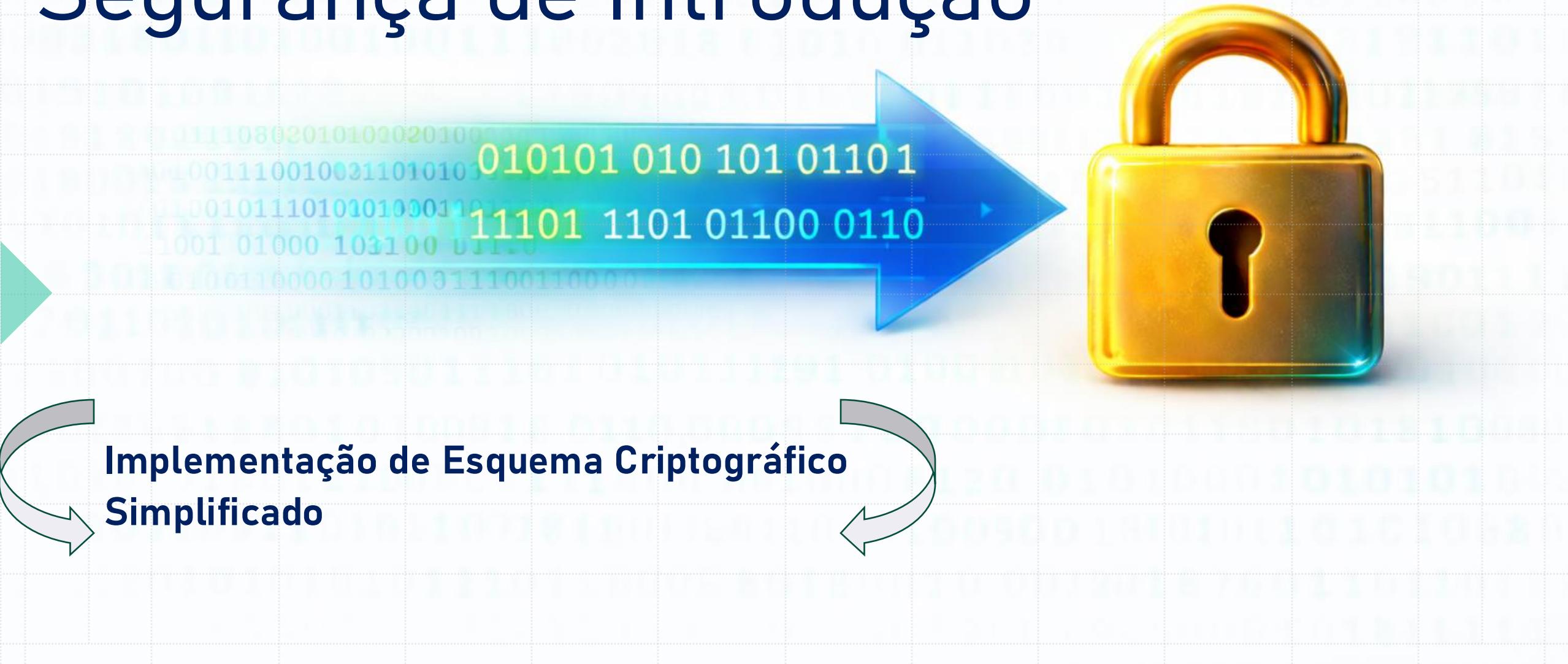


Trabalho de Auditoria e Segurança de Introdução



Este projeto implementa um algoritmo de criptografia simétrica determinística, inspirado na temática do TFT (Teamfight Tactics). O objetivo é demonstrar, de forma didática, os conceitos de Confusão, Difusão e Efeito Avalanche, utilizando módulos de geração de chaves (GEN), encriptação (ENC) e descriptação (DEC) sobre dados binários.

Geração de Chaves (GEN)

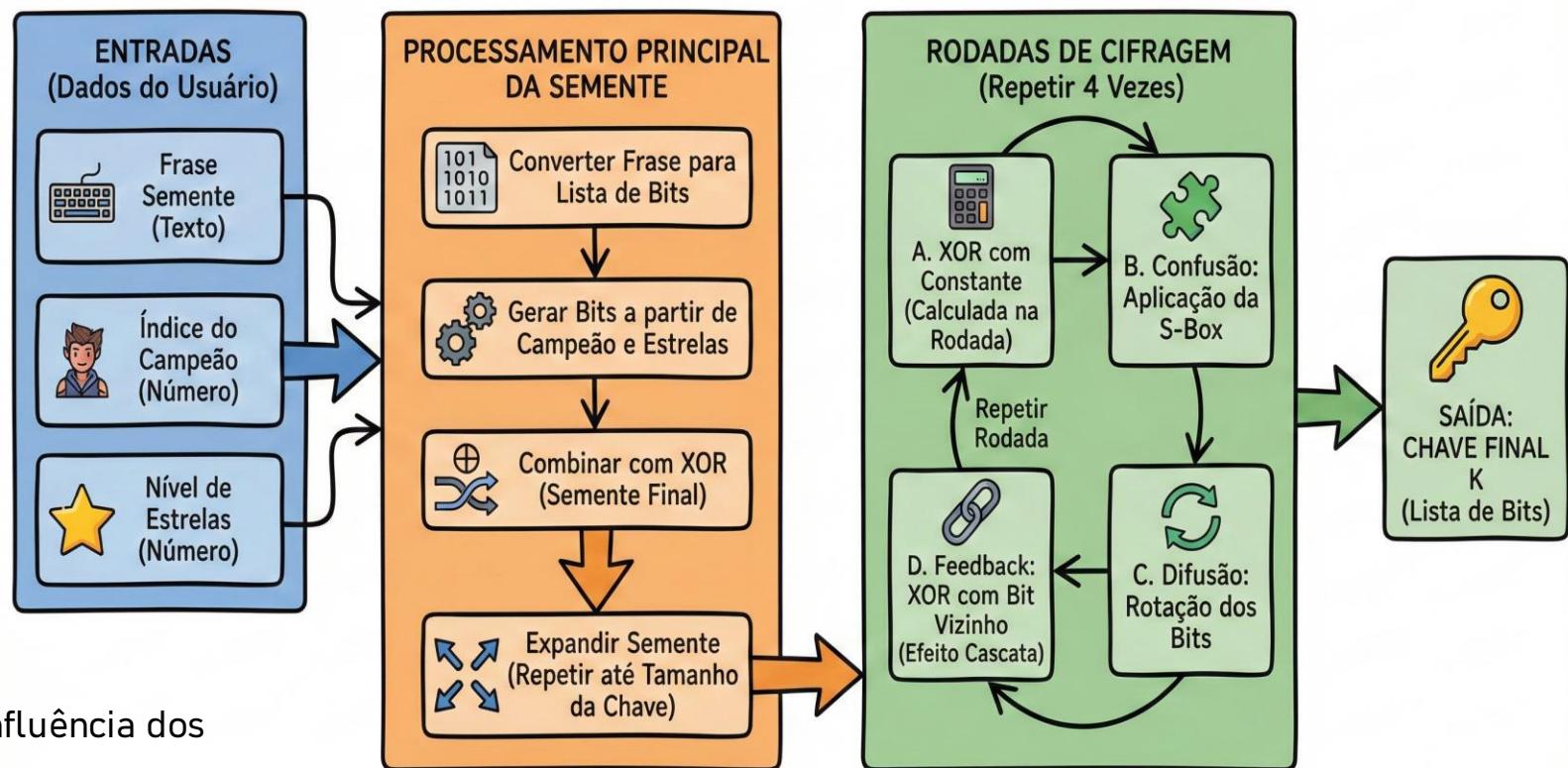
O módulo GEN funciona como a criação de uma *build* no TFT, onde as escolhas do jogador definem um resultado único.

Entradas do Jogador

- Seed (senha)** → base da estratégia
- Campeão** → personagem escolhido
- Estrelas** → nível de evolução

Processamento da Chave

- Conversão das entradas para **bits**
- Combinação das informações usando **XOR**
- 4 rodadas de processamento**, aplicando:
 - Confusão (S-Box)** → embaralha os bits
 - Difusão (rotação + feedback)** → espalha a influência dos bits



Assim como no jogo, **mudar um único detalhe** (seed, campeão ou estrelas) gera uma **chave completamente diferente**.



Encriptação da Mensagem (ENC)

Após a criação da *build* no GEN, a encriptação (ENC) representa o momento em que essa build entra em combate. A mensagem original é convertida em bits e misturada à chave por meio de um XOR inicial, garantindo proteção desde o início.

O processo executa 2 rodadas de uma rede SPN simplificada, aplicando:

Confusão (S-Box) → embaralha os bits

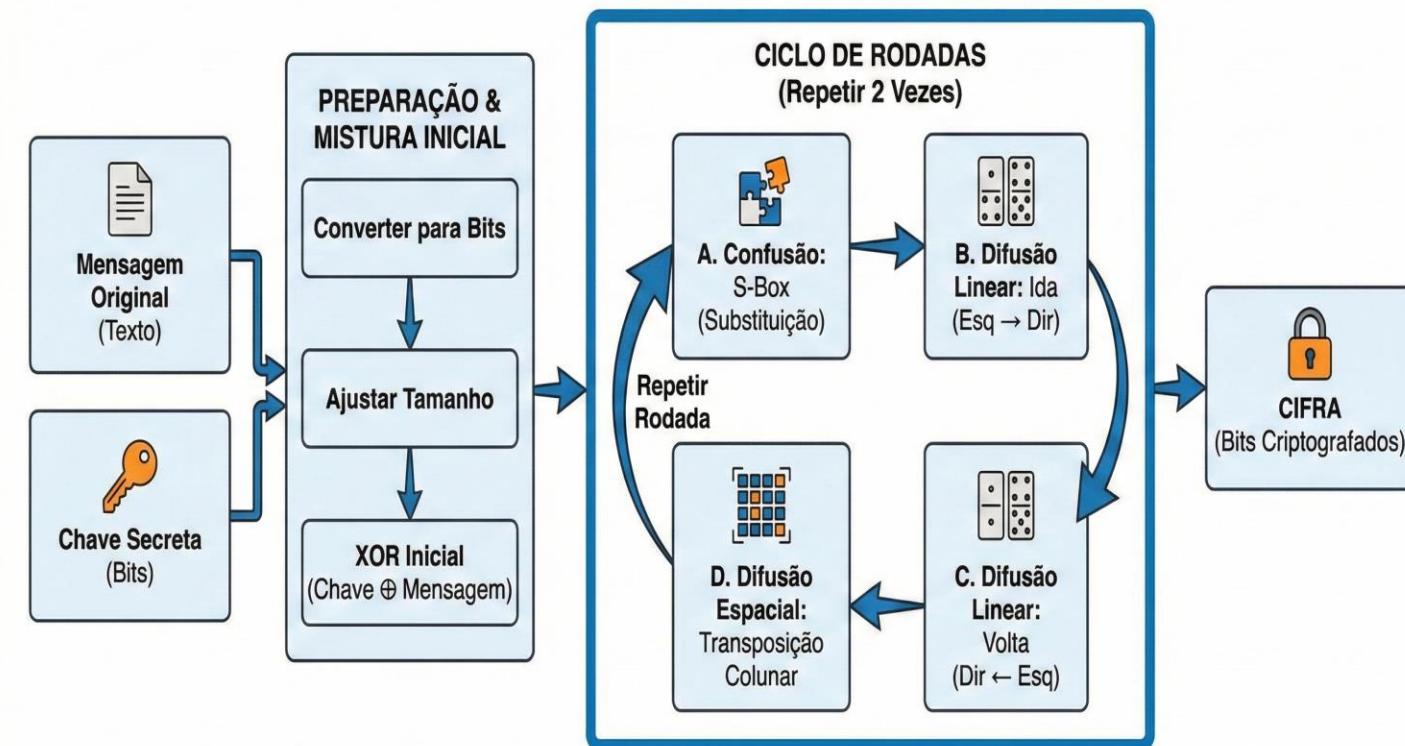
Difusão bidirecional → um bit influencia os demais

Difusão espacial (transposição) → evita padrões previsíveis

Assim como no jogo, pequenas mudanças na mensagem ou na chave geram uma cifra completamente diferente.

Processo de Encriptação Simétrica (ENC)

Fluxo didático das etapas do arquivo 'enc.py' em Português



→ Descrição da Mensagem (DEC)

Após a batalha da ENC, a descrição (DEC) representa o caminho de volta ao estado original. Usando a mesma chave gerada no GEN, o algoritmo desfaz cada etapa da encriptação na ordem inversa (LIFO), como se estivesse revertendo todos os buffs aplicados durante o combate.

Processo de Descrição

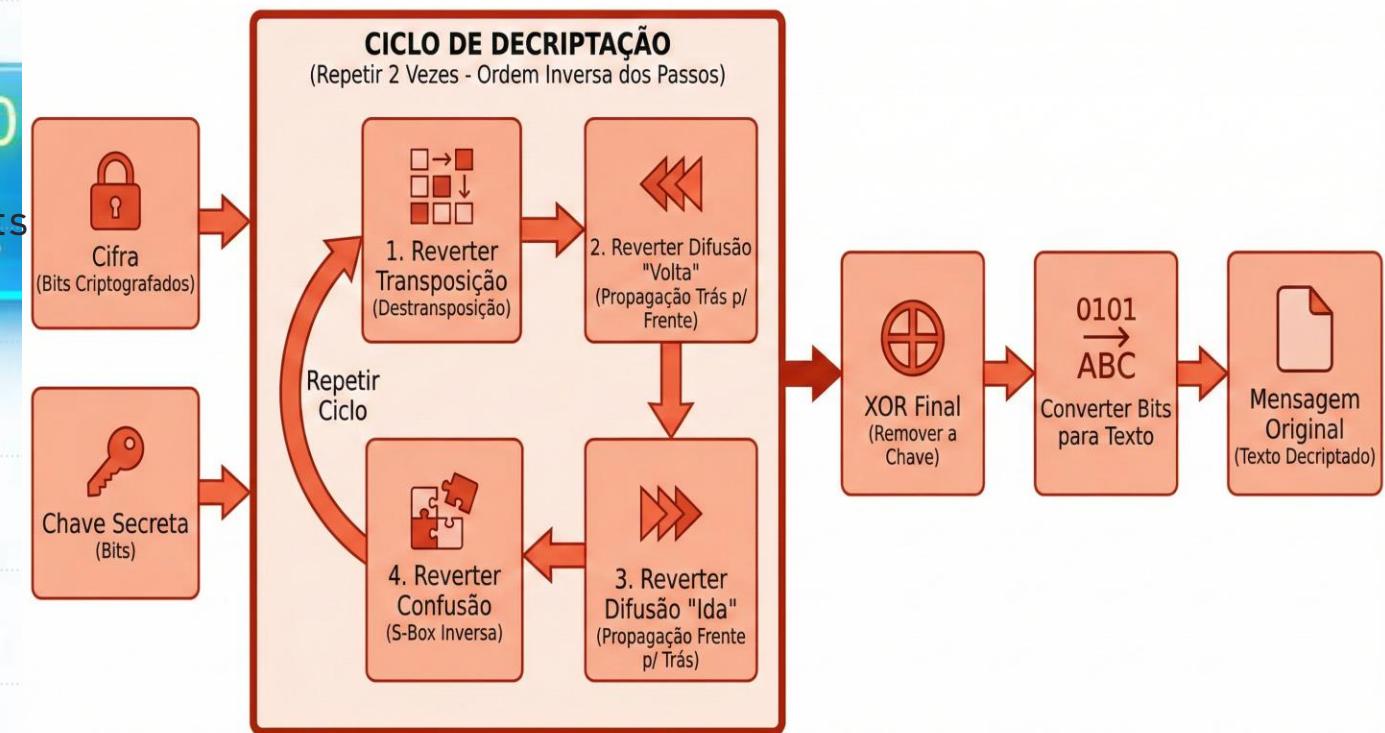
Em **2 rodadas**, o sistema executa:

- **Reversão da transposição** → retorna os bits às posições originais
- **Reversão da difusão** → desfaz a propagação dos bits
- **S-Box inversa** → remove a confusão aplicada
- **XOR final com a chave** → elimina o efeito da criptografia

Ao remover todos os efeitos corretamente, a **mensagem original é recuperada exatamente**, sem perda de dados.

Processo de Decriptação Simétrica (DEC) - O Caminho Inverso

Diagrama didático das etapas do arquivo 'dec.py' em Português, desfazendo a encriptação.



Testes e Métricas de Avaliação

Para validar os critérios exigidos na especificação, o projeto inclui um script automatizado de testes (`testes.py`), responsável por avaliar a qualidade e a segurança do algoritmo implementado.

- Os testes executados cobrem os seguintes aspectos:

- Tempo de Execução**

Mede o tempo médio das funções ENC e DEC em múltiplas execuções, avaliando a eficiência do algoritmo.

- Análise de Colisões (Chaves Equivalentes)**

Verifica a integridade do espaço de chaves, garantindo que chaves diferentes ($K_1 \neq K_2$) não gerem a mesma cifra para uma mesma mensagem.

- Teste de Difusão (Efeito Avalanche na Mensagem)**

Avalia o impacto da alteração de 1 bit na mensagem original, buscando uma taxa de alteração próxima de 50%, conforme esperado para máxima difusão.

- Teste de Confusão (Efeito Avalanche na Seed/Chave)**

Avalia o impacto da alteração de 1 bit na seed, verificando se a chave e a cifra resultante sofrem alterações significativas.

Os testes demonstram que o algoritmo apresenta boa difusão, alta confusão, ausência relevante de colisões e desempenho adequado. Pequenas alterações na mensagem ou na chave geram impactos significativos na cifra, comprovando a eficiência do modelo implementado.



Assim como no TFT, uma pequena mudança na estratégia — seja no campeão, nas estrelas ou na build — resulta em um resultado completamente diferente.