UNIVERSIDADE ESTADUAL DE SANTA CRUZ
PATRICK SILVA FERRAZ

GERADOR DE CHAVES
REDE DE FEISTEL

ILHÉUS-BA
2017

SUMÁRIO

1. Descrição

Este projeto especifica os métodos utilizados para desenvolvimento do algoritmo de geração de chaves para serem aplicadas na Rede de Feistel. A geração é realizada através de permutações de deslocamento em uma chave de entrada de 8 *bytes* (64 *bits* – 8 caracteres), gerando 16 chaves de saída como resultado. Os próximos tópicos descrevem tais utilizações.

2. Objetivo

Este projeto tem por objetivo desenvolver um algoritmo que gere 16 chaves de 48 bits a partir de uma chave de 64 bits (8 caracteres) para posterior aplicação na rede de Feistel.

3. Métodos

O algoritmo foi desenvolvido e compilado no sistema operacional *Linux OpenSuse Leap 42.2* através editor *Kate 16.08.2*, utilizado a linguagem de programação C com as bibliotecas “stdio”, “stdlib” e “ctype” e compilado utilizando o *gcc 4.8.5*.

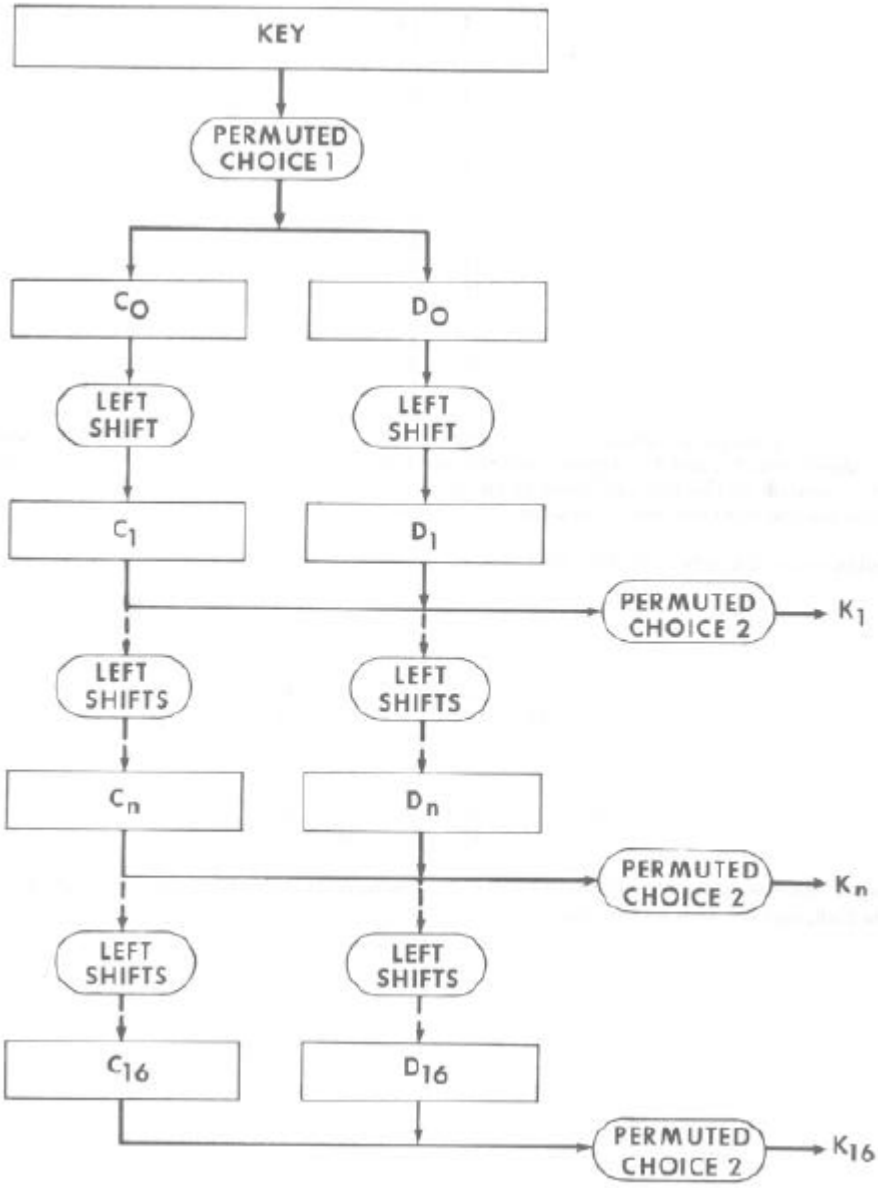

Imagem 1. *Key schedule calculation*

Na imagem 1, pode-se observar o fluxo principal da geração de chaves utilizada, no qual em um bloco de 8 *bytes*, representado pela KEY, é realizado uma *permuted choice 1* (PC1) resultando em dois blocos de 28 *bits* (um bloco de 56 *bits*) C0 e D0, ou seja, há uma perda de 8 *bits*, respectivos aos *bits* múltiplos de 8.O truncamento da *permuted choice 1* pode ser observado na imagem 2.

***PC*-1**

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Imagem 2. *Permuted choice 1*

Após a PC1 ser realizada, antes de cada geração de chaves, são realizados deslocamentos de *bits* individualmente em cada bloco Ci e Di, para posterior aplicação das *permuted choice 2*. Os deslocamentos podem ser de 1 ou 2 unidades binárias dependendo da chave gerada. A imagem 3 representa as escolhas de deslocamentos antes de cada geração de Ki.

Iteration Number	Number of Left Shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

Imagem 3. Deslocamentos por iteração

Posteriormente, as chaves são geradas através da *permuted choice 2*, conforme representado na imagem 4.

PC-2

14	17	11	24	1	5
3	28	15	6	21	10
23	19	12	4	26	8
16	7	27	20	13	2
41	52	31	37	47	55
30	40	51	45	33	48
44	49	39	56	34	53
46	42	50	36	29	32

Imagem 4. *Permuted choice 2*

4. Desenvolvimento

O documento que contém o algoritmo foi nomeado como “geradorChaves.c”.

4.1. Comando de compilação

A compilação do algoritmo foi realizada no gcc através do sistema operacional *Linux* (Código 1), portanto para gerar o executável para o *Windows*, basta compilar novamente no sistema de escolha.

$ *gcc geradorChaves.c -o geradorChaves*

Código 1.

4.2. Comando de execução

Tomando como exemplo uma chave “teste123”, o comando de geração de chaves é representado através da execução do Código 2.

// Digite a chave de 64 bits

$ *./geradorChaves teste123*

Código 2.

4.3. Testes realizados

Entradas inválidas:

- Diferente de dois argumentos (Código 3).

// 1 argumento (1 = geradorChaves)

$ *./geradorChaves*

// Mais de 2 argumentos

$ *./geradorChaves teste123 outro123*

Código 3.

5. Desenvolvimento

Toda documentação pode ser encontrada através do link:

https://github.com/patricksferraz/uesc_criptografia

6. Referências

[1] CSRC – **Computer Security Resource Center**. Disponível em: https://csrc.nist.gov/csrc/media/publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf <último acesso em: 01 de novembro de 2017>