Assignment 3.2: Human Resources (HR) OQS Implementation
Group Members: Abhishek Mahajan, Gavin Zhao, Nate Lavoy, Qahtan Al Jammali, Tejaswini Ojha

Introduction

The objective of this project is to protect the valuable assets dealt with in the Human Resources (HR) Domain using cryptographic techniques and libraries. Through a comprehensive threat modeling process, Personally Identifiable Information (PII), financial data, sensitive records, and federated learning infrastructure were identified as vulnerable assets. Such sensitive information faces threats from both insider and outsider attacks, such as phishing scams, data breaches, and disgruntled employees.

Scope

This implementation of cryptographic techniques helps ensure information confidentiality of PII within the human resources domain, increasing security against inside and outside actors. The techniques used in our implementation can be recreated to protect data that is not limited to PII or the HR sector.

Problem Statement

Personally Identifiable Information is a critical part of HR operations and its scope includes personal data (eg. SSNs, contact information, addresses, and emergency contacts), financial information (eg. bank account details, salary information, and tax documents), and medical records (eg. insurance information, health benefits, and medical history). PII can be targeted by cybercriminals to commit identity theft or financial fraud. Outsider knowledge of medical information is not only a breach of privacy, but it can create an opening for mistreatment through insurance discrimination, decreased employment opportunities, or personalized scams. PII privacy is protected by laws such as GDPR (General Data Protection Regulation), CCPA (California Consumer Privacy Act), and HIPAA (Health Insurance Portability and Accountability Act).

A typical HR department may not have access to such a complete set of sensitive information, but data such as employee records and payroll information are certainly susceptible to malicious attacks. Standard encryption techniques do well enough to protect our sensitive information, making it unusable to those that manage to intercept it. But this won't be true for long. The computation power of quantum computers poses a prominent threat to traditional encryption protocols that rely on mathematically complex algorithms. Quantum technology is still years away from becoming a reality, but it is important to transition to post-quantum safe cryptography before then. Practices such as "Harvest Now, Decrypt Later" involve the interception of sensitive data in its encrypted form. Although many types of data will become irrelevant or expire when quantum computing becomes available, PII such as social security numbers and medical history will certainly still be relevant and can be used maliciously by cybercriminals in the future.

We aim to protect Personally Identifiable Information by transitioning HR data protection to Post-Quantum Safe Cryptography. A hybrid quantum and classical encryption will be implemented on sensitive PII data, ensuring resilience to future quantum-enabled decryption and adherence to state-of-the-art cryptographic standards.

Solution Approach

Our solution aims to create a robust, multilayered defense system by combining quantum and classical encryption algorithms into a hybrid system, mitigating the risk of a single point of failure. Our cryptographic implementation uses Kyber1024 for quantum-resistant key encapsulation, RSA-3072 for traditional security, AES-256 for symmetric encryption, and SHA-256 for key combination and hashing. Two sets of public and private keys are generated using Kyber and RSA. Then two shared secrets are generated for symmetric encryption. They are then concatenated and hashed to produce the hybrid shared secret. A file with sensitive information is encrypted using the shared secret, and decrypted using the same. The implementation uses liboqs from the Open Quantum Safe project.

Implementation ()

First the host who will receive the encrypted data (receiver) creates two sets of public/private key pairs: one using Kyber and one with RSA. The public keys are sent to the host who will send the encrypted data (sender).

The sender creates two random shared secrets, concatenates them, and then hashes the product with SHA-256. This hash is then used as the symmetric key. The AES algorithm is then used to encrypt the data file using the key. The sender encapsulates the shared secrets, one with the Kyber public key and one with the RSA public key. The sender then sends the encrypted file and the two encapsulated shared secrets back to the receiver.

The receiver then decapsulates the shared secrets with the two corresponding private keys, then concatenates and hashes to get the symmetric key. The receiver then decrypts the original data.

For more information on how to run the code that emulates this, see the README.md

Testing and Validation

To ensure the effectiveness of our implementation, a series of comprehensive tests were conducted. The primary focus of these tests was to evaluate the accuracy of encryption and decryption, measure the performance overhead introduced by the cryptographic algorithms, and assess the resilience of the system against simulated attacks. For accuracy testing, sample files containing synthetic Personally Identifiable Information (PII), such as fabricated employee records and payroll information, were encrypted using the hybrid cryptographic method. These encrypted files were then decrypted using the corresponding private keys to ensure that the original data was restored without any discrepancies. This process validated the correctness of our encryption and decryption pipelines, confirming that the implementation was lossless.

Performance testing was conducted to measure the computational overhead introduced by the hybrid encryption scheme. Files of varying sizes, from small text documents under 1 KB to larger datasets exceeding 100 MB, were tested to observe the scalability of the system. Metrics such as the time taken for encryption, decryption, and key generation were recorded and analyzed.

The resilience of the implementation was evaluated through simulated attack scenarios. In one scenario, we mimicked a man-in-the-middle (MITM) attack by intercepting encrypted data during transmission. The system's use of quantum-safe cryptography algorithms like Kyber1024 ensured that intercepted ciphertext could not be decrypted without the corresponding private keys. Another test involved simulating a brute-force attack to assess the feasibility of decrypting the ciphertext without the appropriate keys. Given the high computational complexity of the hybrid encryption method and the robust key lengths used, these attacks were rendered infeasible. Additionally, the system was tested against the "Harvest Now, Decrypt Later" threat, wherein attackers might collect encrypted data today to decrypt it using quantum computing in the future. The adoption of Kyber1024, a post-quantum cryptographic algorithm, provided strong assurances against such future risks.

Results and Observations

The results of the implementation and testing phase demonstrated significant improvements in the security and resilience of the system. The use of Kyber1024 ensured that sensitive HR data was protected against future quantum decryption threats. The hybrid approach combining Kyber, RSA, and AES offered a layered defense mechanism that mitigated the risk of a single point of failure. Encryption and decryption accuracy tests confirmed that the implementation was reliable and capable of restoring original data without any errors.

In terms of performance, the hybrid encryption scheme handled both small and large datasets efficiently. While there was a noticeable increase in processing time for larger files, this was expected due to the added computational complexity of combining multiple cryptographic algorithms. The performance overhead was deemed acceptable, particularly for applications where security is prioritized over real-time processing. The system’s scalability was also evident, as it successfully managed encryption and decryption tasks across a range of file sizes without degradation in performance.

The robustness of the system was validated through attack simulations. The encryption scheme demonstrated strong resistance to interception, brute force, and quantum-related threats. By integrating both classical and quantum-safe cryptographic techniques, the system effectively addressed the potential vulnerabilities associated with each method individually. This dual-layered approach enhanced the overall security posture, providing a future-proof solution for sensitive HR data protection.

Challenges and Limitations

While the implementation met its objectives, a few challenges and limitations were observed. The hybrid encryption scheme introduced some computational overhead due to the combination of quantum-safe and classical algorithms, making it less suitable for high-performance or real-time scenarios without further optimization. Key management complexity was another issue, as handling multiple keys for Kyber, RSA, and AES required robust protocols to ensure security and prevent vulnerabilities like unauthorized access or key loss. Developing automated key management or using hardware security modules would address this in production environments.

We had planned to create a web application for visualizing the encryption and decryption, but it was not fully developed due to time constraints, limiting its potential to demonstrate usability and practicality in real-world HR settings. Fully developing this tool in future iterations could make the system more accessible to non-technical users. Lastly, while Kyber1024 is designed to resist quantum decryption, its effectiveness depends on future quantum computing advancements. Continuous monitoring and adaptation to emerging technologies will be crucial to maintain the system’s long-term security.

Future Work

The findings from this project identified key areas for improvement. Addressing computational overhead could involve optimizing cryptographic operations through parallel processing or algorithmic refinements, potentially reducing processing times without compromising security. Exploring alternative quantum-resistant algorithms may also lead to more efficient solutions.

Key management remains a critical focus; integrating automated systems or hardware security modules could streamline key generation, storage, and distribution while minimizing human error. Completing the development of the web application would enhance usability, allowing HR professionals to visualize encryption processes and better understand post-quantum cryptography.

Testing the system in real-world HR environments would provide valuable insights into its practicality and effectiveness. Integrating with existing HR information systems (HRIS) and expanding the solution to protect other sensitive data could further enhance its versatility and applicability.

Conclusion

This project successfully demonstrated the feasibility of transitioning HR data protection to post-quantum safe cryptography. By combining Kyber1024 for quantum resistance, RSA-3072 for classical security, and AES-256 for symmetric encryption, the implementation provided a robust, multilayered defense system. Testing validated the accuracy, scalability, and resilience of the hybrid encryption scheme, while simulated attacks confirmed its effectiveness against both current and future threats.

Despite challenges such as computational overhead and key management complexity, the project achieved its objective of future-proofing HR data against emerging quantum computing capabilities. The findings from this work underscore the importance of proactive adoption of post-quantum cryptographic techniques and provide a foundation for further research and development in the field of data security. With continued optimization and real-world testing, this implementation has the potential to set a new standard for protecting sensitive information in the HR domain and beyond.

For more reading, please refer to the README.md file that shows how to set up the project and test it out yourself. There are also photos of the incomplete web application included in a separate Github branch to visualize a potential platform for employee use.

Pyproject.toml contains all the dependencies.

Please follow the logic in the README.md to deploy the project, as no main.py is needed.