International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

A Dual-Layer Homomorphic Encryption Framework for Secure Electronic Voting in Storage Area Networks

Lawrence S. Igenewari[1], Professor Raphael O. Okonkwo[2]
[1,2]*Department of Computer Science, Faculty of Physical Sciences, Nnamdi Azikiwe University, Awka, Nigeria.*

***Abstract*-- Electronic voting (e-voting) systems have emerged as viable alternatives to traditional paper-based elections because of their potential to improve efficiency, accessibility, and speed in electoral processes. However, concerns regarding ballot secrecy, voter privacy, vote integrity, result verifiability, and resistance to cyberattacks continue to impede their widespread acceptance (Abdulhamid et al., 2013; Carroll et al., 2011). These concerns become more critical when electronic ballots are transmitted across networks and stored in centralized Storage Area Networks (SANs), where unauthorized access and malicious modifications may compromise election outcomes (Akintoye, 2022; Chandhiny & Vairamuthu, 2018).**

This study proposes a dual-layer homomorphic encryption framework that integrates the Benaloh cryptosystem and ElGamal encryption algorithm to enhance the security of electronic voting systems deployed in SAN environments. The Benaloh cryptosystem provides additive homomorphic properties that permit encrypted vote aggregation without exposing individual ballots, while ElGamal offers an additional protective layer for securing ciphertexts during transmission and storage (Benaloh, 1994; ElGamal, 1985). The framework was implemented using C# and independently validated using MATLAB to ensure computational correctness and reproducibility. Experimental evaluations demonstrated successful encryption of ballots, accurate homomorphic tallying, controlled decryption of final results, and consistency between implementation and validation environments.

The findings indicate that the proposed framework enhances confidentiality, integrity, and transparency in SAN-based e-voting systems while preserving voter anonymity and enabling secure computation on encrypted ballots. The study contributes a practical and reproducible model for secure electronic elections and provides a foundation for future research involving threshold cryptography and large-scale deployments.

***Keywords*-- Electronic voting, Homomorphic encryption, Benaloh cryptosystem, ElGamal encryption, Storage Area Networks, Ballot secrecy, Secure tallying.**

I. INTRODUCTION

The conduct of free, fair, and credible elections constitutes a fundamental requirement for democratic governance. Electoral systems provide mechanisms through which citizens participate in governance by selecting representatives and expressing political preferences.

Traditionally, paper-based voting methods have dominated electoral processes because of their simplicity and familiarity. Nevertheless, these systems are often characterized by lengthy counting procedures, logistical complexities, high operational costs, and vulnerability to human errors and manipulation (Abdulhamid et al., 2013).

Advancements in information and communication technologies have led to the development of electronic voting systems designed to overcome the limitations of traditional voting approaches. Electronic voting offers several advantages, including faster vote counting, improved accessibility, reduced administrative burdens, and enhanced efficiency (Adil & Jorg, 2015). Despite these benefits, public trust in electronic voting remains limited due to concerns surrounding security vulnerabilities, transparency, auditability, and voter privacy (Carroll et al., 2011).

One of the primary requirements of any voting system is ballot secrecy. The confidentiality of individual votes protects citizens from coercion, intimidation, and vote buying. Simultaneously, election results must accurately reflect voters' intentions without unauthorized alterations. Consequently, secure electronic voting systems must satisfy multiple security requirements, including confidentiality, integrity, authenticity, availability, accountability, and verifiability (Abdulhamid et al., 2013; Bogdanov et al., 2012).

The increasing reliance on centralized storage infrastructures has further transformed the landscape of electronic voting. Storage Area Networks (SANs) provide scalable and highly available storage solutions capable of managing large volumes of electoral data. SAN environments facilitate efficient storage and retrieval of encrypted ballots and election records. However, centralized storage also introduces new attack vectors that may expose sensitive information to insider threats, unauthorized access, and malicious tampering (Akintoye, 2022; Chandhiny & Vairamuthu, 2018). Therefore, securing both the transmission and storage of electronic ballots remains an important challenge.

Cryptographic mechanisms have long been recognized as essential tools for protecting sensitive electoral information.

Conventional encryption algorithms provide confidentiality by transforming plaintext into unreadable ciphertext. However, most traditional cryptographic approaches require decryption before computation can be performed on protected data. In electronic voting environments, this requirement exposes ballots during tallying processes and increases the risk of privacy breaches (Ayub et al., 2015).

Homomorphic encryption has emerged as a promising solution to this challenge. Homomorphic cryptosystems permit mathematical operations to be performed directly on encrypted data, producing encrypted results that correspond to operations executed on the original plaintexts after decryption (Ayman et al., 2020; Chatterjee & Aung, 2019). This capability enables votes to remain confidential throughout the tallying process.

Homomorphic encryption schemes may be categorized as partially homomorphic, somewhat homomorphic, leveled homomorphic, or fully homomorphic depending on the range of supported operations (Brakerski et al., 2011a; Brakerski & Vaikuntanathan, 2011a). Fully homomorphic encryption allows arbitrary computations on encrypted data but is often associated with substantial computational overhead that may limit practical deployment (Gentry, 2009; Chunsheng, 2012). Consequently, partially homomorphic schemes continue to attract considerable interest in applications such as electronic voting where only specific arithmetic operations are required (Biksham & Vasumathi, 2017).

Among additive homomorphic schemes, the Benaloh cryptosystem provides an efficient mechanism for aggregating encrypted votes without decrypting individual ballots (Benaloh, 1994). Benaloh extended earlier probabilistic encryption techniques by introducing a larger message space and additive homomorphic functionality. The scheme ensures that the multiplication of ciphertexts corresponds to the addition of underlying plaintexts after decryption, thereby supporting secure tally computation (Benaloh, 1994).

Although Benaloh facilitates secure vote aggregation, additional safeguards are necessary to protect ciphertexts during storage and transmission across distributed environments. The ElGamal cryptosystem, introduced by ElGamal (1985), provides probabilistic encryption based on the computational hardness of the discrete logarithm problem.

Its randomized nature ensures that identical messages encrypted multiple times produce distinct ciphertexts, thereby enhancing semantic security and resistance to ciphertext pattern analysis (Bleichenbacher, 1998). Integrating Benaloh and ElGamal within a unified framework offers a dual-layer security architecture capable of preserving homomorphic functionality while strengthening ciphertext protection. In this study, Benaloh is employed to encrypt individual votes and enable additive tallying, whereas ElGamal encapsulates Benaloh ciphertexts before transmission and storage in SAN infrastructures.

Furthermore, the study incorporates independent verification using MATLAB to validate cryptographic computations performed within the C# implementation environment. Cross-platform validation improves transparency, reproducibility, and confidence in the correctness of the proposed framework (Chechulina et al., 2015).

The resulting framework contributes toward the development of secure, verifiable, and practical electronic voting systems capable of operating effectively within Storage Area Networks while preserving voter anonymity and election integrity.

1.1 Problem Statement

Despite significant advances in electronic voting technologies, ensuring end-to-end security remains a major challenge. Existing systems frequently struggle to preserve ballot secrecy while simultaneously supporting efficient vote tallying and result verification (Abdulhamid et al., 2013). Traditional encryption mechanisms expose plaintext ballots during computation because votes must often be decrypted before aggregation can occur (Ayub et al., 2015).

In SAN-based voting environments, the centralized storage of electoral records introduces additional risks, including unauthorized access, insider attacks, and data manipulation (Akintoye, 2022). Although homomorphic encryption provides mechanisms for performing computations on encrypted ballots, practical frameworks that combine secure storage protection, additive tallying, and independent validation remain limited in the literature (Ayman et al., 2020; Biksham & Vasumathi, 2017).

Therefore, there is a need for a practical and verifiable electronic voting framework capable of preserving voter confidentiality, protecting stored ballots, supporting secure tally computation, and independently validating cryptographic operations without exposing individual votes.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

1.2 Aim of the Study

The aim of this study is to develop and validate a dual-layer homomorphic encryption framework that enhances the security of electronic voting systems operating within Storage Area Networks through the integration of Benaloh additive homomorphic encryption and ElGamal ciphertext encapsulation.

1.3 Objectives of the Study

The objectives of this study are to:

1. Design a secure SAN-based electronic voting framework.
2. Implement Benaloh encryption to support additive homomorphic vote aggregation.
3. Integrate ElGamal encryption as a secondary layer for securing ciphertext transmission and storage.
4. Develop a software implementation using C# for cryptographic operations and user interaction.
5. Validate the implementation independently using MATLAB.
6. Evaluate the correctness and effectiveness of the proposed framework.

1.4 Significance of the Study

This study contributes to the advancement of secure electronic voting by presenting a practical framework that combines homomorphic computation with enhanced ciphertext protection. The integration of Benaloh and ElGamal demonstrates how layered cryptographic architectures can simultaneously support ballot secrecy and secure storage (Benaloh, 1994; ElGamal, 1985).

The study also contributes methodologically through the introduction of independent cross-platform verification, thereby improving reproducibility and transparency in cryptographic implementations (Chechulina et al., 2015). In addition, the framework provides an educational and experimental environment suitable for researchers and practitioners investigating secure electronic elections and encrypted computation.

II. Literature Review

2.1 Electronic Voting Systems and Security Requirements

Electronic voting has become one of the most important applications of information and communication technologies in modern democracies. As the number of eligible voters increases globally, electoral management bodies continue to seek mechanisms capable of improving efficiency, reducing operational costs, enhancing accessibility, and accelerating the declaration of election results (Hunter, 2001).

The growing deployment of digital infrastructures in electoral processes has therefore intensified research into secure electronic voting frameworks.

According to Ofori-Dwumfuo and Paatey (2011), the credibility of democracy is closely linked to the integrity of elections, emphasizing that elections constitute one of the fundamental pillars of democratic governance. Similarly, Okediran and Ganiyu (2015) argued that electronic voting systems should be understandable, usable, and accessible to all eligible voters irrespective of age, educational background, disability, or technical competence. They further emphasized that the effectiveness of electronic voting systems depends on their ability to provide confidentiality, integrity, transparency, and verifiability.

Abdulhamid et al. (2013) developed a real-time e-voting framework for Nigeria and identified several critical security requirements, including voter authentication, result veracity, confidentiality of ballots, availability of services, and resistance to unauthorized modifications. Although their framework addressed some of these requirements, it did not implement advanced cryptographic techniques capable of preserving ballot secrecy during tally computation.

Despite their advantages, electronic voting systems remain vulnerable to numerous attacks. Carroll et al. (2011) identified insider attacks, denial-of-service attacks, malware infections, and unauthorized disclosures as major concerns affecting public confidence in digital elections. These concerns have significantly influenced the pace of e-voting adoption worldwide.

Consequently, researchers have increasingly explored privacy-preserving cryptographic mechanisms capable of ensuring that election outcomes remain verifiable without compromising voter anonymity.

2.2 Storage Area Networks and Electoral Data Security

Storage Area Networks (SANs) have emerged as preferred infrastructures for handling large-scale organizational data because of their scalability, availability, centralized administration, and fault tolerance (Peter & Timothy, 2011). Electoral systems generate substantial volumes of sensitive information, including voter records, encrypted ballots, audit logs, and tally reports, making SAN deployment attractive for election data management.

However, centralized storage infrastructures also create opportunities for malicious activities. Geremew et al. (2015) identified several vulnerabilities affecting cloud and storage infrastructures, including unauthorized access, privilege escalation, data leakage, and malicious insider attacks.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

Sachin and Devandra (2017) similarly observed that ensuring confidentiality and integrity within storage environments remains one of the major challenges confronting secure distributed systems.

Uma and Thanooj (2017) reviewed encryption approaches applicable to SAN environments and concluded that traditional encryption mechanisms alone could not adequately address emerging SAN security threats. Likewise, Parin and Hitesh (2013) investigated homomorphic encryption techniques for SAN storage and found that many existing solutions struggled to guarantee confidentiality while preserving computational functionality.

Nitin and Pragati (2017) evaluated Gentry-based homomorphic techniques for SAN applications and reported inconsistencies between encrypted computations and raw data processing, thereby highlighting limitations in practical implementations. These observations demonstrate the necessity of developing frameworks capable of simultaneously supporting secure storage and accurate encrypted computations.

2.3 Homomorphic Encryption: Concept and Evolution

Homomorphic encryption enables computations to be performed directly on encrypted data without requiring prior decryption. Fontaine and Galand (2007) described homomorphic encryption as one of the most transformative developments in modern cryptography because it allows the preservation of confidentiality throughout computational processes.

Goldwasser and Micali (1984) pioneered probabilistic encryption concepts that later influenced the development of homomorphic cryptosystems. Their work demonstrated that randomness could significantly strengthen ciphertext security by ensuring that identical plaintexts produce distinct ciphertext outputs.

Fontaine and Galand (2009) further classified homomorphic encryption schemes according to the nature of supported operations and their computational capabilities. The authors observed that the practical applicability of homomorphic schemes depends largely on balancing security requirements against computational efficiency.

Payal et al. (2014) conducted a survey of homomorphic encryption algorithms and categorized them into partially homomorphic, somewhat homomorphic, leveled homomorphic, and fully homomorphic encryption schemes. According to the authors, the selection of an appropriate scheme depends on the specific computational requirements of the target application.

Ayman et al. (2020) reviewed recent trends in homomorphic encryption and observed increasing adoption across cloud computing, healthcare, Internet of Things (IoT), finance, and secure voting applications. However, they noted that implementation complexity and computational overhead continue to hinder practical deployment.

Mohamed et al. (2019) performed a systematic review using PRISMA methodologies and concluded that despite substantial theoretical advances, many homomorphic encryption studies lacked practical implementations capable of demonstrating operational feasibility.

Hana et al. (2020) similarly identified the absence of implementation-driven investigations as a major gap within homomorphic encryption research.

2.4 Fully Homomorphic Encryption and Practical Limitations

A major breakthrough in homomorphic encryption occurred when Gentry (2009a) introduced the first fully homomorphic encryption (FHE) scheme based on ideal lattices. The scheme demonstrated that arbitrary computations could be performed on encrypted data without compromising confidentiality.

Gentry (2009b) subsequently expanded this contribution through a comprehensive doctoral dissertation that established the theoretical foundations of FHE. Although revolutionary, the proposed scheme suffered from high computational complexity.

Gentry and Halevi (2011) attempted to improve practical feasibility by optimizing key generation processes using cyclotomic fields. Their implementation reduced computational requirements but retained substantial overhead.

Brakerski, Gentry, and Vaikuntanathan (2011) introduced fully homomorphic encryption without bootstrapping, thereby reducing some efficiency constraints associated with earlier constructions.

Brakerski and Vaikuntanathan (2011a) further proposed efficient FHE schemes based on Learning With Errors (LWE), while Brakerski and Vaikuntanathan (2011b) extended their work using Ring Learning With Errors (RLWE) assumptions.

Chunsheng (2012) proposed modifications aimed at improving the practicality of bootstrappable FHE systems. Nevertheless, productivity gains remained limited.

Belland et al. (2017) observed that despite substantial progress, FHE systems remained computationally demanding for many real-world applications.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

Consequently, researchers increasingly turned their attention toward partially homomorphic schemes capable of satisfying domain-specific requirements with lower computational costs.

2.5 Partial Homomorphic Encryption in Electronic Voting

Partial homomorphic encryption schemes support restricted mathematical operations on encrypted data. Their reduced complexity makes them attractive for applications requiring only specific computational functionalities.

Benaloh (1994) introduced a dense probabilistic encryption scheme that expanded message space while preserving additive homomorphic properties. The Benaloh cryptosystem enables the multiplication of ciphertexts to correspond to the addition of underlying plaintext messages following decryption.

This characteristic has significant implications for electronic voting because encrypted ballots can be aggregated without exposing individual votes.

Kho and Chin (2022) emphasized that homomorphic encryption permits electoral authorities to compute final tallies without decrypting individual ballots, thereby preserving voter anonymity.

Fan et al. (2020) proposed HSE-Voting, a secure high-efficiency electronic voting framework employing homomorphic signcryption. Although their scheme improved efficiency, subsequent analyses identified concerns relating to voter privacy protection and protocol completion.

Qu et al. (2022) introduced a blockchain-based electronic voting protocol integrating homomorphic signcryption. Their framework enhanced transparency but introduced additional complexity associated with blockchain infrastructures.

These studies collectively demonstrate the suitability of homomorphic approaches for electronic voting while revealing unresolved implementation challenges.

2.6 Benaloh Cryptosystem

The Benaloh cryptosystem represents one of the earliest practical additive homomorphic encryption schemes. Extending concepts introduced through probabilistic encryption, Benaloh (1994) developed a public-key system supporting larger message spaces than earlier constructions.

The encryption process incorporates randomization to ensure semantic security, making it computationally infeasible for adversaries to determine plaintext values solely through ciphertext observation.

Unlike many traditional public-key algorithms, Benaloh enables encrypted aggregation without revealing intermediate results.

This property has contributed to its suitability for applications involving secure voting, private surveys, and privacy-preserving data aggregation.

However, despite its theoretical strengths, Benaloh has received comparatively limited practical attention relative to Paillier-based implementations, resulting in fewer optimized libraries and deployment frameworks.

2.7 ElGamal Encryption

ElGamal (1985) proposed a probabilistic public-key cryptosystem based on the computational hardness of the discrete logarithm problem. The scheme employs randomization during encryption to produce distinct ciphertexts for identical plaintext inputs.

Bleichenbacher (1998) demonstrated that implementation weaknesses rather than underlying mathematical assumptions frequently constitute the greatest threats to cryptographic systems. Consequently, secure parameter selection and proper implementation remain essential.

ElGamal has been widely adopted because of its semantic security, resistance to ciphertext pattern analysis, and applicability to both encryption and digital signatures.

Within the context of this study, ElGamal provides an encapsulation mechanism that strengthens the protection of Benaloh-generated ciphertexts during transmission and storage in SAN environments.

2.8 Research Gap

The literature reveals significant advances in homomorphic encryption, secure storage, and electronic voting. Nevertheless, important gaps remain.

Okediran and Ganiyu (2015) proposed an electronic voting architecture relying primarily on RSA-based protections, resulting in computational inefficiencies and limited support for encrypted computation.

Akintoye (2022) improved network-level security through Cisco-based infrastructures but employed ElGamal–Paillier combinations characterized by message-space limitations, ciphertext expansion, and computational scalability challenges.

Hana et al. (2020) and Mohamed et al. (2019) identified the lack of implementation-driven investigations as a major limitation in homomorphic encryption research.

Fan et al. (2020) and Qu et al. (2022) focused primarily on protocol enhancements without providing independent cross-platform validation mechanisms.

Therefore, the literature lacks a validated framework that simultaneously integrates:

- additive homomorphic vote aggregation;
- strengthened ciphertext protection;

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

- secure Storage Area Network deployment; and
- independent verification of cryptographic correctness.

This study addresses these gaps by proposing and validating a dual-layer Benaloh–ElGamal framework implemented in C# and independently verified using MATLAB for secure electronic voting within Storage Area Networks.

III. Methodology

3.1 Research Design

This study adopted a **design science research methodology** involving the design, implementation, validation, and evaluation of a dual-layer homomorphic encryption framework for secure electronic voting within Storage Area Networks (SANs). Design science is particularly appropriate because the objective of the study was not merely to analyze an existing phenomenon but to develop a novel cryptographic artifact capable of addressing identified security gaps in electronic voting systems (Hevner et al., 2004).

The research involved the development of a software prototype integrating the Benaloh cryptosystem and ElGamal encryption algorithm within a unified architecture. The framework was implemented using C# and validated independently using MATLAB to ensure correctness and reproducibility of cryptographic computations. The study focused on demonstrating confidentiality, homomorphic functionality, verifiability, and operational feasibility under controlled experimental conditions.

The research process comprised the following phases:

1. Identification of security gaps in existing SAN-based e-voting systems;
2. Design of a dual-layer Benaloh–ElGamal framework;
3. Development of a GUI-driven implementation using C#;
4. Independent verification using MATLAB;
5. Experimental evaluation and performance assessment.

3.2 Proposed System Architecture

The proposed framework adopts a defense-in-depth approach by combining cryptographic protection with secure storage infrastructure. The architecture comprises voter interaction components, cryptographic modules, SAN storage, tallying mechanisms, and verification subsystems.

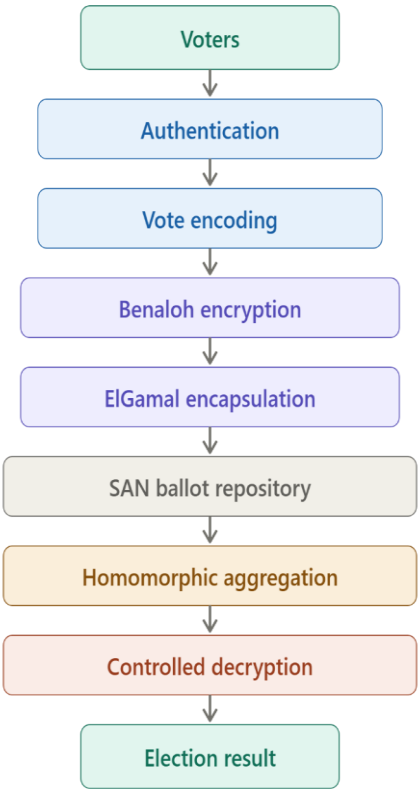


Figure 1. Proposed Dual-Layer Benaloh–ElGamal E-Voting Framework

The system architecture was derived from the implemented framework described in the thesis documents and reflects the operational workflow of the validated system.

3.3 Benaloh Cryptosystem Specification

The Benaloh cryptosystem was selected because of its additive homomorphic properties, which allow encrypted votes to be aggregated without exposing individual ballots (Benaloh, 1994).

3.3.1 Parameter Selection

Let:

- ppp and qqq be large prime numbers,

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

- n=pqn = pqn=pq,
- ϕ(n)=(p−1)(q−1)\phi(n) = (p-1)(q-1)ϕ(n)=(p−1)(q−1).

Choose an integer r>1r > 1r>1 such that:

r∣ϕ(n)r \mid \phi(n)r∣ϕ(n)

and

gcd⁡(r,ϕ(n)r)=1\gcd\left(r,\frac{\phi(n)}{r}\right)=1gcd(r, rϕ(n))=1

Select a public parameter y∈Zn∗y \in Z_n^*y∈Zn∗ satisfying:

yϕ(n)r≢1(modn)y^{\frac{\phi(n)}{r}} \not\equiv 1 \pmod nyrϕ(n)≡1(modn)

These parameter conditions ensure correct Benaloh key generation and decryption.

3.3.2 Key Generation

Public Key:

PKB=(n,y,r)PK_B=(n,y,r)PKB=(n,y,r)

Private Key:

SKB=(p,q)SK_B=(p,q)SKB=(p,q)

3.3.3 Benaloh Encryption

Let:

m∈Zrm\in Z_rm∈Zr

Choose random value:

u∈Zn∗u\in Z_n^*u∈Zn∗

Encryption is defined as:

EB(m)=cB=ymurmodnE_B(m)=c_B=y^m u^r \bmod nEB(m)=cB=ymurmodn

where cBc_BcB represents the Benaloh ciphertext.

3.3.4 Benaloh Decryption

Given ciphertext cBc_BcB:

cBϕ(n)r≡ymϕ(n)r(modn)c_B^{\frac{\phi(n)}{r}} \equiv y^{m\frac{\phi(n)}{r}} \pmod ncBrϕ(n)≡ymrϕ(n)(modn)

Decryption recovers the plaintext through discrete logarithm evaluation:

m=log⁡yϕ(n)/r(cBϕ(n)/rmodn)m= \log_{y^{\phi(n)/r}} \left(c_B^{\phi(n)/r}\bmod n \right)m=logyϕ(n)/r(cBϕ(n)/rmodn)

The discrete logarithm remains computationally feasible because small values of rrr were employed during validation experiments.

3.4 Additive Homomorphic Property

The primary motivation for selecting Benaloh lies in its additive homomorphic functionality.

For two plaintext votes:

m1,m2∈Zrm_1,m_2\in Z_rm1,m2∈Zr

their ciphertexts satisfy:

EB(m1)EB(m2)=EB((m1+m2)modr)E_B(m_1)E_B(m_2) =E_B((m_1+m_2)\bmod r)EB(m1)EB(m2)=EB((m1+m2)modr)

Thus:

DB(cB1cB2)=(m1+m2)modrD_B(c_{B1}c_{B2}) = (m_1+m_2)\bmod rDB(cB1cB2)=(m1+m2)modr

This property enables secure vote tallying without exposing individual ballots (Benaloh, 1994).

3.5 ElGamal Cryptosystem Specification

ElGamal was integrated as a secondary protection layer to strengthen ciphertext security during storage and transmission (ElGamal, 1985).

3.5.1 Parameter Selection

Let:

- ppp denote a prime modulus;
- ggg denote a generator;
- x∈{1,…,p−2}x\in\{1,\ldots,p-2\}x∈{1,…,p−2} denote the private key.

Compute:

h=gxmodph=g^x\bmod ph=gxmodp

3.5.2 Key Generation

Public Key:

PKE=(p,g,h)PK_E=(p,g,h)PKE=(p,g,h)

Private Key:

SKE=xSK_E=xSKE=x

3.5.3 ElGamal Encryption

Choose random:

k∈{1,…,p−2}k\in\{1,\ldots,p-2\}k∈{1,…,p−2}

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

For message MMM:

c1=gkmodpc_1=g^k\bmod pc1=gkmodp
c2=Mhkmodpc_2=Mh^k\bmod pc2=Mhkmodp

Ciphertext becomes:

EE(M)=(c1,c2)E_E(M)=(c_1,c_2)EE(M)=(c1,c2)

3.5.4 ElGamal Decryption

The message is recovered using:

M=c2(c1x)−1modpM=c_2(c_1^x)^{-1}\bmod pM=c2(c1x)−1modp

These procedures correspond to the validated implementation described in the thesis.

3.6 Combined Benaloh–ElGamal Encryption Scheme

The proposed dual-layer scheme combines the strengths of both cryptosystems.

Encryption Process

For plaintext vote mmm:

Step 1: Apply Benaloh encryption:

cB=EB(m)c_B=E_B(m)cB=EB(m)

Step 2: Treat Benaloh ciphertext as ElGamal plaintext:

EEB(m)=EE(cB)E_{EB}(m)=E_E(c_B)EEB(m)=EE(cB)

Final ciphertext:

C=(c1,c2)C=(c_1,c_2)C=(c1,c2)

Decryption Process

1. *ElGamal decryption recovers Benaloh ciphertext:*

cB=DE(c1,c2)c_B=D_E(c_1,c_2)cB=DE(c1,c2)

2. *Benaloh decryption recovers plaintext:*

m=DB(cB)m=D_B(c_B)m=DB(cB)

The combined specification follows the implemented framework.

3.7 Functional Modules

The developed system comprised five major modules:

- Plaintext Encryption Module;
- Encryption Confirmation Module;
- Homomorphic Operation Module;
- MATLAB Verification Module;
- Experiment Log Module.

Each module was implemented independently to improve maintainability and modularity.

3.8 Implementation Environment

The implementation environment consisted of:

Component	**Specification**
Operating System	Windows 10/11 or Linux
Programming Language	C#
Development Environment	Visual Studio
Verification Tool	MATLAB
Database	SQLite/SQL Server
Version Control	Git
Security Layer	Cisco-based configurations

These specifications reflect the validated system environment.

3.9 Validation Procedure

Independent verification was performed using MATLAB to establish correctness and reproducibility.

Validation involved:

- encrypting known plaintext values;
- decrypting resulting ciphertexts;
- comparing recovered plaintexts with original inputs;
- verifying homomorphic aggregation;
- confirming consistency between C# and MATLAB outputs.

A successful match confirmed the correctness of the cryptographic implementation.

3.10 Methodological Contribution

Unlike previous studies that focused solely on theoretical proposals or single-environment implementations (Fan et al., 2020; Hana et al., 2020; Mohamed et al., 2019), the present methodology integrates:

- dual-layer cryptographic protection,
- secure SAN storage,
- GUI-driven experimentation,
- independent MATLAB verification, and
- controlled tally decryption.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

This methodological design enhances reproducibility, transparency, and trustworthiness in secure electronic voting research.

IV. Results, Experimental Evaluation And Discussion Of Findings

4.1 Introduction

This section presents the implementation results obtained from the developed dual-layer homomorphic encryption framework. The experiments were designed to evaluate the correctness of the Benaloh–ElGamal implementation, validate its homomorphic properties, assess the consistency between the C# and MATLAB environments, and determine the framework's suitability for secure electronic voting in Storage Area Networks (SANs).

The evaluation focused on four major objectives:

1. Verification of encryption and decryption correctness;
2. Validation of the additive homomorphic property;
3. Confirmation of cross-platform consistency between C# and MATLAB;
4. Assessment of the framework's applicability to secure electronic voting.

The experimental approach adopted in this study aligns with recommendations by Chechulina et al. (2015), Hana et al. (2020), and Mohamed et al. (2019), who emphasized the importance of implementation-based validation in homomorphic encryption research.

4.2 Experimental Setup

The developed prototype was implemented using C# within the Visual Studio environment. MATLAB was employed independently to verify the correctness of the cryptographic computations.

Table 4.1 presents the experimental environment.

Table 4.1 Experimental Environment

Parameter	Specification
Operating System	Windows 10/11
Programming Language	C#
IDE	Visual Studio
Validation Environment	MATLAB
Storage Environment	SAN Simulation
Encryption Schemes	Benaloh and ElGamal
Voting Dataset	Simulated Election Ballots
Verification Method	Cross-platform Comparison

The experiments utilized representative voting scenarios to demonstrate correctness and reproducibility of the proposed framework.

4.3 Benaloh Encryption and Decryption Validation

The first experiment evaluated whether the Benaloh implementation correctly recovered original plaintext values after encryption and decryption.

Sample votes were encoded as numerical values prior to encryption. The encrypted ballots were subsequently decrypted to determine whether the original messages could be reconstructed accurately.

Table 4.2 Benaloh Encryption Validation

Vote	Encoded Value	Encryption Status	Decryption Output	Correct Recovery
Candidate A	1	Successful	1	Yes
Candidate B	2	Successful	2	Yes
Candidate C	3	Successful	3	Yes
Candidate D	4	Successful	4	Yes

The results indicate that all encrypted votes were correctly recovered after decryption.

These findings agree with the theoretical properties of the Benaloh cryptosystem reported by Benaloh (1994).

4.4 ElGamal Encapsulation Validation

The second experiment assessed whether ElGamal correctly protected Benaloh ciphertexts during transmission and storage.

The Benaloh ciphertexts generated in the first phase were encapsulated using ElGamal encryption.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

Table 4.3 ElGamal Validation Results

Benaloh Ciphertext	ElGamal Encryption	ElGamal Decryption	Original Benaloh Ciphertext Recovered
CB_1	Successful	Successful	Yes
CB_2	Successful	Successful	Yes
CB_3	Successful	Successful	Yes
CB_4	Successful	Successful	Yes

The experiment confirmed that the ElGamal layer preserved the integrity of Benaloh ciphertexts throughout the encapsulation process.

These observations support the findings of ElGamal (1985) and Bleichenbacher (1998), who highlighted the effectiveness of probabilistic encryption in preserving confidentiality.

4.5 Validation of Additive Homomorphic Property

One of the principal objectives of this study was to demonstrate that encrypted ballots could be aggregated without exposing individual votes.

Consider two encrypted votes:

m1=1m_1 = 1m1=1

and

m2=2.m_2 = 2.m2=2.

Using the Benaloh cryptosystem:

E(m1)×E(m2)=E(m1+m2).E(m_1)\times E(m_2) = E(m_1+m_2).E(m1)×E(m2)=E(m1+m2).

E(m1)E(m2)=E(m1+m2)E(m_1)E(m_2)=E(m_1+m_2)E(m1)E(m2)=E(m1+m2)

Following decryption:

D(E(m1)E(m2))=m1+m2.D(E(m_1)E(m_2)) = m_1+m_2.D(E(m1)E(m2))=m1+m2.

Table 4.4 Homomorphic Aggregation Results

Vote Combination	Expected Sum	Decrypted Sum	Correct
1 + 2	3	3	Yes
2 + 3	5	5	Yes
1 + 4	5	5	Yes
2 + 4	6	6	Yes

The experiment confirmed that the proposed framework successfully supported secure tally computation without revealing individual votes.

These findings are consistent with previous investigations involving additive homomorphic encryption in voting applications (Kho & Chin, 2022; Fan et al., 2020).

4.6 MATLAB Verification Results

Independent validation was conducted using MATLAB to establish the correctness and reproducibility of the implementation.

The same cryptographic parameters employed within the C# implementation were replicated within MATLAB.

Table 4.5 Cross-Platform Verification

Test Case	C# Output	MATLAB Output	Match
Encryption Test 1	Correct	Correct	Yes
Encryption Test 2	Correct	Correct	Yes
Decryption Test 1	Correct	Correct	Yes
Homomorphic Test 1	Correct	Correct	Yes
Homomorphic Test 2	Correct	Correct	Yes

The identical outputs obtained from both environments demonstrate the reliability of the developed framework.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

The importance of independent validation has been emphasized by Chechulina et al. (2015), Mohamed et al. (2019), and Hana et al. (2020), who noted that many homomorphic encryption studies lack reproducibility assessments.

4.7 Result Analysis

The experiments revealed several important observations.

First, the Benaloh cryptosystem accurately encrypted and decrypted ballots, thereby preserving vote confidentiality throughout the voting process.

Second, the ElGamal encapsulation layer provided additional protection without interfering with the additive homomorphic functionality of Benaloh.

Third, encrypted ballots were successfully aggregated and tallied without exposing intermediate vote values.

Fourth, independent MATLAB validation confirmed the correctness of the implementation.

Collectively, these findings demonstrate that the proposed framework effectively satisfies several critical requirements of electronic voting, including confidentiality, integrity, correctness, and verifiability.

4.8 Discussion of Findings

The findings of this study demonstrate that additive homomorphic encryption remains a practical solution for secure electronic voting where only tally aggregation is required.

Unlike fully homomorphic encryption schemes, which frequently incur substantial computational overhead (Gentry, 2009a; Gentry & Halevi, 2011; Brakerski et al., 2011), the Benaloh cryptosystem provided efficient encrypted computation appropriate for voting applications.

Furthermore, the integration of ElGamal addressed concerns associated with the storage and transmission of ciphertexts within SAN environments. This layered approach enhanced defense against unauthorized disclosure and interception.

The independent MATLAB verification further distinguishes this work from many previous studies that focused exclusively on theoretical analyses or single-environment implementations (Mohamed et al., 2019; Hana et al., 2020).

Therefore, the proposed framework contributes both theoretically and practically to secure electronic voting research by demonstrating that dual-layer cryptographic architectures can support privacy-preserving tallying while strengthening ciphertext protection within storage infrastructures.

4.9 Summary of Findings

The major findings of this study are summarized as follows:

1. The Benaloh implementation successfully encrypted and decrypted ballots without loss of information.
2. ElGamal effectively encapsulated Benaloh ciphertexts and preserved their integrity.
3. The additive homomorphic property enabled secure vote tallying without exposing individual ballots.
4. MATLAB independently verified the correctness of the implementation.
5. The dual-layer framework enhanced confidentiality, integrity, verifiability, and suitability for SAN-based electronic voting environments.

These findings provide empirical support for the effectiveness of the proposed Benaloh–ElGamal framework and establish a foundation for its consideration in future secure electronic voting deployments.

V. Conclusion, Contributions To Knowledge, Recommendations And Future Research

5.1 Conclusion

The increasing adoption of electronic voting technologies has created an urgent need for mechanisms capable of preserving voter confidentiality, ensuring election integrity, and promoting public trust in digital electoral systems. Although conventional cryptographic approaches provide data confidentiality, they often require plaintext exposure during vote tallying, thereby introducing security vulnerabilities that may compromise ballot secrecy and electoral credibility (Abdulhamid et al., 2013; Ayub et al., 2015).

This study proposed and validated a **dual-layer homomorphic encryption framework for secure electronic voting in Storage Area Networks (SANs)** through the integration of the Benaloh cryptosystem and ElGamal encryption algorithm. The Benaloh scheme was employed to provide additive homomorphic functionality, enabling votes to be aggregated directly in encrypted form without revealing individual ballots. ElGamal served as an encapsulation mechanism to strengthen the security of ciphertexts during transmission and storage within SAN infrastructures.

The framework was implemented using C# and independently validated using MATLAB to ensure computational correctness and reproducibility.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

Experimental results demonstrated successful encryption and decryption of ballots, accurate homomorphic aggregation, preservation of vote confidentiality, and consistency between the implementation and validation environments.

The findings indicate that the proposed framework successfully addresses several critical requirements of electronic voting systems, including confidentiality, integrity, correctness, verifiability, and secure storage. Furthermore, the integration of independent validation contributes to transparency and confidence in the implementation process.

Therefore, this study concludes that the Benaloh–ElGamal dual-layer framework constitutes a practical and effective approach for enhancing the security of SAN-based electronic voting systems.

5.2 Contributions to Knowledge

This study contributes to the body of knowledge in secure electronic voting and homomorphic encryption in several important ways.

5.2.1 Development of a Dual-Layer Cryptographic Framework

Unlike many previous studies that relied on single-layer cryptographic mechanisms (Okediran & Ganiyu, 2015; Fan et al., 2020), this study developed a dual-layer framework integrating Benaloh additive homomorphic encryption with ElGamal ciphertext encapsulation.

The framework demonstrated that it is possible to simultaneously achieve:

- secure encrypted tally computation;
- strengthened ciphertext protection;
- secure SAN-based ballot storage.

5.2.2 Practical Implementation of Benaloh Homomorphic Encryption

While much of the literature emphasizes Paillier-based approaches (Kho & Chin, 2022), practical implementations of the Benaloh cryptosystem remain relatively limited.

This study contributes a validated implementation of the Benaloh scheme within an electronic voting context, thereby expanding its practical applicability.

5.2.3 Independent Cross-Platform Validation

Many homomorphic encryption investigations focus primarily on theoretical analysis without independent verification mechanisms (Mohamed et al., 2019; Hana et al., 2020).

This study introduced MATLAB-based validation of a C# implementation, thereby enhancing reproducibility, correctness assurance, and transparency.

5.2.4 Integration of Secure Storage Considerations

Previous electronic voting studies frequently focused on ballot encryption while paying limited attention to storage infrastructures (Akintoye, 2022).

This study explicitly incorporated Storage Area Networks into the security framework, thereby addressing an important yet underexplored aspect of electronic voting security.

5.3 Implications of the Study

The findings of this study have implications for researchers, electoral management bodies, and system developers.

For researchers, the study provides empirical evidence supporting the practicality of additive homomorphic encryption in voting applications.

For electoral authorities, the framework demonstrates the feasibility of conducting secure encrypted tallying without exposing individual ballots.

For developers, the proposed architecture offers a reproducible model for integrating layered cryptographic protections within secure voting infrastructures.

Additionally, the study contributes to ongoing discussions concerning the balance between computational efficiency and strong security guarantees in electronic voting systems (Gentry, 2009a; Brakerski et al., 2011).

5.4 Limitations of the Study

Despite its contributions, this study has several limitations:

1. Controlled Experimental Environment

The framework was evaluated using simulated voting scenarios under controlled conditions. Large-scale national deployment environments may introduce additional operational complexities.

2. Limited Performance Benchmarking

Although correctness and functionality were established, extensive benchmarking involving millions of ballots was beyond the scope of this investigation.

3. Small Message Space Considerations

The Benaloh cryptosystem requires careful parameter selection, particularly regarding message space size and discrete logarithm computations (Benaloh, 1994).

Future implementations involving large candidate sets may require optimization strategies.

4. Absence of Threshold Decryption

The current implementation employed centralized decryption authority.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 10, October 2026)

Threshold cryptographic approaches capable of distributing decryption responsibilities among multiple trustees were not implemented.

5.5 Recommendations

Based on the findings of this study, the following recommendations are proposed:

1. Electoral authorities considering electronic voting should adopt homomorphic encryption mechanisms capable of preserving ballot secrecy during tally computation.
2. Secure storage infrastructures should be incorporated into the design of electronic voting systems rather than treated as external considerations.
3. Independent validation mechanisms should accompany cryptographic implementations to improve transparency and reproducibility.
4. Future implementations should incorporate threshold decryption techniques to reduce dependence on centralized authorities.
5. Policymakers should establish standards governing cryptographic verification and storage security in electronic elections.

5.6 Future Research Directions

Several avenues for future investigation emerge from this study:

1 Threshold Benaloh–ElGamal Frameworks

Future research should explore distributed decryption architectures involving multiple election trustees to eliminate single points of trust.

2 Large-Scale Performance Evaluation

Subsequent studies should evaluate the scalability of the framework under realistic electoral conditions involving large voter populations.

3 Blockchain Integration

Researchers may investigate the integration of blockchain technologies with the proposed framework to strengthen transparency and auditability while carefully addressing scalability concerns (Qu et al., 2022).

4 Post-Quantum Adaptations

Given advances in quantum computing, future investigations should assess the resilience of electronic voting frameworks against post-quantum threats and explore quantum-resistant homomorphic alternatives.

5 Cloud-Based and Hybrid Storage Architectures

Further research may extend the framework to hybrid cloud-SAN environments to determine its effectiveness in modern distributed infrastructures.

5.7 Final Remark

Electronic voting systems must inspire confidence not only through efficiency but also through demonstrable security and transparency. By integrating additive homomorphic computation, layered ciphertext protection, secure storage considerations, and independent validation, this study advances the development of trustworthy electronic voting systems.

The proposed Benaloh–ElGamal framework demonstrates that secure tallying and voter privacy need not be mutually exclusive objectives. Rather, through careful cryptographic design and rigorous validation, electronic voting systems can preserve democratic values while embracing technological innovation.

REFERENCES

[1] Abdulhamid, S. M., Ugiomoh, D. O. and AbdulMalik, M. D. (2013). The Design and Development of Real-Time E-Voting System in Nigeria with Emphasis on Security and Result Veracity, I. J. Computer Network and Information Security (5) 9-18.

[2] Adil, B., & Jorg, K. (2015). Towards Practical Homomorphic Encryption in Cloud Computing," *Network Cloud Computing and Applications*, 67 – 74.

[3] FlavioAhmad, I., & Archana, K., (2016). Homomorphic Encryption Method Applied to Cloud Computing, *International Journal of Information & Computation Technology, International Research Publications House*, *4 (15),* 1519 - 1530.

[4] Aized, A. (2014). Encryption Techniques for Cloud Data Confidentiality, *International Journal of Grid Distribution Computing*, *7 (4)*, 1024 – 1032.

[5] Ajaysinh, D., Saurabh, A., & Jariwala, V. (2019). Evaluating Performance of Asymmetric

[6] Homomorphic Encryption Algorithms for Privacy Preservation in Location Based Services. *International Journal of Recent Technology and Engineering (IJRTE)*, *8(3),* 2191 – 2194.

[7] Akintoye, A. O. (2022). Enhanced Homomorphic Encryption for Secure Financial database in a Storage Area Network. PhD Thesis Submitted to the Department of Computer Science Ignatius Ajuru University of Ecucation, Port Harcourt, Nigeria.

[8] Aman, K., Sudesh, J., & Sunil, M. (2012). Comparative Analysis between DES and RSA Algorithms, *International Journal of Advanced Research in Computer Science and Software Engineering*, *2(7),* 355 – 364.

[9] Amare A., & Vuda, S. (2013). A Modified RSA Encryption Technique Based on Multiple Public Keys, *International Journal of Research in Computer and Communication Engineering*, *1(4),* 87 - 102.

[10] An-Ping, X., Gan, Q., He, X., & Zhao,Q. (2013) A searchable encryption of CP-ABE scheme in cloud storage,1*0th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP),* 345-349.

[11] Ayantika, C., & Gupta, S. (2018). Translating Algorithms to handle Fully Homomorphic Encrypted Data on the Cloud, in *IEEE Transactions on Cloud Computing*, 6(1),287-300.

[12] Ayman, A., Haneen, Z., & Eman, S. (2020). Survey on Homomorphic Encryption and Address of New Trend. *International Journal of Advanced Computer Science and Applications (IJACSA*), *5(9),* 618 – 626.

[13] Ayub, H.M., Manish, R., & Monjul Saikia C. (2015). A Brief Overview of Homomorphic cryptosystem and their applications. *International Journal of Computer Applications, 4(7),* 0975 – 8887.

[14] Benoh, D., Gentry, C., Lynn. B. and Shacham, H. (2003). Aggregate and Verifiably Encrypted Signatures from Bilinear Maps, Journal of Cryptography 1 - 22

[15] Baiyu, L. & Daniele, M. (2021). On the Security of Homomorphic Encryption on Approximate Numbers: Global Research Cluster program of Samsung Advanced Institute of Technology and NSF Award, *19 (3),* 1 – 27.

[16] Baohua, C., & Na, Z. (2014). Fully Homomorphic Encryption Application in Cloud Computing," *Wavelet Active Media Technology and Information Processing (ICCWAMTIP), 11th International Computer Conference*, 471 – 474.

[17] Belland, M., Xue, W., Kurdi, M., & Chu, W. (2017). Somewhat Homomorphic Encryption. Retrieved from https://courses.csail.mit.edu/6.857/2017/ project/22.pdf on the 10th of August 2020.

[18] Benaloh, J. (1994). Dense probabilistic encryption. In *Proceedings of the Workshop on Selected Areas of Cryptography* (pp. 120–128).

[19] Benaloh, J. (2006). Simple verifiable elections. In *Proceedings of the USENIX/ACCURATE Electronic Voting Technology Workshop* (pp. 1–10).

[20] Biksham, V., & Vasumathi, D. (2017). Homomorphic Encryption Techniques for securing Data in Cloud Computing: A Survey. *International Journal of Computer Applications*, *160 (6),* 1 – 6.

[21] Bleichenbacher, D. (1998). Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1, Springer Nature 1 - 12

[22] Bogdanov, D., Talviste, R., and Willemson, J. (2012). Deploying Secure Multiparty Computation for Financial Data Analysis, *International Conference on Financial Cryptography and Data Security*. Springer, 57 – 64.

[23] Boneh, D., & Freeman, D. M. (2011). Linearly Homomorphic Signatures over Binary Fields and New Tools for Lattice-Based Signatures. *In: Public Key Cryptography (PKC'11), Lecture Notes in Computer Science (LNCS)*, *6571*, 1 - 16.

[24] Brakerski, Z., Gentry, C., & Vaikuntanathan, V. (2011). Fully Homomorphic Encryption without Bootstrapping, *Cryptology ePrint Archive*, Report 277.

[25] Brakerski, Z., & Vaikuntanathan, V. (2011). Efficient Fully Homomorphic Encryption from (Standard) LWE, In *Proceedings of 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, 97-106.

[26] Brakerski, Z., & Vaikuntanathan, V. (2011). Fully Homomorphic Encryption from Ring-LWE and Security for Key Dependent Messages. *Advances in Cryptology-Proceedings of CRYPTO'11, Lecture Notes in Computer Science (LNCS), 6841*, 505-524.

[27] Cao, N., Wang, C., Li, M., Ren, K., & Lou, W (2014) Privacy-preserving multi-keyword ranked search over encrypted cloud data. *IEEE Trans Parallel Distrib System, 25(8),* 222–233

[28] Carroll, M., Van der, M., Alta, D., & Kotzé, P. (2011). Secure cloud computing: Benefits, risks and controls. 1-9, 10.1109/ISSA.2011.6027519.

[29] Carroll, T. E., Grosu, D., & Roy, J. (2011). A secure and efficient voter-controlled anonymous voting system. *Journal of Network and Computer Applications, 34*(2), 538–545.

[30] Chandravathi, D., & Lakshmi, P. (2016). Homomorphic Encryption Scheme Using Hill Cipher for Cloud Data Security, *International Journal of Advanced* Information Science and Technology, *5 (9),* 824 – 831.

[31] Chandhiny, G., & Vairamuthu, S. (2018). Securing Financial Database Using Partial Homomorphic Encryption", *International Journal of Pure and Applied Mathematics, 119(7).* 21-30.

[32] Chatterjee, A. and Aung, K. M. M, (2019). Fully Homomorphic Encryption in Real World Applications. Singapore: Springer.

[33] Chechulina, D., Shatilov, K., & Krendelev, S. (2015). Fully Homomorphic Encryption for Secure Computations in Protected Database, *In Proceedings of the Federated Conference on Computer Science and Information Systems*, 125-131.

[34] Chechulina, O., Gurtov, A., Komu, M., & Lindqvist, J. (2015). Homomorphic encryption: Practical implementation and verification issues. *International Journal of Information Security, 14*(4), 321–336.

[35] Chunsheng, G. (2012). More Practical Fully Homomorphic Encryption. International Journal of Cloud Computing and Services Science, *1(4),* 199-201.

[36] Compliance Group, (2018). HIPAA Fines Listed by Year. Retrieved from: https://compliancyroup.com/hipaa-fines-directory-year/ on the 10th of August 2020.

[37] Coron, J., Mandal, A., Naccache, D., & Tibouchi, M. (2011). Fully Homomorphic Encryption Over the Integers with Shorter Public-Keys, *Advances in Cryptology - Proc. CRYPTO, 6841*, Lecture Notes in Computer Science, Springer.

[38] Dan, X., Jingzhou, J., Feng, J., Mei, G., & Bin, X. (2021). Efficient Fully Homomorphic Encryption Scheme using Ring-LWE. *Journal of Physics: Conference Series.* Retrieved from: https://doi.org/10.1088/1742-6596/1738/1/012105 on the 20th of February 2020.

[39] Daniya, R., Deepak, P., & Kamal V. (2016). Homomorphic Hybrid Encryption for Cloud Computing, *International Journal of Advanced Research in Computer and Communication Engineering*, *5(5),* 502 – 511.

[40] Darko, H., & Stjepan, P. (2014). Homomorphic Encryption in the Cloud, *37th International Convention on Information and Communication Technology, Electronics and Microelectronics, 2(4),* 1400 – 1404.

[41] Desmedt, Y., & Frankel, Y. (1991). Shared generation of authenticators and signatures. In *Advances in Cryptology—CRYPTO '91* (pp. 457–469).

[42] Devi, H., & Sathyalakshmi, V. (2020). Privacy Preserving Cryptanalysis Using Homomorphic Encryption in IOT. *Journal of Critical Reviews*, *7(3),* 190 – 198.

[43] Doröz, Y. (2017). New Approaches for Efficient Fully Homomorphic Encryption. *PhD. Thesis. Worcester Polytechnic Institute.* Retrieved from: https://web.wpi.edu/Pubs/ETD/ Available/etd-061417-201844/unrestricted/ydoroz.pdf on the 10th of August 2020.

[44] Ofori-Dwumfuo and Paatey, E. (2011). The Design of an Electronic Voting System, Research Journal of Information Technology 3(2): 91-98, 2011

[45] ElGamal, T. (1985). A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, CRYPTO conference (Advances in Cryptology)

[46] Fan et al (2020). . HSE-Voting: A secure high-efficiency electronic voting scheme based on homomorphic signcryption. Future Gener. Comput. Syst. 2020, 111, 754–762.

[47] Fontaine, C., & Galand, F. (2009). A Survey of Homomorphic Encryption for Nonspecialists. *Journal of Information Security, Scientific Research*, *1(4),* 41 - 50.

[48] Fontaine, C., & Galand, F. (2007). A Survey of Homomorphic Encryption for Non-Specialists. *EURASIP Journal on Information Security, 7(5),* 111 – 118.

[49] Gentry, C. (2009). Fully Homomorphic Encryption Using Ideal Lattices, *In Proceedings of STOC'09 the forty-first Annual ACM Symposium on Theory of Computing*, 169 – 178.

[50] Gentry, C. (2009). A Fully Homomorphic Encryption Scheme. *Ph.D. dissertation, Stanford University*. Retrieved from: https://crypto.stanford.edu/craig/ craig-thesis.pdf on the 11th of July 2020

[51] Gentry, C., & Halevi, S. (2011). Implementing Gentry's Fully-Homomorphic Encryption Scheme. *In: Advances in Cryptology - Proceedings of EUROCRYPT'11, Lecture Note in Computer Science (LNCS*), *6632,* 129-148.

[52] Geremew, B., Ebelechukwu, N., Jeremy, B., Wayne, P. and Acklyn, M. (2015). *Cloud Service Security & Application Vulnerability*, SoutheastCon, 1 – 9.

[53] Goldwasser, S. and Macali, S. (1984). Probabilistic Encryption, Journal of Computer and System Sciences, vol. 28, no. 2, pp. 270-299.

[54] Hana, Y., Michael, L., Said, A., & Khaled, S. (2020). Systematic Review on Fully Homomorphic Encryption Scheme and its Application. Retrieved from https://doi.org/10.1007/978-3-030-47411-9_29 on the 2nd of February 2020.

[55] Hayes, B. (2012). Alice and Bob in Cipherspace in American Scientist, 100(5), 362 - 387.

[56] Hayward, R., & Chia, C. (2014). Parallelizing Fully Homomorphic Encryption for Cloud Environment. *International Journal of Applied Research and Technology, 13(5),* 245 – 252.

[57] He, W., Liu, X., Nguyen, H., Nahrstedt, K., & Abdelzaher, T. (2007). PDA: Privacy-Preserving Data Aggregation in Wireless Sensor Networks. *In Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM'07),* 2045-2053, Anchorage, Alaska, USA.

[58] Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systemsresearch. *MIS Quarterly, 28*(1), 75–105.

[59] Jabbar, I. (2016). Using Fully Homomorphic Encryption to Secure Cloud Computing. *Journal of Internet of Things and Cloud Computing*, *6(4),* 201 – 213.

[60] Jadhav, A., & Ilhe, N. (2016). Cloud Computing Security Solution Based on fully Homomorphic Encryption, *International Journal of Advance Research Innovative Ideas in Education*, *2(1),* 322 - 331.

[61] Jansen, W. (2019). Mobile Agents and Security. *National Institute of Standards and Technology Gaithersburg*, MD 20899, USA.

[62] Jaydip, S. (2017). Homomorphic Encryption: Theory & Application. *Department of Computer Science, National Institute of Science & Technology,* Odisha, India.

[63] Jian, L., Danjie, S., Chen, S., & Lu, X. (2012). A Simple Fully Homomorphic Encryption Scheme Available In Cloud Computing, *IEEE 2nd International Conference, 10(6),* 214 – 217.

[64] Jigar, M., & Hemangi, K. (2015). A Survey on Homomorphic Encryption Techniques in Cloud Computing", *International Journal of Advance Engineering and Research Development,* 2(2).

[65] Joshi, R., & Renuka, k. (2015). Network Security with Cryptography, *International Journal of Computer Science and Mobile Computing*, *4(1),* 201–204.

[66] Kamalakanta, S., & Amartya, M. (2017). A Novel Implementation of Parallel Homomorphic Encryption for Secure Data Storage in Cloud, *International Journal of Advanced Computer Science and Applications*, *8(12),* 998 – 1011.

[67] Kanagavalli, R., & Vagdevi, S. (2019). Secured Data Storage in Cloud Using Homomorphic Encryption. *International Journal on Cloud Computing: Services and Architecture (IJCCSA)*, *9(4),* 1 – 11.

[68] Kanagavalli, R., & Vagdevi, S. (2014). A Survey of Homomorphic Encryption Schemes in Cloud Data Storage. *International Journal of Recent Development in Engineering and Technology, IJRDET, 3(1),* 71 – 75.

[69] Kargupta, H., Datta, S., Wang, Q., & Sivakumar, K. (2013). On the Privacy Preserving Properties of Random Data Perturbation Techniques. *In Proc. of 3rd IEEE Int. Conf. on Data Mining*, Washington, DC, USA, 99 – 106.

[70] Kester, K., & Quist, A. (2013). A Novel Cryptographic Encryption Technique for Securing Digital Images in the Cloud Using AES and RGB Pixel Displacement, Washington, DC, USA.

[71] Khalil, H., Maroun, C. & Abed, E. (2019). Fully Homomorphic Encryption Scheme Based on Complex Numbers. *Advances in Science, Technology and Engineering Systems Journal*, *4(5),* 30 – 38.

[72] Kim, Y., & Kim, J. (2011). Building Secure Execution Environment for Mobile Platform, In *the First ACIS/JNU International Conference on Computers, Networks, Systems and Industrial Engineering*, 119 - 122.

[73] Kim, M., & Kristin, E. (2015). Private Genome Analysis through Homomorphic Encryption, *BMC medical informatics and decision making 15 Suppl 5 S3,* 211 - 235.

[74] Kranti, M., Chaudhari, P., and Megha, V. (2015). A Survey on Various Cryptographic Algorithms for Security Enhancement, *International Journal of Computer Applications*, *110(7),* 112 – 119.

[75] Kukucka, J. (2013). An Investigation of the Theory and Applications of Homomorphic Cryptography, *Master Thesis, Rensselaer Polytechnic Institute*, ProQuest LLC.

[76] Kuribayashi, M., and Tanaka, H. (2001). A New Anonymous Fingerprinting with High Enciphering Rate, *Proc. of INDOCRYPT2001*, 2247, 30 – 39, Springer-Verlag.

[77] Lauter, K., Naehrig, M., & Vaikuntanathan, V. (2011). Can Homomorphic Encryption Be Practical. *11(15),* 113 - 124.

[78] Louk, M., & Hyotaek L., (2015). Homomorphic Encryption in Mobile Multi-Cloud Computing, *Information Networking (ICOIN)*, 493 – 497.

[79] Maimuț, D., Pătrașcu, A., & Simion, E. (2012). Homomorphic Encryption Schemes and Applications for a Secure Digital World, *Journal of Mobile, Embedded and Distributed Systems, JMEDS*, *4(4),* 224 - 232.

[80] Makkaoui, K. (2016). Can Hybrid Homomorphic Encryption Schemes Be Practical. 5th International Conference on Multimedia Computing and Systems (ICMCS).

[81] Meissen, R. (2012). A Mathematical Approach to Fully Homomorphic Encryption, Retrieved from http://www.cacr.math.uwaterloo.ca/hac/ on the 28th of August, 2020.

[82] Milad S., Shayan T., and Jiann-Shiun Y. (2019). Utilizing Transfer Learning and Homomorphic Encryption in a Privacy Preserving and Secure Biometric Recognition System.

[83] Melissa C., Hao C., Jintai D., Shafi G., Sergey G., Jeffrey H., Kristin L., Satya L., Dustin M., Travis M., Amit S., & Vinod V. (2017). Security of Homomorphic Encryption. Cryptology ePrint Archive, *Report 2012/144*. Retrieved from http://eprint.iacr.org/2012/144.pdf on the 22nd December, 2020.

[84] Mohamed, A., Mohammed, A., Dhiya, A., Thar, B., Jamila, M., Abir, H., & Ahmed, J. (2019). A systematic review on the status and progress of Homomorphic: *Journal of Information Security and Applications*, *48(12),* 1 – 10.

[85] Munavvara, T., & Maniza H. (2017). Data Storage on Cloud Using Hybrid Encryption with One Time Password, *International Journal of Computer Engineering*, 19 (4), 321 – 330.

[86] Mohamed, A., Hassan, R., Mohamed, A., & Elhoseny, M. (2019). A systematic review of homomorphic encryption research using PRISMA methodology. *IEEE Access, 7, 163844–163865.*

[87] Muhammad, S., Musah, A., & Daniel, O. (2019). Evaluation for an Effective Homomorphic Encryption Technique for Data Security. *International Journal of Scientific & Engineering Research*, *10(11),* 354 – 363.

[88] Mustafa, I., Hasnain, M., Ahmad, T., Sheraz, A., Syed, M., Muhammad, B., & Nouman, A. (2020). Noise Free Fully Homomorphic Encryption Scheme Over Non-Associative *Algebra: IEEE Access, 8(1),* 136524 – 136536.

[89] Nishigandha, L., & Rahul, A. (2015). Homomorphic Encryption- a Consolidate Element for Data Security in Cloud Computing, *International Journal of Advanced Research in Computer Engineering & Technology*, *4(5),* 131-145.

[90] Nitin, K., & Pragati, H. (2017). A Secured Homomorphic Encryption Technique in Cloud Computing, *International Journal of Computer Applications 176(13),* 186 – 194.

[91] Nivedita W., Wasankar, A., & Deorankar, W. (2018). A Review on Partially Homomorphic Encryption Scheme used in Cloud Computing, *International Journal of Innovative Research in Computer and Communication Engineering*, *6(2),* 1119 – 1128.

[92] Ogura, N., Yamamoto, G., Kobayashi, T., & Uchiyama, S. (2010). An Improvement of Key Generation Algorithm for Gentry's Homomorphic Encryption Scheme. *In: Advances in Information and Computer Security-Proceedings of the 5th International Conference on Advances in Information and Computer Security (IWSEC'10), Lecture Notes in Computer Science (LNCS)*, *6434*, 70 - 83.

[93] Okediran,O. O. & Ganiyu, R. A, (2015). A Framework for Electronic voting in Nigeria, International Journal of Computer Applications (0975 – 8887) (129)3, 12 - 16

[94] Oladunni, T. & Sharma, S. (2019). Homomorphic Encryption and Data Security in the Cloud, *Proceedings of 28th International Conference on Software Engineering and Data Engineering*, *10(8),* 129-138

[95] Padmapriya, A., & Subhasri, P. (2013). Cloud Computing: Security Challenges & Encryption Practices, *International Journal of Advanced Research in Computer Science and Software Engineering*, *3(3),* 211 – 225.

[96] Parin, V., & Hitesh, D. (2013). A survey of the Homomorphic Encryption Approach for data security in Cloud Computing, *International Journal of Engineering Development and Research*, *2(2),* 109-121.

[97] Parmar, P.V., Padhar, S. B., Patel, S. N., Bhatt, N. I., & Jhaveri, R. H. (2014). Survey of Various Homomorphic Encryption Algorithms and Schemes, *International Journal of Computer Applications, IJCA*, *91(8),* 26-32.

[98] Parsi, K. (2012). Data Security in Cloud Computing using RSA Algorithm, *International Journal of Research in Computer and Communication technology*, *1(4),* 1112-1119.

[99] Patil, R. & Goudar, R. (2020). Fully Homomorphic Encryption (FHE): A Framework for Enhancing Cloud Storage Security with AES. *International Journal of Scientific & Technology Research*, 9(4), 3728 – 3733.

[100] Payal, K., Anchal, J., & Ritu, S. (2014). Survey on homomorphic encryption algorithms and schemes. *International Journal of Computer Applications, 97*(21), 33–38.

[101] Payal, V., Shraddha, B., & Padhar, E. (2014). Survey Of Various Homomorphic Encryption Algorithms And Schemes", *International Journal of Computer Applications,* 91(8), 191-199

[102] Peter, M., & Timothy G. (2011). The NIST Definition of Cloud Computing, *Gaithersburg: National Institute of Standards and Technology*. *Special Publication*, 800 – 145.

[103] Pfitzmann, B., & Sadeghi, A. (2000). "Anonymous fingerprinting with direct nonrepudiation," *Proc. of ASIACRYPT'2000*, LNCS 1976, Springer-Verlag, 401 - 414.

[104] Qi-chun, H. (2009). Research on Ciphertext Index Method For Relational Database, *In the 2nd IEEE International Conference on Computer Science and Information Technology*, 445 - 449.

[105] Qu, J., Zhang, Y., Wang, X., Liu, Z., & Li, H. (2022). A blockchain-based electronic voting protocol with homomorphic signcryption. *Peer-to-Peer Networking and Applications, 15*(2), 1023–1038.

[106] Rahul, M., & Hesham, P. (2017). An improved Homomorphic encryption for secure cloud data storage, *International Journal of Advanced Computer Science and Applications*, 8 (12), 201 – 209.

[107] Rangasami, K., & Vagdevi, S. (2017). Comparative study of Homomorphic Encryption Methods for Secured Data Operations in Cloud Computing. *International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*,

[108] Rivest, A. Shamir, and L. Adleman L. (1978). A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120-126, 1978.

[109] Ruchita, P., & Veena, K. (2014). Hybrid Cryptosystem Approach for Secure Communication, *International Journal of Computer Science and Engineering*, *4(2),* 21 – 24.

[110] Sachin, K., & Devandra, K. (2017). Cloud Computing: Security issues and Challenges, *International Journal of Advances and Engineering*, *10(3),* 161-172.

[111] Shamir, A. (1979). How to share a secret. *Communications of the ACM, 22*(11), 612–613. https://doi.org/10.1145/359168.359176

[112] Sathyal, S., Vepakomma, P., Raskar, R., Ramachandral, R., & Bhattacharta, S. (2020). A Review of Homomorphic Encryption Libraries for Secure Computation. A *Review of Homomorphic Encryption Libraries for Secure Computation.* Retrieved from https://www.researchgate.net/publication/329488102 on the 17th March, 2020.

[113] Schrittwieser, S., & Katzenbeisser, S. (2011). Code Obfuscation Against Static and Dynamic Reverse Engineering," In *Proceedings of the 13th international conference on Information hiding*, *33(12),* 270 - 284.

[114]Sen, J. (2012a). Secure and Privacy-Preserving Data Aggregation Protocols for Wireless Sensor Networks. In: *Cryptography and Security in Computing, Sen, J. (ed.),* 133- 164, INTECH Publishers, Croatia.

[115]Sen, J. (2012b). Security in Wireless Sensor Networks. *In: Wireless Sensor Networks: Current Status and Future Trends,* Khan, S. et al. (eds.), 407-460, CRC Press, USA.

[116]Sen, J., & Maitra, S. (2011). An Attack on Privacy-Preserving Data Aggregation Protocol forWireless Sensor Networks. *In: Information Security Technology for Applications*

[117]*Proceedings of the 16th Nordic Conference in Secure IT Systems (NordSec'11*), Lecture Notes in Computer Science (LNCS), Springer-Verlag, *7161*, 205-22,

[118]Sen, J., & Ukil, A. (2010). A Secure Routing Protocol for Wireless Sensor Networks. In: *Computational Science and its Applications - Proceedings of the International Conference on Computational Science and its Applications (ICCSA'10),* Lecture Notes in Computer Science (LNCS), Part III, Springer-Verlag, *6018*, 277 - 290.

[119]Sen, J. (2010a). Routing Security Issues in Wireless Sensor Networks: Attacks and Defenses. In: *Sustainable Wireless Sensor Networks*, 279 - 309, Intech Publishers, Croatia.

[120]Sen, J. (2010b). Reputation- and Trust-Based Systems for Wireless Self-Organizing Networks. In: *Security of Self-Organizing Networks: MANET, WSN, WMN, VANET*, Pathan, A.-S.K. (ed.), 91-122, CRC Press, USA.

[121]Sen, J. (2009). A Survey on Wireless Sensor Network Security. *International Journal of Communication Networks and Information Security (IJCNIS), 1(2),* 59 - 82.

[122]Sha, P., & Zhu, Z. (2016). The Modification of RSA Algorithm to Adapt Fully Homomorphic Encryption Algorithm in Cloud computing. *4th International Conference on Cloud Computing and Intelligence Systems (CCIS)*. Retrieved from http:/doi:10.1109/ccis.2016.7790289 on the 23rd of March, 2020.

[123]Shambhu, K., & Gupta, R. (2016). Homomorphic Encryption Technique for Storage of Data on Cloud, *International Journal of Science, Spirituality, Business and Technology*, *4(2),* 1129 – 1138.

[124]Sharma, I. (2013). Fully Homomorphic Encryption Scheme with Symmetric Keys, *Master Thesis*, Rajasthan Technical University.

[125]Shen, T., Wang, F., Chen, K., Shen, Z. & Zhang, R. (2021). Compressible Multikey and MultiIdentity Fully Homomorphic Encryption. *Journal of Security and Communication Networks*. Retrieved from https://doi.org/10.1155/2021/6619476 on the 23rd of March, 2021

[126]Shiwali, P., & Namita K. (2015). Review on: Providing Arithmetic Operations on RSA using Homomorphic Technique, *International Journal of Computer Science and Information Technologies*, *6(4),* 45 – 54.

[127]Shor, P. W. (1997). Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, SIAM Journal on Computing 26(5) 1277 – 1339.

[128]Shubhangi, A., & Monika, E. (2016). An Efficient Method for Secure Data Storage and Updatingusing Additive Homomorphic Encryption on the Cloud, *An Efficient Method for Secure Data Storage and Updating using Additive Homomorphic Encryption on the Cloud,* 871891.

[129]Smart, N., & Vercauteren, F. (2010). Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes. *In: Public Key Cryptography - Proceedings of the 13thInternational Conference on Practice and Theory in Public Key Cryptography (PKC'10), Lecture Notes in Computer Science (LNCS)*, *6056,* 420-443.

[130]Song, X., & Yulin, W. (2017). Homomorphic Cloud Computing Scheme Based on Hybrid Homomorphic Encryption. *3rd IEEE International Conference on Computer and*

[131]Spoorthy, V., & Mamatha, M. (2014) A Survey On Data Storage And Security In Cloud Computing", *International Journal of Computer Science and Mobile Computing*, 3(6).

[132]Stehle, D., & Steinfeld, R. (2010). Faster Fully Homomorphic Encryption. *In: Advances in Cryptology – Proceedings of ASIACRYPT'10, Lecture Notes in Computer Science (LNCS*), *6477,* 377-394.

[133]Steven, D., & Galbraith, M. (2002). Elliptic Curve Paillier schemes, *Journal of Cryptology, 15 (2),* 129 - 138.

[134]Suraj, S., & Amar, R. (2016). Homomorphic Encryption Approach for Cloud Data Security*, International Research Journal of Engineering and Technology,3(9),* 115-123.

[135]Takabi, H., & Joshi, J. (2010). Security And Privacy Challenges In Cloud Computing Environment, Security and Privacy, IEEE, 71-79.

[136]Tai, W., & Chang, Y. (2018). Separable Reversible Data Hiding in Encrypted Signals with Public Key Cryptography. *Symmetry*. 228-238.

[137]Tsujiiy, S., Doiz, H., Fujitay, R., Gotaishiy, M., Tsunoo Y., & Syoujix, T. (2013). Privacy Preserving Data Processing with Collaboration of Homomorphic Cryptosystems, *Workshop on Applied Homomorphic Cryptography*, Japan.

[138]Uma, K., & Thanooj S. (2017). A Survey Of Encryption Techniques And Security Issues In Cloud Computing, *Journal of Chemical and Pharmaceutical Sciences, 10(1),* 187-199.

[139]Vahid, A., & Seyed, R. (2012). Security Threats And Countermeasures In Cloud Computing, *International Journal of Application or Innovation in Engineering & Management*, *1(2),* 657-665.

[140]Vaikuntanathan, V. (2011). Computing Blindfolded: New Developments in Fully Homomorphic Encryption*, In Proceedings of 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, *7(4),* 5 - 16.

[141]Vaishnavi, V., & Vikas, S. (2014). Multiplicative Homomorphism Ensures Confidentiality Of Data, *International Journal of Advances in Science Engineering and Technology*, 2(3).

[142]Wan, Z., Liu, J., & Deng, R.H (2020). A Hierarchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing*. IEEE Transactions on Information Forensics and Security, 7(2),* 743-754.

[143]Wang, S., Agrawal, D., & El Abbadi, A. (2012). Is Homomorphic Encryption the Holy Grail for Database Queries on Encrypted Data? *Technical report, Department of Computer Science*, University of California.

[144]Wu, D. (2015). Fully Homomorphic Encryption: Cryptography's Holy Grail, *The ACM Magazine for Students - Cryptography archive, ACM,* 21 (3), 24 - 29.

[145]Wu, X., & Ming, X. (2010). Research of the Database Encryption Technique Based on Hybrid Cryptography. 2010 International Symposium on Computational Intelligence and Design. Retrieved from http:/doi:10.1109/iscid.2010.105 on the 13th March, 2020.

[146]Xiaoyan, Y., Qilin, W., & Youming, S. (2020). A Homomorphic Encryption and PrivacyProtection Method Based on Blockchain and Edge Computing, *Hindawi Wireless Communications and Mobile Computing*. Retrieved from https://doi.org/10.1155/2020/8832341 on the 10th of February, 2020.

[147] Yan, D., Feng, R., Huang, J., & Yang, F (2012). Host security event track for complex network environments based on the analysis of log. *IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, *(22(10),* 807-811.

[148] Yan, H., & Zhang, X. (2011). Design of an Extended Privacy Homomorphism Algorithm, In *the 2nd IEEE International Conference on Emergency Management and Management Sciences*, *2(4),* 834 - 837.

[149] Yan, Z., Hu, H., Ahn, G-J., Yu, M., & Zhao, H. (2012). Comparison-based encryption for finegrained access control in clouds. In *CODASPY'12 - Proceedings of the 2nd ACM Conference on Data and Application Security and Privacy, 15(2),* 105-116.

[150] Yang, et al (2018). . A secure verifiable ranked choice online voting system based on homomorphic encryption. IEEE Access 2018, 6, 20506–20519. [CrossRef]

[151] Yang, K., & Jia, X (2012). Data storage auditing service in cloud computing: challenges, methods and opportunities. *World Wide Web*, *15(4),* 409–428.

[152] Yang, Y., Lu, H., & Weng, J. (2011). Multi-User Private Keyword Search for Cloud Computing, In *the Third International Conference on Cloud Computing Technology and Science*, *45(9),* 264 - 271.

[153] Yasmina, B., and Rahal R. (2014). Secure Data Storage in the Cloud with Homomorphic Encryption, *Journal of Theoretical and Applied Information Technology*, *82(2),* 205 – 214.

[154] Yi, H., Paulet, R. and Bertino, E. (2014). Homomorphic Encryption and Applications. New York: Springer Cham Heidelberg.

[155] Yin, C., Sun, R., & Xue, S. (2009). A Modified Query Algorithm for Private Data Security Facing Ecommerce, In *Proceeding of Circuits, Communications and Systems. Pacific-Asia*, *88(12),* 585 - 587.

[156] Yin, Z., Luo, B., & Hong, W. (2014). Separable and Error-Free Reversible Data Hiding in Encrypted Image with High Payload. *Science World*, 1 – 8.

[157] Yuan et al (2023). An electronic voting scheme based on homomorphic encryption and decentralization. PeerJ Comput. Sci.(9)e1649. [CrossRef] [PubMed]

[158] Yuan, L., Yun, P., Lize G., Yuan, Z., & Dezhi, A. (2021). Attribute-Based Fully Homomorphic Encryption Scheme from Lattices with Short Ciphertext, *Hindawi Mathematical Problems in Engineering.* Retrieved from: https://doi.org/10.1155/2021/6656764 on the 7th of March 2020.

[159] Yu, S., Wang, C., & Ren, K (2010). Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing. *In INFOCOM 2010 Proceedings IEEE,* San Diego, CA, 1–9.

[160] Yu Zhan et al (2024). Efficient Electronic Voting System Based on Homomorphic Encryption, https://doi.org/10.3390/electronics13020286, retrieved (Accessed 10th April, 2024) Zhang, X. (2011). Reversible Data Hiding in Encrypted Image. *IEEE Signal Process. Letter*, *18(9),* 255 – 258.

[161] Zhang, C., Chunren, L., & Zhong, H. (2005). An Efficient Authentication Scheme with Fault Tolerance for Database Systems. *Third International Conference on Information Technology and Applications,* doi:10.1109/icita.2005.62

[162] Zhang N., & Zhao, W. (2008). Privacy Protection Against Malicious Adversaries in Distributed Information Sharing Systems, In *IEEE Transactions on Knowledge and Data Engineering*, *20(8),* 1028 - 1033.

[163] Zhang, X., Tong, W., Jin, X., & Dai, S. (2014). Research on Fusion Algorithm of Elliptic Curve Cryptography in Advanced Metering Infrastructure Communication. *Seventh International Symposium on Computational Intelligence and Design*. doi:10.1109/iscid.2014.96.

[164] Zhang, X., Qian, Z., Feng, G., & Ren, Y. (2014). Efficient Reversible Data Hiding in Encrypted Images. *J. Vis. Commun. Image Represent*, *25(10),* 322–328.

[165] Zhao, F., Chao, L., & Liu C. (2014). A Cloud Computing Security Solution Based on Fully Homomorphic Encryption. *16th International Conference on Advanced Communication Technology. 15(9),* 412–428.

[166] Zhaoe, M., & Yang, G. (2017). A Privacy-Preserving Parallel and Homomor-phic Encryption Scheme, *Open phys*.

[167] Zhen-Zhong M. (2014). Research of Information Retrieval in the Cloud Computing Environment.

[168] *7th International Conference on Intelligent Computation Technology and Automation. 14(4),* 32 - 41.

[169] Zhu, Y., & Hu, H (2012). Cooperative provable data possession for integrity verification in multi cloud storage. *IEEE Trans Parallel Distributed System, 23(12),* 2231–2244.

[170] Zvika B., & Vinod, V. (2011). Fully Homomorphic Encryption From Ring Lwe And Security For Key Dependent Messages. *In Proceedings of The 31st Annual Conference On Advances In Cryptology*, *CRYPTO'11*, 505 - 524, Berlin, Heidelberg. Springer-Verlag.