

Segurança Computacional
Alexandre Souza Costa Oliveira
170098168
Universidade de Brasília - UnB

Trabalho 02:

1. Compilação e utilização:

A implementação do código para AES foi feita em **C++** com gcc 10.3.0 em ambiente Windows (versão 10).

Para compilação do arquivo, basta entrar na pasta dele com o cmd ou powershell e digite a linha de comando:

```
g++ aes.cpp main.cpp -o main
```

A imagem que deseja cifrar ou decifrar deve estar no formato Bitmap 24 bits e deve ser colocada no diretório do main.exe e a senha deve ser colocada no arquivo key.txt com um tamanho de 16 caracteres (16 bytes).

Para executar o programa, digite no cmd ou powershell estando dentro da pasta do arquivo main.cpp:

```
./main
```

Siga os passos da interface para cifrar ou decifrar sua imagem e quando estiver concluído, a imagem aparecerá cifrada ou decifrada no diretório do main.exe com o nome cifrado.bmp ou decifrado.bmp.

Não se esqueça de especificar o formato .bmp ao digitar o nome da imagem que deseja cifrar ou decifrar e também não esqueça de salvar a hash gerada no modo CTR para uso posterior caso queira decifrar.

2. Sobre o trabalho:

Neste trabalho será abordado a implementação do algoritmo de cifra de bloco, Advanced Encryption Standard (AES) com os modos de operação ECB e CTR. O AES foi criado com o objetivo de substituir o algoritmo DES, pois o mesmo foi quebrado. Com o AES é possível cifrar blocos de 128, 192 e 256 bits, utilizando uma chave secreta e neste trabalho estaremos utilizando blocos de 128 bits.

Para transformar um texto simples em um texto cifrado, o AES realiza algumas rodadas com operações para criar um texto cifrado seguro e essas operações são:

- **AddRoundKey** - nessa operação, cada byte do bloco de texto é combinado com um byte da chave secreta daquela rodada.
- **SubBytes** - substitui os bytes do bloco de texto por outros bytes presentes em uma tabela pré calculada.
- **ShiftRows** - desloca os bytes para a esquerda, ou para a direita caso queira decifrar.
- **MixColumns** - operação de difusão, onde é realizada uma transformação linear invertível trocando 4 bytes de cada coluna.

O AES puro é capaz de realizar a cifra apenas do bloco determinado, então quando temos um arquivo ou texto grande, ele não conseguirá sozinho cifrar isso. Para isso é

utilizado os modos de operação ECB e CTR, onde o ECB é considerado um modo de operação não seguro, já que blocos idênticos são cifrados de maneiras idênticas, sendo possível assim quebrar a cifra após analisar os padrões. Já o CTR é mais seguro que o ECB, pois para cada bloco de texto simples a ser cifrado, ele irá incrementar um contador junto com uma hash para garantir que cada bloco seja cifrado de maneira diferente aos anteriores.

3. Implementação:

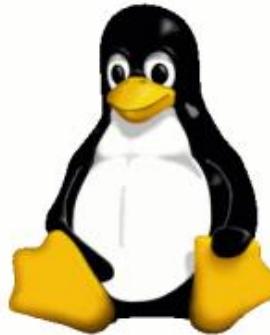
Neste trabalho, foi criada uma classe chamada Aes para realizar todas as operações do Aes em conjunto com o ECB ou CTR. O algoritmo cifra uma imagem do formato Bitmap 24 bits. O resultado poderá ser visto no mesmo diretório do algoritmo.

O conjunto de métodos da classe AES são:

```
Faz a combinação entre o bloco e chave atuais  
void AddRoundKey(unsigned char *bloco, unsigned char *keyBloco);  
Faz a substituição de bytes pela tabela pré calculada sBox:  
void SubBytes(unsigned char *bloco);  
Faz o deslocamento para esquerda:  
void ShiftRows(unsigned char*bloco);  
Faz a difusão com transformação linear invertível:  
void MixColumns(unsigned char *bloco);  
Cria a expansão de chave para no total de 14 rodadas:  
void KeySchedule(unsigned char *key);  
  
Função inversa de SubBytes:  
void InverseSubBytes(unsigned char *bloco);  
Função inversa de ShiftRows (desloca para direita):  
void InverseShiftRows(unsigned char*bloco);  
Função inversa de MixColumns baseada em cálculo da Wiki:  
void InverseMixColumns(unsigned char *bloco);  
  
Determina se será cifrado ou decifrado com modo ECB ou CTR  
void SetCTR(bool value);  
  
Faz a combinação da chave com o contador e hash garantido chave diferentes  
em CTR:  
void KeyCounter(unsigned char *chaveExpandida, unsigned char *hash, int  
índice);  
Gera uma hash de 16 bytes aleatório:  
void GenerateHash(unsigned char *hash);  
  
Realiza a cifra em ECB ou CTR:  
void Cifrar(string texto, string chave, int rounds);  
Realiza a decifração em ECB ou CTR:  
void Decifrar(string texto, string chave, int rounds);
```

Alguns testes foram realizados com 1, 5, 9 e 13 rodadas, utilizando 3 imagens diferentes com os dois modos implementados, ECB e CTR. Foi utilizada duas imagens com detalhes mais minimalistas e uma selfie, pois, poderemos visualizar como o ECB é menos seguro em imagens mais minimalistas (menos detalhes e diferenças de bits). Os resultados podem ser vistos abaixo:

Imagens utilizadas:



Modos / Rodadas	1	5	9	13
ECB	A highly noisy, multi-colored version of the penguin image, showing significant loss of detail.	A slightly less noisy version than 1 rodada, but still very grainy.	The noise level is reduced further, though it remains quite grainy.	The noise level is significantly reduced, making the penguin's features more recognizable.
CTR	A version of the penguin image where horizontal bands of noise are visible, indicating a different type of corruption.	The horizontal banding is reduced compared to 1 rodada.	The horizontal banding is almost entirely removed.	The image appears mostly noise-free.
ECB	A highly noisy version of the abstract drawing, with colors appearing as small dots.	The drawing is partially visible through the noise.	The drawing is more clearly visible.	The drawing is very clear and recognizable.
CTR	A version of the abstract drawing where colors are washed out and appear as uniform shades.	The colors are more distinct but still lack depth.	The colors are well-defined.	The colors are vibrant and the drawing is clear.

ECB (selfie)				
CTR (selfie)				

Para cifrar no modo ECB, foi implementado para que o texto simples fosse dividido em pedaços de 16 bytes cada e assim o algoritmo Aes é aplicado em cada um desses pedaços, com a mesma expansão de chave. Isso implica que, blocos de texto idênticos serão cifrados de maneira idêntica, causando assim uma falha de segurança, e isso pode ser visualizado nos resultados acima.

Já o modo CTR, foi implementado também dividindo o texto simples em pedaços de 16 bytes, mas com o adicional de uma hash e um contador de blocos. A cada bloco, a expansão de chave é combinada com o hash e o contador, assim criando uma expansão de chave única para cada bloco de texto simples. A hash é retornada ao usuário em formato de hexadecimal para facilitar na leitura.

Interface do programa:

```
Bem vindo ao Cifrador 2000
Para realizar a cifra, voce precisara de uma imagem no formato .bmp no diretorio do programa.

Nome do arquivo BMP: alex.bmp
Lendo arquivo...
Leitura concluida.

Qual modo deseja utilizar?
1. ECB
2. Decifrar ECB
3. CTR
4. Decifrar CTR
Opcao: 3
Quantos rounds ?
Resposta: 13
Cifrando no modo CTR...
Gerando Hash...
Sua hash eh: b4 e 63 76 96 b1 93 2e 5b 5e 7d 8a c2 5e dc c6
Cifrado com sucesso, arquivo gerado: result.bmp
PS C:\Users\Alexandre\Documents\UnB\SC\Trabalhos\AES> |
```

4. Dificuldades:

Pelo fato do algoritmo AES utilizar matrizes, foi relativamente difícil de entender como exatamente funciona a aplicação das operações de cada rodada, como MixColumns e ShiftRows. Foi necessário entender primeiro como eram formadas as linhas e colunas e assim o código teve que ser alterado diversas vezes para que realmente funcionasse da maneira correta. O método de MixColumns envolve uma multiplicação de matrizes, mas com uma matemática diferente da convencional, utilizando xor e polinômios. Para que esse método realmente funcionasse neste trabalho, foi necessário utilizar tabelas pré-calculadas de valores para realizar a multiplicação (xor).

O trabalho em si era para ter sido feito com imagens de formato JPEG. Porém o cabeçalho deste formato é muito confuso, e então foi realizado uma adaptação para funcionar com imagens Bitmap.

Não consegui utilizar corretamente o OpenSSL para gerar uma hash e então foi implementado para gerar uma “pseudo” hash com 16 bytes gerados aleatoriamente. (O extremo calor de 34°C dificultou muito a concentração nessa parte)

5. Referências:

Tabelas pré-calculadas e explicação sobre MixColumns:

https://en.wikipedia.org/wiki/Rijndael_MixColumns#Implementation_example

Tabela S-Box e sua inversa:

https://en.wikipedia.org/wiki/Rijndael_S-box

Explicação geral sobre AES:

https://www.youtube.com/watch?v=04HK1UHmxcs&ab_channel=AvelinoMorganti

Assinaturas de arquivos:

https://en.wikipedia.org/wiki/List_of_file_signatures

Explicação sobre expansão de chave:

https://www.brainkart.com/article/AES-Key-Expansion_8410/

Um pouco sobre ECB e CTR:

[https://pt.wikipedia.org/wiki/Modo_de_opera%C3%A7%C3%A3o_\(criptografia\)](https://pt.wikipedia.org/wiki/Modo_de_opera%C3%A7%C3%A3o_(criptografia))