Universidade Federal Fluminense

Bacharelado em Ciência da Computação

Introdução à Criptografia - 2013.2

Professor: Luis Antonio Brasil Kowada

Trabalho 1 - Implementação de Cifra de blocos

Alunos: Carlos Filipe Marques Teixeira Jr. (cfmtjr@gmail.com)
Guilherme Rolim e Souza (loxorolim@gmail.com)
Mateus Carvalho Azis (mateusazis@gmail.com)

Descrição

Esse trabalho é uma implementação de três algoritmos criptográficos: AES com chave de 128 bits, AES apenas com a operação AddRoundKey e AES com Cifra de Vigenére. O objetivo é possibilitar que esses algoritmos sejam utilizados para codificar e decodificar imagens provenientes de arquivos, usando esquemas de encadeamento de blocos ECB e CBC.

Operações sobre imagens

Para realizar a codificação e decodificação das imagens, este trabalho faz uso da biblioteca DevIL. Foram utilizados os binários pré-compilados da versão 1.7.8 sem Unicode para Windows 32-bits, utilizando linkagem dinâmica. Os algoritmos de codificação, porém, foram feitos de forma a funcionarem com dados provenientes de qualquer tipo de fonte. As imagens são apenas as vias de entrada e saída de dados.

Instruções de compilação

A seguir, explicamos como compilar o projeto dentro do ambiente Windows.

A) Visual Studio

A.1) Abra o arquivo de *solution* TrabalhoCriptografia.sln utilizando o Visual Studio Express 2013 ou mais novo.

A.2) Acesse o menu *BUILD* e escolha a opção *Build Solution* para compilar o projeto.

B) GCC e MinGW

B.1) Abra uma janela do terminal e navegue até a pasta TrabalhoCriptografia

B.2) Digite o comando:

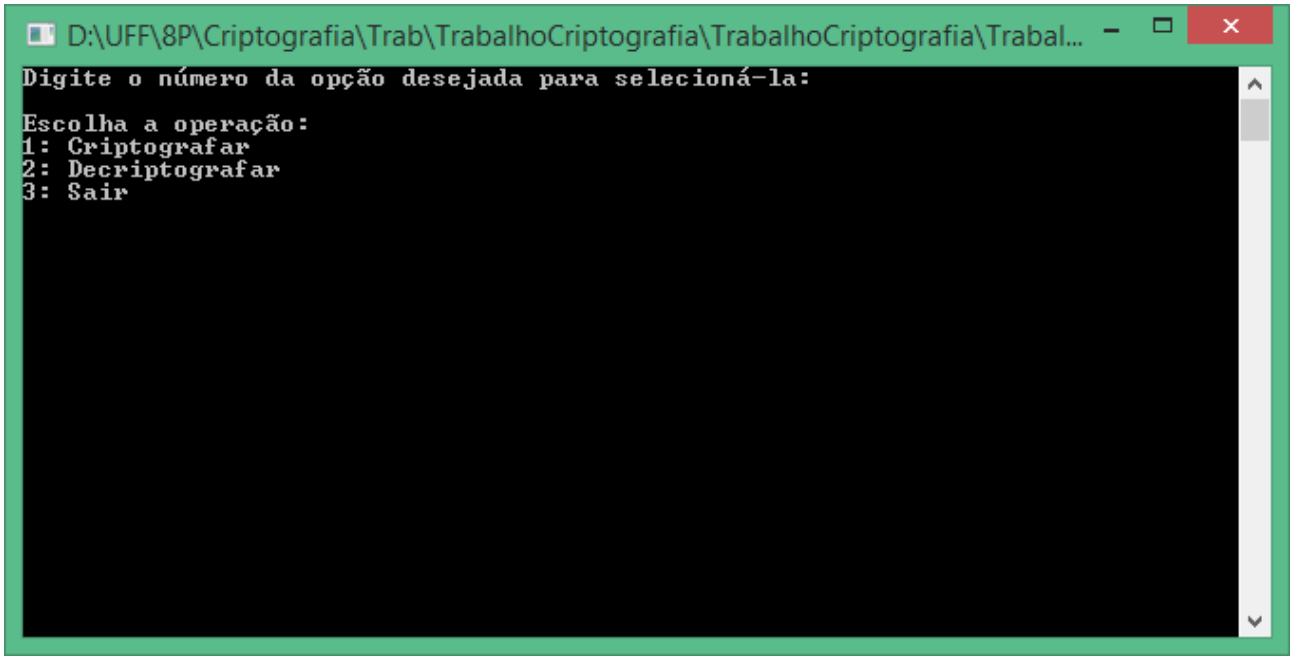
mingw32-make Makefile trabalho

Em ambos os casos, deve-se manter uma cópia da biblioteca dinâmica localizada em devil/lib/DevIL.dll na mesma pasta do executável.

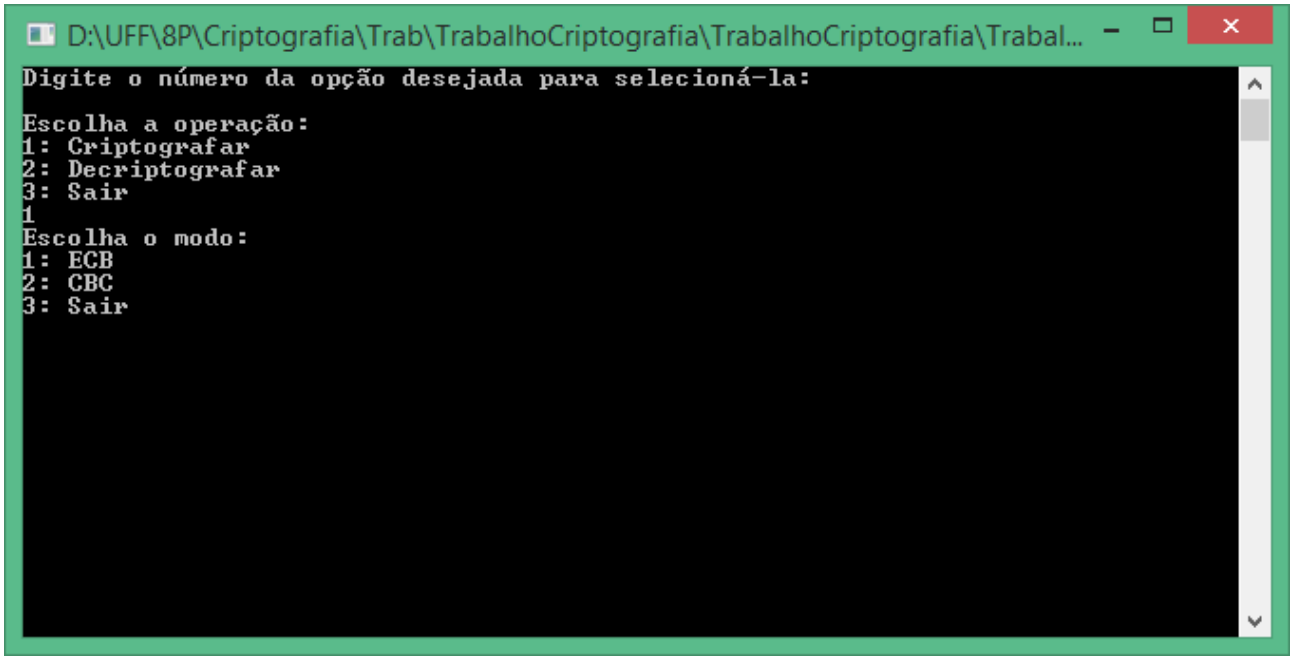
Este projeto foi compilado com sucesso utilizando o Visual Studio Express 2013 e também com GCC 4.8.1 para Windows.

Guia de uso

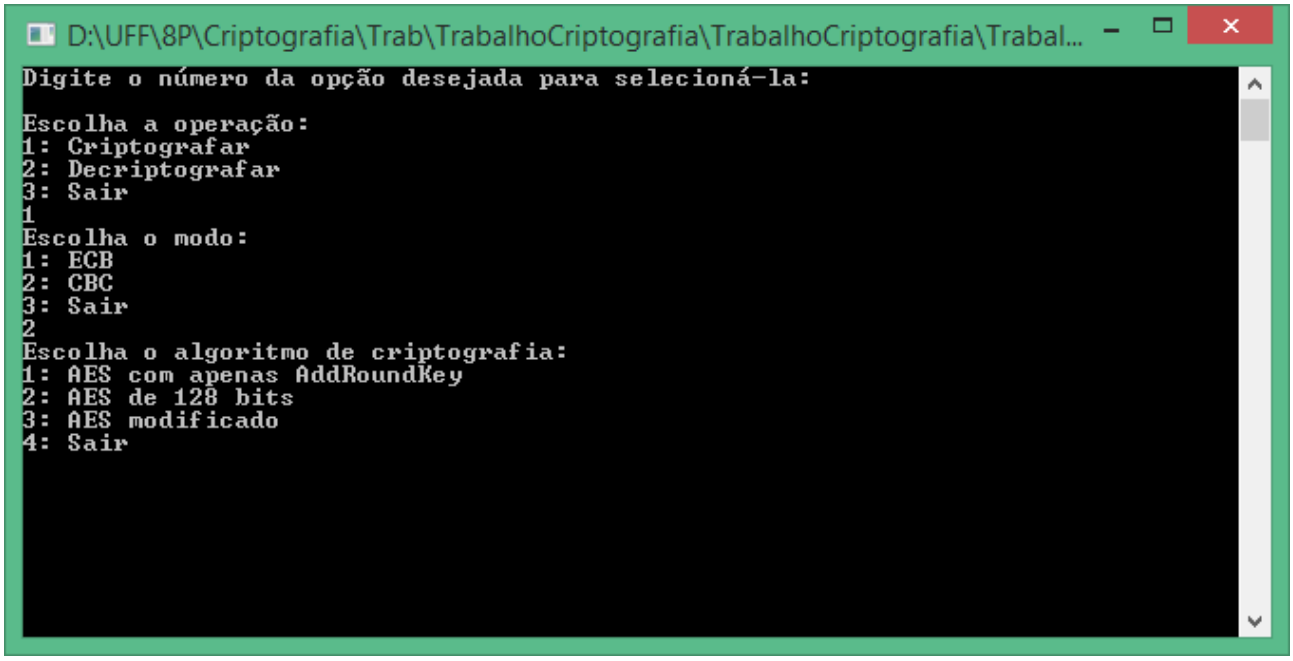
Execute o arquivo **TrabalhoCripto.exe**. Será exibida a tela a seguir.



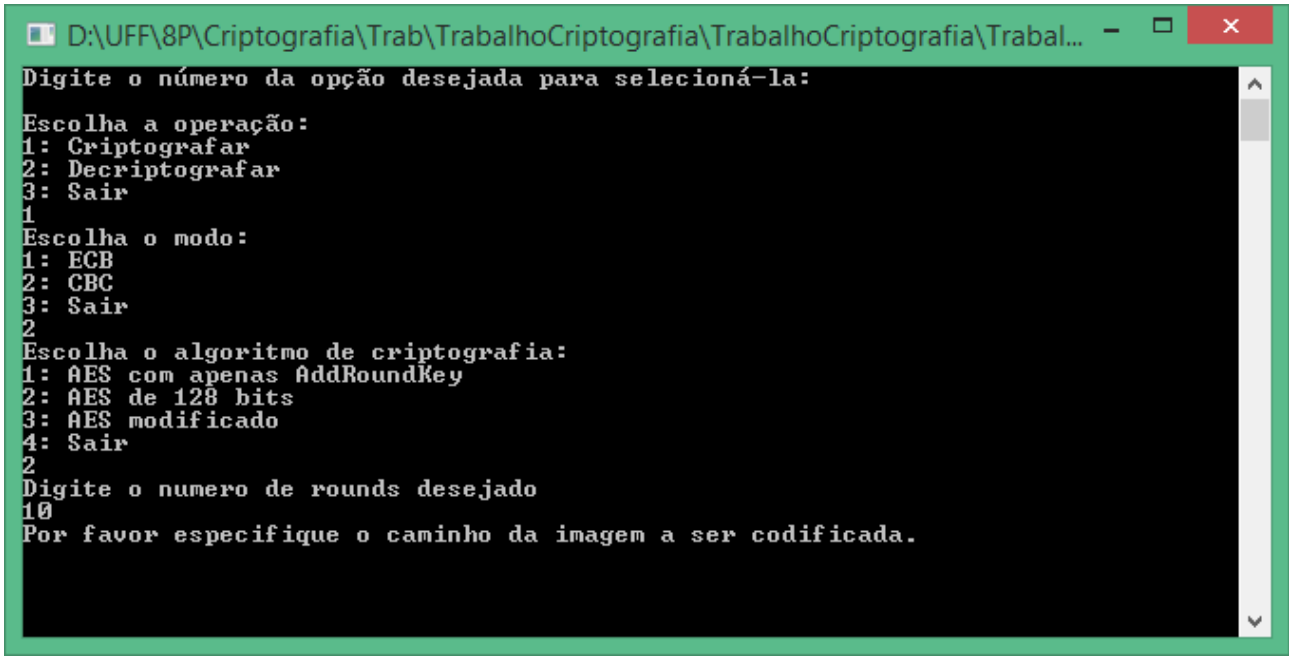
Digite "1" para codificar uma imagem. Na próxima janela, escolha "1" novamente para realizar a codificação com encadeamento de blocos do tipo *Electronic codebook*. Se desejar utilizar *Cipher-block chaining*, escolha "2".



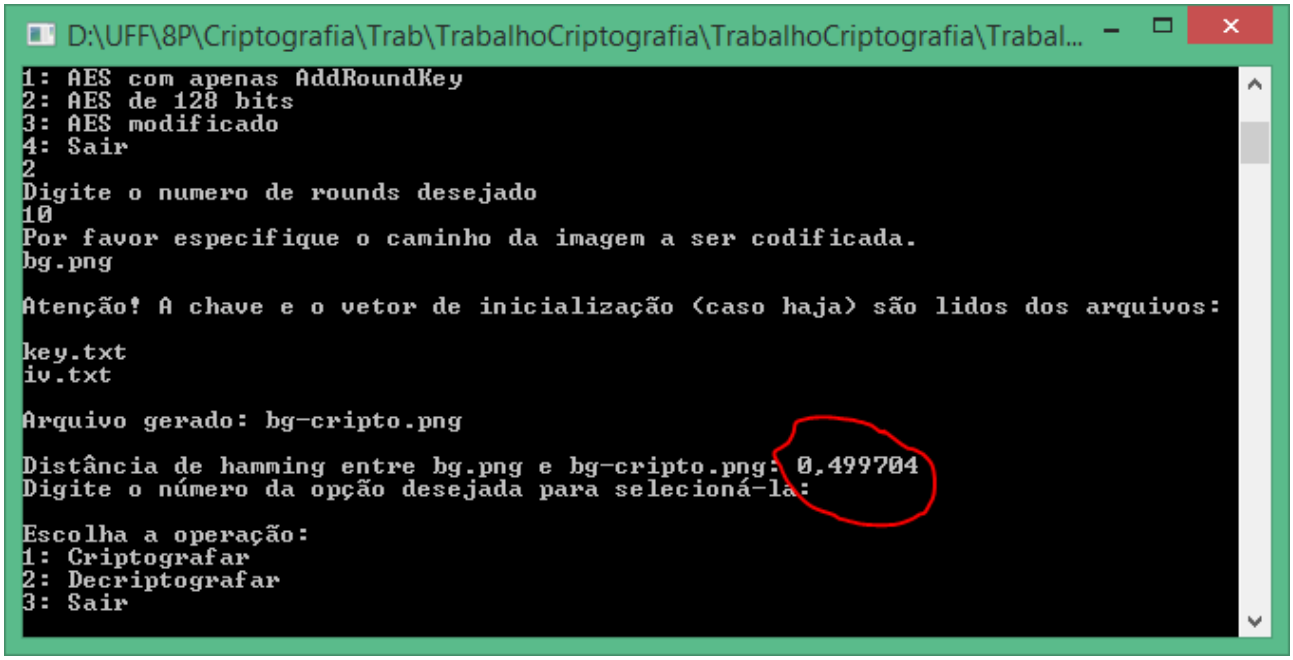
A seguir, são apresentados os métodos de codificação disponíveis. Escolha "1" para realizar o AES apenas com a etapa *AddRoundKey*, "2" para realizar o AES tradicional com chave de 128 bits ou "3" para realizar o AES com Cifra de Vigenére.



Em seguida, digite um número inteiro para indicar a quantidade de *rounds* a serem executados pelo procedimento (normalmente, 10 *rounds*).



Finalmente, digite o nome do arquivo de imagem a ser codificado. São aceitos arquivos **.bmp** de 24 bits ou **.png**. Pressione ENTER e o processamento terá início. Ao término, será exibida a Distância de Hamming entre a imagem original e a codificada. Para um arquivo cujo nome segue o formato **nome.extensão**, o resultado será salvo em **nome-cripto.extensão**.



O programa volta ao menu inicial.

Exemplos de resultados

A seguir, mostramos um exemplo de resultado obtido usando AES de 128 bits com CBC e 10 *rounds*.

Imagem original:

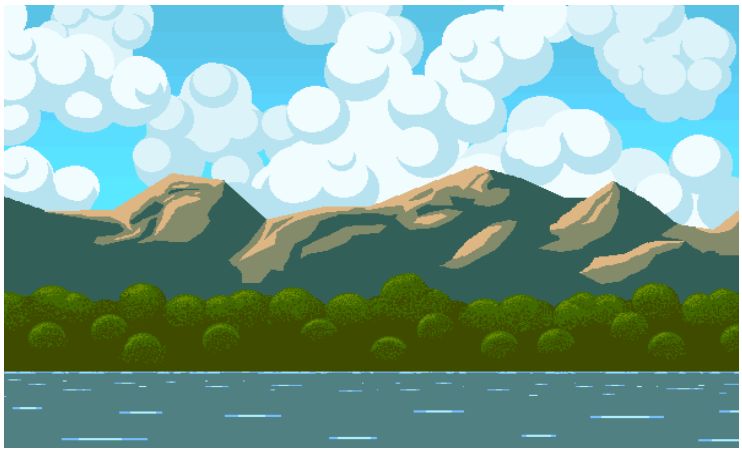
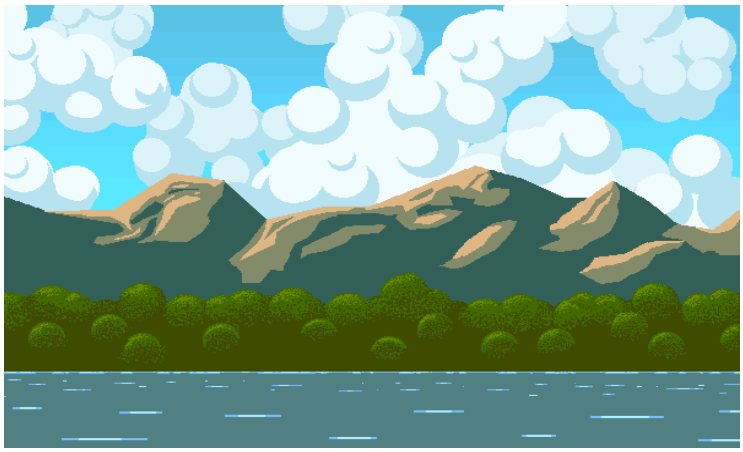


Imagem codificada:

Imagem decodificada:



Descrição dos arquivos fonte

main.c

Arquivo principal. Faz a inicialização da biblioteca de imagens, exibe a interface ao usuário e chama os métodos de codificação adequados.

AES.c

Implementação do AES de 128 bits.

AlternativeAES.c

Implementação alternativa do AES de 128 bits com os métodos de transposição e de substituição trocados. Implementa a Cifra de Vigenere.

AddRoundKeyAES.c

Implementação do AES de 128 bits utilizando apenas a etapa *AddRoundKey.*

Images.c

Realiza a comunicação de entrada e saída de dados através de imagens e executa os algoritmos adequados sobre esses dados.

utils.c

Implementação de funções auxiliares úteis à implementação dos demais algoritmos.