# Lab 07

Auditoria e Segurança de SI

Prof. Roberto Cabral

4 de maio de 2023

# Criptografia Simétrica com Openssl

Este Lab consiste em utilizar o openssl como ferramenta para um melhor entendimento da criptografia simétrica. A biblioteca openssl implementa funções criptográficas básicas e disponibiliza várias funções utilitárias. A implementação é considerada eficiente e segura e está disponível em praticamente todas as distribuições Unix/Linux e Mac OS; mais detalhes podem ser encontrados em: `http://www.openssl.org/`

## Codificação e decodificação base64

O método de codificação `base64` é muito utilizado para transferência de dados binários por meios de transmissões que lidam apenas com texto, como por exemplo para enviar arquivos anexos por e-mail.
Para converter um arquivo `teste.pdf` na sua representação `base64`, deve-se utilizar o seguinte comando :

```
base64 < teste.pdf
```

Para fazer a transformação reversa, deve-se utilizar o seguinte comando:

```
base64 -d < arquivo.base64
```

Em ambos os casos, o resultado será gerado no terminal.
**Obs.:** `base64` **não é um método de criptografia.**

## Visualização de arquivos binários

Para visualizar o conteúdo de arquivos binários, sugiro o uso dos comandos `hexdump` ou `xxd`.

```
hexdump arquivo.bin
xxd arquivo.bin
xxd -b arquivo.bin
```

Para editar o conteúdo, pode-se utilizar o `hexedit`.

## Encriptação Simétrica com OpenSSL

A encriptação simétrica é feita com o subcomando `enc`. Digite `openssl enc --help` no terminal parar ver as opções.
Neste laboratório, utilizaremos os seguintes algoritmos:

- AES convencional com chaves de 128 bits: `-aes-128-ecb`
- AES convencional com chaves de 256 bits: `-aes-256-ecb`

Esses parâmetros devem ser passados ao OpenSSL. ECB indica o modo de uso Electronic Codebook (ECB), onde cada bloco do texto claro é encriptado com a chave criptográfica.
A flag -K indicar a chave criptográfica a ser utilizada. O parâmetro espera que a chave seja passada em hexadecimal. Por exemplo, para encriptar um texto no arquivo textoClaro.txt com a chave (hexa) 34e2b0d2fd7ac13814ceb3c750a726f2 (chave de 128 bits), deve-se utilizar o comando:

```
openssl enc -aes-128-ecb -in textoClaro.txt -out
textoEncriptado.bin -K 34e2b0d2fd7ac13814ceb3c750a726f2
```

A decriptação funciona de forma similar, mas passando o parâmetro **-d**.

## Geração de Chaves

O openssl disponibiliza um gerador de números aleatórios confiável através do comando `rand`. Ele recebe como parâmetro o numero de bytes do número. Para usar com o comando anterior de encriptação, é necessário que a chave seja exibida em hexadecimal (usando o parâmetro -hex). Por exemplo, para uma chave de 128 bits pode ser gerada pelo comando a seguir:

```
openssl rand -hex 16
```

## Atividade

1. Gerar duas chaves de encriptação para AES 128 bits ou 256 bits.
2. Encriptar um arquivo de texto qualquer (exemplo) e uma das chaves criadas.
3. Decriptar o texto utilizando diferentes chaves e analisar o resultado.
4. Modificar alguns bits do texto encriptado e tentar decriptá-lo. Analisar o resultado.
5. Construir um arquivo de texto que possua um padrão que se repete várias vezes. Observar o padrão no arquivo encriptado.

6. Enviar seu nome encriptado (em base64) para o professor usando o algoritmo `aes-128-ecb` e a senha, disponível na pasta do drive compartilhada entre você e o professor.

Deverá ser feito um relatório descrevendo as atividades realizadas. As chaves simétricas usadas devem ser apresentadas no relatório no formato base64. Os arquivos usados na encriptação e decriptação devem ser disponibilizados na pasta do Drive compartilhada com o professor [1]. No Moodle deve ser enviado apenas o relatório (em PDF). O relatório deverá ser entregue até o dia 10 de Maio de 2023.

**Obs.: A atividade é individual.**
**Obs2.: Atividade baseada no laboratório 2.1 do Professor Ricardo da Rocha, disponível em** `http://ww2.inf.ufg.br/~ricardo/cripto/`**.**

[1] `https://drive.google.com/drive/folders/1oh7xGTiyT7NMIodZoRuXwWJqJMmgdAt4?usp=share_link`