**Al-Azhar University-Gaza**

**Deanship of Postgraduate Studies**

**Faculty of Engineering & Information Technology**

**Master of Computing and Information Systems**

Designing an Authentication Framework Using Homomorphic Encryption: A Study to Balance Security and Usability

**By:**

**Walid Fared M. Murad**

**Supervisor**

**Associate Professor**

**Dr. Ahmed Y. Mahmoud**

**A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master in Computing and Information Systems**

**February– 2025**

جامعة الازهر– غزة عمادة الدراسات العليا

كلية الهندسة وتكنولوجيا المعلومات

ماجستير الحوسبة ونظم المعلومات

**تصميم إطار عمل للمصادقة باستخدام التشفير المتجانس: دراسة لتحقيق التوازن بين الأمان وسهولة الاستخدام**

**إعداد الباحث :**

**وليد فريد مراد مراد**

**إشراف:**

**استاذ مشارك**

د. أحمد يحيى محمود

**قدمت هذه الرسالة استكمالاً لمتطلبات الحصول على درجة الماجستير في الحوسبة ونظم المعلومات**

**رجب 1146**

# بسم الله الرحمن الرحيم

(قل هل يستوي الذين يعلمون والذين لا يعلمون انما يتذكرّ أولوا الألباب)

صدق الله العظيم

**Dedication**

to my beloved homeland, a symbol of resilience and defiance in the face of injustice and destruction

To the souls of our righteous martyrs

To our brave prisoners

To my precious mother

To my great father

To my beloved wife

To my sister and brothers

To all best friends

To my supervisor

**Acknowledgment**

As In the shadow of the unjust war on our beloved Gaza Strip, I have reached the end of this long and winding journey, feeling immense gratitude to many individuals who have supported me along the way.

My deepest thanks go to my supervisor, Associate Professor Dr. Ahmad Y. Mahmoud. His unwavering guidance, support, and encouragement were the foundation upon which I built this thesis. I am eternally indebted to him for the time and energy he invested in me and my research.

I would also like to extend my gratitude to the examination committee: Professor Dr. Sami S. Abu Nasser.

To my family and friends, who were my constant source of support and motivation, I am forever grateful. Their love, understanding, and encouragement were the wind beneath my wings.

I am also thankful to Al-Azhar University for providing the necessary resources essential to completing this research.

Finally, I would like to express my gratitude to the universe for the fortunate moments that led me to this path and for the boundless inspiration that guided me throughout.

**ABSTRAT**

In the face of growing concerns about digital security and user convenience, this thesis proposes a robust authentication framework utilizing Homomorphic Encryption. The proposed framework seeks to balance the critical need for high-level security with the demand for user-friendly systems, addressing a significant gap in existing authentication mechanisms.

Homomorphic Encryption enables operations on encrypted data without decryption, offering an innovative approach to secure sensitive information while maintaining functionality. The study begins by analyzing the limitations of traditional authentication methods, focusing on vulnerabilities in security and inefficiencies in usability. It then outlines the theoretical foundations of Homomorphic Encryption and its applicability to authentication systems.

The practical component involves designing and implementing a prototype authentication system based on the proposed framework. Developed using PHP& JavaScript, this system demonstrates the feasibility of integrating Homomorphic Encryption to achieve enhanced security without compromising user experience. Through rigorous testing and evaluation, the framework is validated against key performance metrics, including encryption overhead, system latency, and user accessibility.

This research contributes to the growing field of secure authentication systems by offering a scalable and efficient solution for environments requiring both high security and usability. It concludes with recommendations for future work to extend the framework's capabilities and explore its potential applications in broader contexts.

By bridging the gap between security and usability, this study aims to advance the state of the art in secure digital interactions, laying the groundwork for innovative authentication solutions in an increasingly connected world.

**Keywords:**

Homomorphic Encryption, Authentication Framework, Biometric Authentication, Usability and Security, Secure Authentication, Data Privacy, Encryption Algorithms, Authentication Protocols, Advanced Cryptography, Secure Data, Data Integrity, Security Usability Metrics, Hashing, Hash, Authentication Protocols,

**الملخــــص**

في ظل تزايد المخاوف بشأن الأمن الرقمي وسهولة استخدام الأنظمة، تقترح هذه الرسالة إطار عمل قويًا للمصادقة يعتمد على التشفير المتماثل. يهدف الإطار المقترح إلى تحقيق توازن بين الحاجة الملحّة إلى مستويات عالية من الأمان ومتطلبات الأنظمة سهلة الاستخدام، مما يعالج فجوة كبيرة في الآليات التقليدية للمصادقة.

يسمح التشفير المتماثل بإجراء العمليات على البيانات المشفرة دون فك تشفيرها، مما يقدم نهجًا مبتكرًا لحماية المعلومات الحساسة مع الحفاظ على الوظائف. تبدأ الدراسة بتحليل قيود طرق المصادقة التقليدية، مع التركيز على نقاط الضعف الأمنية وعدم كفاءة الاستخدام. ثم تستعرض الأسس النظرية للتشفير المتماثل وإمكانية تطبيقه في أنظمة المصادقة.

يشمل الجزء العملي تصميم وتنفيذ نموذج أولي لنظام مصادقة قائم على الإطار المقترح. تم تطوير هذا النظام باستخدام لغة البرمجة PHP& JavaScript ، حيث يُظهر إمكانية دمج التشفير المتماثل لتحقيق أمان معزز دون التضحية بتجربة المستخدم. من خلال الاختبارات والتقييمات المكثفة، تم التحقق من كفاءة الإطار بناءً على معايير أداء رئيسية، مثل عبء التشفير، وزمن استجابة النظام، وسهولة الوصول للمستخدم.

تساهم هذه الدراسة في مجال أنظمة المصادقة الآمنة من خلال تقديم حل قابل للتوسع وفعّال للبيئات التي تتطلب أمانًا عاليًا وسهولة استخدام في الوقت نفسه. وتختتم بتوصيات للعمل المستقبلي لتوسيع قدرات الإطار واستكشاف تطبيقاته المحتملة في سياقات أوسع.

من خلال سد الفجوة بين الأمان وسهولة الاستخدام، تهدف هذه الدراسة إلى تعزيز التفاعلات الرقمية الآمنة، مما يمهد الطريق لحلول مصادقة مبتكرة في عالم متصل بشكل متزايد

**الكلمات المفتاحية:**

التشفير المتماثل ، إطار عمل المصادقة ، سهولة الاستخدام والأمان، المصادقة الآمنة ، خصوصية البيانات، خوارزميات التشفير ، بروتوكولات المصادقة، التشفير المتقدم ، امن البيانات ، سلامة البيانات ، مقاييس سهولة استخدام الأمان ، التجزئة، بروتوكولات المصادقة .

**Chapter 1**

**Introduction**

**INTRODUCTION**

**1.1 Introduction**

Authentication is a critical process in ensuring the identity of users and devices particularly in applications like online banking, e-commerce, and access control systems. As cyber threats and identity theft continue to rise, ensuring robust and secure authentication methods has become a pressing necessity. Traditional authentication techniques, such as passwords and tokens, often fall short due to vulnerabilities like phishing, brute force attacks, and credential theft, necessitating the development of more secure and reliable solutions [1].

Homomorphic encryption, a transformative advancement in cryptography, offers the unique capability to perform computations on encrypted data without the need for decryption. This feature has significant implications for authentication systems, particularly in environments that require high security and data privacy. For example, in scenarios such as healthcare systems, homomorphic encryption can authenticate users while preserving the confidentiality of sensitive medical records. Similarly, it can be employed in cloud-based services to verify identities without exposing raw data to the service provider [2].

However, the challenge lies in balancing the strong security features of homomorphic encryption with user convenience and system efficiency. An overly complex authentication process might deter users, while a simplified process could compromise security. This research aims to design a framework for authentication that leverages homomorphic encryption to strike an optimal balance between security and usability [3].

By addressing these challenges, the proposed framework seeks to enhance the security of authentication systems while ensuring they remain user-friendly and efficient, paving the way for broader adoption in sensitive domains.

**1.2 Statement of the Problem**

The increasing reliance on digital systems for sensitive operations, such as online banking, healthcare record management, and secure communications, underscores the critical importance of robust authentication mechanisms. Traditional authentication methods, including password-based systems and even some biometric approaches, are increasingly susceptible to threats such as data breaches, phishing attacks, and impersonation.

To address these challenges, homomorphic encryption emerges as a promising technology capable of enabling secure computations on encrypted data without the need for decryption. While this approach offers a high level of security, its practical implementation in authentication systems faces notable barriers, including computational overhead, scalability issues, and ensuring usability for end-users.

The balance between security and usability is a persistent challenge in the field. Excessive focus on security may lead to overly complex systems that deter users, while prioritizing ease of use can leave systems vulnerable to attacks. This research seeks to bridge this gap by designing and evaluating a framework for authentication that leverages homomorphic encryption to ensure data confidentiality while maintaining a user-friendly experience.

Moreover, the study addresses the pressing need for real-world applicability by focusing on optimizing computational performance and exploring practical use cases in sectors such as healthcare and financial services, where data security is paramount. By tackling these challenges, this research aims to contribute a scalable, secure, and user-centric solution to the evolving demands of authentication systems.

**1.3 Objectives**

The primary objectives of this research are to design and implement a framework for authentication that utilizes homomorphic encryption, balancing security and usability. The study aims to improve performance while protecting sensitive data from potential threats. Additionally, it focuses on ensuring resilience against common cryptographic attacks and analyzing the usability and efficiency of the proposed solution in various scenarios. By achieving these objectives, the research seeks to contribute a practical and secure authentication method suitable for real-world applications.

**1.4 Significance of the Thesis**

The importance of this thesis stems from its contribution to tackling one of the most serious concerns in the current digital era: guaranteeing safe yet user-friendly authentication techniques. With the exponential rise of sensitive data processed and stored across cloud platforms, healthcare systems, and financial institutions, standard authentication techniques have proven ineffective in combating sophisticated assaults. This study provides a unique authentication system based on homomorphic encryption that allows calculations on encrypted material without needing decryption.

This capability is especially important in contexts where data privacy is critical. For example, the healthcare industry requires strong systems that not only protect patient records but also enable secure authentication while maintaining data security. Similarly, cloud-based services require authentication solutions that ensure data integrity and privacy while being accessible and efficient for consumers.

This thesis seeks to enhance the field of authentication systems by emphasizing the difficult balance between security and usability. It offers a scalable solution to the inherent trade-offs between severe security needs and convenience of use, resulting in greater adoption across several domains. Furthermore, the thesis emphasizes the rising significance of homomorphic encryption as a transformational technology, shining light on its practical application in authentication systems.

The findings of this study are intended to aid companies, academics, and governments looking to improve data security while maintaining inclusiveness and efficiency in authentication procedures

**1.5 Thesis Structure**

The Master's thesis will consist of six main chapters, providing a comprehensive exploration of the research topic. Chapter 1 introduces the study by outlining the objectives, motivations, and significance of developing a framework for authentication using homomorphic encryption, with a focus on balancing security and usability. Chapter 2 establishes the theoretical foundation by presenting essential concepts related to authentication, homomorphic encryption, and their applications, serving as a robust knowledge base for the research. Chapter 3 reviews related literature, analyzing prior studies and identifying research gaps that this study aims to address. Chapter 4 Design of the Integrated Authentication Framework, including the design of the proposed framework, the tools utilized, and the mechanisms for encryption and verification. Chapter 5 presents and analyzes the experimental results, evaluating the performance of the proposed model against key criteria and demonstrating its ability to balance security and usability. Finally, Chapter 6 concludes the thesis by summarizing the findings, discussing the scientific contributions, and providing recommendations for future work to enhance the proposed framework and broaden its application

**Chapter 2**

**Theoretical Foundation**

**Introduction**

Secure and effective authentication forms the cornerstone of any reliable information system, and establishing a solid theoretical foundation is an urgent necessity to achieve a practical balance between security and usability Chapter 2 establishes the essential theoretical foundation for our homomorphic‑encryption‑based authentication framework, beginning with a concise survey of existing authentication methods—from passwords and smart cards to biometric systems—and their inherent trade‑offs between security assurances and user burden. We then introduce homomorphic encryption (HE), detailing its core principle of computing on encrypted data, distinguishing PHE, SHE, and FHE variants, and reviewing prominent schemes such as BFV, CKKS, and TFHE, along with their performance constraints. Building on this cryptographic backdrop, we examine real‑world HE applications in privacy‑preserving data aggregation, secure cloud services, and prior HE‑driven authentication efforts, identifying efficiency and usability gaps. Finally, we define the dual metrics—rigorous security models and standardized usability scales—that will guide our evaluation. Together, these elements unite advanced cryptographic innovation with human‑centered design, paving the way for an authentication solution that achieves both robust security and seamless user experience.

**2.1. Overview of Authentication Systems**

Authentication systems form the bedrock of contemporary information security, governing access to sensitive resources across a multitude of digital environments. Over time, these systems have diversified from simple, knowledge‑based schemes—such as passwords and PINs—to more advanced possession‑ and inherence‑based mechanisms, including hardware tokens and biometric identifiers. Each class of authentication offers a distinct balance of security strength, usability, and implementation complexity. In the following subsections, we categorize and examine the principal authentication paradigms, assess their operational workflows and threat mitigations, and highlight the inherent trade‑offs that inform the design of robust yet user‑centric access control frameworks.[4]

**2.1.1. Knowledge-Based Authentication (Passwords, PINs)**

Knowledge-Based Authentication (KBA) represents a foundational class of authentication mechanisms that rely on the user's ability to recall or recognize pre-established secret information. Typical implementations include **passwords**, **Personal Identification Numbers (PINs)**, and **answers to security questions**. These systems are rooted in the principle of “something the user knows,” and have long been the dominant paradigm in digital identity verification due to their simplicity and ease of deployment.[4]

* From a **security perspective**, however, KBA exhibits well-documented limitations. Repeated empirical studies have shown that user-generated passwords often lack sufficient entropy, making them vulnerable to **dictionary attacks**, **brute-force attempts**, and increasingly sophisticated **phishing schemes**. Furthermore, users tend to reuse credentials across multiple platforms, compounding the risk of **credential stuffing** and **cross-site compromise**.[5]
* From a **usability standpoint**, KBA also introduces significant friction. Users frequently struggle to remember complex passwords, especially when required to change them periodically or conform to stringent formatting rules. This usability burden often leads to insecure workarounds—such as writing passwords down, using predictable sequences, or relying on browser-based password managers—each introducing new attack surfaces.
* Despite the proliferation of password policies and best practices (e.g., NIST SP 800-63B guidelines), research indicates that such measures often have **limited effectiveness** in mitigating KBA vulnerabilities. Consequently, KBA is increasingly viewed as insufficient when used in isolation, particularly in high-security or privacy-sensitive applications. Its role is shifting toward **complementary use in multi-factor authentication (MFA)** frameworks, where it is combined with possession or biometric factors to enhance overall assurance.
* Several scholarly efforts have modeled the **risk profile of KBA** using formal threat models. For example, [4] introduced an empirical framework to evaluate web authentication schemes along axes such as **resilience to exposure**, **server leakage resistance**, and **recoverability**, where KBA scored poorly in most categories. These findings reinforce the consensus that while KBA remains ubiquitous, its effectiveness is constrained by human factors and structural vulnerabilities.[4][6]

**2.1.2. Possession-Based Authentication (cards, .....)**

Possession-Based Authentication (PBA) is founded on the principle of verifying identity through something the user has. This category includes a broad range of physical and digital tokens such as smart cards, hardware tokens, mobile-based authenticator apps, USB security keys (e.g., YubiKey), and One-Time Password (OTP) devices. In contemporary systems, PBA is widely adopted as a core component in multi-factor authentication (MFA) architectures, serving as a safeguard against credential-based attacks that compromise knowledge factors alone.

From a security standpoint, PBA offers improved resilience against remote attacks, particularly credential phishing, keylogging, and brute-force attempts. Since authentication is tied to the physical or digital possession of a token, an attacker must obtain or intercept the device itself—posing a higher barrier compared to password-based systems. Advanced forms of PBA, such as public key cryptography–based hardware tokens (e.g., FIDO2/WebAuthn), ensure cryptographic proof of possession without exposing secrets during transmission, reducing susceptibility to man-in-the-middle (MITM) and replay attacks.[7][8]

However, possession-based methods are not without limitations. The physical nature of tokens introduces operational risks such as loss, theft, damage, or device incompatibility. Moreover, some systems may rely on software tokens (e.g., mobile OTP apps), which remain vulnerable to malware, device compromise, or SIM swapping attacks. These weaknesses can erode the assumed security benefits unless combined with other factors (e.g., biometric verification or contextual checks).[7]

From a usability and deployment perspective, PBA presents both advantages and challenges. While hardware tokens can offer seamless tap-based authentication (e.g., NFC cards), users may find carrying and managing physical devices cumbersome. Software-based possession (e.g., Google Authenticator, Authy) improves convenience but depends on smartphone access and technical literacy. Additionally, recovery mechanisms for lost tokens can introduce new vulnerabilities or onboarding friction, requiring careful system design.

Recent studies have proposed context-aware possession authentication, in which device usage is validated alongside metadata such as geolocation, device fingerprinting, or behavioral signatures. This direction seeks to enhance security while maintaining usability, especially in mobile-first environments.

PBA strengthens authentication by anchoring identity verification to a physical or digital object. While more robust than knowledge-based approaches in isolation, its effectiveness hinges on secure token management, user behavior, and system architecture—making it most effective as a complementary rather than standalone solution.[8][4]

**2.1.3. Inherence-Based Authentication (Biometrics)**

**I**nherence-based authentication refers to verification mechanisms that rely on an individual’s unique and intrinsic characteristics, which can be physiological (e.g., fingerprints, iris, facial features) or behavioral (e.g., voice patterns, keystroke dynamics, gait). [9]These biometric traits are largely immutable, inherently tied to the user, and therefore difficult to replicate or share, making them a highly attractive component in modern authentication systems[10].

1. **Scientific Foundations**

The core premise of biometric authentication lies in pattern recognition and feature extraction. Each modality follows a structured pipeline: data acquisition → pre-processing → feature extraction → template generation → matching. Advances in signal processing and machine learning have significantly improved the accuracy and robustness of biometric systems. For example, convolutional neural networks (CNNs) are now widely used in facial recognition and fingerprint classification, achieving false match rates (FMR) below 0.001% under controlled conditions .[9]

However, biometric authentication is not without inherent limitations. Unlike knowledge-based or possession-based credentials, **biometric identifiers cannot be revoked or changed** once compromised. This permanence introduces a significant privacy risk in the event of data breaches. Moreover, biometric systems must contend with intra-user variability (e.g., changes in voice due to illness or variations in fingerprint quality due to skin conditions), which can negatively affect system performance.

To address these challenges, research has focused on **template protection mechanisms**, such as cancelable biometrics and biometric cryptosystems. These techniques aim to ensure that biometric templates, if compromised, cannot be reverse-engineered to reconstruct the original biometric signal, while still allowing accurate matching.

1. **Security and Usability Considerations**

From a security perspective, biometric systems provide high resistance to impersonation and phishing, as the factor is inherently bound to the individual. However, spoofing attacks—such as using 3D facial masks or synthetic fingerprints—remain viable threats, particularly in systems lacking liveness detection mechanisms.[9]

In terms of usability, biometrics are generally perceived as user-friendly due to their non-reliance on memory or physical tokens. Nonetheless, practical deployments reveal challenges: environmental conditions (e.g., lighting for facial recognition, noise for voice recognition), sensor quality, and user behavior all impact performance and user acceptance.

Implementation architectures vary between **on-device (local) matching**, which enhances privacy by keeping biometric templates on the user’s device, and **remote (server-side) matching**, which facilitates centralized control but introduces concerns about **template storage, transmission security, and compliance with privacy regulations**. To address these challenges, modern systems increasingly integrate **privacy-preserving cryptographic techniques**, including **homomorphic encryption**, **secure multi-party computation**, and **differential privacy**, enabling biometric verification without exposing raw templates..[9]

**2.1.4. Existing Frameworks and Their Security/Usability Trade-offs**

Modern authentication frameworks are increasingly adopting multi-factor authentication (MFA) models that integrate two or more authentication factors—typically combining knowledge (e.g., passwords), possession (e.g., tokens or devices), and inherence (e.g., biometrics). While MFA offers substantial improvements in resilience against various attack vectors, such as credential stuffing, phishing, and brute-force attacks, it also introduces **new complexities and usability challenges** that must be carefully balanced.[10][11]

1. **Comparative Analysis of Frameworks:**

Several frameworks have emerged to operationalize MFA in both enterprise and consumer contexts. These include:

* **FIDO2/WebAuthn**, which emphasizes passwordless authentication through public-key cryptography and platform authenticators.[12]
* **OAuth 2.0 with MFA extensions**, widely used in federated identity management systems.[13]
* **Risk-based authentication (RBA)** frameworks, which dynamically adjust authentication requirements based on contextual risk signals.[14]

**Each framework attempts to balance three critical dimensions**:

1. **Security Robustness**:  
   The use of independent, heterogeneous factors significantly increases resistance to credential compromise. For example, OTP-based possession factors mitigate phishing risk, while biometric factors add non-repudiation.[15]
2. **Usability and Accessibility**:  
   As the number of authentication steps increases, so does user friction. Repeated biometric prompts or token requests can hinder task completion, especially in time-sensitive contexts.[10][15]
3. **Operational Cost and Complexity**:  
   The deployment of hardware tokens, biometric readers, and HE infrastructure introduces both capital and maintenance costs, which must be justified by the security gains.[16]
4. **The Security–Usability Trade-Off:**

Research consistently emphasizes the inverse relationship between security and usability. Systems with the highest security—such as three-factor authentication with encrypted biometrics—often impose cognitive and operational burdens on users, reducing adoption and increasing error rates. Conversely, overly simplified systems (e.g., password-only) are more vulnerable to compromise but offer better usability [10]

A study [61] demonstrated that increasing authentication complexity beyond two factors yields diminishing returns in security while significantly lowering user satisfaction and system engagement. Hence, the effectiveness of any authentication framework hinges on its ability to **adapt to context**, providing stronger security when necessary without overwhelming users in routine scenarios.

**2.2. Fundamentals of Homomorphic Encryption (HE)**

The concept of homomorphic encryption (HE) traces back to 1978, when Rivest, Shamir, and Dertouzos introduced the notion of “privacy homomorphisms” in conjunction with RSA, demonstrating that multiplication operations can be performed directly on RSA-encrypted data without decryption [17]. This early insight seeded interest in performing computations on ciphertexts. Over the decades, partially homomorphic schemes—like Paillier for additive operations—emerged, reinforcing the potential of privacy-aware encrypted computation .[18][19]

A seminal moment occurred in 2009, when [20] the first feasible Fully Homomorphic Encryption (FHE) scheme. Built on lattice-based cryptography and leveraging bootstrapping, this breakthrough unlocked the theoretical possibility of unlimited encrypted computation, transforming the cryptographic landscape.

Between 2011 and 2012, second-generation HE schemes such as BGV, BFV, and CKKS were introduced. These adopted the more efficient Ring-Learning With Errors (RLWE) assumption, enabling reduced key sizes, improved noise management, and enhanced performance—making homomorphic operations over integers and real/floating-point data increasingly practical.[55]

At the core, HE relies on the hardness of lattice-based problems, such as (R)LWE, which are considered secure even against quantum attacks. RLWE, in particular, provides a compact algebraic structure that reduces ciphertext expansion and computation overhead compared to non-ring approaches [55].

**To bridge theory and practice, several robust HE libraries have emerged**:

* Microsoft SEAL offers well-optimized, easy-to-use implementations of BFV and CKKS, employing Residue Number System (RNS) techniques and parallel polynomial arithmetic. It supports both C++ and .NET platforms [56].
* PALISADE, evolved from DARPA-funded SIPHER, provides modular support for multiple HE schemes including BGV, BFV, CKKS, and TFHE, with an emphasis on hardware acceleration .[57]
* HElib, originally released by IBM in 2013, focuses on BGV and advanced packing methods, primarily serving academic research needs [.
* TFHE, designed for fast Boolean-circuit evaluation, supports per-gate bootstrapping allowing efficient encrypted control-flow logic.[58]

These libraries—complete with detailed tutorials, developer-friendly APIs, and advanced optimizations like SIMD packing and RNS arithmetic—have matured homomorphic encryption from abstract theory to deployable solutions across privacy-critical sectors such as healthcare, finance, and secure AI.

**2.2.1. Core Concepts: Computing on Encrypted Data**

The central idea behind HE is the preservation of algebraic operations across the encryption boundary. That is:

***(b\*Enc(a=Enc(b)—Enc(a))***

Where \* represents addition or multiplication, and Enc() denotes the encryption function. This feature allows third-party servers to perform computations without accessing the raw data, aligning with the principles of **confidentiality** and **data minimization**.[19][20][21]

HE builds upon several mathematical foundations, primarily lattice-based cryptography and learning with errors (LWE/RLWE) assumptions. The **semantic security** of HE schemes ensures that no information about the plaintext can be inferred from the ciphertext, even during computation.[22]

This capability makes HE suitable for secure authentication frameworks where encrypted credentials or biometric data can be matched without exposure, thus mitigating risks such as **data leakage**, **insider threats**, and **reconstruction attacks**.[19]

**2.2.2. Types of Homomorphic Encryption (PHE, SHE, FHE)**

HE schemes can be broadly categorized based on the types and number of operations they support:

* **Partially Homomorphic Encryption (PHE)**  
  Supports either addition (e.g., Paillier) or multiplication (e.g., RSA) but not both. Suitable for applications where only one operation is needed, such as voting systems or linear statistics.[20][21]
* **Somewhat Homomorphic Encryption (SHE)**  
  Allows a limited number of both addition and multiplication operations, constrained by a “noise budget” that increases with each operation. Examples include early schemes by Gentry (2009) and others.[22]
* **Fully Homomorphic Encryption (FHE)**  
  Supports an arbitrary number of additions and multiplications, allowing for complex functions (e.g., comparison, machine learning) to be executed over encrypted data. While computationally intensive, FHE represents the “gold standard” for privacy-preserving computation.[23][24][25]

.

***Table 2.1: Comparison between PHE, SWHE, and FHE [21][22][23][24][25]***

|  |  |  |  |
| --- | --- | --- | --- |
| **Criteria** | **Partial Homomorphic Encryption (PHE)** | **Somewhat Homomorphic Encryption (SWHE)** | **Fully Homomorphic Encryption (FHE)** |
| Supported Operations | Supports one operation: either addition or multiplication Note: Multiple operations are not allowed without re-encryption | Supports both addition and multiplication Can perform approximately 4–10 operations before bootstrapping is required to reduce noise | Supports unlimited arithmetic operations (addition and multiplication) Typically requires bootstrapping every 50–100 operations to maintain acceptable noise levels |
| Encryption Time (ms) | Approximately 5–10 ms | Approximately 20–50 ms | Approximately 200–1000 ms |
| Decryption Time (ms) | Approximately 5–10 ms | Approximately 20–50 ms | Approximately 200–1000 ms |
| Ciphertext Size | 1–3 times the size of the original data | 3–7 times the size of the original data | 10–20 times the size of the original data |
| Key Size | Approximately 1–2 KB | Approximately 2–5 KB | Approximately 10–30 KB |
| Efficiency & Computational Complexity | High efficiency due to the simplicity of operations Low computational complexity | Moderate efficiency Computational complexity increases moderately due to noise accumulation | Low efficiency because of high computational complexity Relies on advanced mathematical models (e.g., lattice-based problems) |
| Noise Management | Limited noise accumulation; one operation does not significantly impact performance | Moderate noise accumulation requiring monitoring and manageable within specified boundaries before bootstrapping | Significant noise accumulation; regular bootstrapping is necessary to reduce noise and ensure correct decryption |
| Key Management | Conventional and simple key generation and storage | Requires a more complex key management system to ensure secure key handling in light of increased system complexity | Advanced key management techniques are required for secure generation, distribution, and protection due to the system's overall complexity |
| Practical Applications | Electronic voting systems, basic data integrity verification | Blockchain applications, medium-level security systems | Cloud computing, large-scale data analysis, AI applications, and medical applications that require high privacy |
| Hardware Acceleration | Can be accelerated by 2–3 times using multi-core processors or GPUs | Speed improvements of 3–4 times can be achieved with specialized acceleration techniques | Acceleration improvements of up to 3–5 times can be reached using parallel processing and ASICs; however, overall system complexity remains high |

**2.2.3. Key Homomorphic Encryption Schemes (e.g., BFV, CKKS, TFHE)**

**Three HE schemes have gained prominence in real-world applications and academic research:**

* **BFV (Brakerski–Fan–Vercauteren)**  
  Ideal for exact integer arithmetic. Well-suited for operations like secure voting, private queries, and encrypted search.[26]
* **CKKS (Cheon–Kim–Kim–Song)**  
  Enables approximate computations on real or floating-point numbers. Particularly relevant for signal processing, biometric comparison, and privacy-preserving machine learning tasks. Its support for “approximate homomorphism” balances accuracy and efficiency, making it the preferred scheme in many AI and authentication scenarios.[27]
* **TFHE (Fast Fully Homomorphic Encryption over the Torus)**  
  Designed for bit-level operations and Boolean circuits. Supports fast encrypted comparisons, logical gates, and decision trees. Particularly useful for control-flow-heavy logic, such as access policies and conditional checks.[28][29]

Each scheme differs in its underlying algebraic structure, supported operations, noise tolerance, and performance characteristics. The choice of scheme is thus guided by application requirements: **precision**, **latency**, **scalability**, and the **nature of the input data**.

***Table 2.1: Comparison of Core Homomorphic Encryption Schemes: BFV, CKKS, and TFHE [26][27][28]***

|  |  |  |  |
| --- | --- | --- | --- |
| **Criterion** | **BFV** | **CKKS** | **TFHE** |
| **Supported Data Type** | Integer values | Real/Floating-point numbers | Binary data (bits, Boolean values) |
| **Computation Precision** | Fully accurate arithmetic | Approximate arithmetic (with controlled error) | Bit-level exactness |
| **Supported Operations** | Addition, multiplication, and comparison over integers | Approximate addition and multiplication on real numbers | Logical gates, comparisons, and conditional circuits |
| **Noise Growth** | Relatively low, increases with complex operations | Moderate; influenced by precision and computational depth | High per operation; mitigated via efficient bootstrapping |
| **Bootstrapping Requirement** | Optional (computationally intensive but feasible) | Optional; often unnecessary for practical applications | Mandatory after every gate, but optimized for performance |
| **Computation Latency** | Moderate; depends on depth of operations | Low to moderate; efficient in practical workloads | Very low at the bitwise level |
| **Key Size** | Relatively large | Large | Larger still; can be reduced with optimization techniques |
| **Typical Applications** | Secure voting, private queries, encrypted database search | Privacy-preserving machine learning, signal processing, biometric analysis | Access control, decision trees, encrypted condition checking |

**2.2.4. Performance Characteristics and Challenges of HE**

Although homomorphic encryption (HE) offers strong privacy-preserving capabilities, it introduces significant computational overhead due to its reliance on intricate polynomial arithmetic and the necessity of careful noise management. Key performance dimensions include:[30][31][32]

* Computation Time: HE operations—particularly homomorphic multiplications and bootstrapping—are substantially slower than their plaintext counterparts, often by several orders of magnitude.[30][33]
* Ciphertext Expansion: Encrypted data grows significantly in size compared to original plaintext, posing challenges for storage efficiency and data transmission.[33]
* Key Management: HE schemes demand the generation and secure handling of multiple types of cryptographic keys, including public, secret, and evaluation keys.[32]
* Noise Accumulation: Each homomorphic operation introduces additional noise to the ciphertext. If this noise surpasses a given threshold, the ciphertext becomes undecryptable. Fully homomorphic encryption (FHE) schemes address this limitation through bootstrapping—a costly but essential procedure for noise reset.[30][31]

To overcome these inefficiencies, modern innovations like as SIMD-style packing (Single Instruction Multiple Data) have been proposed to enable batch processing while lowering per-operation cost. Furthermore, current HE libraries, like as **Microsoft** **SEAL**, **PALISADE**, and **TenSEAL**, provide solid implementations and developer-friendly APIs that simplify integration into secure authentication systems .[33]

**2.3. Homomorphic Encryption in Security Applications**

Homomorphic Encryption (HE) has emerged as a powerful enabler for secure computation across a range of security-sensitive applications. By allowing computations on encrypted data, HE facilitates the development of systems that uphold data confidentiality, integrity, and user privacy without sacrificing functionality. This capability is particularly significant in contexts where trust boundaries are weak—such as cloud computing, multi-party authentication, and data aggregation across untrusted domains.[34]

**2.3.1. Privacy-Preserving Data Aggregation**

One of the most foundational and impactful applications of homomorphic encryption (HE) lies in the domain of privacy-preserving data aggregation. Traditional data aggregation methods inherently require decryption of individual user data at a centralized aggregator before any meaningful computation can be performed. This practice not only violates the principle of data minimization but also exposes sensitive information to significant risks of unauthorized access, data breaches, and non-compliance with regulatory standards such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA).[34]

HE offers a paradigm-shifting solution by enabling direct aggregation operations over ciphertexts, thereby eliminating the need to reveal individual data points. Partially homomorphic encryption schemes—such as the Paillier cryptosystem, which supports additive operations on encrypted integers—are particularly effective for such use cases. A prominent example is found in smart grid infrastructures, where electricity consumption readings from individual households are encrypted locally and then transmitted to the utility provider. Using HE, the provider can compute the total or average consumption across households without ever accessing or decrypting the individual usage data. This guarantees that user-level information remains confidential throughout the aggregation process while still enabling accurate operational analytics [62].[34]

In more advanced scenarios, modern leveled or approximate HE schemes—such as BFV and CKKS—introduce additional performance enhancements through techniques like Single Instruction Multiple Data (SIMD) batching. This feature allows multiple plaintext values to be packed into a single ciphertext and processed simultaneously, significantly reducing computational overhead and bandwidth requirements. As a result, large-scale sensor networks, smart city infrastructures, and Internet of Things (IoT) ecosystems can benefit from privacy-preserving aggregation mechanisms that are both secure and computationally scalable.[35][36]

From a systems perspective, this application of HE not only preserves user privacy at the edge but also facilitates the implementation of decentralized trust models. It empowers data controllers to comply with legal frameworks while maintaining operational efficiency and unlocking value from encrypted analytics without compromising confidentiality. As such, privacy-preserving data aggregation stands as a critical pillar in the advancement of secure-by-design architectures across modern digital ecosystems.

**2.3.2. Secure Cloud Computing**

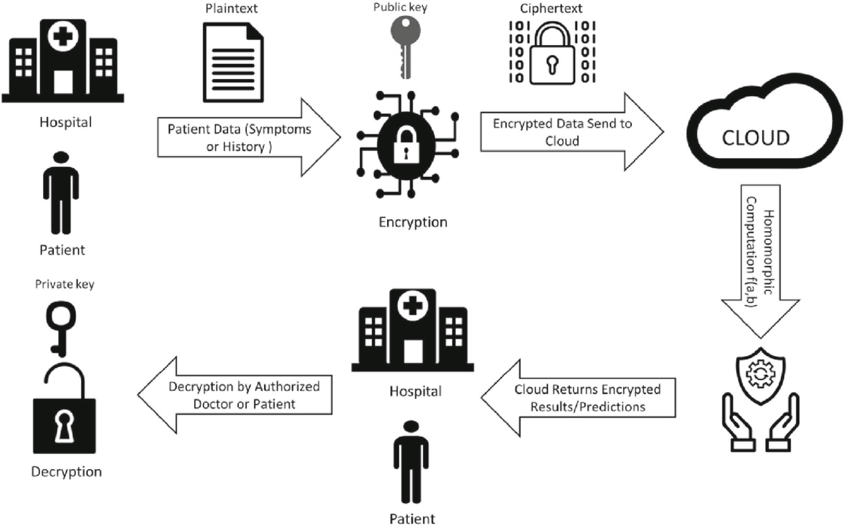
Cloud environments are inherently characterized by a lack of complete trust, as data processing takes place on infrastructure managed by third-party providers rather than data owners themselves. Within this context, homomorphic encryption (HE) introduces a transformative paradigm by enabling computation on encrypted data without requiring decryption, thereby ensuring that sensitive information remains confidential throughout its lifecycle. This cryptographic capability permits the secure outsourcing of computation while preserving data privacy, integrity, and compliance with regulatory frameworks.[34][37]

**Several practical applications exemplify the power of HE in trustless cloud scenarios:**

* Encrypted Search: HE allows users to perform keyword searches over encrypted data residing in cloud storage—typically implemented through HE-compatible inverted indices—without disclosing the content of the query or the underlying documents.
* Privacy-Preserving Machine Learning: By leveraging schemes such as CKKS, machine learning models can be trained and evaluated on encrypted datasets, facilitating secure inference and federated learning across distributed entities (e.g., hospitals or financial institutions) without compromising data confidentiality.
* Confidential Access Control: Using TFHE, access control policies and authorization rules can be represented and evaluated directly in encrypted form. This enables secure policy enforcement without revealing access logic, user roles, or permissions.[37]

**Through these mechanisms, HE empowers the development of “computation-as-a-service” paradigms in which cloud servers remain computationally functional yet semantically unaware of the data they handle. This not only enhances security but also strengthens legal accountability for data processors under stringent data protection regulations.**

***Figure 2.1: Encrypted Authentication Workflow in Cloud Environment [38]***

****

**2.3.3. Previous Applications of HE in Authentication**

Homomorphic encryption (HE) has demonstrated notable potential in reshaping the design of secure authentication protocols by enabling cryptographic computation on encrypted inputs without prior decryption. Traditional authentication systems typically rely on transmitting or exposing sensitive user credentials—such as passwords, biometric templates, or token responses—to centralized verification servers. This process creates inherent vulnerabilities, including the risk of interception, unauthorized access, or compromise due to server-side breaches. In contrast, HE introduces a privacy-preserving paradigm wherein authentication decisions are derived exclusively from encrypted data, thereby significantly minimizing the attack surface and limiting exposure of personal information during the verification process.

**Several academic implementations have validated the feasibility and effectiveness of HE in various authentication scenarios:**

* **Biometric Matching on Encrypted Templates:** Bringer et al. (2009) demonstrated one of the pioneering implementations of HE in biometric authentication, using Paillier encryption to compare encrypted iris templates without decrypting them. Subsequent research leveraging more advanced schemes such as CKKS and TFHE has led to improved accuracy and reduced computational latency, enabling efficient matching of complex biometric vectors like fingerprints and facial embeddings in an encrypted domain.[39]
* **Encrypted Challenge-Response Protocols:** Homomorphic operations have been integrated into interactive authentication frameworks wherein both the generation of challenges and the validation of responses are conducted on encrypted data. Such designs ensure end-to-end confidentiality and resilience against replay attacks, even in hostile environments.[40]
* **HE-Enabled Password Verification**: Gasti et al. (2012) introduced a system in which hashed user credentials remain encrypted throughout the verification lifecycle. This eliminates any exposure of the hash values—even in the event of a compromised verification server—thus offering stronger protection against offline dictionary attacks.[41]

While these implementations demonstrate the security-enhancing capabilities of HE, they are accompanied by operational trade-offs that warrant consideration:

* **Increased Computational Latency**: The added cryptographic complexity associated with encryption, homomorphic evaluation, and decryption introduces delays that are non-trivial in time-sensitive applications. Practical adoption therefore requires optimizations such as parallel computation, algorithmic tuning, or hardware acceleration (e.g., FPGAs or GPUs) to achieve real-time performance.[42][43][44]
* **Complexity in Key Management**: HE schemes often necessitate elaborate key structures, including public/secret keys, evaluation keys, relinearization keys, and bootstrapping parameters. Secure distribution and rotation of these cryptographic materials introduce additional system overhead and governance challenges.[45]

Despite these constraints, the application of HE in authentication remains highly attractive in high-assurance environments—such as remote biometric verification, access control to critical infrastructure, and federated identity frameworks spanning organizational or national boundaries—where the preservation of confidentiality is paramount.

**2.4. Defining and Measuring Security and Usability**

In authentication system design, success hinges on striking the right security–usability balance. Authentication that is overly complex may thwart threats, but often frustrates users, reducing adoption and encouraging insecure workarounds. Conversely, prioritizing convenience alone risks vulnerability. As such, a nuanced, adaptive design is imperative.

Recent studies spotlight **Risk‑Based Authentication (RBA)** as a compelling solution. [46] demonstrate that RBA dynamically adjusts friction—e.g., requiring a one-time code only for unusual logins—thereby delivering comparable security to 2FA with significantly improved usability and user satisfaction [46][47]. However, their work also highlights that misconfiguration of risk thresholds can either lower security or irritate users unnecessarily [47].

Complementary research reveals that **time satisfaction**, not raw speed, heavily influences perceived usability and security in token-based approaches [48]. These insights emphasize that **UX perceptions**—not only system performance metrics—impact acceptance and trust in authentication mechanisms.

Meta-analyses in usable security [59] show a recurring theme: users trade convenience for security only to a point. Systems that appear too cumbersome—such as mandating frequent 2FA or complex password policies—often lead users to reuse passwords or bypass authentication [59]. They conclude that designing for optimal **cognitive load**, memorability, and satisfaction is crucial.

Current standards (NIST, ISO) and frameworks like AuthGuide (2020) incorporate these findings by recommending **adaptive, context-aware authentication** that adjusts based on risk factors such as device consistency or geolocation [60]. Such systems align security requirements to real-time assessments, empowering both protection and usability.

**2.4.1. Security Metrics (e.g., Threat Models, Attack Vectors, Cryptographic Strength)**

Security in authentication refers to the system’s ability to resist attacks, preserve confidentiality, ensure data integrity, and enforce non-repudiation. The evaluation of security in this context requires consideration of multiple metrics and threat models.

**A. Common Threat Models:**

1. Brute-force and dictionary attacks: Attacker tries large sets of possible passwords.
2. Phishing and social engineering: User is tricked into revealing credentials.
3. Credential stuffing: Automated injection of stolen credentials from other systems.
4. Replay and man-in-the-middle (MitM) attacks: Adversary intercepts and reuses authentication data.
5. Biometric spoofing: Presentation of fake biometric samples.

**These threats demand multi-layered defense mechanisms, including encryption, multi-factor authentication, and context-aware validation.[46]**

**B. Quantitative Security Metrics:**

***Table 2.3: Quantitative Security Metrics in Authentication Systems [46][47]***

|  |  |
| --- | --- |
| Metric | Definition |
| False Acceptance Rate (FAR) | Proportion of unauthorized users wrongly accepted as valid. |
| False Rejection Rate (FRR) | Proportion of authorized users wrongly rejected. |
| Equal Error Rate (EER) | Point at which FAR = FRR; lower EER indicates higher accuracy. |
| Resistance to Attack Vectors | Evaluation of system’s resilience against specific attacks (e.g., spoofing). |
| Cryptographic Strength | Bit-length and mathematical robustness of underlying encryption schemes. |

**C. Cryptographic Validation:**

In homomorphic encryption–based authentication systems, cryptographic soundness is critical. For instance, schemes based on lattice problems (e.g., RLWE – Ring Learning With Errors) offer post-quantum resistance, ensuring longevity of security in future computing contexts. The security of the system must therefore be evaluated not only through penetration testing, but also via formal cryptographic proofs and complexity-theoretic analysis.[49][50]

**2.4.2. Usability Metrics (e.g., System Usability Scale (SUS), Task Success Rate, User Satisfaction)**

While robust security is essential, the adoption and effectiveness of an authentication system largely depend on its usability—especially in high-frequency use cases like employee logins or mobile access.

**A. Usability Dimensions**

1. Efficiency: Time and steps required to complete the authentication.
2. Learnability: How easily users can understand and perform the process.
3. Memorability: Ability of users to recall credentials or procedures after non-use.
4. Error Tolerance: System behavior when the user makes mistakes.
5. User Satisfaction: Subjective perception of comfort, trust, and control.

**B. Quantitative and Qualitative Usability Measures:**

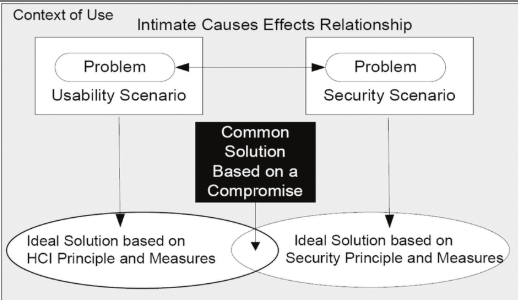
***Table 2.4: Usability Evaluation Metrics for Authentication Systems [49][50]***

|  |  |
| --- | --- |
| Metric | Description |
| System Usability Scale (SUS) | A 10-item Likert scale providing a global view of subjective usability. |
| Task Completion Time | Time taken to complete the authentication process. |
| Task Success Rate | Proportion of successful logins without help or retries. |
| User Error Rate | Frequency of mistakes (e.g., wrong password, bad fingerprint scan). |
| Cognitive Load (NASA-TLX) | Assessment of mental effort required by the authentication process. |

**C. Balancing Security and Usability:**

Studies such as [51] have shown that increasing security steps can weaken the user experience, while simpler systems (like password-only authentication) are easier to use but offer lower security [51]. point out that the mental and physical burden of authentication can lead users to avoid it or create workarounds, introducing additional security risks.[52][53] Therefore, modern systems rely on risk-based dynamic authentication (context-aware authentication) to provide strong security without placing excessive burden on the user.[51][53]

***Figure 2.2: The Security–Usability Trade-Off Curve [54]***

****

**2.5 Summary**

In this chapter, we surveyed the core authentication paradigms—knowledge‑, possession‑, and inherence‑based methods—outlining their security strengths and usability limitations. We then introduced homomorphic encryption (HE), categorized its partially, somewhat, and fully homomorphic schemes, and compared key instantiations (BFV, CKKS, TFHE) in terms of precision, noise management, and performance. Finally, we explored HE’s applications in privacy‑preserving data aggregation, secure cloud computing, and HE‑enabled authentication protocols, and defined quantitative metrics for both security (FAR, FRR, EER, cryptographic strength) and usability (SUS, task time, success and error rates, cognitive load).

In Chapter 3, we will present a comprehensive literature review covering multi‑factor and HE‑enhanced authentication frameworks, analyzing existing implementations, performance evaluations, and open research challenges that inform our proposed framework**.**

**Chapter 3**

**Literature Review**

**Literature Review**

1. **Introduction**:

This chapter presents a comprehensive review of the literature related to authentication and homomorphic encryption. It examines authentication mechanisms, including password-based, biometric, and multi-factor authentication, highlighting their strengths, limitations, and challenges in balancing security and usability.

Additionally, the chapter explores homomorphic encryption, its classifications (partial, semi-homomorphic, and fully homomorphic), and its applications in secure data processing. The integration of homomorphic encryption with authentication systems is also analyzed, identifying research gaps and technical challenges.

The literature review will also serve as a foundation for the development and evaluation of model the proposed authentication security while maintaining usability Additionally, this chapter will provide an overview of the concepts and theories that form the foundation of this research and will be used throughout the thesis to provide a theoretical and empirical foundation for the proposed authentication framework.

1. **Related Work**

This survey [64] explores the evolving landscape of authentication mechanisms, particularly in the context of Multi-Factor Authentication (MFA) as a response to the limitations of Single-Factor Authentication (SFA) in internet-based services. The authors classify MFA techniques into biometric and non-biometric approaches, highlighting a central challenge in achieving a balance between security and accuracy. The survey notes a divergence in research trends: while biometric-focused studies emphasize improving recognition accuracy, non-biometric studies often concentrate on layering multiple authentication factors to enhance security.   
The paper contributes a detailed comparative analysis of authentication protocols, security requirements, and secure one-time passcode generation and distribution mechanisms. Additionally, it provides a comprehensive review of cancelable biometrics—including their techniques, vulnerabilities, and design requirements. Concluding with a summary of key challenges and future research directions, the study serves as a valuable reference for researchers aiming to develop secure and user-friendly authentication systems.

To improve knowledge of the real-world ramifications of these difficulties., the paper [65] addresses the emerging safety and security challenges associated with authenticating human operators of safety-critical devices, particularly in the medical field. It emphasizes that while authentication mechanisms (e.g., passwords, fingerprints) aim to enhance security, they may inadvertently introduce new risks such as reduced usability, delays, and unsafe user behaviors.  
The authors propose a holistic design approach that balances security, safety, usability, and system performance. Using a medical case study, the research conducts both qualitative and probabilistic trade-off analyses to evaluate how different authentication methods and their specific implementation parameters affect overall risk.  
Notably, the probabilistic model quantifies risk as a function of both accidental and malicious causes, allowing designers to optimize authentication mechanisms based on environmental factors and anticipated attack types. The study highlights the complex interdependence between system attributes and human behavior, offering a structured methodology for guiding the design of secure, usable, and efficient authentication solutions.

Building on the recognition of dynamic user environments, the paper [66] proposes an advanced Identity and Access Management (IAM) framework designed for financial institutions across public and private sectors, emphasizing robust data protection and adaptive user authentication. The system integrates a hash-based data security algorithm with a Cyber Water Swarm Optimization-based IAM mechanism, enhanced by a Deep Hill Prophet learning strategy. The core innovation lies in the dynamic calculation of a trust score for each user, based on behavioral and contextual factors. This score governs access permissions to sensitive financial data.  
Users with high trust scores are issued secret encryption keys to access encrypted data, maintaining security even in case of attempted unauthorized access. The framework was validated using a real-time financial transaction dataset simulated in the MATLAB environment, demonstrating its ability to manage live data streams while ensuring secure identity management. Results confirm that combining advanced hashing and AI-driven optimization techniques within IAM systems provides a scalable, adaptable, and highly secure solution for modern financial cybersecurity challenges.

To complement these approaches and provide a forward-looking perspective, the paper [67] offers a comprehensive survey of continuous authentication, which seeks to provide passive and seamless user verification by leveraging various sensor data, such as biometric, behavioral, and context-based characteristics. While this approach enhances user convenience and security, it introduces significant privacy concerns because of the sensitive nature of the transmitted personal data, which often lies beyond the user's direct control.  
The paper identifies key security challenges, including poor matching rates that can undermine the system's effectiveness by inaccurately verifying user identity, and susceptibility to replay attacks, where attackers intercept and reuse valid authentication data to gain unauthorized access. To address these issues, the study examines different continuous authentication methods, discussing their associated security, privacy, and usability concerns, and compares privacy-preserving strategies to mitigate these risks. It provides recommendations for designing systems that ensure security and privacy without compromising user convenience, such as implementing robust matching algorithms to enhance accuracy, using encryption and anonymization to protect personal data, and creating systems that prioritize both user experience and high-security standards.  
Ultimately, the survey emphasizes the importance of striking a balance between security, privacy, and usability for the successful deployment of continuous authentication systems.

This study [68] presents a cloud-based adaptive multi-factor multi-layer authentication framework designed to strengthen cloud security by preventing unauthorized access and data breaches. The framework incorporates access control and intrusion detection mechanisms, along with an automated selection of authentication methods based on context, ensuring a low false positive rate while maintaining effective user verification. It employs multiple authentication factors, including factor length, validity, and value, as well as user geolocation and browser confirmation techniques. Additionally, the system integrates AES-based encryption to safeguard user login data and directory information from unauthorized disclosure. A key aspect of the framework is its dynamic adaptability, adjusting authentication strength according to contextual risk levels to provide strong identity verification without compromising usability. The proposed system demonstrates exceptional performance in detecting and blocking malicious users and intruders, significantly reducing the likelihood of intentional attacks on cloud data and services.

Building upon this foundation, the paper [69] introduces a decentralized authentication architecture that eliminates the vulnerabilities of centralized systems. By leveraging artificial intelligence (AI) and homomorphic encryption (HE), the system performs identity verification directly on encrypted data, greatly reducing privacy risks. While the cloud-based model focuses on dynamic contextual adaptation, this AI-driven solution enhances privacy and decision-making intelligence, especially in sensitive domains like healthcare and finance.

Expanding on the use of HE in biometric systems, the study [70] presents THRIVE—a privacy-preserving biometric authentication framework using threshold homomorphic encryption. By splitting the decryption key between the user and verifier, the system ensures that no single party can compromise stored biometric templates. THRIVE complements the decentralized AI-HE model by applying homomorphic techniques specifically to biometric modalities while integrating two-factor authentication for enhanced protection under adversarial conditions.

The paper [71] further strengthens this direction by introducing a quantum-resilient biometric authentication scheme based on the McEliece post-quantum encryption algorithm. Unlike most existing HE solutions, this approach adheres to NIST cryptographic standards and supports device-independent secure authentication. By utilizing Face Net and feature transformation for encrypted biometric matching, the system ensures irreversibility, revocability, and unlink ability while significantly reducing storage and communication overhead—achieving over 90.5% compression compared to similar methods.

Recent research has increasingly focused on Fully Homomorphic Encryption (FHE) due to its powerful capability to perform computations on encrypted data without decryption. This foundational concept was initially outlined in a study that categorizes homomorphic encryption into PHE, SHE, and FHE, examining their cryptographic bases and practical challenges. This study laid the groundwork for understanding how homomorphic encryption schemes contribute to secure data operations in modern digital environments.[72]

Building on these theoretical foundations [73], another comprehensive survey delved into the application of HE specifically in healthcare settings—an area with heightened privacy demands. The study compared various HE schemes (PHE, SHE, FHE, FLHE) and assessed their roles in protecting sensitive medical data, such as EHRs, genomic sequences, and imaging, while also addressing attack vectors like side-channel and lattice-based attacks. The research emphasized HE’s practical utility in real-world, privacy-critical environments.

Moving from healthcare to broader cryptographic innovation, [74] one paper introduced a leveled FHE scheme grounded solely on the Learning With Errors (LWE) assumption. This work contributed significantly by eliminating ring-based assumptions, employing a novel relinearization technique and dimension-modulus reduction method to reduce ciphertext size and enhance decryption efficiency. An application in Private Information Retrieval (PIR) was proposed, demonstrating the scheme’s practicality and scalability for secure data retrieval.

To assess how these schemes perform in actual computing environments, another study tested FHE using the Microsoft SEAL library on both traditional PCs and edge devices like the NVIDIA Jetson Nano. The results revealed a trade-off between security and performance—while PCs handled FHE reasonably well, resource-constrained IoT platforms experienced computational bottlenecks. This highlighted a critical gap between theoretical feasibility and real-time applicability, especially in edge computing contexts.[75]

In response to the need for both data privacy and authenticity during encrypted computation, a pivotal work [76] proposed the first fully homomorphic authenticated encryption scheme. This system merged FHE with homomorphic authenticators to ensure verifiable and private computation results. By introducing a new security model, the study addressed previous shortcomings in combining encryption with authenticity. Its multi-dataset support and amortized efficiency mark a significant leap toward trustworthy data computation in sensitive domains.

Acknowledging FHE’s inherent performance limitations, [77] a recent survey analyzed acceleration techniques developed between 2019 and 2022. These techniques were divided into algorithmic-level and hardware-level optimizations. The study provided a much-needed systematic comparison and proposed evaluation metrics, offering a roadmap for improving the practicality of FHE in real-world systems.

The study [78] introduces Cipher Face, a homomorphic encryption-based framework for secure facial recognition in cloud environments. By enabling encrypted Euclidean and Cosine distance calculations, Cipher Face effectively mitigates privacy risks associated with conventional biometric embeddings and demonstrates practical scalability and security benefits.

The contribution [79], Blind-Touch, proposes a hybrid neural network-based fingerprint authentication system that divides processing between client-side plaintext computation and server-side encrypted operations using HE. It further incorporates a homomorphic-compatible data compression technique to support scalable, real-time authentication for large user bases while maintaining data confidentiality.

The study [80] explores a deep learning model using a Convolutional Neural Network (CNN) trained on encrypted biometric data via the CKKS encryption scheme, implemented with Ten SEAL. This approach performs all fingerprint matching operations entirely in the encrypted domain, achieving high accuracy and adherence to biometric protection standards such as ISO/IEC 24745, despite the encryption overhead.

The study presents a comprehensive framework for secure image processing that integrates homomorphic encryption and RSA to protect embedded sensitive data during machine learning tasks. The system leverages logistic regression, feature extraction, and polynomial approximation, demonstrating the viability of homomorphic encryption in secure visual data applications.[81]

The study [82] proposes a dual-layered security framework that synergistically combines blockchain technology with homomorphic encryption to improve both traceability and data confidentiality in supply chain operations. By using blockchain as an immutable and decentralized ledger, the system ensures transparent verification of product authenticity, while homomorphic encryption facilitates secure data analytics without exposing sensitive information. This dual approach not only mitigates data breaches but also aligns with data protection regulations, positioning the model as a resilient solution for global supply chains.

The study [83] builds on this vision by addressing collaborative healthcare data analysis, introducing a framework that integrates secret-sharing, secure multiparty computation (SMPC), and homomorphic encryption within a Hyperledger Fabric-based blockchain. This model allows stakeholders to perform statistical computations on encrypted health data without revealing individual records. Smart contracts encrypted using homomorphic techniques support real-time, privacy-preserving analytics, demonstrating the feasibility and scalability of such solutions in sensitive domains like healthcare research and policy-making.

Expanding the application of privacy-preserving mechanisms to access control in Cyber-Physical Systems (CPSs), study [84] investigates the limitations of Attribute-Based Access Control (ABAC) in protecting sensitive user attributes during authentication. To address these privacy gaps, the authors introduce a zero-knowledge proof (ZKP)-based enhancement that enables the verification of user attributes without exposing the underlying data. The proposed model is particularly suited for high-stakes environments, such as industrial mining sites, where both safety and data confidentiality are critical.

The study [85] introduces two novel authentication protocols that leverage homomorphic encryption and oblivious transfer to safeguard sensitive user information throughout the authentication lifecycle. By focusing on behavioral biometrics—such as swipe gestures and keystroke dynamics—the protocols achieve high authentication accuracy and low computational overhead, confirming their practical viability for real-time deployment. This work establishes a foundational approach to privacy-preserving continuous authentication, particularly suitable for mobile and ubiquitous computing environments.

Complementing this contribution, the study [86] provides a comprehensive survey of the application of homomorphic encryption in securing biometric systems. It categorizes existing techniques based on biometric modalities (e.g., facial recognition, iris scans, fingerprints) and outlines challenges such as computational complexity and scalability. Moreover, the survey highlights the integration of homomorphic encryption with machine learning and blockchain to enhance system accuracy and decentralization. This holistic perspective reinforces the relevance of homomorphic encryption as a cornerstone in the development of resilient, secure biometric systems.

Expanding the discussion beyond purely biometric contexts, the study presents a robust cloud-based security framework integrating multiple encryption methodologies for the protection of sensitive healthcare data. Although it does not directly apply homomorphic encryption, the proposed system combines key aggregate cryptosystems, enhanced Diffie-Hellman key exchange, and hybrid cryptographic algorithms to ensure end-to-end data confidentiality and secure transmission. This work contributes to the practical realization of secure cloud authentication systems and highlights the potential of hybrid cryptographic strategies in sectors where privacy and efficiency are paramount.[87]

**Table 3.1 Literature review Summary**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Application / Contribution** | **Key Findings** | **Methodology** | **Focus Area** | **Title** |
| **Provides foundational understanding and future research directions for secure and user-friendly authentication systems** | Biometric methods aim to improve accuracy; non-biometric focus on multi-layered security; identified key challenges and gaps in usability vs. security | Comparative survey and classification of MFA techniques; analysis of security protocols and cancelable biometrics | Multi-Factor Authentication (MFA) and Cancelable Biometrics | Current Multi-factor of Authentication: Approaches,  Requirements, Attacks and Challenges |
| Provides a structured methodology for designing authentication systems that balance risk, usability, and security | Authentication can introduce new risks; probabilistic modeling helps optimize trade-offs between security, safety, and usability | Holistic design approach; qualitative and probabilistic trade-off analysis using medical case study | Authentication in Safety-Critical Systems (e.g., medical devices) | Authentication for Operators of Critical Medical Devices: A Contribution to Analysis of Design Trade-offs |
| Proposes an AI-driven IAM framework for real-time secure identity management in finance | Dynamic trust score enhances secure access control; framework is scalable, adaptable, and secure | Hash-based security algorithm + Cyber Water Swarm Optimization + Deep Hill Prophet learning; tested on simulated financial dataset (MATLAB) | Adaptive Identity and Access Management (IAM) in Financial Sector | A Comprehensive Framework for Cybersecurity in Identity and Access Management Systems |
| Provides design guidelines for secure and user-friendly continuous authentication systems | Identifies privacy risks, matching errors, and replay attacks; recommends robust matching, encryption, and anonymization | Comparative survey of continuous authentication techniques; analysis of security, privacy, and usability concerns | Continuous Authentication and Privacy Protection | Security, Privacy, and Usability in Continuous Authentication: ASurvey |
| Enhances cloud security while maintaining usability through dynamic multi-factor authentication | Achieves low false positive rates, adapts authentication strength to risk levels, and effectively blocks malicious access | Design and implementation of a context-aware multi-layer authentication system with AES encryption and intrusion detection | Adaptive Multi-Factor Authentication in Cloud Environments | Strengthening Cloud Security: An Innovative Multi-Factor  Multi-Layer Authentication Framework for Cloud  User Authentication |
| Strengthens security and privacy for sensitive sectors like finance and healthcare through decentralized, encrypted authentication | Improves confidentiality, reduces attack surfaces, enhances scalability and robustness | Design of an AI-powered authentication system operating on encrypted data without decryption | Decentralized Authentication integrating AI and Homomorphic Encryption | AI-DRIVEN DECENTRALIZED  AUTHENTICATION SYSTEM USING  HOMOMORPHIC ENCRYPTION |
| Enables strong biometric authentication without exposing raw biometric data, suitable for privacy-conscious applications | Protects biometric templates, ensures privacy during authentication, resists malicious adversaries, and remains practically efficient | Design of threshold encryption-based enrollment and authentication protocols splitting the decryption key between user and verifier | Privacy-preserving Biometric Authentication using Homomorphic Encryption | THRIVE: threshold homomorphic encryption  based secure and privacy preserving biometric  verification system |
| Enables real-time, privacy-preserving biometric authentication resilient against future quantum threats | Provides quantum-resistant, device-independent biometric authentication with enhanced privacy (irreversibility, revocability, unlink ability) and reduced storage/communication overhead | Utilization of McEliece-based homomorphic encryption with feature transformation and Face Net embeddings for biometric matching | Post-Quantum Biometric Authentication using Homomorphic Encryption | Post-Quantum Biometric Authentication Based on  Homomorphic Encryption and Classic McEliece |
| Serves as a foundational reference for implementing HE to enhance security and privacy in information systems | Highlights strengths, limitations, and real-world applicability of different HE types for secure computation | Categorization and analysis of PHE, SHE, and FHE schemes; discussion of cryptographic algorithms and practical challenges | Types and Applications of Homomorphic Encryption (HE) | HOMOMORPHICE  NCRYPTION  :A COMPREHENSIVE  STUDYOFTYPES,  TECHNIQUES, ANDREAL-WORLD  APPLICATIONS |
| Positions HE as a long-term privacy-preserving solution for healthcare data analytics | Highlights HE's role in securing EHRs, genomics, telehealth; identifies potential threats and mitigation strategies | Comparison of PHE, SHE, FHE, and FLHE; analysis of attacks and defenses; application reviews in healthcare | Application of Homomorphic Encryption (HE) for protecting healthcare data | A comprehensive survey on secure healthcare data processing with homomorphic encryption: attacks and defenses |
| Provides an efficient, assumption-light foundation for future FHE systems, demonstrated through PIR protocol | Achieves shorter ciphertexts, lower decryption complexity, efficient encrypted function evaluation | Introduces new relinearization and dimension-modulus reduction techniques; constructs a single-server PIR protocol | Construction of FHE based only on LWE without ring assumptions | EFFICIENT FULLY HOMOMORPHIC ENCRYPTION FROM  (STANDARD) LWE |
| Highlights practical feasibility and limitations of FHE deployment in IoT and edge computing | PCs handle FHE well; edge devices face performance and scalability challenges | Experimental evaluation using Microsoft SEAL on PC and NVIDIA Jetson Nano | Privacy-preserving data processing in IoT and data lakes using FHE | Empirical Study of Fully Homomorphic Encryption Using  Microsoft SEAL |
| Advances secure processing on encrypted data, enabling practical applications needing both privacy and verified computation | Achieves strong privacy and authenticity; efficiently handles multiple datasets | Introduced new unified security model; proposed first construction combining FHE with homomorphic authenticators | Secure computations with both privacy and authenticity using homomorphic encryption | Secure Fully Homomorphic  Authenticated Encryption |
| Supports enhancing FHE practicality, encouraging wider adoption in real-world applications | Provided comparative analysis of methods; identified strengths, weaknesses, and research gaps | Categorized optimization techniques into algorithmic and hardware levels; proposed evaluation metrics | Improving performance of Fully Homomorphic Encryption (FHE) | Practical solutions in fully homomorphic  encryption: a survey analyzing existing  acceleration methods |
| Facilitates privacy-preserving biometric authentication; open-sourced for broader adoption | Enabled encrypted similarity computations; demonstrated scalability, security, and practical viability | Developed Cipher Face framework; conducted experimental validation with different models and embedding sizes | Secure facial recognition using FHE | CIPHERFACE: A FULLY HOMOMORPHIC ENCRYPTION-DRIVEN  FRAMEWORK FOR SECURE CLOUD-BASED FACIAL  RECOGNITION |
| Provides scalable, accurate, and privacy-focused fingerprint authentication for online and cloud services | Achieved 93.6% F1-score (PolyU) and 98.2% (SOKOTO); matches 5,000 fingerprints in 0.65 seconds | Client-side neural network preprocessing + server-side homomorphic encrypted computation; homomorphic data compression; clustered server architecture | Privacy-preserving fingerprint authentication in web/cloud environments | Blind-Touch: Homomorphic Encryption-Based Distributed Neural Network  Inference for Privacy-Preserving Fingerprint Authentication |
| Real-time privacy-preserving fingerprint authentication for cloud-based applications in sensitive sectors | Achieved 99.06% test accuracy, 99.19% TAR, 0% FAR, 0.40% EER; processing per fingerprint 0.136 s | CNN feature extraction with CKKS FHE (Ten SEAL); encrypted classification and similarity checks (Euclidean distance) | Secure fingerprint recognition using deep learning and FHE | Touch of Privacy: A Homomorphic Encryption-Powered Deep Learning Framework for Fingerprint Authentication |
| Enhancing security of visual data in sensitive digital applications | Demonstrated viability and effectiveness of homomorphic encryption for image processing compared to traditional methods | Image preprocessing, logistic regression training, feature extraction, polynomial approximation; RSA encryption for images | Secure processing of encrypted image data | A Comprehensive Study on the Homomorphic  Encryption for Secure Image Data Processing |
| Combining blockchain and homomorphic encryption to mitigate counterfeit goods and data breaches risks | Improved trust, resilience, and regulatory compliance in supply chain operations | Dual-layered framework: blockchain for traceability + homomorphic encryption for data privacy | Secure supply chain management | Integrating Blockchain and Homomorphic Encryption to Enhance Security and Privacy in Project Management and Combat Counterfeit Goods in Global Supply Chain Operations |
| Framework for privacy-aware health data sharing and analysis | Enables secure, efficient, and scalable statistical analysis while preserving patient confidentiality | Integrates secret-sharing, SMPC, and HE within Hyperledger Fabric blockchain | Secure collaborative healthcare analytics | Pri Collab Analysis: privacy-preserving healthcare collaborative analysis  on blockchain using homomorphic encryption and secure multiparty  computation |
| Enhances privacy and security in CPSs, especially in critical environments like mining | Demonstrates feasibility of protecting sensitive user attributes during access control decisions | Introduces ZKP-based privacy-preserving protocol for ABAC systems | Secure authentication and access control in CPSs | Privacy-preserving attribute-based access  control using homomorphic encryption |
| Advances secure and seamless continuous authentication without exposing sensitive biometric or behavioral data | Achieves high authentication accuracy with low computational overhead in real-time settings | Designs two protocols using homomorphic encryption and oblivious transfer for behavioral biometric authentication | Continuous user authentication with data privacy | Privacy-preserving continuous authentication using behavioral  biometrics |
| Guides future research on optimizing HE for secure, efficient, and trustworthy biometric systems | Highlights the need to address computational complexity, scalability, and real-time application feasibility | Reviews HE techniques, integration with ML, DL, and blockchain; analyzes challenges and future directions | Securing biometric systems (fingerprint, face, iris) using HE | A Review of  Homomorphic Encryption for  Privacy-Preserving Biometrics |
| Advances cloud security strategies for sensitive sectors like healthcare | Strengthens data confidentiality, integrity, and transmission security without directly using homomorphic encryption | **Combines SEKAC (double encryption), Improved Diffie-Hellman with ABE, and clustering-based secured transmission using enhanced algorithms** | Data privacy and secure transmission in cloud computing (healthcare focus) | A Secure Framework For Enhancing Data Privacy And  Access Control In Healthcare Cloud Management Systems |

1. **Summary**

This literature review chapter explores recent advancements in authentication, particularly integrating privacy-preserving techniques such as homomorphic encryption and decentralized architectures. It highlights the role of homomorphic encryption in biometric authentication, enabling secure computations on encrypted data to protect sensitive biometric information throughout the authentication process.

The chapter examines how artificial intelligence and machine learning, combined with homomorphic encryption, enhance the performance and accuracy of biometric recognition systems, especially in edge computing and cloud environments. It also discusses secure biometric data processing using advanced cryptographic tools to ensure privacy during storage, transmission, and matching phases.

Additionally, the chapter analyzes decentralized authentication systems and privacy-preserving frameworks like blockchain and zero-knowledge proof techniques, which enable secure access control and collaborative data processing without exposing user attributes. It also explores continuous authentication mechanisms that utilize behavioral biometrics and privacy-preserving cryptographic protocols for seamless real-time user verification.

Finally, the chapter reviews broader applications of these security methods in cloud computing and real-world systems, including healthcare, law enforcement, and supply chain management, where data confidentiality, integrity, and user trust are essential. This review sets the stage for the next chapter, which will introduce the technical foundation of the proposed authentication framework, detailing its architectural components, encryption techniques, evaluation methodologies, experimental results, and performance analysis.

**Chapter 4**

**Design of the Integrated Authentication Framework**

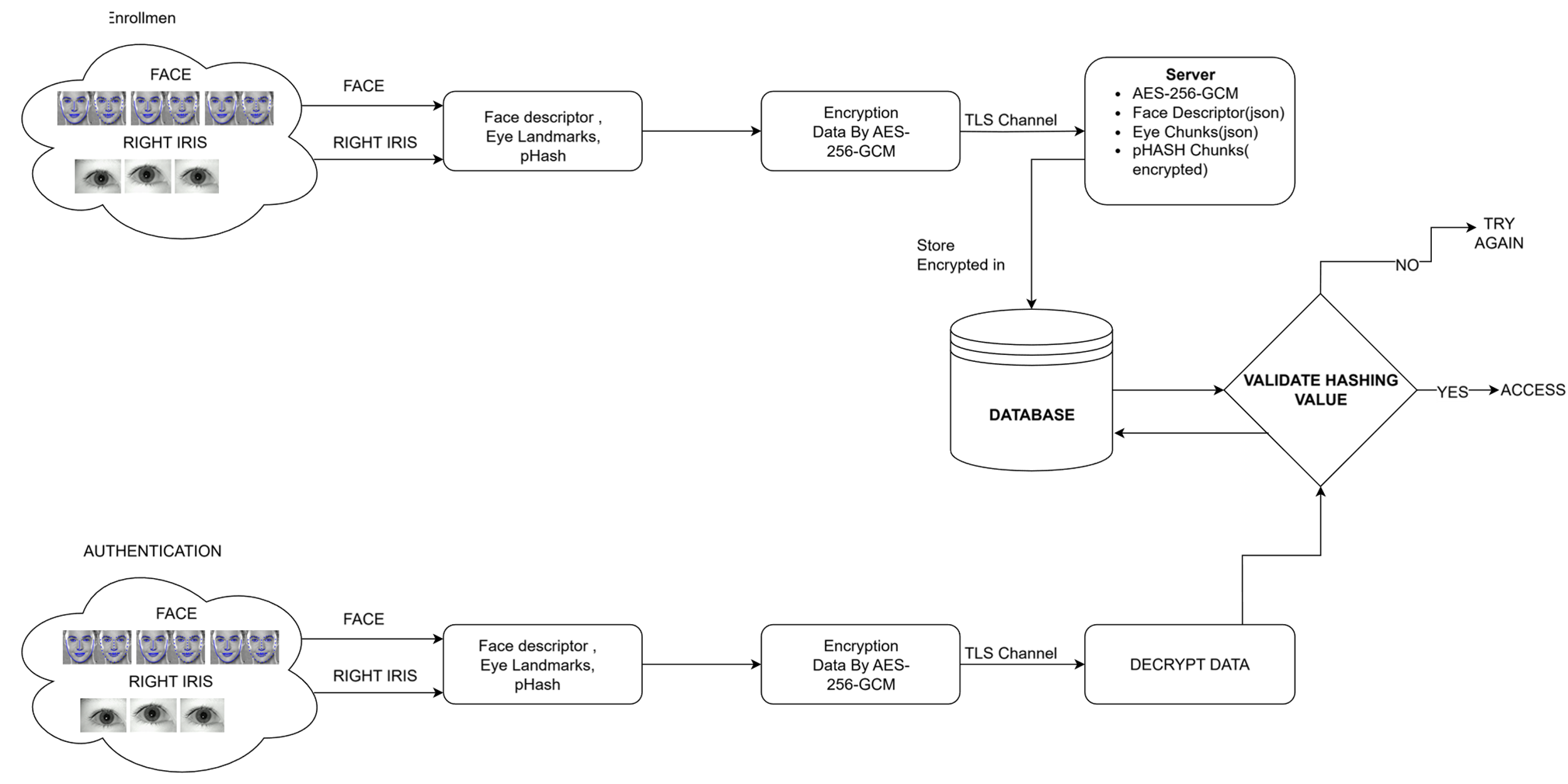
**4.1. Design Goals and Principles**

The design of the proposed authentication framework was guided by a number of goals that directly address both security and usability requirements. **The first goal is enhanced security,** achieved through the integration of multi-modal biometrics combining face descriptors and iris features. This approach reduces the likelihood of spoofing or impersonation attacks by requiring consistent verification across independent biometric traits. **A second goal is usability and accessibility**, whereby the system was implemented using a standard web camera and a simple interface to ensure that users can perform both enrollment and login without technical expertise or specialized hardware. **A third goal is efficiency and real-time performance**. By relying on lightweight but accurate models such as face-api.js for facial recognition and MediaPipe FaceMesh for iris detection, the framework ensures that authentication can be performed rapidly with minimal delay. Another important goal **is data privacy and protection**, accomplished through the encryption of biometric perceptual hashes (pHash) using AES-256-GCM and binding the ciphertext to a unique user identifier. **Finally, scalability and extensibility** were maintained by adopting a modular design that allows the future integration of additional biometric modalities or advanced cryptographic methods such as homomorphic encryption.

To achieve these goals, a number of principles were followed throughout the design. The principle of **multi-factor biometric integration ensures** that authentication depends on both facial descriptors and iris-based features, increasing reliability through feature fusion. The principle **of normalization and robust feature extraction** was applied by adjusting features relative to the eye center and width, thereby producing stable representations that are invariant to distance, scale, and orientation. The principle **of privacy-preserving storage avoids saving raw** images in the database; instead, descriptors, landmarks, and pHash values are stored in encrypted form, reducing the risk of identity theft. The principle **of security by design** incorporates encryption as a core element rather than an optional layer, with the use of Additional Authenticated Data (AAD) in AES-GCM to bind encrypted data to specific users and prevent cross-record substitution. Finally, the principle of a **user-centric experience ensures** that the framework maintains a balance between security and convenience, with streamlined workflows that allow users to complete authentication tasks through minimal interactions such as opening the camera and capturing features.

**4.2. Architectural Overview of the Proposed Framework**

The proposed authentication framework follows a client–server architecture that separates biometric feature extraction from secure storage and verification. This design ensures that the computationally intensive tasks of feature detection are performed locally on the client, while sensitive operations such as encryption, decryption, and database management are executed on the server. The architecture is divided into three main components: the client-side modules, the server-side modules, and the communication channels that connect them.



System Architecture Diagram

**4.2.1. Client-Side Components**

he client side is responsible for capturing and preprocessing biometric data. A standard web browser with access to a webcam is sufficient to perform all client-side operations, ensuring wide accessibility. The primary components include:

* **Camera Interface**: Provides real-time video input using WebRTC APIs and enables users to start, pause, or stop camera streams with minimal interaction.
* **Face Recognition Module**: Utilizes face-api.js models (Tiny Face Detector, Face Landmark 68, and Face Recognition Net) to extract a 128-dimensional face descriptor for each user.
* **Iris Feature Extraction Module**: Employs MediaPipe FaceMesh to detect detailed eye and iris landmarks. Normalized geometric features are computed and partitioned into fixed-length chunks to provide consistent and robust representations.
* **Perceptual Hashing Module**: Crops the detected facial and iris regions from video frames, generates 32×32 grayscale images, applies a two-dimensional DCT, and derives 64-bit perceptual hashes (pHash). These hashes are segmented into chunks for storage and verification.
* **User Interface**: Implements a lightweight design using TailwindCSS and Boxicons to ensure usability. Users can easily complete registration or authentication processes through simple interactions such as capturing facial and iris data.

**4.2.2. Server-Side Components**

The server side is responsible for securely storing, encrypting, and verifying biometric data. It is implemented using PHP scripts and a MySQL database. The main components are:

* **Data Storage Layer**: Stores user identifiers, names, face descriptors, iris landmark chunks, and perceptual hash values. Sensitive data such as pHash chunks are stored both in encrypted form using AES-256-GCM.
* **Encryption and Decryption Module**: Implements AES-256-GCM for biometric data protection. Each ciphertext is generated with a random initialization vector (IV) and authenticated using the user ID as Additional Authenticated Data (AAD). This ensures integrity and prevents misuse across records.
* **Verification Engine**: Handles login requests by comparing incoming biometric features against stored records. Verification involves calculating Euclidean distances between face descriptors or iris feature vectors and computing Hamming distances between perceptual hashes. Thresholds are applied to determine acceptance or rejection.
* **Database Management**: Maintains relational storage of user records with timestamped entries, enforcing consistency and scalability for multi-user environments.

**4.2.3. Communication Channels**

The communication between the client and server relies on **HTTP(S) requests** with JSON payloads. The channels are designed to be lightweight while maintaining confidentiality and integrity:

* **Data Transmission**: Client-side JavaScript sends biometric descriptors, iris chunks, and perceptual hashes in JSON format to PHP endpoints (upload.php for registration and verify.php for authentication).
* **Secure Transport**: All communication is intended to operate over **TLS/HTTPS**, ensuring that biometric data is encrypted during transit and protected from interception or tampering.
* **Response Handling**: The server responds with plain-text confirmation messages indicating success or failure of registration and verification. This feedback is immediately presented to the user via the browser interface.

**4.3. The Cryptographic Techniques**

The proposed authentication framework employs modern cryptographic techniques to ensure the protection of biometric information throughout the processes of storage, transmission, and verification. In the current implementation, the system relies on AES-256-GCM, a symmetric encryption algorithm, which provides confidentiality, integrity, and authenticity of the biometric templates, particularly the perceptual hash (pHash) chunks of both face and iris images. The use of AES-256-GCM guarantees that even if intercepted, the encrypted data cannot be understood or tampered.

**4.3.1. Rationale for Selecting an HE Scheme**

The adoption of Homomorphic Encryption is motivated by the need to perform biometric verification without exposing sensitive data in plaintext form. In traditional approaches, even if perceptual hashes or feature vectors are encrypted at rest, they must be decrypted during comparison, thereby creating potential vulnerabilities. By contrast, an HE scheme allows operations such as **Euclidean distance** or **Hamming distance** to be approximated directly on encrypted values. This property enhances privacy by ensuring that biometric data remains encrypted throughout the authentication process.

Among the available HE schemes, **Partially Homomorphic Encryption (PHE)** and **Somewhat Homomorphic Encryption (SHE)** provide efficient support for limited operations, while **Fully Homomorphic Encryption (FHE)** supports arbitrary computations. Considering the trade-off between **security and usability**, the framework prioritizes schemes that enable simple arithmetic operations (additions and multiplications) on encrypted feature chunks, thereby allowing comparison metrics to be computed securely without incurring excessive computational overhead.

**4.3.2. Key Generation and Management Protocol**

Secure key management is fundamental to the effectiveness of the proposed framework. In the implemented system, **symmetric keys for AES-256-GCM** are generated and managed exclusively on the server side. Keys are created as **256-bit random values** and stored as **Base64-encoded environment variables**, avoiding any hard-coded secrets within the application code. This practice ensures that the cryptographic material remains isolated from the application layer and can be updated without code modifications.

The secret key is never exposed to the client side; all encryption and decryption operations take place within the controlled server environment. To mitigate long-term risks, the framework supports **key rotation**, whereby keys can be periodically replaced. Upon rotation, new biometric templates may be required to ensure consistency with the updated encryption key. This management protocol guarantees that biometric data—particularly perceptual hashes of the face and iris—remain confidential and tamper-resistant throughout their lifecycle, while keeping the encryption mechanism both practical and secure.

**4.3.3. Encryption and Decryption Processes**

The framework distinguishes between the **symmetric encryption process** already

* **AES-256-GCM (Current Implementation)**
  + Encryption: Each perceptual hash chunk is serialized as JSON, encrypted with a 256-bit symmetric key, and combined with a randomly generated 12-byte initialization vector (IV). The user ID is incorporated as Additional Authenticated Data (AAD) to bind ciphertexts to specific records. The result is stored in the database as a Base64-encoded blob.
  + Decryption: During verification, the ciphertext is decoded, the IV and authentication tag are extracted, and the AES-GCM process restores the original JSON string if and only if the integrity check passes.

**4.4. Detailed Protocol Flows**

The integrated authentication framework defines two main protocol flows: the **User Enrollment Protocol**, which governs the process of registering a new user, and the **Homomorphic Authentication Protocol**, which ensures that authentication is carried out securely using encrypted biometric data. Both protocols are designed to preserve security and usability while ensuring that sensitive biometric information never leaves the client side in raw form.

**4.4.1. User Enrollment Protocol**

The enrollment protocol establishes the initial biometric reference templates that will be used for subsequent authentication attempts. The process proceeds as follows:

1. **Initialization**: The user initiates the enrollment phase by accessing the registration interface through a standard web browser. The client device activates the webcam to capture real-time video streams.
2. **Feature Extraction**:
   * Facial features are processed using face-api.js, which extracts a 128-dimensional face descriptor.
   * Iris and eye features are detected using MediaPipe FaceMesh, generating normalized landmark vectors and geometric chunks.
   * Perceptual hashes (pHash) are computed from cropped regions of the face and iris using a Discrete Cosine Transform (DCT)-based algorithm.
3. **Data Preparation**: Extracted descriptors, feature chunks, and pHash values are serialized into JSON format. Plain values are retained for reference, while sensitive data are prepared for encryption.
4. **Encryption**: The perceptual hash chunks are encrypted using **AES-256-GCM**, with the user identifier bound as Additional Authenticated Data (AAD). This ensures that encrypted values are inseparably linked to the correct user record.
5. **Transmission and Storage**: The client transmits the prepared dataset—including user ID, name, descriptors, feature chunks, and encrypted pHash values—over HTTPS to the server. The server validates the input and stores it in the MySQL database with timestamps.
6. **Confirmation**: Upon successful insertion, the server returns a confirmation message to the client, indicating that enrollment is complete and the user is now registered.

This protocol ensures that all sensitive data is stored in encrypted form, while the plain descriptors and feature chunks are only used to support robust verification thresholds.

**4.4.2. The Homomorphic Authentication Protocol**

The implemented authentication protocol relies on **AES-256-GCM**, a symmetric encryption scheme, to secure biometric data during verification. The protocol ensures confidentiality and integrity of perceptual hash chunks and feature vectors, while enabling efficient matching on the server side. The steps are as follows:

1. **Login Request**: The user initiates an authentication attempt by activating the webcam through the login interface. The client captures a live video stream.
2. **Feature Re-Extraction**:
   * A new face descriptor is generated using face-api.js.
   * Eye and iris landmarks are extracted with MediaPipe FaceMesh.
   * Perceptual hashes (pHash) of both face and iris regions are recalculated in real time.
3. **Encryption at the Client**: The newly generated biometric hashes are encrypted using **AES-256-GCM** with a 256-bit symmetric key. A random Initialization Vector (IV) is applied, and the **user ID** is incorporated as Additional Authenticated Data (AAD) to bind ciphertexts to the correct user record.
4. **Secure Transmission**: The encrypted data are transmitted to the server over a TLS-protected channel, ensuring that even if intercepted, the ciphertext cannot be tampered with or decrypted without the shared secret key.
5. **Decryption on the Server**: The server uses the same symmetric key to decrypt the ciphertext. The integrity of the data is verified through the authentication tag embedded by AES-GCM.
6. **Similarity Computation**: Once decrypted, the server compares the received face descriptor and iris features with the stored reference templates. Euclidean distance is applied for descriptors, while Hamming distance is computed for perceptual hashes.
7. **Decision Making**: If the computed similarity values fall within the predefined thresholds, authentication is granted. Otherwise, the login attempt is denied.
8. **Response to Client**: The server communicates the result (success or failure) back to the client.

This protocol achieves a balance between **security and efficiency**, ensuring that biometric data remain protected during both transmission and storage while still enabling real-time authentication performance.

**4.5. Security Model and Threat Analysis**

The security model of the implemented authentication framework is built on the integration of biometric recognition techniques with symmetric encryption using **AES-256-GCM**. The model assumes realistic threat actors and defines the capabilities against which the system is expected to maintain security. The following subsections present the assumed threat landscape and analyze the framework’s resistance to common attacks.

**4.5.1. Assumed Threat Actors and Capabilities**

The framework anticipates the presence of the following threat actors:

1. **External Network Adversaries**
   * Capabilities: Intercepting communications between the client and server, attempting to perform man-in-the-middle (MITM) attacks, replay attacks, or brute-force attempts on transmitted data.
   * Objective: Gaining unauthorized access by capturing or modifying biometric templates during transit.
2. **Malicious Insiders or Database Attackers**
   * Capabilities: Obtaining unauthorized access to the backend database through misconfigurations, SQL injection, or server compromise.
   * Objective: Extracting or manipulating stored biometric templates for impersonation or identity theft.
3. **Client-Side Attackers**
   * Capabilities: Attempting to spoof the biometric capture process, for example by replaying pre-recorded videos or static images to the client interface.
   * Objective: Circumventing liveness checks to impersonate legitimate users.
4. **Brute-Force Attackers**
   * Capabilities: Repeatedly attempting to match against stored biometric references with random or manipulated data.
   * Objective: Forcing false positives by exploiting system thresholds.

**4.5.2. Analysis of Resistance to Common Attacks (e.g., Server Breach, Man-in-the-Middle)**

The framework incorporates several defense mechanisms that mitigate the risks posed by the above threat actors:

1. **Resistance to Server Breach**
   * All perceptual hash chunks are stored in the database in an **AES-256-GCM encrypted format**. Even if an attacker gains access to the database, the ciphertext remains unintelligible without the symmetric key.
   * The use of **Additional Authenticated Data (AAD)** bound to the user ID ensures that ciphertexts cannot be trivially transplanted or reused across records.
2. **Resistance to Man-in-the-Middle (MITM) Attacks**
   * Biometric data are encrypted before transmission and sent over **TLS-secured channels**, preventing adversaries from reading or modifying the data in transit.
   * The authentication tag provided by GCM mode guarantees that any modification to the ciphertext will be detected upon decryption.
3. **Replay Attack Mitigation**
   * Because biometric data are re-captured in real time for every authentication attempt, static replays of old encrypted values are ineffective. Each session produces unique ciphertexts due to the use of **random IVs** in AES-GCM.
4. **Spoofing and Presentation Attacks**
   * Real-time capture of both **facial descriptors** and **iris features** makes simple image-based spoofing more difficult. Although full anti-spoofing techniques (e.g., liveness detection) are not yet integrated, the dual-modality approach increases robustness compared to single-biometric systems.
5. **Resistance to Brute-Force Attempts**
   * Similarity thresholds for both **Euclidean distance (face descriptors)** and **Hamming distance (pHash)** restrict acceptance to high-confidence matches.
   * Repeated failed login attempts can be monitored and rate-limited by the server to further mitigate brute-force risks.

**Chapter 5**

**Experimental Results and Discussion**

**Experimental Results and Discussion**

**5.1 Introduction**

In this chapter, the results of the proposed model and methodology are presented. The practical implementation was carried out using software interfaces for biometric image processing. Specifically, face-api.js was employed to extract facial features, while MediaPipe FaceMesh was utilized to accurately capture eye landmarks. Additionally, pHash algorithms were used to generate visual representations. Symmetric encryption techniques, namely AES-256-GCM, were applied to ensure data security.

This chapter begins with an overview of the tools and programming languages used during the implementation process. It is followed by a detailed analysis of the results obtained through user registration and biometric verification using facial and ocular data. The limitations of the proposed model are then discussed, along with its performance in practical environments. Finally, potential future improvements are suggested to enhance the model’s applicability and robustness.

**5.2 Language and tools used**

In this study, we employed a set of tools and technologies to design and implement the proposed biometric authentication framework. The primary programming languages used were **PHP** for server-side logic and **JavaScript** for client-side operations, particularly for real-time biometric data processing. For facial recognition, we integrated the **face-api.js** library, while **MediaPipe FaceMesh** was utilized to extract iris landmarks with high precision. To generate perceptual hashes (pHash) of facial and iris images, we implemented a custom algorithm based on the Discrete Cosine Transform (DCT). The storage and management of biometric templates were handled by a **MySQL** database, chosen for its efficiency and reliability. All cryptographic operations, including the symmetric encryption of sensitive data, were carried out using **AES-256-GCM**, ensuring both confidentiality and integrity. The application was hosted and tested locally using the **XAMPP Apache server**, which provided an integrated development environment. Finally, **Visual Studio Code** was used as the primary code editor, offering robust debugging and productivity features. These tools were selected to provide a balance between usability, performance, and security throughout the development lifecycle of the proposed model.

**5.3 Implementation and results**

**5.4 Summary**

**Chapter 6**

**Conclusion and Future Work**