Criptografia Simétrica e Assimétrica, Hash, e Assinatura Digital

Segurança da Informação

Charles Tim Batista Garrocho

Instituto Federal de São Paulo – IFSP
Campus Campos do Jordão

garrocho.ifspcjo.edu.br/SEGA6

charles.garrocho@ifsp.edu.br

Curso Superior de TADS

Introdução

Criptografia (do grego esconder+escrever).

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

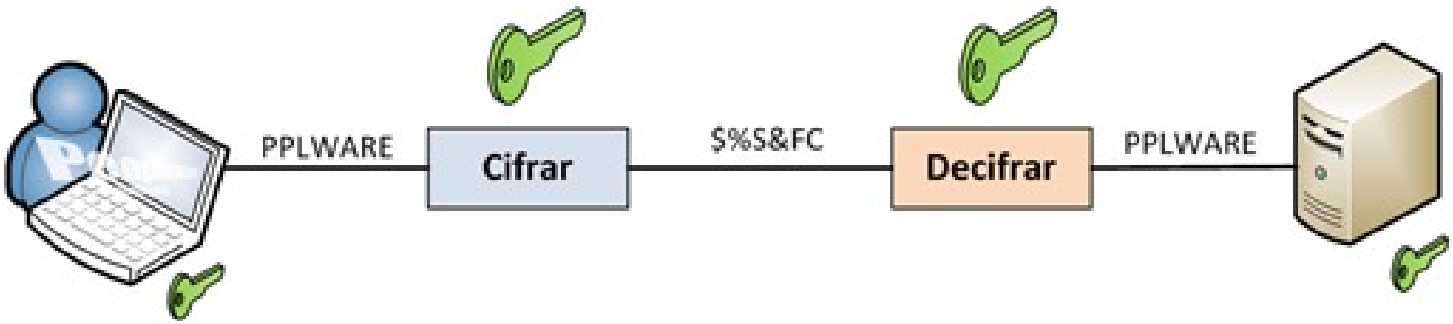
A encriptação é o processo de **transformação** de uma informação original, numa informação ilegível, para terceiros.

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 partilhada entre o emissor e o receptor. Desta forma, a chave que é usada para cifrar é a mesma que é usada para decifrar.



Criptografia Simétrica em Python

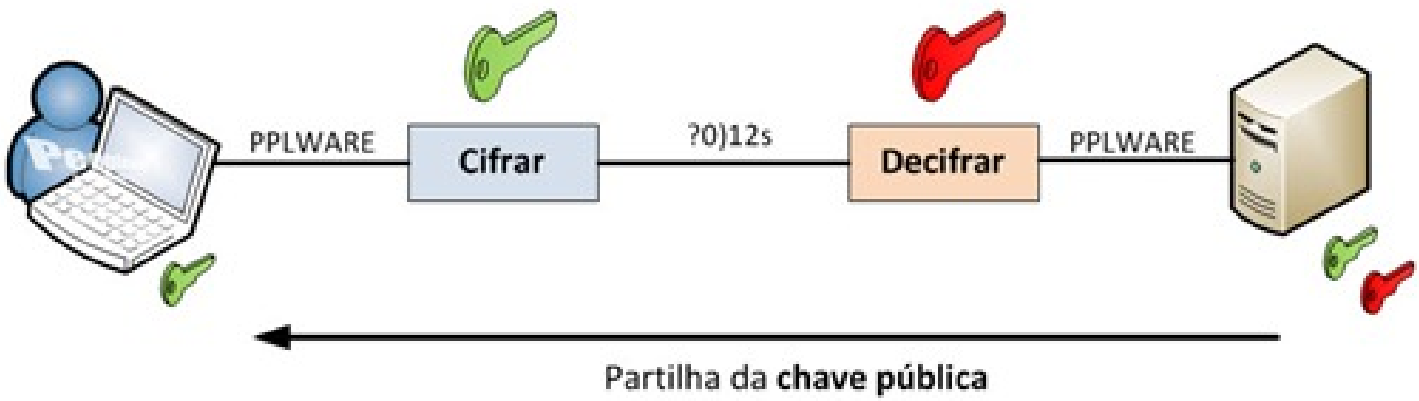
```
from Crypto.Cipher import AES

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.

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.



Criptografia Assimétrica em Python

```
from Crypto.PublicKey import RSA

chave_privada = RSA.generate(1024)
chave_publica = key.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\xdcr&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.

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.

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'
```

Outros Algoritmos de Hash

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)

Certificado Digital

Os computadores e a Internet são largamente utilizados para o processamento de dados e para a troca de mensagens e documentos entre **cidadãos, governo e empresas**.

No entanto, estas transações eletrônicas necessitam da adoção de **mecanismos de segurança** capazes de garantir autenticidade, confidencialidade e integridade às informações eletrônicas.

A **certificação digital** é a tecnologia que provê estes mecanismos. No cerne da certificação digital está o certificado digital, um documento eletrônico que contém o nome, e um número público exclusivo denominado chave pública.

A **chave pública** serve para validar uma **assinatura** realizada em documentos eletrônicos.

INSTITUTO FEDERAL

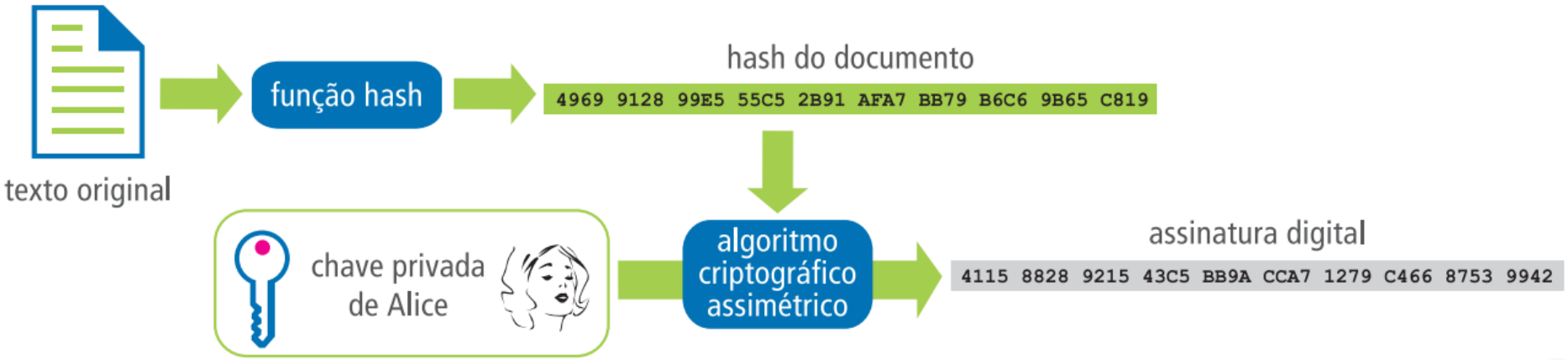
Certificado Digital

A utilização de certificado digital proporciona:

- **Privacidade**: Garantia de que as informações trocadas nas transações eletrônicas não serão lidas por terceiros.
- **Integridade**: Garantia de que as informações trocadas nas transações eletrônicas não foram alteradas desde que foram assinadas.
- **Autenticidade**: Garantia de identidade da origem e destino da transação.
- **Assinatura Digital**: Assinatura eletrônica baseada em métodos criptográficos que é gerada a partir de um conjunto de regras e que atribui ao documento a possibilidade de confirmar, com segurança, sua integridade e a identificação do autor do documento eletrônico.
- **Não Repúdio**: É a garantia de que somente o titular do Certificado Digital poderia ter realizado determinada transação, impedindo que os integrantes de uma transação venham a contestar ou negar uma transação após sua realização.

Assinatura Digital

O mesmo método de autenticação dos algoritmos de criptografia de chave pública operando em conjunto com uma função resumo, também conhecido como função de hash, é chamada de **assinatura digital**.



Este pode ser comparado a uma **impressão digital**, pois cada documento possui um valor único de resumo e até mesmo uma pequena alteração no documento, como a inserção de um espaço em branco, resulta em um resumo completamente diferente.

Assinatura Digital

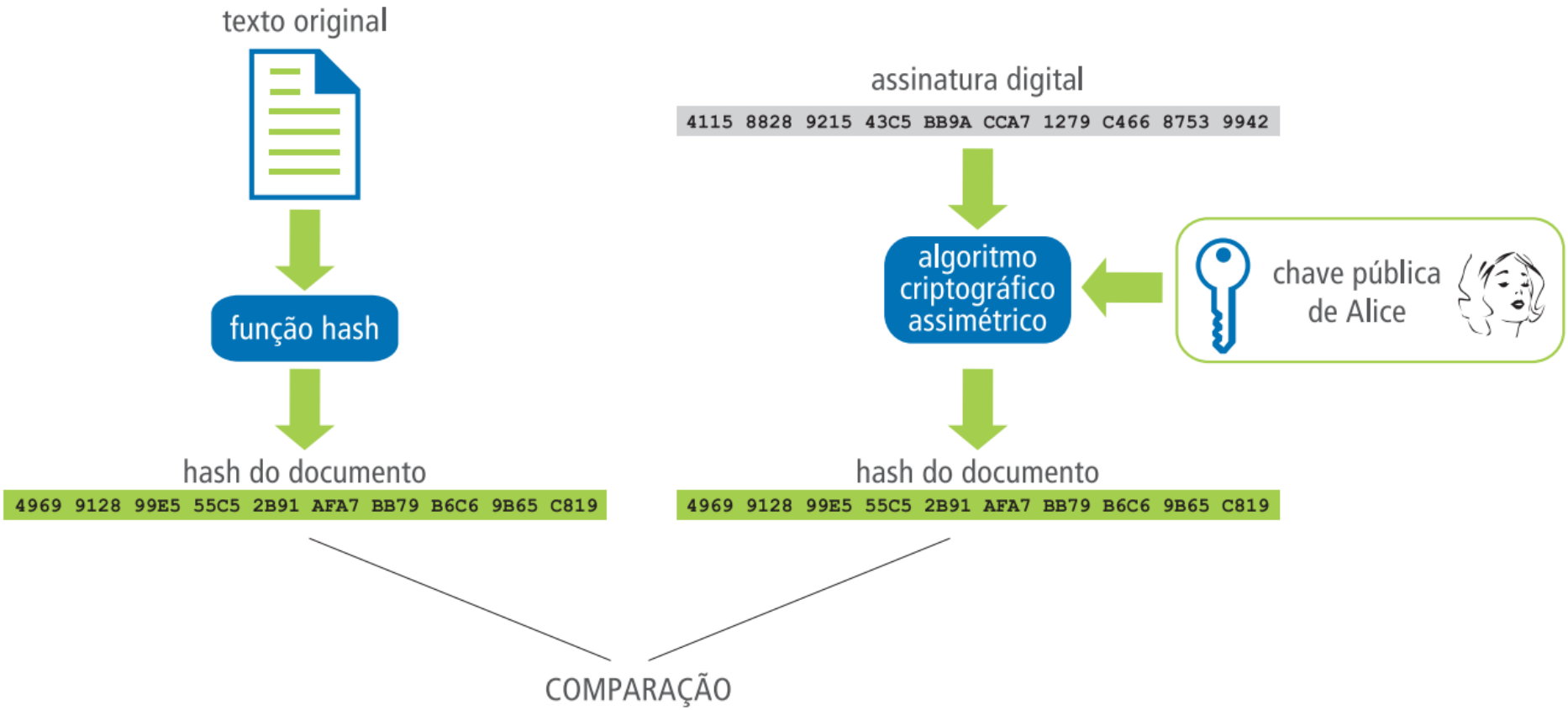
Para **comprovar** uma assinatura digital é necessário inicialmente realizar algumas operações.

Calcular o resumo criptográfico do documento e decifrar a assinatura com a chave pública do signatário.

Se forem iguais, a assinatura está correta, o que significa que foi gerada pela chave privada corresponde à chave pública utilizada na verificação e que o documento está **íntegro**.

Caso sejam diferentes, a assinatura está incorreta, o que significa que pode ter havido **alterações** no documento ou na assinatura pública.

Assinatura Digital



Exercícios Práticos

Esta atividade deverá envolver toda a turma. Vocês devem criar um servidor web (com Flask ou Django, utilizando a linguagem Python) que realize a assinatura digital de arquivos PDF.

Basicamente o servidor irá ter uma página web de login através de seu siape (código único de 7 dígitos inteiros). O sistemas deverá permitir cadastro.

Após o login o usuário poderá submeter um arquivo PDF e assiná-lo digitalmente (utilize os códigos de exemplo no site).

Lembre-se, você deverá armazenar as chaves privadas no servidor, e quando o cliente estabelecer um acesso, ele terá uma chave pública para assim poder assinar documentos em sua máquina, de forma que o PDF e assinatura sejam submetidos ao servidor, e o servidor possa comparar os hash da assinatura e do arquivo recebido.