



**UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE - UFRN**  
**INSTITUTO METRÓPOLE DIGITAL - IMD**  
**BACHARELADO EM TECNOLOGIA DA INFORMAÇÃO - BTI**

**MATHEUS RANGEL DE MELO - 20170070014**  
**MAX WILLIAM SOUTO FILGUEIRA - 20170061294**

**RELATÓRIO CHAT SEGURO**

**NATAL, 2019**

<b>Modos de Operação de Cifras</b>	<b>2</b>
Electronic CodeBook	2
Cipher Block Chaining	2
Cipher Feedback	2
Output Feedback	3
Counter	3
<b>Chat Seguro Implementação</b>	<b>3</b>
S-DES	3
RC4	3
Diffie Hellman	3
Chat	3
<b>Dificuldades</b>	<b>4</b>

## Modos de Operação de Cifras

### Electronic CodeBook

Considerado o modo mais simples de criptografia, a mensagem é dividida em blocos e cada bloco é criptografado separadamente. Uma grande desvantagem desse método é que blocos idênticos da mensagem também são idênticos na mensagem cifrada, não ocultando padrões de dados. No geral esse método não é recomendado e não oferece nenhuma vantagem além da simplicidade.

### Cipher Block Chaining

Nesse modo em cada bloco do texto simples é aplicada uma função XOR junto com o bloco cifrado anterior, assim todos blocos processados dependem do resultado cifrado do bloco anterior.

A principal vantagem com isso é que mesmo que dois blocos idênticos da mensagem original dificilmente gerarão o mesmo bloco cifrado, escondendo melhor os padrões de blocos que possam existir na mensagem original.

A desvantagem é que o processo de cifragem e decifragem não pode ser paralelizado.

### Cipher Feedback

A mensagem é processada em blocos de tamanho predefinido de bits. O texto cifrado anterior é usado como entrada para o algoritmo criptográfico, aumentando a aleatoriedade da cifra gerada.

## Output Feedback

Similar o cipher feedback, porém como entrada do algoritmo criptográfico é saída DES fazendo um XOR bit a bit.

## Counter

Similar aos métodos anteriores porém é usado o valor de um contador criptografado. A cada bloco o contador é incrementado. Vantagem desse método é que permite a paralelização do algoritmo de cifragem pois o resultado de um bloco não depende do bloco anterior.

# Chat Seguro Implementação

O chat foi implementado com Python e a abordagem escolhida foi cliente servidor. Os clientes se conectam no servidor podendo ter inúmeros clientes conversando entre si.

## S-DES

O primeiro algoritmo para criptografia implementado foi s-des, a cifragem é feita byte a byte do bytearray que é passado como parâmetro junto com a chave que deve ser utilizada, como é utilizado apenas os 10 primeiros bits da chave decidimos utilizar apenas um inteiro para representar a chave ignorando os 22 bits restantes do 32.

## RC4

Seguindo o mesmo padrão de entrada do S-DES, o RC4 recebe um bytearray e um chave inteira como entrada e retornar o bytearray cifrado.

## Diffie Hellman

Para o diffie Hellman decidimos implementar uma classe para encapsular os valores das chaves privada, pública e a gerada pelo algoritmo, ele foi bastante simples de implementar tendo em vista que a responsabilidade é permitir que o servidor e cliente compartilhem a chave que será utilizada no RC4 ou S-DES de forma segura.

## Chat

Utilizamos o código de um chat previamente implementado(fonte: <https://pythonprogramming.net/sockets-tutorial-python-3/>) e apenas adicionamos uma camada para criptografar e descriptografar as mensagem.

O servidor roda por padrão em todas as interfaces disponíveis e na porta 1234, mas pode ser facilmente alterado modificando no código do server.py e do client.py.

Como não foi implementada uma interface gráfica para o chat, as mensagens não aparecem em tempo real para os clientes pois o terminal fica aguardando o input do usuário, ou seja, as mensagens enviadas são recebidas apenas quando o cliente aperta enter liberando o loop do input.

Para executar o servidor e cliente é necessário ter apenas o python 3.7 instalado, e executar o python server.py e o python client.py respectivamente, vários clientes podem se conectar no servidor e conversar entre si.

Assim que executar o cliente deve ser informado um username qualquer que será utilizado no chat, após isso a chave é compartilhada através do Diffie Hellman e o S-DES é utilizado para criptografar as mensagens.

```
Accepted new connection from 127.0.0.1:52512, username: Lucas
Received message from Lucas: bytearray(b'Oi pessoal')
107
Accepted new connection from 127.0.0.1:52513, username: Matheus
96
Accepted new connection from 127.0.0.1:52514, username: Max
Received message from Lucas: bytearray(b'Oi pessoal')
Received message from Matheus: bytearray(b'Eae man')
Received message from Lucas: bytearray(b'Funcionou!!!')
Received message from Max: bytearray(b'Incrivel!!!!!!!!!!!!!!')
```

```
C:\Users\mathe\PycharmProjects\ChatSeguro>python client.py
Username: Max
Max >
Lucas > Oi pessoal
Matheus > Eae man
Lucas > Funcionou!!!
Max > Incrivel!!!!!!!!!!!!!!
Max >
```

## Dificuldades

Não tivemos.