Universidade de Brasília
Faculdade Gama
Curso – Sistemas Embarcados.
Professor: Edson Mintsu Hung / Renato Coral.
Datas: 01/12/2015 a 10/12/2015
Aluno: _____ **Matrícula:**_____

Nota

Instruções (Leia com Atenção!!!)

- Esta avaliação é individual e só será considerada após o visto no laboratório.
- É expressamente proibido qualquer tipo de consulta (professor, livros, apostilas, colega, etc.) em qualquer etapa da prova. Caso se constate algum tipo de tentativa, a prova receberá nota ZERO.

Trabalho Final de Sistemas Embarcados

Questão única

Nesta questão iremos simular um sistema de interceptação (que será simulada por meio de arquivos), decodificação de informações esteganografadas em vídeos (sequencia de imagens) e verificação de autenticidade. A esteganografia é uma técnica utilizada para esconder uma informação dentro de outra. Observe que seu propósito é diferente ao da criptografia, onde o objetivo é proteger o conteúdo da mensagem.

Uma das formas mais comuns da esteganografia é utilizando a técnica denominada *Least Significant Bits* (LSB), que consiste em utilizar os bits menos significativos para codificar a mensagem a ser escondida. Por exemplo: Suponha que iremos codificar a letra A (ASCII 65 ou 01000001) em 8 bytes da imagem abaixo utilizando apenas o bit menos significativo:

01011101 11010000 00011100 10101100
11100111 10000111 01101011 11100011

resultando na seguinte imagem:

01011100 11010001 00011100 10101100
11100110 10000110 01101010 11100011

Como foi utilizado apenas um bit menos significativo, obteremos uma imagem com informação esteganografada similar à imagem original (antes de se embutir a informação). Pois o sistema visual humano é pouco susceptível às pequenas variações na imagem. No geral, as técnicas de esteganografia permitem esconder texto, áudio, imagens e diversas outras informações sem comprometer significativamente a percepção visual da imagem original. Existem várias aplicações comerciais para este tipo de técnica como: sistemas de entretenimento, verificação de integridade, marcas d'água digitais, etc. O processo de esteganografia para cada pixel foi feita da seguinte forma:

Pixel original: [A B C D E F G H]
Pixel a ser escondido: [a b c d e f g h]
Informação esteganografada com 3 bits: [A B C D E c b a]

Portanto, neste exemplo, o pixel extraído da esteganografia é: [a b c 0 0 0 0 0].

Entretanto a esteganografia por LSB é muito vulnerável, pois é possível detectar e extrair a informação escondida com facilidade utilizando algumas técnicas de esteganálise. E portanto, foi inserido ao sistema de esteganografia um processo simples de embaralhamento. Neste caso, a esteganografia foi implementada em apenas um ou nenhum pixel para cada bloco de tamanho 3x3 pixels da imagem da seguinte forma:

1	4	7
2	5	8
3	6	9

e o valor 0 indica que não existe nenhum pixel que foi esteganografado naquele bloco. Esta posição é obtida externamente utilizando uma chave decimal de forma circular. E cada bloco é analisado sem que haja sobreposição entre eles. Já os blocos da borda que não possuírem tamanho 3x3 foram desconsiderados, ou seja, não possuem informação esteganografada. Concatenado à informação esteganografada, foi embutido um *hash* MD5 com 128 bits para que possamos verificar se a informação transmitida foi modificada. Este processo aplicará uma marca d'água digital no vídeo, de forma a possibilitar a verificação da sua integridade. Neste projeto deve-se implementar a extração da esteganografia e o desembaralhamento em *threads* separadas.

Neste caso, iremos utilizar um programa externo relativamente comum no Linux denominado *md5sum*, mas por segurança, **não será permitido o uso da chamada de sistema system().** O comando *md5sum nome_do_arquivo* gera como saída no standard output o valor do *hash* e o nome do arquivo de saída (nome_do_arquivo).

Mas, antes de enviar a imagem escondida para o servidor, o sistema deve verificar a integridade do sinal e em seguida inserir um cabeçalho à imagem processada. O formato que será utilizado é o PGM (*portable graymap*), que contém um cabeçalho e a matriz correspondente da imagem. O cabeçalho pode ser feito inserindo a seguinte informação: P5 W H M I,

onde P5 indica o formato PGM, W = largura em pixels, H = altura em pixels, M = valor máximo do pixel e I = imagem desesteganografada. Neste caso, o valor dos pixels são representados por um byte, ou seja, variam entre 0 e 255.
Uma funcionalidade básica que o sistema cliente deve ter é informar o número do processo e em que etapa do processo a execução se encontra ao enviar um sinal para o mesmo, juntamente com a estimativa de porcentagem que já foi executado além de mostrar localmente um resultado parcial da imagem extraída da esteganografia. Por fim, ao receber o sinal de interrupção SIGINT, o código deve abortar o processamento inserindo zeros até o preenchimento dos pixels da imagem processada, em seguida inserir o cabeçalho e enviar ao servidor a imagem parcialmente processada.
No caso do servidor, o programa deve solicitar ao cliente um resultado parcial juntamente com uma estimativa de porcentagem do processo de extração que já foi executado ao pressionar ctrl+C.
O código do servidor deve rodar em um PC e o código do cliente no Raspberry Pi. E o envio de informação para o servidor deve ser por meio de um protocolo que maximiza a obtenção de informações não corrompidas.
Descreva no relatório se existe alguma vantagem (ou desvantagem) em utilizar um RTOS (sistema operacional em tempo-real). Explique como o projeto em questão poderia ser estruturado para tirar vantagem de um RTOS, como por exemplo, o Linux+Xenomai.

Observações:

1) **O *hash* MD5 do vídeo é calculado utilizando o vídeo com a imagem esteganografada e os valores correspondentes ao *hash* zerados.**
2) **É possível visualizar os quadros do vídeos colocando cabeçalhos em cada imagem (quadro).**
3) **O aluno deve indicar o processo de compilação, utilização e chamada para os programas.**
4) **O servidor também deve ser implementado.**
5) **Alguns arquivos de teste deverão ser baixados da página do curso.**