method and pasted onto a cover picture to increase mobile banking security levels and prevent it from being used by unauthorized persons. The fetch picture tab is shown in Figure 4.23.

Figure 4.23

Fetch Picture Tab (Researcher, 2023).



Select "Canvas Image" to Hide Message.

« Final Thesis T... › Cover Images

Search Cover Images

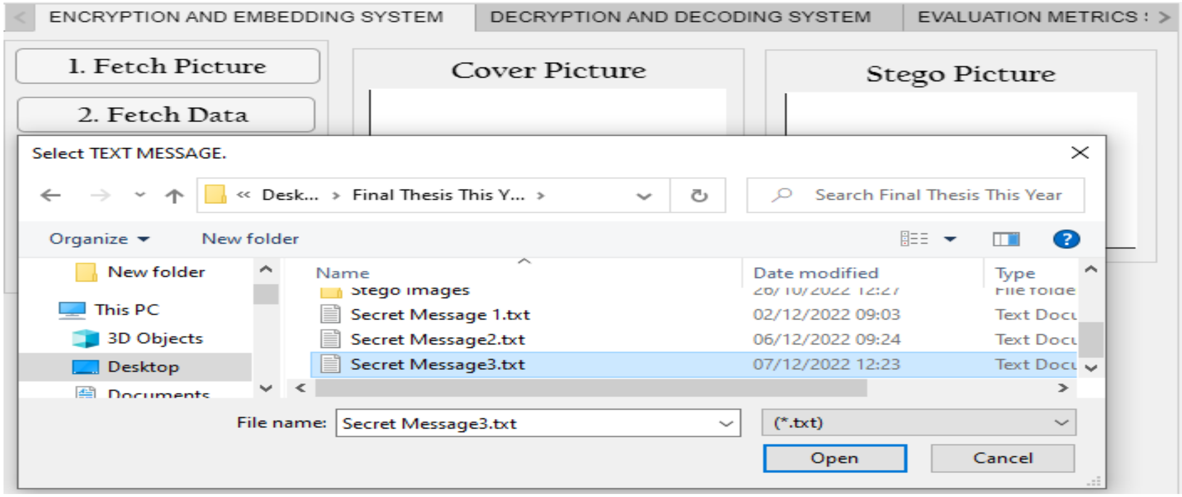
File name: Peppers.tiff

(*.tiff)

The system's fetch picture tab is utilized to upload RGB image called peppers. Then, a credit card number is uploaded onto the system via fetch data tab. Figure 4.23 reveals fetch data tab.

Figure 4.24

Fetch Data Tab (Researcher, 2023).



Select TEXT MESSAGE.

« Desk... › Final Thesis This Y... ›

Search Final Thesis This Year

| Name | Date modified | Type |
|---|---|---|
| Stego Images | 20/10/2022 12:27 | File folde |
| Secret Message 1.txt | 02/12/2022 09:03 | Text Docu |
| Secret Message2.txt | 06/12/2022 09:24 | Text Docu |
| Secret Message3.txt | 07/12/2022 12:23 | Text Docu |

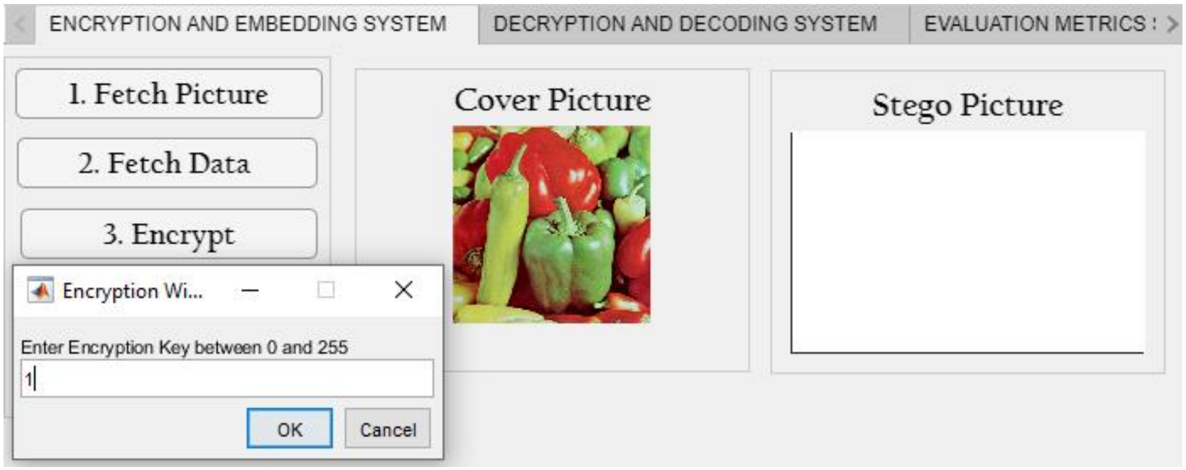
File name: Secret Message3.txt

(*.txt)

The system's fetch data tab is used to upload a credit card number sample containing sixteen characters. After successfully uploading credit card number sample, AES encryption is applied to convert it to ciphertext before being embedded onto the cover image.  Encrypt tab is exbibited in Figure 4.25.

Figure 4.25

Encrypt Tab (Researcher, 2023).



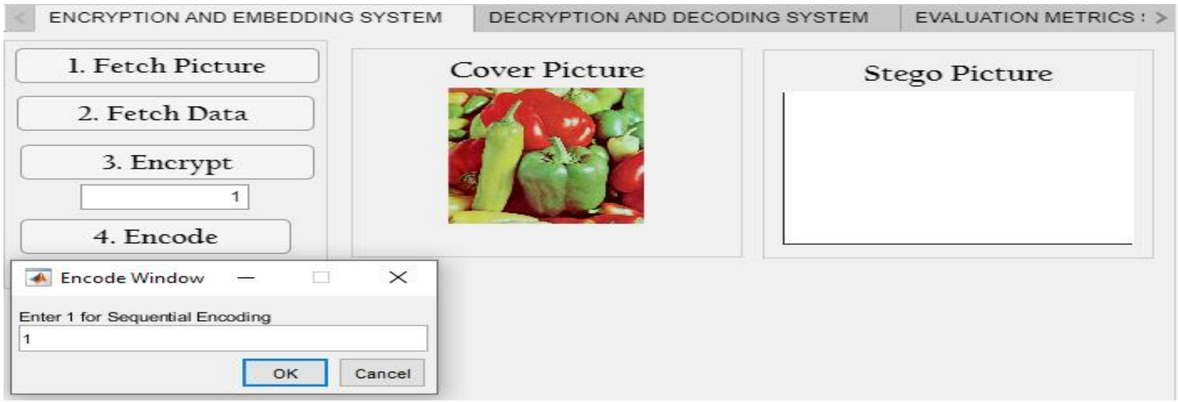
Enter Encryption Key between 0 and 255

1

The function of AES encryption is to transform the credit card number that has already been uploaded into the system to ciphertext. Ciphertext is unreadable format of a message and in this simulation our message is the credit card number. The same key used in this step is the same that will be utilized during decryption process. Encryption tab is exhibited in Figure 4.26.

Figure 4.26

Encode Tab (Researcher, 2023).



Encode Window

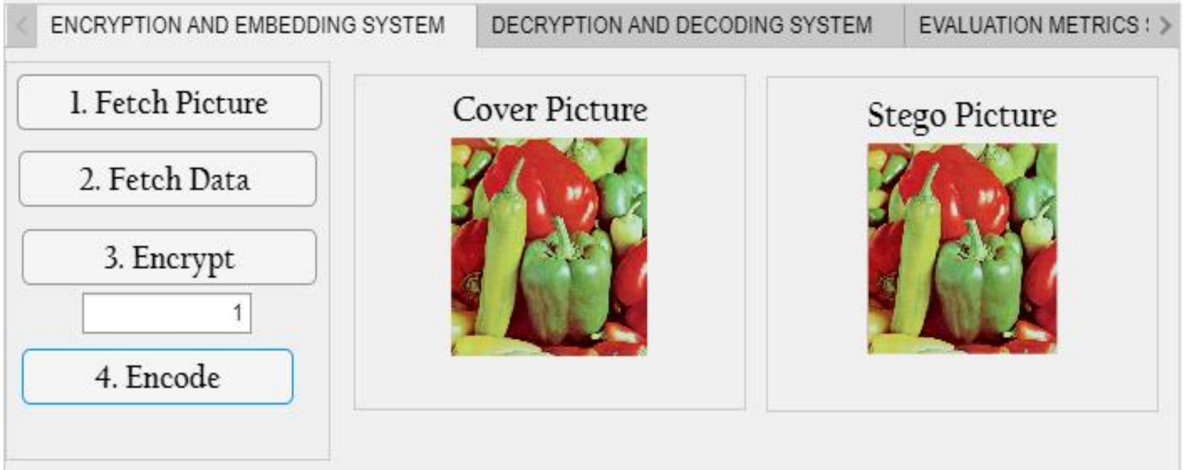
Enter 1 for Sequential Encoding

1

In this simulation test, sequential encoding is used in which a key is used to pick picture pixels and embed the message sequentially. LSB substitution was used to encode the image pixels sequentially as shown in Figure 4.27.

Figure 4.27

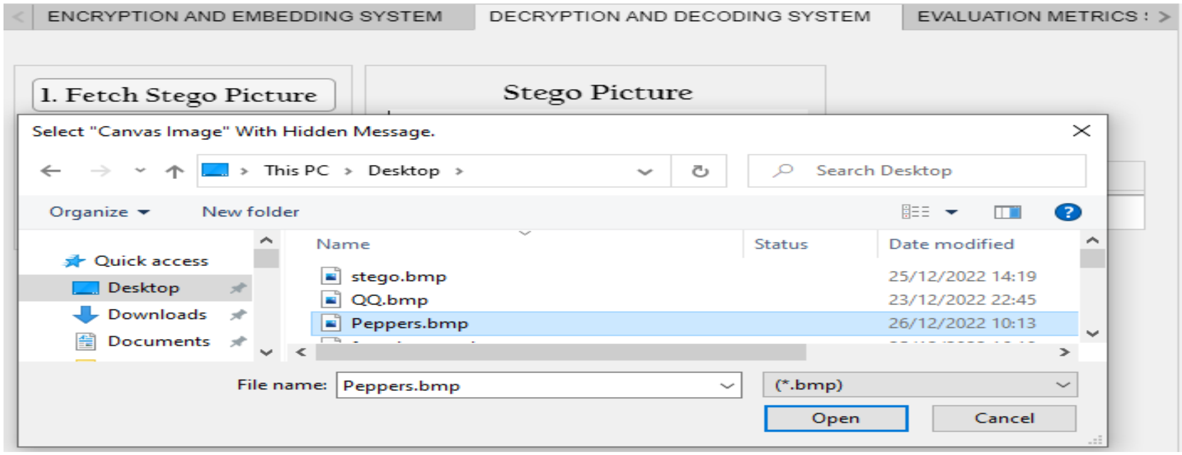
Cover and Stego Picture Display Interface (Researcher, 2023).



The cover image and stego-image are exhibited in their respective UIAXes as shown in Figure 4.27. The stego picture is then saved in bmp format. In mobile banking applications, this picture is anticipated to be transmitted to a receiver to retrieve hidden information and use it to securely conduct online banking. Simulation test three on the encryption and embedding system was successfully implemented and stego-image saved successfully. The next component is decryption and decoding in which three tabs are utilized. Figure 4.28 exhibits the fetch stego picture tab.

Figure 4.28

Fetch Stego Picture Tab (Researcher, 2023).



Select "Canvas Image" With Hidden Message.

| Name | Status | Date modified |
|---|---|---|
| stego.bmp | | 25/12/2022 14:19 |
| QQ.bmp | | 23/12/2022 22:45 |
| Peppers.bmp | | 26/12/2022 10:13 |

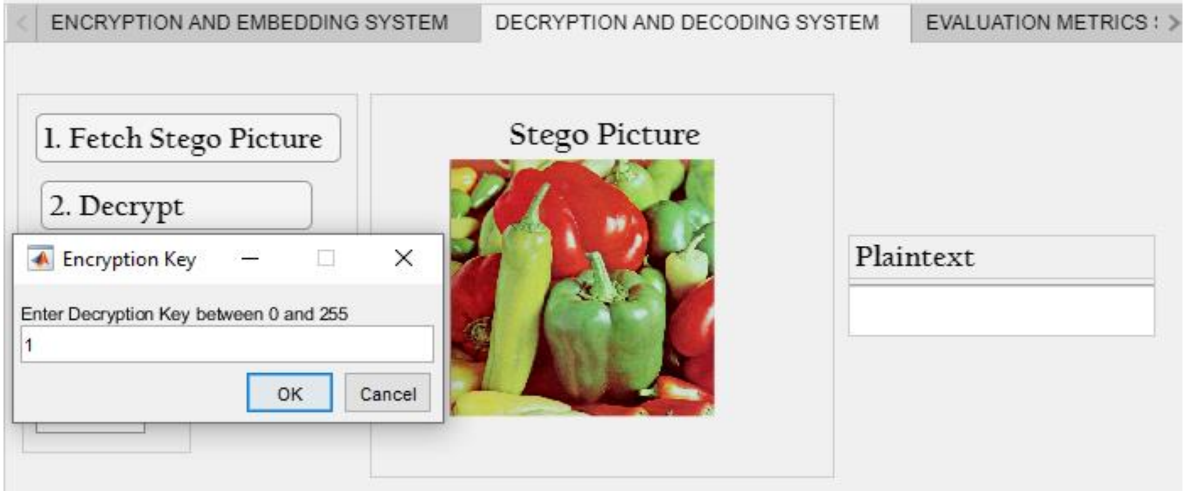
File name: Peppers.bmp

(*.bmp)

The fetch stego picture tab upon provokes an open dialog form to locate the stego-images on the computer and uploads them into UIAXes platform. In simulation test three, the system's fetch stego picture tab is utilized to upload a stego -image of peppers with bmp format. Once the stego-image is uploaded successfully, it is exhibited in the respective UIAXes. Figure 4.29 puts on view the peppers stego-image.

Figure 4.29

Decrypt Tab (Researcher, 2023).



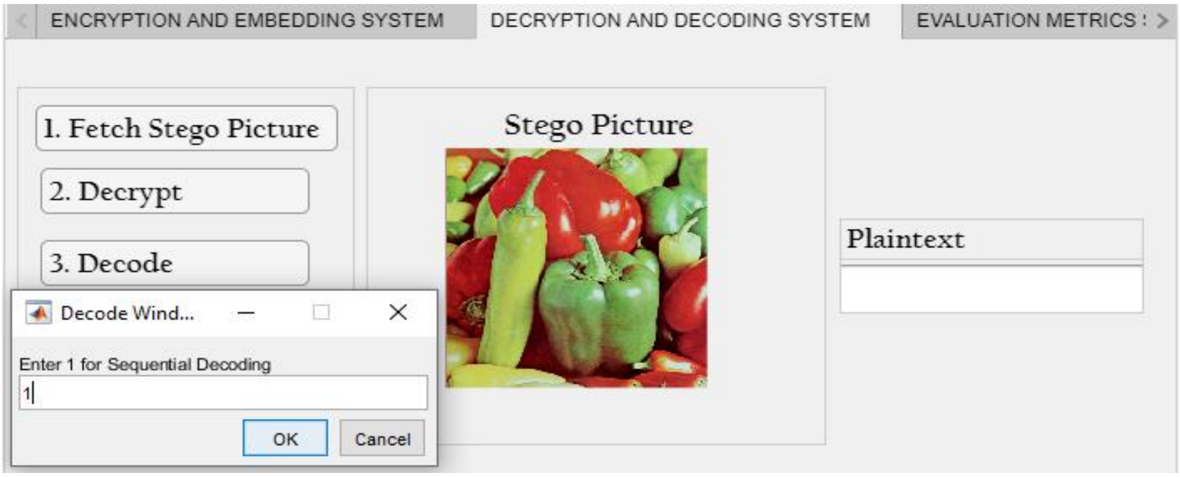
Enter Decryption Key between 0 and 255

1

The decrypt tab provokes an interface for a user to enter decryption key. After successfully fetching the stego-image, the credit card number is restructured from ciphertext to plaintext. The same key that is used in this step corresponds to one used during encryption process. Figure 4.30 depicts decrypt tab.

Figure 4.30

Decode Tab (Researcher, 2023).



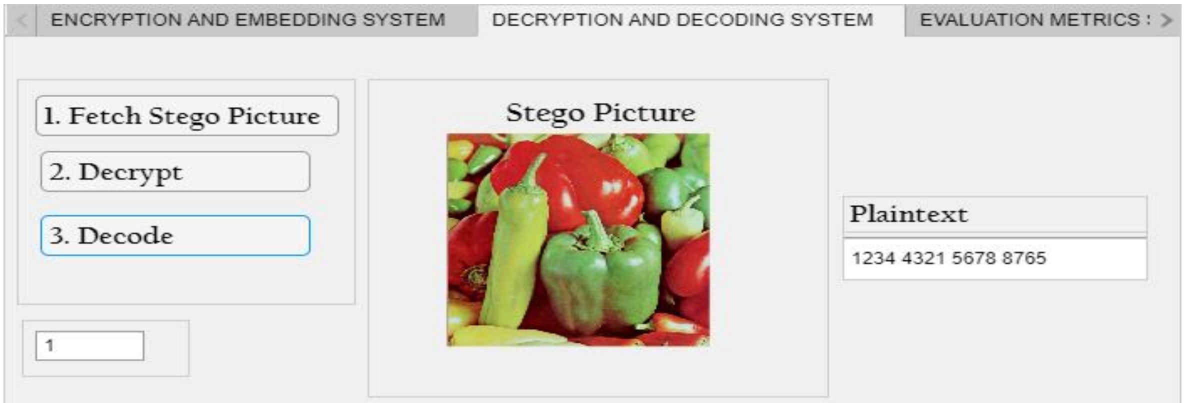
Enter 1 for Sequential Decoding

1

This tab inspires a popup window in which a user can enter a decoding style. In this simulation test, sequential decoding is used in which a key is used to pick image pixels and decode embedded message bits sequentially. Figure 4.31 exhibits successfully decoded credit card number in plaintext form.

Figure 4.31

Credit Card Number Display (Researcher, 2023).



1

1234 4321 5678 8765

Simulation test three was successfully implemented by encrypting and embedding credit card number into a cover-image and decrypting and decoding credit card number securely on the developed system.

### 4.4 Evaluation and Results

This objective sought to evaluate the developed LSB-AES hybrid algorithm and performance metrics comparison in order to establish findings of similar studies that had been done. Figure 4.32 reveals the evaluation metrics MSE, PSNR and entropy results conducted on six cover images Airplane, Female, House, Couple, Peppers and Sailboat.

Figure 4.32

MSE, Entropy and PSNR Results (Researcher, 2023)

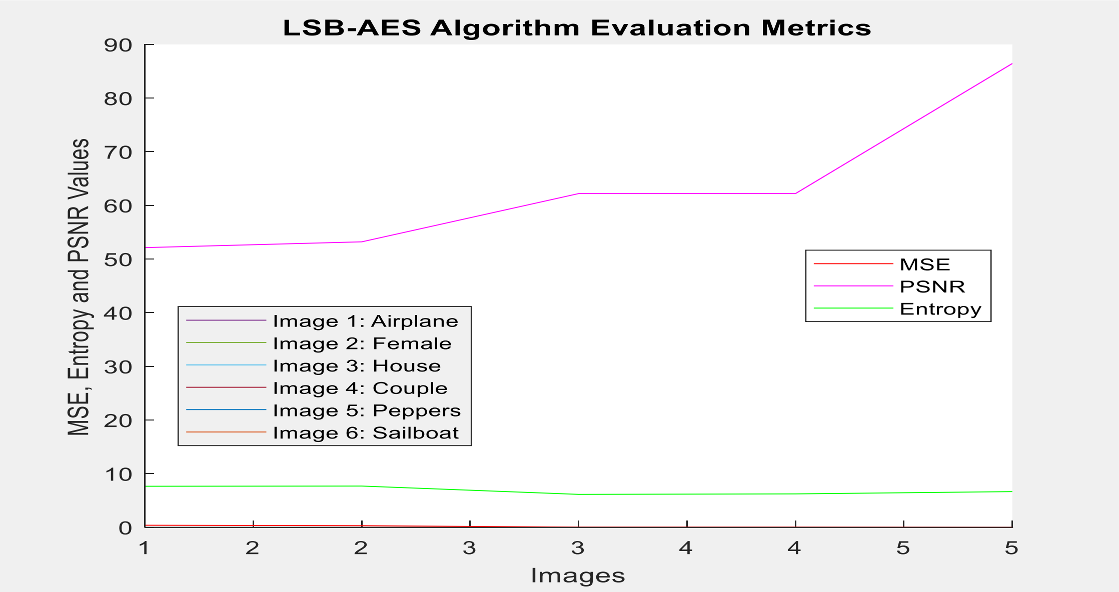


Figure 4.32 illustrates MSE, PSNR and entropy values on the y axis and images used in the test experiment on the x axis with the numbers 1, 2, 3, 4, 5, 6 representing the Airplane, Female, House, Couple, Peppers and Sailboat respectively. To achieve better text concealment in an image, MSE should be low. In the simulation experiments detailed in Figure 4.32, MSE values ranged from 0.0001297 to 0.0005646, indicating minimal error. Lower MSE values suggest that the stego images are very similar to the original cover images (Sogaard, Krasula, Shahid, Temel, Brunnstrom, & Razaak, 2016). These results align with the MSE findings of Abikoye, Ogundokun, Misra, and Agrawal (2022), which ranged from 0.0002124 to 0.0009422, as well as with Msallam (2020), whose MSE values ranged from 0.0011 to 0.0015. A lower MSE is recommended for more effective data concealment (Singh, Choubisa, & Soni, 2020).

The PSNR values for the cover and stego pictures ranged from 80.65 to 87.04, as shown in Figure 4.32. Higher PSNR values indicate better imperceptibility, suggesting that the stego images were robust and it would be difficult to detect the hidden message. These findings are consistent with the PSNR results reported by Abikoye, Ogundokun, Misra, and Agrawal (2022), which ranged from 78.389 to 84.858, as well as with the PSNR results of Msallam (2020), which were between 82.3599 and 83.0220. A PSNR value above 40 dB generally indicates that a stego image has adequate imperceptibility (Setiadi & Jumanto, 2018). Additionally, this study's results align with those of Bandekar and Suguna (2018), whose findings showed high PSNR values ranging from 48.55 to 55.38 dB, further indicating excellent imperceptibility.

The entropy values of the pictures, as shown in Figure 4.32, were 6.295 to 7.67. Entropy measures the security level of the system, with ideal values approaching 8 (Hari, Syaiful, Moses, & Atika, 2017). The system demonstrated acceptable entropy levels, indicating high security and suitability for mobile banking applications. The entropy findings from this study align with those reported by Jain and Kanwal (2016), whose values ranged from 6.926 to 7.607, and Msallam (2020), whose values were between 7.1914 and 7.4518. In contrast, Al-Amri, Hamood, and Farhan (2023) reported lower entropy values, ranging from 4.1443 to 5.447.

### 4.4.1 MSE Analysis

Six cover images were used in simulation experiments, and the results showed that the associated stego images had low MSE values and were quite similar. MSE values of cover and stego images that are closer to zero are better and indicate that the images are similar

(Sara, Akter, & Uddin, 2019). The following were the MSE values for the images: Couple 0.0005646, Sailboat 0.0001297, Peppers 0.0001297, Airplane 0.0001488, Female 0.0004425, House 0.0005035. These values infer that the images used in the developed algorithm were of good quality. The MSE values for these photos are displayed in Figure 4.32.

A study by Elshazly, Safey, Abdelwahab, Fikry, and Elaraby (2016) on a robust image steganography technique using LSB in the spatial domain reported an average MSE value of 0.17884. Kumar & Singh (2017) conducted a study on LSB of color images and achieved an average MSE value of 0.012386. Another research was conducted by Essa, Abdullah, and Al-Dabbagh (2018) on steganography technique using genetic algorithm from which an average MSE value was 0.224. A study by Darwis, Junaidi, Shofiana, and Wamiliana (2021) on digital image steganography using center-embedded pixel positioning reported an average MSE value of 0.454. Figure 4.33 show the four researches and the average MSE values.

Figure 4.33

MSE Performance (Researcher, 2023).

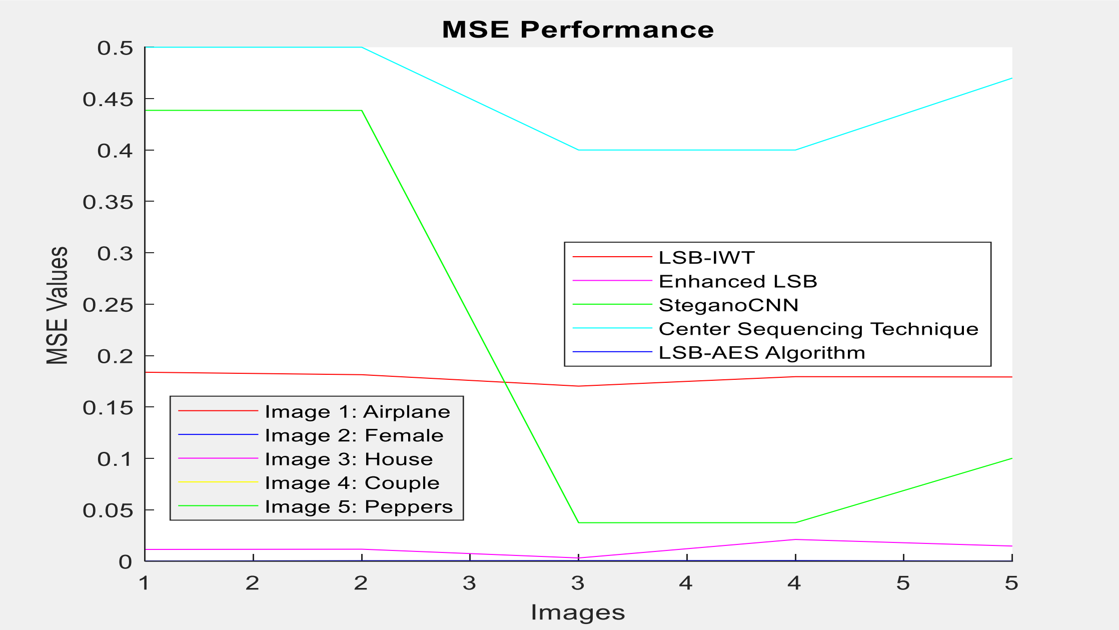


Figure 4.33 average MSE values on the y axis and images used in the test experiment on the x axis with the numbers 1, 2, 3, 4, 5 representing the Airplane, Female, House, Couple and Peppers respectively. The average MSE is a comparison from four studies: Elshazly *et al*., (2018) who developed an algorithm known as LSB-IWT, Essa *et al*. (2018) who developed an algorithm known as Enhanced LSB, Darwis *et al*., (2021) who developed an algorithm known as center sequencing technique based on LSB and Kumar and Singh (2017) whose algorithm is known as SteganoCNN. Findings from their studies indicate that they had good MSE results which pass the acceptable minimum threshold of being close to zero. When MSE values are close to zero, it indicates that the cover and stego pictures are identical (Hashim, Rahim, Johi, Taha, &Hamad, 2018). The average MSE values of Elshazly *et al*., (2018) and Kumar and Singh (2017) are 0.17884 and 0.012386 respectively which are much bigger compared to the average MSE value from this study which is

0.00035782. additionally, Figure 4.33 illustrates average MSE values of two more researchers; Essa *et al.,* (2018) and Darwis *et al.,* (2021), which demonstrates that their average MSE results met the acceptable minimum threshold, being close to zero. The average MSE values of the two authors were 0.224 and 0.454 respectively which were bigger compared to the average MSE values from this study which was 0.00035782. This infers that the pictures used in the developed algorithm were of good quality.

### 4.4.2 PSNR Analysis

The benchmark values for PSNR should be more than 39 dB to indicate good imperceptibility (Sukumar, Subramaniyaswamy, Vijayakumar, & Ravi, 2020; Lakshmi, Srinives, Kumar, & Chandra, 2016). In the developed algorithm, the PSNR readings were as follows: Sailboat 87.14 dB, Peppers 87.04 dB, Couple 80.65 dB, House 81.14 dB, Female 84.71 dB and Airplane 86.44 dB.  These values indicate that the images utilized in the developed algorithm had good imperceptibility and passes the minimum threshold of 39 decibel. The PSNR values for these photos are shown in Figure 4.32.

The algorithm proposed by Muhammad, Sajjad, Mehmood, Rho, and Baik (2016) for image steganography using an uncorrelated color space achieved an average PSNR value of 61.706. Heidari and Farzadnia (2017) conducted a study on quantum LSB-based steganography of color images and achieved an average PSNR value of 55.591. Another research was conducted by Muhammad, Sajjad, Mehmood, Rho, and Baik (2018) on image steganography using uncorrelated color space from which an average PSNR value was 52.352. Finally, research by Abdelraout (2021) on image steganography based on visual color sensitivity yielded average PSNR value of 84.478. Figure 4.34 illustrates PSNR performance.

Figure 4.34

PSNR Performance (Researcher, 2023).

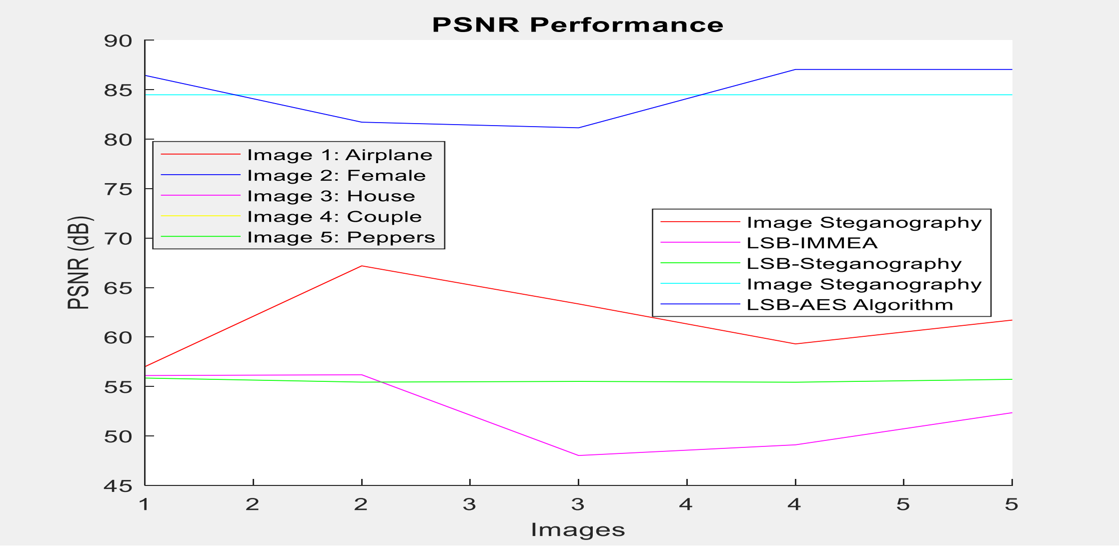


Figure 4.34 illustrates PSNR comparison from four studies: Muhammad *et al*., (2016) developed image steganography algorithm, Heidari and Farzadia (2017) developed LSB steganography algorithm. Their findings indicate that they both achieved good results. However, when compared with the developed LSB-AES hybrid algorithm, the average PSNR was enhanced from 61.706 and 55.591 respectively on the two researches on Figure 4.34 to 84.68. This implies that the developed algorithm had good imperceptibility than the two researches.

Similarly, Figure 4.34 illustrates PSNR values from two other studies; Muhammad *et al.*, (2018) who developed LSB-IMMEA algorithm and AbdelRaout (2021) who developed image steganography algorithm. Findings from their studies indicate that they had good

PSNR results which pass the acceptable minimum threshold of more than 39 dB (Sukumar, Subramaniyaswamy, Vijayakumar, & Ravi, 2020). However, when comparing the average values of the two researchers to the PSNR values of this study, there was an enhancement from 52.353 and 84.478 to 84.674 on the developed algorithm. This infers that the PSNR value of the developed algorithm had good imperceptibility than the two researchers.

### 4.4.3 Entropy Analysis

Entropy analysis was used to determine the security of the images used in the study. For a system to be robust, entropy values should be close to the benchmark value of 8 (Hari, Moses, Syaiful, & Atika, 2017). The following entropy values were found via simulation tests conducted on the cover photos and their corresponding stego-images: Sailboat 7.762, Female 6.898, House 7.069, Couple 6.295, Airplane 6.664, and Peppers 7.670. These entropy values are close to the acceptable benchmark of 8 inferring that the images have good security for embedding secret messages.

Figure 4.35

Entropy Performance (Researcher, 2023).

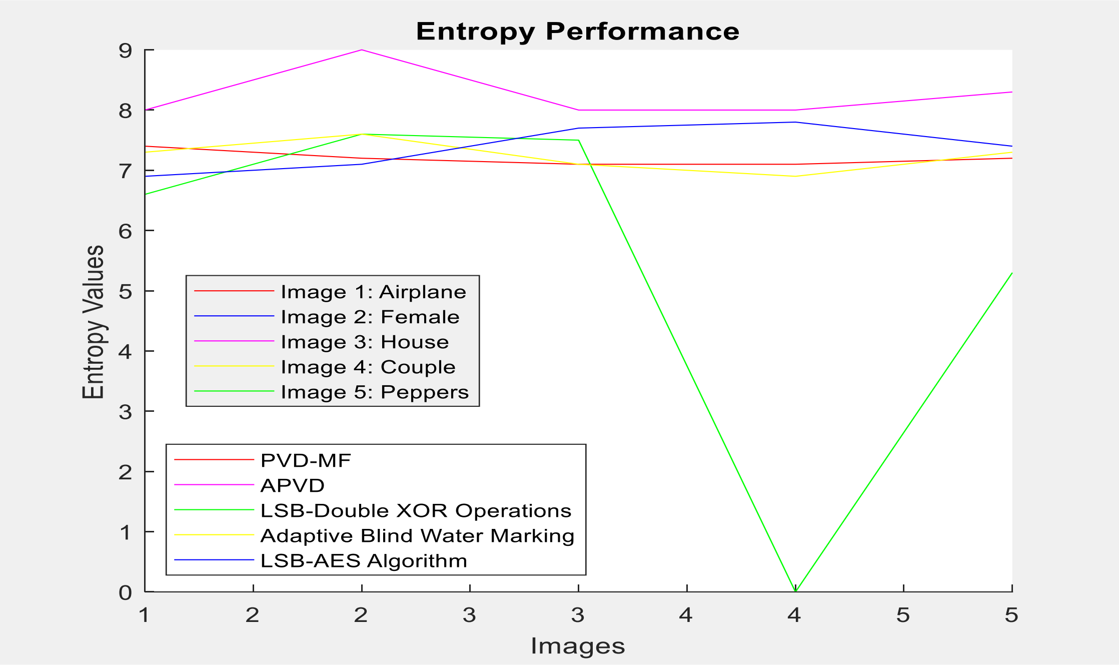


Figure 4.35 illustrates entropy values from four studies; Saha *et al*., (2014) and Rusuma *et al*., (2018). Findings from their studies indicate that they had good entropy results which are close to 8. Whenever entropy values are close to 8, it infers that stego-picture has embedded data and is considered to have good security. The average entropy values of the two authors were 7.25965 and 7.993275 respectively. When comparing the values with the developed algorithm, Rusuma *et al*., (2018) had higher entropy values. This infers those images utilized for the study had good security than the developed LSB-AES hybrid algorithm.

Additionally, Figure 4.35 illustrates average entropy values from Ahmed and Ahmed (2020) and Sharma *et al*., (2021). Findings from their studies indicate that they had good entropy results which are close to 8. The average entropy values of the two authors were

7.2474 and 7.22847 respectively. Compared to the entropy values of the developed algorithm, the developed algorithm offers better entropy, indicating greater security than the approaches of the two authors.

The overall performance of the developed LSB-AES hybrid algorithm shows that it is more robust than other algorithms. The developed LSB-AES hybrid algorithm exhibited higher entropy values. Higher entropy values indicate more complexity and less predictability of the stego images which is desirable to avoid detection of hidden messages.

Higher PSNR values indicate better quality of the stego image, inferring that the embedding process has introduced minimal distortion. When compared to other studies, this study outperforms all other studies with higher PSNR values. Additionally, the developed LSB-AES hybrid algorithm exhibited lower MSE values which indicate better quality of the stego image and is desirable detection of hidden messages.

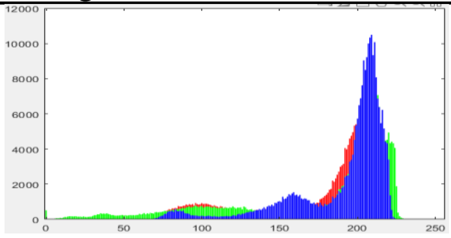
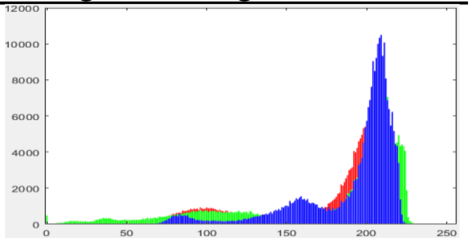
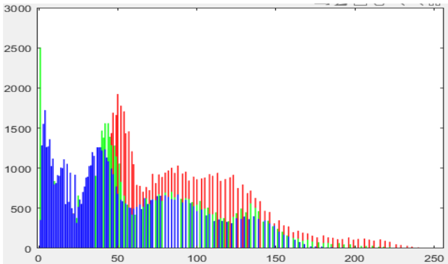
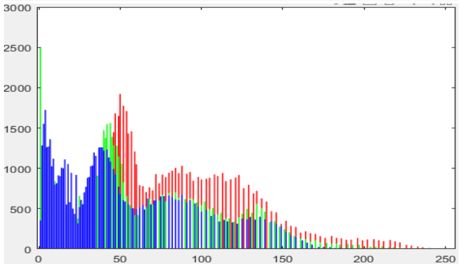
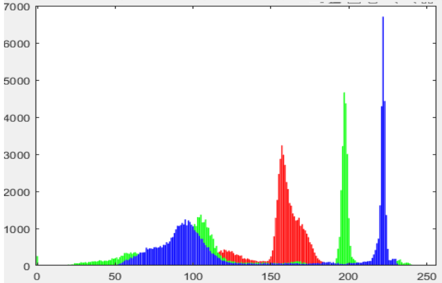
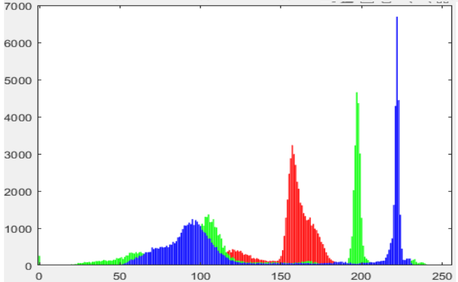
### 4.5.5 Histogram Analysis

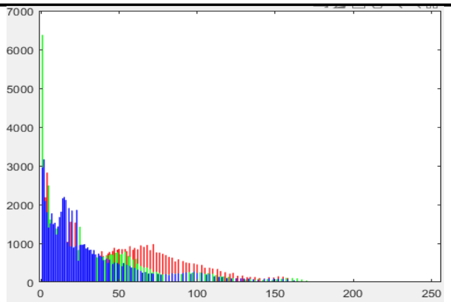
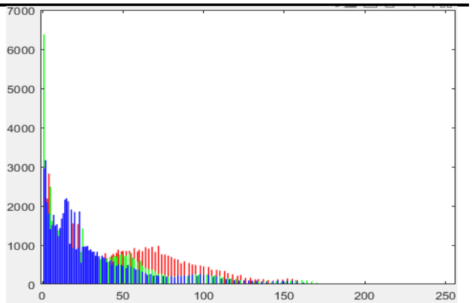
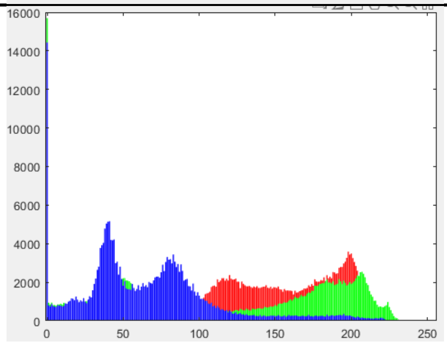
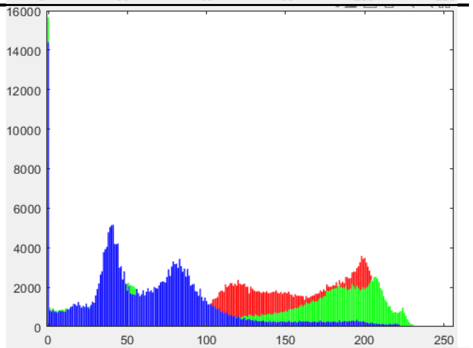
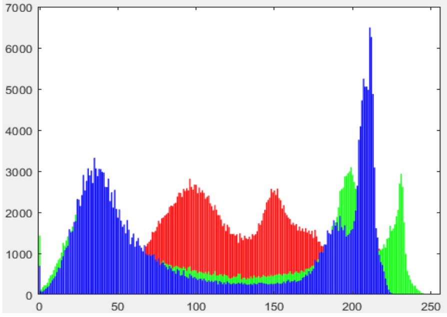
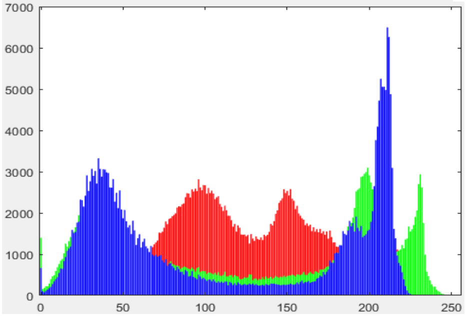
A simulation experiment was conducted to examine whether the color distribution changes when text is embedded into cover pictures. The RGB color distribution histograms reveal minimal differences between the images, indicating that the proposed algorithm is both reliable and well-suited for mobile banking applications. It is challenging for an adversary to detect hidden messages within the images. These findings are consistent with a study by Tauhid, Tasnim, Noor, Faruqui, and Yousuf (2018), which showed that the differences between cover and stego images were nearly imperceptible, reflecting the improved performance of their algorithm.

Additionally, Beza (2018) developed a secure mobile banking framework combining LSB steganography with AES encryption for Multimedia Messaging Service (MMS). The simulation results demonstrated that the AES algorithm outperforms others in terms of encryption time, power efficiency, and low memory consumption. Histograms generated from simulation experiments are displayed in Table 4.2.

Table 4.2

Cover Picture and Stego Pictures with their Histograms (Researcher, 2023).

| Cover Picture | Stego Picture | Histogram of Cover Picture | Histogram of Stego Picture |
|---|---|---|---|
| | |  |  |
| | |  |  |
| | |  |  |

|  |  |  |  |
|---|---|---|---|
| | |  |  |
| | |  |  |
| | |  |  |

The histograms displayed on table 4.2 demonstrate that there was hardly any slight distortion between the cover and stego pictures, even with hidden messages included in each of them. A cover picture is a media that is used to hide a message while a stego picture is a media that contains hidden message. The cover and stego pictures were extremely similar, indicating that there was only slight distortion if any when the hidden message was inserted into the cover pictures. This is because LSB substitution technique was utilized to substitute message binary digits into the picture pixels.

In terms of analysis of image quality, the PSNR values ranged from 16.652 to 35.687 decibels. These results are not consistent with the minimum PSNR threshold values which should be above values of 40 (Lakshmi, Srinives, Kumar, & Chandra, 2016). This infers that the type of images that were used in his simulation experiments were not of good quality.

The algorithm proposed by Panwar, Damani, and Kumar (2018), which combined modified LSB steganography with AES encryption, was evaluated using two standard analysis metrics: MSE and PSNR. The MSE values ranged from 0.00036 to 0.00149, while the PSNR values ranged from 76.38 to 82.49. From this study, low MSE values indicate good quality images and high PSNR values of above 40 decibels infer good imperceptibility. The modified LSB combined with AES encryption proved supportive in conserving safety of hidden secret data in stego-pictures. It is therefore consistent with our current research because it utilizes hybridization with low MSE and high PSNR evaluation metrics values.

The current study results are consistent with findings of a hybrid algorithm that combines AES and Blowfish algorithm proposed by Purwinarko and Hardyanto (2018). Findings from this study infers that although AES algorithm is secure, it needs to be enhanced by combining it with other algorithms to avoid attacks which occur due to the vulnerability of the s-box in AES algorithm (Verma, Guha, & Mishra, 2018). This study reveals that a hybrid algorithm that combines AES with another algorithm such as Blowfish provides more security and prevent hackers from MITM attack.

A hybrid algorithm that combines ECC and image steganography for secure mobile banking applications was proposed by Karthikeyan, Vijayarajan, and Manikandan (2011). In this proposed algorithm, ECC was employed to strengthen the overall security, while steganography was used to protect information as it is transmitted from the bank server to the client application. The results demonstrated that the algorithm addressed the security weaknesses of RGPS and SMS banking, offering mutual authentication, non-repudiation, and data integrity. Additionally, the system showed that the embedding capacity of the image pixels was greater compared to the SMS packet capacity. Histogram analysis of the images indicated that there were no significant changes after embedding the data. This study used ECC and LSB steganography. The study's results are consistent with results of this study in which histogram of cover and stego pictures do not show any significant changes.

Histogram analysis results from this study indicate that it is not easy to detect anomalies resulting from embedding data into the cover images since the cover and stego images were similar when assessed using the human eye. Similarly, the results indicate that the developed algorithm is robust against attacks such as MITM.

### 4.6 Steganalysis and LSB-AES On-Transit User-Data Protection Algorithm

The developed algorithm was implemented at the transport layer of the OSI model, which is responsible for ensuring that data is fully transferred between servers or systems. The Transmission Control Protocol (TCP) at this layer manages end-to-end error recovery and ensures complete data transmission (Fraihat, 2020). Implementing the algorithm at this layer addresses the vulnerability where MITM attacks occur, involving the interception and manipulation of communication between two parties.

Steganalysis is a method of locating and extracting information that has been embedded in a cover picture. Both computer forensics and internet crime use steganalysis (Ibrahim, 2007). According to (Meghanathan & Nayak, 2010), specific and generic algorithms are used for image steganalysis. Generic methods need extensive statistical analysis and intricate computations and operate independently of the steganographic algorithms. They deliver a solid and enhanced outcome (Amirtharajan & Rayappan, 2013). The algorithm employed in each steganalysis relies on the steganography algorithm and the image format (Amirtharajan & Rayappan, 2013; Meghanathan & Nayak, 2010).

The main approaches used in steganalysis are statistical analysis and visual analysis (Curran & Devitt, 2008; Wang & Wang, 2004). Human eyes are required for visual analysis. Human abilities are used in visual analysis to look for concealed data and covert communication. When sensitive data is inserted in the picture's portion, visual inspection by the eyes can discern and successfully detect concealed signals (Curran & Devitt, 2008; Wang & Wang, 2004). Statistical analysis can spot minute changes in the image's statistical features brought on by steganography, even though steganography is typically detected via statistical analysis (Curran & Devitt, 2008; Wang & Wang, 2004). The general public has access to a wide range of steganalysis tools, such as SteganalyzerAS and Steganalyzer SS, which are examples of technologies available for this purpose (Green & Levstein, 2015).

This study demonstrates MITM attacks in mobile banking applications and used visual analysis to find concealed information in a stego-picture. Visual analysis was employed and judged the cover and stego-pictures using MSE, PSNR, and histograms as exhibited in Figure 4.32. Low MSE indicates stego pictures were similar to or an exact replica of the cover picture. This indicates that the stego and cover pictures were similar or identical. Second, the stego and cover pictures have PSNR values of more than 40db. Thirdly, histograms from the developed method show that there were little to no differences between the cover and stego pictures and therefore an adversary cannot obtain clues about any hidden messages in the pictures using human eyes. This illustrates that the pictures had adequate imperceptibility.

MITM attack is not feasible in the developed system because even if the enemy manages to intercept a message in the stego-picture, the message is invincible with the naked eye. This is because the findings of the simulation tests show that given the results of MSE, PSNR, and histograms, it would be complicated to find a concealed message in the stego-pictures. Similar to this, the algorithm incorporated a synergy of two algorithms; AES and LSB in the system's simulation test experiment. The secret message is encrypted and then incorporated onto the stego-picture. This suggests that even if MITM used other steganalysis methods to determine that a concealed message exists, he or she would not succeed in decrypting the message from the stego-picture since they lack the decryption key. The created technique is hence resistant to MITM assaults. Table 4.3 illustrates cover pictures and stego pictures which infer the difficulty to conceive which picture contains encrypted message.

Table 4.3

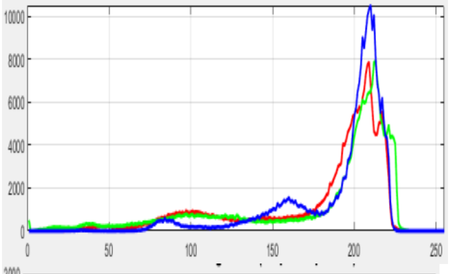
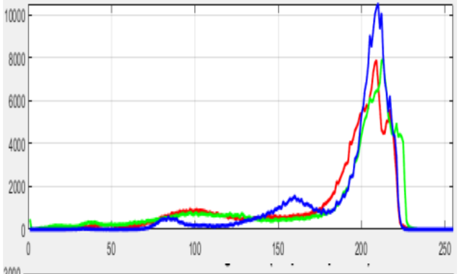
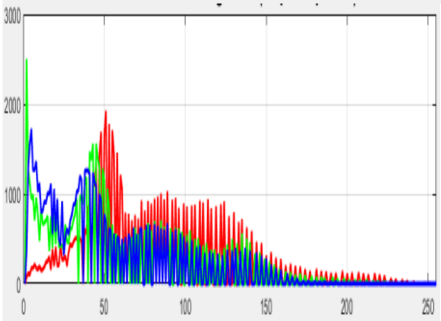
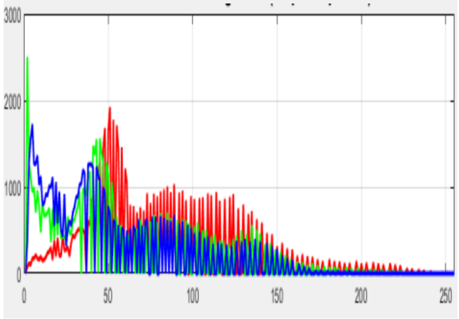
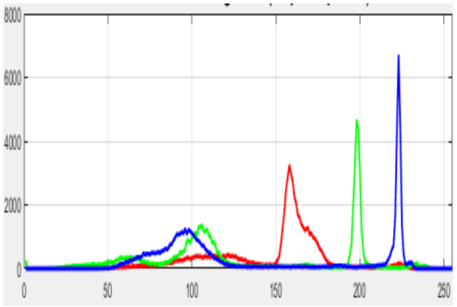
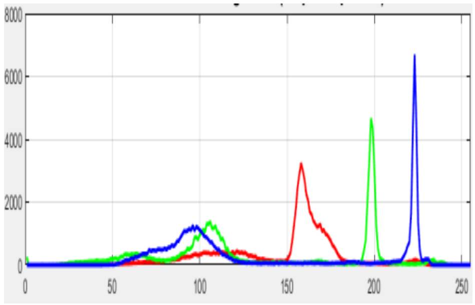
Cover Pictures and Stego Pictures on Visual Analysis Attacks (Researcher, 2023).

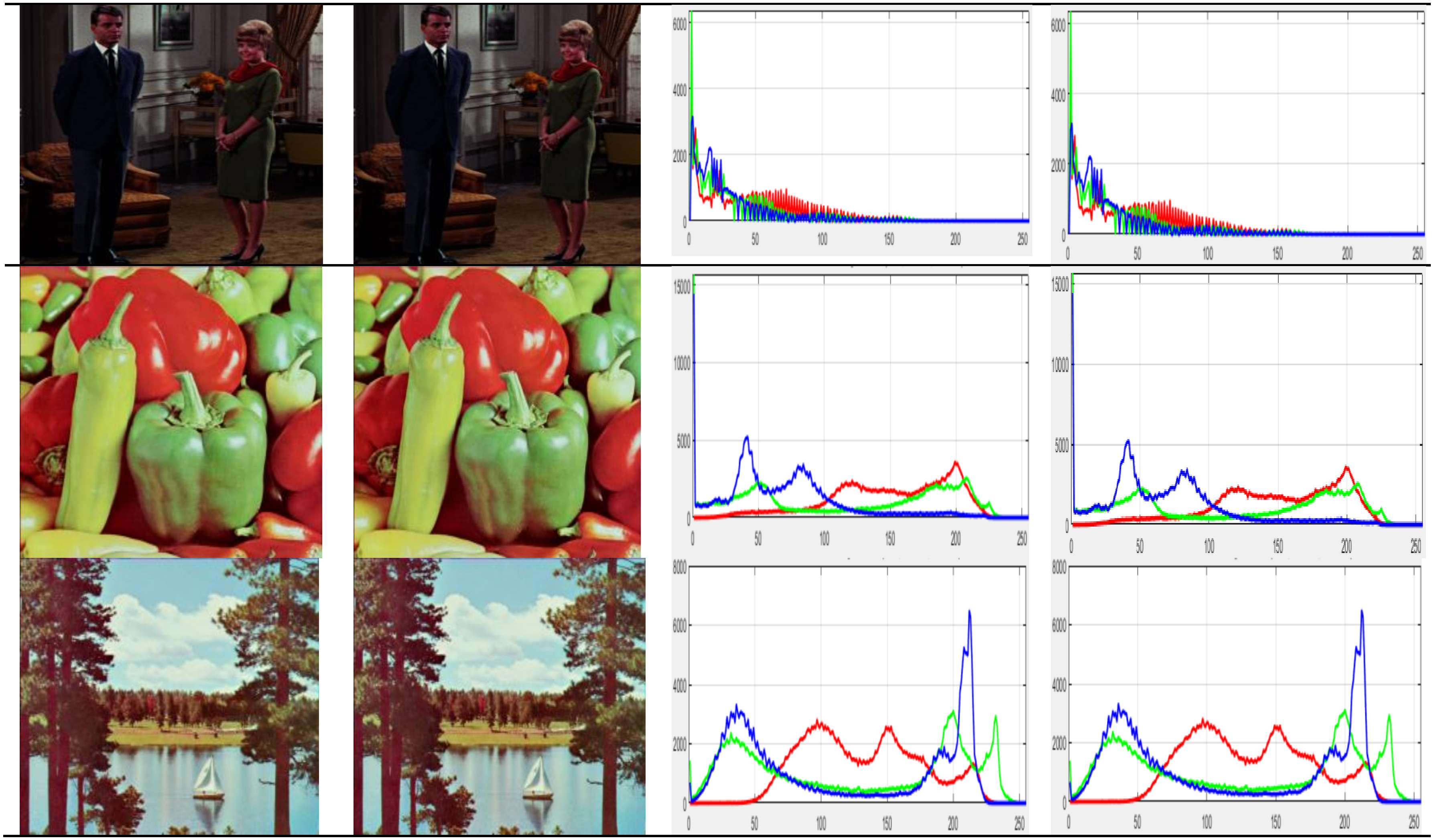
| Name | Cover Picture | Stego Picture |
|---|---|---|
| Airplane | | |
| Female | | |
| Peppers | | |

Table 4.3 illustrates three cover pictures and their corresponding stego-pictures. From the three sets of pictures, it can be inferred that using visual analysis, it is complicated for an adversary to detect existence of a message from either of the pictures because they closely resemble even though a secret message is embedded on the corresponding stego-pictures. Likewise, even if an adversary were to detect the presence of a message in the stego-pictures, extracting the message would be challenging due to its encryption. Table 4.4 presents the histogram analysis in response to statistical attacks.

Table 4.4

Histogram Analysis on Statistical Attacks (Researcher, 2023).

| Cover Picture | Stego Picture | Histogram of Cover Picture | Histogram of Stego Picture |
|---|---|---|---|
| | |  |  |
| | |  |  |
| | |  |  |


6000
4000
2000
0
0
50
100
150
200
250
15000
10000
5000
8000

Histogram analysis common statistical analysis technique was utilized to check robustness of a stego-picture against attacks. Histogram analysis can be performed on a stego-picture and compared to the cover picture. Deviations in the histogram of the stego-picture relative to the cover picture can indicate the presence of hidden data. To counter visual analysis attacks, the histogram of the stego-picture should closely resemble that of the cover picture. The first recorded statistical attack method is the histogram attack (Fridrich & Goljan, 2002). This approach is based on the observation that, with LSB embedding, odd pixel values tend to either decrease or remain unchanged, while even pixel values typically increase by 1. Consequently, during embedding, pairs of values such as (2_i, 2_(i+1)) are swapped.

A statistical measure can be used to determine whether the embedding function’s asymmetry indicates that the even values follow the known distribution. The expected distribution of the values in the presence of hidden information is then shown in a theoretical histogram of the color distribution, and adjacent value pairs are once more constructed. For each pair, the discrepancy between actual and anticipated occurrence frequencies is sought for. In our study, the observation from Table 4.4 infers that when the secret messages were concealed in the cover picture, the tonal distribution was unaffected.

The histograms of the cover pictures and stego pictures for each of the images used are identical. This is because LSB substitution technique was used to embed the message binary digits onto the picture pixels. When the histograms are plotted, the resultant stego histogram resembles the histogram of the cover picture.

# CHAPTER FIVE

# DISCUSSION

## 5.1 Overview

In this chapter, the following are presented; results and discussions on operation of mobile banking applications that allow users to access banking services remotely, security threats affecting user data on transit in mobile banking applications and techniques used to secure user data on transit in mobile banking applications.

## 5.2. Operation of Mobile Banking Applications

This objective aimed to assess the functionality of mobile banking applications, focusing on how they enable users to access banking services remotely. It sought to recommend best practices for bank customers to securely use these applications. The goal was to determine the operational effectiveness of mobile banking applications and demonstrate that following best practices can help mitigate potential threats. Section 5.3 further discusses the potential threats to mobile banking applications.

This study found out that, depending on technology employed, mobile banking has both server-side and client-side technologies. Server-side technology is used by the bank's or service provider's server, which keeps customer data secure. Client data is entered manually by the user, automatically encrypted by the program, or encrypted before being saved in the application when using client-side technology in the programs, solutions, and service offers developed or installed on a customer's SIM or smart phone (techopedia, 2022; Rawat & Agrawal, 2015).

A consumer must register with the bank and receive authorization to access the mobile banking services before utilizing the mobile banking applications. An OTP or other security measure is delivered to the account holder's mobile device. HTTPS ensures secure connection, user data privacy, authentication, and integrity. Mobile banking applications should implement HTTPS to protect user information.

Beginning with the establishment of a private conversation between a client's mobile banking application and the bank server, secure mobile banking is achieved through establishment of secure connection. TLS handshake is being used to create this secure session. In order to defend against hackers, TLS makes sure that all data transferred between a user and a bank server is encrypted (Curguz, 2016).

Following the establishment of a secure connection, the client or user is authenticated to the bank's server using login details registered at the banks server so they can use mobile banking services after the server has verified the authentication methods that were used. If authentication is unsuccessful; however, the user is prompted to repeat the login procedure (Waghmare, Golekar, Hatwar, Parimal, & Hiware, 2017)

This study is supported by findings of Waghmare, Golekar, Hatwar, Parimal, and Hiware (2017), who illustrates that operation of mobile banking, begins with establishment of a safe link by utilizing TLS handshaking procedure uniting the client and server applications. Two stage authentications are utilized to access the bank server once a secure connection has been established. To access mobile banking services remotely, a customer must pass the first level of authentication, which utilizes usernames and passwords, and the second level, which utilizes TFA Authentication.

It is worth noting that access to banking services using mobile banking applications is remote in nature and as such, the customer using mobile banking applications is responsible for providing the correct login credentials since there is no physical teller to verify their identity. Therefore, users of mobile banking applications need to be aware of best practices that can help thwart potential threats from cybercriminals in order to access banking services securely.

This study identified several security measures and best practices that users of mobile banking applications can adopt to ensure secure and remote banking. For users who utilize smartphones with authentication features such as username and password, and TFA, MFA is considered as the best practice because it requires it requires utilization of more than two factors of authentication which is essentially difficult for cybercriminals to break, as supported by findings of (La-Polla, Martinelli, & Sgandurra, 2013). Additionally, users of mobile banking applications should utilize complex passwords because they complicate the adversary's efforts to crack or guess the passwords.

Additionally, this study discovered that numerous mobile banking applications are available for download from third-party stores, which users can install on their smartphones. However, these applications often contain viruses and therefore not recommended for use. Therefore, this study established that users should download and install mobile banking applications that are authorized from official bank websites.

This study established that whenever updates for operating system and mobile banking applications are available, it is the responsibility of the user to download and install security patches which resolves recent vulnerabilities and threats. Additionally, users should install mobile antivirus software in order to annihilate mobile malware.

Lastly, this study established that users of mobile banking applications should be aware of social engineering attacks which are directed to unsuspecting users in order to extort confidential information and even to steal money. Therefore, users should be alert on people who solicit confidential information in the name of assisting them, as supported by findings of Dasgupta, Roy, and Nag, (2017).

### 5.3. Threats Affecting User-Data on Transit in Mobile Banking Applications

This objective aimed to assess the security threats impacting user data in transit within mobile banking applications. It identified various types of threats to user data in transit, allowing for the development of appropriate security techniques to address these threats, as discussed in Section 5.4.

Threats commonly affecting various OSI layers include MITM attacks, active and passive eavesdropping, DoS attacks. Additional threats to mobile banking encompass mobile malware, packet sniffing, DNS poisoning, SSL stripping, session hijacking, XSS, and SQL injection attacks. MITM can be either active or passive. In an MITM attack, the secure communication between a mobile banking application and the bank's server is intercepted and divided into two separate segments: one between the mobile banking application and the attacker, and the other between the attacker and the bank’s server. When a MITM attack is successful, both the mobile banking application and the bank's server remain unaware of the compromised communication path. Active MITM attacks can be initiated through various methods, such as DNS spoofing, SSL hijacking, and ARP cache poisoning (Bhattacharya 7 Reddy, 2022; Kaka, Sastry, & Maiti, 2017).

This research affirms findings of Bojjagani and Sastry (2017), who established that MITM attack is the most effective active network attack, and that an adversary can launch many other related attacks from it, including DoS attacks, session prediction attacks, account lockout attacks, and HTTP smuggling, among others. As a result, the network suffers serious damage in this assault since traffic is either intercepted or diverted to the intruder. The most frequent type of assaults in the DoS category arise in the application layer of the OSI model protocols arising from bugs in protocols used at this surface, such as TCP/IP and UDP. Construction of DoS attacks can be automated using software like BackTrack and Metasploit, among other tools.

Finally, this study found out that threats that arise from social engineering can be identified, but they are difficult to stop. Some of the different techniques used to launch social engineering are baiting, pretexting, scareware, phishing, and spear phishing. The defenses against social engineering attacks include giving people the tools they need to recognize and avoid them by educating individuals about safety of sensitive user data, alerting security personnel about suspicious activity, organizing security orientations for new bank employees and customers.

## 5.4 Techniques Securing User-Data on Transit in Mobile Banking Applications

This objective aimed to assess the techniques employed to secure user data in transit within mobile banking applications, assessing their strengths, weaknesses, and operational structures. The study identified various methods used to protect user data from threats such as MITM attacks, DoS attacks, eavesdropping, and social engineering. These methods include symmetric and asymmetric algorithms, steganographic techniques, and authentication mechanisms.

This study found out that mobile banking utilizes AES algorithm due to its complexity with less resource usage than asymmetric schemes like RSA and ECC algorithms. Additionally, even though Blowfish algorithm like AES is a swift block cryptosystem, it is not utilized for mobile banking because of its small block sizes and is susceptible to cryptanalysis attacks (Khelifi, 2013).

This study determined that RSA offers essential and dependable security features, including privacy, confidentiality, authentication, integrity, and non-repudiation. Consequently, RSA is employed in network security because of its robust public-key cryptosystem, which is challenging for hackers to compromise. However, RSA is vulnerable to timing attacks, brute force attacks, and key-related vulnerabilities. Therefore, due to these risks, RSA should not be used in critical systems like mobile banking. In addition to the RSA algorithm, this study found that Elliptic Curve Cryptography (ECC) performs slower than AES. ECC is also vulnerable to electromagnetic analysis attacks making it unsuitable for mobile banking (Mitra, Jana, Bhattacharya, Pal, & Poray, 2017).

Finally, this study found out that steganographic techniques such as LSB can be used to embed data on a media and send it over wireless networks without the knowledge of adversaries. However, this technique when utilized alone is susceptible to steganalysis attacks and thus not suitable for mobile banking.

## 5.5 Development of LSB-AES On-Transit User-Data Protection Algorithm

LSB-AES hybrid algorithm was developed by merging two established algorithms LSB and AES encryption algorithm. The LSB-AES on-transit user-data protection algorithm starts its operation by loading a cover image. This step is followed by confidential data that need to be hidden in the cover image. AES is then applied to encrypt confidential data to secure it before embedding in the cover image. LSB embedding is then applied to hide encrypted data into the LSB bits of the cover image. The stego-image produced is read to the system in order to start the process of extracting hidden data. LSB decoding is applied in order to extract encrypted data from the LSB of the stego-image. AES decryption is then applied in order to decrypt extracted data to retrieve the original confidential information.

The algorithm's system design interface incorporates three major components. The first component with the title ENCRYPTION AND EMBEDDING SYSTEM is responsible for fetching cover pictures from the computer, fetching data, applying encryption key using AES scheme and embedding encrypted data into a cover picture using LSB substitution technique. This component was designed with the following tabs: Fetch Picture, Fetch Data, Encrypt, and Encode tabs. The system also contains two UIAXes where the cover and stego pictures are displayed upon provoked by respective callback tabs as well as an edit textfield where encryption key can be displayed.

The second component with the title DECRYPTION AND DECODING SYSTEM is responsible for fetching the stego picture from the computer, and decrypting ciphertext to recover plaintext message. This component was designed with the following three tabs in mind: Fetch Stego Image, Decrypt, and Decode. The component also contains UIAXes where a stego picture is displayed upon provoked by the respective tab and an edit textfield where a decryption key can be displayed.

The third component with the title EVALUATION METRICS SYSTEM is responsible for calculating and displaying MSE, PSNR, Entropy and Histograms upon provoked by the respective tabs. This component also contains three UIAXes, one for cover picture, the second for stego picture and the third for histograms. Finally, the component contains three edit textfields which are MSE, PSNR and Entropy.

### 5.6 Evaluation of LSB-AES On-Transit User-Data Protection Algorithm

Findings from simulation tests conducted on six cover images show that their corresponding stego images were substantially similar with low MSE values which ranges from 0.0001297 to 0.0005035.  Lower MSE values which is closer to zero indicates good quality embedding. Findings from MSE values of the images used in the study indicate good quality embedding.

PSNR which is a ratio between the maximum possible value of a signal and the power of distortion should be higher to indicate a better quality of embedding. PSNR values in the study were between 80.65 to 87.14 dB and thus indicates that there was better embedding and thus good imperceptibility.

Entropy which is the corresponding states of intensity level which individual pixels in an image can adapt was found out to range between 6.295 to 7.762 in this study. For an image to have good entropy, it's values should be close to the value 8. Thus, findings from this study show that the entropy of the images used is good and indicate the robustness of the developed LSB-AES algorithm.

Histogram analysis shows that there is no distortion between the cover and stego images, even when confidential messages are hidden in them. The cover and stego images were similar. This is because LSB substitution technique was utilized to substitute message binary digits into image pixels.

# CHAPTER SIX

# CONCLUSION AND RECOMMENDATIONS

## 6.1 Overview

Mobile banking applications offers solutions to clients delivered by banks that authorize customers to request and obtain banking services remotely utilizing mobile devices. Mobile banking is developed in view of client and server-side technologies. Server-side technologies are administered on the server of a bank while client technologies are administered using a customer's SIM card or software program downloaded and installed on a smart phone.

In order to access banking services remotely using client application installed on the users' smart phone, a user needs to download a specific bank application and install in on the smart phone. This is followed by configuration from the bank and affirmation on the customer application. After successful configuration, the bank customer replaces login password with user preferred strong password or any other mechanism available on the application such as biometric authentication that utilizes fingerprint, iris, and face, among others.

Following the establishment of a secure connection, the client or user is authenticated to the bank's server using login details registered at the banks server so they can use mobile banking services after the server has verified the authentication methods that were used. If authentication is unsuccessful; however, the user is prompted to repeat the login procedure.

It is worth noting that access to banking services using mobile banking applications is remote in nature and as such, the customer using mobile banking applications is responsible for providing the correct login credentials since there is no physical teller to verify their identity. Therefore, users of mobile banking applications need to be aware of best practices that can help thwart potential threats from cybercriminals in order to access banking services securely.

Therefore, users of mobile banking applications should utilize complex passwords because they complicate the adversary's efforts to crack or guess the passwords, whenever updates for operating system and mobile banking applications are available, it is the responsibility of the user to download and install security patches which resolves recent vulnerabilities and threats, users should install mobile antivirus software in order to annihilate mobile malware and lastly, users of mobile banking applications should be aware of social engineering attacks which are directed to unsuspecting users in order to extort confidential information and even to steal money.

This research established that even though consumers conduct mobile banking securely using their smart phones, there are a number of deficiencies in mobile banking applications. Threats prevalent in some of the OSI layers include MITM, active and passive eavesdropping assaults, and various forms of DoS attacks. Other threats to mobile banking include mobile malware, packet sniffing, DNS poisoning, SSL strip session hijacking, site-to-site scripting, and SQL injection assaults.

Findings from this research indicate that the different forms of threats that affect mobile banking channel are aimed at gaining unauthorized access to customers' bank accounts remotely to siphon money and confidential information.

This research established numerous techniques protecting user data in mobile banking. Some of the techniques established were symmetric and asymmetric cryptographic schemes, authentication mechanisms SFA, TFA, MFA, steganography and combination of two or more cryptosystems. Some examples of symmetric cryptographic schemes include AES, Blowfish, and Twofish algorithms, whereas asymmetric cryptographic schemes include RSA, ECC and DSA.

This study further established that AES is the preferred cryptosystem that is used in high demand security systems such as banks, government systems and wireless communications because it is a strong cryptosystem and that there have been no specific attacks against it so far. Blowfish and Twofish are other symmetric algorithms which are quite fast in execution but have not been recommended by NIST for high security systems such as mobile banking because they are vulnerable to attacks.

The study established that steganography can be used to conceal data by embedding it onto a cover picture such that an adversary cannot identify the existence of a message on the picture. This study utilized picture steganography to hide data. However, when steganography is used alone to embed data, it is vulnerable to steganalysis attacks. Lastly the study established that two or more encryption systems can be combined to provide security of mobile banking systems.

The developed LSB-AES on-transit user-data protection algorithm was developed and implemented using MATLAB software by utilizing AES algorithm and LSB substitution algorithms. The developed system had three major components: encryption and embedding, decryption and decoding, and evaluation metrics. Three simulation tests were conducted using three different cover images and confidential messages. The simulation tests were successfully accomplished through encrypting a message and embedding it on a cover-picture and finally decrypting and decoding confidential message from a stego-picture.

The security properties of the AES and LSB algorithms were integrated in the LSB-AES hybrid algorithm. An essential element of the LSB replacement technique was the ability to conceal a secret message on the cover-image without the message's presence being discovered. Similarly, the AES method made a significant contribution by encrypting a secret message to prevent attackers from decrypting it. As a result, using the LSB-AES hybrid method together is more secure than using the two techniques independently.

The pdeveloped algorithm was evaluated using MSE, PSNR, entropy and histograms. The system revealed low MSE values and higher PSNR values. Histogram analysis was carried out on six cover images. The histograms revealed that there were minimal to no significant changes between the cover and stego-pictures. This infers that it is difficult for MITM to discern that there are hidden messages in the stego-pictures. Entropy was also utilized to measure the security level of the developed system. System values indicate that values were close to entropy threshold of 8 and therefore the system was robust and secure.

To test security tamperproof of the developed algorithm using MITM attacks, a simulation was conducted using visual analysis in order to investigate concealed information in our stego-pictures. Visual analysis was used to judge cover and stego pictures using MSE, PSNR, and histograms. Low MSE values inferred stego and cover pictures were similar. Secondly, PSNR values were more than 40Db which informed adequate imperceptibility level. Thirdly, histograms from the algorithm showed that there were little to no differences between the cover and stego pictures.

MITM attack was not feasible in the developed system because even if the enemy manages to intercept the stego-picture, the message is invincible with the naked eye. This is because the findings from the simulation tests show that, given the results of MSE, PSNR, and histograms, it would be complicated to find a concealed message in the stego pictures. Similar to this, the program incorporated a synergy of two algorithms; AES and LSB from the system's simulation test experiment. The secret messages were encrypted and then incorporated onto the stego-pictures. This suggests that even if MITM used other steganalysis methods to determine that a concealed message exists, he or she cannot succeed in decrypting the message from the stego-picture since they lack the decryption key. The proposed algorithm is hence resistant to MITM assaults.

The developed LSB-AES algorithm exhibited low MSE values which were close to zero and PSNR values greater than the acceptable threshold of 40 dB. The inference from the PSNR values informs that the pictures had good imperceptibility and therefore challenging for an adversary to discover existence of hidden messages on the stego-pictures.

Entropy values were close 8 inferring that the developed algorithm was robust. Histograms of cover and stego pictures were almost similar when secret messages were embedded onto cover pictures inferring that the proposed algorithm is immune to statistical attacks against MITM attacks.

## 6.2 Recommendations

Utilization of mobile banking application for accessing banking services remotely by customers is conducted over wireless networks. As such the technology employed (client-server) infrastructure is susceptible to threats. Therefore, this study recommends users of mobile banking applications to employ best practices such as multi-factor authentication, installation of mobile antivirus to annihilate mobile malware, use of secured wireless networks, installation of applications downloaded from official bank websites and regularly update mobile operating systems as well as mobile banking applications whenever updates are available. These best practices will enable customers to mitigate threats and access banking services remotely and securely.

This study recommends that banks should develop secure systems for mobile banking applications that are robust to combat emerging cybercrime activities. This is because, as technology advances, so does cybercriminals advance their technology in regard to developed systems. This is also beneficial to banks since the image of the banks is preserved and mobile banking adoption will be improved.

To policy makers, this study recommends that the developed hybrid algorithm is secure for utilization in mobile banking applications and that awareness of best practices about best practises in using mobile banking application should be adopted in order to minimize threats to mobile baking applications. In the recent past banks and governments have fallen victims of cybercrimes and money lost. Legislation can therefore evaluate and amend security of mobile banking policies to incorporate the developed hybrid algorithm.

This study provides the needed knowledge for policy analysts and provides awareness of secure techniques that can be adopted for mobile banking application development. This then will help policy analysts to raise public awareness campaigns about threats that affect user-data in mobile banking and propose adoption of the developed secure hybrid algorithm for mobile banking.

The pool of knowledge from this study is important to scholars especially in information security. Particularly, this study has rich content on how mobile banking applications operate to enable user's access banking remotely and shows emerging threats to mobile banking and the best practices that can be followed to minimize cyberattacks on the mobile banking application channel. Additionally, this study has put forth a robust LSB-AES on-transit user-data protection algorithm which serves as a milestone towards development of secure systems for mobile banking. This source of knowledge will also aid in preparation of in-depth content that can be disseminated to learners and interested parties in the society. The study recommends that in order to broaden their knowledge base, scholars and students should make use of the theoretical and practical knowledge gained from this study.

## 6.3 Recommendations for Further Studies

This study sought to develop a hybrid algorithm that merges LSB and AES algorithm to yield a robust security system for mobile banking applications. The study recommends integration of Artificial Intelligence (AI) algorithms with hybrid algorithms to enhance the security of mobile banking applications. Hybrid algorithms integrate multiple cryptographic techniques to leverage their strengths and mitigate weaknesses.

A combination of AI with hybrid algorithms will result to enhanced encryption in which AI optimizes hybrid encryption algorithms by selecting the most secure and efficient cryptographic methods based on the context. For example, AI can dynamically choose between RSA, AES or ECC based on the type of data and the current threat landscape.

AI algorithms can monitor and analyze real-time data to adapt security protocols. This implies that if a potential threat is detected, the system can automatically switch to a more hybrid algorithm or increase the complexity of encryption. Similarly, AI can enhance the effectiveness of hybrid algorithms by continuously monitoring for anomalies in encrypted data and if unusual patterns are detected, AI can trigger additional layers of encryption or other security measure to protect sensitive information.

Additionally, by integrating AI to hybrid algorithms, threats can easily be predicted and prevented. This can be achieved by analyzing vast amounts of data to predict potential security threats and adjust hybrid algorithms accordingly. This will prevent attacks before they occur. Lastly, AI can enhance MFA by integrating biometric data with hybrid algorithms. This ensures that even if one factor is compromised, additional layers of security protect the user account

# REFERENCES

Aas, J., Barnes, R., Case, B., Durumeric, Z., Eckersley, P., Flores-López, J. A., Halderman, A., Hoffman-Andrews, J., Kasten, J., Rescorla, E., Schoen, S., & Warren, B. (2019). Let's Encrypt: An Automated Certificate Authority to Encrypt the Entire Web. *In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1-15.

AbdelRaouf, A. (2020). A New Data Hiding Approach for Image Steganography based On Visual Color Sensitivity. *Multimedia Tools and Applications.* https://doi.org/10.1007/s11042-020-10224-3

AbdelWahab, O.F., Hussein, A.I., Hamed, H.F.A., Kelash, H.M., Khalaf, A.A.M. (2021). Efficient Combination of RSA Cryptography, Lossy, and Lossless Compression Steganography Techniques to Hide Data. *Procedia Computer Science* 182, pp. 5–12.

Abdullah, A. A. (2017). Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt data. Accessed on 24/12/2022 at 19.25 pm from https://www.researchgate.net/publication/317615794

Abdullah, M., & Rana, M.E. (2021). Vulnerabilities in Public Key Cryptography. Proceedings of the *3rd International Conference on Integrated Intelligent Computing Communication and Security*, vol. 4, pp. 627-631

Abikoye, O.C., Ogundokun, R.O., Misra, S., & Agrawal, K. (2022). Analytical Study on LSB-based Image Steganography Approach retrieved on 23/12/2022 at 23.06 pm https://www.researchgate.net/publication/358997836_Analytical_Study_on_LSB-Based_Image_Steganography_Approach. Doi: 10.1007/978-981-16-8484-5_43

Abirami, J., Devakunchari, R., & Valliyammai, C. (2015). A Top Web Security Vulnerability SQL Injection Attack: Survey. *7th International Conference on Advanced Computing*, IEEE, pp. 1-9.

Abood, O.G., & Guirguis, S.K. (2018). A Survey on Cryptography Algorithms. *International Journal of Scientific and Research Publications*, vol.8 (7), pp. 495-516

Abomhara, M., & Koien, M. (2015). Cyber security and the Internet of Things: Vulnerabilities, Threats, Intruders, & Attacks. *Journal of Cyber Security and Mobility,* vol. 4, pp. 65-68

Abuhamad, M., Abusnaina, A., Nyang, D.H., & Mohaisen, D. (2020). Sensor Based Continuous Authentication of Smartphones' Users Using Behavioral Biometrics: A Contemporary Survey. *IEEE Internet of Things Journal*, pp. 1-19. Retrieved on 6/10/2022 at 10.47 AM from https:// www.cs.ucf.edu/~mohaisen/doc/iotj20.pdf

Abuzalata, M., Alqadi, Z., Al-Azzeh, J., & Jaber, Q. (2019). Modified Inverse LSB Method for Highly Secure Message Hiding. *International Journal of Computer Science and Mobile Computing*, vol.8 (2), pp. 93-103.

Acharya, S., & Joshi, S. (2020). Impact of Cyber-attacks on Banking Institutions in India: A Study of Safety Mechanisms and Prevention Measures. *Palarch's Journal of Archeology of Egypt*, vol.17 (6), pp 4656-4670.

Acharya, K., Sajwan, M., & Bhargaya, S. (2013). Analysis of Cryptographic Algorithms for Network Security. *International Journal of Computer Applications Technology and Research*, vol. 3 (2), pp.130- 135

ACSC (2022). Information Security Manual. Guidelines for Cryptography. Retrieved https://www.cyber.gov.au/sites/default/files/2022-06/22.%20ISM%20-%20Guidelines 20 for%20Cryptography%20%28June%202022%29.pdf

Adebayo, O.S., Ganiyu, S.O., Osang, F.B., Ajiboye, S.S., Olamilekan, K.M., & Abdulazeez, L. (2022). Data Privacy System using Steganography and Cryptography. *International Journal of Mathematical Sciences and Computing,* vol 2, pp. 37-45

Agrawal, D.P., & Wang, H. (2018) Computer and Cyber Security. Auerbach Publications, New York. Retrieved from https://doi.org/10.1201/9780429424878 on 11/1/2023 at 10.00 am

Ahmad, S. A., & Garko, A. B. (2019). Hybrid Cryptography Algorithms in Cloud Computing: A Review. *15th International Conference on Electronics, Computer and Computation*, https://doi:10.1109/icecco48375.2019.9043

Akhtar, D.R., & Inaam, M.D. (2016). *Research Design*: Research in Social Science: Interdisciplinary Perspectives.

Akinyede, R. O., & Esese, O. A. (2017). Development of a Secure Mobile E-Banking System. *International Journal of Computer*, vol.26 (1), pp 23-42

Alabaichi, A., Ali, M.A., Al-Dabbas, K., & Salih, A. (2020). Image Steganography using Least Significant Bit and Secret Map Techniques. *International Journal of Electrical and Computer Engineering*, vol.10 (1), pp. 935-946.

Aleroud, A., & Zhou, L. (2017). Phishing Environments, Techniques, and

Countermeasures. *A Survey of Computer & Security*, vol 68, pp. 160-196

Al-Halabi, Y.S. (2020). A Symmetric Key-Based Steganography Calculation for Anchored Information. *Journal of Theoretical and Applied Information Technology,* vol. 98(1), pp. 103-123

Alaca, F., & van Oorschot, P.C. (2016). Device Fingerprinting for Augmenting Web Authentication: Classification and Analysis of Methods. *32nd Annual Conference on Computer Security Applications.* New York, NY, USA: ACM

Alancy, M., Alomrani, R., Alqarni, B., & Almutairi, S. (2023). Image Steganography Using LSB and Hybrid Encryption Algorithms. *Applied Sciences*, 13 https://doi.org/10.3390/app132111771

Alexan, W., Hamza, A., & Medhat, H. (2019). An AES Double-Layer based Message Security Scheme. *2019 International Conference on Innovative Trends in Computer Engineering, Aswan, Egypt*, pp. 86-91

Ali, G., Dida, M., & Sam, A.E. (2020). Evaluation of Key Security Issues Associated With Mobile Money Systems in Uganda. *Information*, vol 11(6), pp. 1-24

Alharbi, F., Chang, J., Zhou, Y., Qian, F., Qian, Z., & Abu-Ghazaleh, N. (2019). Collaborative Client-Side DNS Cache Poisoning Attack. *2019-IEEE Conference on Computer Communications,* pp.1153–1161.

Alibadi, S.H., & Sadkhan, S.B. (2018). A Proposed Security Evaluation Method for Bluetooth E0 Based on Fuzzy Logic, *2018 International Conference on Advanced Science and Engineering (ICOASE)*, Duhok, Iraq, 2018, pp. 324-329, Doi: 10.1109/ICOASE.2018.8548918.

Al-Khater, W.A., Al-Maadeed, S., Ahmed, A.A., Saliq, A.S., & Khan, M.K. (2020).

Comprehensive Review of Cybercrime Detection Techniques. *IEEE Access,* vol. 8, pp. 137293-137311

Al-Marghilani, A. (2021). Comprehensive Analysis of IOT Malware Evasion Techniques *Engineering, Technology & Applied Sciences Research*, vol, 11(4), pp. 7495-7 500

Almajed, H.N., A. Almogren, S., & Altameem, A. (2019). A Resilient Smart Body Sensor Network through Pyramid Interconnection, *IEEE Access*, vol. 7, pp. 51039–51046.

Al-Rikabi, H. T., & Hazim, H. T. (2021). Enhanced Data Security of Communication System Using Combined Encryption and Steganography. *International Journal of Interactive Mobile Technologies*, 15(16), pp. 144–157.

Alsaawy, Y., Abi Sen, A.A., Alkhodre, A.B., Bahbouh, N.M., Baghanim, N.A., & Alharbi, H.B. (2021). Double Steganography - New Algorithm for More Security. *2021 8th International Conference on Computing for Sustainable Global Development*, pp. 370-374.

Alleaume, R., Branciard, C., Bouda, J., Debuisschert, T., Dianati, M., Gisin, N., Godfrey, M., Grangier, P., Langer, T., Lutkenhaus, N., Monyk, S., Painchault, P., Peev, M., Poppe, A., Pornin, T., Rarity, J., Renner, R., Ribordy, R., Riguidel, M., Salvail, L., Shields, A., Weinfurter, H., Zeilinger, A. (2014). Using Quantum Key Distribution for Cryptographic Purposes: A survey, *Theory of Computer Science*, vol. 560 (1), pp. 62–81.

Al-Amri. R.M., Hamood, D.N., Farhan, A.K. (2023). Image Steganography Based on Chaotic Function and Randomize Function. *Iraqi Journal of Computer Science*

*And Mathematics*, vol. 4(1), pp. 71-86.

Al-Azzeh, J., Zahran, B., Alqadi, Z. (2018). Salt and Pepper Noise: Effects and Removal, *International Journal on Informatics Visualization*, vol.2 (4), pp. 252-256.

Al-Farawn, A., Rjeib, H. D., Ali, N. S., & Al-Sadawi, B. (2020). Secured E-payment System Based on Automated Authentication Data and Iterated Salted Hash Algorithm. *Telecommunication, Computing, Electronics and Control*, vol .18(1), pp. 538-55.

Alharbi, F., Zhou, Y., Qian, Z., & Abu-Ghazaleh, N. (2022). DNS Poisoning of Operating System Caches: Attacks and Mitigations. *IEEE Transactions on Dependable and Secure Computing*, vol 19(4), pp. 2851-2863

Alhawamleh, A.M.K. (2012). Web-based English Placement Test System. Ph.D. Dissertation, Universiti Utara Malaysia

Alia, M. A., & Yahya, A. A. (2010). Public – Key Steganography Based on Matching Method, *European Journal of Scientific Research*, vol. 40(2), pp. 223–231

Ali, G., Dida, M.A., & and Sam, A.E. (2020). Evaluation of Key Security Issues Associated with Mobile Money Systems in Uganda, Information, vol. 11(6), p. 309.

Almomani, A.B.B., Gupta, S., Atawneh, A., Meulenberg, & Almomani, E. (2013). A Survey of phishing Email Filtering Techniques. *IEEE Communications Surveys And Tutorials*, vol 15(4), pp. 2070-2090

Al-Omari, Z.Y., & Al-Taani, A.T. (2017). Secure LSB steganography for Colored Images using Character-color Mapping. *8th International Conference on Information and Communication Systems pp*. 104- 110.

Altwairqi, F., AlZain, M.A., Soh, B., Masud, M., & Al-Amri, J. (2019). Four Most Famous Cyber Attacks for Financial Gains, *International Journal of Engineering and Advanced Technology*, vol. 9, pp. 2131–2139

Almuhammadi, S., & Al-Shaaby, A.  (2017). A survey on Recent Approaches Combining Cryptography and Steganography, *Computer Science Information Technology*, pp. 63-74, Doi: 10.5121/csit.2017.70306

Al-Riyami, A., Zhang, N., &Keane, J. (2016). Impact of hash value truncation on ID anonymity in WSN. Ad Hoc Networks, 45, 80-103.

Al-Shaaby, T. (2017). Cryptography and Steganography: New Approach Transactions on Networks and Communications 5 (6).

Al-Shabi, M.A. (2019). A Survey on Symmetric and Asymmetric Cryptography Algorithms in Information Security. *International Journal of Scientific and Research Publications*, vol. 9(3), pp. 576-589

Al-Shaaby, A., & AlKharobi, T. (2017). Cryptography and Steganography: New Approach. *Transactions on Networks and Communications*, vol.  5(6), pp. 25-38, Doi: 10.14738/tnc.56.3914

Amhed, A., & Naeem, M. (2022). Analysis of most common Encryption Algorithms. *International Journal of Engineering and Applied Computer Science*, vol.4 (2), pp. 8-13

Ambhire, V.R., & Teltumde, P.S. (2011). Information Security in Banking and Financial Industry. *International Journal of Computational Engineering and Management*, vol. 14, pp. 101-105

Amirtharajan, R., & Rayappan, J. (2013). Steganography time to time: A review, *Research Journal of Information Technology,* vol. 5, pp. 58-66

Amrita, K.M., Gupta, N., & Mishra, R. (2018). An Overview of Cryptanalysis on AES. *International Journal of Advance Research and Engineering*, vol. 7(1), pp. 638-646

Anada, H., Yasuda, T., Kawamoto, J., Weng, J., & Sakurai, K. (2019). RSA Public Key With Inside Structure: Proofs of Key Generation and Identities for Web-of-Trust. *Journal of Information Security*, vol 45, pp. 10-19.

Andersson, O. (2013). Slowloris. Retrieved July 6, 2022 from https://github.com /Ogglas/ Orignal-Slowloris-HTTP-DoS/blob/master/slowloris.pl.

Andre, P., Wild, M., Smith, K., & Markmann, T. (2018). Use of Cryptographic Hash Functions in XMPP.

AndroBugs (2015). AndroBugs. https://github.com/AndroBugs

Anu, B., & Koilakuntla, M. (2014). Hiding Secret Information using LSB-based Audio Steganography. *2014 International Conference on Soft Computing and Machine Intelligence*, IEEE, pp.56-59

Anudini, A., Gayamini, G., & Weerawardane, T. (2022). Cloud Data Security System using Cryptography and Steganography: A Review. *International Journal of Scientific and Research Publications*, vol 12(9), pp. 275-281.

Anushka, X.K. (2020). Cryptography used in WhatsApp. *Science Open*. https://www.scienceopen.com/hosted-document?doi=10.14293/S2199-1006.1.SOR-. PPV1NF8.v1

Anwar, N.B., Hasan, M., Hasan, M., Loren, J.Z., & Hossain, S.M.J. (2019). Comparison

Study of Cryptography Algorithms and Its Applications. *International Journal of Computer Networks and Communication Security*, vol.7 (5), pp. 96-103.

Apache (2019). Apache Core Features. Retrieved on July, 2022 at 9.48 AM from https://httpd.apache.org/docs/2.4/mod/core.html.

Arachchilage, N.A.G., Love, S., & Beznosov, K. (2016). Phishing Threat Avoidance Behavior. An Empirical Investigation. *Computers in Human Behavior*, vol. 60, pp. 185-197

Arora, A., Singh, M.P., Thakral, P., & Jarwal, N. (2016). Image Steganography using Enhanced LSB Substitution Technique. *In proceedings of the 4th International Conference on Parallel Distributed and Grid Computing*, Waknaghat, pp. 386-389

Arnott, D., & Pervan, G. (2014). A critical Analysis of Decision Support Systems Research Revisited: The Rise of Design Science. *Journal of Information Technology*, vol. 29(14), pp. 269– 293.

Arshah, R.A., Hammood, W.A. & Kamaludin, A.  (2018). An Integrated Flood Warning. and Response Model for Effective Flood Disaster Mitigation Management. *Advanced Science Letters*, vol. 24(10): p. 7819- 7823

Arya, S., & Malhotra, M. (2016). Effective AES Implementation. *International Journal Of Electronics and communication Engineering and Technology*, vol 7(1), pp. 1-9

Arya, R. P., Mishra, U., Bansa, A., & Email, W. S. (2013). A Survey on Recent Cryptographic Hash Function Designs. *International Journal of Emerging* Technologies in Computer Science, vol. 2(1), pp 1-6

Arya, A., & Soni, S. (2018). Performance Evaluation on Secret Image Steganography Techniques using Least Significant Bit Methods. *International Journal of Computer Science Trends and Technology* vol. 6(2), pp. 160-166

Astuti, Y. P., Setiadi, D. R. I. M., Rachmawanto, E. H., & Sari, C. A. (2018). Simple and Secure Image Steganography using LSB and triple XOR operation on MSB. *International Conference on Information and Communications Technology,* ICOIACT, 2018

Athidas, G., & Alamelu, K. (2018). Security Issues in Mobile Banking. Shanlex *International Journal of Management*, vol. 6(1), pp. 6-10.

Atmowardoyo, H. (2019). Research Methods in TEFL Studies. Descriptive Research, CaseStudy, Error Analysis and R&D. *Journal of Language Teaching and Research*, vol. 9(1), pp. 197-204

Aung, P.P., & Naing, T.M (2014). A Novel Secure Combination Technique of Steganography and Cryptography. *International Journal of Information Technology, Modeling and Computing*, vol. 2, pp. 55-62.

Avinash, S. (2015). SSL Stripping for Newbies, retrieved on 31/12/2022 on 10.45 am from https://www.linkedin.com/pulse/ssl-stripping-newbies-avinash-sm

Avornyo, P., Fang, J., Opoku Antwi, C., Aboagye, M. O., & Boadi, E. A. (2019). Are customers still with us? The influence of optimum stimulation level and IT-specific traits on mobile banking discontinuous usage intentions. *Journal of Retailing and Consumer Services*, vol. 47, pp. 348–360.

Awad, A.T. (2018). Introduction to Information Security Foundations and Applications. Research Gate, https:www.researchgate.net/publications/325170901

Aye, A.M. (2018). LSB Based Image Steganography for Information Security System. *International Journal of Trend in Scientific Research and Development*, vol 3(1), pp. 394-400

Ayyoub, B.Z.B., Nader, J., & Al-Qadi, Z. (2019). Suggested Method to Create Color Image Features Vector. *Journal of Engineering and Applied Sciences* vol. 14(1), pp. 2203-2207.

Ayushi, A. (2010). A Symmetric Key Cryptographic Algorithm. *International Journal of Computer Application*, vol. 1(15), pp. 1-4

Bahtiyar, S., Gür, G. (2014). Altay, L. Security Assessment of Payment Systems under PCI DSS Incompatibilities. *IFIP International Information Security Conference*; Springer: Berlin/Heidelberg, Germany, pp. 395–402.

Bandekar, P.P., & Suguna, G.C. (2021). LSB Based Text and Image Steganography using AES Algorithm. *Proceedings of the International Conference on Communication and Electronics System*, pp 782-788

Barhoom, T. S., & Mousa, S. M. A. (2015). A Steganography LSB Technique for Hiding Image within Image Using Blowfish Encryption Algorithm. *International Journal of Research in Engineering and Science*, vol. 3(3), pp. 61-66.

Basavala, S.R., Kumar, N., & Agarrwal, A. (2012). Authentication: An Overview, its Types and Integration with Web and Mobile Applications. *IEEE International Conference on Parallel, Distributed and Grid Computing.* doi:10.1109/pdgc.2012.6449853

Bellas, J., Koraus, M., Kombo, F., & Koraus, A. (2016). Electronic Banking Security and Customers Satisfaction in Commercial Banks. *Journal of Security and*

*Sustainability Issues*, vol 5(3), pp. 412-422.

Belazi, A., Abd El-Latif, A.A., Diaconu, A-V., Rhouma, R., Belghith, S. (2017). Chaos-based partial image encryption scheme based on linear fractional and lifting wavelet transforms, *Optical Lasers Engineering*, vol.88, pp. 37–50.

Beno, R., & Poet, R. (2020). Hacking Passwords that Satisfy Common Password Policies. *13th International Conference on Security of Information and Networks*, pp. 1-3

Beza, T. (2018). Secure Mobile Banking Framework by using Cryptography and Steganography Methods. *Global Strategy Journal*, vol. 6(8), pp. 863-882.

Bhadauria, R., Sanyal, S. (2012). Survey on Security Issues in Cloud Computing and Associated Mitigation Techniques. *International Journal of Computer Applications*, vol. 47(18), pp.47-66.

Bhanot, R., & Hans, R. (2015). A Review and Comparative Analysis of Various Encryption Algorithms. International Journal of Security and its Applications, vol 9(4), pp. 289-306.

Bhargava, S., & Mukhija, M. (2019). Hide Image and Text Using LSB, DWT and RSA Based on Image Steganography. *ICTACT Journal on Image and Video Processing*, pp.1940-1946

Bhasker, D. (2019). Staying Ahead of the Race-Quantum Computing and Cybersecurity. Vol 7(1), retrieved from on 26/7/2022 on 20.58 PM from https://csiac.org/articles/staying-ahead-of-the-race-quantum-computing-and-cybersecurity/

Bhateja, N., Sikka, S., & Malhotra, A. (2021). A Review of SQL Injection Attack and

Various Detection Approaches. *Smart and Sustainable Intelligent Systems*, pp.481-489 Bilal, M., & Kang, S.-G. (2017). A secure key agreement protocol for dynamic group, *Cluster Computing*, vol. 20 (3), pp. 2779–2792.

Bhattacharya, I., & Reddy, P.S. (2022). Packet Sniffer. *Journal of Engineering Sciences,* vol. 13(6), pp. 204-211

Binny, A., & Koilakuntla, M. (2014). Hiding Secret Information using LSB Based Audio Steganography. *2014 International Conference on Soft Computing and Machine Learning Intelligence*, doi: 10.1109/ISCMI.2014.24

Biswajita, D., Pal, P.K., & Bandyopadhyay, S.K. (2016). Multi-bit Data Hiding in Randomly Chosen LSB Layers of an Audio. *2016 International Conference on Information Technology,* pp. 283-287, doi:10.1109/ICIT.2016.063

Biswas, C., Gupta U. D., & Haque, M. M. (2019). An Efficient Algorithm for Confidentiality, Integrity and Authentication Using Hybrid Cryptography and Steganography. *International Conference on Electrical, Computer and Communication Engineering*, pp. 1-5. doi:10.1109/ECACE.2019.8679136.

Bogos, C-E, Mocanu, R., & Simion, E. (2023). A Security Analysis Comparison between Signal, WhatsApp, and Telegram. https://eprint.iacr.org/2023/071.pdf

Bojjagani, S., & Sastry, V. N. (2017). A Secure End-to-End SMS-based Mobile Banking Protocol. *International Journal of Communication Systems*, vol. 30(15)

Bojjagani, S., & Sastry, V. N. (2017). VAPTAi: A Threat Model for Vulnerability Assessment and Penetration Testing of Android and Ios Mobile Banking Apps. 3rd International Conference on Collaboration and Internet Computing, pp. 77-86

Bonde, S.Y., & Bhadade, U.S. (2017). Analysis of encryption algorithms (RSA, SRNN and 2 key pair) for information security. *International Conference on Computing, Communication, Control and Automation.*

Boonkrong, S. (2021). Methods and Threats of Authentication, in Authentication and Access Control. *Springer*, pp. 45-70

Boureanu I., Owesarski P., Vaudenay S. (2014) Applied Cryptography and Network Security. ACNS 2014. Lecture Notes in Computer Science, vol 8479. Springer, Cham

Boussif, M. (2022). Scalable Implementation of Array of 8-bit-Based RSA with Large Key Size. *5th International Conference on Advanced Systems Emergent Technologies*, pp. 375-380.

Brandt, M., Dai, T., Klein, A., Shulman, H., & Waidner, M. (2018). Domain Validation for MiM-resilient PKI. *In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2060–2076.

Bualsauod, E. H., &Asem, M. O. (2019). A Study on the Effects of Online Banking Quality Gaps on Customers Perception in Saudi Arabia. *Journal of King Saud University Engineering Sciences*

Bunder, M., Nitaj, A., Susilo, W., & Tonien, J. (2017). A Generalized Attack on RSA Type Cryptosystems. *Theories in Computer Science*, vol. 704, pp. 74–81.

Bush, R., & Austein. (2013). The Resource Public Key Infrastructure to Router Protocol. RFC 6810. Retrieved July 22, 2022 from https://tools.ietf.org/html/rfc6810.

Business Daily (2020). JKUAT Students Charged with KES 24.4 million bank Theft.

https://www.businessdailyafrica.com/bd/economy/jkuat-students-sh24-4-million-bank-theft-2722874.

Butt, M.A.; Ajmal Z.; Khan, Z.I.; Idrees, M.; Javed, Y. An In-Depth Survey of Bypassing Buffer Overflow Mitigation Techniques. Appl. Sci. 2022, 12, 6702. Retrieved from https://doi.org/_10.3390/app12136702

Caliwag, J.A., Pagaduan, R.A., Castillo, R.E., & Ramos, W.V.J. (2019). Integrating the Escaping Technique in Preventing Cross Site Scripting in an Online Inventory System. *Proceedings of the 2nd International Conference on Information Science and Systems*, pp. 110-114. https://doi.org/10.1145/3322645.3322696

Calzavara, S., Rabitti, A., & Bugliesi, M. (2018). Semantics-Based Analysis of Content Security Policy Deployment. ACM Transactions on the Web, 12, Article No, 10. https://doi.org/10.1145/3149408

Cambiaso, E., Papaleo, G., & Aiello, M.  (2019). Slowcomm: Design, Development and performance evaluation of a new slow DoS attack. *Journal of Information Security and Applications*, vol. 3, pp. 23–31.

Cambiaso, E., Papaleo, G., Chiola, G., & Aiello, M. (2015). Designing and modeling the slow next DoS attack. In Proceedings of the International Joint Conference on Computational Intelligence in Security for Information Systems pp. 249–259.

Camillo, M. (2017).  Cybersecurity: Risks and Management of Risks for Global Banks and Financial Institutions. *Journal of Risk Management in Financial Institutions*, vol. 10 (2): pp. 196- 200.

Carranza, R., Díaz, E., Sánchez-Camacho, C., & Martín-Consuegra, D. (2021). E-

Banking Adoption: An opportunity for Customer Value Co-creation. *Frontiers in Psychology*, 11, 4003.

Castle, S., Pervaiz, F., Weld, G., Roesner, F., & Anderson, R. (2016). Let's Talk Money: Evaluating the Security Challenges of Mobile Money in the Developing World. *In The Proceedings of the 7th annual symposium on Computing for Development*, ACM 4, pp. 1-10.

Cavus, N., Mohammed, Y.B., & Isah, M.L. (2023). Examining User Verification Scheme Safety and Secrecy Issues Affecting Mobile Banking. Systematic Literature Review. *Sage Journals*, vol 3 (1), https://doi.org/101177/2182440231152379

Chaimaa, B., Najib, E., & Rachid, H. (2021). E-banking Overview: Concepts, Challenges and Solutions. *Wireless Personal Communications*, vol.  117(2), pp. 1059-1078

Central Bank Kenya, Kenya National Bureau of Statistics & FSD Kenya (2019).

Chaimaa, B., Najib, E., & Rachid, H. (2021). E-banking Overview: Concepts, Challenges and Solutions. *Wireless Pers Communication*, vol. (117), pp. 1059–1078.

Chande, M.K., Lee, C.C., Li, C-T. (2018). Cryptanalysis and improvement of an ECDLP based proxy blind signature scheme, *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 21(1), pp. 23-34.

Chandra, S. (2014). A Study and Analysis on Symmetric Cryptography. *International* Conference on Science, Engineering and Management Research, pp. 1-8, doi:10.1109/icsemr.2014.7043664

Chauhan, S., Jyotsna, Kumar, J., & Doegar, A. (2017). Multiple Layer Text Security using Variable Block Size Cryptography and Image Steganography, *3$^{rd}$ International Conference on Computational Intelligence & Communication Technology,* pp. 1-7, Doi: 10.1109/CIACT.2017.7977303.

Cheddad, A., Condell, J., Curran, K., & Mc-Kevitt, P. (2010). Digital Image Steganography: Survey and Analysis of Current Methods. *Signal processing*, vol. (90), pp. 727-752.

Chen, L., Yan, Z., Zhang, W., & Kantole, R. (2015b). TruSMS: a trustworthy SMS spam control system based on trust management. *Future Generation Computer Systems*, 49, 77–93.

Chen, H. C., Nshimiyimana, A., Damarjati, C., & Chang, P. H. (2021). Detection and Prevention of Cross-Site Scripting Attack with Combined Approaches. *International Conference on Electronics, Information, and Communication*, pp. 1-4. https://doi.org/10.1109/ICEIC51217.2021.9369796

Cheng, M. G., & Guo, R. (2010). Analysis and Research on HTTPS Hijacking Attacks. *2$^{nd}$ International Conference on Networks Security, Wireless Communications and Trusted Computing*, IEEE, pp. 223-226.

Chikouche, S. L., & Chikouche, N. (2017). An Improved Approach for LSB-Based Image Steganography using AES Algorithm. *The 5$^{th}$ International Conference on* Electrical Engineering-Boumerdes, Algeria

Chhikara, S., & Kumar, R. (2020). An Information Theoretic Image Steganalysis for LSB Steganography. *Acta Cybernetics*, vol. 24, pp. 593-612

Chiola, G., Cambiaso, E., & Aiello, M. (2019). Introducing the SlowDrop attack. *Computer Networks*, vol. 150, pp. 234–249.

Chowdhury, A.R., Mahmud, J., Kamal, A.R.M., Hamid, Md. A & Member. (2018). MAES: Modified Advanced Encryption Standard for Resource Constraint Environment. *IEEE Sensors Applications*, pp. 1-6

Chung, T., van Rijswijk-Deij, R., Chandrasekaran, B., Choffnes, D., Levin, D., Maggs, B.M., Mislove, A., & Wilson, C. (2017). A Longitudinal, End-to-End View of the DNSSEC Ecosystem. *26th USENIX Security Symposium Association*, pp. 1307–1322.

Citi (2018). Mobile Banking One of Top Three Most Used Apps by Americans. Retrieved from https://www.citigroup.com/citi/news/2018/180426a.htm on 11/1/2023 at 10.51 am

C Insights (2015). Mobile Banking Security: Challenges, Solutions. USA, Report.

Cleveland, C.E. (2016). A Study on how Mobile Banking has Affected Banking Industry: Has Mobile Banking Improved Bank Performance? Honors Theses, 228

Common Lounge (2020). Retrieved from https://www.commonlounge.com/ discussion/d95616beecc 148daaa23f35178691c35

Conti, M., Dragoni, N., & Lesyk, V. (2016). A Survey of Man in the Middle Attacks. *IEEE Communications Surveys & Tutorials*, vol 18(3), pp. 2027-2051

Costantino, G., La Marra, F.A., Martinelli, F., & Matteucci, I. (2018). CANDY: A Social Engineering Attack to Leak Information from Infotainment System, *2018 IEEE 87th Vehicular Technology Conference* pp. 1-5, Doi: 10.1109/VTCSpring. 2018.8417879.

Cover, T.M., & Thomas, J.A. (2012). *Elements of Information Theory*. John Willy & Sons, NJ, USA.

Curguz, J. (2016). Vulnerabilities of SSL/TLS Protocol. *Computer Science and Information Technology*, pp. 245-256, DOI:10.5121/csit.2016.60620

Curran, K., & Devitt, J. (2008). Image Analysis for Online Dynamic Steganography Detection, *Computer and Information Science*, vol. 1, pp. 32

Czimer, J., & Kiszely, R. (2020). Dos and dents' of Immediate Payment Implementation-The Hungarian Story. *Economy and Finance*, vol. 7(3), pp. 280-293 Doi: 103308/EF.2020.3.2

Daemen, J., & Rijmen, V. (1999). AES Proposal: Rijndael Document Version 2. AES Algorithm Submission. Retrieved on 21/12/2022 at 10.33 am, from https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf.

Dalkilic, M.E., & Gungor, C. (2000). An Interactive Cryptanalysis Algorithm for the Vigenere Cipher, Lecture Notes in Computer Science January 2000, retrieved from https://www.researchgate.net/publication/221

Damghani, H., & Hosseinian, H. (2017). Overview of High Dynamic Range Technology (HDR): Capabilities; Benefits; Application and Standards in Television Receiver, 14th Media Technology Conference on, Islamic Republic of Iran Broadcasting, Tehran, Iran, December 2017.

Dantas, Y.G., Nigam, V., & Fonseca, I.E. (2014). A selective defense for application layer DDoS attacks. *In Proceedings of the Joint Intelligence and Security Informatics Conference*, pp. 75–82.

Darwis, D., Junaidi, A., Shofiana, D.S., & Wamiliana (2021). A New Digital Image Steganography based on Center Embedded Pixel Positioning. *Cybernetics and*

*Information Technology*, vol. 21(2), pp. 89-104

Das, D. (2022). An Efficient Light-Weight LSB Steganography with Deel Learning Steganography. https://arxiv.org/ftp/arxiv/papers/2211/2211.08680.pdf

Das, R., & Tuithung, T. (2012). A Novel Steganography Method for Image Based on Huffman Encoding. *2012 3rd National Conference on Emerging Trends and Applications in Computer Science*, pp. 14-18.

Dasgupta, D., Roy, & A., Nag, (2017). A. multi-factor authentication, in Advances in User Authentication, *Springer, Cham, Switzerland*, pp. 185–233,

Date, S., Waghmare, A., Sharma, N., & Chavan, S. (2017). USSD-Based Universal Application. *International Journal of Scientific Research in Computer Science Engineering and Information Technology*, vol 2(2), pp. 692

Demilie, W.B., & Deriba, F.G. (2022). Detection and Prevention of SQLI Attacks and Developing Comprehensive Framework using Machine Learning and Hybrid Techniques. *Journal of Big Data*, vol 9 (124), pp. 2-30

Dhaief, Z. S., Maryoosh, A. A., & Ali, R. (2020). Hiding Encrypted Text in Image using Least Significant Bit Image Steganography Technique. *International Journal of Engineering Research and Advanced Technology*, vol. 6(8), pp 63-75.

Dhamija, A., & Dhaka, V. (2015). A Novel Cryptographic and Steganographic Approach for Secure Cloud Data Migration in Green Computing and Internet of Things. *2015 International Conference on Green Computing and Internet of Things*, pp. 346-351, IEEE.

Deshmukh, P. (2016). An Image Encryption and Decryption using AES algorithm, *International Journal of Scientific& Engineering Research*, vol 7(2), pp. 210-213.

Devi, A.S. (2015). Performance analysis of Symmetric Key Algorithms: DES, AES. *International Journal of Engineering and Computer Science*, vol (6), pp. 12646-12651,

Digital Transformation Cyber Security News (2016). Tesco Bank Fined £16.4m after Hackers siphoned £2.26m from customers in 2016**.** Retrieved on 31/12/2022 https://www.thedrum.com/news/2018/10/01/tesco-bank-fined-164m-after-hackers-siphoned-226m-customers-2016

Dsniff (2022). Retrieved on 31/12/2022 from http://monkey.org/∼dugsong/dsniff/

Domain, W. T. I. S. (2018). A Review and Open Issues of Diverse Text Watermarking Techniques in Spatial Journal of Theoretical and Applied Information Technology 96 17.

Doukas, N., Stavroulakis, P., & Bardis, N. (2021). Review of Artificial Intelligence Cyber Threat Assessment Techniques for Increased System Survivability, in Malware Analysis Using Artificial Intelligence and Deep Learning. *Springer*, pp. 207–222.

Dresch A., Lacerda D. P., Antunes, J. A. V. (2015a). *Design science research* Springer. pp. 67–102.

Durey, A., Laperdrix, P., Rudametkin, W., & Rouvoy, W. (2021). FP-redemption: Studying Browser Fingerprinting Adoption for the sake of Web Security. Detection of Intrusions and Malware, and Vulnerability Assessment. Cham: *Springer International Publishing*, pp. 237–257.

Ecer, F. (2018). An Integrated Fuzzy AHP and ARAS mode to Evaluate Mobile Banking

Services. *Technological and Economic Development Economy*, vol. 24(2), pp. 670-695. Doi: 10.3848/20294913.2016.1255275.

Ee, S.J., Tien, M., Jeshua, W., Yap, J. S., Lee, S. C. Y., Tuz, Z. F. (2020). Active and Passive Security Attacks in Wireless Networks and Prevention Techniques. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.12972857.v1

Ekawade, S., Mule, S., & Patkar, U. (2019). Phishing Attacks and Its Preventions. *Imperial Journal of Interdisciplinary Research*, vol. 4(2), pp. 24-48

El-Abbadi, N. E., Al-Zubaidi, E. A., & Razzaq, H. S. (2020). Image Quality Assessment Tools. *Journal of Xi'an University of Architecture and Technology*, Vol. 12(3), pp. 1260-1276

Elkhodr, M. (2015). Keyloggers. *Digital Investigation* vol. 2(1), pp. 1–6

Ellavarason, E., Guest, R., & Deravi, F. (2018). A framework for Assessing Factors Influencing user Interaction for Touch-based Biometrics. *26th European Signal Processing Conference*, pp. 553-557

Elshazly, E.A., Safey, A., Abdelwahab, S., Fikry, R.M., Elaraby, S.M., Zahram, O., & El-kordy, M. (2016). FPGR Implementation of Robust Image Steganography Technique based on LSB in Spatial Domain. *International Journal of Computer Applications*, vol. 145(12), pp. 43-52

EMBASB (2022). European Mobile Banking Apps White Paper. Retrieved from https://www.kartensicherheit.de/media/banking_mobile_apps_-_white _paper_-_eshard.pdf

Encryption Consulting (2024). What is Blowfish in Security? Who uses Blowfish?

Retrieved from ttps://www.encryptionconsulting.com/education-center/what-is-blowfish/

Essa, R.J., Abdullah, N.A.Z, Al-Dabbagh, R.D.D. (2018). Steganography Technique Using Generic Algorithm. *Iraqi Journal of Science*, vol. 59(3A), pp. 1312-1325

Ettercap (2022). ARP poisoning tool, accessed on 2/1/2023 at 7.00 am from http://ettercap.sourceforge.net/

Etienne, E. (2018). Elementary Statistical Methods of Cryptography, Master's Thesis.

Fan, M., Liu, J., Wang, W., Li, H., Tian, Z., Liu, T. (2017). DAPASA: Detecting Android piggybacked Apps through Sensitive Subgraph Analysis. *IEEE Transactions on Information Forensics & Security*. doi:10.1109/TIFS.2017.2687880.

Fang, W., Chen, W., Zhang, W., Pei, J., Gao, W., & Wang, G. (2020). Digital Signature Scheme for Information Non-repudiation in Blockchain: A State-of-the-Art Review. *EURASIP Journal on Wireless Communications and Networking*, vol. (1), pp. 1-15.

Farham, H., & Alwan, Z. A. (2018). Improved Method using a two XOR to binary Image In RGB Color Image Steganography. *International Journal of Engineering and Technology,* vol. 7(4), pp 4296-4299.

Farley, R., & Wang, X. (2012). VoIP shield: A transparent protection of deployed VoIP systems from SIP-based exploits. *Proceedings of the 2012 Network Operations and Management Symposium*, pp. 486–489.

Farooqi, S., Feal, A., Lauinger, T., MaCoy, D., Shafiq, Z., & Vallina-Rodriguez, N. (2020). Understanding Incentivized Mobile Application Installs on Google Play Store. *Proceedings of the ACM Internet Measurement Conference.*

Fauzan, M.A., & Paulus, E. (2018). A Framework to ensure Data Integrity and Safety. *Journal of Computing and Applied Informatics*, vol. 1(2), pp. 1–12

Felt, A.P., & Wagner, D. (2011). Phishing on Mobile Devices. *IEEE Workshop on Web 2.0 Security and Privacy, San Francisco*, CA, USA

FinMark Trust (2007). Mobile Banking Technology Options. An Overview of the Different Mobile Banking Technology Options, and their Impact on the Mobile Banking Industry, pp. 33

FIPS 197 (2022). Advanced Encryption Standard (AES). Computer Security https://doi.org/10.6028/NIST.FIPS.197.upd1.ipd

Foroughi, B., Iranmanesh, M. & Hyun, S.S. (2019). Understanding the Determinants of Mobile Banking Continuance Usage Intention. *Journal of Enterprise Information Management*, vol. 32(6), pp. 1015– 1033.

Forouzan, B.A. (2007). *Cryptography and Network Security*. Special Edition, Tata MacGraw-Hill Publishing Company Limited, New Delhi

Foozy, C.F.M., Ahmad, R., Abdollah, M.F., Yusof, R., Mas'ud, M.Z. (2011). Generic taxonomy of social engineering attack and defense mechanism for handheld computer study. *In Proceedings of the Malaysian Technical Universities International Conference on Engineering and Technology*, Batu Pahat, Malaysia, 13–15 November 2011; pp. 1–6.

Fraihat, A. (2020). Computer Networking Layers Based on the OSI Model. *The Mattingly Publishing Co., Inc*, vol 83, pp. 6485-6495

Fridrich, J., & Goljan, M. (2002). Practical Steganalysis of Digital Images: State of the art. Security and Watermarking of Multimedia Contents IV, *International Society for Optics and Photonics*, vol.4675, pp. 1–13.

Gambhir, A., & Mishra, A. R. (2015). A New Data Hiding Technique with Multilayer Security System.

Ganesh, R.S., Nagaraj, V., Sivakumar, S.A., & Shankar, B.M. (2020). An Intelligent and Hybrid Method of Combining Spatial Domain and Frequency Representation for Digital Image Steganography. *3rd International Conference on Intelligent Sustainable Systems.* Doi: 10.1109/icss49785.2020.9315918, pp. 1057-1061

Gaur, N., A. Mehra., & Kumar, P. (2018). Enhanced AES Architecture using Extended Set ALU at 28nm FPGA. *5th International Conference on Signal Processing and Integrated Networks* (SPIN).

Gauravram, P. (2003). Cryptographic Hash Functions: Cryptanalysis, design and Applications. Ph.D. thesis, Brisbane, Australia: Faculty of Information Technology, Queensland University of Technology.

Gbo (2021). The Rise of Mobile Banking in the UK- Global Business Outlook https://www.globalbusinessOutlook.com/the-rise-of-mobile-banking-in-the-uk

Geetha, R., Padmavanthy, T., Thilagam, T., & Lallithasree, A. (2019). Tamillian Cryptography Efficient Hybrid Symmetric Key Encryption Algorithm. *Wireless Personal Communications*. Doi: 10.1007/s 11277-019-070136

Geerts, G.L. (2011). A Design Science Research Methodology and its Application to Accounting Information Systems Research. *International Journal of Accounting Information Systems,* vol 12(2), pp. 142-151

GitHub. (2017). DHCPing. Retrieved July 22, 2022 at 9.52 AM from https://github.com/kamorin/DHCPig.

GitHub. (2017). dns-flood-ng. Retrieved July 22, 2022 at 10 AM from https://github.com/cmosek/dns-flood-ng.

GitHub. (2018). HTTP Unbearable Load King. Retrieved July 22, 2022 at 11.00 AM from https://github.com/grafov/hulk

GitHub. 2019. Low Orbit Ion Cannon. Retrieved July, 22, 2022 from https://github.com/NewEraCracker/LOIC.

Ghafir, I. (2016). Social engineering attack strategies and defense approaches. *In Proceedings of the IEEE International Conference on Future Internet of Things and Cloud*, Vienna, Austria, 22–24 August 2016; pp. 1–5.

Ghali, Z. (2021). Motives of Customers' E-loyalty Towards E-banking Services: A Study in Saudi Arabia. *Journal of Decision Systems*, pp. 1-22

Granjal, J., Monteiro, E., Silva, J-S. (15). Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues, *IEEE Communications Surveys & Tutorials*, vol. 17(3), pp. 1294 – 1312.

Goel. A., Sharma, D.K., & Gupta, K.D. (2022). Leobat: Lightweight Encryption and OTP Based Authentication Technique for Securing IOT Networks. *Expert Systems*, Vol. 39(5), p. 788

Golait, D., & Hubballi, N. (2017). Detecting anomalous behavior in VoIP systems: A discrete event system modeling. *IEEE Transactions on Information Forensics and Security*, vol. 12 (3), pp. 730–745.

Gonzalez, H., Gosselin-Lavigne, M.A., Stakhanova, N., & Ghorbani, A.A. (2015). The

Impact of Application-layer Denial-of-Service Attacks. In Case Studies in Secure Computing: Achievements and Trends. CRC Press, Boca Raton, FL, 261–272

Gordon, S., & Ford, R. (2006). On the Definition and Classification of Cybercrime. *Journal in Computer Virology*, vol 2(1), pp. 13-20

Green, J., Levstein, I., Boggs, C.R.J., & Fenger, T. (2015). Steganography Analysis: Efficacy and Response-Time of Current Steganalysis Software, *Journal of Computer Science*, vol. 9, pp. 236-44.

Gu, Y.Q., He, C., Liu, F.G., & Ye, J. (2021). Raman Ink for Steganography. *Adv. Optical Mater*, vol. 9, https://doi.org/10.1002/adom.202002038

Gulshan, K. (2016). Denial of service attacks: An updated perspective. *Systems Science & Control Engineering*, Vol. 4(1), pp. 285–294.

Guo, C., Campbell, B., Kapadia, A., Reiter, M. K., & Caine, K. (2021). Effect of Mood, Location, Trust, and Presence of Others on Video-Based Social Authentication. In *30th USENIX Security Symposium*

Gupta, S., & Gupta, B.B. (2018). XSS-Secure as a Service for the Platforms of Online Social Network-Based Multimedia Web Applications in Cloud. *Multimedia Tools and Applications*, vol. 77, pp. 4829-4861.

Gupta, S., & Gupta, B.B. (2016). XSS-SAFE: A Server-Side Approach to Detect and Mitigate Cross-Site Scripting (XSS) Attacks in JavaScript Code. *Arabian Journal for Science and Engineering*, vol. 41, pp. 897-920. https://doi.org/10.1007/s13369-015-1891-7

Gupta, K., Silakari, S. (2012). Novel approach for Fast Compressed Hybrid Color Image Cryptosystem *Advanced Engineering Software*, vol. 49, pp. 29–42.

Gupta, B.B., Gupta, S., & Chaudhary, P. (2017). Enhancing the Browser-Side Context-Aware Sanitization of Suspicious HTML5 Code for Halting the DOM-Based XSS Vulnerabilities in Cloud. *International Journal of Cloud Applications and Computing*, vol. 7, pp. 1-31. https://doi.org/10.4018/IJCAC.2017010101

Gupta, S., & Sharma, J. (2012). A Hybrid Algorithm based on RSA and Diffie-Hellman. *International Conference on Computational Intelligence and Computing Research* (ICCIC), pp. 1-4

Gupta, B.B., Tewari, A., Jain, A.K., & Agrawal, D.P. (2017). Fighting against Phishing Attacks. State of the Art and Future Challenges, *Neural Computing and Applications*, vol. 28(12), pp. 3629-3654

Gutub, A., & Al-shaarani, F. (2020). Efficient Implementation of Multi-image Secret Hiding-based on LSB and DWT Steganography Comparisons, *Arabia Journal For Science Engineering*, vol 45, pp. 2631-2644

Gwahula, R. (2016). Risks and Barriers Associated with Mobile Money Transactions in Tanzania. Business Management and Strategy, vol. 7(2), pp. 121-139

Haddaji, R. (2016). Comparison of Digital Signature Algorithm and Authentication Schemes for H.264 Compressed Video. *Int. J. Adv. Comp. Sci App*, vol. 7(9) pp. 357-363.

Hanif, Y., & Lallie, H.S. (2021). Security Factors on the Intention to use Mobile Banking Applications in the UK Older Generation. A Mixed Method Study using Modified UTAUT and MTAM- with Perceived Cyber Security Risk and Trust. *Technology And Society*, vol 67, 101693, https://doi.org/10.1016/j.techsoc.2021.101693

Hanis, S., & Amutha, R. (2017). Double image compression and encryption scheme using logistic mapped convolution and cellular automata, *Multimedia Tools Applications*, vol. 77 (6), pp. 6897–6912.

Hari, R.E., Syaiful, A.R., Moses, S.D-R.I., & Atika, S.C. (2017). A Performance Analysis StegoCrypt Algorithm Based on LSB-AES 128-bit in Various Image Sizes. *IEEE International Seminar on Application for Technology of Information and Communication* Semarang, Indonesia, doi:10.1109/ISEMANTIC.2017.8251836

Harinath, K., Raja, A., Suneel, V., Muzafer, S., & Rajesh, K. (2024). Multi-Format Steganography in Network Security. *International Journal of Advanced Research in Computer and Communication Engineering*, vol 13(4), pp. 814-821.

Hasham, S., Joshi, S., & Mikkelsen, D. (2019). Financial crime and fraud in the age of cybersecurity. *McKinsey & Company*, p. 1-11.

Hashim, MM., Rahim, M.S.M., Johi, F.A., Taha, M.S., & Hamad, H.S. (2018). Performance Evaluation Measurement of Image Steganography Techniques with Analysis of LSB based on Variation Image Formats. *International Journal of Engineering and Technology*, vol. 7(4), pp. 3505-3514.

Hassan, M.A., & Shukur, Z. (2021). A Secure Multi-Factor User Authentication Framework for Electronic Payment System. In Proceedings of the 3rd International Cyber Resilience Conference (CRC), Langkawi Island, Malaysia, 29–31 January 2021; pp. 1–6.

Hayikader, S., Hadi, F.N.H.B.A., & Ibrahim, J. (2016). Issues and Security Measures of Mobile Banking Apps. *International Journal of Scientific and Research Publications*, vol 6(1), pp. 36-41.

Heidari S, Farzadnia E (2017) A Novel Quantum LSB-based Steganography Method using the Gray Code for Colored Quantum Images. *Quantum Information Processing*, vol.16 (10), pp. 242

Herxberg, A. (2022). Foundations of Applied Cryptography and Cybersecurity. Accessed ttps://www.researchgate.net/publication/323243320_Foundations_of_Applied_Cryptography_and_Cybersecurity, pp. 1-643 on 23/12/2022 at 9.18 pm

Hoffman, P., Sullivan, A., &Fujiwara, K. (2019). DNS Terminology, Technical Report. Retrieved from: https://tools.iet;org/html/rfc8489 on 31/12/2022

Hossain, M.A., & Ahmed, F., (2014). Evaluating the Impact of Mobile Banking Deployment for Microfinance, *University of Dhaka Journal of Marketing,* vol. 2012 15, pp. 144–157.

Hossain, D., Paul, A., & Islam, H. (2018). Survey of the Protection Mechanisms to the SSL-based Session Hijacking Attacks. *Network Protocols and Algorithms,* Vol. 10(1), pp. 83-108

Haseeb, K., Islam, N., Almogren, A., Din, I.U., Almajed, H.N., & Guizani, N. (2019). Secret Sharing-based Energy-aware and Multi-hop Routing Protocol for IoT based WSNs. *IEEE Access*, vol. 7, pp. 79980–79988.

Hosseinian, H., Damghani, H., Damghani, L., Nezam, G., & Hosseinian, H. (2019). Home appliances energy management based on the IoT system, *International Journal of Nonlinear Analysis and Applications*, vol. 10(1), pp. 168-175

Hosseinian, H., Damghani, H., Damghani, L., Kouhi, E., & Kouhi, M. (2019). Asia's Cities: The way to Going Smart, *International Journal of Nonlinear Analysis and Applications*, vol. 10(2), pp. 142-152

Hsiao, F-H. (2017). Applying elliptic curve cryptography to a chaotic synchronization System: neural-network-based approach, *International Journal of Systems Science,* vol. 48 (14), pp.  3044-3059.

Huang, YF., & Tang, S. (2016). Covert Voice over Internet Protocol Communications based on Spatial Model. *Science China Technological Sciences*, vol. 59(1), pp. 117-127.

Hubballi N., & Tripathi, N. (2017). A Closer Look into DHCP Starvation Attack in Wireless Networks. *Computers & Security*, vol. 65, pp. 387–404.

Hubballi, N., & Tripathi, N. (2017). An event-based technique for detecting spoofed IP packets. Journal of Information Security and Applications vol. 35, pp. 32–43.

Hussain, M., Wahab, A. W. A., Idris, Y. I. B., Ho, A. T., & Jung, K. H. (2018). Image Steganography in Spatial Domain: A survey Signal Processing: *Image Communication* 65 46-66.

Hussain, I., Djahel, S., Zhang, Z., & Nait-Abdesselam, F. (2015). A comprehensive study of flooding attack consequences and countermeasures in session initiation protocol SIP. *Security and Communication Networks*, vol. 8(18), pp. 4436–4451.

Hussaina, M., Wahaba, A.W.A., Idris, Y.I.B., Antony, T.S., Jung, K.H. (2018). Image Steganography in Spatial Domain: A Survey. *Signal Processing: Image Communication,* vol. 65, pp. 46-66

Hsieh, M. (2018). The USC-SIPI Image Database. Retrieved from USC University of Southern California: http://sipi.usc.edu/database/

Ibrahim, A. (2007). Steganalysis in Computer Forensics, *in Australian Digital Forensics* Conference, pp. 10.

Internet Systems Consortium. 2020. Bind 9. Retrieved on 31/12/2022 at 20.37 pm from https://www.isc.org/bind/.

Islam, M.R., Siddiqa, A., Uddin, M.P., Mandal, A.K., & Hossain, M.D. (2014). An Efficient Filtering Based Approach Improving LSB Image Steganography using Status Bit along with AES Cryptography, *International Conference on Informatics, Electronics and Vision*, pp. 1-6

ISO/IEC 27002:2013. Information Technology-Security Techniques-Code of Practice for Information Security Controls,
available at: http://www.iso27001security.com/html/27002.html

Jahan, I., Asif, M., Rozario, L.J. (2015). Improved RSA cryptosystem based on the study of number theory and public key cryptosystems. American Journal of Engineering Research, vol 4(1), pp. 143-149

Jain, P., & Kanwal, N. (2016). Image Steganography in RGB Color Components using Improved LSB Technique Image Pattern Compression using Weighted Principal Components Algorithm. *Indian journal of science and technology*, vol. 9, pp. 1-4

James, M., Kumar, D.S., & Scholar, P.G. (2016). An Optimized Parallel Mix Column and Sub bytes Design in Lightweight Advanced Encryption Standard. *International Journal of Computational Engineering Research*, vol 6(3), pp. 25-28

Jan, A., Parah, S.A., Hussan, M., & Malik, B.A. (2022). Double Layer Security using Crypto-stego Techniques: A Comprehensive Review. *Health Technol*, vol. 12, pp. 9-31. https://doi.org/10.1007/s12553-021-00602-1

Jani, H.B. (2015). Latest Side Channel Attacks and its Countermeasures Attacks: Attacks Based on Cryptography. International Journal of Computer Science and

Information Technology Research, vol 3(1), pp. 427-441

Javeed, D., Badamasi, U.M., Ndubuisi, C.O., Soomro, F., & Asif, M. (2020). Man in the Middle Attacks: Analysis, Motivation and Prevention: *International Journal of Computer Networks and Computing Security*, vol. 8(7), pp 52-57

Jawad, L.M., Sulong, G. (2015). Chaotic map-embedded Blowfish algorithm for security Enhancement of color image encryption, *Nonlinear Dynamics*, vol. 81 (4), pp. 2079–2093.

Jazi, H.H., Gonzalez, H., Stakhanova, N., & Ghorbani, A.A. (2017). Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling. *Computer Networks,* vol. 121, pp. 25–36.

Jiang, F., Fu, Y., Gupta, B.B., Liang, Y., Rho, S., & Lou, F. (2020). Deep Learning Based Multi-Channel Intelligent Attack Detection for Data Security. *IEEE Transactions on Sustainable Computing*, vol. 5, pp. 204-212.

Jibril, A. B., Kwarteng, M. A., Chovancova, M., & Denanyoh, R. (2020). Customers' Perception of cybersecurity threats toward E-banking Adoption and Retention: A conceptual study. *In International Conference on Cyber Warfare and Security*, pp. 270.

Jin, X., Hu, X., Ying, K., Du, W., Yin, H., Peri, G.N. (2014). Code Injection Attacks on html5-based Mobile applications: Characterization, Detection and Mitigation. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale,* AZ, USA, 3–7 November 2014; pp. 66–77.

JKUAT students charged with hacking bank, stealing millions (2021, July 1). Business Daily

Jøsang, A. (2017, September). A Consistent Definition of Authorization. *In International Workshop on Security and Trust Management*, pp. 134-144

Józef, L., Mazurczyk, W., & Szczypiorski, K. (2014). Principles and Overview of Network Steganography. *IEEE Communications Magazine*, vol 52(5), pp. 225-229

Joshi, K. (2018). A New Approach of Text Steganography using ASCII Values. *International Journal of Engineering Research and Technology*, vol 7(5), pp. 490-493.

JScape (2022). What is AES Encryption? How does it Work? Retrieved from: https://www.jscape.com/blog/aes-encryption on 9/12/2022 at 8.49 am

Juniper Research (2019). Mobile Banking Users to Reach 2 billion by 2020, Representing More than 1 in 3 of Global Adult Population. Retrieved on 11/1/2023 at 10.53 am from https://www.juniperresearch.com/press/digital-banking-users-to-reach-2-billion.

Kadhum, R.N., & Ali, N.H.M. (2022). Using Steganography Techniques for Implicit Authentication to Enhance Sensitive Data Hiding. *International Journal of Non-*Linear Analysis Applications, vol. 13(1), pp. 3973-3983.

Kadhim, I. J., Premaratne, P., Vial, P. J., & Halloran, B. (2019). Comprehensive survey of Image steganography: Techniques, Evaluations, and trends in future research. *Neurocomputing,* vol. 335, pp. 299-326.

Kahate, A. (2011). *Cryptography and Network Security*, 2nd Edition, Tata Mc-Grew-Hill Publisher Ltd.

Kak, A. (2019). AES: The Advanced Encryption Standard. Pardue University, pp. 11-20

Kaka, J. G., Ishaq, O. O., & Ojeniyi, J. O. (2020). Recognition-based Graphical Password Algorithms: A Survey. IEEE 2nd International Conference on Cyberspace, pp. 44–51.

Kaka, S., Sastry, V.N., & Maiti, R.R. (2017). On the MitM Vulnerability in Mobile Banking Applications for Android Devices, *2016 International Conference on Advanced Networks and Telecommunications Systems.* Accessed on 16/7/2022 at 16.04 PM from https://dl.acm.org/doi/abs/10.1109/ANTS.2016.7947811

Kakadel, R. B., & Veshne, N. (2017). Unified Payment Interface. A Way Towards Cashless Economy. International Research Journal of Engineering and Technology, vol. 4, pp. 762-766

Kamal. R. (2020). *Mobile Computing*, 2nd Ed, Oxford University Press, Idia

Kang, J. (2018). Mobile payment in Fintech environment: Trends, security challenges, and services. Human Centered Computer Information Science, vol.8 (32).

Kang, K., Pang, Z., Da Xu, L., Ma, L., Wang, C. (2014). An interactive trust model for Application Market of the Internet of Things. *IEEE Transactions on Industrial Informatics*, vol. 10 (2), pp. 1516-1526.

Kanta, A., Coray, S., Coisel, I., & Scanlon, M. (2021). Cracking in Digital Forensic Investigation? Analysing the Guesability of Over 3.9 billion real-world accounts. *Forensic Science International Digital Investigation*, vol. 37, pp. 301186

Karthikeyan, B., Kosaraju, A.C., & Gupta S, S. (2016). Enhanced Security in Steganography using Encryption and Quick Response Code. *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, pp. 2308-2312.

Karthikeyan, T., Vijayarajan, V., & Manikandan, K. (2011). Secure Mobile Banking Application using Elliptic Curve Cryptography and RGB Intensity Based Randomized Pixels with Variable Bits Image Steganography. *International Journal of Advanced Research in Computer Science*, vol 2(1), pp. 387-390.

Kaur, S., Bansal, S., & Bansal, R.K (2014). Steganography and Classification of Image Steganography Techniques. *International Conference on Computing for Sustainable Global Development,* New Delhi, India, Doi: 10.1109/India.com.2014.6828087

Kaur, G., Pande, B., Bhardwaj, A., Bhagat, G., & Gupta, S. (2018). Efficient Yet Robust Elimination of XSS Attack Vectors from HTML5 Web Applications Hosted on OSN-Based Cloud Platforms. *Procedia Computer Science*, vol. 125, pp. 669-675. https://doi.org/10.1016/j.procs.2017.12.086

Kaur, H., & Kakkar, A. (2017). Comparison of Different Image Formats using LSB Steganography. *4th International Conference on Signal Processing Computing and Control*, pp. 97-101

Kaur, M., Raj, M., Lee, H.N. (2022). Cross Channel Scripting and Code Injection Attacks on Web and Cloud-Based Applications: A Comprehensive Review. *Sensors*, 22, 1959.

Kavitha, K. (2015). Mobile Banking Supervising System-Issues, Challenges and Suggestions to improve Mobile Banking Services. *Advances in Computer Science: An International Journal,* vol. 4, pp. 65-67.

Kelley, S. (2020). Dnsmasq-network services for small networks, Retrieved on 31/12/2022 from: https://www.thekelleys.org.uk/dnsmasq/doc.html

Kenya Revenue Authority lost $39 million to hacker (2017, March 2022). BBC News

Keoh, S., Kumar, S., Tschofenig, H., 2014. Securing the Internet of Things: A Standardization Perspective, *IEEE Internet of Things Journal*, pp. 2-12

Khelifi, A. (2013). Enhancing Protection Techniques of E-Banking Security Services Using Open-Source Cryptographic Algorithms. 14th *International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pp. 89-95. DOI 10.1109/SNPD.2013.47

Khelifi, A., Aburrous, M., Talib, M. A., & Shastry, P. V. S. (2013). Enhancing Protection Techniques of E-Banking Security Services Using Open-Source Cryptographic Algorithms. *14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*. IEEE, DI: 10.1109/SNDP.2013.47

Khidzir, N.Z., Daud, K.A.M., Ismail, A.R., Ghani, M.S.A.A., & Ibrahim, M.A.H. (2018). Information Security Requirement: The Relationship between Cybersecurity Risk Confidentiality, Integrity and Availability in Digital Social Media. *In Regional Conference on Science, Technology and Social Sciences*, Springer, pp. 229-237

Kiat, L.S., Obaja, M.A., Wei, L., & Hui, O.C. (2017). Malware Text DB: A Database for Annotated Malware Articles. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 1557-1567, Doi: 10.18653/v1/P17-114

Kieseberg, P., Fruhwirt, P., Schrittwieser, S., & Weippl, E. (2015). Security Tests for

Mobile Applications Why using tls/ssl is not Enough. *In IEEE Eighth International Conference on Software Testing, Verification and Valid in Section VIIation Workshops*, pp. 1–2

Kirsten, S. (2016) Cross Site Scripting (XSS) Software Attack. https://owasp.org/www-community/attacks/xss

Kizza, J.M. (2020). Access Control and Authorization. Guide to Computer Network Security Texts in Computer Science, *Springer International Publishing*, Cham pp. 187-206

Koblitz, N. (1987). Elliptic Curve Cryptosystems, *Mathematics of computation*, vol. 48 (177), pp. 203–209

Kolla, A. (2017). List of 10 Best Steganography Tools to Hide Data, Geek Dashboard, Available: https://www.geekdashboard.com/best-steganography-tools/.

Kourouma, M.K., Warren, R.P., Atkins-Ball, D.S., Jackson, L., Gwee, N., Trivedi, S.K., & Breaux T. (2022). Investigating Wireless and Internet of Things Technologies Security Threats and Attacks. *International Journal of Wireless and Mobile Networks*, vol. 14(3), pp. 1-19

Krishna, A.V.N., & Babu, A.V. (2010). Role of Statistical Tests in Estimation of the Security of a New Encryption Algorithm, *International Journal of Advancements in Technology*, vol. 1(1), pp. 13-25

Krishnamurthy, G.N., Ramaswamy, V. (2009). Encryption Quality Analysis and Security Evaluation of CAST-128 Algorithm and its Modified Version using Digital Images, *International Journal of Network Security & Its Applications*, vol.1(1)

Krombholz, K., Hobel, H., Huber, M., & Weippl, E. (2015). Advanced Social

Engineering attacks. *Journal of Information Security*, vol (22), pp. 113–122.

Kruse, L.C., Seidel, S., & Purao, S. (2016). Making Use of Design Principles, *Proceedings of the 11th International Conference on Tackling Society's Grand Challenges with Design Science*, vol. 1961, pp. 37–51.

Kumar, M. (2015). A Report on Digital Signature. Retrieved on 23/10/2022 from htttps://www.academia.edu/24747203/Seminar_Report_On_DIGITAL_SIGNATURE

Kumar, P.R., Raj, P. H., & Jelciana, P. (2017). Exploring Data Security Issues and Solutions in Cloud Computing. *6th International Conference on Smart Computing And Communications*, ICSCC 2017, 7-8, India

Kumar, R. H., Kumar, P. H., Sudeepa, K. B., & Aithal, G. (2013). Enhanced Security System using Symmetric Encryption and Visual Cryptography**.** *International Journal of Advances in Engineering & Technology*; vol. 6 (3), pp.1211-1219.

Kumar, V., Pathak, V., & Badal, N. (2022). Complex Entropy based Encryption and Decryption Technique for Securing Medical Images. *Multimedia Tools Appl*, vol 8, pp. 37441-37459

Kumar, M., & Singh, G. (2017). Block-based Image Steganography using Entropy with LSB and 2-bit Identical Approach. International Journal of Computer *Applications*, vol. 171(8), pp. 12-15

Kumar, N., Thakur, J., Kalia, A. (2011). Performance Analysis of Symmetric Key Cryptography Algorithms: DES, AES and Blowfish. *International Journal of Engineering Sciences*, vol.4, pp.28-37.

Kumar, V., Kumar, R., Pandey, S. (2018). A computationally efficient centralized group key distribution protocol for secure multicast communications based upon RSA public key cryptosystem, *Computer Information Science*, pp. 1–14

Kundu, R., & Dutta, A. (2020). Cryptographic Hash Functions and Attacks- A Detailed Study. International Journal of Advanced Research in Computer Science. Available Online at www.ijarcs.info. Doi: http://dx.doi.org/10.26483/ijarcs .v11i2.6508

Kunhoth, J., Subramanian, N., Al-Maadeed, S., & Bouridane, A. (2023). Video Steganography: Recent Advances and Challenges. *Multimedia Tools and Applications*, vol 82, pp. 41943-41985

Lakshmi, S.B., Srinives, S., Kumar, Chandra, M.B. (2016). Steganography based Image Sharing with Reversibility. *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 19(1), pp. 67-80

Lala, S.K., Kumar, A., & Subbulakshmi, T. (2021) Secure Web Development Using OWASP Guidelines. *5th International Conference on Intelligent Computing and Control Systems Madurai*, pp. 323-332. https://doi.org/10.1109/ICICCS51141 .2021.9432179

Lamprecht, C., & Guetterman, T. (2019). Mixed methods in accounting: A Field-based analysis. *Meditari Accountancy Research*, vol.27 (6), pp.921-938.

La-Polla, M., Martinelli, F., Sgandurra, D. (2013). A Survey on Security for Mobile Devices, *in Communications Surveys & Tutorials, IEEE*, vol. 15(1), pp. 446-471

Le, G., Jingqiang, L., Bo, L., Jiwu, J., & Jing, W. (2015). Protecting Private Keys against Memory Disclosure Attacks using Hardware Transactional Memory. *In 2015 IEEE Symposium on Security and Privacy*, San Jose, California.

Leonov, P. Y., Vorobyev, A. V., Ezhova, A. A., Kotelyanets, O. S., Zavalishina, A. K., & Morozov, N. V. (2021). The Main Social Engineering Techniques Aimed at Hacking Information Systems. *Ural Symposium on Biomedical Engineering, Radio electronics and Information Technology* (USBEREIT). IEEE 10.1109/ USBEREIT51232.2021.9455031

Li, Y., Wang, C., & Chen, H. (2017). A hyper-chaos-based image encryption algorithm using pixel-level permutation and bit-level permutation, *Opt. Lasers Eng*, vol. 90, pp.  238–246.

Libicki, M. (2018). Could the Issue of DPRK Hacking Benefit from Benign Neglect? *Georgetown Journal of International Affairs* 19, pp. 83-89. doi:10.1353/gia.2018.0010.

Liliya, R.A., Evgeny, K.A., Igor, B.O., & Stanislav, V.S. (2017). Increasing the Lifetime of Symmetric Keys for the GCM Mode by Internal Re-keying," IACR Cryptology ePrint Archive 2017.

Lin, X., Sun, L., & Qu, H. (2018). An Efficient RSA-based Certificateless Public Key Encryption Scheme. *Discrete Applied Mathematics*, vol. 241(2018), pp. 39-47

Liu, Z., Huang, X., Hu, Z., Khan, M.K., Jeong Seo, H.W.A., & Zhou, L. (2017). On Emerging Family of Elliptic Curves to Secure Internet of Things: ECC Comes of Age, *IEEE Transactions on Dependable and Secure Computing*, vol. 14 (3), pp 237-248.

Liu Z., Wenger E., Großschädl, J. (2014) MoTE-ECC: *Energy-Scalable Elliptic Curve Cryptography for Wireless Sensor Networks*, vol 8479, pp. 362-379.

Lucks, S. (2004). Design Principles for Iterated Hash Functions.

https://eprint.iacr.org/2004/253

Lundgren, B., & Möller, N. (2017). Defining Information Security. *Science and Engineering Ethics,* doi:10.1007/s11948-017-9992-1

Lupu, C., Gaitan, V.G., & Lupu, V. (2015). Security enhancement of internet banking applications by using multimodal biometrics. *IEEE 13th International Symposium on Applied Machine Intelligence and Informatics*.

Luvanda, A. (2014). Proposed Framework for Securing Mobile Banking Applications from Man in the Middle Attacks. *Journal of Information Engineering and Applications,* vol. 4(12), pp. 20–28

Luy, E., Karatas, Z. Y., & Ergin, H, (2016). Comment on ‘An Enhanced and Secured RSA Key Generation Scheme. *Journal of Information Security Application,* vol. 30, pp. 1–2

Macharia. K. W. (2021). Cryptographic Hash Functions. Retrieved on 26/10/2022 from https://www.researchgate.net/publication/351837904_Cryptographic_Hash_Functions

Maetouq, A., Daud, S. M., Ahmad, N.A., Maarop, N., Sjarif, N. N. A., & Abas, H. (2018). Comparison of Hash Function Algorithms against Attacks: A Review. *International Journal of Advanced Computer Science and Applications* vol 9(8), pp. 98-103

Mahd, S.A., & Khodher, M.A. (2021). An Improved Method for Combining LSB and MSB Based on Color Image RGB. *Engineering and Technology Journal,* vol 39 (1), pp. 231-242.

Majjed, L.O.A. (2023). Image Types and Formats. Retrieved from https://www.slideshare.net/ssuserff72e4/lecture-22023pdf-253271077

Malaquias, R.F., & Silva, A.F. (2020). Understanding the use of Mobile Banking in Rural Areas of Brazil. Technology in Society, 62, 101260

Malhotra, A., Cohen, I.E., Brakke, E., & Goldberg, S. (2016). Attacking the network time protocol. In Proceedings of the Network and Distributed System Security Symposium (NDSS'16).

Malhotra, A., & Goldberg, S. (2016). Attacking NTP's authenticated broadcast mode. Computer Communication Review vol. 46(2), pp. 12–17.

Malhotra, A., Van Gundy, M., Varia, M., Kennedy, H., Gardner, J., & Goldberg, S. (2017). The security of NTP's datagram protocol. *In Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 405–423.

Mali, K., Chakraborty, S.H., & Roy, M. (2015). A Study on Statistical Analysis and Security Evaluation Parameters, *International Journal of Scientific and Engineering Research*, vol. 3(8), pp. 339-343.

Mandal, S., & Singh, A.K.  (2021). Journal of Emerging Technologies and Innovative Research Manikandan, V.M., & Masilamani, V. (2019). A Novel Reversible Data Hiding Scheme that Provides Image Encryption. *Journal of Image and Graphics*, Vol. 6(1), pp. 64-68.

Marashdeh, Z., Suwais, K., & Alia, M. (2021). A Survey on SQL Injection Attack: Detection and Challenges. *International Conference on Information Technology*, pp. 957-962

Marashdih, W., Zaaba, Z. F., Suwais, K., &. Mohd, N.A. (2019). Web Application

Security: An Investigation on Static Analysis with other Algorithms to Detect Cross Site Scripting, *Procedia Computer Science*, vol. 161, pp. 1173–1181, 2019.

Marella, P., Straub, J., & Benard, B. (2019). Development of a Facial Feature Based Image Steganography Technology. *International Conference on Computational Science and Computational Intelligence,* pp. 675-678

Mastkar, N., Isankar, M., Sheikh, F. R., Singh, D., & Ramteka, S. (2018). Survey Paper on Securing Online Transaction using Cryptography and Steganography. *International Journal of Scientific Research in Science, Engineering and Technology*, vol. 4(4), pp 463-465

Matrouk, K., Al- Hasanat, A., Alasha'ary, H., Al-Qadi, Z., &Val-Shalabi, H. (2019). Analysis of Matrix. *International Journal of Computer Science and Mobile Computing*, vol.8 (3), pp. 76-90.

Meghanathan, N., & Nayak, L. (2010). Steganalysis Algorithms for Detecting the Hidden Information in Image, Audio and Video Cover Media, *International Journal of Network Security & Its application,* vol. 2, pp. 43-55

Meghashree, B.S., & Sujatha, B.R. (2018). AES based Image Encryption and Decryption Using Matlab. *International Journal of Engineering Research and Technology*. vol 6(3), pp. 1-3

Meng, X., & Zheng, X. (2015). Cryptanalysis of RSA with small Parameter Revisited. *Information Process,* Lett 115(11), pp. 858-862

Mhato, D., & Yadav, K. (2017). RSA and ECC. A Comparative Analysis. *International Journal of Engineering and Research*, vol 12(19), pp. 9053-9061

Miah, S.J., & Genemo, H. (2016). A Design Science Research Methodology for Expert

Systems Development. *Australian Journal of Information Systems*, vol 20, pp. 1-29

Miller, V.S. (1985). Use of Elliptic Curves in Cryptography, *Conference on the Theory* And Application of Cryptographic Techniques, pp. 417–426

Mishra, A.K. (2017). Digital Signature: The Need of Cashless Society. CreateSpace Independent Publishing Platform. ISBN-13: 978-1546382539

Mishra, M., Tiwari, G., & Yadav, A. K. (2014). Secret Communication using Public Key Steganography. *In Recent Advances and Innovations in Engineering* (ICRAIE), pp. 1-5, IEEE.

Mizrahi, T. (2012). Slave diversity: Using multiple paths to improve the accuracy of clock synchronization protocols. *In Proceedings of the International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication*, pp 1 - 6.

Mizrahi, T. (2014). Security Requirements of Time Protocols in Packet Switched Packets Retrieved July 17th, 2022 from https://tools.ietf.org/html/rfc7384.

Mitra, S., Jana, B., Bhattacharya, S., Pal, P., & Poray, J. (2017). Quantum cryptography: Overview, security issues and future challenges. *4th International Conference on Opto-Electronics and Applied Optics* doi:10.1109/optronix.2017.8350006

MobSF (2017). Mobile-Security-Framework-MboSF, https://github.com/MobSF/ Mobile-Security-Framework-MobSF

Modube, A.O., Adedoyin, A.E., Titilayo, A.O., & Deborah, F.O. (2021). A Comparative Analysis of LSB, MSB and PVD Based Image Steganography. *International Journal of Research and Review*, vol. 8(9), pp. 373-377.

Moody, G. D., Siponen, M., & Pahnila, S. (2018). Toward a unified model of information Security Policy Compliance, *MIS Quarterly*., 42

Mogos, G., & Jamail, N. S. M. (2021). Study on Security Risks of E-banking System. *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 21(2), pp. 1065-1072.

Mohamed, K. S. (2020). Cryptography Concepts: Integrity, Authentication, Availability, Access Control, and Non-repudiation. *New Frontiers in Cryptography*, pp. 41-63). Springer, Cham.

Mollah, M. B., Azad, M. A. K., Vasilakos, A. (2017). Security and privacy challenges in Mobile cloud computing: Survey and way ahead, *Journal of Network and Computer Applications,* vol. 84, pp. 38–54.

Mohammadi, H. (2015). A Study of Mobile Banking Loyalty in Iran. *Computers in Human Behavior,* vol, 44, pp. 35-47

Mohammed, S. (2016). Secure Hash Design and Implementation Based on MD5 and SHA-1 using Merkle-Damgard Construction. *International Journal of Advanced Research, Computer Science and Technology*, vol 4(2), pp 92-94

Mohamed, K.S. (2020). New Frontiers in Cryptography (Quantum, Blockchain, *Lightweight, Chaotic and DNA)* doi:10.1007/978-3-030-58996-7

Mohammed, A., Argabi, A., & Alam, I. (2019). A new Cryptographic Algorithm AEDS Advanced Encryption and Decryption Standard) for data security. *International Advanced Research Journal of Science Engineering and Technology,* vol. 6(10), pp. 1–7.

Mohd, B.J., Abed, S., & Alouneh, T.A.S. (2012). FPGA Hardware and the LSB

Steganography Method. *International Conference on Computer, Information and Telecommunication System*, pp. 14-16

Molato, M.R.D., & Gerardo, B.D. (2018). Cover Image Selection Technique for Secured LSB-based Image Steganography. *International Conference on Algorithms, Computing & Artificial Intelligence, Association for Computing Machinery, Article* 17, pp. 1-6

Morkel, T. (2012). Image Steganography Applications for Secure Communication. Doctoral dissertation, University of Pretoria.

Mouton, F., Leenen, L., & Venter, H.S. (2016). Social Engineering Attack Detection Model. SEADMv2, *2015 International Conference on Cyberworlds*, pp. 217-223

Moxie Marlinspike's sslstrip (2009). Retrieved on 31/12/2022 on 10.00 am http://www.thoughtcrime.org/ software/sslstrip/mpss. (2018). Myanmar Payment Services.

Mritha, R., & Isa, N.A.M. (2015). Fast Retrieval of Hidden Data using Enhanced Hidden Markov Model in Video Steganography. *Applied Soft Computing*, vol. 34, pp. 744-757.

Msallam, M.M. (2020). A Development of Least Significant Bit Steganography Technique *Iraqi Journal of Computers, Communications, Control and Systems Engineering* Vol, 20(1), pp. 31-39.

Mtaho, A.B. (2015). Improving Mobile Money Security with Two-Factor Authentication. *International Journal of Computer Application*, pp. 109, 9–15.

Muhammad, K., Ahmad, J., Sajjad, M., & Zubair, M. (2015). Secure Image

Steganography using Cryptography and Image Transposition. *ArXiv, abs/1510.04413*.

Muhammad, K., Ahmad, J., Farman, H., & Zubair, M. (2014). A Novel Image Steganographic Approach for Hiding Text in Color Images using HSI Color Model, *Middle-East Journal of Scientific Research*, vol. 22(5), pp. 647-654.

Muhammad, K., Sajjad, M., Mehmood, I., Rho, S., & Baik, S.W. (2016). A Novel Magic LSB Substitution Method (M-LSB-SM) using Multi-level Encryption and Achromatic Component of an Image. *Multimedia Tools Applications*, vol. 75(22), pp. 14867–14893

Muhammad, K., Sajjad, M., Mehmood, I., Rho, S., & Baik, S.W. (2018). Image Steganography using Uncorrelated Color Space and its Application for Security of Visual Contents in Online Social Networks. *Future Generation Computer Systems*, vol. 86, pp. 951–960

Munga, G.N. (2010). The Impact of Mobile Banking. A Case Study of M-PESA in the Kenyan Society. Master’s Thesis, University of Nairobi

Mushenko, A., Zolkin, A., & Yatsumira, A. (2021). Steganography Analysis of Chaotic Carrier Signal Transmission with Non-linear Parametric Modulation. *International Russian Automation Conference IEEE*, pp. 1018-1023 Doi: 10.1109/RusAutoCon52004.2021.9537422.

Muslim, M. A. & Prasetiyo, B. (2015). Data Hiding Security using Bit Matching-based Steganography and Cryptography without Change the Stego Image Quality. *Journal of Theoretical & Applied Information Technology*, pp. 82

Mustafa, C., & Wisam, E. (2018). New LSN-based Color Image Steganography Method to Enhance the efficiency of Payload Capacity, Security and Integrity Check. *Sadhana,*Vol. 43(5), pp. 68, Doi: 10.1007/s 12046-018-0848-4

Nagaraj, K. (2023). TwoFish Encryption: A Comprehensive Guide. Understanding the Key Features Strategy and Weaknesses of TwoFish Encryption. https://cyberWing.medium.com/twofish-encryption-a-comprehensive-guide 2023-b3ad0f844870

Ndatinya, V., & Xiao, Z. (2015). Network Forensic Analysis using Wireshark, *International Journal of Sensor Networks*, vol. 10(2).

Neamah, I. (2015). A Modification of ElGamal Cryptosystem Using Statistical Methods, *European Journal of Scientific Research*, vol. 133(1), pp. 20-25

NerdWallet (2022). Mobile Banking is safe and secure. Here is why. Retrieved from https://campaignlp.constantcontact.com/em/1107988473808/0f210eec-6845-49ad-8418-b2c91407f976 on 25/4/2023 at 9.20 am

Nerwal, B., Mohapatra, A. K., & Usmani, K. A. (2019). Towards a Taxonomy of Cyber Threats against Target Applications. *Journal of Statistics and Management Systems.* Vol 22(2), pp. 301-325. DOI: 10.1080/09720510.2019.1580907

Nie, J., & Hu, X. (2008). Mobile banking information security and protection methods. In: 2008. *International Conference on Computer Science and Software Engineering Mobile*, pp. 587–590

Nie, T., & Zhang, T. (2009). A study of DES and Blowfish encryption algorithm. In TENCON *2009 IEEE Region 10 Conference*, pp. 1-4.

Nikam, N.R., Priyanka, R., Vakhariya, R.R., Mohite, S.K., & Magdum, C.S. (2020). Data

Integrity: An Overview. *International Journal of Recent Scientific Research* Vol 11(6), pp. 38762-38767.

Nikiforakis, N., Younan, Y., & Joosen, W. (2010). HProxy: Client-side detection of SSL stripping attacks, in Detection of Intrusions and Malware, and Vulnerability Assessment, Bonn, Germany: *Springer*, vol. 6201, pp. 200–218.

Nisha, S., & Farik, M. (2017). RSA Public Key Cryptography Algorithm: A Review. *Information Journal of Scientific and Technology Research*, vol 6(7), pp. 187-191

NIST (2022). Advanced Encryption Standard. Computer Security, Cryptography. Federal Information Processing Standards Publications, pp. 1-38 retrieved on 20/12/2022 At 8.30 am from https://doi.org/10.6028/NIST.FIPS.197-upd1.ipd

Nithya, V., Pandian, S.L., & Malarvizhi, C. (2015). A Survey of Detection and Prevention of Cross-site Scripting Attack. *International Journal of Science and its Applications*, vol, 9(3), pp. 139-152

NLnet Labs (2020). Unbound DNS Resolver. Retrieved on 31/12/2022 at 20.40 pm from https://nlnetlabs.nl/projects/unbound/about/

Nosrati, L., & Bidgoli, A. M. (2016). A Review of Different Encryption Algorithms for Security of Mobile Banking. *International Journal of Engineering and Technical Research*, vol. 5(3), pp 119-123.

Nosrati, L., & Bidgoli, A.M. (2016). A Review of Mobile Banking Security, *Proceedings* of IEEE *Canadian Conference on Electrical and Computer Engineering*, Doi: 10.1109/CCECE.2016.7726820

Nugroho, Y.S., Gunawan, D., Puspa P. D. A., Islam, S., & Alhefdhi, A. A. (2022). Study of Vulnerability Identifiers in Code Comments: Source, Purpose, and Severity. *Journal of Communications Software and Systems*, vol. 18(2), pp. 165–174.

Nwoye, C.J. (2015). Design and Development of an E-Commerce Security using RSA Cryptosystem. *International Journal of Innovative Research in Information Security*, vol 2(6), pp. 5-17.

Nyimbili, F., & Nyimbili, L. (2024). Types of Purposive Sampling Techniques with Their Examples and Applications in Qualitative Research Studies. *British Journal of Multidisciplinary and Advanced Studies*, vol 5(1), pp. 90-99

OECD (2020), Digital Disruption in Banking and its Impact on Competition http://www.oecd.org/daf/competition/digital-disruption-in-financial-markets.htm

Ogundokun, R.O., Abikoye, O.C., Misra, S., Awotunde, J.B. (2020). Modified Least Significant Bit Technique for Securing Medical Images. *Lecture Notes in Business Information Processing* 402, pp. 553-565

Okpara, O.S., Bekaroo, G. (2017). Fingerprint-Based Authentication in M-wallets using Embedded Cameras. In Proceedings of the 2017 *IEEE International Conference on Environment and Electrical Engineering* and 2017 IEEE Industrial and Commercial Power Systems Europe, Milan, Italy, 6–9 June; pp. 1–5.

Oluwakemi, A.C., Kayode, A. S., & Ayotunde, O. J. (2012). Efficient Data Hiding System using Cryptography and Steganography, *International Journal of Applied Information Systems,* vol. 4 (11), pp. 6–11

Omar G. A., Elsadd, M. A., & Guirguis, S. K. (2017). Investigation of Cryptography

Algorithms used for Security and Privacy Protection in Smart Grid. *In Power Systems Conference, 2017 Nineteenth International Middle East, IEEE,* 644-649, (December 2017).

Omar, A., Hammood, M.N.M.K., Muamer, N., Mohammed., & Waleed, A. (2017). Hammood Issues and Challenges of Video Dissemination in VANET and Routing Protocol: Review. *Journal of Engineering and Applied Sciences*, vol 12(11), pp. 9266-9277.

Ometov, A., Bezzateev, S., Mäkitalo, N., Andreev, S., Mikkonen, T., Koucheryavy, Y. (2018). Multi-Factor Authentication. A Survey. *Cryptography,* vol. 2(1), pp. 2-31 https://doi.org/10.3390/cryptography2010001

Owusu Kwateng, K., Atiemo, K. A. O., & Appiah, C. (2019). Acceptance and use of mobile banking: An application of UTAUT2. *Journal of Enterprise Information Management*, 32, 118–152.

Padmavathi, B., & Kumari, S. R. (2013). A Survey on Performance Analysis of DES, AES and RSA algorithm along with LSB substitution. *International Journal of Science and Research,* India.

Panda, P. (2016). A Security Analysis of the Top 500 Global E-Commerce Mobile Apps In USA, UK, Australia, Singapore, and India. Technical Report. AppKnox

Panda, P. (2015). Security Report of Top 100 Mobile Banking Apps-APAC. Technical Report. AppKnox

Panja, B., Fattaleh, D., Mercado, M., Robinson, A., & Meharia, P. (2013). Cybersecurity in Banking and Financial Sector: Security Analysis of a Mobile Banking Application. *International Conference on Collaboration Technologies and Systems* (CTS)

Panwar, S., Damani, S., & Kumar, M. (2018). Digital Image Steganography using Modified LSB and AES Cryptography. *International Journal of Recent Engineering Research and Development*, vol.3(6), pp. 18-27.

Papaspirou, V., Maglaras, L., Ferrag, M. A., Kantzavelou, I., Janicke, I., & Douligeris, C. (2021). A novel Two-Factor HoneyToken Authentication Mechanism. *International Conference on Computer Communications and Networks IEEE*, pp. 1–7

Patel, K., Utareja, S., & Gupta, H. (2013). Information Hiding using Least Significant Bit Steganography and Blowfish Algorithm, *International Journal of Computer Applications*, vol. 63(13), pp. 24–28

Patil, R.K. (2020). Investigation on Data Security Threats and Solutions. *International Journal of Innovative Science and Research Technology*, vol.5 (1), pp. 79-8

Patil, S. S., & Goud, S. (2016). Enhanced Multi-Level Secret Data Hiding. *International Journal of Scientific Research in Science, Engineering and Technology*, pp. 846-850

Patil, P., Narayankar, P., Narayan, D.G., & Meena, S.M. (2016). A Comprehensive Evaluation of Cryptographic Algorithms: DES, *Procedia Computer Science* vol. 78, pp. 617–624

Peffers, K., Tuunanen, T., Gengler, C.E., Rossi, M., Hui, W., Virtanen, V., & Bragge, J.

(2020). Design Science Research Process: A Model for Producing and Presenting Information Systems Research. https: doi.org/10.48550/arXiv.2006.02763

Peng, Y., Lao, Y., & Li, P. (2021). Robust Watermarking for Deep Neural Networks via Bi-Level Optimization. Proceedings of the IEEE/CVF *International Conference on Computer Vision*, pp. 14841-14850

Pesante, L. (2017). Introduction to Information Security. Accessed on 6/1/2021 from https://www.us-cert.gov/sites/default/files/publications/infosecuritybasics.pdf, Accessedhttps://www.uscert.gov/sites/default/files/publications/infosecuritybasics .pdf,

Phishingpro (2016). Everyone is a target, http: www.razorthorn.co.uk/wp-content/uploads 2017/01/phishing-stats-2016.pdf

Pillai, B., Mounika, M., Rao, P.J., & Sriram, P. (2016). Image steganography method using K-means clustering and encryption techniques. *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, PP. 1206-1211.

Pirandola, S., Andersen, U. L., Banchi, L., Berta, M., Bunandar, D., Colbeck, R., Englund, D., Gehring, T., Lupo, C., Ottaviani, C., Pereira, J.L., Razavi, M., Shamsul, J., Shaari, Tomamichel, M., Usenko, V. C., Vallone, G., Villoresi, P., & Wallden, P. (2020). *Advances in quantum cryptography Advances in Optics and Photonics*, vol. 12(4), pp. 1012-1236.

Pohlig, P., & Hellman, M. (1978). An Improved Algorithm for Computing Logarithms over GP(p) and Its Cryptographic Significance, *IEEE Transaction on information theory*, vol. 24(1)

Popoutsakis, M., Fysarakis, K., Spanoudakis, G., Ioannidis, S., & Koloutsou, K. (2021). Towards a Collection of Security and Privacy Patterns. *Applied Sciences*, vol 11(4), pp. 2-41

Positive Technologies (2018). Financial Application Vulnerabilities. https://www.ptsecurity.com/ww-en/analytics/financial-application-vulnerabilities

Prandini, M., Ramili, M., Carroni, W., & Callegati, F. (2020). Splitting the HTTPS Stream to attack Secure Web Connections. *IEEE Security and Privacy*, pp. 80-84

Prasad, K.M., Reddy, A.R.M., & Rao, K.V. (2014). DoS and DDoS Attacks: Defense, Detection, and Traceback Mechanisms: A Survey. *Global Journal of Computer Science and Technology*, vol 14(7), pp. 15-32

Praseed, A., & Thilagam, P. S. (2019). DDoS attacks at the application layer: Challenges and research perspectives for safeguarding web applications. *IEEE Communications Surveys & Tutorials*, vol. 21(1), pp. 661–685.

Pradham, A., Sahu, A. K., Swain, g., & Sekhar, K. R. (2016). Performance Evaluation Parameters of Image Steganography Techniques in: *Proceedings of the 2016 International Conference on Research Advances in Integrated Navigation Systems* Doi: 10.1109/RAINS.2016.776499

Prakash, K. (2023). CIA Triad in Cyber Security: Definition, Examples, Importance. https://www.knowledgehut.com/blog/security/cia-in-cyber-security.

Purnama, B., & Rohayani, H. (2015). A New Modified Caesar Cipher Cryptography Method with Legible Ciphertext from a Message to be Encrypted, *Procedia Computer. Science*, vol. 59, pp. 195-204

Purwinarko, A., & Hardyanto, W. (2018). A Hybrid Security Algorithm AES and

Blowfish for Authentication in Mobile Applications. *Scientific Journal of Informatics*, vol 5(1), pp. 76-80.

Putra, D.S.K., Sadikin, M. A., Windarta, S. (2017). Secure Mobile Banking Authentication Scheme using Signcryption, Pair-based Text Authentication, and Contactless Smart Card. *International Conference on Quality in Research: International Symposium on Electrical and Computer Engineering*, pp 230-234

QARK (2017). Tool to look for several security related Android applications vulnerabilities, https://github.com/linkedin/qark

Raharja, P.S.J., & Tresna, R. (2019). Adoption of Information and Communication Technology on Enhancing Business Performance: Study on Creative Industry SMEs in Bandung City, Indonesia. *Review of Integrative Business and Economics Research*, vol. 8 (3), pp. 20–30.

Rajeev, S., & Geetha, G. (2012). Cryptographic Hash Functions: A Review. *International Journal of Computer Science Issues*, vol 9, pp. 461 - 479

Rahav, A. (2018). The Secret Security Wiki. Accessed on 2/1/2023 at 22.04 pm from https://doubleoctopus.com/security-wiki/authentication/single-factor-authentication/

Rajaiah, M., Basha, S.A., Hidayathulla, D., Reddy, V.A., Kusuma, S., & Ahmmad, S.A. (2023). Image Encryption using AES Feature Extraction and Random No. Generation. *European Journal of Molecular and Clinical Medicine*, vol 10(2), pp. 453-460

Rajendran, S., & Doraipandian, M. (2017). Chaotic Map Based Random Image

Steganography using LSB Technique, *International Journal of Network Security*, vol. 19, pp. 593-598

Ramadhan, M.J., & Elleithy, K.M. (2016). A video Steganography Algorithm based on Kanade-Lucas-Tomasi Tracking Algorithm and Error Correcting Codes. *Multimedia Tools and Applications, vol.* 75(17), pp. 10311-10333.

Ramya, K. (2013). Design and Implementation of Digital Signatures. *International Journal of Engineering Research and Technology*, vol 2(2), pp. 1386-1390

Ranjan, S., Swaminathan, R., Uysal, M., Nucci, A., & Knightly, E. (2009). DDoS-Shield: DDoS-resilient Scheduling to Counter Application Layer Attacks. *IEEE/ACM Transaction on Networking*, vol. 17(1), pp. 26–39.

Ravi, S., Joshi, K., & Nandal. R. (2021). An Adapted Approach of Image Steganography Using Pixel Mutation and Bit Augmentation. Smart Computing Techniques and Applications. Springer, Singapore, pp. 217-224.

Rawat, A., & Agrawal, D. (2015). An Enhanced Message Digest Hash Algorithm for Information Security. *International Journal of Recent Research in Electrical and Electronics Engineering*, vol 2(1), pp. 54-62

Raza, M. (2023). Denial-of-Service Attacks in 2023. History Techniques and Prevention. https://www.splunk.com/en_us/blog/learn/dos-denial-of-service-attacks.html.

Reaves, B., Bowers, J., Scaife, N., Bates, A., Bhartiya, A., Traynor, P., Butler, K.R.B. (2017). Mo(bile) money, Mo(bile) problems: Analysis of Branchless Banking Applications. *ACM Transactions on privacy and security*, vol. 20(3), pp. 1–31.

Rehman, T. U. (2021). Cybersecurity for E-Banking and E-Commerce in

Pakistan. *Handbook of Research on Advancing Cybersecurity for Digital Transformation*, pp. 373-403

Rob, S. (2018). Cryptographic Key Management - the Risks and Mitigation, Available: https://www.cryptomathic.com/newsevents/blog/cryptographic-key-management-therisks-and-mitigations.

Roopesh, K., & Kumar, Y.A. (2021). Development of Novel Algorithm for Data Hiding On Mobile Application. *International Journal of Advanced Computer Science Applications*, vol. 12(8), pp. 223-230

Rouse, M. (2017). Single-Factor Authentication. Accessed on 2/1/2023 at 22.02 pm from https://searchsecurity.techtarget.com

Roy, S., & Islam, M.M. (2022). A Hybrid Secured Approach Combining LSB Steganography and AES Using Mosaic Images for Encrypting Data Security. *SN Computer Science*, vol 3(153), DOI: https://doi.org/10.1007/S42979-022-01046-8

Russo, B., Camilli, M., & Mock, M. (2022). Detecting Weak Self-Admitted Technical Debt. Retrieved on 31/12/2022 at 14.20 pm from ttps://arxiv.org/abs/2205.02208

Rwisa, S., Kissaka, M., & Kapis, K. (2020). A Metric for Evaluating Security Models based on Implementation of Public Key Infrastructure. *International Journal of Wireless and Microwave Technologies*, vol. 6, pp. 27-35

Sachin, D., & Gupta, R. (2021). Analysis of Various Data Security Techniques of Steganography. A Survey. *Information Security Journal*: A Global Perspective, vol. 30(2), pp. 63-87

Safia, A.L., Gutub, A., & Ghamdi, M.A. (2019). Enhancing Arabic Text Steganography for Personal usage Utilizing Pseudo Spaces. *Journal of King Saud University Computer and Information Sciences*, pp. 2-12

Saha, B.J., Kabi, K.K., Pradhan, C., & Bisoi, A. (2014). Comparative Study of Image Encryption using 2D Chaotic Map. *IEEE 2nd International Conference on Information Systems and Computer Networks,* doi: 10.1109/iciscon.2014.6965227

Sahin, M., Ünlü, T., Hébert, C., Shepherd, L.A., Coull, N., & Lean, C.M. (2022). Measuring Developers' Web Security Awareness from Attack and Defense Perspectives. IEEE Security and Privacy Workshops (SPW), San Francisco, 22-26 May 2022, 31-43

Sahoo, S.R., & Gupta, B.B. (2019). Classification of Various Attacks and Their Defense Mechanism in Online Social Networks: A Survey. *Enterprise Information Systems*, vol. 13, pp. 832-864. https://doi.org/10.1080/17517575.2019.1605542

Saini, A., & Vandana, D. (2022). A Study on Modified RSA Algorithm in Network Security. International Research Journal of Modernization in Engineering Technology Science, vol 4(4), pp. 1461-1465

Sakr, R.H., Omara, F., & Nomir, O. (2013). A Comparative Study of Security Algorithms for Cloud Computing, *International Journal of Intelligent Computing and Information Science,* vol.13, pp. 73-84

Salahdine, F., & Kaabouch, N. (2019). Social Engineering Attacks: A Survey. Future Internet. Doi:10.3390/fil 104089

Salihu, M.N., Jimoh, A., Salihu, S., & Modi, B. (2024). Enhancing Cybersecurity with Practical Cryptographic Hash Algorithms. *International Conference Technology*

*Innovation and Industrial Management Sustainable Development*, Poland, pp. 732-748

Salim, A., Sagheer, A. M., & Yaseen, L. (2020). Design and Implementation of a Secure Mobile Banking System based on Elliptic Curve Integrated Encryption Schema Springer Nature Switzerland. ACRIT 2019, CCIS 1174 pp 424-438 https://doi.org/10.1007/978-030-78752-5_33

Sang, N.M. (2021). Critical Factors affecting Consumer Intention of using Mobile Banking Applications during CCOVID-19 Pandemic: An Empirical Study from *Vietnam Journal of Asian Finance, Economics and Business*. Vol. 8(11), pp. 157-167.

Sanganagouda, J. (2011). USSD: A Communication Technology to Potentially Ouster SMS Dependency. *International Journal of Research and Reviews in Computer Science,* vol 2(2), pp. 10

Sankpal, L., Rathod, A., Kodre, B., Sayyed, N., & Sayta, R. (2017). Location Based Encryption for Secure Banking Transactions in Mobile Data Environment. *International Journal of Advance Engineering and Research Development* Vol. 4(1), pp 499-503

Sara, U., Akter, M., & Uddin, M.S. (2019). Image Quality Assessment through FSIM, SSIM, MSE and PSNR-A Comparative Review. *Journal of Computer and Communications,* vol. 7(3), pp. 8-18.

Sarfraz, M., Alsoraya, D., AlBathali, A., & Al-Mayyas, O. (2021). An Automated Management System for Student e-Services. *Information Sciences Letters*, vol. 10(1), p. 9.

Sarker, S., Chatterjee, S., Xiao, X., Elbanna, A. (2019). The socio-technical axis of Cohesion for the IS discipline: its historical legacy and its continued relevance, *MIS Q.* 43 (3) pp. 695–719

Sarkar, P.G., & Fitzgerald, S. (2016). Attack on SSL: A Comprehensive study of beast, crime, time, breach, lucky 13 and RC4 biases, San Francisco, CA, isec Partners.

Sarkar, A., & Karforma, S. (2018). Image Steganography using Password-based Encryption Technique to secure E-Banking Data. *International Journal of Applied Engineering Research*, vol. 13(22), pp. 15477-15483

Sattar, B., Sadkhan, S.B., & Reza, D.M. (2017). Investigation of the best structure for the nonlinear combining function, Annual Conference on New Trends in Information & Communications Technology Applications.

Sattar, B., Sadkhan, S.B., & Jawad, S.F. (2020). Security Evaluation of Cryptosystems based on Orthogonal Transformation, *6th International Engineering Conference Sustainable Technology and Development (IEC)*, Erbil, Iraq, 2020, pp. 222-226

Scapy. (2019). Welcome to Scapy's Documentation! Retrieved July 6, 2022 from https://scapy.readthedocs.io/en/ latest/.

Science, C., & Bridgeport, B. (2015). A Novel Video Steganography Algorithm in the Wavelet Domain Based on the KLT Tracking Algorithm and BCIT Codes Doi: 10.1109/IEC49899.2020.9122828

Schneier, B. (1994). The Blowfish Encryption Algorithm. *Dr. Dobb's Journal-Software Tools for Professional Programmer*, vol. 19(4), pp. 38-43.

Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms and Source Code

In C, vol 2, pp. 216-222.

Sealpath (2020). Protecting the three states of data. Data Protection. Retrieved from https://www.sealpath.com/blog/protecting-the-three-states-of-data/

Sedeeq, T. T. (2013). Steganography in Colored Images. *International Journal of Computer Science and Information Security*, vol. 11(4), pp 87-92

SentinelOne (2019). Financial Cyber Threats: 10 Cases of Insider Bank Attacks.

Retrieved from https://www.sentinelone.com/blog/financial-cyber-threats-10-cases-of-insider-bank-attacks/

Seo, J. H. (2020). Efficient Digital Signatures from RSA without Random Oracles *Information Science* 512, pp. 471–480.

Sepehri-Rad, A., Sadjadi, S., & Sadi-Nezhad, S. (2019). An Application of DEMATEL for Transaction authentication in Online Banking. *International Journal of Data and Network Science*, vol. 3(2), pp. 71-76.

Setiadi, D. R. I. M. (2019). Improved Payload Capacity in LSB Image Steganography using Dilated Hybrid Edge Detection, *Journal of King Saud University, Computer and Information Sciences*, pp 1-11.

Setiadi, D.I.M (2021). PSNR vs SSIM: Imperceptibility Quality Assessment for Image Steganography. *Multimedia Tools Appl*, vol 8, pp. 8423-8444

Setiadi, D.R.I.M., & Jumanto, J. (2018). An Enhanced LSB-Image Steganography Using Hybrid Canny-Sobel Edge Detection. *Cyber Information Technology*, vol. 18(2), pp 74-78

Setyaningsih, E., Wardoyo, R., & Sari, A.K. (2020). Securing Color Image Transmission

Using Compression Key Generator and Efficient Symmetric Key Distribution. Digital Communications and Networks. Doi: 10.1016/j.dcan.2020.02.001

Shaanika, I.N. (2022). The use of Mixed Methods as a Research Strategy in Information Systems Studies. Proceedings of the *13th International Conference on Society and Information Technology*, pp. 50-55 https://doi.org/10.54808/ICSIT2020.01.50

Shachi, M., Shourav, N. S., Ahmed, A. S. S., Brishty, A. A., & Sakib, N. A Survey on Detection and Prevention of SQL and NoSQL Injection Attack on Server-side Applications. *International Journal of Computer Applications*, vol. 975, p. 8887.

Shahid, S., Islam, J.U., Malik, S., & Hasan, U. (2022). Examining Consumer Experience In Using Mobile Banking Applications. A Study of its Antecedents and Outcomes. *Journal of Retailing and Consumer Services*, vol 65, pp. 102870.

Shalini, S., & Usha, S. (2011). Prevention of Cross-site Scripting Attacks on Web Applications in the Client-side. *International Journal of Computer Science Issues* Vol. 8(4), pp. 650.

Shaikh, A.A. and Karjaluoto H. (2015). (Eds.), Marketing and mobile financial services: a global perspective on digital banking consumer behavior, Routledge, New York, pp. 73-91

Shailender, G., Ankur, G., & Bharat, B. (2012). Information Hiding Using Least Significant Bit Steganography and Cryptography, *International Journal of Computer Network and Information Security*, vol. 6(27), pp. 27–34

Shallal, Q.M., & Bokhari, M.U. (2016). A Review on Symmetric Key Encryption

Techniques in Cryptography. *International Journal of Computer Application*, vol. 147 (10), pp. 43-48

Shannon, C.E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, vol 27(3), pp. 379-423

Shukla, A. K. (2021). Proposed E-Services Quality Model and its Impact on the Indian. Society. *Turkish Journal of Computer and Mathematics Education*, vol. 12(10), pp. 4231-4242

Shaikh, A.A. and Karjaluoto, H. (2019). *Mobile financial services*, in Shaikh, A.A. and Marketing and mobile financial services: a global perspective on digital banking consumer behavior, Routledge, New York, pp. 1-26.

Shaikh, A. A., & Karjaluoto, H. (2015). Mobile banking adoption: A literature review. *Telematics and Informatics*, vol. 32(1), pp. 129-142.

Shankar, A., & Rishi, B. (2020). Convenience matter in mobile banking adoption intention, *Australasian Marketing Journal*, Elsevier Ltd,

Sharma, K., Agrawal, A., Pandey, D., Khan, R.A., & Dinkar, S.K. (2019). RSA Based Encryption Approach for Preserving Confidentiality of Big Data. *Journal of King Saudi University – Computer and Information Sciences*, vol. 34(2022), pp. 2088-2097. https://doi.org/10.1016/j.jksuci.2019.10.006

Sharma, N., & Bohra, B. (2017). Enhancing Online Banking Authentication using Hybrid Cryptographic method. In Proceedings of the 3rd *International Conference on Computational Intelligence and Communication Technology*, Ghaziabad, India, 24–26 November 2017; pp. 1–8

Sharma, R., Ganotra, R., Dhall, S., & Gupta, S. (2018). Performance Comparison of

Steganography Techniques. *International Journal of Computer Network and Information Security*, vol. 9, pp. 37-46

Sharma, L., & Mathuria, M. (2018). Mobile Banking Transaction using Fingerprint Authentication. 2nd International *Conference on Inventive Systems and Control*, pp. 1300- 1305.

Sharma, H., Mishra, D.C., Sharma, R.K., & Kumar, N. (2021). Multi-image Steganography and Authentication using Crypto-Stego Techniques. *Multimedia Tools and Applications,* vol 80, pp. 29067-29093

Sharma, M. H., MithleshArya, M. & Goyal, M. D. (2013). Secure Image Hiding Algorithm using Cryptography and Steganography. *Journal of Computer Engineering*, vol. 13(5), pp. 1-6

Sharma, M.K., Mohd, N., & Sharma, R. (2014). A New Steganography Technique Based on Difference Scheme of RGB Channels and Text using Histogram Analysis. *International Journal of Engineering Research and Applications,* vol. 4(5), pp. 64-69

Shayganmehr, M., & Montazer, G. A. (2021). A Novel Hybrid Assessment Model to Evaluate E-services Websites of Iranian Municipalities. *Artificial Intelligence Review*, pp. 1-35.

Shekyan, S. (2017). Application Layer DoS attack simulator. Retrieved July 20th, 2022 from https://github.com/ shekyan/slowhttptest.

Shoukat, I.A., Bakar, K.A., & Iftikhar, M. (2011). A Survey about the Latest Trends and Research Issues on Cryptographic Elements. *International Journal of Computer Science Issues*, vol 8(3), pp. 140-149

Silva-Garc´ıa, V., Flores-Carapia, R., Renteria-Marquez, C., Luna-Benoso, B., Aldape-Perez, M. (2018). Substitution box generation using Chaos: An image encryption application, *Application Mathematics Computing*, vol. 332 (2018), pp. 123–135.

Simplilearn (2022). Digital Signature Algorithm (DSA) in Cryptography: How it works and Advantages. Retrieved on 30th July, 2022 at 14.30 form https://www.simplilearn.com/tutorials/cryptography-tutorial/digital-signature-algorithm

Singh, S.J., & Attri, V.K. (2015). Dual Layer Security of Data using LSB Image Steganography Method and AES Encryption Algorithm. *International Journal of Signal Processing Image Processing and Pattern Recognition*, vol. 8(5), pp. 259-266

Singh, N., Meherhomji, V., & Chandavarkar, B.R. (2020). Automated versus Manual Approach of Web Application Penetration Testing. *11th International Conference on Computing, Communication and Networking Technologies Kharagpur*, 1-3 July 2020, pp. 1-6. https://doi.org/10.1109/ICCCNT49239.2020.9225385

Singh, K., Singh, P., & Kumar, K. (2018). User Behavior Analytics-based Classification of Application Layer HTTP-GET Flood Attacks. *Journal of Network and Computer Applications* vol. 112, pp. 97–114.

Singh, K., Singh, P., & Kumar, K. (2018). User Behavior Analytics-based Classification of Application Layer HTTP-GET Flood Attacks. *Journal of Network and Computer Applications,* vol. 112, pp. 97–114.

Singh, V., Choubisa, M., & Soni, G. K. (2020). Enhanced Image Steganography

Technique for Hiding Multiple Images in an Image using LSB technique. TEST Engineering and Management, pp. 30561-30565

Singh, K.J., & De, T. (2017). MLP-GA based algorithm to detect application layer DDoS Attack. *Journal of Information Security and Applications*, vol. 36, pp. 145–153.

Singh, A., & Malik, S. (2013). Securing Data by Using Cryptography with Steganography, *International Journal of Computer Network and Information Security*, vol. 3(5), pp. 404–409

Singhai, D., & Gupta, C. (2018). A Review of Various Image Encryption Technique Using AES and Random RGB Substitution. *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 7(1), pp 210-215

Sneha, P.S., Sankar, S., & Kumar, A.S. (2020). A Chaotic Color Image Encryption Scheme Combining Walsh–Hadamard Transform and Arnold–Tent Maps. *Journal of Ambient Intelligent Human Computer*, vol. 11, pp. 1289–1308 https://doi.org/10.1007/s12652-019-01385-0

Sohal, M., & Sharma, S. (2022). BDNA-A DNA inspired Symmetric Key Cryptographic Technique to Secure Cloud Computing. *Journal of King Saud University Computer Information Sciences*, vol. 34(1), pp. 1417-1425

Stallings, W. (2014). *Cryptography and Network Security Principles and Practice*. 6th Ed. Prentice Hall, Upper Saddle River.

Stritter, B., Freiling, F., König, H., Rietz, R., & Ullrich, S. (2016). Cleaning up Web 2.0's Security Mess-At Least Partly. IEEE Security and Privacy, vol. 14(4), pp. 8-57. https://doi.org/10.1109/MSP.2016.31

Symantec, T.M (2012). Don't be afraid of mobile banking apps. Retrieved from http://www.banktech.com/channels/dont-be-afraid-of-mobile-banking-apps/240006734 on 15/7/2022

Sridevi, D. R., Vijaya, P., & Rao, K. S. (2013). Image Steganography Combined with Cryptography. International Journal of Computers & Technology, vol. 9(1), pp. 976-984

StarTrinity (2019). StarTrinity SIP Tester (Call Generator, Simulator)—VoIP Monitoring and Testing Tool. Retrieved February 6, 2022 from http://startrinity.com/VoIP/SipTester/SipTester.aspx.

Statista, Smartphones (2020). Statistics & Facts, Statista, Hamburg, Germany, retrieved from https://www.statista.com/topics/840/smartphones/.

Statista. (2019b). Share of U.S. Mobile Phone Banking Users 2016, retrieved from https://www.statista.com/statistics/244414/percentage-of-us-mobile-phoneusers-who- use-mobile-banking/

Stevens, C., (2020). Assembling Cybersecurity: The politics and materiality of Technical Malware Reports and the Case of Stuxnet. *Contemporary Security Policy*, vol.41 (1): p. 129-134

Stinson, D.R. (2006). *Cryptography, Theory and Practice*, Third edition, Chap-man & Hall/CRC

Sodhi, G. K., & Gaba, G. S. (2018). An Efficient Hash Algorithm to preserve Data Integrity. *Journal of Engineering Science and Technology*, vol. 13(3), pp. 778-789

Sogaard, J., Krasula, L., Shahid, M., Temel, D., Brunnstrom, K., & Razaak, M. (2016). Applicability of Existing Objective Metrics of Perceptual Quality for Adaptive

Video Streaming. *Society for Imaging Science and Technology IS&T International Symposium on Electronic Imaging*, DOI: 10.2352/ISSN.2470-1173.2016.13. IQSP-206

Sounthiraraj, D., Sahs, J., Greenwood, G., Lin, Z., & Khan, L. (2014). Smvhunter: Large scale, automated detection of ssl/tls man-in-the-middle vulnerabilities in android apps. *In Proceedings of the 21st Annual Network and Distributed System Security Symposium 2014*

Souppaya, M., Scarfone, K. (2013). Guide to Malware Incident Prevention and Handling for Desktops and Laptops. *NIST Special Publication*, vol. 800, pp. 83

Sukumar, AK., Subramaniyaswamy, V., Vijayakumar, V., Ravi, L. (2020). A Secure Multimedia Steganography Scheme using Hybrid Transform and Support Vector Machine for Cloud-based Storage. *Multimedia Tools and Applications.* Doi: 10.1007/s11042-019-08476-2.

Suzuki, Y.E., & Monroy, S.A.S. (2022). Prevention and Mitigation Measures against Phishing emails: a sequential Scheme Model. *Security Journal*, vol. 35(4), pp. 1162-1182

Taba, M.S., Rahim, M.S.M., Lafta, S.A., Hashim, M.M., & Alzuabidi, H.M. (2019). Combination of Steganography and Cryptography: A Short Survey. *2nd International Conference on Sustainable Engineering Techniques*, pp. 1-13, Doi: 10.1088/1757-899x/518/052003

Taha, M.S., Shafry, M., Zeebaree, D., Hashim, M.M., Khalid, H.N. (2020). Information Hiding: Tools for Securing Biometric Information. *Technology Reports of Kansai University*, pp. 1383-1394.

Taher, M.M., Ahmad, A.R., Hameed, R.S., Mokri, S.S. (2022). A Literature Review Various Steganography Methods. *Journal of Theoretical and Applied Information Technology*, vol. 100(5), pp. 1412-142

Tahir, A.S. (2015). Design and Implementation of RSA Algorithm using FPGA. *International Journal of Computer and Technology*, vol 14(12), pp. 6361-6367

Talasila, S., Vijaya, Kumar, G., Vijaya Babu, E., Nainika, K., Veda, S.M., & Mohan, P. (2024). The Hybrid Model of LSB-Technique in Image Steganography using AES and RSA Algorithms. *Springer,* Singapore, https://doi.org/10.1007/978-981-99-8451=0_34

Talom, F.S.G., & Tengeh, R.K. (2019). The Impact of Mobile Money on the Financial Performance of the SMEs in Douala, Cameroon, *Sustainability*, vol. 12 (1), pp. 1

Tajpour, A., Heydari, M.Z., Masrom, M., & Ibrahim, S. (2010). SQL injection Detection and Prevention Tools Assessment. *3rd International Conference on Computer Science and Information Technology,* Vol. 9, pp. 518-522.

Talom, F.S.G., & Tengeh, R.K. (2020). The Impact of Mobile Money on the Financial Performance of SMEs in Douala, Cameroon, *Sustainability*, vol. 12, pp. 183

Tarawneh, H., Otair, M., Alomari, A., & Altarawneh, M. (2012). Mobile Banking Based on Standalone Mobile Application Clients: A Suggested Mobile Banking Solution for Banks in Jordan. International Journal of Advanced Research in Computer Science, Vol. 3(1), pp. 49-58

Tariq, N. (2018). Impact of Cyberattacks on Financial Institutions. *Journal of Internet Banking and Commerce*, vol 23(2), pp. 1-11

Tasevski, I., & Jakimoski, K. (2020). Overview of SQL Injection Defense Mechanisms. *28th Telecommunications Forum*, pp. 1- 4. IEEE

Tauhid, A., Tasnim, M., Noor, S.A., Faruqui, N., &Yousuf, M.A. (2018). A Secure Image Steganography using Advanced Encryption Standard and Discrete Cosine Transform. *Journal of Information Security*, vol 10(3), pp. 117-129.

Taylor, A. R. E. (2021). Standing by for Data Loss: Failure, Preparedness and the Cloud. Ephemera: Theory & Politics in Organization, vol. 21(1).

Techopedia (2022). Client/Server Architecture. Retrieved on 6/8/2022 at 11.58 AM from https://www.techopedia.com/definition/438/clientserver-architecture

Techopedia (2013). Digital Signature Algorithm. Retrieved from https://www.techopedia.com/definition/27504/digital-signature-algorithm-dsa

Tello-Rodríguez, M., Ocharán-Hernández, J.O., Pérez-Arriaga, J.C., Limón, X., Sánchez-García, Á.J. (2021). A Design Guide for Usable Web APIs. *Trudy ISP RAN/Proc. ISP RAS*, vol. 33(1), pp. 173-188

Thapar, S. S., Sarangal, H. (2018). A Study of Data Threats and the Role of Cryptography Algorithms. *IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference.* doi:10.1109/IEMCON.2018.8614943

Tran, H.T.T., & Corner, J. (2016). The impact of communication channels on mobile banking adoption. *International Journal of Bank Marketing*, vol. 34(1), pp. 78-109.

The Standard (2017). Man Charged with Hacking KRA and Causing Sh 4 billion Loss. https://www.standardmedia.co.ke/amp/business/article/2001233552/man-charged-with-hacking-kra-and-causing-sh4b-loss

The Register (2017). Hackers delight: Mobile bank app security flaw could have smacked millions, https://www.theregister.co.uk/2017/12/11/mobile-banking-security-research

The Straits Times (2015). Over $ 7000 lost in malware attack at fake banking portal, http://www.straitstimes.com/singapore/over-7000-lost-in-malware-attack-at-fake-banking-portal

The Telegraph (2017). Flaw discovered in banking apps leaving millions vulnerable to Hack, http://www.telegraph.co.uk/science/2017/12/06/flaw-discovered-banking-apps-leaving-millions-vulnerable-hack

Thirumaran, N., Soni, S., & Brendha, G. (2015). Design of Context-Aware Interactive Voice Response System. *Advances in Natural and Applied Sciences*, vol. 9(6), pp. 669-675

Thirumaran, M., Soni, S., & Gayathry, B. G. (2015). An intelligent interactive voice response system for banking domain. In Proceedings of the 2015 *International Conference on Advanced Research in Computer Science Engineering & Technology* (ICARCSET 2015), pp. 1-6.

Thiyagarajan, R, & Meenakshi, P. B. (2019). An enhancement of EAACK using P2P ACK and RSA public key cryptography Meas. *International Journal Meas Confederation*, vol. 136, pp. 116–121.

Thomas, B.A. (2016). Encryption Algorithms: A Survey, vol. 4(2),

Thomas, S., E, Philip, S. T., Nazar, S., Mathew, A., & Joseph, N. (2012). Advanced Cryptographic Steganography using Multimedia Files. *International Conference on Electrical Engineering and Computer Science* (ICEECS-2012).

Thompson, C., Leininger, R., & Bhatt, R. (2017). Mobile Banking Applications: Security Challenges for Banks. Retrieved on 31/12/2022 at 12.38 from https://www.accenture.com/t20180223T145013Z__w__/usen/_acnmedia/PDF-49/Accenture-Mobile-Banking-Apps-SecurityChallenges-Banks.pdf.

Tiwari, K., & Gangurde, S. (2020). LSB Steganography using Pixel Locator Sequence With AES, https//:arxiv.org/pdf/2012.022494

Tobah, Y., Kwong, A.,Kang, I., Genkin, D., & Shin, K.G. (2022). Combining Specter and Rowhammer for New Speculative Attacks. *In Proceedings of the IEEE Symposium on Security and Privacy, San Francisco, CA, USA*, pp. 1-18

Tripathi, N., Mehtre B.M. (2014). Analysis of various ARP poisoning mitigation Techniques: A comparison. In Proceedings of the *International Conference on Control, Instrumentation, Communication, and Computational Technologies*, pp. 125–132.

Tripathi, N., & Hubballi, N. (2016). A probabilistic anomaly detection scheme to detect DHCP starvation attacks. In Proceedings of the *International Conference on Advanced Networks and Telecommunications Systems*, pp. 1–6.

Tripathi, N., Hubballi, N., & Singh, Y. (2016). How secure are web servers? An empirical study of slow HTTP DoS attacks and detection. *In Proceedings of the International Conference on Availability, Reliability, and Security*, pp. 454–463.

Tripathi, N., & Hubballi, N. (2018). Slow Rate Denial of Service Attacks against HTTP/2 and Detection. *Computers & Security*, vol. 72, pp. 255–272.

Tripathi, N., & Hubballi, N. (2018). Detecting stealth DHCP starvation attack using

Machine learning approach. *Journal of Computer Virology and Hacking Techniques*, vol. 14(3), pp. 233–244.

Tripwire. (2016). DYN Restores Service After DDoS Attack Downed Twitter, Spotify, Others. Retrieved July 23, 2022 from https://www.tripwire.com/state-of-security/latest-security-news/dyn-restores-service-ddos-attack-broughttwitter-spotify-others/.

Tuli, P., & Sahu, P. (2013). System Monitoring and Security Using Keylogger, *International Journal of Computer Science and Mobile Computing,* vol.2, pp. 106-111

Unit 4 Lab 4: Data Representation and Compression, pp. 6 (edc.org)

USC-SIPI Image Database. http://sipi.usc.edu/database/ (accessed on 2nd March 2023)

US Department of the Treasury. Financial Services Sector-Specific Plan; US Department of the Treasury: New York, NY, USA, 2015.

Vairaprakash, G., Kannan, S., Mahajan, T.M. (2017). Cryptographic Tree and Its Key Management for Securing Outsourced Data in the Cloud. *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 7(5), pp. 255-259, 2017.

Valmik, N.K., & Kshirsagar, V.K. (2014). Blowfish Algorithm. *Journal of Computer Engineering,* vol 16, pp. 80-83

Vedaraj, M. (2018). A Hybrid Data Encryption Technique using Twofish and Elgamal for Cloud Computing, vol. 8(1473), pp. 1473–1478

Venkataksrishnan, S., Kaushik, A., & Verma, J.K. (2020). Sentiment Analysis on Google Play Store Data using Deep Learning. *Applications of Machine Learning,* *Springer*, pp. 15-30.

Verma, A., Kaur, S., & Tech, B.C.M. (2016). Improvement of the Performance and Security of Advanced Encryption Standard using AES algorithm and Comparison with Blowfish Research Scholar. *International Research Journal of Engineering and Technology*, vol 3(10), pp. 660-674.

Verma, A., Guha, P., & Mishra, S. (2016). Comparative Study of Different Cryptographic Algorithms. *International Journal of Emerging Trends and Technology in Computer Science*, vol 5(2), pp. 58-63.

Vijayakumar, P., Vijayalakshmic, V., & Zayaraz, G. (2016). An Improved Level of Security for DNA Steganography using Hyperelliptic Curve Cryptography. *Wireless Personal Communications*, vol. 89, pp. 1221-1242.

Vincent, O.R., Okediran, T.M., Abayomi-Ali, A.A., & Adenitan, O.J. (2020). An Identity-Based Elliptic Curve Cryptography for Mobile Payment Security. *SN Computer Science*, vol 1(2), pp. 1-12.

Waghmare, M., Golekar, P., Hatwar, A., Parimal, R., & Hiware, A. (2017). Enhancement of Security in Mobile Banking Applications. *International Journal for Scientific Research and Development*, vol. 5(1), pp. 110-112.

Wang, X., Feng, D., Lai, X., Yu, H. (2004). Collisions for Hash Functions MD4, MD5, HAVAL-128, and RIPEMD. Jinan250100, China: The School of Mathematics and System Science, Shandong University. https://eprint.iacr.org/2004/199.pdf

Wang, M., & Long, Y. (2020, November). SM9 Digital Signature with Non-repudiation. *In 2020 16th International Conference on Computational Intelligence and Security* (CIS), IEEE, pp. 356-361.

Wang, L., Song, H., & Liu, P. (2016). A novel hybrid color image encryption algorithm using two complex chaotic systems, Opt. Lasers Eng. Vol. 77. Pp. 118–125.

Wang, H., & Wang, S. (2004). Cyber Warfare: Steganography vs. Steganalysis, Communications of the ACM, vol. 47, pp. 76-82

Wazid, M., Zeadally, S., & Das, A. K. (2019). Mobile Banking Evolution and Threats: Malware Threats and Security Solutions. *IEEE Consumer Electronics Magazine* Vol 8(2), pp 56-60 Doi: 10.1109/mce.2018.2881291

Weetracker (2021). The Quirky Past of Kenyan Mobile Banking was "Mobile Banking" Literally. Retrieved from https://weetracker.com/2021/07/22/mobile-banking-kenya-history/

Wireshark network protocol analyzer (2022) accessed on 2/1/2023 from https://www.wireshark.org/

Xu, G., Xie, X., Huang, S., Zhang, J., Pan, L., & Lou, W. (2020). A Novel Policy-Based XSS Defense Mechanism for Browsers. *IEEE Transactions on Dependable and Secure Computing,* vol. 19, pp. 826-878. https://doi.org/10.1109/TDSC.2020.3009472

Xu, C., Zhao, G., Xie, G., & Yu, S. (2014). Detection on application layer DDoS using random walk model. *In Proceedings of the International Conference on Communication*, pp. 707-712.

Yadav, M., & Dhankar, A. (2015). Image Steganography Techniques: A Review, *International Journal for Innovative Research in Science and Technology*, vol. 2(2), pp. 243-248.

Yan, P., & Yan, Z. (2017). A survey on dynamic mobile malware detection. *Software Quality Journal*, pp. 1-29

Yang, A. S. (2009). Exploring adoption difficulties in mobile banking services, Canadian Journal of Administrative Sciences, vol. 26(2), pp. 136-14

Yang, J., & Peng, B. (2017). An optimized Algorithm based on Generalized Difference Expansion Method used for HEVC Reversible Video Information Hiding *17th IEEE International Conference on Communication Technology,* vol. 5, pp. 1668-1672

Yassein, M. B., Aljawarneh, S., Qawasmeh, E., Mardini, W., & Khamayseh, Y. (2017). Comprehensive Study of Symmetric Key and Asymmetric Key Encryption Algorithms in Engineering and Technology. *International Conference,* pp. 1-7 IEEE. Doi: 10.1109/ICEngTechnol.2017.8308215

Yigit, Y., & Karabatak, M. (2019). A Steganography Application for Hiding Student Information into an Image. *In proceedings of the 7th International Symposium on Digital Forensics and Security, Barcelos, Portugal,* pp. 1-4

Yildirim, N., & Varol, A. (2019). Research on Security Vulnerabilities in Online and. Mobile Banking Systems. *7th International Symposium on Digital Forensics and Security*

Yildirim, N., & Varol, A. (2018). A Research on Security Vulnerabilities in Online and Mobile Banking Systems. *7th International Symposium on Digital Forensics and Security,* pp. 1-5

Zahran, B., AL-Azzeh, J., Al Qadi, Z., Al-Zoghoul, M., & Khawatreh, S. (2018). A

Modified LBP Method to Extract Features from Color Images. *Journal of Theoretical and Applied Information Technology*, vol. 96(10).

Zargar, S.T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE Communications Surveys & Tutorials 15, 4 (2013), 2046–2069.

Zay, K. (2019). Design and Implementation of Electronic Payment Gateway for Secure Online Payment System. *International Journal of Trend in Scientific Research and Development*, vol 3(5), pp. 1329-1334.

Zekauskas, P., Vveinhardt, J., & Andriukaitiene, R. (2017). Philosophy and Paradihm of Scientific Research. Retrieved on 2/8/2022 at 15.44 PM, from htttps://www.intechopen.com/chapters/58890

Zhang, M., & Tong, X.-J. (2015). A new Algorithm of Image Compression and Encryption Based on Spatiotemporal Cross Chaotic System, *Multimedia Tools Applications*, vol. 74 (24), pp. 11255–11279.

Zang, Q., & Qunding. (2015). Digital Image Encryption based on Advanced Encryption Standard Algorithm, *5th International Conference on Instrumentation and Measurement, Computer, Communication and Control*, pp. 1218-1221.

Zhe, D., Qinghong, W., Naizheng, S., &Yuhan, Z. (2017). Study on Data Security Policy Based on Cloud Storage. *IEEE 3rd International Conference on Big Data Security on Cloud.* Doi: 10.1109/BigDataSecurity 2017.12

Zhijun, W. (2021). Steganography and Steganalysis in Voice over IP: A Review. *Sensors*, vol. 21(4), p. 1032.

Zhou, L., & Nunes, M. (2016). Formulating a framework for desktop research in Chinese information systems. In Handbook of Research on Innovations in Information Retrieval, Analysis, and Management (pp. 307-325). IGI Global.

Zhou, T. (2018). Examining Users Switch from Online Banking to Mobile Banking. *International Journal of Networking and Virtual Organizations*, vol. 18(1), pp. 51–66. https://doi.org/10.1504/IJNVO.2018.090675

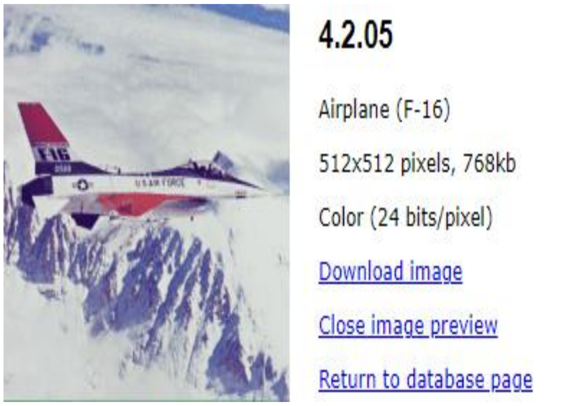
Zhu, L., Hu, Z., Heidemann, J., Wessels, D., Mankin, A., & Somaiya. N. (2015) Connection-Oriented DNS to Improve Privacy and Security. *In Proceedings of the IEEE Symposium on Security and Privacy*, vol. (15) pp. 171–186.

Zubair, M., Ali, A., & Anam, S. (2023). A DDDAS-Based Impact Area Simulation Study of Highway Abnormalities. *In Proceedings of the MOL2NET22 Conference on Molecular, Biomedical & Computational Sciences and Engineering*, 8th ed, pp. 1-31

# APPENDICES

## APPENDIX I: USC-SIPI IMAGE DATABASE

Link: https://sipi.usc.edu/database/



4.2.05

Airplane (F-16)

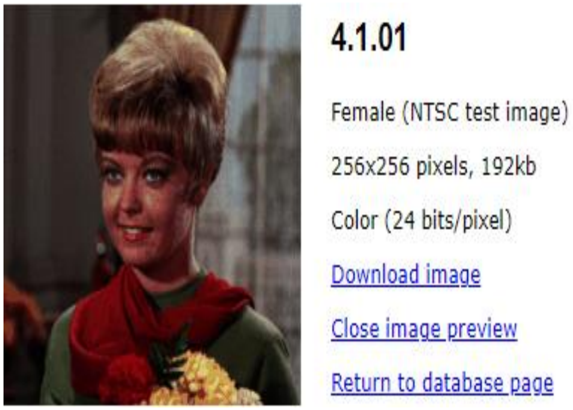
512x512 pixels, 768kb

Color (24 bits/pixel)

Download image

Close image preview

Return to database page



4.1.01

Female (NTSC test image)

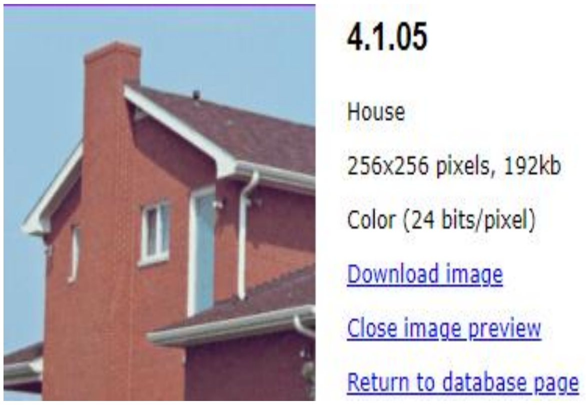
256x256 pixels, 192kb

Color (24 bits/pixel)

Download image

Close image preview

Return to database page



4.1.05

House

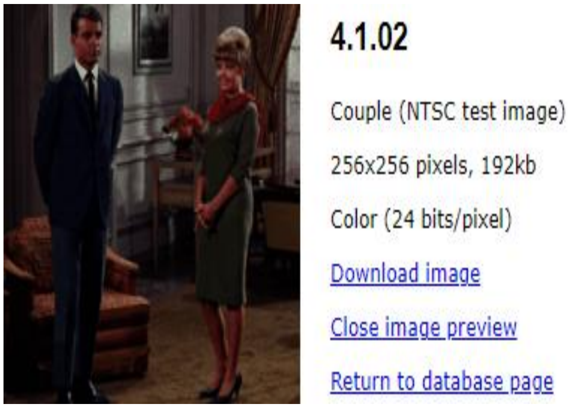
256x256 pixels, 192kb

Color (24 bits/pixel)

Download image

Close image preview

Return to database page



4.1.02

Couple (NTSC test image)

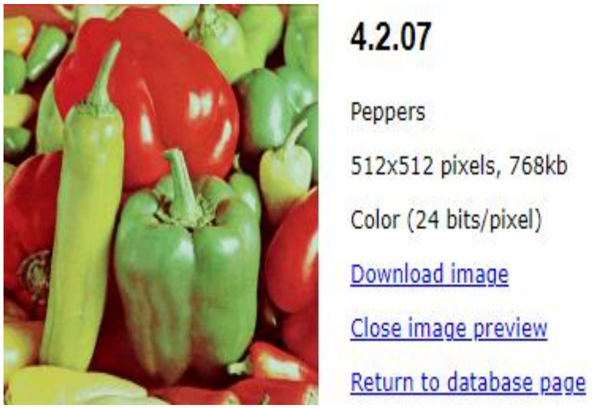
256x256 pixels, 192kb

Color (24 bits/pixel)

Download image

Close image preview

Return to database page



4.2.07

Peppers

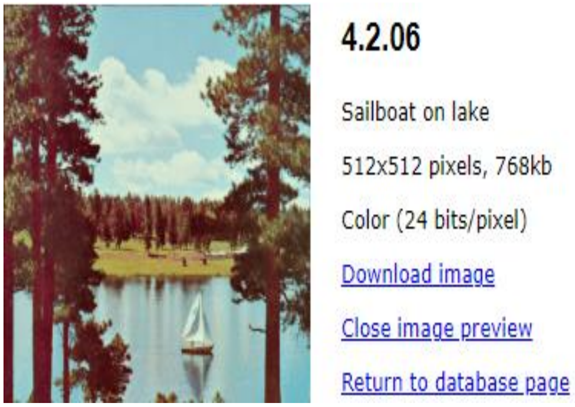
512x512 pixels, 768kb

Color (24 bits/pixel)

Download image

Close image preview

Return to database page



4.2.06

Sailboat on lake

512x512 pixels, 768kb

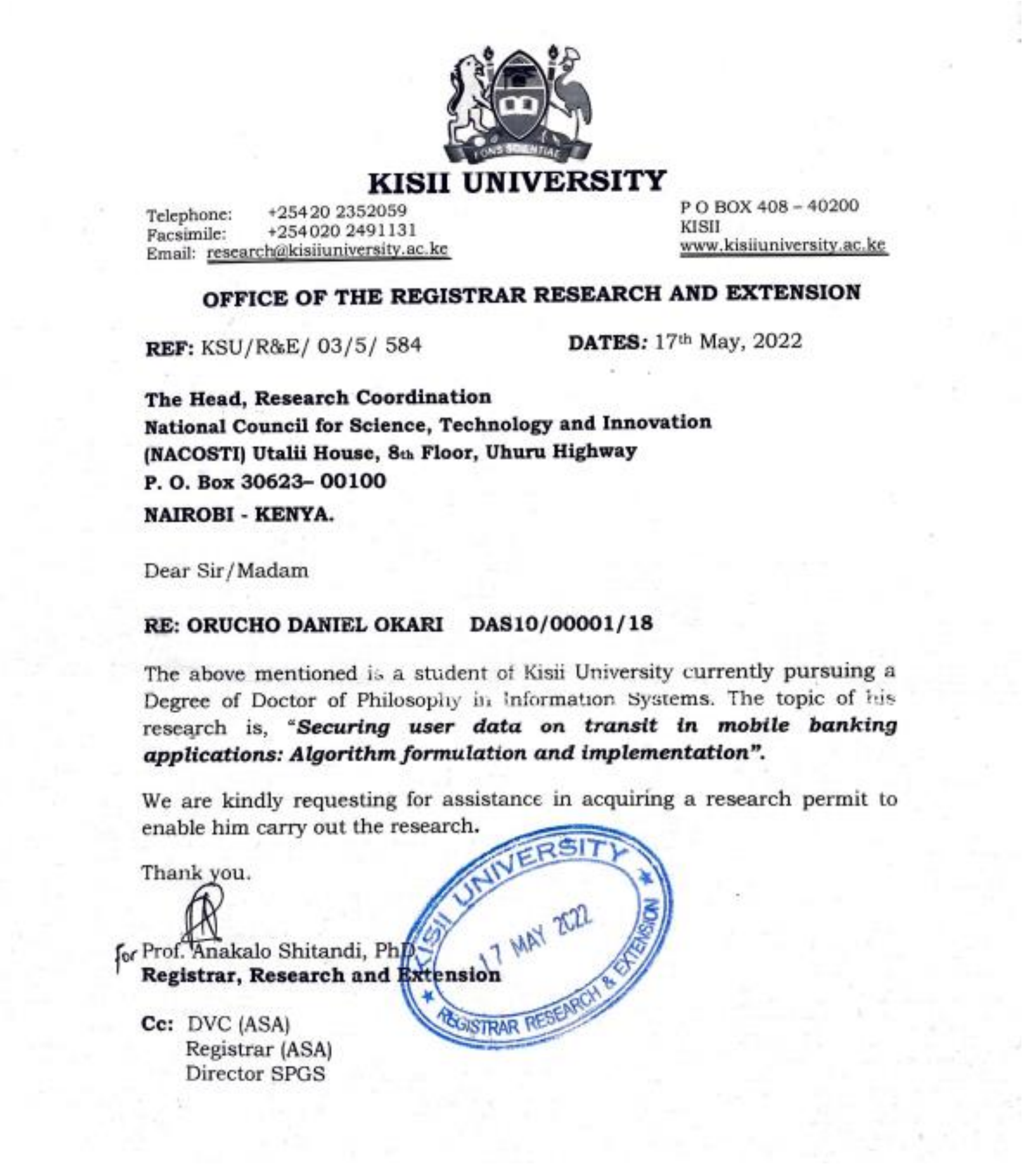
Color (24 bits/pixel)

Download image

Close image preview

Return to database page

# APPENDIX II: LETTER OF INTRODUCTION FROM THE UNIVERSITY



FONS SCIENTIAE

**KISII UNIVERSITY**

Telephone: +25420 2352059
Facsimile: +254020 2491131
Email: research@kisiiuniversity.ac.ke

P O BOX 408 – 40200
KISII
www.kisiiuniversity.ac.ke

**OFFICE OF THE REGISTRAR RESEARCH AND EXTENSION**

**REF:** KSU/R&E/ 03/5/ 584 **DATES***:* 17th May, 2022

**The Head, Research Coordination**
**National Council for Science, Technology and Innovation**
**(NACOSTI) Utalii House, 8th Floor, Uhuru Highway**
**P. O. Box 30623– 00100**
**NAIROBI - KENYA.**

Dear Sir/Madam

**RE: ORUCHO DANIEL OKARI DAS10/00001/18**

The above mentioned is a student of Kisii University currently pursuing a Degree of Doctor of Philosophy in Information Systems. The topic of his research is, ***"Securing user data on transit in mobile banking applications: Algorithm formulation and implementation".***

We are kindly requesting for assistance in acquiring a research permit to enable him carry out the research.

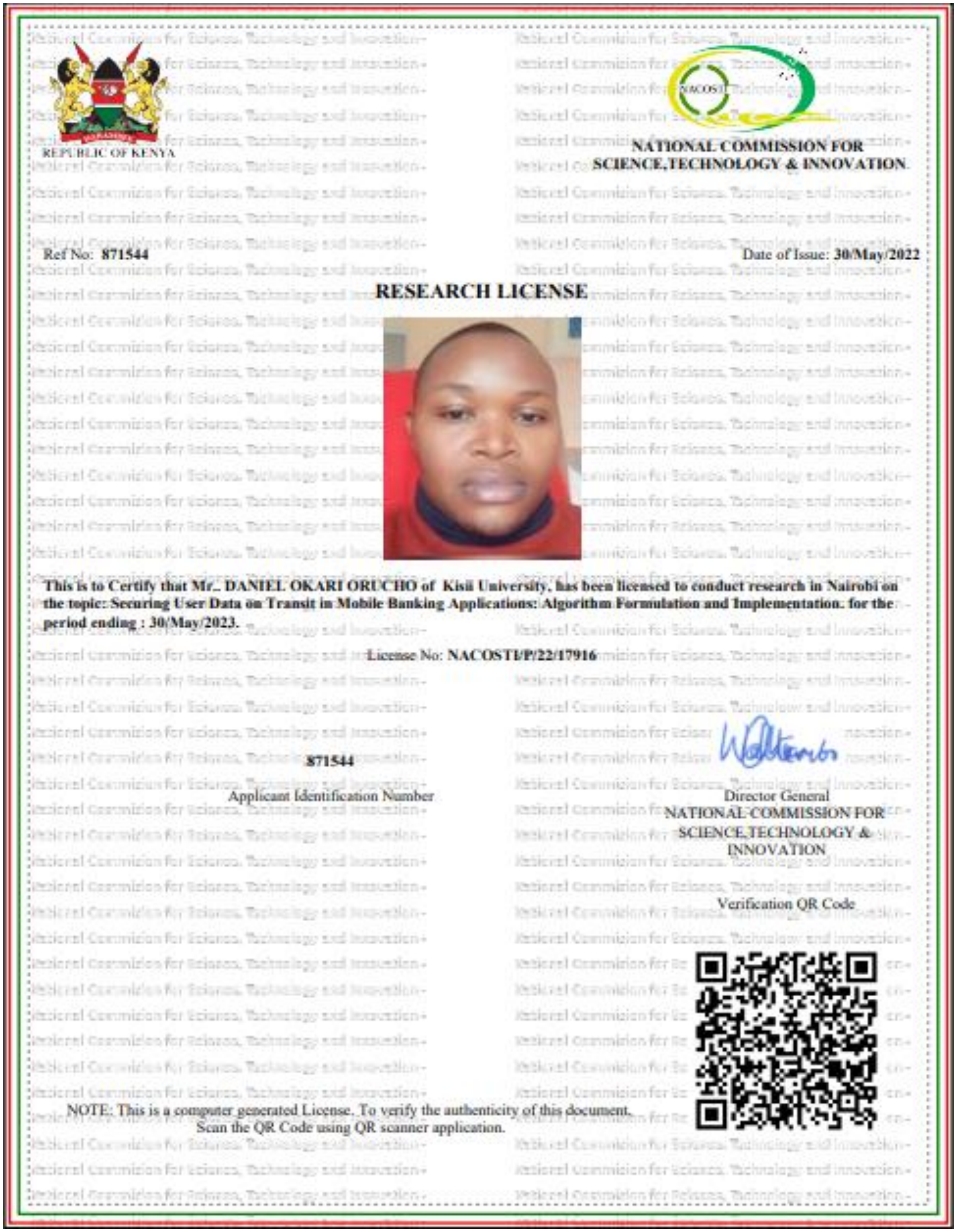
Thank you.

for Prof. Anakalo Shitandi, PhD
**Registrar, Research and Extension**

KISII UNIVERSITY ★
17 MAY 2022
REGISTRAR RESEARCH & EXTENSION ★

**Cc:** DVC (ASA)
Registrar (ASA)
Director SPGS

## APPENDIX III: RESEARCH PERMIT



REPUBLIC OF KENYA

NATIONAL COMMISSION FOR
SCIENCE,TECHNOLOGY & INNOVATION

Ref No: **871544**

Date of Issue: **30/May/2022**

**RESEARCH LICENSE**

**This is to Certify that Mr.. DANIEL OKARI ORUCHO of Kisii University, has been licensed to conduct research in Nairobi on the topic: Securing User Data on Transit in Mobile Banking Applications: Algorithm Formulation and Implementation. for the period ending : 30/May/2023.**

License No: **NACOSTI/P/22/17916**

**871544**

Applicant Identification Number

Director General
NATIONAL COMMISSION FOR
SCIENCE,TECHNOLOGY &
INNOVATION

Verification QR Code

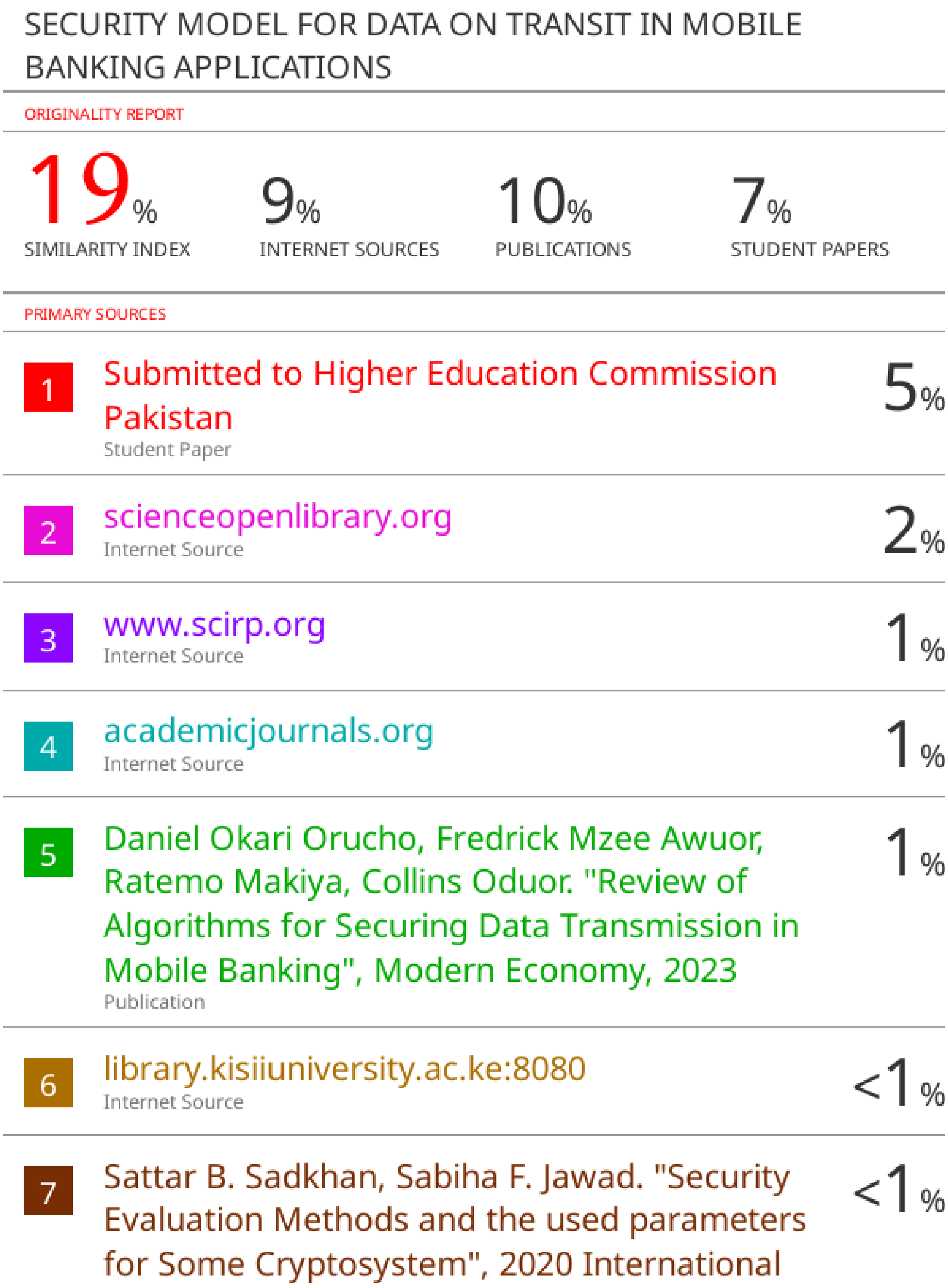
NOTE: This is a computer generated License. To verify the authenticity of this document,
Scan the QR Code using QR scanner application.

## APPENDIX IV: SOURCE CODE LSB-AES ON-TRANSIT USER-DATA PROTECTION ALGORITHM

https://ww2.mathworks.cn/matlabcentral/fileexchange/173575-lsb-aes-hybrid-algorithm

\

## APPENDIX V: PLAGIARISM REPORT



SECURITY MODEL FOR DATA ON TRANSIT IN MOBILE BANKING APPLICATIONS

ORIGINALITY REPORT

| 19% | 9% | 10% | 7% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to Higher Education Commission Pakistan<br>Student Paper | 5% |
| 2 | scienceopenlibrary.org<br>Internet Source | 2% |
| 3 | www.scirp.org<br>Internet Source | 1% |
| 4 | academicjournals.org<br>Internet Source | 1% |
| 5 | Daniel Okari Orucho, Fredrick Mzee Awuor, Ratemo Makiya, Collins Oduor. "Review of Algorithms for Securing Data Transmission in Mobile Banking", Modern Economy, 2023<br>Publication | 1% |
| 6 | library.kisiiuniversity.ac.ke:8080<br>Internet Source | <1% |
| 7 | Sattar B. Sadkhan, Sabiha F. Jawad. "Security Evaluation Methods and the used parameters for Some Cryptosystem", 2020 International | <1% |