International Journal of Computer Techniques–IJCT Volume 13 Issue 2, March-April - 2026
Open Access and Peer Review Journal ISSN 2394-2231 https://ijctjournal.org/

AN INTELLIGENT CYBERSECURITY FRAMEWORK FOR MULTI-MODAL DATA ENCRYPTION AND DECRYPTION IN WEB APPLICATIONS

GHIRIJHA V P , LAKSHMI PRIYA S J, NIGIL CHRYSO PRAISES N C

Department Of Information Technology, Arunachala College Of Engineering For Women.

ABSTRACT

The rapid growth of web applications has led to an exponential increase in the volume and diversity of data handled online, including text, images, audio, video, and structured logs—collectively referred to as multi-modal data. Traditional security mechanisms often apply uniform encryption strategies that fail to address the distinct characteristics and threat profiles of different data modalities. This paper proposes an intelligent cybersecurity framework that integrates adaptive encryption, machine learning–based threat assessment, and secure key management to provide robust, end-to-end protection for multi-modal data in web applications. The framework dynamically selects encryption algorithms and key lengths based on data type, sensitivity, and contextual risk, thereby optimizing both security strength and system performance. Experimental evaluation demonstrates improved resistance to common web-based attacks while maintaining acceptable latency and scalability.

Keywords: Cybersecurity, Multi-Modal Data, Web Applications, Encryption, Decryption, Machine Learning, Adaptive Security

1. Introduction

Web applications have become integral to modern digital ecosystems, supporting e-commerce, healthcare, finance, education, and social networking platforms. These applications routinely process multi-modal data such as textual user inputs, multimedia uploads, transactional records, and real-time sensor streams. As data diversity increases, so does the attack surface, exposing applications to threats including data breaches, man-in-the-middle attacks, injection attacks, and unauthorized access.

Conventional encryption techniques typically employ static algorithms and keys, regardless of data modality or risk context. While effective to a certain extent, such approaches may lead to unnecessary computational overhead or insufficient protection for high-risk data. Recent advancements in artificial intelligence and machine learning enable systems to make context-aware decisions, opening new possibilities for adaptive cybersecurity solutions.

This research introduces an intelligent cybersecurity framework designed specifically for multi-modal data encryption and decryption in web applications. By combining adaptive

cryptographic mechanisms with intelligent threat analysis, the proposed system aims to enhance data confidentiality, integrity, and availability.

2. Related Work

Existing research in web security primarily focuses on single-modal data protection, secure communication protocols, and access control mechanisms. Symmetric algorithms such as AES and asymmetric algorithms such as RSA and ECC are widely adopted for data encryption and key exchange. Hybrid encryption models combine these techniques to balance security and performance.

Recent studies have explored the application of machine learning for intrusion detection, anomaly detection, and risk assessment. However, limited work has addressed the integration of intelligent decision-making with encryption strategies tailored to different data modalities. Moreover, most existing frameworks do not dynamically adjust encryption parameters based on real-time threat intelligence.

The proposed framework bridges this gap by introducing a unified architecture that supports multi-modal data handling, adaptive encryption selection, and intelligent security monitoring.

3. Proposed Intelligent Cybersecurity Framework

3.1 System Architecture

The proposed framework consists of the following core components:

1. **Multi-Modal Data Analyzer:** Identifies data type (text, image, audio, video, structured data) and sensitivity level.
2. **Risk Assessment Engine:** Utilizes machine learning models to evaluate contextual risk based on user behavior, request patterns, and historical attack data.
3. **Adaptive Encryption Module:** Selects suitable encryption algorithms (e.g., AES-256, ChaCha20, RSA-2048, ECC) and key sizes based on analyzer and risk engine outputs.
4. **Secure Key Management System:** Handles key generation, storage, rotation, and revocation using secure hardware or software vaults.
5. **Decryption and Access Control Layer:** Ensures that only authenticated and authorized entities can decrypt and access protected data.

3.2 Encryption and Decryption Workflow

When data is submitted to the web application, the system first classifies the data modality and sensitivity. The risk assessment engine then computes a threat score. Based on these parameters, the adaptive encryption module applies an optimal encryption strategy before data storage or transmission. During retrieval, the decryption process is triggered only after successful authentication and authorization checks.

3.3 Intelligent Decision-Making

Machine learning models, such as decision trees or lightweight neural networks, are trained on historical security logs to predict potential threats. This enables proactive strengthening of encryption for high-risk scenarios, such as suspicious login attempts or anomalous data access patterns.

4. Software Requirements Specification (Based on Implementation)

4.1 Overview

The developed system is a Flask-based web application that provides an intelligent cybersecurity framework for multi-modal data encryption, decryption, and basic machine learning–based analysis. The application supports text, image, video, and password data, integrating cryptographic techniques and an SVM model to enhance data security in web environments.

4.2 Functional Requirements

4.2.1 File Upload and Handling

- The system shall allow users to upload files through a web interface.
- The system shall securely store uploaded files in a predefined server directory.
- The system shall validate file names using secure filename handling.
- The system shall support multiple file types including text, image, and video files.

4.2.2 Text Data Encryption and Decryption

- The system shall encrypt text files using symmetric key encryption (Fernet).
- The system shall decrypt encrypted text files upon user request.
- The system shall generate encrypted and decrypted output files automatically.
- The system shall display appropriate success or failure messages to the user.

4.2.3 Image Processing and Security

- The system shall preprocess uploaded image files for machine learning analysis.
- The system shall apply an SVM-based prediction model on image data.
- The system shall store prediction results in a downloadable text file.
- The system shall encrypt image files using AES symmetric encryption.
- The system shall decrypt image files using the corresponding AES key.
- The system shall securely generate and store encryption keys for images.

4.2.4 Video Data Encryption and Decryption

- The system shall support encryption of video files using AES encryption.
- The system shall generate a unique encryption key and initialization vector for each video.
- The system shall decrypt encrypted video files using the stored key.
- The system shall notify the user if decryption fails due to missing or invalid keys.

4.2.5 Password Encryption and Decryption

- The system shall allow users to encrypt plain-text passwords.
- The system shall decrypt encrypted password tokens upon request.
- The system shall ensure that password data is never stored in plain text.

4.2.6 Download Management

- The system shall allow users to download encrypted, decrypted, and prediction result files.
- The system shall securely transmit files to the user upon download request.

5. Non-Functional Requirements

5.1 Security Requirements

- The system shall use industry-standard cryptographic algorithms such as AES and Fernet.
- The system shall ensure confidentiality of uploaded and processed data.
- The system shall prevent unauthorized file access through controlled routing.
- The system shall handle encryption keys securely during runtime.

5.2 Performance Requirements

- The system shall perform encryption and decryption with minimal delay for small to medium-sized files.
- The system shall handle concurrent user requests efficiently in a web environment.
- The system shall ensure acceptable response time for SVM-based image prediction.

5.3 Usability Requirements

- The system shall provide a simple and user-friendly web interface.
- The system shall display clear messages for encryption, decryption, and prediction operations.

- The system shall support common file formats used in web applications.

5.4 Scalability Requirements

- The system shall be modular to allow future integration of additional data types.
- The system shall support extension to advanced machine learning or deep learning models.
- The system shall be deployable on local servers or cloud platforms.

5.5 Reliability Requirements

- The system shall handle incorrect inputs gracefully without crashing.
- The system shall detect and report decryption failures.
- The system shall maintain consistent behavior across repeated operations.

6. Software Requirements

- **Operating System:** Windows / Linux
- **Programming Language:** Python 3.x
- **Web Framework:** Flask
- **Machine Learning Library:** Scikit-learn
- **Cryptography Libraries:** Cryptography (Fernet), PyCryptodome (AES)
- **Image Processing:** Pillow (PIL)
- **Data Handling:** NumPy, Joblib
- **Web Technologies:** HTML, CSS (Frontend)
- **Server:** Flask Development Server / WSGI Server

7. Hardware Requirements

- **Processor:** Dual-core or higher
- **RAM:** Minimum 8 GB
- **Storage:** 256 GB or more
- **Network:** Stable Internet connection

8. Constraints

- Encryption keys for images and videos are stored locally in key files.
- The system operates in a controlled server environment.
- Debug mode is enabled during development.

- Large file processing may be limited by system memory.

9. Assumptions

- Users upload valid and supported file formats.
- The SVM model is pre-trained and correctly loaded.
- The server environment is secure and trusted.

10. Implementation Details

The framework is implemented as a modular middleware layer compatible with modern web technologies. Backend services are developed using secure server-side frameworks, while encryption operations leverage standardized cryptographic libraries. RESTful APIs facilitate seamless integration with existing web applications. Emphasis is placed on minimizing performance overhead through efficient algorithm selection and caching strategies.

11. Experimental Evaluation

11.1 Experimental Setup

The framework is evaluated using a prototype web application handling multi-modal datasets. Performance metrics include encryption/decryption latency, throughput, and system scalability under varying load conditions. Security evaluation considers resistance to common web attacks such as brute force, replay, and data tampering.

11.2 Results and Discussion

Experimental results indicate that adaptive encryption significantly reduces unnecessary computational cost for low-risk data while providing enhanced protection for sensitive data. The intelligent framework demonstrates improved detection of anomalous behavior and maintains secure operations with minimal impact on user experience.

12. Security Analysis

The proposed framework strengthens confidentiality through strong, context-aware encryption and ensures integrity via secure key management and access controls. Availability is maintained by balancing security measures with system performance. The integration of intelligent risk assessment further enhances resilience against evolving cyber threats.

13. Conclusion and Future Work

This paper presents an intelligent cybersecurity framework for multi-modal data encryption and decryption in web applications. By combining adaptive cryptographic techniques with machine learning–based risk assessment, the framework addresses the limitations of traditional static security approaches. Future work will focus on incorporating advanced deep learning

models, supporting edge and cloud-native deployments, and conducting large-scale real-world evaluations.

14. References

1. Stallings, W. (2019). Cryptography and Network Security: Principles and Practice. Pearson Education.

2. Bishop, M. (2018). Computer Security: Art and Science. Addison-Wesley Professional.

3. Schneier, B.(2015). Applied Cryptography Protocols, Algorithms, and Source law in C. Wiley.

4. Alsmadi, I., & Xu, S. (2020). "Security of AI and AI for Security: Challenges and Opportunities." IEEE Security & Privacy, 18(3), 14–25.

5. Shaukat, K., Luo, S., Varadharajan, V., & Hameed, I. A.(2020). " A Survey on Machine Learning ways for Cybersecurity. " IEEE Access, 8, 222310 – 222354.

6. Dua, S., & Du, X. (2016). Data Mining and Machine Learning in Cybersecurity. CRC Press.

7. Patel, A., Taghavi, M., Bakhtiyari, K., & Javadi, B. (2013). "An intrusion detection and prevention system in cloud computing: A systematic review." Journal of Network and Computer Applications, 36(1), 25–41.

8. Zhang, C., & Wang, J. (2021). "AI-driven encryption for secure multimedia transmission." International Journal of Information Security, 20(2), 151–168.

9. Shafiq, M., & Ahmad, H. F. (2020). "Deep learning-based network intrusion detection systems: A review." IEEE Access, 8, 219650–219670.

10. Wang, P., Chen, C., & Zhao, L.(2022). " A mongrel Cryptographic Model forMulti-Modal Data Protection in Web operations. " Journal of Information Security and Applications, 66, 103115.