Trabalho Segurança Computacional

(Relatório AES-128 modo ECB e CTR)

Gustavo Vieira de Araújo - 211068440

02 de setembro de 2024

Professor: João José Costa Gondim

INTRODUÇÃO

O relatório em questão, descreve a implementação do algoritmo de criptografia simétrica AES-128 em Python, o qual possui a cifração e decifração de dados. Desse modo, é utilizado uma estrutura de matrizes 4x4 para manipular o estado do texto ao longo das operações do AES, empregando o modo de operação CTR.

DESCRIÇÃO DAS CIFRAS, MODO DE OPERAÇÃO E OPERAÇÕES IMPLEMENTADAS

AES-128 (ECB):

O AES-128 é um algoritmo de criptografia simétrica que opera em blocos de 128 bits, com chaves de 128 bits e máximo de 10 rodadas. Diante disso, este algoritmo realiza múltiplas rodadas de substituições, permutações e operações aritméticas, aplicadas sobre os blocos de dados em matrizes 4x4, conhecidas como estados, a fim de que no final tenhamos a cifra do bloco inicial.

Modo de Operação CTR (Counter):

O modo de operação CTR estende o AES, transformando ele em um cifrador de fluxo, o qual o contador é cifrado e combinado com o bloco de 128 bits inicial para produzir o texto cifrado. Esse modo permite operações paralelas, bem como garante que cada bloco seja tratado independentemente, logo tratar grandes volumes de dados com esse algoritmo é mais eficiente (leva menos tempo).

Operações Implementadas:

As operações básicas do AES implementadas nos códigos são:

- **SubBytes:** Esta operação substitui cada byte da matriz de estado por um byte correspondente da S-Box, introduzindo não linearidade no algoritmo.

- **ShiftRows:** A segunda linha da matriz de estado é deslocada uma posição para a esquerda, a terceira linha duas posições, e a quarta linha três posições, contribuindo para a difusão dos dados.
- **MixColumns:** Cada coluna da matriz de estado é transformada usando operações aritméticas sobre os bytes da coluna, espalhando a influência dos bytes sobre toda a matriz.
- **AddRoundKey:** Esta operação aplica uma chave de rodada ao estado, utilizando uma operação XOR entre a matriz de estado e a chave derivada da chave principal.

Além disso, existem outras funções de auxílio na implementação, que para fins de objetividade não serão citadas aqui, uma vez que o objetivo delas na maior parte dos casos são conversões, manipulações de objetos, visualização e etc, além disso essas funções estão bem comentadas e descritas no código de implementação.

DESCRIÇÃO DAS IMPLEMENTAÇÕES

Implementação da Cifração (AES_cifracao.py):

O código de cifração começa com a função “obter_matrizes_do_texto”, que converte o texto de entrada em uma ou mais matrizes 4x4. Isso é feito segmentando o texto em blocos de 16 bytes e populando uma matriz com esses bytes.

A função “cifrar_texto”implementa as operações do AES-128 (ECB) descritas anteriormente. Ela aplica repetidamente as operações de SubBytes, ShiftRows, MixColumns, e AddRoundKey para cada rodada, com exceção da última, onde a MixColumns não é aplicada.

Adicionalmente, o código integra o modo CTR, gerando um contador inicial, que é incrementado a cada bloco de texto, cifrado e combinado com o bloco correspondente do texto original.

Implementação da Decifração (AES_decifracao.py):

O código de decifração segue uma estrutura similar ao de cifração, começando com a função “obter_matrizes_estado”, que reconstroi as matrizes de estado a partir do texto cifrado.

A função “decifrar_texto” reverte o processo de cifração, aplicando as operações inversas das rodadas do AES. Ela recupera o texto original, processando o contador cifrado e revertendo os efeitos das operações SubBytes, ShiftRows, e MixColumns, seguindo a ordem inversa.

A integridade e a corretude da decifração são garantidas pelo uso correto da chave e do contador, assegurando que o texto decifrado corresponda exatamente ao texto original antes da cifração.

DESCRIÇÃO DO ARQUIVO DE TESTES

O arquivo “testes” foi projetado para validar a implementação manual do AES em modo ECB e CTR, comparando seus resultados com aqueles produzidos pela biblioteca “pycryptodome”.

Testes de Cifração:

- **Teste da Implementação Manual (ECB):** Aqui, a função “criptografar_texto” é utilizada para cifrar um texto utilizando 10 rodadas. O resultado é comparado com a saída gerada pela função equivalente da biblioteca “pycryptodome”.
- **Teste da Implementação da Biblioteca (ECB):** Esse teste utiliza a função encrypt do módulo AES da biblioteca “pycryptodome” para cifrar o mesmo texto e chave. Os resultados são comparados com os obtidos manualmente.
- **Teste da Implementação Manual (CTR):** No modo CTR, o texto é cifrado usando a função “criptografar_texto_ctr”, que incorpora um contador. O número de rodadas é configurável, e o resultado é comparado com a cifra da biblioteca.
- **Teste da Implementação da Biblioteca (CTR):** A biblioteca “pycryptodome” é utilizada para cifrar o texto no modo CTR, e o resultado é comparado com a implementação manual.

Testes de Decifração:

- **Teste da Implementação Manual (ECB):** A função “decifrar_matriz” é utilizada para reverter a cifra ECB manualmente gerada. O resultado é comparado com a saída da biblioteca.
- **Teste da Implementação da Biblioteca (ECB):** A função decrypt da biblioteca “pycryptodome” é usada para decifrar um bloco previamente cifrado no modo ECB. A comparação é feita com a saída da implementação manual.

- **Teste da Implementação Manual (CTR):** A função “decifrar_texto_ctr” reverte o processo de cifração no modo CTR, utilizando a mesma chave e contador para verificar a consistência dos resultados.
- **Teste da Implementação da Biblioteca (CTR):** A função decrypt da biblioteca “pycryptodome” é usada para decifrar o texto cifrado no modo CTR. Novamente, os resultados são comparados com a implementação manual.

CONCLUSÃO

A implementação do AES no modo ECB e CTR realizada nos arquivos “AES_cifracao.py” e “AES_decifracao.py” demonstra uma aplicação prática primitiva desse algoritmo, o qual segue os padrões solicitados e realiza a criptografia/descriptografia como solicitado. Ademais, os testes demonstram que a implementação manual do AES, tanto no modo ECB quanto no modo CTR, produzem resultados consistentes com a biblioteca “pycryptodome”, logo válida a corretude das operações de cifração e decifração implementadas.

REFERÊNCIAS

Informações sobre modo ECB e CTR:

https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#ECB

https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#CTR

Documentação Numpy e Pycryptodome:

https://numpy.org/doc/

https://www.pycryptodome.org/

Informações sobre o AES-128:

https://pt.wikipedia.org/wiki/Advanced_Encryption_Standard

https://pt.stackoverflow.com/questions/43492/como-funciona-o-algoritmo-de-criptografia-aes