Trabalho de Segurança Computacional

Luis Fernando Lamellas - 19/0016841
Universidade de Brasília

***Abstract*—O projeto tem como objetivo descrever o processo de implementação de diversas funções e algoritmos empregados na cifração de mensagens, visando a aquisição de conhecimento sobre criptografia.**

I. Introdução

O Advanced Encryption Standard (AES) é um algoritmo de criptografia simétrica, ou seja, utiliza a mesma chave para cifrar e decifrar dados. Sendo um marco na segurança da informação, é empregado para proteger comunicações digitais, arquivos e transações online. O AES substituiu o Data Encryption Standard (DES) devido à necessidade de um padrão mais robusto e seguro.

O algoritmo opera em blocos de dados e suporta chaves de 128, 192 e 256 bits, oferecendo flexibilidade na escolha do nível de segurança conforme a aplicação. Suas operações fundamentais, como SubBytes, ShiftRows, MixColumns e AddRoundKey, aplicadas em várias rodadas, conferem-lhe uma resistência notável contra diversos ataques cibernéticos.

Neste estudo, é proposto o desenvolvimento do Advanced Encryption Standard (AES) sem o auxílio de bibliotecas de criptografia convencionais. Essa abordagem proporcionará uma compreensão aprofundada das operações fundamentais do AES. Além disso, também serão implementados o modo de operação ECB(Eletronic Codebook) e CTR(Counter).

II. Desenvolvimento

A. Advanced Encryption Standard

1) state-from-bytes(): A função aceita uma sequência de bytes (data) e organiza esses bytes em uma lista de listas, representando assim a matriz do estado do AES. Cada sublista contém 4 bytes, e o número total de sublistas é determinado pela divisão do comprimento total dos bytes por 4.

2) key-expansion(): No AES, a chave original fornecida pelo usuário é expandida para gerar um conjunto de subchaves que serão utilizadas nas várias rodadas do algoritmo.

A expansão de chave envolve a geração de subchaves adicionais a partir da chave mestra inicial. Essas subchaves são utilizadas nas diversas etapas do processo de cifragem, garantindo uma variação dinâmica e complexa ao longo das rodadas. A key expansion é realizada através de uma série de operações, incluindo substituições de bytes, operações de permutação e a aplicação de funções não-lineares.

Essa abordagem aumenta a resistência do sistema a ataques criptoanalíticos, garantindo que a chave original seja distribuída de maneira mais extensa e eficaz ao longo do processo de cifragem.

3) add-round-key(): Essa operação ocorre em cada rodada do algoritmo e consiste na combinação bitwise (ou exclusivo, XOR) do estado atual do bloco de texto com a subchave correspondente àquela rodada. Tal combinação introduz a influência da chave na transformação do bloco de texto, sendo crucial para a confusão e difusão características da criptografia moderna.

4) sub-bytes(): Nessa etapa, cada byte do bloco de texto é substituído por outro byte de acordo com uma tabela de substituição predefinida, chamada de S-Box.

O S-Box é uma matriz de substituição não linear que introduz não apenas confusão, mas também aumenta a resistência do algoritmo a ataques criptoanalíticos. A substituição é realizada de forma independente para cada byte no bloco de texto, proporcionando uma transformação não linear que contribui para a segurança global do sistema.

5) shift-rows(): Nessa etapa, os bytes nas linhas da matriz do bloco de texto são deslocados ciclicamente para a esquerda, com a quantidade de deslocamento dependendo do número da linha.

Esse deslocamento confere uma propriedade de difusão ao algoritmo, garantindo que alterações em um byte do bloco de entrada afetem múltiplos bytes no bloco de saída. Contribuindo para a dispersão de informações ao longo do bloco de texto, aumentando a complexidade do processo de cifragem.

6) mix-columns(): Essa operação é aplicada a cada coluna da matriz do bloco de texto e envolve transformações lineares que misturam os bytes dentro de cada coluna.

Cada byte em uma coluna é multiplicado por uma constante pré-definida e, em seguida, a soma desses produtos é calculada. Essa soma resulta em um novo valor que substitui o byte original na coluna. A função mix-columns() contribui para a difusão e a confusão, garantindo que alterações em um byte do bloco de entrada afetem múltiplos bytes no bloco de saída.

B. Modo de Operação CTR

O modo CTR é uma forma de operar o algoritmo AES (Advanced Encryption Standard) para criptografar dados de forma eficiente. Em vez de criptografar blocos de dados diretamente, o modo CTR utiliza um contador, que é criptografado e combinado com os dados originais por meio de uma operação de ou exclusivo (XOR).

A ideia é gerar uma sequência de blocos de chave pseudoaleatórios usando o contador como entrada para a cifra AES. Esses blocos pseudoaleatórios são então combinados com os blocos de dados originais usando a operação XOR para produzir o texto cifrado. O contador é incrementado para cada

bloco de dados, garantindo que cada bloco de texto cifrado seja diferente, mesmo que os dados originais se repitam.

O modo CTR é popular porque permite a paralelização da operação de criptografia e descriptografia, o que o torna eficiente em termos de desempenho. Além disso, oferece boa segurança quando usado corretamente.

III. Resultados

A. AES Encryption

O modo de operação ECB (Electronic Codebook) do AES cifra cada bloco de texto de forma independente. Para este estudo foi utilizado como dado a ser cifrado a string ”AES - Advanced Encryption Standard”. Para criptografá-lo através da implementação do AES desenvolvida, foi utilizada a chave ”aesEncryptionKey”. Como resultado do ciframento obteve-se a seguinte string em hexadecimal: ”b5346f6c76 ab583c7318 cfe398ce1f ea6b52e9e3 02b5a9a9f7 44a9784339 92f89c4b36 7fabcf7ce9 43491bb9b4 6baa1f”

A fim de verificar o resultado encontrado, foi utilizada uma biblioteca de criptografia para cifrar a mesma string com a mesma chave. O resutlado encontrado foi idêntico, sugerindo que a implementação do algoritmo foi correta.

B. Modo de Operação CTR

A mesma estratégia de verificação foi usada para o modo de operação CTR. Além da mesma string de entrada e da mesma chave, foi escolhido um contador incial com o valor de ”2623891”. Após realizada a transformação em cifra, obteve-se a seguinte string em hexadecimal: ”b550bd470a 5f7c5dd6bc fb0bc74351 93efa5746b 539ff9496c 128204858d 0f92a8e23a 5a83cb4915 2375d43c7a 297ddd”

Também foi aferido um resultado positivo quando contraposto com a cifração do algoritmo da biblioteca, sugerindo uma implementação correta desse modo de cifração.

IV. Conclusão

O AES desempenha um papel crucial na segurança da informação, sendo amplamente reconhecido como um algoritmo robusto e confiável para a cifragem de dados sensíveis. No entanto, é importante destacar que o AES, por si só, não é suficiente para garantir a segurança completa em todas as situações.

A importância do AES reside na sua capacidade de fornecer uma cifragem forte e eficaz, resistente a uma variedade de ataques criptoanalíticos conhecidos. No entanto, a escolha do modo de operação é igualmente vital para garantir a segurança geral do sistema. A combinação correta do algoritmo AES com os modos de operação apropriados contribui para a construção de sistemas criptográficos robustos e resilientes.