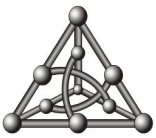

UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL
Faculdade de Computação

Microprocessador criptográfico

Trabalho 1

Algoritmos e Programação I – 2018

1 Descrição

Figura 1: Imagem de *iStockphoto*.

Criptografia é uma subárea da Computação que estuda técnicas de comunicação segura na suposta presença de intrusos. Atualmente, a área se preocupa em construir e analisar protocolos que impeçam a influência de intrusos e que estão relacionados com aspectos de segurança da informação, tais como confidencialidade e integridade dos dados e autenticação. As aplicações são muitas, como segurança em cartões bancários, senhas em computadores, comércio eletrônico, etc. A Matemática e a Engenharia Elétrica também são áreas de suporte à Criptografia, além da própria Computação. A Criptografia é uma das subáreas responsáveis pelo aparecimento do computador, motivada pelas grandes guerras do século passado. Os métodos usados em Criptografia têm se tornado cada vez mais complexos e suas aplicações cada vez mais abrangentes.

Os professores da Faculdade de Computação da UFMS têm ficado muito preocupados com suas pesquisas. Sempre que um professor está próximo a anunciar um resultado novo de uma pesquisa sua, um outro professor lhe toma a frente e faz o anúncio primeiro, roubando-lhe o tempo investido, os resultados e o reconhecimento da comunidade científica. Como isto tem se tornado uma rotina constante, os professores já estão muito desconfiados uns dos outros e o ambiente de trabalho tem se tornado insuportável e cada vez mais desetimulante. Medidas drásticas têm sido aventadas, como a abertura das caixas de mensagens de todos os professores, escutas telefônicas, câmeras ou mesmo a substituição dos seus computadores por calculadoras eletrônicas.

Para restabelecer o clima de cooperação, paz e harmonia que sempre houve entre os professores da FACOM, você foi convocado(a) a decrifrar as mensagens que são enviadas pelos computadores dos professores.

Todas as mensagens dos computadores da FACOM são transmitidas antes passando por um microprocessador critográfico simples, que embaralha os bits de cada byte da mensagem. Um exemplo de um microprocessador como este é mostrado na figura 2.

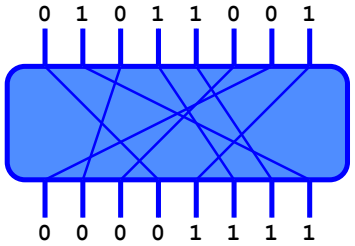

Figura 2: Um microprocessador criptográfico que embaralha os 8 bits de entrada. Os bits de entrada sempre são fornecidos de acordo com a permutação $(7, 6, 5, 4, 3, 2, 1, 0)$. Nesta figura, os bits de entrada são embaralhados de tal forma que os bits de saída obedecem à permutação $(1, 5, 2, 7, 0, 4, 3, 6)$.

Os professores da FACOM desconhecem este processo de cifragem de suas mensagens, mas já há desconfiança que algum deles tenha interceptado as mensagens cifradas, estudado-as, e tenha então conseguido quebrar esse código, alterando mensagens importantes.

Neste programa, você deverá espionar as mensagens que são enviadas dos computadores da FACOM. Você receberá chaves de decifragem dos microprocessadores de criptografia instalados nos computadores da FACOM e mensagens cifradas. Sua tarefa é decifrar essas mensagens.

2 Entrada

A entrada é composta de vários casos de teste. Um caso de teste é composto por uma linha que contém apenas um número inteiro positivo k, com $1 \leq k \leq 20$, que indica a quantidade de casos de teste. Cada caso de teste é composto por uma linha inicial que contém uma permutação do conjunto $\{0, 1, 2, 3, 4, 5, 6, 7\}$, cada número separado por um espaço, representando a chave de criptografia usada no microprocessador. Na próxima linha são fornecidos 3 números inteiros d, o e h, separados por um espaço, que indicam as bases numéricas das codificações dos números na entrada: o número d indica que a cada d números temos um número fornecido na base decimal; o número o indica que a cada o números temos um número fornecido na base octal; e o número h indica que a cada h números temos um número fornecido na base hexadecimal. Caso contrário, isto é, se um número fornecido na entrada não é múltiplo de d, o e h, então a entrada é fornecida na base binária. Empates são resolvidos da seguinte ordem: hexadecimal, octal e decimal. Por exemplo, se $d = 3$ e $o = 4$, então o 12^{o} número na entrada é fornecido na base octal. Na próxima linha é fornecido um número inteiro n, com $1 \leq n \leq 1000$, que indica quantos bytes serão transmitidos na mensagem. Depois disso, n números são fornecidos, um a cada linha, na sua respectiva base de numeração.

3 Exemplo de entrada

```
2

1 5 2 7 0 4 3 6
3 5 7
22
01110000
00111101
108
01111101
075
180
AC
01110100
101
004
01111101
004
10101100
74
354
11110100
00110101
180
01100100
064
7D
01000100

6 1 0 4 7 2 3 5
2 3 5
14
00111010
039
217
131
23
001
00110011
055
047
0B
00011011
043
00000111
035
```

4 Saída

Para cada caso de teste da entrada seu programa deve produzir a impressão dos bytes como representações gráficas de caracteres. A primeira linha identifica o caso de teste, no formato "Teste k:", onde k é iniciado a partir de 1. A seguir, imprima a representação gráfica do caractere decifrado correspondente a cada caractere cifrado fornecido na entrada. Separe os casos de teste com uma linha em branco.

5 Exemplo de saída

```
Teste 1:
Encontrei o resultado!

Teste 2:
Senha quebrada
```

6 Instruções para entrega

1. **Compilador**

   Os(as) professores(as) usam o compilador da linguagem C da coleção de compiladores GNU `gcc`, com as opções de compilação `-Wall -ansi -pedantic` para corrigir os programas. Se você usar algum outro compilador para desenvolver seu programa, antes de entregá-lo verifique se o seu programa tem extensão `.c`, compila sem mensagens de alerta e executa corretamente.

2. **Forma de entrega**

   A entrega do trabalho deve ser realizada diretamente no Sistema de Suporte a Disciplinas da FACOM (EAD/FACOM), na disciplina de Algoritmos e Programação I. Depois de acessar o EAD da disciplina vá até o tópico e escolha "Entrega do trabalho 1". Você pode entregar o trabalho quantas vezes quiser até às **23 horas e 55 minutos** do dia **22 de abril de 2018**. A última versão entregue é aquela que será corrigida. Encerrado o prazo, não serão mais aceitos trabalhos.

3. **Atrasos**

   Trabalhos atrasados não serão aceitos. Não deixe para entregar seu trabalho na última hora. Para prevenir imprevistos como queda de energia, problemas com o sistema, e/ou falha de conexão com a internet, sugerimos que a entrega do trabalho seja feita pelo menos um dia antes do prazo determinado.

4. **Erros**

Trabalhos com erros de compilação receberão nota ZERO. Faça todos os testes necessários para garantir que seu programa está livre de erros de compilação.

5. **O que entregar?**

   Você deve entregar um único arquivo contendo APENAS o seu programa fonte. NÃO entregue qualquer outro arquivo, tal como o programa executável, já compilado.

6. **Verificação dos dados de entrada**

   Não se preocupe com a verificação dos dados de entrada do seu programa. Seu programa não precisa fazer consistência dos dados de entrada. Isto significa que se, por exemplo, o seu programa pede um número entre 1 e 10 e o usuário digita um número negativo, uma letra, um cifrão, etc, o seu programa pode fazer qualquer coisa, como travar o computador ou encerrar a sua execução abruptamente com respostas erradas.

7. **Arquivo com o programa fonte**

   Seu arquivo contendo o programa fonte na linguagem C deve estar bem organizado. Um programa na linguagem C tem de ser muito bem compreendido por uma pessoa. Verifique se seu programa tem a indentação adequada, se não tem linhas muito longas, se tem variáveis com nomes significativos, se possui funções precisas, entre outros. Não esqueça que um programa bem descrito e bem organizado é a chave de seu sucesso. Não esqueça da documentação de seu programa.

   Dê o nome do seu usuário do servidor da FACOM para seu programa e adicione a extensão `.c` ao final do nome deste arquivo. Por exemplo, `marco_aurelio.c` é um nome válido.

8. **Saída**

   Observe que a saída do seu programa é enxuta. Em particular, sempre que ocorre alguma interação do programa com o(a) usuário(a), nenhuma mensagem é emitida na saída. Por exemplo, na leitura de um número, não imprima qualquer mensagem como `"Informe um número: "`.

9. **Conduta Ética**

   O trabalho deve ser feito INDIVIDUALMENTE. Cada estudante tem responsabilidade sobre cópias de seu trabalho, mesmo que parciais. Não faça o trabalho em grupo e não compartilhe seu programa ou trechos de seu programa. Você pode consultar seus colegas para esclarecer dúvidas e discutir idéias sobre o trabalho, ao vivo ou no fórum de discussão da disciplina, mas NÃO copie o programa!

   Trabalhos considerados plagiados terão nota ZERO.