External modifications of Advanced Encryption System

ABSTRACT

The goal of this paper presents a novel approach to enhance the performance and resistance to attacks of the AES algorithm through hybridization using **cascading encryption** and tackle problems in the algorithm which makes it susceptible such as:

* Weak Key generation
* Insufficient Key Lengths
* Weak Random Generator
* Weak Key Exchange Protocols

This paper proposes a scheme that incorporates two famous algorithms, RSA and AES taking advantages of both the algorithms to combat the above-mentioned vulnerabilities.

An Overview of AES:

The Advanced Encryption Standard (AES) is a widely used symmetric-key encryption algorithm designed to secure sensitive data. It was established as the standard encryption algorithm by the U.S. National Institute of Standards and Technology (NIST) in 2001, replacing the Data Encryption Standard (DES). AES is a block cipher, meaning it processes data in fixed-size blocks, and it supports key lengths of 128, 192, and 256 bits. AES works on Bytes of plaintext and has certain number of rounds associated with the key-size.

* 128 bit key – 10 rounds
* 192 bit key – 12 rounds
* 256 bit key – 14 rounds

AES has excellent computation time; encryption and decryption processes are relatively fast compared to some other encryption algorithms. The low computational overhead makes it suitable for real-time applications and scenarios with resource constraints. This means that it can be implemented with other strong algorithms which have a rather slow computation speed.

Each round of AES is comprised of four steps in total:

* SubBytes
* ShiftRows
* MixColumns
* Add Round Key

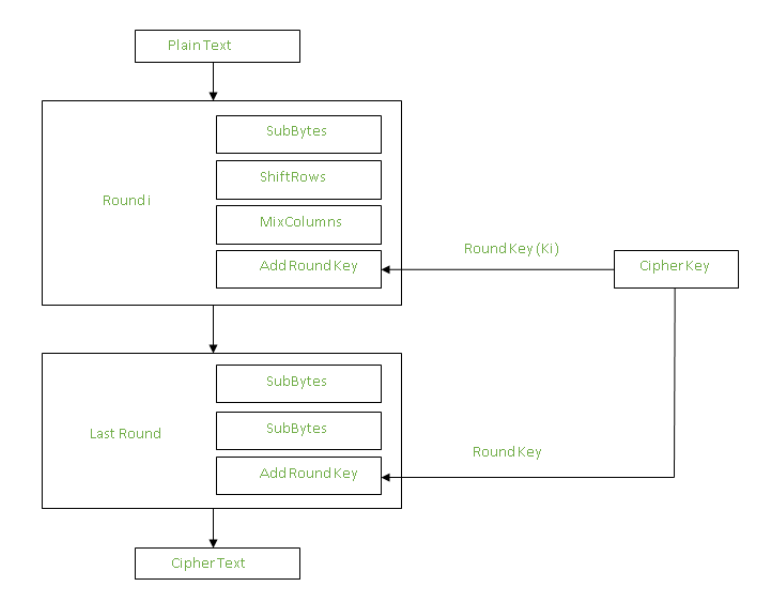
These four steps contribute to the confusion and diffusion of the algorithm.

1. Confusion: Confusion involves making the relationship between the ciphertext and the key as complex and unpredictable as possible. In other words, changing one bit of the key or plaintext should ideally affect many bits in the ciphertext. This can be improved by modifying SubBytes and Add Round Key.

In SubBytes we can improve the implementation of S-boxes by adding computations(using non-linear functions) or a larger size box.

1. Diffusion: Diffusion involves spreading the influence of a single input or key bit across many output or ciphertext bits. The objective is to ensure that each bit of the output depends on multiple bits of the input.

All the four steps occur in a single round, except the last round which does not contain MixColumns.



An Overview of RSA Key Generation:

RSA algorithm’s strength lies in its Key Generation which depends on difficulty of factoring the product of two large prime numbers. This ensures its security but results in slow computation speeds during encryption and decryption. As key length increases for better security, computational overhead also increases, making key generation, encryption, and decryption more computationally intensive.

RSA’s Key Generation steps are as follows:

1. Select two big prime numbers p and q. The strength of the key generation lies in the size of the numbers.
2. Compute n=p\*q and finding Euler Totient function of n=(p-1)\*(q-1)
3. Deciding a public key e: choosing e over a range of numbers which are coprime to the Euler Totient function of n. This key is public and is used for encryption.
4. Getting a respective private key d: d=1\*(e)^-1 mod Euler Function of n

RSA encryption and decryption:

1. ciphertext C ≡ M^e (mod n) where M is the plaintext in numeric form
2. Plaintext M ≡ C^d (mod n)

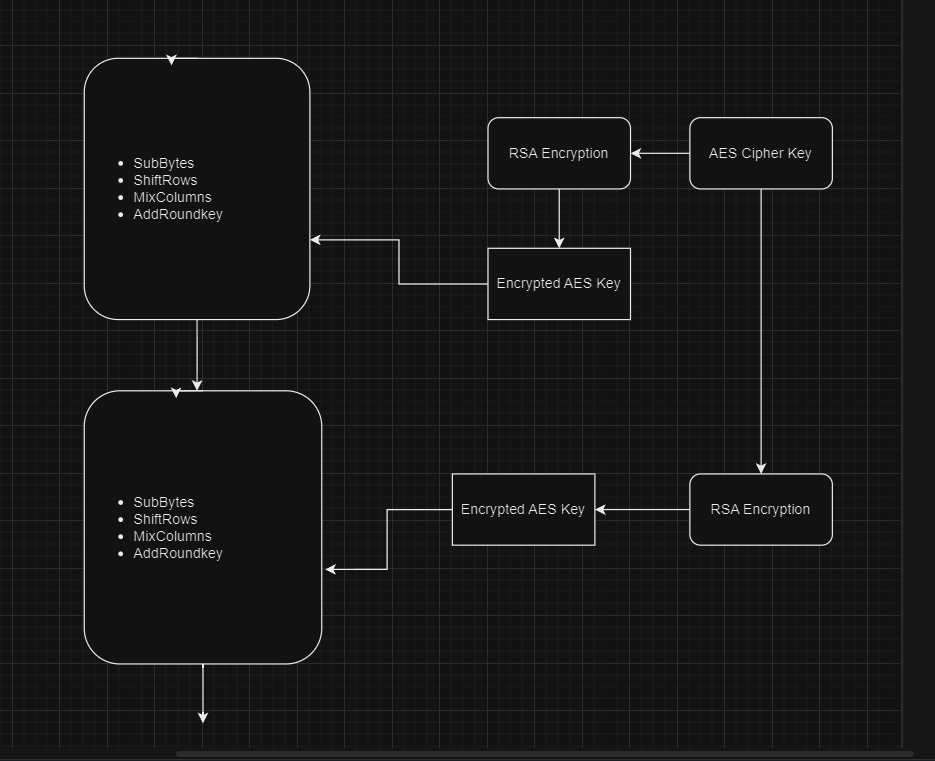
AES and RSA Hybrid Algorithm Key generation

For Encryption:

1. Generate RSA Key Pair (public(e,n), private(d,n))
2. Generate AES Key
3. Encrypt AES Key with RSA public key and use it in AddRoundKey Step.
4. Encrypt the actual data with the encrypted AES key

For Decryption:

Decrypt the corresponding AES keys (as it is symmetric in nature) Following the four steps of AES, decrypt the actual data.



Analysis and results:

Below is the graph for the decryption/Encryption time with a plain text of size:

* 32KB
* 64KB
* 128KB
* 256KB
* 512KB
* 1024KB
* 2048KB
* 4096KB.

A graph with numbers and a line

Description automatically generated

A graph with numbers and a line

Description automatically generated

References:

1. [Vol 9\_1\_1.pdf (scienpress.com)](https://www.scienpress.com/Upload/JAMB/Vol%209_1_1.pdf)- Modification and customization of cryptographic algorithms-George Marinakisnm
2. <https://www.geeksforgeeks.org/advanced-encryption-standard-aes/>
3. <https://www.geeksforgeeks.org/advanced-encryption-standard-aes/>
4. <https://www.sciencegate.app/app/document#/10.4028/www.scientific.net/amr.989-994.1861/related-documents> -Research on Improved AES algorithm