Universidade de Brasília
Ciência da computação
Segurança Computacional

Cifra de Bloco e modo operação CTR

Talles Marcelo de Mesquita Caldeira Mat:20/0060295

4 de novembro de 2023

1 Pré-requisitos

1. Versão do Python 3.x.x >=
2. Acesso ao prompt de comando(cmd)

2 Introdução

A segurança da informação é altamente dependente da criptografia, uma disciplina imprescindível para resguardar informações confidenciais de quaisquer acessos não autorizados. O algoritmo AES, devido à sua eficiência e confiabilidade, é amplamente empregado na proteção de dados. Porém, é importante destacar que a segurança dos dados criptografados está diretamente influenciada pela qualidade da implementação realizada. Dito isso, devemos analisar que o código desenvolvido está embasado em uma implementação idealizada e não deverá ser levado em consideração para ambiente de produção ou algo do tipo, ou seja, somente para fins educacionais.

2.1 O que é Cifra de Bloco e modo operação CTR?

Uma cifra de bloco é um tipo de algoritmo de criptografia simétrica que opera em blocos fixos de dados de tamanho predeterminado. Cada bloco de dados é cifrado e decifrado de forma independente. A cifra de bloco mais conhecida e amplamente utilizada é o AES (Advanced Encryption Standard), que opera em blocos de 128 bits. Outras cifras de bloco incluem DES (Data Encryption Standard) e Triple DES.

O modo de operação CTR (Counter Mode) é um dos modos de operação utilizados com cifras de bloco, como o AES, para cifrar dados de forma segura. Neste modo, em vez de cifrar diretamente os dados originais, os dados são combinados com um valor chamado contador (counter) para gerar um keystream (fluxo de chaves), que é usado para cifrar os dados.

3 Desenvolvimento

3.1 Tabelas S-box e Round Constant

A tabela S-box é uma tabela de substituição utilizada nas etapas de substituição do AES. Ela é responsável por substituir bytes de entrada por bytes de saída durante a criptografia. A tabela S-box é uma matriz 16x16, onde cada elemento é um valor hexadecimal de 8 bits (um byte). Ela é usada nas etapas de SubBytes tanto na criptografia quanto na descriptografia. Por outro lado, a tabela de Round Constants é usada no processo de expansão da chave para gerar chaves de rodada adicionais usadas durante a criptografia

3.2 Geração de Chave

Idealmente a chave secreta seria gerada de acordo com o tamanho escolhido pelo usuário, entretanto por questões de tempo hábil não foi possível implementar de tal forma pois o mesmo apresentava muitos bugs. A chave poderia ter comprimentos diferentes, mas os tamanhos mais comuns são 128, 192 ou 256 bits. Dependendo do tamanho da chave, o AES escolherá o número apropriado de rodadas (10, 12 ou 14).

3.3 Expensão da chave

A chave de entrada é expandida para gerar um conjunto de chaves de rodada. Isso é feito usando a função key-expansion. A função key-expansion cria várias chaves intermediárias com base na chave de entrada. Essas chaves intermediárias são usadas nas várias rodadas do AES.

As chaves intermediárias geradas na etapa anterior são usadas em cada rodada do AES. O número de rodadas é determinado pelo tamanho da chave: 10 rodadas para chave de 128 bits, 12 rodadas para chave de 192 bits e 14 rodadas para chave de 256 bits. As chaves intermediárias são adicionadas às porções de dados durante a criptografia e subtraídas durante a descriptografia.

3.4 Criptografia e Descriptografia

Com as chaves de rodada geradas, o algoritmo AES é aplicado aos blocos de dados durante a criptografia e descriptografia. Cada rodada do AES envolve várias etapas, incluindo substituição de bytes, transposição de linhas, mistura de colunas e adição de chaves de rodada.

3.5 Substituição de Bytes (SubBytes)

A etapa de substituição de bytes é a primeira camada de transformação do AES. Nesta etapa, cada byte dos dados de entrada é substituído por outro byte de acordo com uma tabela de substituição chamada S-box. Essa substituição é feita de maneira não linear, introduzindo confusão nos dados. A S-box é uma tabela fixa que é usada tanto na criptografia quanto na descriptografia. Ela ajuda a proteger contra ataques estatísticos e criptanálise.

3.6 Transposição de Linhas (ShiftRows)

Após a substituição de bytes, a etapa de transposição de linhas reorganiza os bytes nos blocos de dados. Cada linha da matriz de dados é deslocada para a esquerda em um número variável de posições. Isso garante que os dados sejam misturados e dispersos por toda a matriz, tornando mais difícil identificar padrões e melhorando a difusão dos dados.

3.7 Mistura de Colunas (MixColumns) - Tive bastante dificuldade

A etapa de mistura de colunas opera nas colunas da matriz de dados. Ela é responsável por combinar e confundir os bytes dentro de cada coluna, introduzindo ainda mais difusão nos dados. Isso é feito por meio de uma operação de multiplicação em um campo finito (GF(2elevado8)), onde os bytes são tratados como elementos desse campo. A mistura de colunas ajuda a garantir que as mudanças em um byte afetem todos os outros bytes da coluna.

3.8 Adição de Chaves de Rodada (AddRoundKey)

A etapa de adição de chaves de rodada é fundamental no processo de criptografia e descriptografia. Em cada rodada (exceto a primeira e a última), a chave de rodada gerada a partir da chave principal é combinada com os dados de entrada por meio de uma operação de ou exclusivo (XOR). Isso introduz a chave atual na transformação dos dados e é crucial para garantir que cada rodada seja única e dependa da chave de rodada correspondente. A adição de chaves de rodada também contribui para a confusão dos dados.

Essas quatro etapas (Substituição de Bytes, Transposição de Linhas, Mistura de Colunas e Adição de Chaves de Rodada) são repetidas em cada rodada do AES, exceto a última, onde a etapa de Mistura de Colunas é omitida na criptografia e invertida na descriptografia. Juntas, essas operações criam uma criptografia robusta e eficaz que protege os dados contra ataques e preserva a confidencialidade das informações durante a comunicação ou armazenamento.

4 Conclusão

Em suma, o código apresentado implementa o algoritmo AES (Advanced Encryption Standard), que é amplamente utilizado para criptografia de dados e é considerado altamente seguro quando implementado corretamente. O algoritmo é composto por várias etapas, incluindo Substituição de Bytes, Transposição de Linhas, Mistura de Colunas e Adição de Chaves de Rodada, que juntas garantem uma forte confidencialidade dos dados.

No entanto, a implementação completa do algoritmo AES pode ser desafiadora devido a várias razões:

Complexidade Matemática: O AES envolve operações matemáticas complexas, como a multiplicação em um campo finito (GF(2elevado8)), que requerem um entendimento profundo da álgebra modular. Erros na implementação matemática podem comprometer a segurança do algoritmo.

Gerenciamento de Chaves: A geração e o gerenciamento adequados de chaves são fundamentais para a segurança do AES. Isso inclui a geração segura de chaves, o armazenamento protegido das chaves e a garantia de que as chaves sejam mantidas em sigilo.

Proteção Contra Ataques: O AES deve ser implementado com considerações de segurança em mente para proteger contra ataques, como ataques de força bruta, ataques de análise diferencial e outros métodos criptoanalíticos.

Dado tal complexidade torna-se um desafio, entretanto para dados didaticos foi concebido de tal forma os aspectos necessarios a implementação