A HYBRID SECURITY MODEL FOR DATA PROTECTION IN PUBLIC CLOUD COMPUTING

DANIEL OKARI ORUCHO

A THESIS SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY IN THE SCHOOL OF COMPUTING AND MATHEMATICS IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF THE DEGREE OF MASTER OF SCIENCE IN CYBER SECURITY OF THE CO-OPERATIVE UNIVERSITY OF KENYA

2025

DECLARATION

Declaration by the Candidate

This thesis is my original work and has not been presented for award of a degree in any other University or for any other award

…….. 24/11/2025 ……

Signature Date

Orucho Daniel Okari: C005/600020/2023

Declaration by the Supervisors

I/We confirm that the work reported in this thesis was carried out by the candidate under our supervision and has been submitted with our approval as university supervisors

…….. 24.11.2025 ……

Signature Date

Dr. Charles Katila, PhD

Department of Computer Science and Information Technology

School of Computing and Mathematics

The Co-operative University of Kenya

…… 24/11/2025 ……

Signature Date

Dr. Ngaira Mandela, PhD

Department of Computing and Informatics

School of Science and Technology

Open University of Kenya

DEDICATION

I dedicate this thesis to my cousin Joakim Okenye and my brother David Orucho. You have always given me the necessary motivation I needed.

ACKNOWLEDGEMENT

I would like to express my deepest gratitude to God, the source of all wisdom and understanding, for His unwavering protection and for granting me the physical and mental strength to complete this thesis.

I am sincerely thankful to Dr. Charles Katila and Dr. Ngaira Mandela for their prompt and thoughtful reviews, which greatly contributed to the refinement of this work.

I also extend heartfelt appreciation to my cousin Joakim Okenye and my brother David Orucho for their unwavering support, encouragement, and belief in me throughout this journey. Their presence and motivation have been invaluable.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ABBREVIATIONS AND ACRONYMS

API:	Application Programming Interface
DDoS:	Distributed Denial of Service
DSA:	Digital Signature Algorithm
DSRM:	Design Science Research Methodology
GUI:	Graphical User Interface
HE:	Homomorphic Encryption
LSB:	Least Significant Bit
MATLAB:	Matrix Laboratory
MFA	Multi-Factor Authentication
MSE:	Mean Squared Error
PIN:	Personal Identification Number
PSNR:	Peak Signal to Noise Ratio
RSA:	Rivest Shamir Adleman
SQL:	Structured Query Language
TFA:	Two-Factor Authentication
USC-SIPI:	University of Southern California Signal and Image Processing Institute
XSS:	Cross-Site-Scripting

DEFINITION OF TERMS

Plaintext: This is the original input data that can be read and makes sense. It can be a message, figures, or any other type of information that can be understood when read.

Ciphertext: This is the resulting product when plaintext is encrypted using a cryptographic algorithm. It ensures confidentiality of information.

Cover-image: This is the original image which is used to hide secret data. It is a medium upon which a file or data can be hidden to avoid detection.

Stego-image: This is the resulting image when data is embedded in a cover-image.

ABSTRACT

This thesis sought to address the following objectives: assess security threats and attacks on data in public cloud computing, assess existing techniques for securing data in public cloud environments, developed a hybrid algorithm to enhance data security in public cloud computing, and evaluated the effectiveness of the proposed hybrid algorithm. This study adopted a positivist research paradigm and employed both descriptive and data science methodologies. Secondary data was collected from peer-reviewed journal articles, conference proceedings, and books, while high-quality images were sourced from the University of Southern California Signal and Image Processing Institute database for algorithm test simulations. Encryption was applied to numerical data before embedding it into cover images, and data analysis was conducted using content analysis, gap analysis, visual quality assessment, statistical evaluation, and comparisons with baseline algorithms. Simulations were performed using MATLAB R2021a on six color images. The study identified various threats to data in public cloud computing, including insecure APIs, legal and compliance issues, data corruption, code injection, insider threats, mobile malware, phishing and social engineering, denial of service, man-in-the-middle attacks, SQL injection, and cross-site scripting. These threats can be mitigated through robust security frameworks such as input validation, encryption, access controls, and continuous monitoring, along with organizational strategies like employee training, legal compliance audits, and regular vulnerability assessments. Techniques for securing data in public cloud environments include cryptographic algorithms like Blowfish, Twofish, RSA, digital signatures, homomorphic encryption, authentication, and data hiding methods. The proposed hybrid algorithm integrated Least Significant Bit substitution with Paillier Homomorphic Encryption and was evaluated using Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), entropy, and histogram analysis. Results showed low MSE values (0.000023–0.00012), high PSNR values (87.3–94.57), and entropy values ranging from 6.4207 to 7.5937, indicating minimal distortion, high reconstruction quality, and strong data complexity. The algorithm maintained high visual and statistical fidelity post-embedding, with perfect correlation and stable entropy values, while minor chi-square fluctuations suggested localized changes without compromising imperceptibility or robustness. Overall, the hybrid algorithm proved effective for secure data hiding in public cloud computing by preserving image quality and ensuring statistical integrity.

CHAPTER ONE

INTRODUCTION

This section outlines the foundational components of the research. It introduces the background of the topic, clarifies the problem under investigation and the objectives of the study. Additionally, the chapter highlights the core assumptions made by the researcher, defines the thematic boundaries within which the study was carried out, and notes the challenges encountered during the research process. The rationale behind selecting this subject is also justified.

1.1 Background of the Study

Cloud computing has rapidly become a cornerstone of modern IT infrastructure, offering scalable, cost-effective, and flexible services through models such as Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). It enables users to access shared digital resources on demand, with deployment models ranging from public and private to hybrid and community clouds (Odun-Ayo, Ananya, Agono, & Goddy-Worlu, 2018; Mell & Grance, 2011).

Despite its transformative benefits (Alkhamese, Shabana, & Hanafy, 2019), cloud computing, particularly in public environments, remains vulnerable to a range of security threats, including insecure APIs, insider attacks, data breaches, and injection-based exploits. These threats often compromise the core principles of data confidentiality, integrity, and availability, deterring organizations from fully embracing cloud solutions (Thabit, Alhomdy, Al-Ahdal, & Jagtap, 2021). While existing security mechanisms such as encryption and access control offer partial protection, they are often either computationally intensive or insufficiently robust

against evolving attack vectors (Butt, Mehmood, Shah, Amin, Shaukat, Raza, Suh, & Piran, 2020).

This creates a critical gap in lightweight, effective, and scalable data protection strategies. To address this challenge, the present study proposes a hybrid algorithm that integrates Least Significant Bit (LSB) steganography with Paillier Homomorphic Encryption. This approach aims to enhance data security in public cloud computing by embedding encrypted data within digital images, thereby ensuring imperceptibility, statistical integrity, and resilience against common detection techniques. By focusing on this specific gap, the study contributes a novel solution to the ongoing challenge of secure data transmission and storage in cloud environments.

1.2 Statement of the Problem

Cloud computing has become a foundational technology for modern enterprises, offering scalable, flexible, and cost-effective access to digital resources (Alkhamese, Shabana, & Hanafy, 2019; Wu, Buyya, & Ramamohanarao, 2020). However, its widespread adoption, particularly in public cloud environments, has exposed critical vulnerabilities in traditional data security mechanisms. Techniques such as Two-Factor Authentication (2FA), Multi-Factor Authentication (MFA), and cryptographic schemes are commonly used to safeguard cloud data, yet they exhibit inherent weaknesses. While 2FA and MFA provide layered identity verification, they remain susceptible to phishing, SIM swapping, and device-level compromise, especially when authentication tokens are transmitted over insecure channels. Even biometric authentication, often considered more secure, can be spoofed using deepfakes or synthetic fingerprints. These vulnerabilities are exacerbated in public cloud settings where perimeter-based security models are ineffective and access is ubiquitous (Odun-Ayo, Ananya, Agono, & Goddy-Worlu, 2018).

Cryptographic schemes, though essential for ensuring data confidentiality and integrity, face challenges in key management, computational overhead, and exposure to insider threats. In distributed cloud environments, encryption keys must be securely stored and transmitted across multiple nodes, increasing the risk of mismanagement or leakage. Moreover, the multi-tenancy architecture of public clouds introduces isolation failures that can result in cross-tenant data breaches (Butt, Mehmood, Shah, Amin, Shaukat, Raza, Suh, & Piran, 2020).

Recent incidents illustrate the severity of these shortcomings. In 2024, a Distributed Denial of Service (DDoS) attack on Microsoft Azure disrupted global operations for nearly 10 hours, exposing deployment flaws in protective systems (Communications Authority of Kenya, 2024). Similarly, between July and September 2024, targeted malware and ransomware campaigns exploited default credentials and unsecured endpoints, affecting critical sectors such as government, healthcare, and industrial control systems (Communications Authority of Kenya, 2024).

These persistent threats and limitations underscore the inadequacy of conventional security mechanisms in addressing the dynamic and complex threat landscape of public cloud computing. Therefore, there is a pressing need for lightweight, resilient, and scalable security solutions that can enhance data protection without compromising performance. This study addresses that gap by proposing a hybrid algorithm that integrates Least Significant Bit (LSB) steganography with Paillier Homomorphic Encryption to improve data security in public cloud environments. The algorithm aims to provide a novel approach to secure data embedding and transmission, ensuring imperceptibility, statistical integrity, and robustness against detection and attack.

1.3 Objectives of the Study

This thesis sought to achieve the following specific objectives:

i. To assess major data security threats and attack vectors in public cloud computing environments through desktop-based literature analysis.
ii. To assess data security techniques employed in public cloud computing through desktop-based literature analysis.
iii. To develop a hybrid algorithm that ensures security of data in public cloud computing
iv. To evaluate the robustness of the developed hybrid algorithm in securing data within public cloud computing environments.

1.4 Research Questions

i. What are the major data security threats and attack vectors affecting public cloud computing environments, as identified through desktop-based literature analysis?
ii. What data security techniques are currently employed in public cloud computing environments, based on findings from desktop-based literature analysis?
iii. How can a hybrid algorithm be developed that secures data in public cloud computing?
iv. How robust is the developed hybrid algorithm in securing data within public cloud computing environments?

1.5 Significance of the Study

This study highlights the urgent need to safeguard data stored in public cloud environments, where security remains a foundational concern. As cloud computing continues to support critical services across sectors, the findings of this thesis offer valuable insights for cloud infrastructure providers. By outlining proven data protection strategies, the study provides a knowledge base that can inform the design of secure, resilient cloud applications.

For the academic community, particularly scholars in information security, the thesis contributes to the growing body of literature on cloud data protection. It offers a foundation for further research and curriculum development, enabling educators to disseminate relevant and practical knowledge to learners and practitioners.

Finally, the study benefits students and emerging researchers by presenting a pool of knowledge that bridges theory and practice. The insights gained can be applied to real-world cloud computing scenarios, fostering innovation and enhancing the security posture of future systems.

1.6 Scope of the Study

This thesis strived to enhance data protection within public cloud environments, specifically examining prevalent security threats and attacks, alongside countermeasures employed to safeguard sensitive information. Moreover, the study undertook design and assess a composite algorithm that ensures robust data security in public cloud platforms. A combination of descriptive and algorithm-centric research designs guided the investigation. Descriptive methodology was utilized to thoroughly analyze and document data-related threats and the defense mechanisms adopted in cloud computing, while the data science approach, focused on artifact and algorithm development, was used in creating and validating the proposed hybrid algorithm.

1.7 Limitation of the Study

This thesis utilized the Signal and Image Processing Institute (SIPI) image dataset to assess image quality during simulation tests. While the dataset offers high-resolution and standardized images suitable for algorithm evaluation, it may not fully represent the diversity of image types encountered in real-world cloud environments, such as medical scans, satellite

imagery, or compressed mobile uploads. This limits the generalizability of the findings across broader application domains. Additionally, the evaluation metrics such as PSNR, MSE, and Entropy, focus primarily on visual fidelity and statistical complexity. These metrics do not capture real-time performance, resistance to adversarial attacks, or scalability under dynamic cloud workloads.

The simulations were conducted in MATLAB R2021a, which may not reflect performance variations across other platforms or programming environments. Furthermore, the study did not incorporate primary data collection or live cloud deployment, which restricts its ability to account for emerging threat patterns or operational constraints. Delimitations include the intentional focus on public cloud environments and image-based steganography, excluding other cloud models and data formats. These boundaries were set to maintain methodological clarity and feasibility but may limit the breadth of applicability.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

The chapter provide an understanding of cloud computing and focused on security challenges related to public cloud computing, including potential security risks and specific types of attacks targeting data. Additionally, the chapter addressed the methodologies and techniques employed to safeguard data within the public cloud. It also evaluated hybrid algorithms, delved into information theory, and highlighted current gaps in the research surrounding cloud security.

2.2 Threats in Public Cloud Computing

2.2.1 Insecure APIs

Cloud Application Programming Interfaces (APIs) are utilized to access sensitive data and applications. However, as the world is interconnected, the APIs are exposed to malicious attacks (Bettendorf, 2021), and insecure because of their design (Salt Labs, 2023). Additionally, APIs are exposed to bot cyberattacks (Chinnasamy, 2022).

2.2.2 Legal and Compliance Issues

When data flows across international borders, the relevant legal, regulatory, and compliance frameworks can become unclear, raising various security issues. Additionally, the variations in privacy, security laws, and regulations across countries, whether at the local, national, or state levels complicate these legal and compliance challenges (Islam, Manivannan, & Zeadally, 2016).

2.2.3 Data Corruption

Cloud-based information is as vulnerable as data stored on unreliable media. Additionally, losing an encryption key could trigger a data disaster (Gupta, Shankar, & Gupta, 2021). Furthermore, having a backup strategy is critical to protect against both deliberate and accidental data losses. To ensure data availability, it's essential that the Content Security Policy includes regular backup processes. Furthermore, “backup data must comply with security protocols to prevent risks like tampering or unauthorized access (Alrasheed, Aied alhariri, Adubaykhi, & El Khediri, 2022).

2.2.4 Code Injection Threat

A virtual attack involving illegal SQL injection can severely affect Software-as-a-Service (SaaS) platforms, especially those with weakly designed applications. This attack is executed through a flawed interface, enabling the unauthorized implementation of SQL statements which is typically not intended for public access (Kaur & Kaimal, 2023).

2.2.5 Insider Threats

Insider threats present a major challenge in cloud computing environment because unauthorized users can misuse their privileges or unintentionally jeopardize data security. Addressing threats that emanate from insider threats requires a blend of technical measures such as robust identity and management systems, and other organizational strategies that can detect and manage these threats (Kandias, Virvilis, & Gritzalis, 2013).

2.2.6 Mobile Malware

Malware is harmful software that is designed to undermine cloud computing security. It brings about data breaches and identifies theft and can spread through executable files or software platforms (Ye, Li, Adjeroh, & Iyengar, 2018). Most common propagation techniques

include drive-by-downloads, backdoors, phishing, removable drives, and using other tools such as exploit kits (Alex, Creado, Almobaideen, Alghanam, & Saadeh, 2023; Caviglione, Choras, Corona, Janicki, Mazurczyk, Pawlicki, & Wasielewska, 2021).

2.2.7 Phishing and Social Engineering

Despite advances in cybersecurity, the human factor remains the most vulnerable point, mainly due to mistakes, limited knowledge, and even clever manipulation. Recently, cybercriminals have adopted sophisticated tools like Natural Language Processing (NLP) to craft highly realistic phishing emails and smishing (SMS-based phishing) messages. These deceptive techniques trick individuals into disclosing sensitive personal details or unknowingly installing malicious software, ultimately resulting in major data leaks and financial damage (Communications Authority of Kenya, 2024).

2.3. Attacks on Data in Public Cloud Computing

This form of cyberattack involves flooding a network or service with an overwhelming volume of data packets. The targeted server becomes overloaded with a continuous stream of connection requests, filling its buffer memory with excessive and redundant data. When this memory is depleted, the server loses its ability to establish new connections. This type of assault primarily affects infrastructure and platform as service levels (Kaur & Kaimal, 2023). Consequently, users are unable to connect to the affected resources or services. The attack can disrupt websites, applications, or entire systems, leading to significant downtime.

2.3.1 Denial of Service

This kind of cyberattack floods a network or service with an overwhelming number of data requests, causing the server to struggle with too many simultaneous connections. As a result, the host’s buffer memory gets overloaded with repetitive and unnecessary data. If that buffer

space runs out, the server loses its ability to establish new connections (Kaur & Kaimal, 2023). This disruption leads to users being cut off from accessing key services or resources, and in severe cases, can take down entire websites, applications, or systems.

2.3.2 Man-in-the-Middle (MITM)

This type of threats involves a hacker covertly inserting themselves into a conversation between two unsuspecting parties. This is done by creating a deceptive bridge that tricks the victim into believing they are communicating securely, while the attacker monitors and alters the exchanged data. Often, the goal is to impersonate one of the participants or steal sensitive information like banking credentials, login passwords, or private data. The consequences can include unauthorized access to accounts or illicit financial transactions. Such attacks are especially common in public Wi-Fi environments, where data packets are often unencrypted and vulnerable. In these settings, cybercriminals reroute traffic through their own devices, capturing and exploiting confidential user data with ease (Conti, Dragoni & Lesyk, 2016).

2.3.3 Structured Query Language (SQL) Injection

Structured Query Language (SQL) injection is a cyberattack that takes advantage of vulnerabilities commonly found in the design of web applications. In this method, attackers embed harmful commands within standard SQL inputs, often through user entry fields. When executed, these commands manipulate the backend database in unintended ways, giving unauthorized users access to confidential data. Once the attack is successful, hackers can retrieve sensitive records, compromise database integrity, and in some cases, even rewrite or delete data altogether (Bhadauria & Sanyal, 2012).

2.3.4 Cross-Site Scripting (XSS)

These assaults occur when threat actor manages to deliver malware to unsuspecting participants via software platforms, typically through scripts that run in their browsers (Bhadauria & Sanyal, 2012; Hydara, Sultan, Zulzalil, & Admodisastro, 2015). One major trigger for XSS attacks is poor validation of user-provided input. This can lead to two critical issues: the input is not correctly neutralized, or the site applies flawed validation logic. Such oversights open up security loopholes that attackers can readily exploit.

2.4 Mechanisms Securing Data in Public Cloud Computing

2.4.1 Symmetric Encryption

This technique uses sole, shared key to carry out both the encoding and decoding of data. This method is particularly effective for securing large volumes of information quickly. Some commonly used symmetric encryption techniques explored in this study include Blowfish and Twofish algorithms. The same secret key is applied during both encryption and decryption, whether it's a word, a number, or a string of characters. For a message to be successfully encrypted and later understood, both the sender and the recipient must possess this identical key (Ahmed & Naeem, 2022).

2.4.1.1 Blowfish Algorithm

Blowfish algorithm offers strong system security. It supports key lengths from 32 to 448 bits, making it a flexible choice for protecting information. Its ability to handle variable key sizes enhances its versatility, while its quick encryption and decryption speeds make it ideal for applications that need rapid data processing. Moreover, Blowfish is customizable and can function with different block sizes, ensuring efficient data transmission with minimal resource consumption (Encryption Consulting, 2024). Although it has a vulnerability related to weak

keys, there have been no successful attacks exploiting this weakness (Schneier, 1994; Schneier, 1996).

Blowfish algorithm is suitable for automatic file encryption systems and 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).

Even though the Blowfish algorithm is a well-regarded encryption algorithm, it has some challenges such as key change impact on speed in which negatively impacts when frequent key changes are required. Secondly there is the issue of the lengthy key schedule which is a drawback especially when a quick setup key is essential. Lastly, Blowfish algorithm is susceptible to Brute-force attacks which compromises security (Encryption Consulting, 2024).

2.4.1.2 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 supports 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 several cryptographic applications (Mandal & Singh, 2021).

The Twofish encryption algorithm processes data by initially splitting the plaintext into four 32-bit segments, totaling 128 bits. The first step is the whitening stage, where each 32-bit segment is combined using an XOR operation with four key words. Then, the algorithm proceeds with a series of 16 rounds, each involving the function g applied to the leftmost two words.

2.4.2 Asymmetric Encryption

The data encrypted with the public key can only be accessed by using the corresponding private key. Notable asymmetric encryption methods include RSA, Elliptic Curve Cryptography (ECC), and Digital Signatures. In this approach, the public key is available to both the sender and the recipient, enabling anyone to send messages securely". However, only the recipient possesses the private key, which is required to decrypt the message, thereby boosting security (Devi, 2015). Asymmetric encryption is generally more secure than symmetric encryption due to the significant computational resources needed to manage encryption processes, including the use of digital certificates, hashing, digital signatures, and encryption protocols.

2.4.2.1 Rivest-Shamir-Adelman Algorithm

In an asymmetric encryption system, two distinct keys are used: one for "encryption and the other for decryption. Rivest-Shamir-Adleman (RSA) is commonly employed in network security to safeguard data. The RSA cryptosystem's central challenges involve the integer factorization problem and the RSA problem, which are linked to finding the Nth root of a

large number, N. Number theory shows that multiplying two prime numbers is relatively simple, but factorizing their product is far more complex. RSA security relies heavily on the difficulty of factorizing large numbers, with key sizes ranging from 2048 to 4096 bits making the factorization task computationally demanding (Bhanot & Hans, 2015).

This is primarily due to the modular exponentiation used in both encryption and decryption, which is computationally intensive. As the key size increases, so does the number of operations required for both encryption and decryption, following a quadratic growth pattern. RSA includes processes such as key generation, encryption, and decryption (Saini & Vandana, 2022).

2.4.2.2 Digital Signatures

DSA operates through discrete logarithmic challenges and modular exponentiation, which are computationally tough to crack using brute-force methods. This strength comes from the "difficulty of solving the discrete logarithm problem, combined with the time-intensive nature of modular exponentiation, which collectively protects DSA from brute-force attacks (Simplilearn, 2022).

This method relies on a hashing algorithm, which produces a unique hash or digest that reflects even the smallest change in the data. This ensures that any alteration to the data, whether intentional or accidental, results in a completely different hash, "alerting the receiver" that the data was tampered with during transmission (Thapar & Sarangal, 2018). Digital signatures enhance security by providing additional layers of non-repudiation, integrity, and authentication (Mishra, 2017). The DSA process unfolds in three steps: key generation, signature creation, and signature verification (Simplilearn, 2022).

DSA technology is widely applied in various systems such as smart cards, ISDN (Integrated Services Digital Network), web applications, and email verification. DSA is advantageous over RSA because it generates keys faster, is more stable and secure, and requires less storage space. This is because DSA uses smaller key sizes, efficient signature representation, and fixed-length components such as r and s (Simplilearn, 2022). A secure digital signature system ensures that breaking the algorithm would require solving the discrete logarithm problem (Stinson, 2006).

Two types of forgery attacks exist: selective and potential forgeries. Existential forgery occurs when a forged message/signature pair is created by an unauthorized individual, while in selective forgeries, the adversary creates a predetermined message/signature pair (Kumar, 2016).

2.4.2.3 Homomorphic Encryption

Homomorphic encryption is a cryptographic technique that enables computations on encrypted data without the need for decryption. The core principle is to develop encryption schemes that support “both addition and multiplication” on ciphertexts. When these encrypted operations are decrypted, the result is equivalent to performing the “same operations on the original data”. This allows users to perform computations on data stored in the cloud without exposing the underlying plaintext (HCLTech, 2022).

The homomorphic encryption (HE) algorithm supports the recreation of decryption operations using specific plug operations that mirror the corresponding read operations (Ladislas, 2023; Ladislas & Phelix, 2023). A cryptosystem is considered homomorphic if it permits operations on the ciphertext. Each operation on the ciphertext is reflected when the message is decrypted. A “mathematical function f (.) is commutative if f (x, y) = f(y, x), meaning operations such as

x + y are homomorphic, whereas operations like x - y are not because they are non-commutative. Homomorphic encryption enables secure scientific computations on the cloud without exposing sensitive inputs (Scheibner, Kiltz, & Struck, 2021; Kareyo, Siraj, & Mohammed, 2020).

2.4.3 Authentication

Two-factor authentication involves two types of verification chosen from two categories of credentials such as “something a user knows and something a user has”. On the other hand, multi-factor authentication involves utilization of something a user knows, something a user has, and something that is inherent in the user (Mohsin, Han, Hammoudeh, & Hegarty, 2017).

2.4.4 Data Hiding Techniques

Data hiding techniques involves inserting information within a cover medium which is then transmitted over the internet. The two main techniques of data hiding are steganography and watermarking.

Steganography is a method that is utilized by hiding messages in a cover medium to avoid detection. It facilitates communication between the sender and receiver possible without suspicion or revealing the presence of the message (Johnson, Zoran, & Sushil, 2001). On the other hand, watermarking is a technique that provides protection of information against illegal data transmission (Tamanna & Ashwani, 2017).

There are five main categories of steganography namely, text, image, audio, video, and network steganography (Gautam & Sharma, 2015; Ismail & Souley, 2018; Thusanth & Gan, 2016; Shefali & Shrivastava, 2015).

2.5 Evaluation of Hybrid Algorithms

Hybrid encryption combines multiple encryption techniques to bolster data protection, offering a more secure way of safeguarding sensitive information. Unlike this, asymmetric key cryptography (or public-key cryptography) utilizes separate key pairs “for the processes of encryption and decryption”.

2.5.1 Complexity Evaluation

The complexity of an encryption method can serve as a criterion for assessing its overall security. An encryption system is considered secure if it cannot be easily broken, even when executed on a machine with limited memory and processing power. However, if the algorithm is tested on a machine with ample memory and high processing capacity, it is said to have a measurable level of security (Krishnamurthy & Ramaswamy, 2009).

2.6 Homomorphic Encryption Theory

The homomorphic encryption enables two primary operations on encrypted data: addition and multiplication (Challa, 2020). Homomorphic encryption utilizes either symmetric or asymmetric keys and features a unique feature known as homomorphic evaluation. These schemes include key generation, encryption, decryption, and evaluation functions. It enables arbitrary computations to be performed on encrypted data, producing results that remains in encrypted form (Gentry, 2009; Gentry, 2010).

2.7 Hybrid Theory for Data Security in Cloud Computing

Hybrid data security models blend several protection techniques to create a strong and well-rounded defense for information. These models often combine tools like encryption, hidden data embedding (steganography), and user access restrictions “to tackle the distinct security challenges posed by cloud-based systems” (Sheeba & Parameswari, 2023).

2.8 Conceptual Model

This study's conceptual model in Figure 2.1 illustrates the relationship between existing cryptographic techniques and the development of a secure hybrid algorithm for public cloud computing.

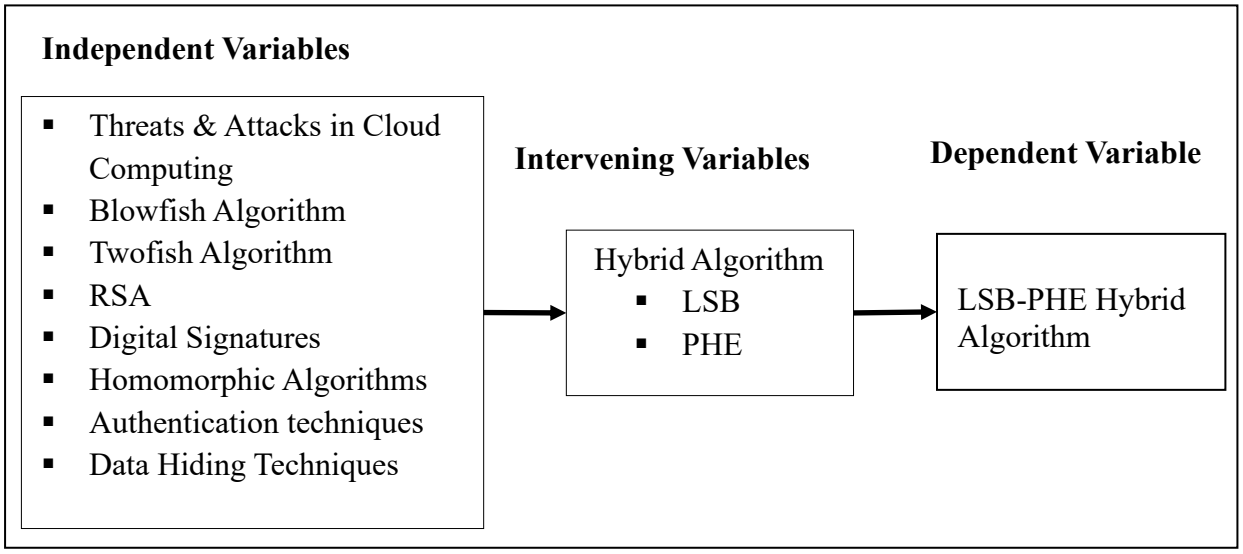

Figure 2.1: Conceptual Model

The independent variables include threats and attacks in cloud computing, alongside established cryptographic and authentication methods such as Blowfish, Twofish, RSA, digital signatures, homomorphic encryption, authentication techniques, and data hiding strategies. These variables represent foundational security mechanisms and challenges.

The intervening variable is a hybrid algorithm that integrates Homomorphic Encryption (HE) and Least Significant Bit (LSB) steganography. This construct enhances the effectiveness of the independent variables by addressing their limitations in confidentiality, integrity, and resilience.

The dependent variable is the resulting LSB-PHE algorithm, which is evaluated for its robustness, scalability, and security performance in cloud environments. The model posits that by combining HE and LSB, the hybrid algorithm can overcome the constraints of traditional methods and offer a more secure solution for cloud data protection.

2.9 Research Gaps

The table 2.1 presents a comparative overview of recent hybrid cryptographic techniques designed to enhance data security in cloud computing environments. While these approaches ranging from homomorphic encryption combined with steganography to advanced optimization algorithms, demonstrate significant potential, several critical gaps remain unaddressed. A recurring limitation across the reviewed techniques is their high computational complexity and resource intensiveness, which poses challenges for real-time or large-scale deployment, especially in resource constrained environments such as mobile cloud systems.

Despite the integration of multiple cryptographic and steganographic methods, scalability and efficiency trade-offs persist, often requiring substantial storage, bandwidth, and processing power. Moreover, while many studies focus on enhancing confidentiality and privacy, few provide a balanced evaluation of robustness, adaptability, and performance under dynamic cloud workloads or adversarial conditions. There is also a lack of standardized benchmarking across these hybrid models, making it difficult to compare their effectiveness or generalize their applicability across different cloud architectures.

Table 2.1: Hybrid Techniques for Security of Data in Cloud Computing

Author	Technique	Limitations	Comment
Abundi, Abdulla, Alotaibi, Asiri, Ahmed, Zamani, Motwakel, & Yaseen (2022)	Homomorphic encryption with Steganography	The system was computationally intensive, needed increased storage requirements, system is complex to implement and maintain	Utilized in data transmission on the Internet of Everything (IoE) environment
Cosseron, Hoffmann, Meaux, & Standaert (2022)	Hybrid Homomorphic Scheme (HHE) featuring Elizabeth stream cipher	The system incurs significant computational overhead, and is complex to implement.	The system provided high security for privacy-preserving computations and secure data processing. The system was applied to homomorphically evaluate a neural network inference
Sunil, Devika, Teja, Sreenivas, Afreed, & Reddy (2024)	Blowfish, AES, RSA, & LSB steganography	A combination of multiple cryptographic algorithms with steganography is computationally intensive, leads to increased storage requirements, scalability challenges, and the system can be complex to implement and maintain	The system was robust due to the combination of multiple cryptographic algorithms and steganographic technique. The system was designed for securing data in electronic communication systems
Muhammed, Isiaka, Asaju-Gbolagade, Adewole, & Gbolagade (2021)	Fully Homomorphic Encryption (FHE) in combination with N-prime model & Matrix operation	The system introduces significant computational complexity, requires increased storage and bandwidth requirements, integration of the N-prime model and Matrix operation adds complexity to its implementation and the system relies on the robustness of the underlying encryption techniques	Utilized in applications that require privacy-preserving computations and secure data processing. The system was primarily applied to secure data in cloud computing environments.
Kumana, Teja, &	Elliptic Galois	The system can be computationally intensive,	Used for enhancing data protection

Kumar (2024)	Cryptography (EGC) Protocol, Matrix XOR Encoding Steganography & Adaptive Firefly Algorithm	the system leads to increased storage requirements, challenges in scaling to handle large volumes of data or a high number of IoT devices, potentially impacting its performance and efficiency, integration of multiple techniques and optimization algorithms can make the system complex to implement and maintain, requiring specialized knowledge and expertise	in IoT environments.
Seth, Dalal, & Kumar (2019).	Paillier Homomorphic Encryption, RSA algorithm, & ElGamal Encryption	Computational complexity, system leads to increased storage and bandwidth requirements which may be a constraint for certain applications, implementation complexity, and challenges in scaling to handle large volumes of data or a high number of devices, potentially impacting its performance and efficiency	Utilized in systems that require data privacy and security
Joseph & Mohan (2022)	Bat algorithm and Cuckoo Search algorithm with the Paillier Homomorphic Encryption scheme.	Computational complexity, increased bandwidth and storage requirements, implementation complexity, and scalability	Utilized for security of data in cloud environments

Additionally, limited attention has been given to lightweight, hybrid encryption models that can maintain strong security guarantees while reducing overhead. Most existing solutions prioritize security at the expense of usability, integration simplicity, or real-time responsiveness.

To address the identified research gaps, this study aims to design a robust yet computationally efficient hybrid encryption algorithm specifically tailored for public cloud environments. It further seeks to evaluate the algorithm's performance, scalability, and resilience using standardized metrics and real-world cloud scenarios. Additionally, the study intends to contribute a comparative framework for assessing hybrid cryptographic models, focusing on both their security capabilities and operational feasibility.

2.10 Summary

This chapter highlighted public cloud computing topics such as overview of cloud computing, cloud computing frameworks, conceptual model, threats in public cloud computing, attacks on data in public cloud computing, mechanisms securing data in public cloud computing, evaluation of hybrid algorithms, and research gaps.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

In this chapter, the following are presented and discussed; research philosophy, research design, dataset, homomorphic encryption scheme, LSB algorithm, design of LSB-PHE algorithm, performance analysis of LSB-PHE hybrid algorithm, results analysis and presentation, and ethical considerations.

3.2 Research Philosophy

Positivism is deeply rooted in scientific investigation (Zekauskas, Vveinhardt, & Andriukaitiene, 2017). It suits this study as it focuses on present-day issues within real-world settings specifically, the security challenges facing public cloud computing. Accordingly, it aims to offer a practical remedy through a hybrid algorithm designed to strengthen security and address known vulnerabilities in cloud environments.

3.3 Research Design

This study utilized a blended approach, assimilating both Descriptive Research Design and Design Science (DS). Descriptive research, by definition, is applied to observe and document phenomena as they occur. Through this approach, the research will explore various data security risks and attacks in public cloud environments while examining the protective measures currently employed.

In addition, desktop research was used to source secondary data. It involved reviewing credible academic materials such as scholarly books, peer-reviewed journal articles, and conference proceedings. According to Zhou and Nunes (2016), desktop research is dependent on publicly accessible data. Desktop research was systematically applied to address two key

objectives of this study: (i) to assess major data security threats and attack vectors in public cloud computing environments, and (ii) to evaluate the data security techniques currently employed in such environments. This involved conducting a comprehensive literature analysis using publicly accessible academic sources, including peer-reviewed journal articles, scholarly books, and conference proceedings. By sourcing materials from reputable databases such as ACM Digital Library, Emerald Insight, and Google Scholar, the study identified prevailing security challenges, such as insider threats, data breaches, and unauthorized access, and critically examined the cryptographic and authentication techniques used to mitigate them. This process provided a foundational understanding of the current landscape of cloud security, informed the development of the hybrid algorithm, and helped pinpoint gaps in existing approaches that the study aimed to address.

The study also incorporated Design Science (DS) as a core methodological approach, particularly in addressing Objective iii: to develop a hybrid algorithm that ensures security of data in public cloud computing. Design Science is a solution-oriented methodology aimed at refining system development and enhancing the effectiveness of IT-based solutions (Dresch, Lacerda, & Antunes, 2015). It supports researchers in expanding their methodological toolkit by emphasizing iterative, artifact-driven inquiry and fostering theory development throughout the research lifecycle, from conceptualization to evaluation (Kruse, Seidel, & Purao, 2016). In this study, DS guided the structured development of the hybrid algorithm by integrating Homomorphic Encryption (HE) and Least Significant Bit (LSB) techniques, ensuring that the resulting solution was both theoretically grounded and practically applicable. As Geerts (2011) notes, DS is particularly well-suited for building and improving application systems, making it an ideal framework for designing a secure, efficient, and scalable encryption model for public cloud environments.

3.4 Sampling Design

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

The experimental simulations utilized the USC-SIPI Miscellaneous Image Database, which is a recognized benchmark in image quality assessment and steganography research. The database is publicly accessible through the USC-SIPI website. The dataset is renown as a standard reference for evaluating image quality in steganographic applications (USC-SIPI Image Database, 2023). It has also played a critical role in advancing theoretical research and image analysis within the domain of digital Steganography（Das, 2022; Panwar, Damani, & Kumar, 2018).

For simulation purposes, six high-resolution color images which include airplane, female, house, couple, peppers, and sailboat, were selected from the dataset. These images were chosen based on their availability in Tagged Image File Format (TIFF), which supports lossless compression, thereby preserving image integrity during data embedding（Unit 4 Lab 4, 2023). Their superior clarity and substantial dimensions make them particularly suitable for Steganographic embedding, ensuring minimal distortion and optimal payload capacity（Majjed, 2023）.

To simulate secure data concealment, numeric data was first encrypted using the Paillier homomorphic encryption scheme, a probabilistic asymmetric algorithm known for its additive homomorphic properties. This approach enabled encrypted computations to be performed directly on ciphertexts without decryption, enhancing both data confidentiality and computational flexibility. The encrypted payload was then embedded into the selected cover images, forming the basis of the experimental dataset. Figure 3.1 displays the images used in the tests.

Figure 3.1: Images for Simulation Trials (USCI-SIPI Image Repository, 2024).

Figure 3.1 presents a selection of images intended for simulation trials. These color systems are derived from the RGB spectrum, which closely mirrors the natural way human vision processes and distinguishes colors (Muhammad, Ahmad, Farman, & Zubair, 2014).

3.5 Homomorphic Encryption Scheme

Homomorphic encryption is a framework which is based on the concept of homomorphism. It helps service providers to perform operations on encrypted data while preserving its functional attributes and encrypted format. Several homomorphic encryption schemes exist, offering a range of efficiencies, security levels, and functionalities that suit various applications. They include partially homomorphic encryption (PHE), Somewhat Homomorphic Encryption (SHE), and Fully Homomorphic Encryption (FHE) (Acar, Aksu, Uluagac, Conti, 2019).

The Paillier encryption algorithm is one of the PHE algorithms which focusses on addition operations. It generates "two distinct keys for encryption and decryption of data". The algorithm has three operations, that is key generation, encryption and decryption (Regueiro, Seco, De Diego, & Lage, 2021).

3.5.1 Key Generation

1. Select two large prime numbers
2. Compute $n = p \times q$
3. Compute $\lambda = lcm(p-1, q-1)$, where lcm is the least common multiple
4. Select a random integer g such that g is a generator in the group n^2
5. Calculate $\mu = \left(L\left(g^{\lambda} \bmod n^2\right)\right)^{-1} \bmod n$, where $L(x) = \frac{x-1}{n}$
6. Shared key is (n, g) and confidential key is (λ, μ)

3.5.2 Encryption

Encoding a message m:

1. Choose an arbitrary integer r where $r = \in z_n$
2. Compute the ciphertext c as $c = g^m . r^n mod\ n^2$.

3.5.3 Decryption

To decrypt cyphertext c:

1) Compute$L(c^{\lambda} mod\ n^2)$.
2) Multiply the result by μ and n to get the plain text message m.

3.6 LSB Algorithm

Least Significant Bit (LSB) is a widely utilized steganographic technique which involves hiding data into the least significant bits of a file, ensuring the original pixel values remain largely unchanged (Sharda & Budhiraja, 2013). Steganographic methods can be categorized

based on type of cover media utilized to conceal data, which include text, image, audio, and video (Mohamad, Sahira, & Yasin, 2018).

When a file, video, text, message is hidden in another medium, the whole process is regulated by a stego-key. This key inhibits the ability to detect the embedded data to authorized parties who are familiar with it (Kheiralla, 2017).

3.6.1 Embedding Algorithm

In this algorithm, the LSB of image pixels are substituted with bits of concealed data. After embedding, the resulting image closely resembles the original image since the LSBs do not alter the image's appearance. This algorithm falls under spatial domain steganography, where crucial information is embedded in the LSB of the cover image (Hazim, 2022; Aljazaery & Alaidi, 2022). The LSB embedding method is depicted in Figure 3.2.

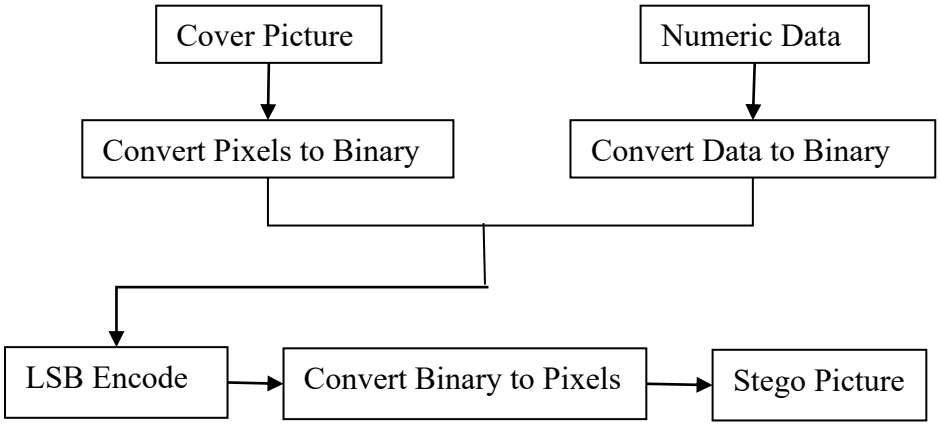

Figure 3.2: LSB Embedding Process

3.6.2 Decoding Algorithm

Begin by opening the stego image. Apply PHE decryption method. Use sequential decoding to retrieve hidden message. Figure 3.3 illustrates the LSB decoding process. This version conveys the same procedure in a unique structure, ensuring clarity and distinctness.

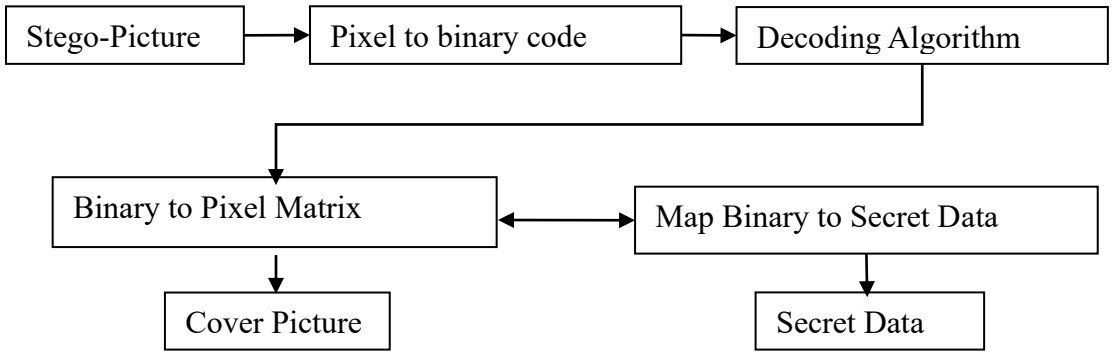

Figure 3.3: LSB Decoding Process

Figure 3.2 illustrates decoding process in which a stego image is fed into the system so that decoding algorithm translates binary code into image pixels to reveal the hidden message.

3.7 Design of LSB-PHE Algorithm

Design of LSB-PHE algorithm utilizes a synergy of the strengths of both algorithms to produce a robust algorithm as illustrated in Table 3.1.

Table 3.1: LSB-PHE Hybrid Algorithm

LSB-RSA Hybrid Algorithm	
Step 1:	Post Cover Image
Step 2:	Post Confidential Information
Step 3:	Employ PHE Encryption
Step 4:	Apply LSB Encoding
Step 5:	Read the Stego-Image for Decoding
Step 6:	Apply LSB Decoding
Step 7:	Apply PHE Decryption
Step 8:	Display Decrypted Data

3.8 Performance Analysis of LSB-PHE Hybrid Algorithm

The proposed hybrid algorithm was analyzed using efficiency of the algorithm in terms of Central Processing Unit (CPU) time which can be evaluated by analyzing its complexity. The

performance complexity of an algorithm is directly associated with its space and time requirements (Aspnes, 2017).

Space and time complexities measure the memory usage and execution time of an algorithm based on the size of the input. Various factors, including hardware, operating systems, processors, compiler software, and others, can influence these complexities but may not be accounted for in algorithm performance comparisons". The primary focus of complexity in algorithm analysis is the manner in which the algorithm operates. Big-O notation, as described by Brownlee (2011) and Cormen, Leiserson, Rivest, and Stein (2001), serves as an effective tool for assessing the asymptotic behavior of algorithm complexities.

Compliance of the developed hybrid algorithm was addressed adhering to established intellectual property laws and avoiding infringement of the existing rights regarding established homomorphic encryption or LSB steganography innovations. A thorough manual of the developed hybrid algorithm and its functionality was preserved.

3.8.1 Security Analysis

Security analysis of the developed hybrid data protection algorithm was achieved by ensuring that LSB substitution technique and Paillier homomorphic encryption were implemented correctly. It involved analysis of MSE and PSNR of the cover and stego images to test imperceptibility. Histogram analysis and chi-square tests were conducted on cover and stego images to evaluate irregularities in pixel distributions that may reveal hidden data. A synergy of the hybrid algorithm was simulated to confirm end-to-end confidentiality. Additionally, in terms of complexity, the algorithm was evaluated in terms of time and space complexity to analyze resource utilization.

3.8.2 Benchmarking Against Standard Algorithms

The developed algorithm was tested and compared with other algorithms in order to establish how well the developed algorithm performs relative to the ones that are already established. The metrics that were used for benchmarking were MSE, PSNR, and Entropy. Benchmarking helped to demonstrate that the hybrid algorithm was tested against recognized frameworks, and also helped to refine the developed hybrid algorithm for better performance.

3.8.3 Simulation and Testing

Simulation tests were conducted on cover and stego images using established metrics such as MSE, PSNR, entropy and histogram analysis on MATLAB software. Numeric data was first encrypted using Paillier homomorphic encryption scheme before being embedded into the cover images to generate stego images. The developed hybrid algorithm demonstrated fully functionalities in key generation, encryption, embedding, decoding and decryption.

3.9 Results Analysis and Presentation

Analysis of results were conducted using the following established metrics; PSNR, MSE, entropy and histograms. MATLAB software was used for simulations and results were displayed in tables and graphs.

3.10 Ethical Considerations

The researcher acquired an introductory letter from the administration of The Co-operative University of Kenya to support their application for an authority license from the National Commission for Science, Technology, and Innovation (NACOSTI), which was a necessary standard procedure for conducting the study. The researcher gathered academic resources from peer-reviewed journal sources such as Google, Google Scholar, Emerald, and ACM, online books, and conference papers. These materials helped address the researcher's

scholarly objectives and contribute to the academic community's awareness of experimental evidence.

For practical aspects, the researcher utilized USC-SIPI image database, which is open and freely accessible. The researcher ensured that there were no privacy concerns or violations when downloading and using the cover images from this database. These images were used solely for the study’s purpose. Additionally, the research community will have access to the results of simulation tests conducted on these images.

CHAPTER FOUR

DATA ANALYSIS, PRESENTATION AND INTERPRETATION

4.1 Introduction

In this chapter, the following are presented and discussed; security threats and attacks on data in public cloud computing, techniques securing data in public cloud computing, development of LSB-PHE data protection algorithm, implementation, and data analysis. Additionally, simulation tests, performance metrics of the proposed algorithm have been discussed.

4.2 Data Analysis of Security Threats and Attacks in Public Cloud Computing

The desktop-based literature review revealed that insecure Application Programming Interfaces (APIs) represent a significant vulnerability within public cloud computing infrastructures. APIs serve as critical gateways to sensitive data and services, yet their exposure to external access makes them prime targets for exploitation. The analysis showed that many APIs are developed with minimal security validation, lacking robust authentication, encryption, and access control mechanisms. This design weakness enables attackers to exploit APIs through methods such as injection attacks, session hijacking, and unauthorized data access. The reviewed studies consistently emphasized that poorly secured APIs contribute to a broader attack surface in cloud environments, underscoring the need for stricter implementation standards and continuous security auditing.

The desktop-based literature review revealed that cross-border data movement in public cloud environments introduces significant legal and compliance challenges. When data is transferred between jurisdictions, organizations often encounter unclear governance boundaries stemming from differences in privacy laws and data protection regulations. This regulatory ambiguity increases the risk of non-compliance, particularly in cases where

national laws conflict or lack harmonization. The analysis highlighted that many organizations struggle to maintain compliance due to limited visibility into where data is stored or processed, which can expose them to legal liabilities and sanctions. These findings underscore the need for clearer international frameworks and stricter contractual controls in cloud service agreements.

The desktop-based literature review highlighted that data stored in public cloud environments is vulnerable to corruption due to several operational and technical factors. Key risks identified include hardware failures, loss of encryption keys, and accidental data deletion—all of which can lead to partial or complete data loss. These incidents pose significant threats to data integrity and availability, often resulting in costly disruptions for organizations. The analysis emphasized that without robust backup and recovery strategies, such data losses can become irreversible. Consequently, regular and automated backup mechanisms were identified as essential safeguards to ensure business continuity and mitigate the impact of data corruption in cloud infrastructures.

The study discovered that SQL injection attacks pose a significant threat to data security in cloud computing environments. These attacks often exploit vulnerabilities in the design of web applications, enabling unauthorized access to databases, retrieval of sensitive records, and corruption of stored data. Such weaknesses compromise the integrity and confidentiality of cloud-hosted information, highlighting the critical need for secure coding practices and regular vulnerability assessments.

Additionally, the research identified that multi-cloud environments introduce further security challenges due to inconsistencies in software design, web browsers, and database systems across different platforms. These vulnerabilities can expose organizations to phishing attacks,

data breaches, and malware injection. To mitigate these risks, the literature recommends implementing strategies such as network segmentation and effective de-militarized zone (DMZ) management, maintaining patch schedules aligned with vendor guidance, hardening servers and firewalls according to best practices, and adopting secure architectural frameworks grounded in formal security design methodologies.

The desktop-based review identified malicious insiders, individuals within an organization who misuse access privileges, as a critical threat to cloud data security. These actors can intentionally or unintentionally access, alter, or leak sensitive information. To mitigate this risk, the literature recommends a multi-layered approach combining technical controls such as identity and access management (IAM) solutions with strict organizational policies focused on monitoring, auditing, and limiting privileged access.

The desktop-based literature review established that malware, a form of destructive software, poses a serious threat to cloud computing security. Malware is commonly deployed through various threat channels, including malicious payloads, phishing advertisements, and infected removable devices, enabling unauthorized penetration into cloud-based systems. Once embedded, malware can compromise system integrity, leading to unauthorized data exposure, data corruption, and disruption of services across cloud infrastructures. These findings underscore the importance of proactive malware detection, user awareness training, and layered security controls to safeguard cloud environments.

The desktop-based literature review identified social engineering as one of the most significant and elusive threats to public cloud computing. Although preventable, social engineering attacks are difficult to detect due to their reliance on human manipulation rather than technological exploitation. The analysis revealed a consistent attack pattern comprising

four stages: researching the target, establishing rapport, leveraging acquired knowledge to execute the attack, and erasing traces of the intrusion (Mouton, Leenen, & Venter, 2016). Unlike conventional cyberattacks, these threats exploit behavioral vulnerabilities, indicating that adversaries often study human behavior to gain unauthorized access to cloud resources (Nerwal, Mohapatra, & Usmani, 2019). This underscores the need for user awareness training, behavioral monitoring, and social engineering-resistant security protocols in cloud environments.

The desktop-based literature review revealed that Denial of Service (DoS) attacks remain a critical threat to public cloud infrastructures. The analysis found that attackers can launch DoS attacks by routing traffic through a personal access point under their control, targeting cloud servers that host sensitive applications such as mobile banking platforms. In such scenarios, the adversary deliberately drops all packet components of a communication flow between the application and the bank's server, leading to connection timeouts and service disruptions. This tactic ultimately exhausts system resources, degrading performance and availability for legitimate users (Kaka, Sastry, & Maiti, 2017). These findings highlight the importance of implementing traffic monitoring, anomaly detection, and rate-limiting mechanisms to defend against DoS threats in cloud environments.

The desktop-based literature review found that application layer Denial of Service (DoS) attacks are among the most frequent vulnerabilities in network protocols, often disrupting the normal operation of network devices and denying access to legitimate users. These attacks are difficult to detect because malicious traffic often mimics legitimate Internet activity, blending abnormal behavior with normal usage patterns (Kaka, Sastry, & Maiti, 2017). The analysis identified two main categories of application layer DoS threats: protocol-specific attacks and

generic attacks (Tripathi & Hubballi, 2018). Examples of protocol-specific attacks include Network Time Protocol (NTP) abuse, timeshifting DoS, Slow HTTP attacks, and Dynamic Host Configuration Protocol (DHCP) hunger assaults, all of which exploit weaknesses in specific network services to exhaust system resources and degrade service availability.

The desktop-based literature review found that Man-in-the-Middle (MITM) attacks, which resemble eavesdropping in nature, pose a serious threat to data confidentiality in public cloud environments. Also referred to as fire brigade attacks, the term originates from the bucket brigade method of firefighting, symbolizing the attacker's position between the sender and receiver of information (Cheng, 2010; Javeed et al., 2020). MITM attacks can be passive, where the attacker silently monitors client-server traffic for future exploitation, or active, where the attacker intercepts and alters the data in transit (Kieseberg, Fruhwirt, Schrittwieser, & Weippl, 2015). These attacks compromise trust and integrity in cloud communications, highlighting the need for strong encryption protocols and secure session management.

The desktop-based literature review identified Cross-Site Scripting (XSS) as a critical programming flaw that occurs when unsanitized user input is processed by a system (Kirsten, 2016). Attackers exploit this vulnerability by injecting malicious scripts into web applications, which are then executed in the user's browser. This can lead to account takeover, session or cookie theft, and traffic redirection to attacker-controlled domains (Agrawal & Wang, 2018; Jiang et al., 2020). The analysis emphasized that any web application, regardless of the programming language used, is susceptible to XSS if proper input validation is not enforced. XSS threats are particularly dangerous when dynamically generated scripts are tampered with by users. Common types of XSS attacks include mutation-based and Document Object Model

(DOM)-based exploits, both of which compromise the integrity and security of cloud-hosted applications (Chen, Nshimiyimana, Damarjati, & Chang, 2021).

The study identified key mitigation techniques for Cross-Site Scripting (XSS) attacks, emphasizing that technical controls must be paired with user awareness. Effective measures include input filtering, HTML entity encoding, data sanitization, Content Security Policy (CSP) enforcement, and data validation. Additional escaping methods, such as attribute value, JavaScript, URL, and CSS escaping, further reduce XSS risks (Sahoo & Gupta, 2019; Gupta & Gupta, 2016).

4.3 Data Analysis of Techniques Securing Data in Public Cloud Computing

4.3.1 Blowfish Algorithm

The study found that the Blowfish algorithm, developed by Bruce Schneier in 1993 (Anwar, Hasan, Hasan, Loren, & Hossain, 2019), is a symmetric cipher that uses identical keys for encryption and decryption. It has a block size of 64 bits (Valmik & Kshirsagar, 2014) and supports key lengths ranging from 32 to 448 bits, with variations of up to 14 rounds (Kumar, Thakur, & Kalia, 2011). Blowfish consumes approximately 5 KB of memory and is considered one of the fastest block ciphers developed, allowing widespread use without concerns over copyright or patent restrictions (Acharya, Sajwan, & Bhargaya, 2013).

The algorithm consists of two parts: key expansion and data encryption. A key of up to 448 bits is expanded into multiple smaller keys totaling 4168 bytes. The most common configuration uses 16 rounds, each relying on key arrangement and substitution using the same data. Operations include XORs and additions on 32-bit words, along with data lookups using four indexed arrays (Nie & Zhang, 2009). Due to its static key structure, Blowfish is suitable for applications like database security and internet commerce. It performs efficiently

on 32-bit microprocessors and is considered a rapid cipher due to its simple round structure (CommonLounge, 2020). However, Blowfish does not support authentication or non-repudiation, and its key schedule is computationally intensive.

4.3.2 Twofish Algorithm

The study also found that Twofish is an enhancement of Blowfish. It is a symmetric key encryption method that uses the same key for both encryption and decryption. Twofish incorporates precomputed key-dependent S-boxes, which are essential components of any block cipher algorithm (Mandal & Singh, 2021). It supports key sizes from 128 to 256 bits and is free to use, fast, flexible, and secure. No successful cryptanalysis of Twofish has been reported, and encryption methods using 128-bit keys or more are theoretically resistant to brute-force attacks. Applications such as 96Crypt by eRightSoft, KeePass, GnuPG, and PGP utilize Twofish (Mandal & Singh, 2021).

Twofish performs well across various hardware platforms and was designed to accommodate performance tradeoffs in speed, key setup, memory usage, and hardware gate count. This makes it a highly adaptable algorithm suitable for diverse cryptographic applications (Mandal & Singh, 2021).

4.3.3 Homomorphic Encryption Algorithm

The study established that Homomorphic encryption is a complex cryptographic technique that allows computations on encrypted data without revealing the plaintext (Alqarni, 2021). This capability is vital for cloud computing, where sensitive data is processed and stored remotely. Homomorphic encryption enables users to perform operations like addition and multiplication on ciphertexts, producing encrypted results that match the output of plaintext operations. This ensures confidentiality and supports meaningful computation in untrusted

environments. Given that cloud users must trust third-party providers with sensitive data, homomorphic encryption offers a critical layer of privacy protection (Sharma, 2024).

4.3.4 Authentication Mechanisms

The study identified authentication as a critical technique for securing data in public cloud computing, encompassing methods such as passwords, smart cards, digital signatures, fingerprints, and other biometrics (Sepehri-Rad, Sadjadi & Sadi-Nezhad, 2019). Authentication mechanisms are categorized into Single-Factor Authentication (SFA), Two-Factor Authentication (TFA), and Multi-Factor Authentication (MFA). SFA relies on a single element like a PIN and is vulnerable to attacks such as brute force, shoulder surfing, and impersonation (Rouse, 2017; Rahav, 2018). TFA enhances security by requiring two elements—typically something the user knows and something they have or are—but still faces limitations against sophisticated threats (Rahav, 2018). MFA further strengthens access control by combining multiple factors, such as a password, a biometric trait, and a physical token. However, MFA is not immune to attacks like shoulder surfing, which can be mitigated through techniques such as facial biometric authentication and the use of One-Time Passwords (OTPs) (Kaka, Ishaq, & Ojeniyi, 2020).

4.3.5 Steganographic Mechanisms

Finally, the study identified steganography as a technique for securing data by hiding it within a cover medium to avoid detection (Kadhim, Premaratne, Vial, & Halloran, 2019). Steganographic methods are evaluated based on imperceptibility, embedding capacity, security, robustness, and computational complexity (Kadhim et al., 2019; Hussaina, Wahaba, Idris, Antony & Jung, 2018). Imperceptibility ensures that the hidden message cannot be detected by the human eye. Metrics such as Mean Squared Error (MSE), Peak Signal-to-Noise Ratio (PSNR), and entropy are used to assess this property.

4.4 Development of LSB-PHE Data Protection Algorithm

The development of LSB-PHE algorithm was first visualized by a conceptual design where two established techniques were merged to produce one robust hybrid algorithm. The purpose of Paillier encryption was to encrypt plaintext data before it is embedded into an image and thus provide confidentiality. On the other hand, the purpose of LSB steganography was to provide concealment of encrypted data thus contributes covert communication. This dual-layered approach infers that even if the image is intercepted, the attacker will be able to see only pixels, while extraction of ciphertext implies that a key will be required for readability, as detailed in Figure 4.1 and Table 4.1 respectively.

Table 4.1: LSB-PHE Data Protection Algorithm

LSB-PHE Hybrid Algorithm	
Step 1:	Enter data to be Encrypted
Step 2:	Apply-PHE Encryption
Step 3:	Upload Cover Image
Step 4:	Embed Encrypted data
Step 5	Upload Stego Image
Step 6:	Decode Ciphertext
Step 7:	Decrypt data

The LSB-PHE algorithm initiates execution by first generating two public keys (n,g) used for encryption and two private keys (lambda, mu) used for decryption. When input of plain data is done in the system, public keys are used to encrypt the data resulting to a ciphertext. The next step is to upload a cover image which is used to embed ciphertext to generate a stego image. In order to retrieve plaintext, the stgo image is uploaded into the system. Here decoding is done in which the ciphertext is retrieved from the stego image. Finally, private keys (lambda, mu) are applied to decrypt and get the original plaintext.

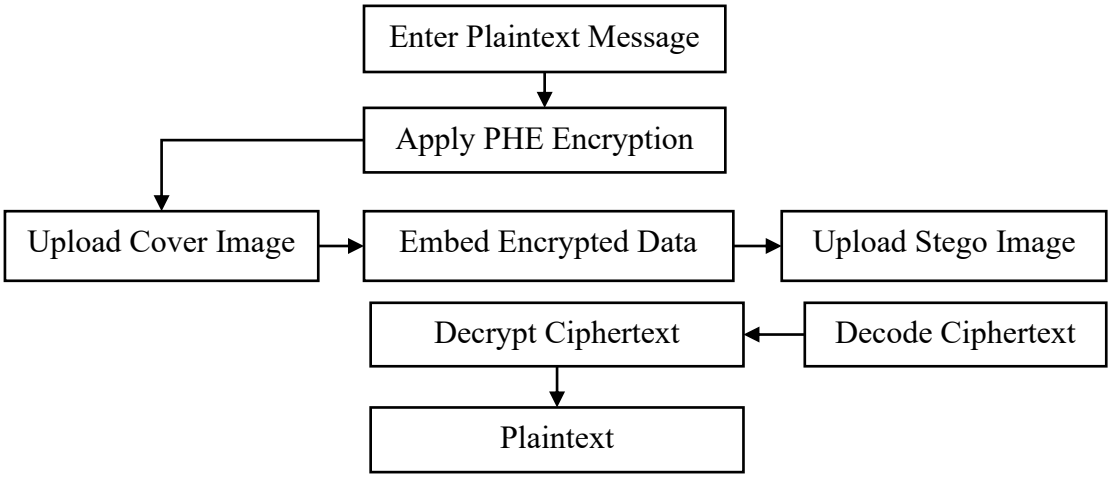

Figure 4.1: Conceptual Design of LSB-PHE Algorithm (Researcher, 2025)

Figure 4.1 portray a conceptual model on how data is encrypted and decrypted in the LSB-PHE algorithm. Figure 4.2 depict a Graphical User Interface (GUI) of LSB-PHE algorithm.

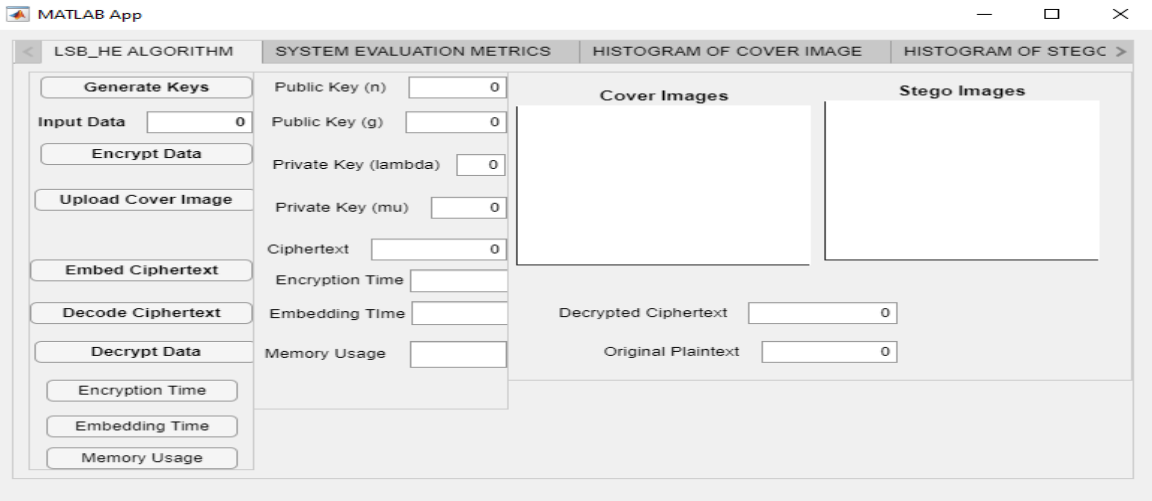

Figure 4.2: GUI of LSB-PHE Data Protection Algorithm (Researcher, 2025)

The GUI contains User Interface Axes (UIAXes) which help in visual data presentation and images display. The GUI also contains different buttons which are programmed to respond when a user interacts with a component. It is the main interface of the algorithm. Figure 4.3 present algorithm evaluation metrics.

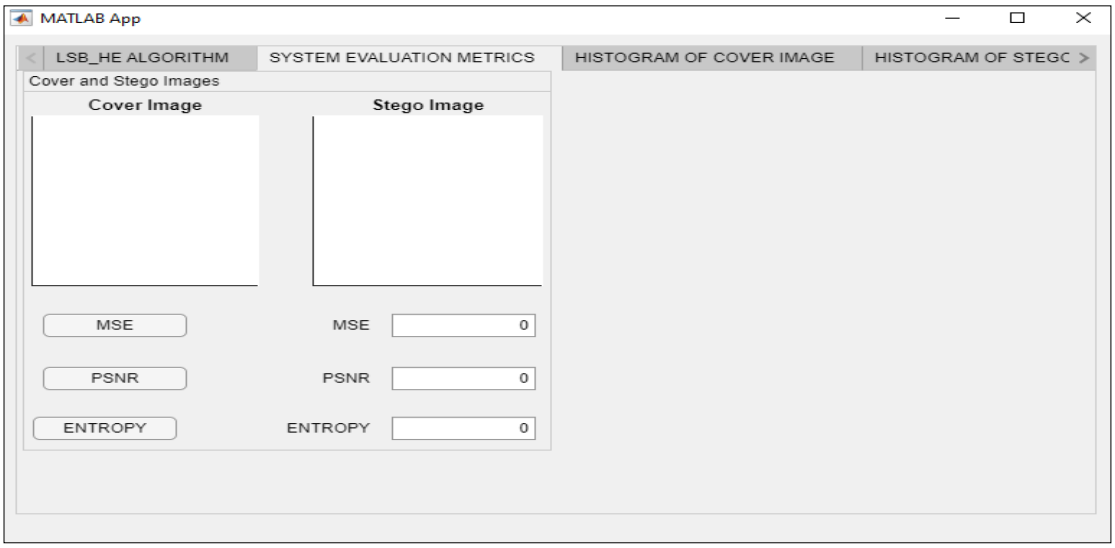

Figure 4.3: GUI of System Evaluation Metrics (Researcher, 2025)

This interface is responsible for displaying the cover and stego images and for calculating MSE, PSNR and Entropy of LSB-PHE data protection algorithm.

4.5 Implementation

LSB-PHE algorithm was implemented in MATLAB App designer with successful simulation which involved key generation, data input, encryption and embedding of data, and decoding and decryption of data.

4.5.1 Development Environment

LSB-PHE data protection algorithm was developed on MATLAB R2021b version that was installed in windows 11 Operating System (OS) with Core i7 processor and 16 GB Random Access Memory (RAM)

4.5.2 Pre-processing

A numeric data was used for simulation tests. Before encryption of data, the algorithm first generates the public keys (n, g) which are used to encrypt the numeric data. Color images

with varied pixel resolution were selected for image quality metrics using MSE, PSNR, and Entropy to analyze if they are suitable in formulating the LSB-PHE data protection algorithm.

4.5.3 PHE Integration

Integration was done by first generating the public keys for encryption and private keys for decryption. Secondly, numeric data is entered into the system and application of paillier encryption is done to generate ciphertext. Thirdly, the ciphertext is embedded into the Red-Green-Blue colour channels of the cover image using LSB substitution technique. Fourthly, extraction and decoding are applied on the stego image to produce the embedded ciphertext. Finally, Paillier decryption function is applied to retrieve plaintext.

4.5.4 LSB Embedding Process

Embedding process is initiated by conversion of ciphertext into a fixed-length binary string using decimal-to-binary function. The binary numbers are then inserted into LSBs of the RGB cover image. The Cover image is then uploaded into MATLAB GUI. A stegancoder function is developed which is responsible for inserting each bit of the ciphertext sequentially into the LSBs of the pixel channels.

4.5.5 Extraction and Decryption Process

Extraction is initiated by uploading the stego image. This image is converted into 8-bit integer format and reshaped into linear array to facilitate bit-level access. The pixel values are sequentially examined to reconstruct the embedded binary stream. The decryption phase follows, where the ciphertext is processed using the Paillier decryption algorithm. This involves retrieving the original primes p and q, computing the private key components, and applying modular arithmetic to recover the plaintext. The decrypted value is then displayed in the user interface, completing the secure retrieval of the hidden message.

4.5.6 MATLAB Code Snippets

1) Key Generation using Paillier Cryptosystem

```
p = 17;
q = 23;
where
“n = p * q;
lambda = lcm (p-1, q-1);
% Choose generator g and validate
valid_g_found = false;
while ~valid_g_found
    g = randi([2, n-1]);
    if gcd(g, n) == 1
        L = @(x) floor ((x - 1) / n);
        L_val = L (mod (powermod (g, lambda, n^2), n^2));
        if gcd (L_val, n) == 1
            valid_g_found = true;
        end
    end
end
mu = modInverse (L_val, n);
```

2) Paillier Encryption of Numeric Plaintext

```
plaintext = 42; % Example numeric value
r = randi ([1, n-1]);
while gcd (r, n) ~= 1
    r = randi ([1, n-1]);
end
ciphertext = mod (powermod (g, plaintext, n^2) * powermod (r, n, n^2), n^2)”;
```

3) Embedding Ciphertext into RGB Cover Image via LSB

```
binaryData = dec2bin (ciphertext, 32); % Convert to binary
img = imread('coverImage.png'); % Load RGB image

stegoImg = stegancoder(img, binaryData); % Custom function handles embedding
inwrite (stegoImg, 'stegoImage.bmp'); % Save stego-image in BMP format
```

4) Extracting and Decoding Ciphertext from Stego-Image

```
stegoImg = imread('stegoImage.bmp');
flatImg = reshape (stegoImg, [], 3); % Flatten RGB for bit access

% Extract LSBs to reconstruct binary string
extractedBits = '';
for i = 1:32
```

```
    channel = mod (i-1, 3) + 1;
    pixelIdx = ceil (i / 3);
    extractedBits(i) = num2str (bitget (flatImg (pixelIdx, channel), 1));
end
decodedCiphertext = bin2dec(extractedBits);
```

5) Paillier Decryption Function

```
function plaintext = paillierDecrypt (ciphertext, p, q, g)
  “
 n = p * q;
    lambda = lcm (p-1, q-1);
    L = @(x) floor ((x - 1) / n);
    mu = modInverse (L (mod (powermod (g, lambda, n^2), n^2)), n);
    plaintext = mod (L (powermod (ciphertext, lambda, n^2)) * mu, n);
end”
```

4.5.7 Performance Considerations

The algorithm’s performance was carefully managed to balance computational efficiency with cryptographic robustness. The use of symbolic computation in MATLAB for modular exponentiation and key generation, particularly in the Paillier encryption and decryption routines, ensured precision but introduced computational overhead, especially for larger ciphertexts. To mitigate this, the implementation employed optimized helper functions such as powermod for efficient modular exponentiation and modInverse for computing modular inverses. Random number generation and coprimality checks were streamlined using bounded integer ranges and loop constraints. Additionally, the LSB embedding and extraction processes were designed to operate directly on image matrices using vectorized operations and bitwise manipulation, minimizing memory usage and execution time.

4.5.8 Validation

Numeric input values were encrypted to generate ciphertext. The ciphertext was then embedded into the LSB of cover images to generate stego images. When plaintext is extracted, it was compared with the original data input to confirm similarity.

4.6 Data Analysis

4.6.1 Visual Quality Metrics

4.6.1.1 MSE Analysis

MSE analysis was done across six standard test images. The MSE values remained notably low, ranging from 0.000023 (Peppers, Sailboat) to 0.00012 (Couple, House), indicating minimal deviation between cover and stego images.

4.6.1.2 PSNR Analysis

Correspondingly, PSNR values were exceptionally high, exceeding 87 dB in all cases and peaking at 94.57 dB for Peppers and Sailboat, which reflects excellent visual fidelity.

4.6.1.3 Entropy Analysis

Entropy values, which assess the randomness and information content, hovered between 6.4207 and 7.5937, with Peppers and Sailboat again exhibiting the highest values, suggesting preserved complexity post-embedding. Collectively, these results confirm that the algorithm achieves high-quality steganography, preserving both image integrity and data entropy while securely concealing encrypted information as shown in Table 4.2.

Table 4.2: MSE, PSNR and Entropy Results (Researcher, 2025)

Image	Value of MSE	Value of PSNR	Value of Entropy
Airplane	0.000038	92.35	6.7025
Female	0.000096	88.55	7.0525
House	0.00012	87.3	6.4961
Couple	0.00012	87.3	6.4207
Peppers	0.000023	94.57	7.5937
Sailboat	0.000023	94.57	7.4842

Table 4.2 show values for MSE, PSNR, and Entropy. Figure 4.4 portray image name against image quality metrics in a graph.

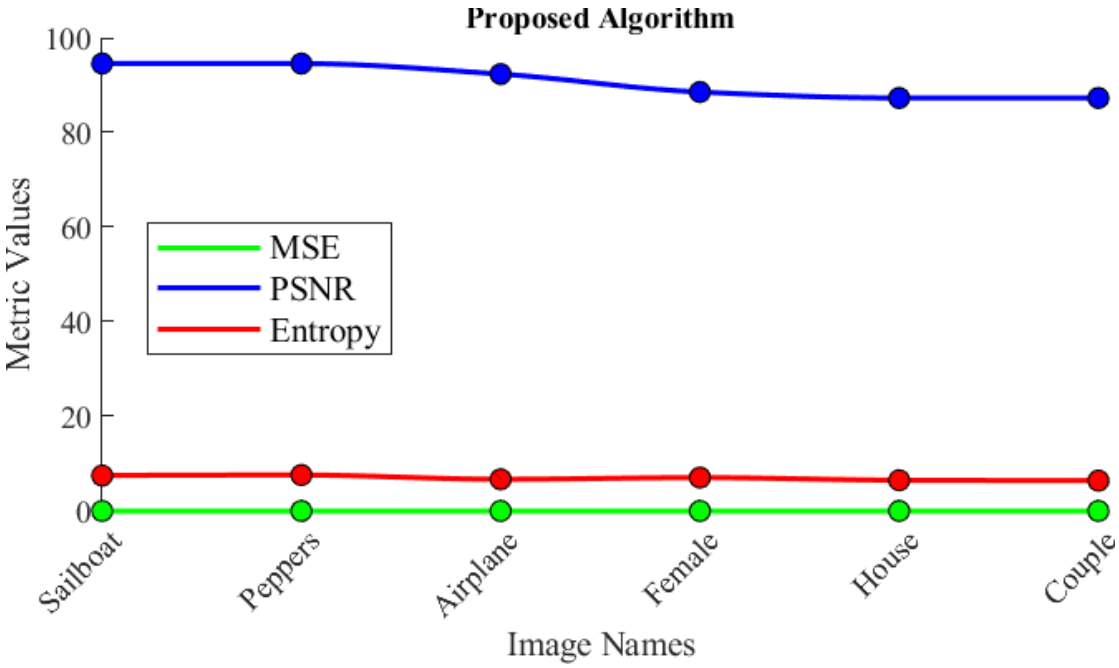

Figure 4.4: Proposed LSB-PHE Data Protection Algorithm (Researcher, 2025)

Among all the tested images, the airplane image demonstrated low MSE value of 0.000038 and high PSNR value of 92.35 decibel (dB) inferring excellent imperceptibility in the process of embedding ciphertext to the cover image. The entropy of airplane image was 6.7025 which imply that the complexity of the image and randomness of ciphertext was preserved, which informs that it is robust to statistical detection techniques.

The Female image generated MSE value of 0.000096, PSNR value of 88.55 dB, and entropy value of 7.0525. This implies that any changes introduced to the image during the embedding process were rarely noticeable. Entropy value of 7.0525 infers that the complexity of the image was maintained after embedding the ciphertext. This suggests that the algorithm can effectively conceal data while preserving visual quality and statistical randomness.

The House image disclosed MSE value of 0.00012 and PSNR value of 87.3 dB. This implies that the image had slightly higher embedding noise when compared to the other images used in the test simulations. Entropy value of the image was 6.4961 which imply a moderate level of embedded information complexity and thus preserved statistical richness. This infers that the algorithm maintained visual as well as structural integrity.

The Couple image attained exactly the same MSE and PSNR values like that of the House image. Entropy values were 6.4207. Even though it was lower, reflecting a uniform texture, the image remains unchanged. This implies that the algorithm maintained its statistical integrity. This infers that the algorithm is robust and can conceal data while maintain perceptual and structural precision.

The Peppers achieved low MSE value of 0.000023 and PSNR value of 94.57 dB. This implies that the image had exceptional visual preservation after embedding ciphertext to the image. Entropy value was 7.5937. This high entropy value indicates that the complexity of the image remained intact or slightly changed. This infers that the algorithm is effective in embedding data, while maintaining visual and statistical transparency.

The Sailboat image posted MSE value of 0.000023 and PSNR value of 94.57 dB, just like the Peppers image. These values indicate that there were no visual changes to the image. Entropy value was 7.4842 which imply that the embedding process was rich in informational complexity. This infers that the Sailboat image is suitable for secure steganographic applications, which offers minimal visual or statistical trail while efficiently concealing embedded data.

The developed algorithm was evaluated using MSE, PSNR, and Entropy with Peppers and Sailboat posting MSE and PSNR values of 0.000023 and 94.57 dB respectively which concludes that the images had perceptual integrity after embedding the ciphertext in the images. The Airplane image had low MSE and high PSNR values of 0.0000338 and 92.35 dB and maintained high entropy values. Female, House, and Couple images showed higher MSE values of up to 0.00012 and lower PSNR values. Their entropy values were above 6.4. These results illustrate that embedding of data was effective with minimal perceptual effect on the images.

4.6.2 Statistical Analysis

4.6.2.1 Histogram Analysis

This study compared the histograms of cover images and stego images visually and statistically and computed the Chi-Square distance, Correlation coefficient and Entropy.

The Chi-Square distance quantifies the difference in pixel intensity distributions between cover and stego images; lower values signify minimal statistical disturbance, a key indicator of subtle embedding. The correlation coefficient evaluates the degree of structural similarity, with high values confirming that pixel patterns remain largely unchanged after embedding, thereby maintaining spatial coherence. Additionally, entropy reflects the image's complexity and information content, preserved or slightly increased entropy values imply that the embedding process retains or even enhances data randomness, offering resistance against entropy-based steganalysis. Together, these metrics affirm that LSB-PHE data protection algorithm achieves both visual transparency and secure, undetectable data hiding. The following Tables depict computations of cover and stego images with their corresponding histograms.

Table 4.3: Comprehensive Image Quality Metrics (Researcher, 2025)

Image	Chi -Square (R)	Chi -Square (G)	Chi -Square (B)	Corr. Coeff (R)	Corr. Coeff (G)	Corr. Coeff (B)	Entropy (Cover)	Entropy (Stego)	Entropy (Change)
Airplane	0.0246	3.5960	0.0175	1.0000	1.0000	1.0000	6.7025	6.7025	0.0000
Couple	0.0295	0.0144	2.0064	1.0000	1.0000	1.0000	6.4207	6.4207	0.0000
Female	2.0063	3.0072	2.0070	1.0000	1.0000	1.0000	7.0525	7.0525	0.0000
House	0.0069	0.0086	0.0034	1.0000	1.0000	1.0000	6.4961	6.4961	0.0000
Peppers	0.0087	0.0141	0.0026	1.0000	1.0000	1.0000	7.5937	7.5937	0.0000
Sailboat	0.0064	0.1108	0.0307	1.0000	1.0000	1.0000	7.4842	7.4842	0.0000

Table 4.3 illustrates that across all tested images; Sailboat, Peppers, House, Female, Couple, and Airplane, the histogram metrics consistently point to a highly effective and imperceptible steganographic process. Chi-square distances remained low in most channels, with only a few moderate deviations (notably in the green channel for Female and Airplane), indicating minimal disruption to pixel intensity distributions. Correlation coefficients were uniformly 1.0000 across all RGB channels, confirming that the overall structural and frequency patterns of the images were perfectly preserved. In terms of entropy, most images showed either no change or negligible increases (on the order of 0.0000 to 0.0001), suggesting that the embedded data had no significant impact on the statistical complexity or information content of the images. Collectively, these findings validate the robustness and stealth of the LSB-PHE data protection algorithm, ensuring strong visual fidelity, high statistical consistency, and resilience against histogram- or entropy-based steganalysis. Figure 4.5 an analysis of the cover and stego images and their generated histograms.

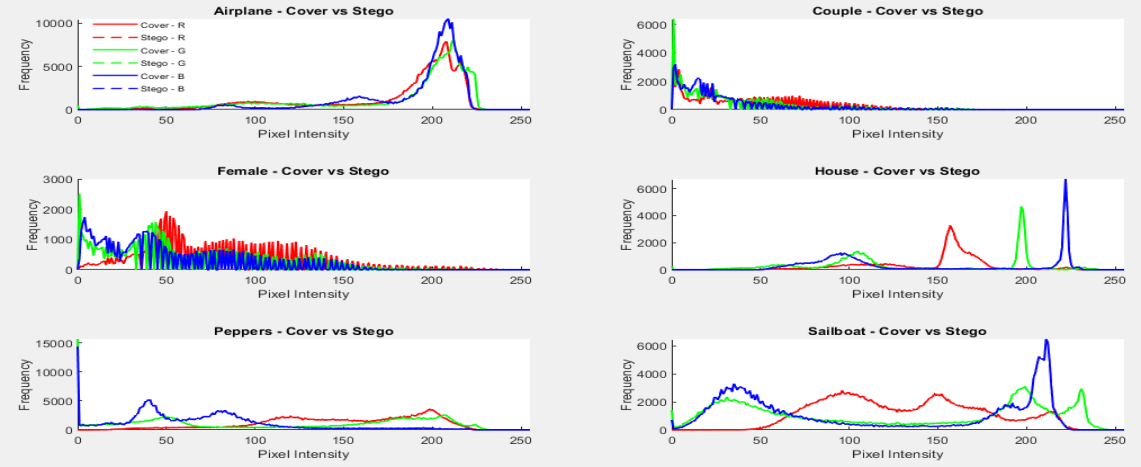

Figure 4.5: Histogram Analysis of Cover Image and Stego Image (Researcher, 2025)

Figure 4.5 demonstrates histogram analysis of 6 cover and stego images and their histograms. The histogram analysis for the Airplane image reveals a strong similarity between the cover and stego versions, both visually and statistically. The red (0.0246) and blue (0.0175) channels exhibit very low chi-square distances, indicating negligible alterations in pixel intensity distributions. Although the green channel shows a slightly elevated chi-square value of 3.5960, suggesting that data embedding may have been more concentrated there, the correlation coefficients for all channels remain perfectly aligned at 1.0000, confirming full preservation of histogram structure. Additionally, the entropy for both images is unchanged at 6.7025, highlighting that the embedding process maintained the image's informational complexity.

The similarity analysis of the Female image show that even though the total distribution of pixel values remails exactly aligned for cover and stego images from correlation coefficients of 1.0000 across all RGB channels, there are perceivable value deviations that can be observed in the green channel with a marginal with towering cui-square value of 3.0072. The slight changes indicate that modifications were introduced during the embedding process. The

overall algorithm indicates that there was no statistical imperceptibility, which infers that it preserved its original texture and complexity.

When the histogram of the House image was analyzed, it discloses a strong correlation between the cover and stego images. The chi-square intervals between the RGB channels were below 0.01 which implies that they preserved statistical consistency of pixel values. The correlation coefficient of the value 1.0000 in all the RGB channels affirm that the image's original texture was preserved and therefore no significant distortion was introduced after embedding process. Further, entropy values of 6.4961 for the cover image and 6.4962 for the stego image were indistinguishable and thus making it unpredictable for data hiding.

The Couple Image's histogram shows that the cover and stego image were closely same. The chi-square values of the RGB channels were 0.0295, 0.0144, and 2.0064 respectively. This implies that there were slight changes in pixel intensity distributions with the blue channel reflecting a higher value than the rest. Correlation coefficient of the value 1.0000 in all the three channels affirm that no distortions were introduced after embedding data on the cover image. Entropy values of 6.4207 for the histograms informs that statistical distribution of pixel intensities remained unchanged. The findings point out that the algorithm camouflages eclipsed data such that it is invisible to the human visual system and difficult to detect.

Analysis of the histogram of Peppers image present significant resemblance between the cover and stego image. The chi-square values in all the RGB channels were 0.0087, 0.0141, and 0.0026 respectively, indicating minimal imperceptible deviations in pixel intensity distributions. Correlation coefficients of 1.0000 in all the RGB channels informs that the hidden data in the histogram of cover image is statistically and visually indistinguishable from

histogram of stego image. Notably, both images preserve same entropy values of 7.5937 and thus exhibiting that the hidden data is invisible both to the human visual system and to analysis.

Presentation of analysis of histogram of Sailboat results close similarity in semblance and pixel-level distribution between the stego and cover images. Chi-square values for RGB channels were 0.0064, 0.1108, and 0.0307 respectively which signals negligible deviations. The correlation coefficients of the value 1.0000 in all the RGB channels indicates preservation of the structural shape of the histograms. Notably, both images preserve same entropy values of 7.4842 and thus exhibiting that the hidden data is invisible both to the human visual system and to analysis.

Histogram analysis of the LSB-PHE data protection algorithm demonstrates that histograms generated from the cover and stego images had no visible distortions or quality loss, embedding process is not visible to statistical analysis and appeared the same to the human visual system. This can be proved by their uniform correlation coefficients of 1.0000 across all RGB channels. Additionally, entropy values remained the same in all the images, which infers that the algorithm is robust and preserves the original texture and pixel-level unpredictability.

4.6.3 Analysis of LSB-PHE Data Protection Algorithm with Baseline Algorithms

This study sought to evaluate the developed LSB-PHE data protection algorithm against other already developed algorithms such as Least Significat Bit (LSB), Least Significant Bit-Advanced Encryption Standard (LSB-AES), Secret Map-Enhanced LSB, and Least

Significant Bit-Homomorphic Encryption (LSB-PHE) algorithms as demonstrated in Figure 4.6.

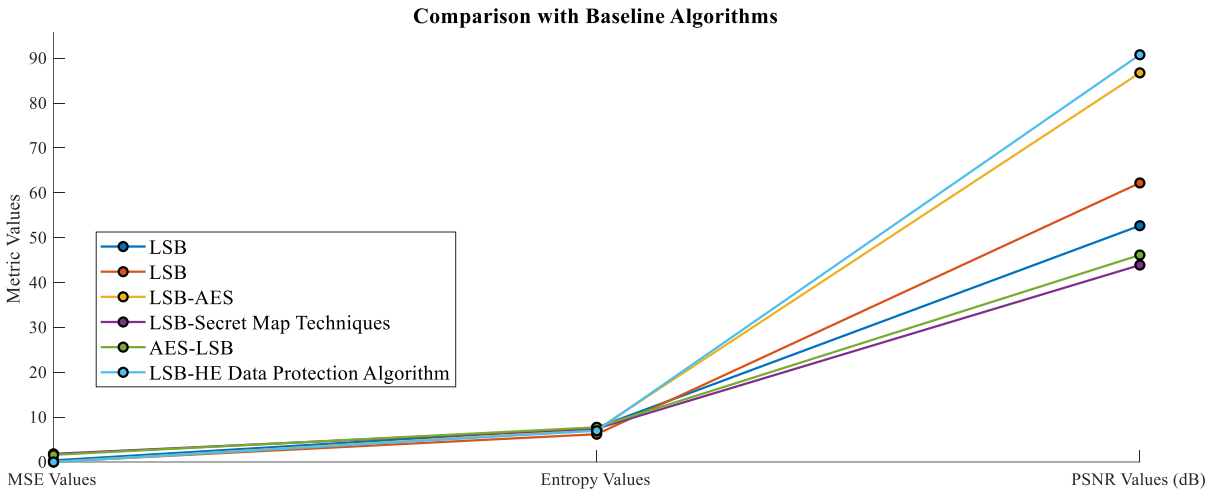

Figure 4.6: Comparison with Baseline Algorithms (Researcher, 2025).

Figure 4.6 illustrates a comparison of baseline algorithms with the proposed algorithm. Ganesh, Nagaraj, and Shankar (2020) employed a traditional LSB technique that yielded moderate imperceptibility, demonstrated by an MSE of 0.3545 and a PSNR of 52.67, indicating visible degradation in the stego image. Nonetheless, the method achieves high entropy (7.6595), suggesting robust data security. By comparison, the proposed LSB-PHE algorithm drastically reduces distortion with a remarkably low MSE of 0.000023 and substantially improves visual quality with a PSNR of 94.57. Despite a slightly lower entropy of 7.5937, it maintains a solid security threshold. This positions LSB-PHE as a more refined solution that optimally balances imperceptibility, visual fidelity, and information-theoretic security.

Alexan, Hamza, and Medhat (2019) presented an LSB-based technique that delivers solid improvements in visual quality, evidenced by a high PSNR of 62.2 and a relatively low MSE

of 0.07817. These results reflect an advancement over conventional LSB methods in terms of imperceptibility. However, their algorithm records a comparatively low entropy of 6.193395, which may signal vulnerabilities in security. In contrast, the proposed LSB-PHE algorithm builds on these strengths while addressing the limitations: it achieves a substantially lower MSE, elevates the PSNR to 94.57, and enhances entropy to 7.5937. This refined performance underscores LSB-PHE's ability to offer superior image fidelity and improved information concealment, making it a more robust alternative in both imperceptibility and data security.

Alabaichi, Al-Dabbas, and Salih (2020) proposed an LSB-based Secret Map technique that introduces spatial variability, effectively enhancing the entropy of the stego image (7.4418), a testament to its information-theoretic security. However, this improvement in randomness comes at the cost of perceptual quality, as evidenced by a high MSE of 1.8133 and a modest PSNR of 43.8953, indicating substantial image degradation. In contrast, the proposed LSB-PHE algorithm represents a significant advancement by achieving a dramatically lower MSE and elevating PSNR to 94.57, thereby preserving visual fidelity to a far greater extent. Although its entropy (67.5937) is marginally lower, it remains within a robust security threshold, highlighting LSB-PHE as a more balanced and efficient steganographic strategy.

Shwaysh, Alani, Saad, and Abdulhussein (2024) proposed an AES-LSB hybrid approach that exhibits remarkable strength in terms of entropy, achieving the highest recorded value among all benchmarked methods (7.7286). This elevated entropy underscores the algorithm's strong encryption-driven randomness and potential for secure data hiding. However, this advantage is offset by reduced imperceptibility, as indicated by a relatively high MSE of 1.5933 and a modest PSNR of 46.1383, both suggesting noticeable distortion in the stego image. In contrast, the proposed LSB-PHE algorithm effectively rebalances this trade-off by

maintaining a respectable entropy level (6.4207 to 7.5937) while significantly enhancing image quality and minimizing distortion. With its superior PSNR (87.3 to 94.57) and exceptionally low MSE, LSB-PHE stands out as a more refined and visually unobtrusive steganographic solution that does not compromise core security principles.

LSB-PHE data protection algorithm records the lowest MSE (0.000023) and highest PSNR (94.57) among the compared models, indicating near-flawless preservation of image quality. Although its entropy value (7.5937) is marginally lower than those of more encryption-intensive algorithms, it remains well within a secure and effective range. This positions LSB-PHE as a refined and balanced solution that advances the steganographic trifecta, imperceptibility, image clarity, and data security, beyond the trade-offs faced by conventional methods.

4.6.4 Complexity of LSB-PHE Algorithm

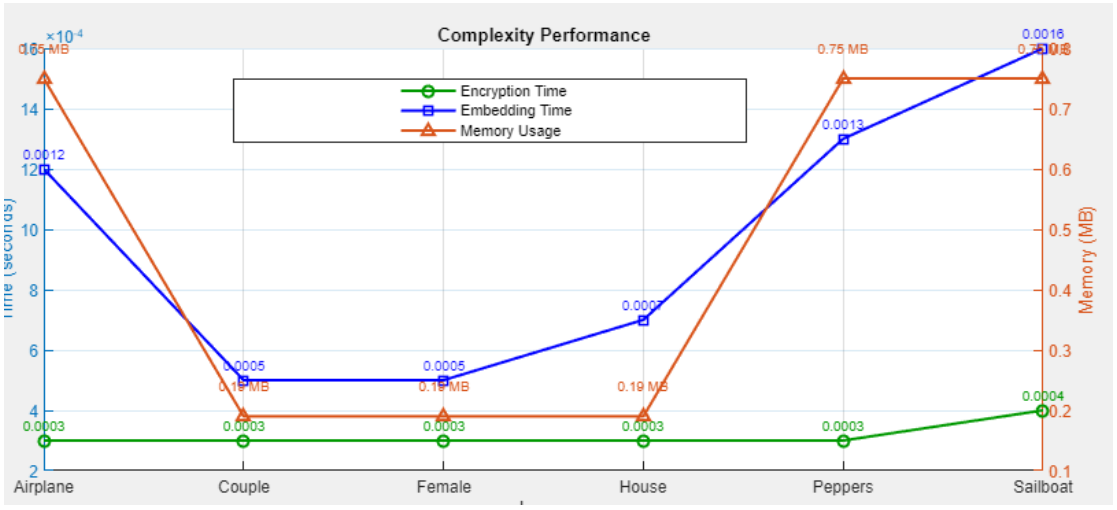

Figure 4.7: Complexity of LSB-PHE Algorithm

Figure 4.7 demonstrate performance metrics of time and space complexity of LSB-PHE algorithm which infer the algorithm effectively combines cryptographic robustness with steganographic agility. Encryption time remains highly consistent across all six test images,

averaging approximately 0.0003167 seconds, with only a marginal increase observed in the Sailboat image (0.0004s). This consistency suggests that the Paillier encryption component is computationally efficient, likely benefiting from optimized key sizes and streamlined modular operations. In contrast, embedding time varies more noticeably, from 0.0005 to 0.0016 seconds, highlighting the influence of image complexity and resolution on LSB substitution.

Memory utilization in images such as Couple, Female and House stood at 0.19 MB while Airplane, Peppers, and Sailboat was 0.75 MB. The differences in memory utilization among the images arise because of image size, and color intensities. These factors impact on the embedding capacity and resource consumption. In general, the algorithm demonstrates excellent scalability, maintaining minimal encryption overhead while adapting efficiently to diverse image profiles. This positions the approach as a practical solution for secure data hiding in resource-constrained environments.

CHAPTER FIVE

DISCUSSION OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS

5.1 Introduction

In this chapter, the following are presented; security threats and attacks on data in public cloud computing, techniques securing data in public cloud computing, development of LSB-PHE data protection algorithm, evaluation of LSB-PHE data protection algorithm, conclusions, recommendations, and suggestions for further research.

5.2 Security Threats and Attack Vectors on Data in Public Cloud Computing

The study confirms that public cloud computing is vulnerable to multiple threats, many of which align with prior research. For instance, insecure APIs—identified here as arising from poor design and misconfiguration—are consistent with findings by Jensen et al. (2009), who emphasized that APIs are often the weakest link in cloud interfaces. This reinforces the need for secure API gateways and continuous monitoring.

SQL injection attacks, as validated in this study, remain a persistent threat. This aligns with Halfond et al. (2006), who documented the widespread impact of SQLi on web applications. However, this study extends the discussion by emphasizing the role of input sanitization in cloud-specific contexts, where multi-tenant architectures amplify the risk.

The identification of XSS vulnerabilities echoes the work of Gupta & Gupta (2018), but this study adds nuance by linking XSS to session hijacking and traffic redirection in cloud-hosted applications. This suggests that cloud platforms require stricter client-side script controls than traditional web environments.

Legal and compliance risks are also highlighted, particularly in cross-border data flows. This finding supports Pearson & Benameur (2010), who warned that jurisdictional ambiguity can

undermine cloud adoption. The study's emphasis on standardized frameworks adds practical urgency to their theoretical concerns.

Insider threats, confirmed here as a major risk, are consistent with Greitzer & Frincke (2010), who argued that behavioral monitoring is essential. This study contributes by proposing a hybrid approach combining IAM systems with organizational policies, bridging technical and human-centered strategies.

Social engineering and malware injection are shown to exploit user unawareness—an insight that complements the behavioral focus of Mouton et al. (2016). By linking these attacks to phishing and payload delivery, the study reinforces the need for user training and endpoint security.

DoS and MITM attacks are discussed in line with Tripathi & Hubballi (2018), but this study adds specificity by describing how attackers route traffic through personal access points. This operational detail enhances understanding of attack vectors in mobile banking scenarios.

5.3 Techniques Securing Data in Public Cloud Computing

The study's evaluation of Blowfish and Twofish algorithms confirms their efficiency but highlights limitations in authentication and non-repudiation. This aligns with Schneier (1993) and Mandal & Singh (2021), but the study goes further by arguing that these limitations make them unsuitable for critical cloud applications—a point not emphasized in earlier literature.

Homomorphic encryption is presented as a breakthrough for privacy-preserving computation. This supports Gentry (2009), who introduced the concept, but the study advances the discussion by applying it to real-world cloud scenarios where third-party trust is minimal. This practical framing strengthens its relevance.

Authentication mechanisms are discussed with a focus on MFA. While Rahav (2018) and Ometov et al. (2018) support MFA's effectiveness, this study critiques its vulnerability to shoulder surfing and proposes mitigation through facial biometrics and OTPs. This adds a layer of operational insight not present in prior work.

Steganography is validated as a data-hiding technique, consistent with Kadhim et al. (2019). However, this study contributes by emphasizing evaluation metrics like MSE and PSNR, which are often overlooked in broader discussions. This technical precision enhances the applicability of steganographic methods in cloud contexts.

5.4 Development of LSB-PHE Data Protection Algorithm

The LSB-PHE data protection algorithm was first visualized in a conceptual model in which LSB substitution technique can be combine with PHE to generate one hybrid algorithm that can benefit from the strengths of each technique. The algorithm was developed and implemented using MATLAB R2021b. Specification of the system were; Windows 11 Operating System with Intel Core i7 processor and 16 GB RAM.

System GUI were developed containing different buttons for executing various commands such as key generation, encryption, image uploads, extraction and decoding, decryption, embedding time, and memory. These buttons were programmed such that there corresponding UIAXes would display appropriate data when clicked.

5.5 Evaluation of LSB-PHE Data Protection Algorithm

The LSB-PHE data protection algorithm was thoroughly tested across a range of standard benchmark images, delivering impressive outcomes in both visual fidelity and statistical robustness. Images like Peppers and Sailboat exhibited outstanding performance, achieving

the lowest Mean Squared Error (0.000023) and the highest Peak Signal-to-Noise Ratio (94.57 dB), reflecting excellent preservation of visual quality after embedding. Similarly, Airplane demonstrated strong results with a low MSE of 0.000038, a PSNR of 92.35 dB, and an entropy value of 6.7025, inferring high resilience to statistical analysis and effective complexity retention.

Conversely, images with more uniform textures, such as Female, House, and Couple, showed slightly elevated MSEs (up to 0.00012) and slightly reduced PSNRs (as low as 87.3 dB), though still within imperceptible distortion thresholds. Consistently high entropy values across all images (exceeding 6.4) confirmed that the algorithm maintained essential structural features. Overall, these results affirm the effectiveness of the LSB-PHE algorithm in securely embedding encrypted content while preserving both perceptual quality and statistical integrity, strengthened by the underlying Paillier cryptographic framework.

The histogram-based analysis of the LSB-PHE algorithm across six benchmark images demonstrates its strong imperceptibility and statistical fidelity. Each image preserved perfect correlation coefficients of 1.0000 across all RGB channels, confirming that the underlying histogram structures remained untouched post-embedding. Although slight chi-square variations appeared particularly in the green channels of Airplane (3.5960) and Female (3.0072), these deviations were minor and localized. In contrast, images such as Peppers, Sailboat, and House reported exceptionally low chi-square values, with Peppers showing just 0.0026 in the blue channel, reaffirming the subtlety of the embedding process.

Complementing these results, entropy values for all image pairs remained stable, ranging from 6.4207 to 7.5937, indicating no measurable loss in informational richness or randomness. These consistent patterns affirm the LSB-PHE algorithm’s effectiveness in

embedding encrypted data inconspicuously, preserving both visual integrity and statistical complexity, essential qualities for secure and reliable steganographic applications.

The comparative evaluation underscores the enhanced performance of the LSB-PHE data protection algorithm relative to various traditional and hybrid steganographic approaches. While the techniques introduced by Ganesh *et al.,* (2020) and Alexan *et al.,* (2019) offer strengths such as elevated entropy or improved perceptual quality, they fall short in achieving a balanced integration of imperceptibility and data security. In contrast, the LSB-PHE algorithm demonstrates superior effectiveness by attaining an exceptionally low MSE of 0.000023 and a high PSNR of 94.57, indicating minimal visual distortion. It also preserves a robust entropy level of 7.5937, ensuring a reliable degree of information concealment despite being slightly lower than that of more encryption-heavy solutions.

In comparison to advanced techniques such as LSB-AES, the Secret Map approach Alabaichi *et al.,* (2020), and AES-LSB hybrid algorithm by Shwaysh *et al.,* (2024), the LSB-PHE algorithm consistently achieves higher image quality while maintaining strong data concealment capabilities. While these baseline methods attain elevated entropy levels which indicate enhanced randomness, they often incur higher MSE and reduced PSNR, signaling greater visual distortion. In contrast, LSB-PHE excels by achieving a well-balanced integration of low distortion and high perceptual quality without substantially compromising security. This makes it a refined and dependable choice for steganographic systems that demand both clarity and cryptographic robustness.

The LSB-PHE algorithm exhibits consistently low encryption times across all six test images, averaging 0.0003167 seconds, while embedding time fluctuates based on image complexity, ranging from 0.0005 to 0.0016 seconds. Memory usage reveals two clear groupings: low-

resolution images such as Couple and Female consume only 0.19 MB, whereas higher-resolution images like Airplane and Sailboat demand up to 0.75 MB. These findings underscore the algorithm's flexibility in handling varied image types, making it a strong candidate for secure data embedding in environments with limited computational resources. Further enhancements could be achieved through parallel embedding techniques and intelligent image selection strategies.

5.6 Conclusions

This study set out to investigate security threats to data in public cloud computing, evaluate existing protection techniques, and develop a hybrid algorithm for enhanced data security. The conclusions below are explicitly aligned with each research objective:

The study confirmed that public cloud environments face diverse threats including SQL injection, XSS, insecure APIs, insider threats, social engineering, malware injection, DoS, and MITM attacks. These threats compromise confidentiality, integrity, and availability of cloud-hosted data. The findings reinforce the need for layered security strategies that combine technical controls with organizational policies.

The study evaluated multiple data protection techniques including symmetric encryption (Blowfish, Twofish), asymmetric encryption (RSA, DS), homomorphic encryption, authentication mechanisms (SFA, TFA, MFA), and steganographic methods. While symmetric algorithms offer speed, they lack non-repudiation. Asymmetric and homomorphic encryption provide stronger confidentiality but are computationally intensive. Authentication and data hiding techniques complement encryption by enhancing access control and stealth.

The study successfully developed the LSB-PHE algorithm, combining Paillier Homomorphic Encryption (PHE) for secure encryption and Least Significant Bit (LSB) steganography for covert data embedding. This hybrid approach addresses both confidentiality and concealment in cloud environments.

Simulation results demonstrated that LSB-PHE achieves high imperceptibility, with low MSE and high PSNR values. Entropy values approached the theoretical maximum, indicating strong resistance to statistical detection. Histogram and correlation analysis showed minimal visual and statistical disturbance. Compared to benchmark algorithms, LSB-PHE exhibited superior image quality and robust security performance.

5.7 Recommendations

Based on the study's findings and performance evaluation of the LSB-PHE data protection algorithm, the following targeted recommendations are proposed:

Cloud security practitioners should integrate statistical validation into steganographic workflows by incorporating histogram correlation and chi-square analysis into routine steganographic evaluations to complement visual inspection. This will improve detection resistance and ensure that embedded data remains imperceptible under forensic scrutiny. Additionally, when selecting or designing data hiding techniques, practitioners should use a tri-metric evaluation framework that equally prioritizes imperceptibility, entropy, and distortion. This avoids over-optimization of one metric at the expense of overall security and usability.

Cloud service providers handling medical records, legal documents, or intellectual property should consider implementing hybrid models like LSB-PHE to enhance confidentiality and stealth. This is especially relevant for services offering secure image transmission or archival.

Providers should adopt modular architectures that allow seamless integration of cryptographic libraries (such as Paillier) with native image processing tools. This improves scalability and simplifies maintenance across diverse platforms.

Policy makers and regulatory bodies should require cloud vendors to implement both encryption and concealment mechanisms for high-risk data categories. This dual-layer approach strengthens compliance with privacy laws and reduces exposure to insider and external threats. Policymakers should also promote standardized testing frameworks that include statistical metrics (such as MSE, PSNR, entropy) for validating data hiding techniques used in regulated sectors like healthcare and finance.

5.8 Suggestions for Further Research

Future research should explore the development of an adaptive LSB embedding algorithm using deep learning techniques such as Convolutional Neural Networks (CNNs) or Transformer-based models. These models can be trained to analyze image content—specifically texture complexity and edge density—and dynamically adjust embedding parameters to optimize imperceptibility and resilience against steganalysis. Evaluation should be conducted using benchmark datasets like BOSSbase or ALASKA2, with performance measured through metrics such as PSNR, SSIM, and detection accuracy against tools like StegExpose or SRNet. Deployment could involve integrating the adaptive model into MATLAB or Python-based cloud security toolkits for real-time image protection.

Further research should also investigate the integration of LSB steganography with advanced homomorphic encryption schemes such as Brakerski-Gentry-Vaikuntanathan (BGV) or Cheon-Kim-Kim-Song (CKKS). These schemes support encrypted arithmetic operations and are ideal for domains requiring secure computation, such as health informatics and financial

analytics. Evaluation should involve embedding encrypted data into high-resolution medical images (e.g., DICOM format) and encrypted financial logs, with performance assessed through encryption/decryption time, computational overhead, and post-decryption accuracy. Implementation can be prototyped using libraries like Microsoft SEAL or PALISADE, and tested on cloud platforms such as Azure or AWS with GPU acceleration.

Another promising direction is the deployment of the LSB-PHE algorithm in real-time cloud communication systems, such as secure telemedicine platforms or encrypted messaging services. Research should focus on latency, throughput, and reliability under varying network conditions, using tools like Wireshark and Apache JMeter for simulation. Deployment should leverage containerized microservice architectures (e.g., Docker with Kubernetes) to ensure scalability, fault tolerance, and seamless integration into production environments.

Finally, future work should consider developing a cross-platform implementation of the LSB-PHE algorithm using performance-optimized languages such as Rust or WebAssembly (WASM). This would enable secure data hiding in browser-based cloud applications and mobile environments. Evaluation should compare execution time, memory usage, and compatibility across desktop, mobile, and browser platforms. Deployment could involve packaging the solution as a lightweight SDK or browser extension for integration into client-side cloud applications.

REFERENCES

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

Abunadi, I., Abdullah, M.H., Alotaibi, S.S., Asiri, M.M., Ahmed, H.M., Zamani, A.S., Motwakel, A., & Yaseen, I. (2022). Optimal Multikey Homomorphic Encryption With Steganography Approach for Multimedia Security in Internet of Everything Environment. *Appl.Sci*, vol 12, 4026, https://doi.org/10.3390/app2084026

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

Agrawal, D.P., & Wang, H. (2018) Computer and Cyber Security. Auerbach Publications, New York. Retrieved from https://doi.org/10.1201/97804 29424878 on 11/1/2023 at 10.00 am

Ahmed, M., Li, Y., Waqas, M., Sheraz, M., Jin, D., & Han, Z. (2018). A Survey on Socially Aware Device-to-Device Communications. *IEEE Commun. Surv. Tutor*, vol. 20, pp. 2169–2197.

Acar, A., Aksu, H., Uluagac, A.C., & Conti, M. A. (2019). Survey on homomorphic encryption schemes: Theory and implementation. *ACM Comput. Surv*, PP. 51-79.

Ajoudanian, S., & Ahmadi, M.R. (2012). A Novel Data Security Model for Cloud Computing. *International Journal of Engineering and Technology*, vol. 4, pp. 326–329.

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

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

Alex, C., Creado, G., Almobaideen, W., Alghanam, O.A., & Saadeh, M. (2023). A comprehensive survey for IoT security datasets taxonomy, classification and

machine learning mechanisms, *Comput. Secur*, vol. 132, doi: 10.1016/j.cose.2023.103283.

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.

Aljazaery, I. A., & Alaidi, A. H. M. (2022). Encryption of Color Image Based on DNA Strand and Exponential Factor. *International Journal of Online & Biomedical Engineering*, vol. 18(3), https://doi.org/10.3991/ijoe.v18i03.28021

Alkhamese, A.Y., Shabana, W.R., & Hanafy, I.M. (2019). Data Security in Cloud Computing Using Steganography: A Review. *In Proceedings of the 2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*, Aswan, Egypt, pp. 2–4.

Alou, B., Hasnain, M., Alharbi, A., Alosaimi, W., Alyami, H., & Ayaz, M. (2021). A Systematic Literature Review on Cloud computing security: Threats and Mitigation Strategies, *IEEE Access*, vol. 9, pp. 57792-57807.

Alqarni, A.A. (2021). A secure approach for data integration in cloud using Paillier homomorphic encryption, *Journal of Basic and Applied Sciences*, vol. 5, no. 2, pp. 15-21, 2021.

Alrasheed, S.H., Aied alhariri, M., Adubaykhi, S.A., & El Khediri, S. (2022). Cloud Computing Security and Challenges: Issues, Threats, and Solutions. *5th Conference on Cloud and Internet of Things (CIoT),* Marrakech, Morocco, pp. 166-172, doi: 10.1109/CIoT53061.2022.9766571. https://ieeexplore-ieeeorg.ezaccess.library.uitm.edu.my/document/9766571

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.

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

Amara, N., Zhiqui, H., & Ali, A. (2017). Cloud computing security threats and attacks

With their mitigation techniques. *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC),* pp. 244 – 251, doi: 10.1109/CyberC.2017.37.

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

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

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.

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

AspnesM J. (2017). Notes on computational complexity theory CPSC 468/568 ch. 9. Pp. 58–59

Aspiring Youths (2025). Advantages and Disadvantages of Steganography, retrieved from https://aspiringyouths.com/advantages-disadvantages/steganography/?form=MG0AV3&MG0AV3

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

Babu, S.S., & Vijayalakshmi, Y. (2020). Enhancement of e-commerce security through Asymmetric key algorithm. *Comput. Commun*, vol. 153, pp. 125–134, Doi: 10.1016/j.comcom.2020.01.033.

Baffle (2023). Advantages and Disadvantages of Homomorphic Encryption, retrieved On 1/3/2025 from https://baffle.io/blog/advantages-and-disadvantages-of-Homomorphic-encryption-2023/?form=MG0AV3

Bansal, P., Sharma, B., & Saxena, M. (2016). Low Error Rate Based Secure Sharing of

Personal Health Record in Cloud Computing Using DWT Steganography. *In Proceedings of the 8th International Conference on Computational Intelligence and Communication Networks (CICN), Tehri, India*, 23–25 December, pp. 428–431.

Bettendorf, M. (2021). API growth continues to skyrocket in 2020 and into 2021. https://blog.postman.com/api-growth-rate/.

Bhadauria, R., & Sanyal, S. (2012). Survey on security issues in cloud computing and Associated Mitigation techniques. *Int J Comput Appl*, vol. 47(18), pp.47–66 57.

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.

Bharadiya, J. P. (2023). Artificial Intelligence in Transportation Systems A Critical Review. *American Journal of Computing and Engineering*, vol. 6(1), pp. 34 - 45. https://doi.org/10.47672/ajce.1487

Bharadiya, J. P. (2023). The Impact of Artificial Intelligence on Business Processes. *European Journal of Technology*, vol. 7(2), pp. 15 - 25. https://doi.org/10.47672/ejt.1488 "

Bharadiya, J. P. (2023). Machine Learning in Cyber security: Techniques and Challenges. *European Journal of Technology*, vol. 7(2), pp. 1 - 14. https://doi.org/10.47672/ejt.1486”

Bharadiya, J. P. (2023). A Comprehensive Survey of Deep Learning Techniques Natural Language Processing. *European Journal of Technology*, vol. 7(1), pp. 58 - 66. https://doi.org/10.47672/ejt.1473

Bhardwaj, P. (2019). Types of sampling in research. Journal of the Practice of Cardiovascular Sciences, 5(3), 157. DOI: 10.4103/jpcs.jpcs_62_19

Bolton, T., Dargahi, T., Belguith, S., Al-Rakhami, M.S., & Sodhro, A.H. (2021). On the Security and Privacy Challenges of Virtual Assistants. *Sensors*, vol. 21, p. 2312.

Brownlee, J. (2011). Clever algorithms: nature-inspired programming recipes. Lulu, pp 404–405

Butt, U.A., Mehmood, M., Shah, S.B., Amin, R., Shaukat, M.W., Raza, S.M., Suh, D.Y., & Piran., M.J. (2020). *A Review of Machine Learning Algorithms for Cloud Computing Security, Electronics*, vol. 9, no. 9, p. 1379.

Caviglione, L., Choras, M., Corona, I., Janicki, A., Mazurczyk, W., Pawlicki, M., &

Wasielewska, K. (2021). Tight arms race: Overview of current malware threats and trends in their detection, *IEEE Access*, vol. 9, pp. 5371–5396

Challa, R. (2020). Homomorphic Encryption: Review and Applications. *Advances in Data Science Management.* Springer Nature Singapore Pte Ltd, pg 273-281

Charles, M., & Delgado, B. (2022). Health Datasets as Assets: Block Chain-Based Valuation and Transaction Methods. Retrieved from https://pmc.ncbi.nlm.nih.gov/articles/PMC9907414/ on 16/1/2025 at 4.16 am

Chaudhary, J. K., Sharma, H., Tadiboina, S. N., Singh, R., Khan, M. S., & Garg, A. (2023). Applications of Machine Learning in Viral Disease Diagnosis. *10th International Conference on Computing for Sustainable Global Development* (INDIACom) (pp. 1167-1172). IEEE.

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

Chinnasamy, V. (2022). Bad bots are coming at APIs! How to beat the API bot attacks? Help Net Security. https://www.helpnetsecurity.com/2022/09/12/api-bot-attacks/.

Cloudflare (2025). What is a Cryptographic Key? https://www.cloudflare.com/en-gb/learning/ssl/what-is-a-cryptographic-key/

Communications Authority of Kenya (2024). Cybersecurity Report. Retrieved from ttps://www.ca.go.ke/sites/default/files/202410/Cyber%20Security%20Report%20Q1%202024-2025.pdf on 19/1/2025 at 13.34 pm

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

Cormen, T. H, Leiserson, C.E., Rivest, R. L., & Stein, C. (2001) *Introduction to algorithms*. MIT Press, Cambridge, p 47

Cosseron, O., Hoffmann, C., Meaux, P., Standaert, FX. (2022). Towards Case-Optimized Hybrid Homomorphic Encryption. In: Agrawal, S., Lin, D. (eds). Advances in Cryptography-ASIACRYPT2022. Lecture notes in Computer Science vol 13793 Springer, Cham, https://doi.org/10.1007/978-3-031-22969-5_2

Cramer,R., Damgård, I., & Nielsen, J.B. (2001). Multiparty computation from threshold

homomorphic encryption. In: *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 280–300.

Crihan, G., Cracium, M., & Dumitriu, L.A. (2023). A Comparative Assessment of Homomorphic Encryption Algorithms Applied to Biometric Information, *Inventions,* vol.8, pp. 102, https://doi.org/10.3390/inventions 8040102

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

Derhab, A., Belaoued, M., Guerroumi, M., & Khan, F.A. (2020). Two-Factor Mutual Authentication Offloading for Mobile Cloud Computing. *IEEE Access* vol. 8, pp. 28956–28969.

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

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.

Douligeris, C. (2021). A novel Two-Factor HoneyToken Authentication Mechanism. *International Conference on Computer Communications and Networks IEEE*, pp. 1–7

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.

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

Encryption Consulting (2024). What is Blowfish in Security? Who uses Blowfish? Retrieved from ttps://www.encryptionconsulting.com/education-center/what-is-blowfish/

Esiner, E., & Datta, A. (2019). Two-factor authentication for trusted third party free Dispersed storage. *Future Gener. Comput. Syst*. Vol. 90, pp. 291–306.

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

Eze V.H.U, Uche, K.C.A., Okafor, W.O., Edozie, E., Ugwu, C.N., & Ogenyi, F.C. (2023). Renewable Energy Powered Water System in Uganda: A Critical Review. *Newport International Journal of Scientific and Experimental Sciences* (NIJSES), vol. 3(3), pp. 140-147.

Flores-Carapia1, R., Silva-García1, V.M., Cardona-López, M.A., & Villarreal-Cervantes, M.G. (2025). A chaotic digital signature algorithm based on a dynamic substitution box. Scientific Reports, retrieved from https://doi.org/10.1038/s41598-024-83943-x on 19/1/2025 at 11.30 am

Gadde, S., Amutharaj, J., & Usha, S. (2023). A Security Model to Protect the Isolation of Medical Data in the Cloud Using Hybrid Cryptography", *Journal of Information Security*, vol. 73, pp. 770–777

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).

Gayam, R. R. (2021). Artificial Intelligence in Healthcare: Advanced Algorithms for Predictive Diagnosis, Personalized Treatment, and Outcome Prediction. *Australian Journal of Machine Learning Research & Applications*, Vol.1(1), pp. 113-131.

GeeksforGeeks (2023). Image Steganography iin Cryptography, geeksforgeeks.org

Gentry, C. (2009). A Fully Homomorphic Encryption Scheme. PhD. Thesis Stanford

Gentry, C. (2010). Computing Arbitrary Functions of Encrypted Data. Comm, ACM Vol 53(3), pp. 97-105

Ghadi (2023). A Study of Modified RSA Cryptosystem. *International Journal of Applied Sciences and Technology,* pp. 51-70

Gharib M., Lollini P., Botta M., Amparore E., Donatelli S., Bondavalli A. (2018). On the Safety of Automotive Systems Incorporating Machine Learning Based Components: A Position Paper; *Proceedings of the 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops,* DSN-W 2018; Luxembourg. 25–28 June 2018.

Ghosh, S., Singh, A.R., Pandey, G., & Lakhanpal, A. (2020). A Novel Solution to Cloud Data Security Issues, *2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN),* Greater Noida, India, 2020, pp.

857-860, doi: 10.1109/ICACCCN51052.2020.9362743. https://ieeexplore-ieeeorg.ezaccess.library.uitm.edu.my/stamp/stamp.jsp?tp=&arnumber=9362743

Ghoul, S., Sulaiman, R., & Shukur, Z. (2023). A Review on Security Techniques in Image Steganography. *International Journal of Advanced Computer Science and Applications*, vol 14(6), pp. 361-385

Ghuge, S.S., Kumar, S., Savitha, S., & Suraj, V. (2020). Multilayer Technique to Secure Data Transfer in Private Cloud for SaaS Applications. *In Proceedings of the2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India*, 5–7 March, pp. 646–651.

Gonzalez, N., Miers, C., Redigolo, F., Simplicio, M., Carvalho, T., Näslund, M., & Pourzandi, M. (2012) A quantitative analysis of current security concerns and solutions for cloud computing. *J Cloud Comput Adv Syst Appl*, vol. 1(1), pp.11

Gu, Y.Q., He, C., Liu, F.G., & Ye, J. (2021). Roman Ink for Steganography. *Adv. Optical Mater*, vol. 9, https://doi.org/10.1002/adom.202002038

Gupta, S., Shankar, G., & Gupta, A. (2021), Cloud Computing: Services, Deployment Models and Security Challenges. *2nd International Conference on Smart Electronics and Communication (ICOSEC).* https://ieeexplore.ieee.org/document/9591794

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. https://link.springer.com/ article/10.1007/s11042-016-3735-1

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, 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

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

Hamza, R., Hassan, A., Ali, A., Bakri Bashir, M., M. Alqhtani, S., Mohmmed Tawfeeg,

T., & Yousif, A. (2022). Towards Secure Big Data Analysis via Fully Homomorphic Encryption Algorithms. *Entropy,* 24, 519. https://doi.org/10.3390/e24040519

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 andCommunication* Semarang, Indonesia, doi:10.1109/ISEMANTIC.2017.8251836

Hazim, H. T. (2022). Secure Chaos of 5G Wireless Communication System Based on IOT Applications. *International Journal of Online & Biomedical Engineering*, vol. 18(12), https://doi.org/10.3991/ijoe.v18i12.33817

HCLTech (2022). Homomorphic Encryption: Exploring Technology Trends and Future Approach: https://www.hcltech.com/sites/default/files/documents/resources/whitepaper/files/2024/06/03/redesigned-homomorphic-encryption-%20exploring-technology-trends-and-future-approach.pdf

Hung, Y.H. (2019) Investigating how cloud computing transforms the development of Industries. *IEEE Access*, vol 7, pp.181505–181517

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

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.

Hutto, C., Gilbet, & VADER, E. (2014). A Parsimonious Rule-Based Model for Sentiment Analysis of social media Text, *International AAAI Conference on Weblogs and social media*, pp. 216–225

Hydara, I., Sultan, A.B.M., Zulzalil, H., & Admodisastro, N. (2015). Current state of Research on cross-site scripting (XSS): A systematic literature review. *Inf Softw Technol*, vol. 58, pp. 170–186

IDC (2022). Cloud Spending to Grow 17% to $88.9 billion in 2022 vs 10% in 2021. Retrieved from ttps://infotechlead.com/cloud/cloud-spending-to-grow-17-to-88-9-bn-in-2022-vs-10-in-2021-idc-74765 on 15/1/2025 at 03.32 am

IEEE Digital Privacy (2025). Types of Homomorphic Encryption, retrieved from Htttps://digitalprivacy.ieee.org/publications/topics/types-of-homomorphic-Encryption/?form=MG0AV3

Islam, S.M.J., Chaudhury, Z.H., Islam, S. (2019). A Simple and Secured Cryptography System of Cloud Computing. *In Proceedings of the 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE), Edmonton, AB, Canada*, 5–8 May, pp. 1–3

Islam, T., Manivannan, D., Zeadally, S. (2016). A classifcation and characterization of security threats in cloud computing. *Int J Next Gener Comput*, vol.7(1), pp. 1071–1081

Ismail, A.A., & Boukari, S. (2019). Performance Analysis of Text and Image Steganography with RSA Algorithm in Cloud Computing, *International Journal of Software Engineering & Applications*, Vol. 9

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

Jansen, W.A., & Grance, T. (2011). Guidelines on security and privacy in public cloud Computing. *NIST Spec Publ* 800(144):10–11

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

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.

Johnson, N. F., Zoran, D., & Sushil, J. (2001). Information Hiding: Steganography and Watermarking-Attacks and Countermeasures, *Springer Science & Business Media*, vol (1)

Joseph, M., & Mohan, G. (2022). Design of a Hybrid Optimization and Homomorphic Encryption for Securing Data in Cloud Environment. *International Journal of Computer Networks and Applications*, vol 9 (4), pp. 385-398

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.

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 16/1/2025 at 16.04 PM from https://dl.acm.org/doi/abs/10.1109/ANTS.2016.7947811

Kalu, C., Uche, A., Val, E., Udoka, H., Kisakye, A., Maxwell, K.F., Wisdom, O.O. (2023). Design of a Solar Powered Water Supply System for Kagadi Model Primary School in Uganda. *Journal of Engineering, Technology, and Applied Science* (JETAS), vol. 5(2), pp. 67-78.

Kandias, M., Virvilis, N., & Gritzalis, D. (2013). The insider threat in cloud computing. *In Proceedings of the 6th International Conference on Critical Information Infrastructures Security (CRITIS 2011)*, Lecture Notes in Computer Science, vol. 6983, pp. 93–103. Springer. https://doi.org/10.1007/978-3-642-41476-3_8

Kareem, S. M., & Rahma, A. M. S. (2020). A novel approach for the development of the Twofish algorithm based on multi-level key space. *Journal of Information Security and Applications*, vol. 50, 102410. doi: 10.1016/j.jisa.2019.102410

Kareyo, M. A., Siraj, K. S., & Mohammed, A. A. (2020). Ensuring data security in cloud computing using homomorphic encryption. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT),* vol. *6*(2), pp. 296–302. https://doi.org/10.32628/IJSRCSEIT

Kaur, M., & Kaimal, A.B. (2023). Analysis of Cloud Computing Security Challenges and Threats for Resolving Data Breach Issues. *International Conference on Computer Communication and Informatics (ICCCI)*, Coimbatore, India, pp. 1-6, doi: 10.1109/ICCCI56745.2023.10128329. https://ieeexploreieeeorg.ezaccess.library.uitm.edu.my/stamp/stamp.jsp?tp=&arnumber=10128329

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

Khari, M., Kumar, M., Vaishali. (2016). Secure data transference architecture for cloud Computing using cryptography algorithms. *In Proceedings of the 2016 3rd International Conference on Computing for Sustainable Global Development* (INDIACom), NewDelhi, India, 16–18 March, pp. 2141–2146.

Kheiralla, F. A. M. (2018). Steganography a New Dawn in the World of Information Security Compared with Cryptography Technology. *International Journal of Innovations & Advancement in Computer Science*, Vol.7(1).

Kieseberg, P., Fruhwirt, P., Schrittwieser, S., & Weippl, E. (2015). Security Tests for Mobile Applicationswhy using tls/ssl is Not Enough. *In IEEE Eighth International Conference on Software Testing, Verification and Valid in Section VIIation Workshops,* pp. 1–2

Kim, B.H., Huang, W., & Lie, D. (2012). Unity: Secure and durable personal cloud storage, Presented at the CCSW ACM Workshop Cloud Computer. Security Workshop, New York, NY, USA, pp. 31–36.

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)

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, R., & Goyal, R. (2019). On cloud security requirements, threats, vulnerabilities and Countermeasures: A survey, *Computer Science Review*, vol. 33, pp. 1–48

Kumar, S., Jafri, S.A.A., Nigam, N., Gupta, N., Gupta, G., & Singh, S.K. (2020). A New User Identity Based Authentication, Using Security and Distributed for Cloud Computing. *IOP Conf. Ser. Mater. Sci. Eng*. Vol.748, pp. 1-10

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.

Kumona, N.R., Teja, B.G., & Kumar, T.D. (2024). Enhancing Data Security in IoT: An Integrated Approach with Cryptography and Steganography. *Journal of Emerging Technologies and Innovative Research*, vol 11(8), pp. 17-26

Ladislas, E.S. (2023). Personalizing Government Services through Artificial Intelligence: Opportunities and Challenges. *Indian Journal of Artificial Intelligence and Neural Networking* (IJAINN), vol. 3(5), pp. 13-18.

Ladislas, E.S., & Phelix, B (2023). Factors Affecting E-government Adoption in the Democratic Republic of Congo. *International Research Journal of Engineering and Technology* (IRJET). Vol. 9(3), pp. 1309-1323.

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

Lavanya, S., & Saravanakumar. (2022). Secured two factor authentication graph-based replication and encryption strategy in cloud computing. *Multimedia Tools and Applications,* pp. 16105– 16125

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, Y., Sun, Y.L., Ryoo, J., Rizvi, S., & Vasilakos, A.V. (2015). A survey of security and Privacy challenges in cloud computing: solutions and future directions. *J Comput Sci Eng,*vol. 9(3), pp. 119–133

Lubega, M., & Karuhanga, M. (2022). On the Eigenvalue problem involving the Robin p(x)-Laplacian. *Annals of Mathematics and Computer Science*, vol. 7(7), pp. 1-11.

Mahmood, T., Fulmer, W., Mungoli, N., Huang, J., & Lu, A. (2019, October). Improving information sharing and collaborative analysis for remote geospatial visualization using mixed reality. *In 2019 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)* pp. 236-247. IEEE

Majjed, L.O.A. (2023). Image Types and Formats. Retrieved from

https://www.slideshare.net/ssuserff72e4/lecture-22023pdf-253271077

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.

Mani, A.C., & Malviya, A.K. (2019). Security Challenges in Cloud Computing Networks. *Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE)*, Sultanpur, India, pp. 8–9 Margaret, K., Semajeri, E., Mbabazi, B.P., & Wycliff, M.Z. (2020). E-Government Development Review in Africa: An Assessment of Democratic Republic of Congo's Global E-Government UN Ranking. *International Journal of Engineering and Information Systems*, vol. 4(11), pp. 47-55.

Mary Sheeba, R., Parameswari, R. (2023). Hybrid Security for Data in Cloud Computing: A Review. In: Dutta, P., Chakrabarti, S., Bhattacharya, A., Dutta, S., Piuri, V. (eds) *Emerging Technologies in Data Mining and Information Security*. Lecture Notes in Networks and Systems, vol 491. Springer, Singapore.

Medvedeva, M.A., & Hashem, A.A.H. (2022). Image Steganography by Homomorphic Encryption with Hybrid Optimization. *Neuro Quantology*, vol. 20(13), pp. 2930-2935

Mell P, & Grance, T. (2011) The NIST definition of cloud computing: recommendations of The National Institute of Standards and Technology. NIST Spec Publ 800–145:1–7

Meng, X., & Zheng, X. (2015). Cryptanalysis of RSA with small Parameter Revisited. *Information Process,* Lett 115(11), pp. 858-862

Mishra, A.K. (2017). Digital Signature: The Need of Cashless Society. CreateSpace Independent Publishing Platform. ISBN-13: 978-1546382539

Mohammed, A.H.Y., Dziyauddin, R. A., & Latiff, L.A. (2025). Current multi-factor Authentication: Approaches, Requirements, Attacks and Challenges. *International Journal of Advanced Computer Science and Applications*, vol. 14(1), pp. 166-178

Moore, N. (2018). How to do Research. *A Practical Guide to Designing and Managing Research Projects*, Cambridge University Press, pp. 106-111

Mohsin, J. K., Han, L., Hammoudeh, M., & Hegarty, R. (2017). Two Factor Vs Multi-factor, an Authentication Battle in Mobile Cloud Computing Environments. *Proceedings of the International Conference on Future Networks and Distributed Systems.* doi:10.1145/3102304.3102343

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. (2016). Social Engineering Attack Examples, Templates and Scenarios. *Compututer & Security*, vol. 59, pp. 186–209.

Mthunzi, S.N., Benkhelifa, E., Bosakowski, T., Guegan, C.G., & Barhamgi M (2020) Cloud computing security taxonomy: From an atomistic to a holistic view. *Future Generation Computer Systems,* vol. 107. Pp. 620–644

Muhammed, K.J., Isiaka, R.M., Asaju-Gbolagade, A.W., Adewole, K.S., & Gbolagade, K.A. (2021). Hybrid Algorithm for Symmetric Based Fully Homomorphic Encryption. In: Florez, H., Pollo-Cattaneo, M.F. (eds). Applied Informatics, ICAI2021 *Communications in Computer and Information Science* vol 1455 Springer, Cham, https://doi.org/10.1007/978-3-030-89654-6_27

Mungoli, N. (2020). Exploring the Technological Benefits of VR in Physical Fitness (Doctoral dissertation, The University of North Carolina at Charlotte).

Muraidhara, P. (2013). Security issues in cloud computing and its countermeasures. *International Journal of Scientific & Engineering Research*, vol. 4(10).

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

NCC Group (2023). The Paillier Cryptosystem with Applications to Threshold ECDSA https://www.nccgroup.com/us/research-blog/the-paillier-cryptosystem-with-applications-to-threshold-ecdsa/?form=MG0AV3

Ndubuisi, S., Adanma, N.O., Nnamchi, Sanya, O.D. Mundu, M.M., Gabriel, V. (2020).

Dynamic analysis of performance of photovoltaic generators under moving cloud conditions. *Journal of Solar Energy Research*. Vol. 5(2), pp. 453-468.

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

Newman, M. (2018). Further Limitations on Information-Theoretically Secure Quantum Homomorphic Encryption. https://api.semanticscholar.org/CorpusID:119185375

Nie, T., & Zhang, T. (2009). A study of DES and Blowfish encryption algorithm. In TENCON IEEE Region 10 Conference.

Nieuwenhoff, T.V.D. (2021). Homomorphic Encryption. The Pros and Cons, https://tvdn.me/fhe/2021-11-14_fully-homomorphic-encryption-pro-con/?form=MG0AV3&MG0AV3

Nnamchi S. N., Sanya, O.D., Zaina, K., Gabriel, V. (2020). Development of dynamic thermal input models for simulation of photovoltaic generators. *International Journal of Ambient Energy*, vol. 41(13), pp. 1454-1466.

Nyimbili, F., & Nyimbili, L. (2024). Types of Purposive Sampling Techniques with them Examples and Applications in Qualitative Research Studies. *British Journal of Multidisciplinary and Advanced Studies*, vol 5(1), pp. 90-99

Odun-Ayo I, Ananya M, Agono F, Goddy-Worlu R (2018) Cloud computing architecture: A Critical Analysis. *In Proceedings of the 18th international conference on computational science and applications*. IEEE, pp 1–7

Odun-Ayo, I., Okereke, C., & Orovwode, H. (2018). Cloud and Application Programming Interface – Issues and Developments. The World Congress on Engineering 2018. https://www.researchgate.net/profile/Isaac-Odun-Ayo/publication/333402621 - Cloud and Application Programming Interface/links/5cebe920458515026a5ee994/Cloud-and-ApplicationProgramming-Interface.pdf?origin=publication detail

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

Orucho, D.O., Awuor, F.M., Makiya, R., & Oduor, C. (2023). An Enhanced Data Transmission in Mobile Banking using LSB-AES Algorithm. *Asian Journal of Research in Computer Science*, vol. 16(1), pp. 43-56.

Paillier, P. (2005). Paillier Encryption and Signature Schemes. In: Van Tilborg, H.C.A (eds). Encyclopedia of Cryptography and Security. Springer, Boston, M.A https://doi.org/10.1007/0-387-23483-7_293

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.

Pant V.K., Prakash, J., & Asthana, A. (2015). Three step data security model for cloud Computing based on RSA and steganography. *In Proceedings of the 2015 International Conference on Green Computing and Internet of Things (ICGCIoT), Greater Noida, India*, 8–10 October, pp. 490–494.

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, H.B., & Kansara, N. (2021). Cloud Computing Deployment Models: A Comparative Study. *International Journal of Innovative Research in Computer Science & Technology,* vol 9(2), pp. 45-50

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

PhoenixNAP (2020). 5 Cloud Deployment Models: Learn the Differences. Retrieved From: ttps://phoenixnap.com/blog/cloud-deployment-models on 8/1/2025 at 9.47 am

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.

Prakriti, G., & Deepak, S. (2015). A Survey on Digital Image Steganography Techniques. *International Journal of Electronics, Electrical and Computational System*, Vol. 4

Premarathne, U., Abuadbba, A., Alabdulatif, A., Khalil, I., Tari, Z., Zomaya, A., & Buyya, R. (2016). Hybrid Cryptographic Access Control for Cloud-Based EHR Systems. *IEEE Cloud Computing*, vol. 3, pp. 58–64.

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

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/

Ramya, K. (2013). Design and Implementation of Digital Signatures. *International Journal of Engineering Research and Technology*, vol 2(2), pp. 1386-1390

Rathod, A., & Patel, K. (2025). Internal Re-keying Based Modified AES. *Indian Journal of Science and Technology*, vol. 18(1), pp. 85-94

Regueiro, C., Seco, I., De Diego, S., Lage, O., & Etxebarria, L. (2021). Privacy-enhancing distributed protocol for data aggregation based on blockchain and homomorphic encryption. *Inf. Process. Manag*, Vol. 58, pp. 102745.

Reza, H., & Sonawane, M. (2016). Enhancing Mobile Cloud Computing Security Using Steganography. *Journal of Information Security*, vol 7, pp. 245–259.

Rouse, M. (2017). Single-Factor Authentication. Accessed on 2/1/2023 at 22.02 pm from https://searchsecurity.techtarget.com

Ryu, J., Kim, K., & Won, D. (2023). A Study on Partially Homomorphic Encryption. *17th International Conference on Ubiquitous Information Management and Communication.* IEEE Splore.

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

Safar, F., & Al King, R. (2023). Data Security in Cloud Computing. International Journal of Wireless and Ad Hoc Communications, vol 7(1), pp. 50-61

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 Defence 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

Sajjad, M., Muhammad, K., Baik, S.W., Rho, S., Jan, Z., & Yeo, S.-S. (2016). Mobile-cloud assisted framework for selective encryption of medical images with Steganography for resource-constrained devices. *Multimed. Tools Appl, vol.* 76, pp. 3519–3536.

Salt Labs (2023). Salt State of API Security Report Q1 2023 https://content.salt.security/state-api-report.html

Saltz, J., Hotz, N., Wild, D.J., & Stirling, K. (2018). Exploring Project Management Methodologies used with Data Science Teams, *American Conference on Information Systems*, and AMCIS 2018 Proceedings 12

Sanghi, A., Chaudhary, S., & Dave, M. (2017). Enhance the Data Security in Cloud Computing by Text Steganography. Lect. Notes Netw. Syst. 2017, 241–248.

Sanyal, A., J. Kusner, M., Gascón, A., & Kanade, V. (2018). TAPAS: Tricks to Accelerate (encrypted) Prediction as a Service. Retrieved from https://proceedings.mlr.press/v80/sanyal18a/sanyal18a.pdf on 16/1/2025 at 4.06 am

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.

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

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*.

Scheibner, J., Kiltz, E., & Struck, P. (2021). Potential of homomorphic encryption for cloud computing use cases in manufacturing. *Journal of Cybersecurity and Privacy,* vol. *3*(1), pp. 44–60. https://doi.org/10.3390/jcp3010004

Scheibner, J., Louis Raisaro, J., Ramón Troncoso-Pastoriza, J., Ienca, M., Fellay, J.,

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.

Sharda, S., & Budhiraja, S. (2013). Image steganography: A review. *International Journal of Emerging Technology and Advanced Engineering*, Vol. 3(1), pp. 707-710.

SentinelOne (2025). What is Ciphertext? Types and Best Practices. SentinelOne.com

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.

Seth, B., Dalal, S., & Kumar, R. (2019). Hybrid Homomorphic Encryption Scheme for Secure Cloud Data Storage. In: Kumar, R., & Wiil, U. (eds). Recent Advances In Computational Intelligence. Studies in Computational Intelligence, vol 823, Springer, Cham https://doi.org/10.1007/978-3-030-12500-4_5

Shefali, N., & Ashish, S. (2015). Designing and Performance of Advanced Steganography System Using RGB Image. *International Journal for Advance Research in Engineering and Technology*, Vol. 3

Shwaysh, M.M., Alani, S., Saad, M.A., & Abdulhussein, T.A. (2024). Image Encryption And Steganography Method Based on AES Algorithm and Secret Sharing Algorithm. *Ingenierie des Systems d'Information*, vol. 29(2), pp. 705-714

Siedlecki, S. L. (2020). *Understanding Descriptive Research Designs and Methods.* Using Research to Advance Nursing Practice. Wolters Cluwer Health, Inc, Cleveland, USA

Singh, G., Kumar, A., & Sandha, K.S. (2021). A Study of New Trends in Blowfish Algorithm. *International Journal of Engineering Research and Applications*, vol. 1(2), pp. 321-326

Sinigaglia, F., Carbone, R., Costa, G., & Zannone, N. (2020) A survey on multi-factor Authentication for online banking in the wild, *Computer Security*, vol. 95 doi: 10.1016/j.cose.2020.101745.

Shannon, C.E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal*, vol 27(3), pp. 379-423

Sharma, L.R. (2023). Choosing Appropriate Probality Sampling Designs in Research. *Primera Scientific Medicine and Public Health*, vol 2(2), pp. 4-21

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, 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, G.P. (2024). Enhancing Cloud Data Security using Homomorphic Encryption Techniques. *International Journal of Trend in Scientific Research and Development*, vol 8(5), pp. 940-946

Shen, J., Gui, Z., Ji, S., Shen, J., Tan, H., & Tang, Y. (2018). Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks. *J. Netw. Comput. Appl,* vol. 106, pp. 117–123.

Simplilearn (2022). Digital Signature Algorithm (DSA) in Cryptography: How it works and Advantages. Retrieved on 30th December, 2024 at 14.30 form https://www.simplilearn.com/tutorials/cryptography-tutorial/digital-signature-algorithm

Sneha, P.S., Sankar, S., & Kumar, A.S. (2020). A Chaotic Colour Image Encryption Scheme Combining Walsh–Hadamard Transform and Arnold–Tent Maps. *Journam of Ambient Intelligent Human Computer,* vol. 11, pp. 1289–1308 https://doi.org/10.1007/s12652-019-01385-0

Stallings, W. (2006). *Cryptography and Network Security Principles and Practice*, Prentice Hall.

Stallings, W. (2014). *Cryptography and Network Security Principles and Practice*. 6th Ed. Prentice Hall, Upper Saddle River.

Stinson, D.R. (2006). *Cryptography, Theory and Practice,* Third edition, Chap-man & Hall/CRC

Stockemer, D., Stockemer, G., & Glaeser. (2019). Quantitative methods for the social Sciences, Springer International Publishing, vol. 50, pp. 185.

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. *Multimed Tools. Appications.*

doi: 10.1007/s11042-019-08476-2.

Sunil, CH., Devika, KD., Teja, S.S., Sreenivas, Ch., Afreed, SK., & Reddy, J.S. (2024). Advanced Data Security using Hybrid Cryptography and Steganography. *International Journal of Research Publication and Reviews*, vol 5(3), pp. 1246-1250

Susilawati, F., Mohamad, N., Sahira, & Yasin, M. (2018). Information Hiding Based on Audio Steganography using Least Significant Bit. *International Journal of Engineering & Technology*, Vol. 7(3.28), pp. 334-336.

Tahir, A.S. (2015). Design and Implementation of RSA Algorithm using FPGA. *International Journal of Computer and Technology*, vol 14(12), pp. 6361-6367

Tamanna & Ashwani, S. (2017). Analysis and Refinement of Steganography Techniques, *International Journal of Computer Applications*, Vol. 170

Thabit, F., Alhomdy, A.P., Al-Ahdal, A.H.A., & Jagtap, P.D. (2021). A new lightweight Cryptographic algorithm for enhancing data security in cloud computing, *Global Transitions Proceedings*, vol. 2(1), pp. 91-99

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

Thokchom, S., & Saikia, D.K. (2019). Privacy preserving and public auditable integrity Checking on dynamic cloud data. *IJ Netw Secur, vol.* 21(2), pp. 221–229

Thusanth, T., & Diane, G. (2016). A prototype tool to demonstrate Steganography principles

Trilogix Cloud (2025). AWS and Cloud Resource Management. Retrieved from: https://trilogix.cloud on 8/1/2025

Tripathi, N., & Hubballi, N. (2018). Slow Rate Denial of Service Attacks against HTTP/2 and Detection. *Computers & Security*, vol. 72, pp. 255–272.

Tubaishat, A. (2019). Security in Cloud Computing: State-of-the-Art, Key Features, Challenges and Opportunities. *In Proceedings of the 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS),* Singapore, 23–25 February 2019, pp. 311–315.

Ullah, I., Boreli, R., & S. Kanhere, S. (2022). Privacy in targeted advertising on mobile Devices, pp. 648-678. https://doi.org/10.1007/s10207-022-00655-x

Unit 4 Lab 4: Data Representation and Compression, pp. 6 (edc.org)

USC-SIPI Image Database. http://sipi.usc.edu/database/ (accessed on 8th September 2024)

Vassilev, V., Phipps, A., Lane, M., Mohamed, K., & Naciscionis, A. (2020). Two-factor authentication for voice assistance in digital banking using public cloud services. *In Proceedings of the 2020 10th International Conference on Cloud Computing, Data Science Engineering (Confluence),* Noida, India, 29–31 January 2020; pp. 404–409.

Vayena, E., & Hubaux, J. P. (2021). Revolutionizing Medical Data Sharing Using Advanced Privacy Enhancing Technologies: Technical, Legal, and Ethical Synthesis. Retrieved from https://pubmed.ncbi.nlm.nih.gov/33629963/

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.

Wikipedia (2023). Paillier Cryptosystem, retrieved on 1/3/2025 from https://en.wikipedia.org/wiki/paillier-cryptosystem? Form=MG0AV3

Wu, C., Buyya, R., & Ramamohanarao, K. (2020). Modeling cloud business customers' Utility Functions. *Future Generation Computer Systems*, vol. 105, pp.737–753

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

Yang, C., & Ye, J. (2015). Secure and efficient fine-grained data access control scheme in Cloud computing. *J High Speed Netw, vol.* 21(4), pp. 259–271 65.

Yazici M., Basurra S., Gaber M. (2018). Edge Machine Learning: Enabling Smart Internet of Things Applications. *Big Data Cogn. Comput*, vol. 2(26). doi: 10.3390/bdcc2030026.

Ye, Y., Li, T., Adjeroh, D., & Iyengar, S.S. (2018). A survey on malware detection using data mining techniques, *ACM Computer. Surveys,* vol. 50(3), pp. 1–40

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

Yousuf, H., Lahzi, M., Salloum, S.A., & Shaalan, K. (2021). Systematic review on fully

homomorphic encryption scheme and its application. In: Al-Emran, M., Shaalan, K., & Hassanien, A. (eds.) Recent Advances in Intelligent Systems and Smart. Studies in Systems, Decision and Control, *Springer Cham* vol. 295, pp. 537–551.

Yu, S., Wang, C., Ren, K., & Lou, W. (2010). Achieving secure, scalable, and fine-grained Data access control in cloud computing. *In: Proceedings of INFOCOM. IEEE*, pp. 1–9

Yuefa, W.D., & Yaqiang, G. (2009). Data security model for cloud computing. *In Proceedings of the 2009 International Workshop on Information Security and Application* (IWISA), Qingdao, China, 21–22 November, pp. 141–144.

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

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.

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: USCI-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

THE CO-OPERATIVE UNIVERSITY OF KENYA
P. O. Box 24814-00502 Karen, Kenya
Telephone: (020)-2430127/2679456/8891401 Fax (020)-8891410
www.cuk.ac.ke

BOARD OF POSTGRADUATE STUDIES

5th April 2025

The Director,
National Commission for Science, Technology& Innovation,
Utalii House, Nairobi.

Dear Sir/Madam,

RE: DANIEL OKARI ORUCHO, REGISTRATION NUMBER: C005/600020/2023

This is to introduce the above named Master of Science in Cyber Security student in the School of Computing and Mathematics at The Co-operative University of Kenya.

He has successfully completed his course work and is proceeding to the field to collect data on data protection in public cloud computing. The title of his research project is **"A Hybrid Security Model for Data Protection in Public Cloud Computing"**

Kindly accord him the necessary assistance.

Yours faithfully,

D. K. Muthoni
Director, Board of Postgraduate Studies.

Copy to: Dean SCM

CUK is ISO 9001: 2015Certified

APPENDIX III: RESEARCH PERMIT

REPUBLIC OF KENYA

NATIONAL COMMISSION FOR SCIENCE,TECHNOLOGY & INNOVATION

Ref No: **329580**

Date of Issue: **19/May/2025**

RESEARCH LICENSE

This is to Certify that Mr.. DANIEL OKARI ORUCHO of The Cooperative University of Kenya, has been licensed to conduct research as per the provision of the Science, Technology and Innovation Act, 2013 (Rev.2014) in Nairobi on the topic: A Hybrid Security Model for Data Protection in Public Cloud Computing for the period ending : 19/May/2026.

License No: **NACOSTI/P/25/4173851**

329580

Applicant Identification Number

Deputy Director
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.

See overleaf for conditions

APPENDIX IV: MATLAB CODE SNIPPETS

Key Generation using Paillier Cryptosystem

```
p = 17;
q = 23;
where
“n = p * q;
lambda = lcm (p-1, q-1);
% Choose generator g and validate
valid_g_found = false;
while ~valid_g_found
   g = randi([2, n-1]);
   if gcd(g, n) == 1
      L = @(x) floor ((x - 1) / n);
      L_val = L (mod (powermod (g, lambda, n^2), n^2));
      if gcd (L_val, n) == 1
         valid_g_found = true;
      end
   end
end
mu = modInverse (L_val, n);
```

Paillier Encryption of Numeric Plaintext

```
plaintext = 42; % Example numeric value
r = randi ([1, n-1]);
while gcd (r, n) ~= 1
   r = randi ([1, n-1]);
end
ciphertext = mod (powermod (g, plaintext, n^2) * powermod (r, n, n^2), n^2)”;
```

Embedding Ciphertext into RGB Cover Image via LSB

```
binaryData = dec2bin (ciphertext, 32); % Convert to binary
img = imread('coverImage.png'); % Load RGB image

stegoImg = stegancoder(img, binaryData); % Custom function handles embedding
inwrite (stegoImg, 'stegoImage.bmp'); % Save stego-image in BMP format
```

Extracting and Decoding Ciphertext from Stego-Image

```
stegoImg = imread('stegoImage.bmp');
flatImg = reshape (stegoImg, [], 3); % Flatten RGB for bit access
```

```
% Extract LSBs to reconstruct binary string
extractedBits = '';
for i = 1:32
    channel = mod (i-1, 3) + 1;
    pixelIdx = ceil (i / 3);
    extractedBits(i) = num2str (bitget (flatImg (pixelIdx, channel), 1));
end
decodedCiphertext = bin2dec(extractedBits);
```

Paillier Decryption Function

```
function plaintext = paillierDecrypt (ciphertext, p, q, g)
  “
 n = p * q;
    lambda = lcm (p-1, q-1);
    L = @(x) floor ((x - 1) / n);
    mu = modInverse (L (mod (powermod (g, lambda, n^2), n^2)), n);
    plaintext = mod (L (powermod (ciphertext, lambda, n^2)) * mu, n);
end”
```

APPENDIX V: SIMILARITY INDEX REPORT

Daniel Orucho

A Hybrid Security Model for Data Protection in Public Cloud Computing

Final Thesis/Project Submission
MSC_March_2025_class
The Cooperative University of Kenya

Document Details

Submission ID
trn:oid:::1:3351218726

Submission Date
Sep 25, 2025, 2:58 PM GMT+3

Download Date
Sep 26, 2025, 8:38 AM GMT+3

File Name
A_HYBRID_SECURITY_MODEL_FOR_DATA_PROTECTION_IN_PUBLIC_CLOUD_COMPUTING.docx

File Size
2.8 MB

114 Pages

22,951 Words

141,061 Characters

turnitin Page 1 of 120 - Cover Page Submission ID trn:oid:::1:3351218726

4% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

- **52** Not Cited or Quoted 4%
  Matches with neither in-text citation nor quotation marks
- **13** Missing Quotations 1%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

3% Internet sources
3% Publications
0% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Daniel Orucho

A Hybrid Security Model for Data Protection in Public Cloud Computing

Final Thesis/Project Submission

MSC_March_2025_class

The Cooperative University of Kenya

Document Details

Submission ID
trn:oid:::1:3351218726

Submission Date
Sep 25, 2025, 2:58 PM GMT+3

Download Date
Sep 26, 2025, 8:38 AM GMT+3

File Name
A_HYBRID_SECURITY_MODEL_FOR_DATA_PROTECTION_IN_PUBLIC_CLOUD_COMPUTING.docx

File Size
2.8 MB

114 Pages

22,951 Words

141,061 Characters

turnitin Page 1 of 116 - Cover Page Submission ID trn:oid:::1:3351218726

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.