



Segurança

Segurança em Sistemas

Vulnerabilidades, Ameaças e Controle

Ataques Ativos

Confidencialidade

Integridade

Disponibilidade

Autenticação

Criptografia

Encriptação Simétrica

Requisitos

Vulnerabilidades

Algoritmos

Stream Ciphers

Autenticação

Message Authentication Code (MAC)

Hash

Public Keys

Diffie-Hellman

Segurança em Sistemas

Vulnerabilidades, Ameaças e Controle

- **Vulnerabilidade:**
 - Uma fraqueza do sistema relevante ao seu design ou implementação;
 - Pode ser explorada para causar danos;
 - Exemplo: falha a verificar a identidade do utilizador.
- **Ameaça:**
 - Um potencial de violação de segurança, que existe quando há uma entidade, circunstância, capacidade, ação ou evento que pode causar danos.

- **Controle:**
 - Mitigação das vulnerabilidades através da implementação de ações/processos que as reduzam.

Ataques Ativos

- Envolvem modificação dos dados transmitidos;
- Tipos:
 - Repetição: captura e envio de pacotes de forma não autorizada;
 - Dissimulação: uma entidade tenta fazer-se passar por outra;
 - Modificação: alteração de parte de uma mensagem legítima;
 - DoS: tentativa de inativar uma linha de comunicação.

Confidencialidade

- As medidas de confidencialidade são projetadas para evitar informações confidenciais de tentativas de acesso não autorizado.

Integridade

- Envolve manter a consistência, precisão e confiabilidade dos dados durante todo o seu ciclo de vida;
- Os dados não devem ser indevidamente alterados e devem ser tomadas medidas para garantir que os dados não possam ser alterados por pessoas não autorizadas (por exemplo, em violação de confidencialidade).

Disponibilidade

- A informação deve ser consistente e prontamente acessível para os acessos autorizados;
- Envolve a manutenção adequada de *hardware* e infraestrutura técnica e sistemas que armazenam e exibem as informações.

Autenticação

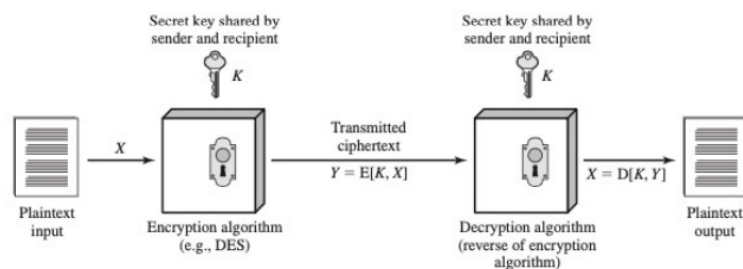
- Define o controle no acesso a certos recursos;
- **Identificação:** quem é o indivíduo que quer aceder ao sistema;

- Poderá ser facilmente conhecida ou previsível;
- **Autenticação:** confirmação da identidade do indivíduo que pretende aceder ao sistema;
 - Deve ser fiável;
 - A autenticação é composta por uma ou mais das seguintes qualidades:
 - **Algo que o utilizador sabe:** Password, pins, *handshakes*;
 - As passwords apresentam diversos problemas (utilização, divulgação, revogação, perda);
 - **Algo que o utilizador é:** dados biométricos, impressão digital;
 - Não é um processo binário.
 - **Algo que o utilizador tem:** *tokens*;

Criptografia

- Pode ser útil **esconder a informação para indivíduos não autenticados**;

Encriptação Simétrica



1. **Entrada:** conteúdo original;
2. **Algoritmo de encriptação:** algoritmo que realiza alterações e substituições ao conteúdo original;
3. **Private Key:** é também um *input* do algoritmo, sendo que as alterações feitas a (1) dependem desta chave;
4. **Conteúdo encriptado:** mensagem revolvida resultante do algoritmo e chave;
5. **Algoritmo de desencriptação:** execução reversa do algoritmo de encriptação, tendo como *input* (3) e (4).

Requisitos

- Mesmo que o código do algoritmo seja descoberto, um possível *hacker* não consiga decifrar a mensagem sem acesso à *private key*;
- Não deverá ser possível fazer *reverse-engineer* do algoritmo com base no conteúdo encriptado e conteúdo desencriptado;
- O destinatário e remetente de uma mensagem encriptada deverão ter obtido cópias da chave secreta de uma forma segura.

Vulnerabilidades

- **Criptanálise:** tentativas de deduzir a chave ou outros pontos do algoritmo;
- **Brute-Force:** tentar todas as combinações de chaves secretas até que o output do algoritmo de descriptação seja "legível".

Algoritmos

- Os algoritmos mais comuns são **block ciphers** (e.g. DES, 3DES e AES):
 - O *input* é processado em blocos com tamanho fixo;
 - Cada bloco encriptado tem o mesmo tamanho que o *input*;
 - Exemplos:
 - Data Encryption Standard (DES):
 - Utiliza operações lógicas e aritméticas em dados binários até 64 *bits*;
 - Chaves de 56 bits;
 - A encriptação ocorre em 16 passos: a cada passo, substitui e baralha o *input* com base nos valores da chave;
 - Triples DES (3DES):
 - Repete o DES 3 vezes (com 2/3 chaves secretas distintas):
 - Constitui uma desvantagem porque o DES foi desenvolvido nos anos 70, e a sua implementação em sistemas modernos não é a mais eficiente;
 - Ambos utilizam blocos de 64 bits, logo há deficiência na eficiência de (des)encriptação;

- Mitiga ataques *bruteforce*;
- Estende a capacidade de resistência a criptoanálise do DES;
- Advanced Encryption Standard (AES):
 - Utiliza blocos de 128 bits e chaves de {128, 192, 256} bits;

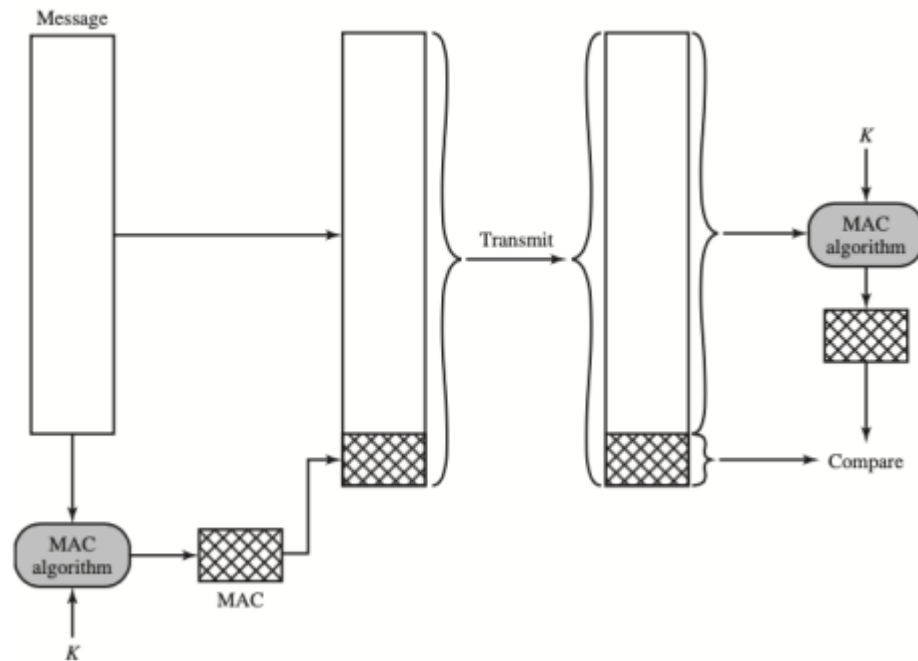
Stream Ciphers

- Encriptam o conteúdo **um *byte*** de cada vez;
- Geralmente mais rápidas que *block ciphers*;

Autenticação

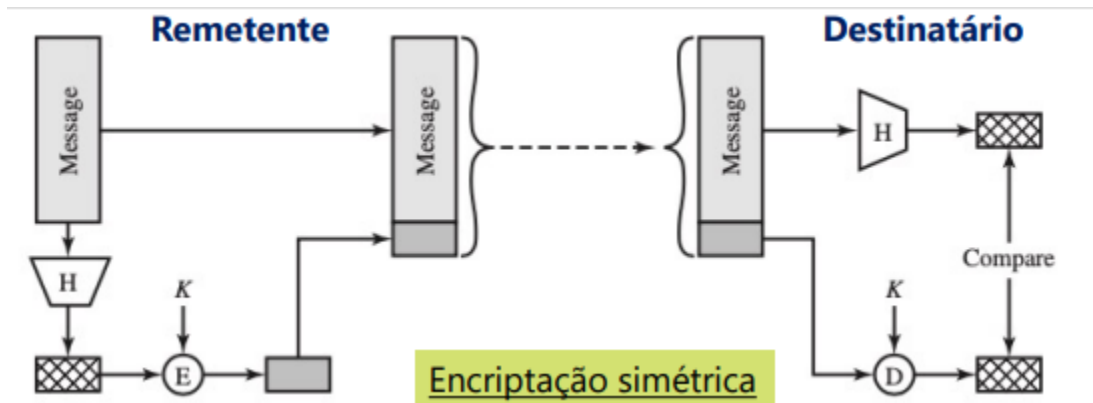
Message Authentication Code (MAC)

- Uma *Private Key* é utilizada para criar um bloco — MAC;
- A mensagem e o código são enviados ao destinatário;
- Assumindo que apenas as partes que comunicam conhecem a *Private Key*:
 - O destinatário garante que a mensagem não foi alterada:
 - se for alterada, o MAC não corresponde-
 - O destinatário garante que a mensagem foi enviada pelo remetente.
 - Se a mensagem contém uma sequência numérica, o MAC garante também que esta sequência não é alterada.

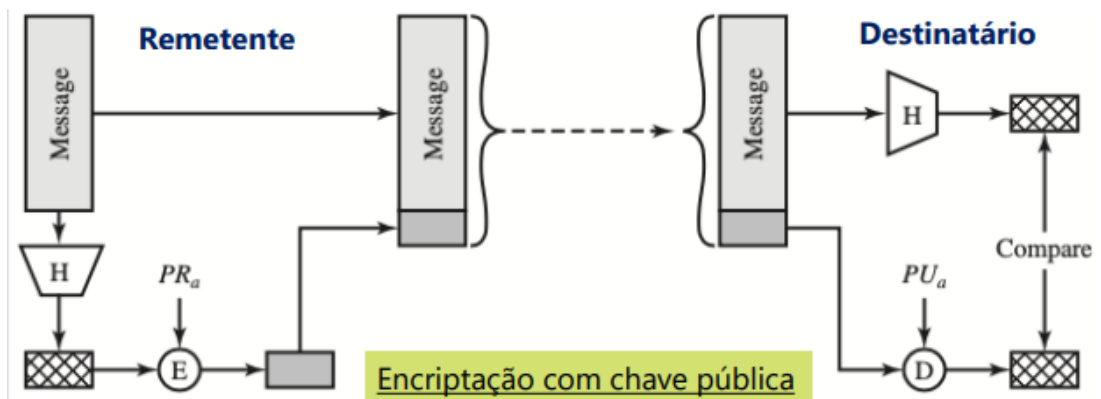


Hash

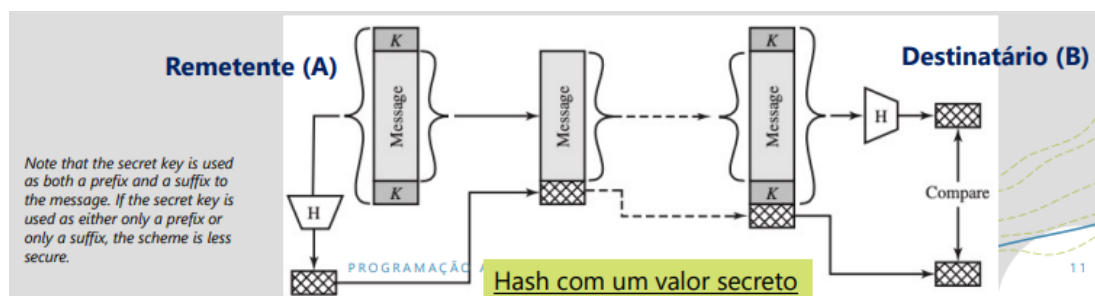
- Uma função H que recebe uma mensagem M como input e produz um output $H(M)$;
- Existem 3 abordagens na implementação deste tipo de autenticação:
 - Encriptação simétrica:



- Encriptação com chave pública:



- Hash com valor secreto:



Requisitos:

- Ser aplicável a qualquer tamanho de dados;
- Produzir um *output* de tamanho fixo;
- A implementação de $H(x)$ deverá ser prática (e.g. a computação $H(x)$ deverá ser fácil);
- **One-way, per-image resistant:** para um código h , é inviável encontrar um tal x tal que $H(x) = h$;
- **Weak-collision resistant:** para um bloco x , é inviável encontrar um $y \neq x$ tal que $H(y) = H(x)$;
- **Strong-collision resistant:** é inviável encontrar um par (x, y) tal que $H(x) = H(y)$.

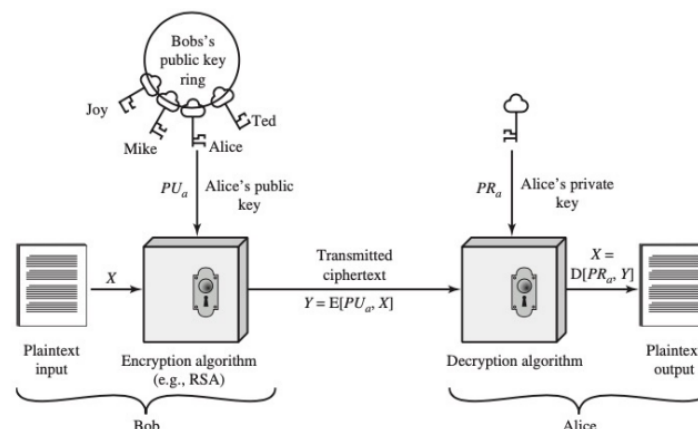
Public Keys

Diffie-Hellman

- Algoritmos baseados em operações matemáticas;
- Algoritmo assimétrico: **prevê a utilização de 2 chaves;**

A implementação com *public keys* ocorre com:

1. *Input* em *plaintext*;
2. Algoritmo de encriptação assimétrico que realiza transformações no *input*;
3. *Public and private keys*: Par de chaves que serão utilizadas para encriptação e desencriptação (que afetam as transformações ao *input* inicial);
4. Texto encriptado;
5. Algoritmo de desencriptação.



- **Processo:**

1. É gerado um par de chaves para (des)encriptação;
2. A chave pública é mantida num registo público; a chave privada é mantida privada;
3. Se A quiser enviar mensagem a B, utiliza a chave pública de B para encriptar a mensagem;
4. Quando B recebe a mensagem, utiliza a sua chave privada para a desencriptar.



A comunicação é segura desde que as chaves privadas sejam protegidas!

- **Requisitos:**

- É computacionalmente inviável para um atacante com uma chave pública, PU_b e um texto encriptado, C , recuperar a mensagem original M ;

- Qualquer uma das chaves poderá ser usada para encriptação desde que a seguinte seja usada para descriptação:
- $M = D[PU_b, E(PR_b, M)] = D[PR_b, E(PI_b, M)]$