Lista de Exercícios: Cifras de Bloco e Hash

Prof. Gabriel Rodrigues Caldas de Aquino

Compilado em:
October 31, 2026

1 Cifra de Bloco

- Explique, com suas palavras, os conceitos de **difusão** e **confusão** segundo Claude Shannon.
- Descreva a **estrutura de Feistel**. Demonstre por que o mesmo algoritmo de Feistel pode ser usado tanto para encriptação quanto para decriptação apenas invertendo a ordem das subchaves K_i. Mostre isso para uma rodada apenas.

2 Calculando uma Rodada da Cifra de Feistel

Considere uma cifra de Feistel com bloco de **8 bits**, dividida em duas metades de 4 bits cada: L_0 (bits mais significativos) e R_0 (bits menos significativos). A função de rodada F recebe 4 bits de entrada e uma subchave K_i, mas, por simplicidade, nesta questão definimos $F(x, K_i) = 1111$ (ou seja, sempre retorna os 4 bits '1', independentemente de x e da subchave).

Utilize:

- Bloco: 10100110
- $F(R_0, K_1) = 1111$

Faça:

1. Calcule L_1 e R_1 (em binário).
2. Concatene L_1 e R_1 para obter o bloco de 8 bits de saída após uma rodada e apresente-o também em binário.
3. Realize o processo inverso (descriptografe o bloco) e mostre que o texto original é recuperado.
4. Explique brevemente por que, com essa função F constante, a cifra se torna insegura, indicando qual propriedade (confusão ou difusão) é prejudicada.

3 Hash simples

Considere a função de hash iterativa simples definida por:

- Recebe de entrada dados de quaisquer tamanhos (Para facilitar utilize dados com tamanho múltipos de 8bits)
- Inicialização $H_0 = 00000000$;
- Funcionamento: separa o dado em blocos de 8 bits
- Em cada bloco de 8 bits B_i, $H_i = H_{i-1} \oplus B_i$.
- Resutado: último bloco de 8 bits

1. Mostre explicitamente (com um exemplo numérico) como é possível reordenar blocos sem alterar o hash.
2. Proponha uma modificação simples (mantendo iteração por blocos) que torne esse esquema resistente à simples reordenação de blocos (não precisa torná-lo criptograficamente seguro, apenas eliminar a propriedade de comutatividade).

4 Propriedades de Hash

1. Que características são necessárias em uma função de hash segura?
2. O que é resistência à resistência à pré-imagem?
3. O que é resistência à segunda pré-imagem?
4. O que é resistência à colisão forte?
5. Quais são as implicações de não se possuir resistência a colisão forte?
6. O que é a função de compactação de hash? O que torna uma função de compactação segura?

5 Colisões em Hash

Uma função de hash produz um resumo de (m) bits.

1. Usando o paradoxo do aniversário, estime o esforço (número de tentativas da ordem) necessário para encontrar uma colisão por ataque de força bruta.
2. Para ($m = 128$) e para ($m = 256$), escreva a ordem de magnitude desse esforço e comente se cada tamanho é suficiente para segurança moderna frente a ataques com recursos massivos.