RELATÓRIO TÉCNICO - TRABALHO EXTRA DE CRIPTOGRAFIA

Aluno: Álisson Rodrigues Fernandes
Matrícula: 510357
Data de Conclusão: 4 de janeiro de 2026
Ambiente de Desenvolvimento: C++ com OpenSSL e bibliotecas padrão
Hardware Utilizado: Computador com 8 núcleos de processamento
Hashes Quebrados com Sucesso: 3/3 (Todos os desafios completados)

1. INTRODUÇÃO

O presente relatório detalha a execução dos três desafios propostos no trabalho extra da disciplina de Criptografia, todos utilizando a função hash SHAKE128 com diferentes configurações de saída. Os desafios exploraram as três propriedades fundamentais de segurança de funções hash: resistência a colisões, resistência a segunda pré-imagem e resistência a pré-imagem.

Este trabalho teve como objetivo prático demonstrar como a redução deliberada do tamanho da saída de uma função hash robusta como SHAKE128 pode comprometer completamente sua segurança, tornando-a vulnerável a ataques computacionais com hardware comum.

2. DESCRIÇÃO DOS DESAFIOS E RESULTADOS OBTIDOS

2.1. Desafio A: Quebra da Resistência a Colisões (4 bytes)

Objetivo: Encontrar duas strings diferentes que gerem exatamente o mesmo hash de 4 bytes.

2.1.1. Análise dos resultados:

- O ataque foi extremamente rápido (0,093 segundos);
- Foram necessárias apenas 35.362 tentativas para encontrar uma colisão;
- A eficiência está alinhada com o paradoxo do aniversário, que prevê que uma colisão em um espaço de 2^{32} possibilidades será encontrada após aproximadamente 2^{16} = 65.536 tentativas;
- As strings encontradas são completamente diferentes, validando a quebra da resistência a colisões.

2.2. Desafio B: Quebra da Resistência a Segunda Pré-imagem (4 bytes)

Objetivo: Encontrar uma string diferente de "Aluno: Álisson Rodrigues" que gere o mesmo hash de 4 bytes.

2.2.1. Análise dos resultados:

- A segunda pré-imagem encontrada foi um número simples: "437443275";
- O ataque utilizou 8 threads para acelerar a busca;
- O hash da string original e da segunda pré-imagem são idênticos nos primeiros 4 bytes;
- Demonstração clara de como sistemas que utilizam apenas parte do hash são vulneráveis.

2.3. Desafio C: Quebra da Resistência a Pré-imagem (34 bits - Hash #3)

Objetivo: Encontrar qualquer string que gere um hash começando com E82C07F0 nos primeiros 34 bits.

2.3.1. Análise dos resultados:

- A pré-imagem encontrada foi o número: 3337893024008014782;
- Foram necessárias mais de 2,7 bilhões de tentativas para encontrar a solução;
- O tempo de execução foi de aproximadamente 40 minutos e 48 segundos;
- O hash completo dos primeiros 5 bytes é E82C07F016, onde os primeiros 34 bits correspondem exatamente ao alvo E82C07F0;
- Este foi o desafio mais computacionalmente intensivo, como esperado.

3. DIFICULDADES ENCONTRADAS

3.1. Desafio A (Colisão)

- **Dificuldade inicial:** Implementar uma estratégia eficiente de geração de strings candidatas;
- **Solução adotada:** Utilização de múltiplos padrões de geração (STR-, NUM-, etc.) para aumentar a diversidade dos candidatos;
- **Aprendizado:** O paradoxo do aniversário é extremamente eficaz na prática, tornando ataques de colisão viáveis mesmo em espaços teoricamente grandes.

3.2. Desafio B (Segunda Pré-imagem)

- **Dificuldade principal**: Codificação correta do caractere "Á" (UTF-8) para garantir que o hash da string alvo fosse calculado corretamente;
- **Problema técnico:** Gerenciamento de contexto OpenSSL em ambiente multithread;
- **Solução:** Criação de um contexto separado para cada thread e uso rigoroso de pares new/free;
- **Aprendizado:** Detalhes de implementação como codificação de caracteres e gerenciamento de memória são críticos em operações criptográficas.

3.3. Desafio C (Pré-imagem)

- **Maior desafio:** Tempo computacional prolongado (mais de 40 minutos);
- **Solução:** Implementação de busca paralela otimizada e configuração de salvamento automático do resultado;
- **Aprendizado:** A paralelização é essencial para ataques de pré-imagem, e mesmo com 34 bits de segurança, o ataque é viável com hardware moderno.

5. CONCLUSÕES E APRENDIZADOS

5.1. Principais Conclusões Técnicas

1. Tamanho do hash é crítico: Reduzir o tamanho do hash de 256 bits (padrão seguro) para apenas 4 bytes (32 bits) transforma um problema computacionalmente impossível em um que pode ser resolvido em frações de segundo.

2. Paradoxo do aniversário é poderoso: O Desafio A demonstrou que encontrar colisões é significativamente mais fácil que encontrar pré-imagens, mesmo com o mesmo tamanho de hash.

3. Paralelização é essencial: O Desafio C só foi viável através do uso eficiente de múltiplos núcleos de processamento, destacando a importância do paralelismo em ataques computacionais modernos.

4. Detalhes de implementação importam: Questões como codificação de caracteres (UTF-8 vs ASCII) e gerenciamento de recursos de bibliotecas criptográficas (OpenSSL) foram decisivas para o sucesso dos ataques.

5.2. Implicações para Sistemas Reais

- **Nunca truncar hashes:** Os resultados mostram que truncar hashes para economizar espaço ou largura de banda pode comprometer completamente a segurança do sistema.
- **Verificação em múltiplas camadas:** Sistemas críticos devem usar múltiplos mecanismos de segurança além de hashes simples.