Criptografia por substituição

Dante Eleutério dos Santos GRR20206686
Richard Fernando Heise Ferreira GRR20191053

[1]Universidade Federal do Paraná
Curitiba – PR – Brasil

1. Introdução

O trabalho consiste na implementação de um algoritmo que utiliza transposição e substituição de caracteres a fim de criptografar um texto claro. Nosso algoritmo em específico recebe o nome de *lingxing* e implementa um algoritmo de substituição simples usando a lógica dee matrizes e reordenação de colunas seguido de uma substituição alfabética simples, os detalhes estarão nas secções abaixo.

2. O algoritmo

O algoritmo funciona da seguinte forma:

Para criptografia:
Primeiro, uma entrada é lida do terminal e armazenada em um buffer de chars, lemos entradas sem quebras de linha, em seguida esse buffer é transformado seguindo o seguinte processo: é criada uma matriz com o texto original, essa matriz possui 4 colunas obrigatoriamente, o texto claro é montado como uma matriz, as colunas são shiftadas para a direita e reescritas na forma embaralhada, para garantir que sempre há uma matriz quadrada, são completadas as lacunas faltantes, caso hajam, usando '˜'. Vamos rodar um exemplo para facilitar a explicação:

texto claro: texto_para_hoje

matriz:
1 2 3 4
t e x t
o _ p a
r a _ h
o j e ˜

Isso se torna o vetor transposto: toroe ajxp etah˜, ou seja, cria-se o vetor com base na leitura das colunas.

Depois, shiftamos a matriz da seguinte forma: 4123, isto é,
4 1 2 3
t e x t
o _ p a
r a _ h
o j e ˜

Ou seja, vetor shiftado fica: e ajxp etah˜toro.
Depois disso, só falta transformar em "mandarim" com uma função de conversão.

Para decriptografar:
A decriptografia é completamente simétrica, isto é, basta desfazer o que foi feito e retirar os ’~’ do texto final.

3. Implementação

Utilizamos um código de conversão de UTF-8 para UTF-32 que encontramos no site rosettacode, visto que a conversão foi extremamente complexa quando tentamos realizá-la. O código está em C, possui Makefile e pode ser rodado seguindo o que explica o arquivo *usage.txt*.

O código está comentado e, no geral, seu funcionamento é simples o bastante para que não haja comentários nesta seção que justifiquem-se. Já que o algoritmo já foi explicado na seção anterior.