UNIVERSIDADE VEIGA DE ALMEIDA

DOUGLAS DA SILVA VEGA | 1220203098
LUIZ MENDES BARBOSA | 1220107933
JULIANO ALFREDO | 1220104685
MARCOS GABRIEL SERAFIM TEIXEIRA | 1220300532
PEDRO HENRIQUE BERARDO MONTEIRO | 1220303469

PROJETO DE EXTENSÃO VII

RIO DE JANEIRO
2025

DOUGLAS DA SILVA VEGA | 1220203098
LUIZ MENDES BARBOSA | 1220107933
JULIANO ALFREDO | 1220104685
MARCOS GABRIEL SERAFIM TEIXEIRA | 1220300532
PEDRO HENRIQUE BERARDO MONTEIRO | 1220303469

PROJETO DE EXTENSÃO VII

Trabalho apresentado como requisito para obtenção de nota na disciplina Projeto de Extensão VII, da Universidade Veiga de Almeida.

Orientador: Fabio Contarini Carneiro

RIO DE JANEIRO
2025

Resumo

Este trabalho apresenta o desenvolvimento de um algoritmo de cifra de blocos simétrica, com foco na proteção de dados sensíveis, atendendo às necessidades propostas pela empresa fictícia Inn Seguros. O algoritmo foi implementado em linguagem Python, sem o uso de bibliotecas externas, e opera sobre blocos de 32 bits utilizando uma chave de 32 bits. A estrutura do algoritmo é composta por três rodadas, cada uma realizando operações de substituição e permutação baseadas em subchaves derivadas da chave principal.

A substituição é realizada por meio de uma operação XOR entre o bloco e a subchave da rodada, enquanto a permutação inverte as metades do bloco, ambas sendo operações reversíveis, o que facilita o processo de descriptografia. O sistema permite criptografar e descriptografar qualquer tipo de arquivo binário, mantendo a simetria entre os processos.

Os testes realizados demonstraram um efeito avalanche eficiente, evidenciando que pequenas alterações na chave ou nos dados de entrada geram grandes variações no texto cifrado. O projeto contribuiu significativamente para o entendimento prático dos conceitos de segurança da informação e criptografia de blocos, além de oferecer uma solução funcional e didática para ambientes controlados.

Palavras-chave: criptografia, cifra de blocos, segurança da informação, efeito avalanche, subchave.

Sumário

1. Introdução

A segurança da informação é um dos pilares fundamentais para empresas que lidam com dados sensíveis, como é o caso da Inn Seguros. Com o aumento exponencial das ameaças cibernéticas e a sofisticação de ataques, tornou-se imprescindível que corporações desenvolvam mecanismos próprios para proteger seus ativos digitais. Nesse contexto, algoritmos de cifra de blocos se destacam como uma solução eficaz para garantir a confidencialidade de informações.

Este trabalho tem como objetivo desenvolver um algoritmo de criptografia simétrica por blocos, utilizando operações simples e reversíveis, capaz de criptografar e descriptografar arquivos de forma segura, eficiente e sem dependência de bibliotecas externas. O algoritmo foi projetado especificamente para atender às necessidades da Inn Seguros, focando na proteção de registros de contratos, dados pessoais e relatórios de sinistros.

2. Justificativa

A decisão de desenvolver um algoritmo de cifra de blocos próprio se justifica por múltiplas razões estratégicas e acadêmicas. Do ponto de vista institucional, a criação de uma solução interna reduz a exposição da empresa a vulnerabilidades de implementações públicas e permite maior controle sobre os mecanismos de segurança utilizados. Além disso, a autonomia na criptografia permite a adaptação do sistema de acordo com as políticas específicas de governança de dados da empresa.

Do ponto de vista técnico-pedagógico, este projeto proporcionou aos desenvolvedores a oportunidade de aplicar conceitos teóricos de criptografia, como substituição, permutação, simetria e efeito avalanche, em um ambiente prático. O algoritmo implementado, utiliza chaves e blocos de 32 bits ao longo de três rodadas, demonstrando um comportamento robusto, com forte difusão e capacidade de mascaramento de padrões, atendendo assim aos critérios fundamentais de segurança exigidos na criptografia moderna.

3. Descrição do Algoritmo

O algoritmo desenvolvido foi escrito em Python puro, sem uso de bibliotecas externas, o que garante sua independência e portabilidade. O código é dividido em funções específicas, cada uma responsável por uma parte do processo de criptografia ou descriptografia. O algoritmo funciona sobre blocos de 32 bits (4 bytes) e utiliza uma chave de 32 bits, com três rodadas de processamento, cada uma com substituição e permutação.

3.1. Derivação de Subchaves

A função gerar_subchaves(chave) aplica rotações circulares à chave principal para gerar três subchaves diferentes. Cada subchave será usada em uma das três rodadas de cifragem. O objetivo é introduzir variação em cada etapa, promovendo confusão e difusão.

```
sub = ((chave << (i + 1)) | (chave >> (32 - (i + 1)))) & 0xFFFFFFFF
```

Essa operação garante que cada rodada use uma subchave distinta e relacionada com a chave principal.

3.2. Operação de Substituição

A substituição é feita com uma operação de XOR entre o bloco de dados e a subchave da rodada. Essa operação é simples, rápida e reversível, o que facilita o processo de descriptografia e sendo fundamental para mascarar os bits originais.

3.3. Operação de Permutação

A permutação troca as duas metades do bloco de 32 bits (ou seja, os 16 bits mais significativos com os 16 menos significativos). Essa técnica garante difusão, espalhando os efeitos da substituição ao longo de todo o bloco.

```
esquerda = (bloco >> 16) & 0xFFFF
direita = bloco & 0xFFFF
bloco = (direita << 16) | esquerda
```

Essa permutação também é auto-inversa, o que simplifica a reversão no processo de decriptografia.

3.4. Processo de Criptografia e Decriptografia

O bloco passa por 3 rodadas onde são aplicadas, sequencialmente, a substituição e a permutação. No processo de decriptografia, aplica-se as mesmas operações em ordem inversa.

3.5. Criptografia de Bloco

A função criptografar_bloco(bloco, chave) aplica as três rodadas do algoritmo sobre um único bloco de 32 bits. Em cada rodada, realiza-se a substituição com a subchave derivada, seguida de uma permutação. O bloco resultante de uma rodada serve como entrada para a próxima.

3.6. Descriptografia de Bloco

A função decriptografar_bloco(bloco, chave) inverte o processo da criptografia, utilizando a mesma lógica de subchaves, mas aplicadas em ordem inversa. Como tanto a substituição via XOR quanto a permutação de metades são operações reversíveis, o bloco original pode ser perfeitamente recuperado.

3.7. Processamento de Arquivo

A função processar_arquivo(nome_entrada, nome_saida, chave, modo) realiza a leitura do arquivo de entrada em modo binário, divide os dados em blocos de 4 bytes (32 bits), aplica a cifra definida e grava o resultado no arquivo de saída. Quando em modo decriptografar, o algoritmo também remove o padding adicionado no final do último bloco.

3.8. Interface de Execução

A função menu() oferece uma interface de terminal simples, onde o usuário pode escolher a operação (criptografar ou decriptografar), indicar os nomes dos arquivos de entrada e saída, e fornecer uma chave de 32 bits no formato hexadecimal.

3.9. Características Gerais do Algoritmo

- Independência de bibliotecas externas.
- Compatível com qualquer tipo de arquivo binário ou textual.
- Simetria entre as operações de criptografia e decriptografia.
- Simplicidade operacional com boa evidência de efeito avalanche.
- Tamanho do bloco: 32 bits.
- Tamanho da chave: 32 bits.
- Número de rodadas: 3.
- Substituição: operação XOR com subchave.
- Permutação: troca das metades do bloco.
- Geração de subchaves: rotação circular da chave original a cada rodada.

4. Efeito Avalanche

Um dos critérios de segurança de um bom algoritmo criptográfico é o chamado efeito avalanche: pequenas alterações na entrada devem provocar grandes mudanças na saída. Nosso algoritmo foi projetado com isso em mente e demonstrou bons resultados nos testes, onde uma mudança de apenas um bit na chave gera uma saída totalmente diferente.

5. Exemplos Práticos com Diferenciação de 1 bit

Entrada: 0x12345678

Chave 1: 0x1A2B3C4D

Saída com a chave 1:

Texto claro: 0x12345678

Após substituição 1: 0x289f6a39

Após permutação 1: 0x6a39289f

Após substituição 2: 0x83245e09

Após permutação 2: 0x5e098324

Após substituição 3: 0x4c3de369

Após permutação 3: 0xe3694c3d

Entrada: 0x12345678

Chave 2: 0x1A2B3C4C (difere por 1 bit)

Saídas com chave 2:

Texto claro: 0x12345678

Após substituição 1: 0x289f6a38

Após permutação 1: 0x6a38289f

Após substituição 2: 0x83245e08

Após permutação 2: 0x5e088324

Após substituição 3: 0x4c3de368

Após permutação 3: 0xe3684c3d

Mesmo com apenas 1 bit diferente na chave, os resultados são completamente distintos, evidenciando o efeito avalanche.

6. Conclusão

O algoritmo desenvolvido atendeu com sucesso às necessidades propostas pela Inn Seguros. Ele foi capaz de realizar tanto a criptografia quanto a decriptografia de forma simétrica, utilizando apenas recursos nativos da linguagem, sem depender de bibliotecas externas.

Os testes realizados demonstraram um efeito avalanche consistente, comprovando a eficácia do modelo. A implementação simples e clara contribuiu para facilitar o entendimento dos conceitos e permitiu aplicar, na prática, os princípios fundamentais da criptografia de blocos.

Além de cumprir sua função como ferramenta de proteção de dados, o projeto se mostrou valioso como exercício de aprendizado, reforçando a importância de entender como os algoritmos de segurança funcionam por dentro e servindo como base sólida para aprofundamentos futuros na área.

7. Referências

STALLINGS, William. Criptografia e segurança de redes: princípios e práticas. 6. ed. São Paulo: Pearson, 2018.

SHANNON, Claude E. Communication theory of secrecy systems. Bell System Technical Journal, v. 28, n. 4, p. 656–715, 1949.

TANENBAUM, Andrew S. Redes de computadores. 5. ed. São Paulo: Pearson, 2011.

KOBLITZ, Ronald. RFC 1321 – The MD5 Message-Digest Algorithm. [S. l.]: Internet Engineering Task Force, abr. 1992. Disponível em: https://datatracker.ietf.org/doc/html/rfc1321. Acesso em: 12 jun. 2025.