Auditoria e segurança da informação Relatório de execução de testes

Nome: Amanda Julia Ferreira, 12211BSI225

Descrição Geral do Trabalho

Este trabalho tem como objetivo implementar um esquema criptográfico simplificado, composto por três funções principais:

GEN(seed): gera uma chave binária a partir de uma semente textual;

ENC(key, mensagem): cifra uma mensagem binária usando a chave;

DEC(key, cifra): decifra a cifra para recuperar a mensagem original. O sistema foi desenvolvido em Python 3.10 e busca atender os critérios de avaliação:

tempo de execução eficiente; baixa chance de chaves equivalentes; boa difusão (efeito avalanche); boa confusão (sensibilidade à seed).

Além disso, o projeto inclui testes para verificar funcionalidade básica, difusão, confusão, equivalência de chaves e desempenho.

1. Geração de Chave (GEN)

A função GEN recebe a seed, aplica SHA-256 e obtém sempre 32 bytes (256 bits). Como a especificação exige uma chave com tamanho $4 \times len(seed)$, o hash é convertido byte a byte em bits e apenas os primeiros bits necessários são usados. Esse procedimento garante o tamanho exato da chave e mantém o caráter determinístico do algoritmo: a mesma seed sempre gera a mesma chave.

2. Geração da Cifra (ENC)

A cifra é produzida em ENC com dois passos: primeiro é feito XOR da mensagem com a chave, criando uma cifra intermediária simples e reversível; em seguida, aplica-se uma difusão reversível por XOR em cadeia, que faz cada bit depender do anterior, espalhando pequenas mudanças pela cifra.

3. Recuperação da Mensagem (DEC)

A função DEC desfaz essas etapas na ordem inversa: primeiro remove a difusão percorrendo o vetor de trás para frente, depois aplica XOR com a chave, recuperando a mensagem original. Além disso, o sistema faz isso trabalhando com listas de bits, mas quando a mensagem é textual, ela também pode ser convertida para bits (8 bits por caractere) antes da cifragem e reconstruída após a decifragem (por não fazer parte das especificações, esses métodos estão na classe de testes).

Foram realizados testes de funcionalidade básica, difusão, confusão, chaves equivalentes e desempenho, com saídas ajustadas para apresentar amostras de mensagem, cifra e decifrado, além de tempos de execução. O resultado demonstra que o esquema é simples, funcional, reversível e adequado para ilustrar conceitos de chave, cifra, difusão e confusão, com execução rápida e resultados reproduzíveis.

==========

TESTE 1: Funcionalidade Básica

==========

Senha: 'auditoria_e_seguranca'

Tamanho chave: 84 bits

Mensagem (amostra 32 bits): 01100111100001001111000100100000

Cifra (amostra 32 bits): 11111110110110001010011000101101

Tempo ENC: 1.00 ms

Decifrado (amostra 32 bits): 01100111100001001111000100100000

Tempo DEC: 0.00 ms

OK: Mensagem recuperada corretamente

==========

TESTE 2: Difusão (Efeito Avalanche)

==========

Alterado: 1 bit da mensagem

Bits alterados na cifra: 20 de 20

Porcentagem: 100.0%

OK: Houve difusão

==========

TESTE 3: Confusão (Mudança na Senha)

==========

Senha 1: 'auditoria_e_seguranca'

Senha 2: 'Auditoria_e_seguranca' (mudou 1 letra)

Bits alterados na cifra: 44 de 84

Porcentagem: 52.4%

OK: Houve confusão

==========

TESTE 4: Chaves Equivalentes

==========

Testadas 23 senhas diferentes

Colisões encontradas: 0

OK: Nenhuma chave equivalente

==========

TESTE 5: Performance

==========

GEN: 0.00 ms

ENC: 1.00 ms

DEC: 0.00 ms

TOTAL: 1.00 ms