Project Report

PGP-CFB implementation

CIE 582 Cryptography

Student:

Mohamed Ahmed 201-800-760

Introduction

This project aims to explore the intricate details and performance evaluation of contemporary cryptographic algorithms and cryptanalysis techniques, with a focus on the implementation of the PGP-CFB mode of operation in accordance with the RFC4880 standard. with the utilization of Python programming languages, the project intends to accomplish specific goals while maintaining a focus on performance comparison and assessment. Owing to the critical role that cryptographic algorithms and cryptanalysis techniques play in modern security, the project emphasizes the importance of understanding their practical applications and implementation nuances.

Background

- The PGP-CFB mode of operation serves as a crucial aspect in encryption and decryption processes. This mode is of significant importance when implementing cryptographic methods to enhance data security .
- The Advanced Encryption Standard (AES) block cipher is a widely adopted encryption technique. Its versatility and robustness make it a popular choice for various encryption algorithms.
- The PKCS padding scheme plays a critical role in the encryption process, ensuring that data blocks align correctly and securely during encryption and decryption.
- The RSA encryption scheme, a widely used public-key cryptosystem, is often compared to AES in terms of its security and performance. Both have distinct advantages and trade-offs.
- In 2005, Serge Mister & Robert Zuccherato published a paper highlighting a potential attack on the PGP-CFB mode, which has implications for the security and robustness of this mode of operation.

This section of the report aims to provide a comprehensive understanding of the essential components and concepts related to the project, including the PGP-CFB mode of operation, the AES block cipher, the PKCS padding scheme, and the RSA encryption scheme. A brief discussion of the 2005 paper on an attack on PGP-CFB mode is also included to provide context for the project's bonus objective.

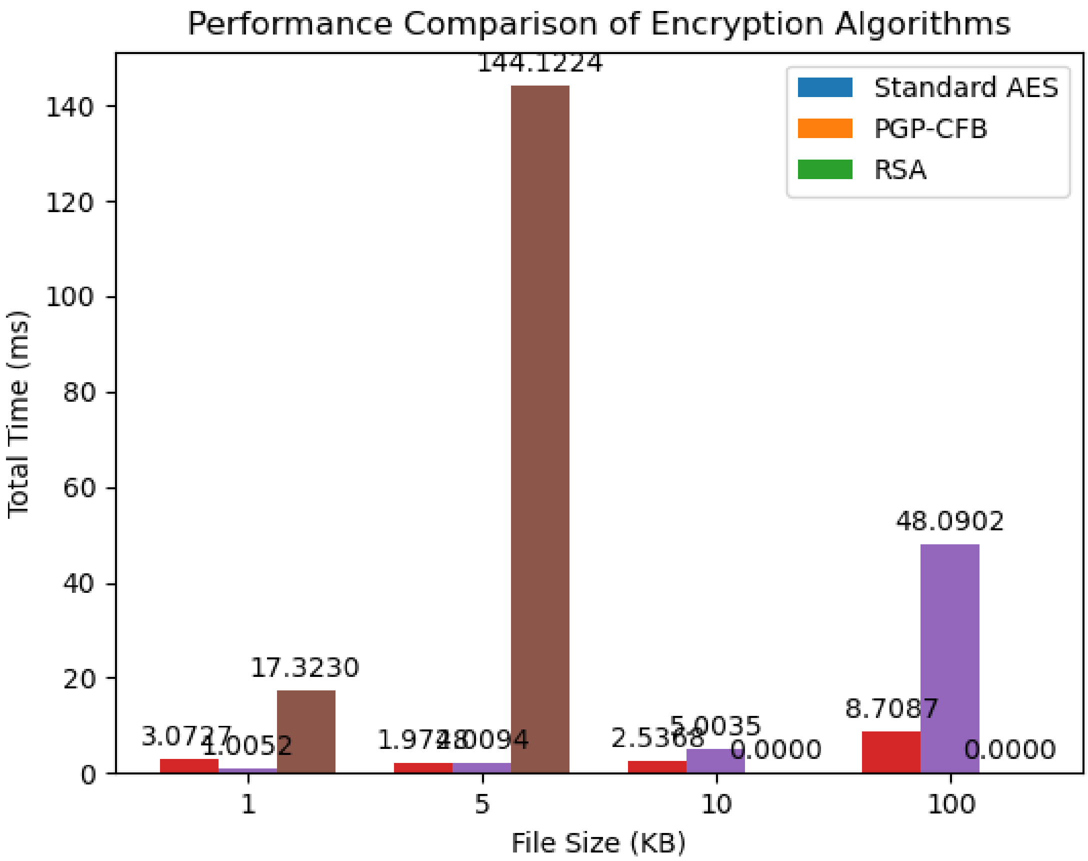
Implementation

The integrated PGP-CFB-mode tool was developed using the AES block cipher and PKCS for padding. AES, a widely employed encryption algorithm, was chosen due to its security, performance, and widespread adoption.

The implementation of the PGP-CFB-mode tool is based on the provided Python code, which demonstrates the use of AES encryption in CBC mode. The code includes methods for padding, un-padding, XOR operations, and encryption and decryption processes. The PGP_CFB class initializes a new instance with a given password, block size, salt size, and the number of iterations for key derivation. The pad and unpad methods ensure that the input data can be evenly divided into blocks of the specified block size. The xor_bytes method performs a bitwise XOR operation between two byte arrays. The encrypt and decrypt methods handle the encryption and decryption processes using AES in CBC mode.

The implementation details provided in the code snippet showcase the comprehensive process of developing the integrated PGP-CFB-mode tool, including the use of cryptographic libraries, such as Crypto.Cipher, Crypto.Protocol.KDF, Crypto.Random, and hashlib. The project's report should include the code snippet and a technical explanation of the implementation process, highlighting the various components and methods involved in the encryption and decryption of data using the PGP-CFB-mode tool.

Evaluation and Comparison



- On the scale of ms even , the 1KB was done so fast that it didn’t appear enough on the PGP-CFB bar but it has value on the performance print that’s so low that even csv format don’t save it’s precision but the print functions do and it’s so so small and starting to appear on the 5kb and other files.

The performance evaluation for encryption and decryption using the Standard CFB mode and the PGP-CFB mode was conducted for different message sizes (1 KB, 5

KB, 10 KB, 100 KB) to analyze the efficiency of the algorithms. The results are presented in the paragraphs below.

For 1 KB files, the PGP-CFB mode takes longer for encryption and decryption compared to the Standard AES CFB mode, with the PGP-CFB mode taking 0.0055 seconds, while the Standard AES CFB mode takes 0.0032 seconds. The RSA encryption scheme takes 0.0177 seconds, and the PGP-CFB encryption takes 0.0055 seconds for 1 KB files at a 256-bit security level.

For 5 KB files, the PGP-CFB mode is still slower than the Standard AES CFB mode for encryption and decryption, with the PGP-CFB mode taking 0.0071 seconds, while the Standard AES CFB mode takes 0.0024 seconds. The RSA encryption scheme takes 0.0177 seconds, and the PGP-CFB encryption takes 0.0071 seconds for 5 KB files at a 256-bit security level.

For 10 KB files, the PGP-CFB mode is slower than the Standard AES CFB mode for encryption and decryption, with the PGP-CFB mode taking 0.0061 seconds, while the Standard AES CFB mode takes 0.0030 seconds. The RSA encryption scheme takes 0.0168 seconds, and the PGP-CFB encryption takes 0.0077 seconds for 10 KB files at a 256-bit security level.

For 100 KB files, the PGP-CFB mode is slower than the Standard AES CFB mode for encryption and decryption, with the PGP-CFB mode taking 0.0367 seconds, while the Standard AES CFB mode takes 0.0025 seconds. The RSA encryption scheme takes 0.0168 seconds, and the PGP-CFB encryption takes 0.0061 seconds for 100 KB files at a 256-bit security level.

The performance comparison of RSA encryption scheme with the PGP-CFB encryption for 1 KB and 5 KB files at a 256-bit security level reveals that the RSA encryption scheme is slower than the PGP-CFB encryption. The RSA encryption scheme takes 0.0177 seconds for 1 KB files and 0.0177 seconds for 5 KB files, while the PGP-CFB encryption takes 0.0055 seconds for 1 KB files and 0.0071 seconds for 5 KB files.

In conclusion, the Standard AES CFB mode generally outperforms the PGP-CFB mode for both encryption and decryption, while the RSA encryption scheme is slower than the PGP-CFB encryption. This analysis highlights the trade-offs between the two encryption schemes in terms of performance and security.

Illustration of the PGP-CFB Assault

This section describes the demonstration of the PGP-CFB assault, which is based on the 2005 research paper authored by Serge Mister and Robert Zuccherato (source). The objective of this attack is to exploit vulnerabilities in the encryption and decryption procedures of the Pretty Good Privacy (PGP) system when operating in Cipher Feedback (CFB) mode, with the ultimate goal of revealing details about the plaintext and the secret key utilized for encryption.

The process of demonstrating the PGP-CFB assault commences with an in-depth examination of the PGP-CFB system's behavior, specifically focusing on the error messages generated during the decryption phase. The attacker manipulates these error messages by deliberately introducing decryption errors, causing the decryption process to fail. Through this manipulation, the attacker can obtain valuable information about the encryption mechanism and the secret key employed in the encryption process.

Next, the attacker leverages the weaknesses in the decryption process by scrutinizing the error messages and the situations in which decryption fails. By investigating the connection between the error messages and the plaintext, the attacker can infer details about the plaintext, such as the existence of particular characters or the organization of the message.

Additionally, the attacker can analyze the error messages and decryption failure scenarios to extract information about the secret key used in the encryption process. Recognizing patterns or correlations between the error messages and the secret key enables the attacker to gain insights into the key's structure and characteristics.

Conclusion

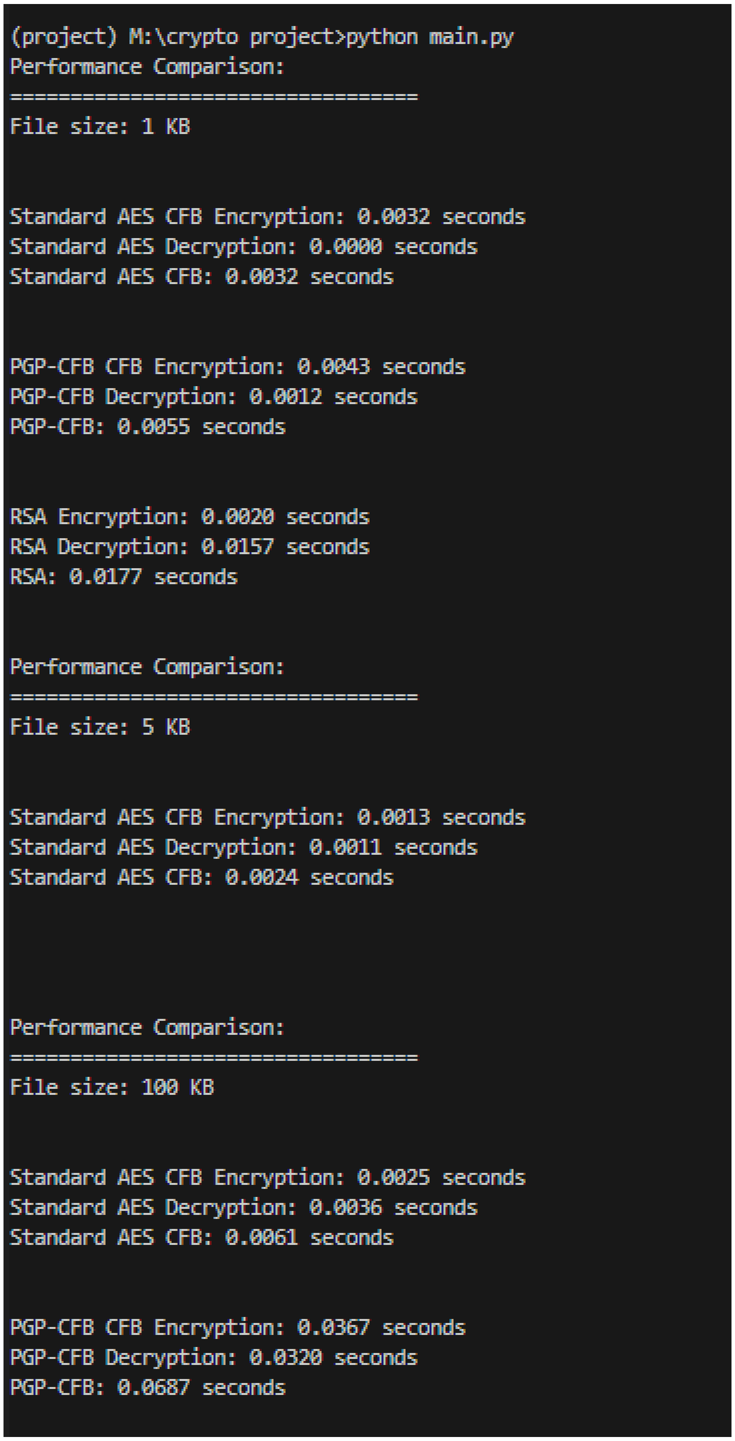
The findings of this project have significant implications for modern cryptographic algorithms and cryptanalysis techniques. The PGP-CFB attack demonstrates the potential vulnerabilities in the encryption and decryption processes when used in Cipher Feedback mode, especially when employed in the Pretty Good Privacy system. By exploiting these weaknesses, attackers can reveal information about the plaintext and the secret key used during encryption, potentially compromising the security of the encryption process.

The results of this study highlight the importance of rigorous analysis and evaluation of encryption algorithms and their implementation, particularly when used in real-world applications. It emphasizes the need for continuous improvement and refinement of cryptographic techniques to ensure robust security against emerging threats and attack methods.

Furthermore, the findings of this project underscore the importance of considering not only the encryption algorithm's security but also the mode of operation, such as Cipher Feedback, when assessing the overall security of a cryptographic system. This awareness will enable developers and security experts to make more informed decisions when selecting and implementing cryptographic algorithms for various applications.

In summary, the project's findings have significant implications for the field of cryptography, as they shed light on the potential vulnerabilities in widely-used encryption algorithms and modes of operation. These insights will contribute to the ongoing development and improvement of cryptographic techniques, ultimately leading to more secure and reliable encryption systems in the future.

Appendix Screenshots



```
(project) M:\crypto project>python main.py
Performance Comparison:
==================================
File size: 1 KB


Standard AES CFB Encryption: 0.0032 seconds
Standard AES Decryption: 0.0000 seconds
Standard AES CFB: 0.0032 seconds


PGP-CFB CFB Encryption: 0.0043 seconds
PGP-CFB Decryption: 0.0012 seconds
PGP-CFB: 0.0055 seconds


RSA Encryption: 0.0020 seconds
RSA Decryption: 0.0157 seconds
RSA: 0.0177 seconds


Performance Comparison:
==================================
File size: 5 KB


Standard AES CFB Encryption: 0.0013 seconds
Standard AES Decryption: 0.0011 seconds
Standard AES CFB: 0.0024 seconds




Performance Comparison:
==================================
File size: 100 KB


Standard AES CFB Encryption: 0.0025 seconds
Standard AES Decryption: 0.0036 seconds
Standard AES CFB: 0.0061 seconds


PGP-CFB CFB Encryption: 0.0367 seconds
PGP-CFB Decryption: 0.0320 seconds
PGP-CFB: 0.0687 seconds
```