Modos de Cifragem em Bloco utilizados pelo DES

Data Encryption Standard (DES) é um sistema de codificação simétricas por blocos de 64 bits, dos quais 8 bits (um byte) servem de teste de paridade (para verificar a integridade da chave). O algoritmo efetua combinações, substituições e permutações entre o texto a ser codificado e a chave, de modo com que as operações possam ser feitas nos dois sentidos (para a descodificação). Diversos modos de operação foram criados, tais como:

1- Electronic CodeBook (ECB)

No modo Electronic CodeBook, a mensagem é dividida em blocos, sendo cada um criptografado separadamente.

A desvantagem é que blocos idênticos de texto plano (texto puro), que são conteúdos de um arquivo sequencial ordinário legível como material textual sem muito processamento, são criptografados em blocos de texto cifrado idênticos; assim, ele também não oculta padrões de dados, ou seja, não oferece uma perfeita confidencialidade de mensagem.

2- Cipher Block Chaining (CBC)

No modo Cipher Block Chaining, a cada bloco de texto simples é aplicada uma função XOR junto com o bloco cifrado anterior antes do texto ser criptografado. Desta forma, cada bloco cifrado fica dependente de todos os blocos de texto simples processados até este momento.

Suas principais desvantagens são que a criptografia é sequência, a mensagem deve ser alinhada de acordo com um múltiplo do tamanho do bloco de cifra e toda a validade de blocos anteriores está contida no bloco de texto cifrado imediatamente anterior.

3- Cipher FeedBack (CFB)

O modo Cipher FeedBack, em contraste com Cipher Block Chaining (CBC) que criptografa um número definido de bits de texto simples por vez, às vezes é desejável criptografar e transferir alguns valores de texto simples instantaneamente, um de cada vez, para o qual o método de "feedback de texto cifrado" é uma das opções, onde o mesmo também faz utilização de um vetor de inicialização

Sua principal vantagem é a opção de recuperação de erros, incluindo erros que adicionam ou excluem blocos de texto cifrado. Outra vantagem é que o processo de descriptografia também usa a cifra de bloco; dependendo de quão diferente é a criptografia e descriptografia de cifra de bloco, isso pode ser conveniente para uma maior segurança.

4- Output FeedBack (OFB)

O modo Output FeedBack possui algumas semelhanças com o modo CFB, pois permite a criptografia de diferentes tamanhos de bloco, mas tem a diferença fundamental de que a saída da função de bloqueio de criptografia é o feedback (em vez do texto cifrado). Em termos de correção de erros, pode tolerar erros de bit de texto codificado, mas é incapaz de se auto-sincronizar depois de perder bits de texto codificado, pois perturba a sincronização do fluxo de chaves de alinhamento.

Um dos problemas desse método é que com a realimentação de saída é que o texto simples pode ser facilmente alterado, mas o uso de um esquema de assinatura digital pode superar esse problema.

5- Counter (CTR)

O modo Counter transforma uma cifra de bloco em uma cifra de fluxo. Ele gera o próximo bloco de keystream, criptografando valores sucessivos de um "contador". O contador pode ser qualquer função que produza uma sequência que tenha a garantia de não se repetir por um longo tempo.

Algumas de suas vantagens são: Eficiência de software e hardware, pré-processamento, pois o trabalho criptográfico na codificação de uma mensagem M é independente de M, onde é usado em alguns ambientes para aumentar a velocidade. Isto é, pode-se calcular o bloco em "ciclos de reposição", mesmo antes de conhecer o texto e as características de eficiência anteriores não são obtidas à custa de segurança. Além disso, ao contrário de outros modos, o manuseio de mensagens de comprimento arbitrário de bits é trivial. Nenhuma parte é desperdiçada ao fazer isso, o texto cifrado C é de o mesmo comprimento que o texto M.

Algumas de suas desvantagens são: não fornece integridade da mensagem; ocorre erro de propagação; suscetível a uso de uso, tal como o usuário reutilizar o “contador” e pode ter interação com cifras fracas.

Implementação do RC4

A principal dificuldade foi entender (e assim implementar) a manipulação com a chave e perceber que o XOR poderia ser feito de forma bem simples em Python. Sua execução se baseia em executar somente o arquivo “main.py” e selecionar a opção desejada. A chave e as mensagens estão em um arquivo de texto. Os testes feitos se encontram no arquivo anexado, onde os arquivos contém a chave, a mensagem, a mensagem criptografada e a mensagem descriptografada.

Implementação do S-DES

Apesar de ter entendido a teoria, no desenvolvimento a dificuldade foi como implementar a criptografia. Outro fator, foram fatores pessoais em relação ao tempo.

Chat Seguro

As principais dificuldades da implementação do RC4 foram inicialmente o desenvolvimento do Chat em si, levando em conta que teve de ser aprimorado o conhecimento de socket e funcionamento de servidor-cliente. O nome de cada cliente fica guardado na própria instância do “client” e os processos de criptografias são feitas pelo lado do cliente, assim no servidor a mensagem já é recebida criptografada.

Para a execução, deve-se iniciar servidor e após isso, os clientes. Primeiramente digitar o nome, caso queira sair, digitar “{quit}”; para mudança de chave, colocar “{chave}” seguido do tipo: “1” para RC4 e “2” para S-DES e em seguida a nova chave, ambos os casos sem espaço.

No presente momento do envio dessa tarefa, só consta a forma RC4 para a criptografia.

Segue em anexo 2 imagens do chat em funcionamento, na primeira imagem inserindo a mudança de chave é a outra o início ao fim da conversa é saindo da mesma: