**CHAPTER TWO**

**LITERATURE REVIEW**

**2.1 OVERVIEW**

This chapter presents a comprehensive review of literature related to file encryption systems. It explores the evolution of encryption technologies, assesses various cryptographic algorithms and their applications, and evaluates existing tools with a focus on usability, performance, and key management. The review aims to identify gaps and inform the design of a robust file encryption system tailored for secure yet user-friendly data protection.

**2.2 CONCEPTUAL FRAMEWORK**

**2.2.1 Data Security and File Encryption**

Data security refers to the process of protecting digital information from unauthorized access, corruption, or theft. File encryption is one of the fundamental methods of securing data by converting plain-text files into cipher-text using mathematical algorithms and keys. Only authorized users with the correct decryption key can reverse this process. Encryption protects data at rest and in transit, ensuring confidentiality and integrity even in the event of interception or data breach.

**2.2.2 Symmetric vs Asymmetric Encryption**

Encryption algorithms are broadly categorized into symmetric and asymmetric cryptography:

* **Symmetric Encryption**: Uses a single key for both encryption and decryption. AES (Advanced Encryption Standard) is the most widely used symmetric algorithm due to its speed and security.
* **Asymmetric Encryption**: Utilizes a pair of keys—public and private. RSA (Rivest–Shamir–Adleman) is the most popular asymmetric algorithm. It is computationally more intensive but essential for secure key distribution.

Hybrid approaches combine both methods to balance performance and security—using RSA to encrypt AES keys, which are then used for file encryption.

**2.3 REVIEW OF RELATED WORKS**

| **Author(s) & Year** | **Study Focus** | **Key Findings** | **Limitations** |
| --- | --- | --- | --- |
| Daemen & Rijmen (2002) | Design of AES | AES is secure and efficient for large-scale file encryption | Requires secure key storage to prevent misuse |
| Rivest, Shamir & Adleman (1978) | RSA cryptosystem | RSA enables secure key exchange and authentication | Slower for encrypting large files |
| Gutmann (2015) | Security architecture | Explores design principles for encryption systems | Focused on theory more than practical applications |
| Singh & Sharma (2020) | Comparative encryption study | AES is faster, RSA is more secure for key exchange | No benchmarking on hybrid approaches |
| Wang & Li (2022) | Hybrid encryption | Combining AES and RSA enhances both speed and security | Adds overhead in key generation |
| Zhang & Chen (2021) | Blockchain for encrypted file storage | Ensures data integrity and traceability | High storage costs and limited scalability |
| Jones et al. (2023) | AI in encryption | AI improves threat detection in encrypted systems | Requires continuous retraining for new threats |

These works demonstrate the ongoing research in balancing encryption strength, speed, and usability. Notably, hybrid models combining AES and RSA offer the best trade-offs.

**2.4 EXISTING TOOLS AND FRAMEWORKS**

A variety of tools exist for file encryption. BitLocker and VeraCrypt are notable full-disk encryption tools, while AxCrypt and Cryptomator are file-based systems.

* **BitLocker (Microsoft, 2021)**: Provides full-disk encryption for Windows. It is seamless for users but only available on certain editions of Windows.
* **VeraCrypt**: Open-source disk encryption tool supporting containers and hidden volumes. However, it may be complex for novice users.
* **AxCrypt**: A simple file encryption tool with integration into Windows Explorer. It uses AES-128/256 but has limited cross-platform compatibility.
* **Cryptomator**: Designed for secure cloud file storage. It encrypts files locally before upload, ensuring security even in third-party clouds.

While effective, these tools lack integrated key management systems or intuitive interfaces, especially for less technical users. There is a need for a solution that simplifies encryption while retaining strong cryptographic integrity.

**2.5 KEY MANAGEMENT STRATEGIES**

Effective key management is central to the success of any encryption system. Poor key storage or distribution undermines even the strongest algorithms. Strategies include:

* **Password-derived keys**: Easy to use but vulnerable to brute force attacks if weak passwords are used.
* **Hardware Security Modules (HSMs)**: Provide secure, tamper-proof storage but are costly and impractical for individual users.
* **Hybrid management**: Combining local password protection with asymmetric key wrapping (e.g., using RSA to encrypt AES keys) provides an effective balance.

Kumar et al. (2019) emphasize that systems should automate secure key generation and use trusted vaults for storage.

**2.6 PERFORMANCE AND USABILITY CONSIDERATIONS**

A good encryption system must not only be secure but also fast and easy to use. Metrics often used in evaluating encryption systems include:

* **Encryption/Decryption Speed**: AES is known for its high throughput.
* **System Overhead**: Excessive CPU or memory usage may deter adoption.
* **User Experience (UX)**: Clear navigation, file selection, and feedback mechanisms are essential for adoption, especially among non-experts.

Studies such as Sharma & Patel (2021) show AES significantly outperforms RSA in performance, while RSA is best reserved for secure key exchange rather than bulk data encryption.

**2.7 GAPS IN EXISTING RESEARCH**

Despite extensive research and tool development, several gaps remain:

* Lack of integrated, hybrid encryption systems with built-in key management and user-friendly interfaces.
* Limited focus on usability in open-source tools—many are command-line driven or poorly documented.
* Scarcity of evaluation on user comprehension, error recovery, and real-world adoption metrics.

These gaps justify the development of a new file encryption system that addresses security, usability, and performance in a cohesive, scalable framework.

**2.8 SUMMARY**

This chapter reviewed the landscape of file encryption, examining key algorithms, existing tools, and relevant academic research. AES and RSA remain dominant due to their respective strengths in speed and security. However, their integration into user-friendly systems is still lacking. The insights gained from this review inform the subsequent design of a hybrid file encryption system that emphasizes strong security, practical usability, and effective key management.