

Criptografia Simétrica e Assimétrica, e Hash

Segurança da Informação

Charles Tim Batista Garrocho

Instituto Federal de São Paulo – IFSP
Campus Campos do Jordão

garrocho.ifspcj.o.edu.br/SEGA6

charles.garrocho@ifsp.edu.br

Curso Superior de TADS



INSTITUTO FEDERAL

Criptografia (do grego esconder+escrever).

A criptografia existe desde a antiguidade, e estava normalmente associada a **atividades militares** e diplomáticas.

A encriptação é o processo de **transformação** de uma informação original, numa informação ilegível, para terceiros.

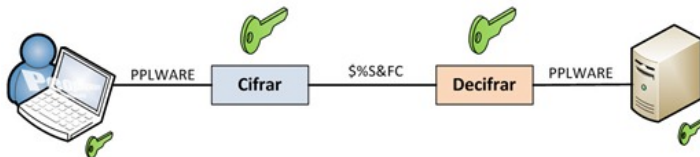
Este mecanismo tem como objetivo o envio de informação confidencial de forma **segura**, sendo apenas possível a sua decodificação por pessoas autorizadas.



Criptografia Simétrica

A criptografia simétrica é também conhecida por **criptografia de chave secreta**. DES , 3DES , AES e RC4 são alguns dos algoritmos que usam criptografia simétrica.

Funcionamento: É usada uma única chave que é compartilhada entre o emissor e o receptor. Desta forma, a chave que é usada para cifrar é a mesma que é usada para decifrar.



INSTITUTO FEDERAL

Criptografia Simétrica em Python

```
from Crypto.Cipher import AES  
  
obj = AES.new('Esta é a chave1', AES.MODE_ECB)  
  
mensagem = "Você está azul"  
  
texto_criptografado = obj.encrypt(mensagem)  
  
texto_criptografado  
'Kmq\xe5\x8dAD\xf0\xfcW\x10\x19\xb9\x16\xd0&'  
  
obj2 = AES.new('Esta é a chave1', AES.MODE_ECB)  
  
obj2.decrypt(texto_criptografado)  
'Você está azul'
```

Você deve utilizar uma mensagem com quantidade de caracteres **múltiplo** do tamanho da quantidade de caracteres da chave.

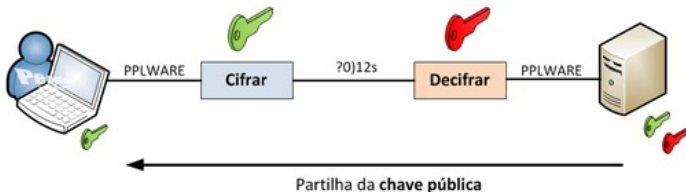


INSTITUTO FEDERAL

Criptografia Assimétrica

A criptografia assimétrica é também conhecida por **criptografia de chave pública**. RSA, DSS e ElGamal são alguns dos algoritmos que usam criptografia assimétrica.

Funcionamento: É usado um par de chaves distintas. Uma **chave pública** é usada para cifrar. Uma **chave privada** é usada para decifrar.



INSTITUTO FEDERAL

Criptografia Assimétrica em Python

```
from Crypto.PublicKey import RSA

chave_privada = RSA.generate(1024)
chave_publica = chave_privada.publickey()
texto_criptografado = chave_publica.encrypt('A mensagem', 32)

texto_criptografado
"\x8dm9\xb47\xd1n\xfc\xd3\x9b\xf0\x8e\xb3\x9c\xc9:\x86\
xa2U\xa0\x80'\xd1%\xef^\xa5\x10\x14U-Hs\x00@_H\xa2\
x17+U.\xb1\x0bs\xe5\xcf\xb9\xc3+\x9c\xd9\x1c\xb8\x8d\xb3q\
x0e\xaa\x82\xf4\x80\x98\xdc r&p\x0c\xf5\xe8\x9f\xf3\x1f\
xbf\xb2\x7f\xb6\>*\x97}\t?d(\xe19\x00\xea\x96\xa2\xf3\
xf2\xdd9\xcc\xfc\x04\xbb\xf1\xad'\xc6@)X:\x86\xd6\xba\
x12\xf6&n#V\x0c\xf0H\xd7\xc78\x18\x8cESq"

chave_privada.decrypt(texto_criptografado)
'A mensagem'
```

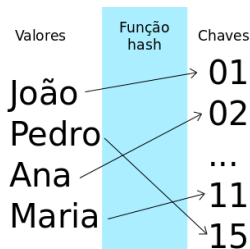
Apenas a chave pública é compartilhada.



INSTITUTO FEDERAL

Função Hash

Uma função hash é um **método criptográfico** que, quando aplicado sobre uma informação, independentemente do tamanho que ela tenha, gera um resultado único e de tamanho fixo, chamado **hash**.



Podemos utilizar hash para: Verificar a integridade de arquivos, e ou gerar assinaturas digitais.



INSTITUTO FEDERAL

MD5 (Message-Digest algorithm 5)

MD5 é uma **função hash criptográfica** de 128 bits unidirecional, descrito na RFC 1321.

Muito utilizado por softwares com protocolo P2P na verificação de **integridade** de arquivos e logins.

```
import md5
m = md5.new()
m.update("A mensagem")
m.digest()
'=5\xe6\xa7I\x1b1.\xf9\x14\x964\xa4\xae~\xce'
```



INSTITUTO FEDERAL

SHA-1 e SHA-2 (Secure Hash Algorithm): Desenvolvido pelo NIST e NSA. Já foram exploradas falhas no SHA.

Projeto em Python (<https://docs.python.org/2/library/hashlib.html>)

SHA-3 (Secure Hash Algorithm): Desenvolvido pelo NIST e NSA. Não possui falhas.

Projeto em Python (<https://pypi.python.org/pypi/pysha3>)

WHIRLPOOL: essa função foi adotada pelo ISO e IEC como parte do padrão internacional ISO 10118-3.

Projeto em Python (<https://github.com/radiosilence/python-whirlpool>)

