Encrypted File System Architecture

SOHAM FALDU 19BCI0024
VIDHI MOTERIA 19BCE0525
RASHI MAHESHWARI 19BDS0006
ROHAN ALLEN 18BCI0247
PHANIDER EKDUKALLA 18BCI0253

Abstract

Increasing thefts of sensitive data owned by individuals and organizations call for an integrated solution to the problem of storage security. Most existing systems are designed for personal use and do not address the unique demands of enterprise environments. An enterprise-class encrypting file system must take a cohesive approach towards solving the issues associated with data security in organizations.

Developing a high security and good speed solution for the problem. Using some of the fastest algorithms like twofish for encryption.

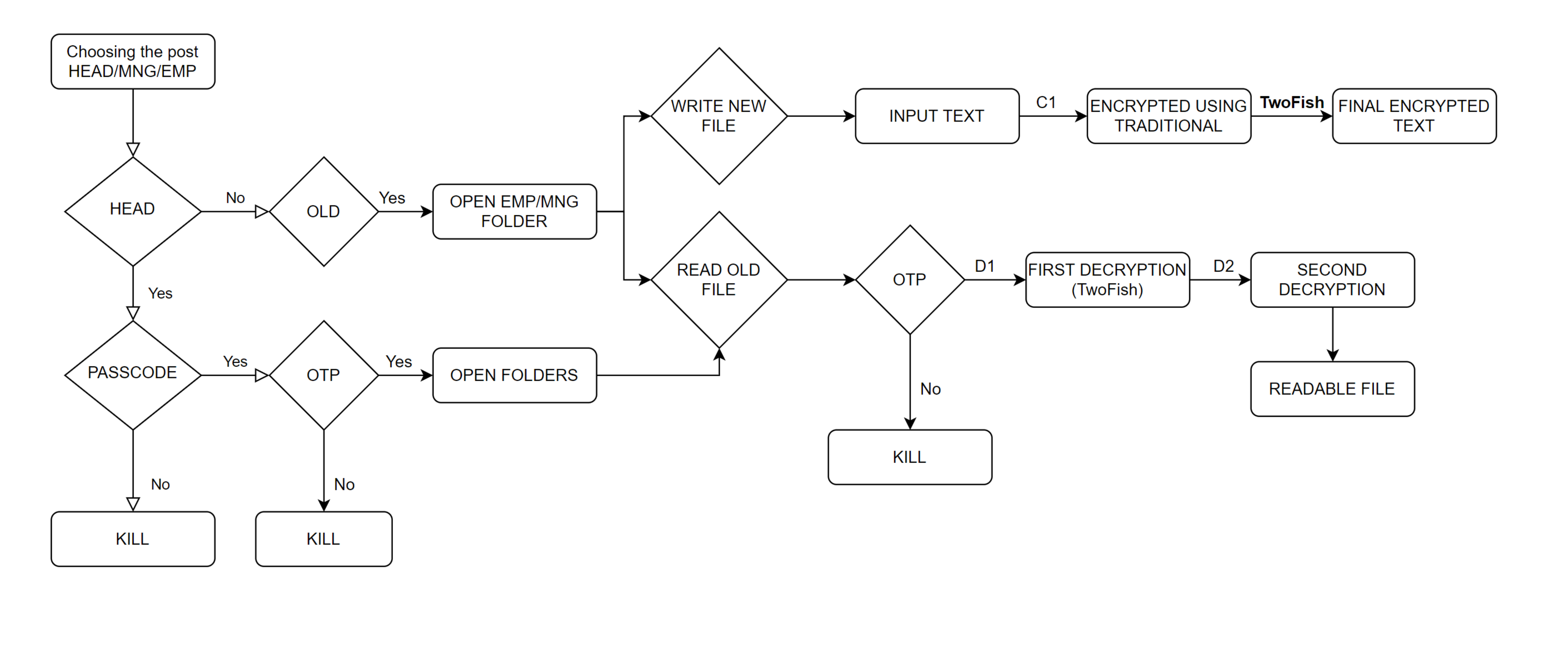
Problem Statement

Developing an encrypted file system that is both secure and fast. Looking over the issue of time complexity with cryptanalysis. The architecture would be for organizational use for large companies. We are going to use some of the fastest encryption algorithm for securing the architecture. Encrypting files of organization so that if the opponent party tries to hack and take the file he/she won’t be able to encrypt it.

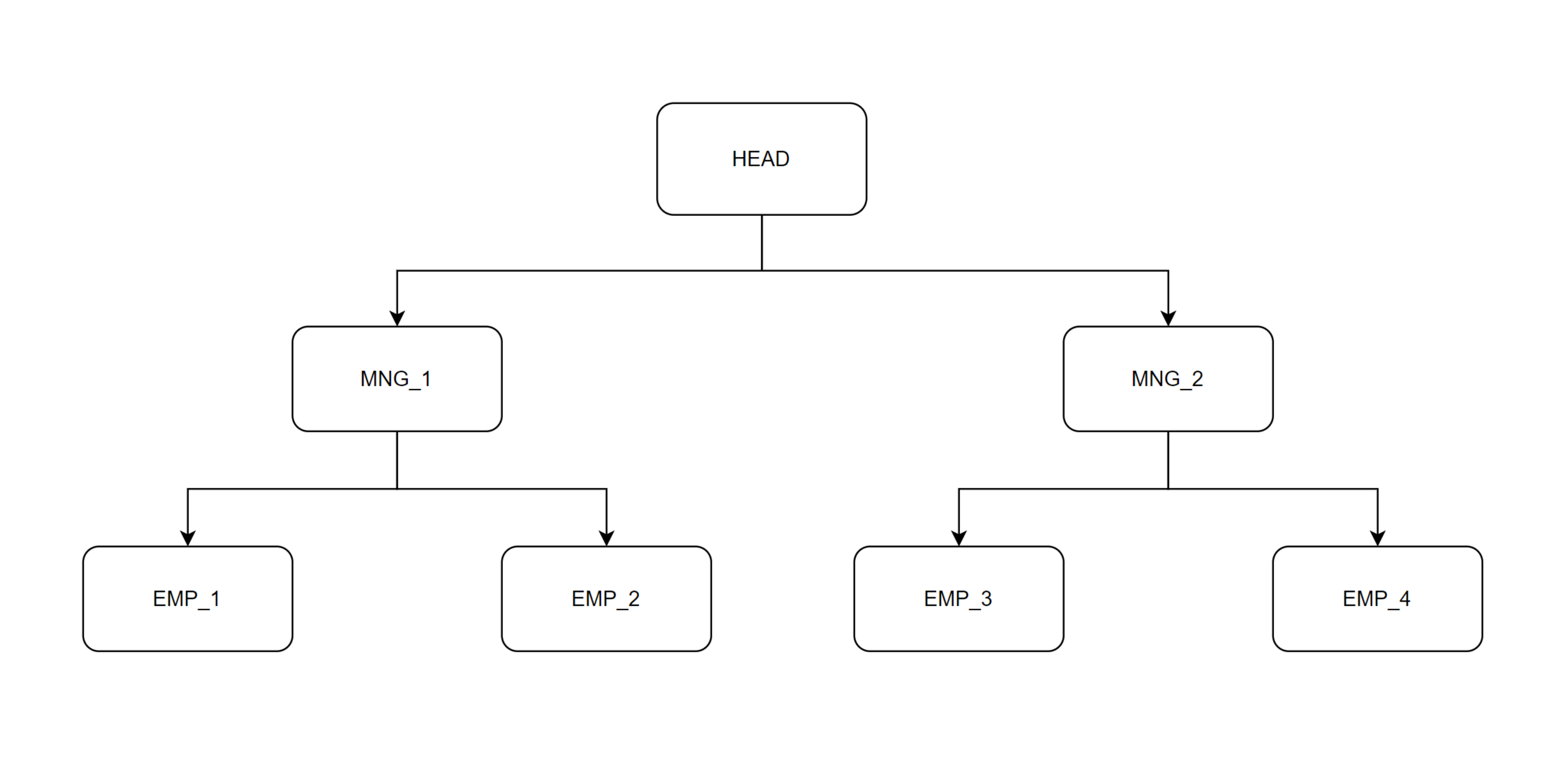
Proposed System

- A 3-nary enterprise system which extensible to n-nary system.
- Used Twofish algorithm (fastest symmetric algorithm known).
- For high security, using OTP protected file so that either the employee who have written it or the head of the employee can read it.
- Using some fast traditional cryptographic algorithm during the input to increase security with out loss of speed.

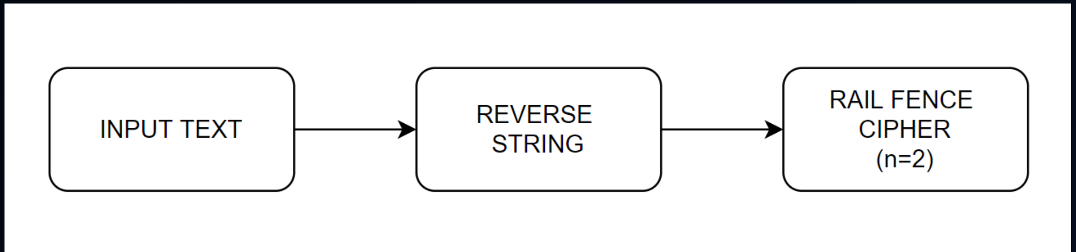
System Design



File System Hierarchy



C1 Encryption Process



- The inputted string will be first be reversed.
- Then we will use the basic rail fence cipher to misplace the letters.
- This small process will increase our security without compromising speed of the process.

Why TwoFish Algorithm?

- Details:

128-bit symmetric cipher

Variable key lengths of 128-192 & 256

16 rounds

Feistel network

- Advantages:

No weak keys

Efficiency

Flexible design

Fastest algorithm

Bruce Schneier:
“But even from a theoretical perspective, Twofish isn't even remotely broken.”

Justification for using TwoFish

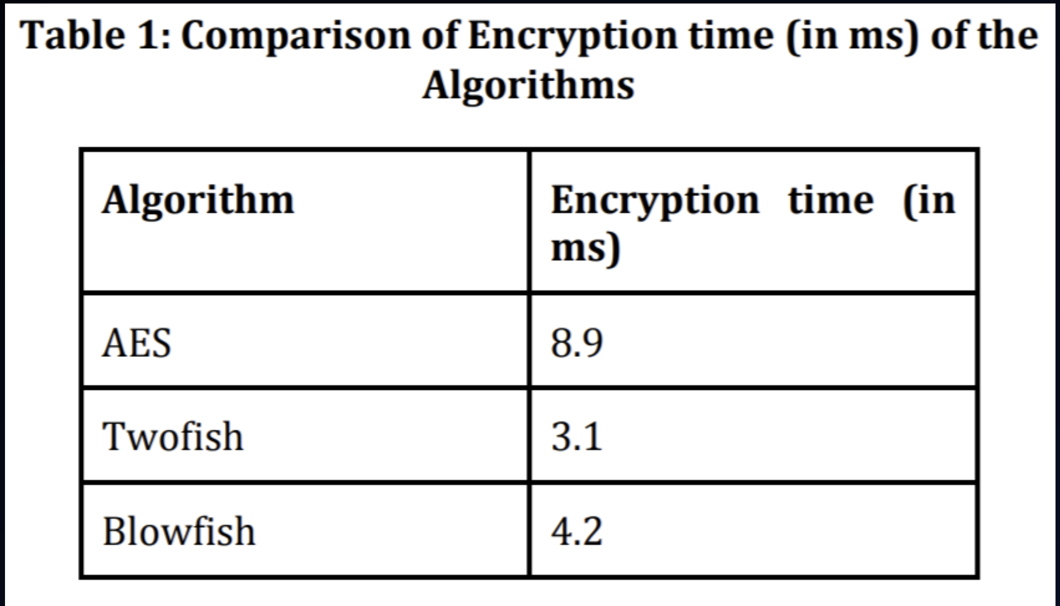


Table 1: Comparison of Encryption time (in ms) of the Algorithms

Algorithm	**Encryption time (in ms)**
AES	8.9
Twofish	3.1
Blowfish	4.2

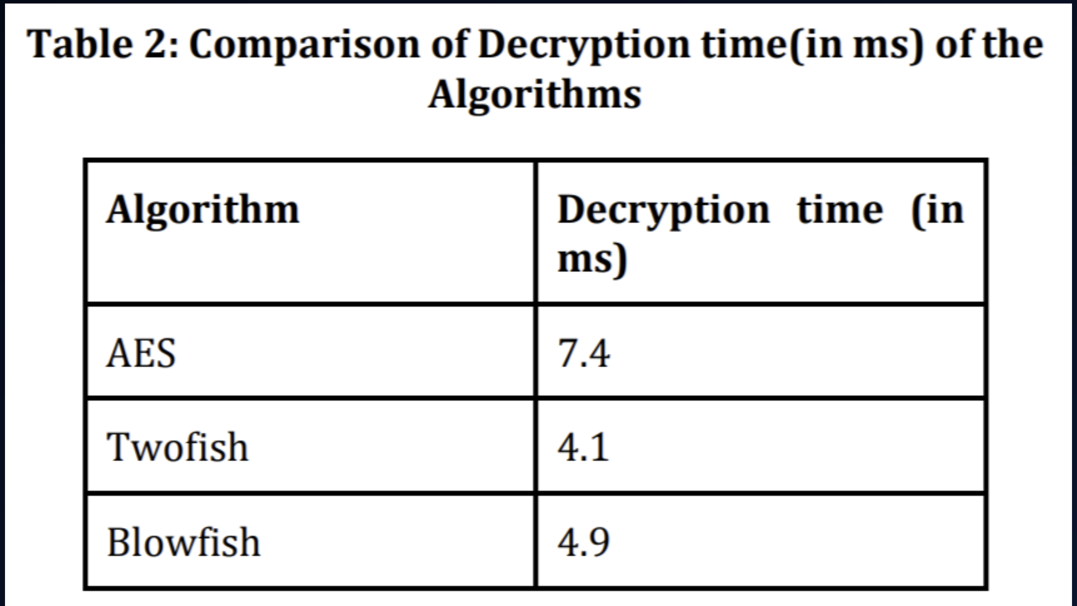


Table 2: Comparison of Decryption time(in ms) of the Algorithms

Algorithm	**Decryption time (in ms)**
AES	7.4
Twofish	4.1
Blowfish	4.9

Cryptanalysis of the encrypted file

- TwoFish algorithm: 2^{256} (256 bit algorithm)
- Rail Fence algorithm: L = length of text, K = possible key, K<L
- Reversing the string will confuse the attacker more. Because even after decrypting rail fence algorithm using brute force method the attacker won't get any meaningful answer in any possible key.
- Rail fence is weak if the small key size is small. But in large company the text is large.
- Final cryptanalysis: $2^{256} \times (L-1)$

Cryptanalysis of the file system

- Our file system is secured using OTP which is shared through SMTP (Simple Mail Transfer Protocol).
- SMTP uses PGP for its security which is not broken till date.
- And PGP is secured using RSA-2048 which is also not broken till date.
- Total Cryptanalysis: 2^{2048}
- For OTP: 62^6 Possible combinations and only 1 try.
- $62^6 = 1.999 \times 10^{13}$

Result and Observations

No of Words	Decryption Time (in seconds)
1000	0.0987
10000	0.689
100000	4.729

Conclusion

- We have formed a extensible file system which uses OTP to protect the file from false use from inside the system.
- And using twofish and other traditional methods we are encrypting the file to preventing the attacker from using the information from other the system where the file is stored.
- Accomplishing the problem, speed with security!