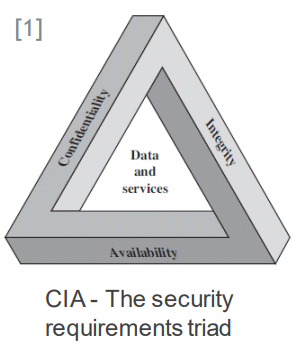
**Segurança do Computador**

**3 Objetivos chave**

**Confidencialidade**

→ Não é disponibilizado para as pessoas não autorizadas (privado).

**Integridade**

→ Dados e programas não podem ser alterados por pessoas in autorizadas.

→ Sistema não é manipulado por pessoas in autorizados.

**Disponibilidade**

→ Sistema funcionar corretamente.

→ Serviços não negados a pessoas autorizadas

**Perda de segurança nos objetivos chave**

**Confidencialidade**

→ A perda de confidencialidade é uma distribuição ou divulgação de informação não autorizada

**Integridade**

→ A perda de integridade é a modificação ou destruição in autorizada da informação.

**Disponibilidade**

→ A perda de disponibilidade é a perturbação ou dificuldade em aceder ou usar informação ou sistemas de informação.

**Autenticidade**

→ Ser genuíno e ser capaz de ser autorizado (confiado) cada input que o sistema recebeu veio de alguém ou lugar de confiança.

**Não repúdio**

→ “Eu fiz algo, e não consigo dizer que não fui eu…”.

**Níveis de impacto**

**Low / Baixos**

**→** Efeito adverso limitado na organização ou indivíduos

**Moderate / Médio**

**→** Efeito adverso sério

**High / Alto**

**→** Severo ou catastrófico efeito adverso

**Algumas definições**

**Ataques de Segurança**

**→** Ação que compromete a segurança da informação.

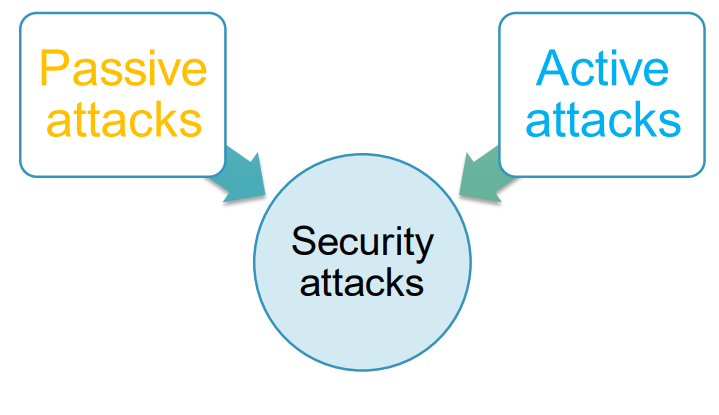
**Mecanismo de Segurança**

**→** Um processo desenhado para detetar, prever e recuperar de um ataque de segurança.

**Serviço de Segurança**

**→** Luta os ataques de segurança, usando um ou mais mecanismos de segurança.

**Ataques de Segurança**

**** **Passivos**

**Objetivo**

**→** Conseguir a informação.

**2 tipos**

**→** Leitura in autorizada dos conteúdos das mensagens ou análise de tráfego.

**Difíceis de detetar!**

**→** Não envolvem nenhuma alteração dos dados.

**Como prevenir**

**→** Criptografia.

**Ativos**

**Replay**

**→** Envolve a captura passiva de uma unidade de dados e a sua subsequente retransmissão.

**Modificação da mensagem**

**→** Uma porção da mensagem é alterada ou as mensagens são atrasadas ou reordenadas.

**Denial of Service (DoS)**

**→** Trava ou não permite o uso normal da comunicação.

**Representação / Personificação**

→ Quando uma entidade se faz passar por outra.

**Cifras Simétricas – Técnicas Clássicas**

**Pontos Chave e Definições**

**Plain Text**

→ Mensagem original, clean

**Texto Cifrado**

→ Mensagem cifrada não pode ser lida, resulta de um algoritmo de cifragem

**Secret Key**

→ Input para o algoritmo de encriptação ou desencriptação.

**Algoritmo de encriptação**

→ Cria a mensagem cifrada do plain text

**Algoritmo de desencriptação**

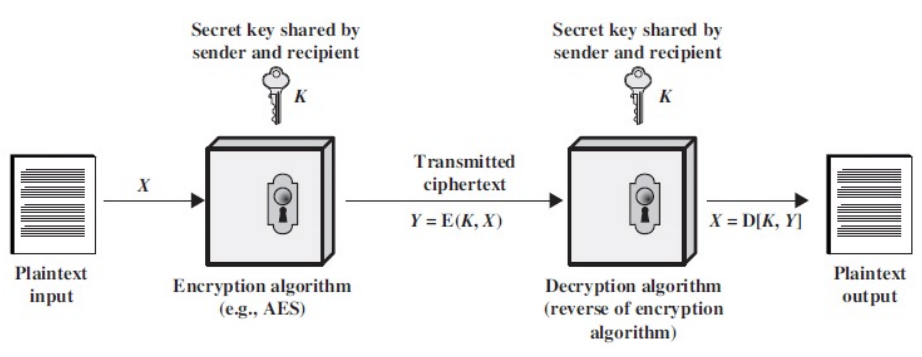
→ Recupera o plain text da mensagem

**Criptografia** → Esquemas usados para encriptar

**Criptoanalise** → técnicas usadas para decifrar a mensagem

Criptografia + Criptoanalise = **Criptologia**

**Criptografia Simétrica**

****

→ Usa a mesma chave para encriptar e desencriptar

→ O oponente não deve ser capaz de desencriptar a mensagem ou descobrir a chave

→ Transforma texto normal em texto encriptado, necessita:

→ Chave secreta

→ Algoritmo de encriptação

→ Não existe necessidade de manter o algoritmo secreto, apenas a chave

→ Ao usar a mesma chave e algoritmo o plain text é recuperado.

**Tipos de Ataque**

**Criptoanalise**

→ Podem ter o texto cifrado e querem descobrir o plain text

→ Podem ter o texto cifrado e querem descobrir a chave.

→ Tenta descobrir padrões.

**Força Bruta**

→ Tentar desencriptar a mensagem de todas as maneiras possíveis.

**Técnicas**

**Substituição**

→ Substituir as palavras ou letras por outras.

**Transposição**

→ Trocar a ordem das letras.

**Máquinas de Notor**

→ Múltiplos estados de encriptação

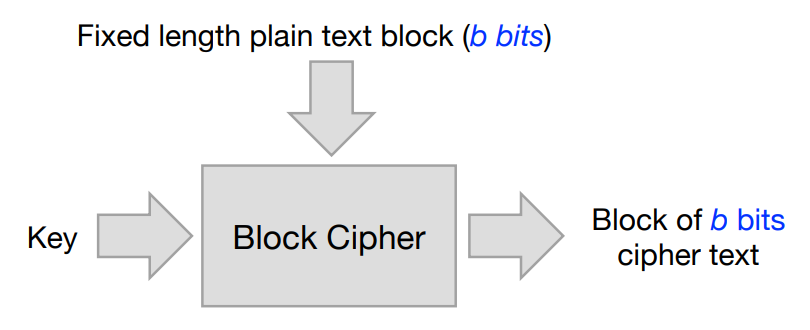
→ Melhor método, antes dos mais modernos

→ Baseado em substituição.

**Cifras Simétricas – (cifra de bloco, modos de operação, padding e tamanho chaves)**

**Cifras em Blocos**

→ Os dados são divididos em blocos de tamanho fixo e encriptados bloco a bloco.



**Feistel structure**

→ O número de vezes que cifra tem de ser exatamente o mesmo número de decifrar

→ Em cada ronda:

→ A substituição é feita em uma metade dos dados processados

→ De seguida é feita a permutação que troca as duas metades.

**Cifra de Sequência**

→ A encriptação é feita bit a bit.

→ Usa uma stream infinita de pseudorandom bits como chave

**Modos de Operação**

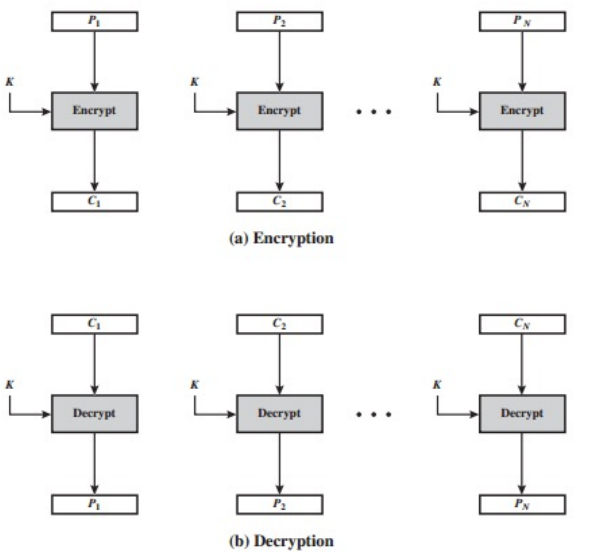
**1. Eletronic Code Book (ECB)**

→ Divide em blocos que usam a mesma chave para decifrar.

→ Boa performance

→ Bom para pequenas quantidades de dados.

→ Atenção: o mesmo plain text, gera o mesmo texto cifrado

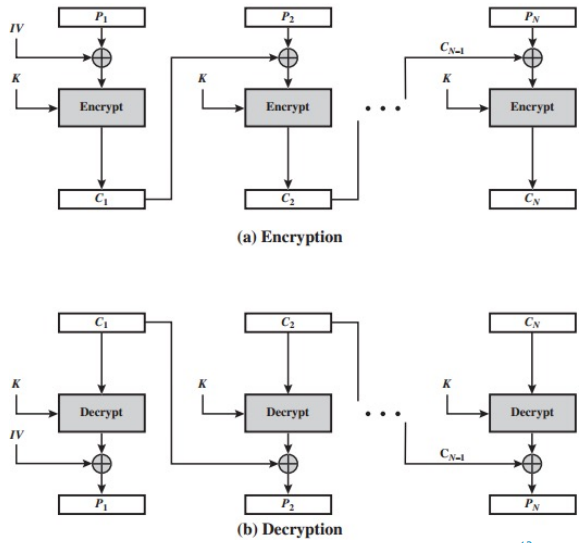


**2. Cipher block chaining (CBC) – Default Operation (Mais Usado)**

→ Resolve o problema do ECB (cifrar o mesmo texto), texto cifrado é sempre diferente, pois o bloco cifrado anteriormente influencia o que vem a seguir.

→ Necessidade do uso do IV

→ IV tem o mesmo tamanho do bloco de texto cifrado e tem de ser secreto



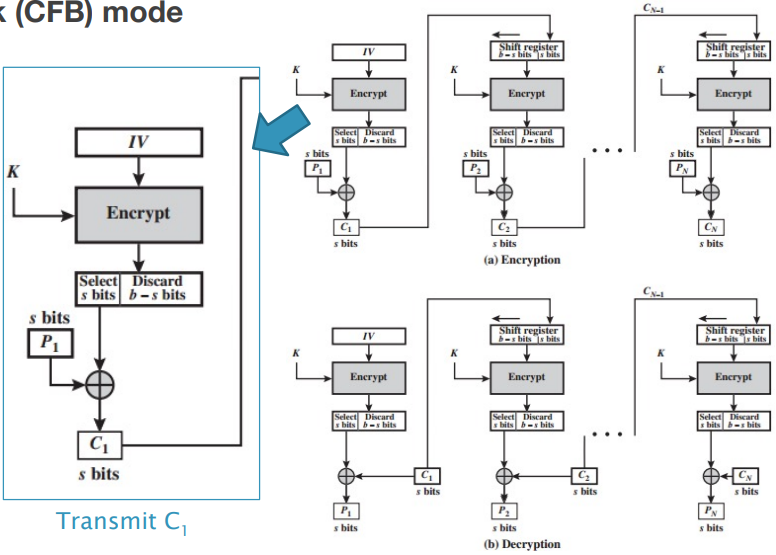
**3. Cipher feedback (CFB)**

→ Texto cifrado depende do texto cifrado atual e anterior.

→ Segundo bloco apenas começa quando o texto anterior for totalmente cifrado e o seu bloco tiver enchido

→ Pudemos ter um output enquanto está a cifrar.

→ Qualquer erro durante a cifragem tem um grande impacto nos blocos posteriores

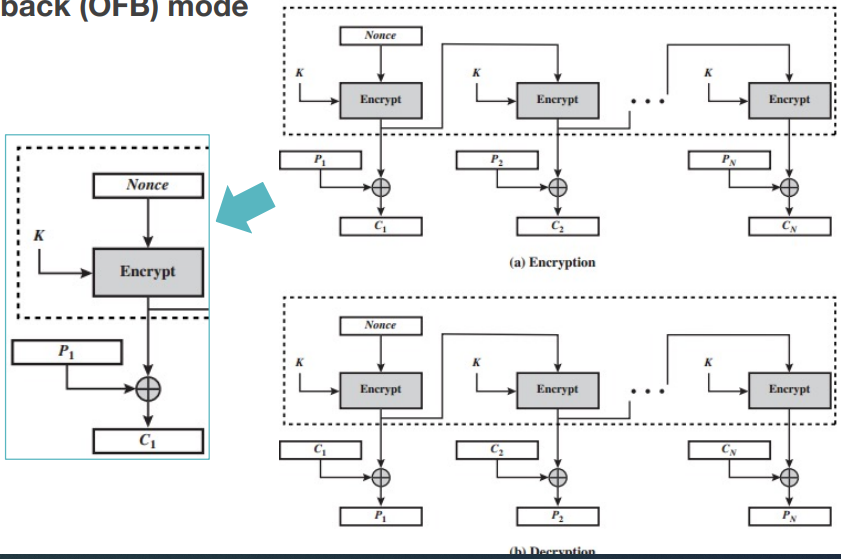


**4. Output feedback (OFB)**

→ Algoritmo que encripta também desencripta

→ Valor aleatório que é gerado para usar na primeira operação de cifragem, não influencia as posteriores (apenas afeta o bloco atual).

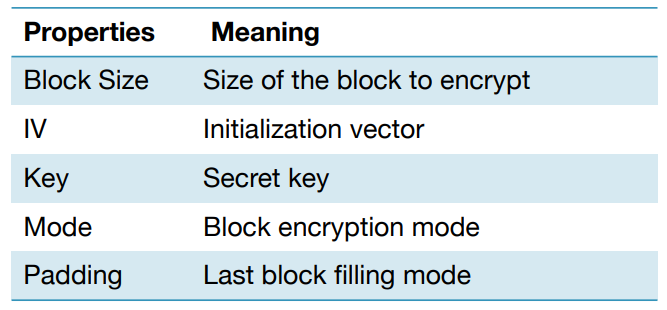
→ Funciona em realtime, o bit é encriptado e enviado



**5. Counter (CTR)**

→ Usa um contador para cifrar, inicializado a um valor, o primeiro bloco usa o contador + chave para cifrar o texto

**Padding** → Usado para encher o último bloco para terem todos o mesmo tamanho, ou seja, quando não há espaço suficiente para encher o último bloco

Informação tem de ser conhecida pelas duas pessoas que partilham.

A chave secreta e o IV têm de ser gerados pelo algoritmo de encriptação

Atualmente as chaves, para haver mais segurança, têm um tamanho de 56 a 128 bits

**Segurança para sistemas criptográficos**

→ Qualidade do algoritmo

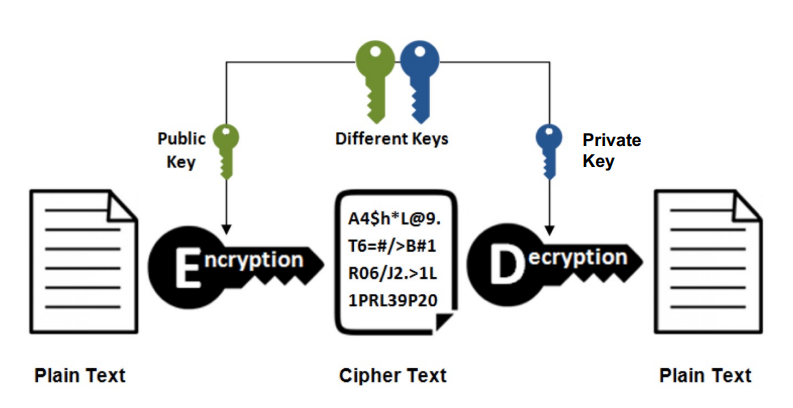
→ Tamanho da chave

Texto cifrado em bytes → têm de ser senore base64 para não perder informação

Write → encripta

Read → desencripta

**Cifras Assimétricas**



**Encriptação utilizando uma chave public é a maior revolução**

→ Não usa substituição ou permutação

→ Funções matemáticas que usam duas chaves separadas

→ Confidencialidade, distribuição de chaves e autenticação

Equívocos em relação a cifras utilizando chaves publicas

→ Algoritmos assimétricos não são mais ou menos seguros que algoritmos simétricos

→ Algoritmos simétricos não estão absoletos

→ Pudemos divulgar a chave públicas, mas apenas essa!

**Terminologia**

**Chave Assimétricas**

→ Duas chaves relacionadas: uma publica e uma privada

→ São utilizadas para realizar operações complementares: encriptar, desencriptar, assinaturas

**Certificado de chave públicas**

→ Um documento digital

→ Liga um nome a uma chave publica

→ Assinado por uma Autoridade de Certificação (chave privada)

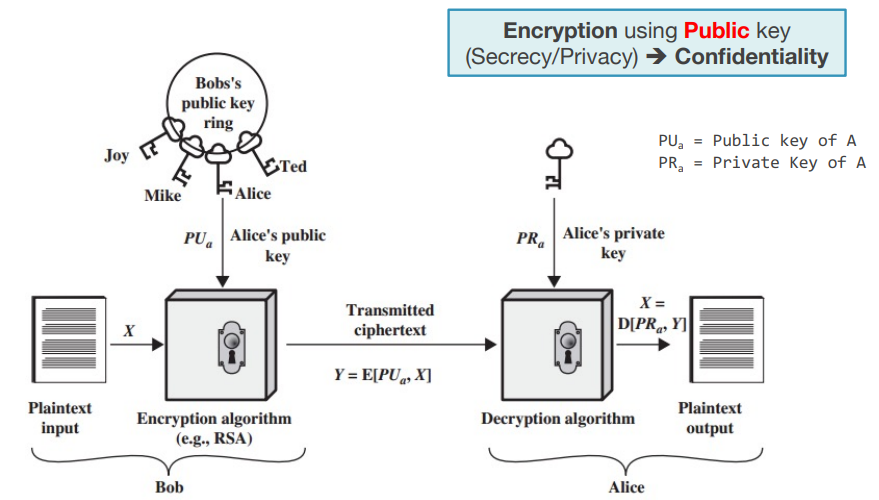
**Princípios de sistemas de criptografia de chaves publicas**

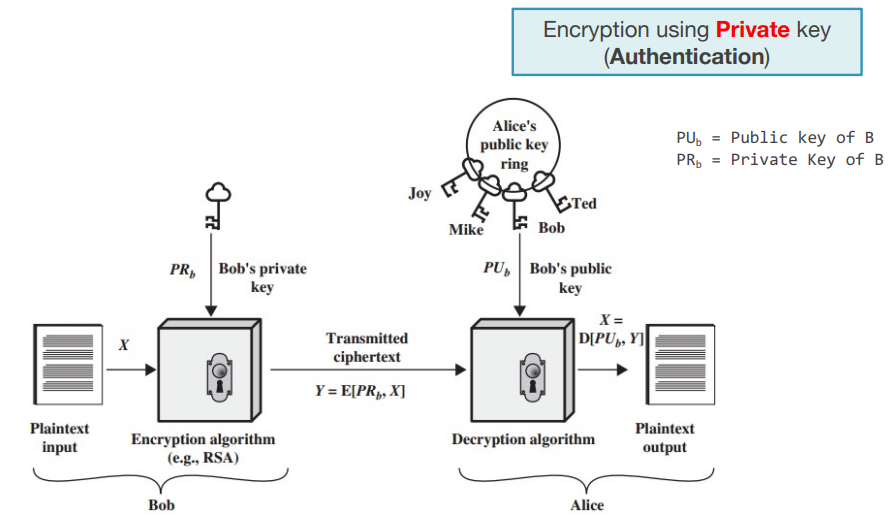
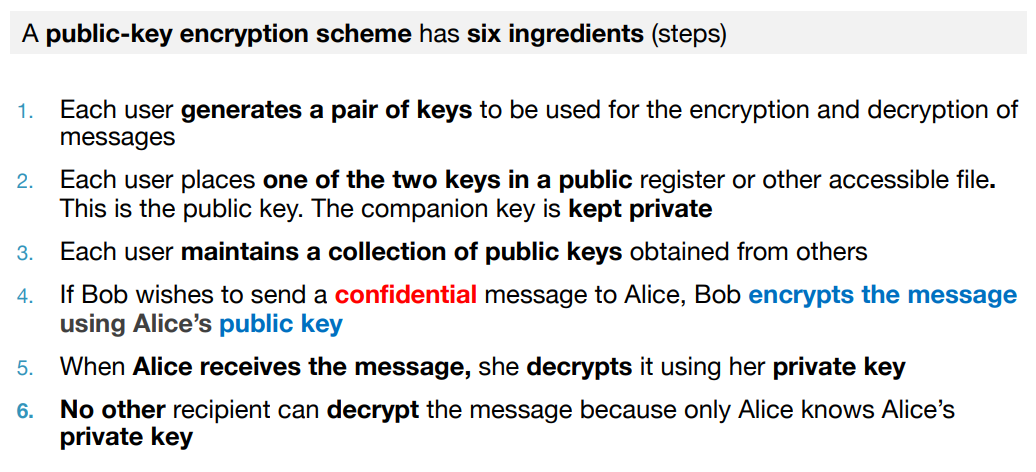
→ Algoritmos assimétricos usa uma chave para encriptação e outra (relacionada) para desencriptar

**Características principais**

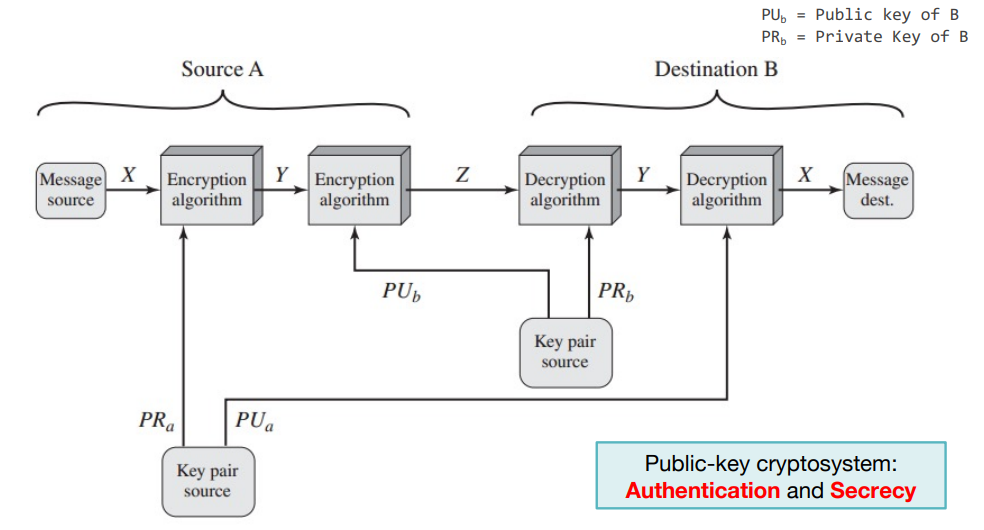
→ Conseguir a chave private através da publica é computacionalmente complicado

→ Qualquer chave pode ser utilizada para encriptar a outra é utilizada para desencriptar

****

****

No entanto, é possível fornecer tanto a Função de autentificação e confidencialidade por um duplo uso do esquema de chave pública

****

**Aplicações para sistemas de criptação de chave publica**

**Encriptação/Desencriptação** → Quem envia encripta a mensagem com a chave publica do recetor

**Assinatura Digital** → Quem envia assina a mensagem com a sua chave privada

**Troca de Chaves** → Cooperação para a sessão das chaves

**Cripto análise de chaves públicas**

**1. Força Bruta**

→ Usar chaves grandes

→ Assimétrico usa one way functions

→ Uma one way function é aquela que mapeia um domínio em um intervalo tal que cada valor da função tem um inverso único, deve de ser fácil de criar mas dificil de reverter

→ A chave deve de ser grande o suficiente, mas pratica para encriptar e desencriptar

**2. Conseguir a chave privada através da pública**

→ Ainda não foi provado que se consegue ter a chave privada através da pública, mesmo que ambas estejam relacionadas.

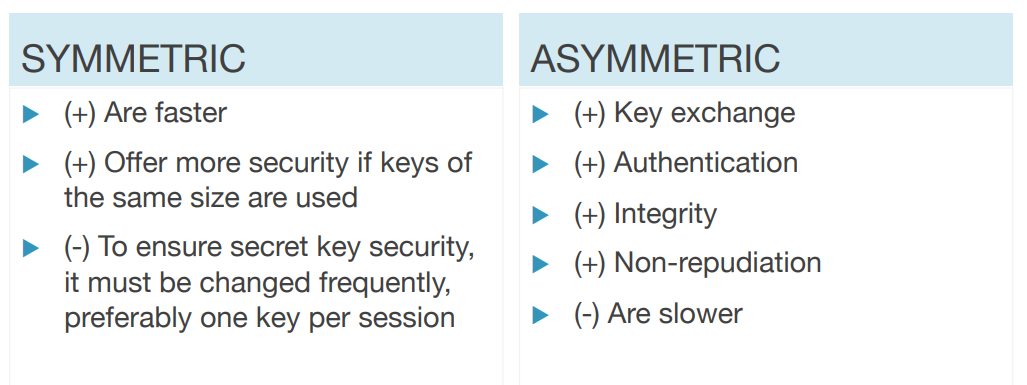
**3. Ataque mensagem provável**

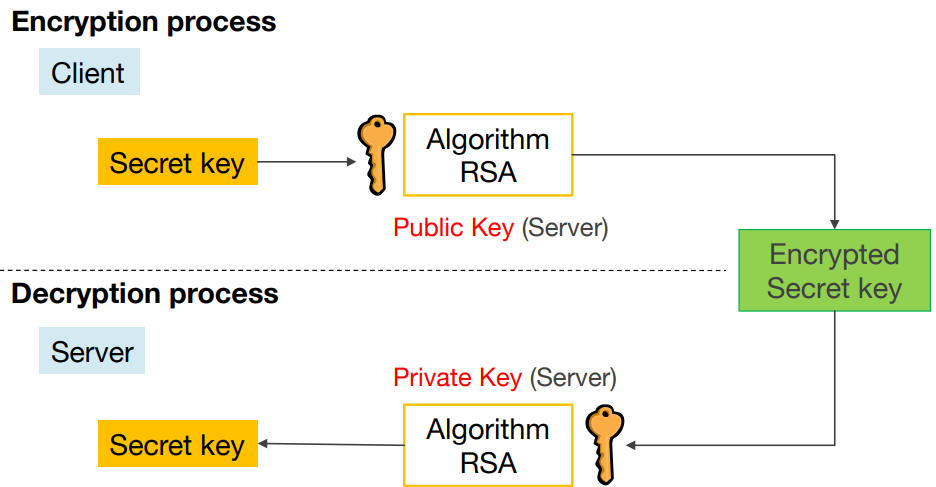
→ Tentar encontrar os bits que realizara, a cifragem e fazer o matching para saber se é o mesmo

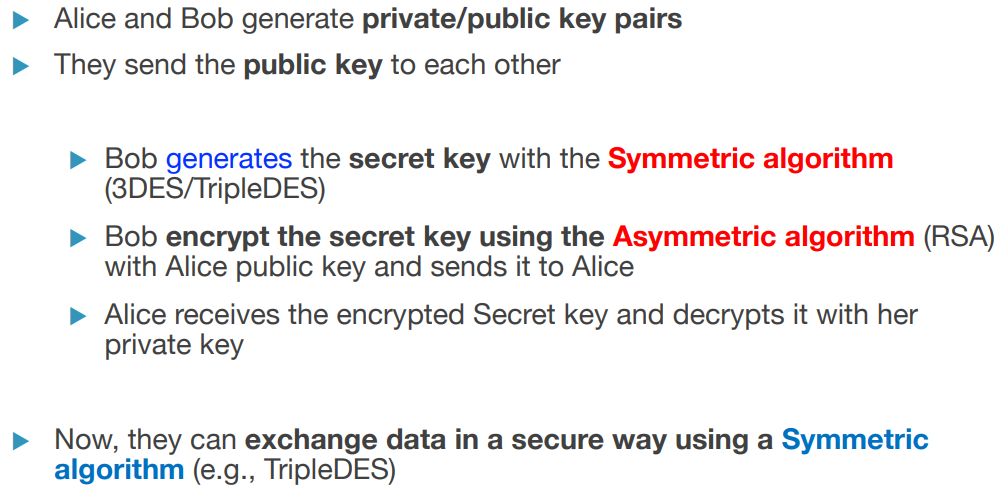
**Problemas com a distribuição de chaves públicas**

→ Man in the middle, alguém pode ficar com a chave pública

**Simétrico vs assimétrico**

****

****

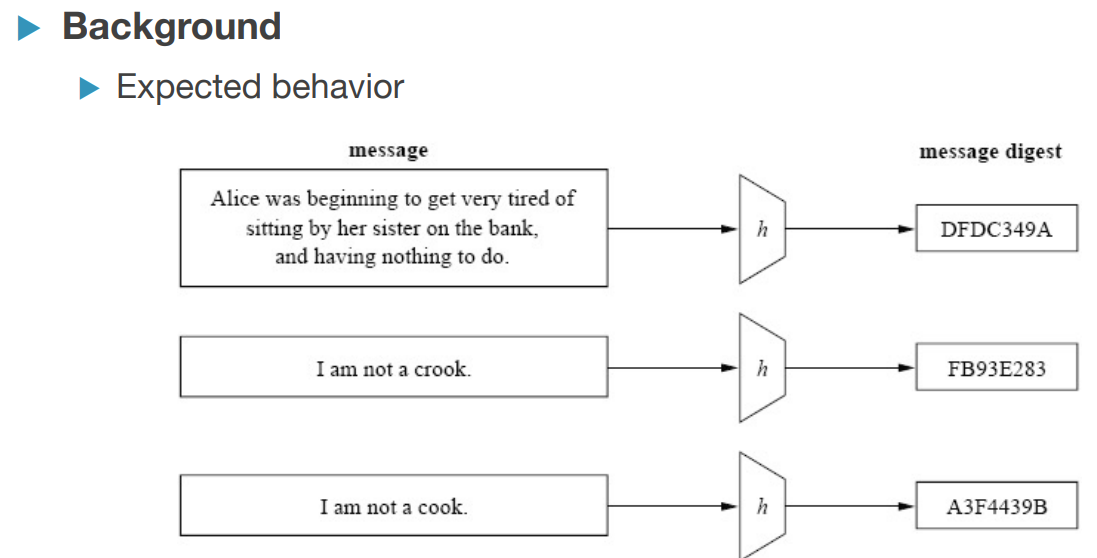
****

**Funções Hash**

Uma boa função hash produz: o resultado da aplicação da função a um grande conjunto de entradas produz saídas uniformemente distribuídas e aparentemente aleatórias.

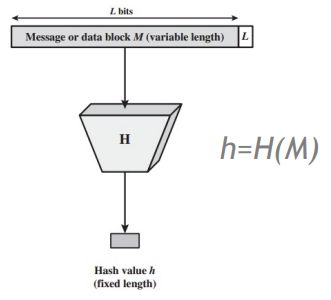
Quando aplicamos uma função hash queremos manter a integridade dos dados, pois qualquer mudança a um bit ou bits, resulta num hash code totalmente diferente

Tipos de algoritmos de hash 🡪 SHA ou Whirlpool



Propriedades Chave:

1. Computacionalmente difícil encontrar o M que produz o h (**one-way property**)



1. Computacionalmente inviável encontrar 2’s M que produzem o mesmo h

Devido a estas características as funções de hash são normalmente utilizadas para saber se os dados foram alterados

* One-way property
* Collisiono-free property

**Autenticação de mensagem**

É um mecanismo ou serviço para verificar a integridade da mensagem

Assegura que os dados recebidos são exatamente os mesmo que os enviados

Quando uma hash function é utlizada para fornecer autenticação de mensagem, a função de hash é normalmente referida no resumo de mensagem

**Aplicações de Criptografia Hash Functions**

**Aplicações**

Autenticação, integridade e confidencialidade da mensagem

* A mensagem mais a concatenação do hash code são encriptadas usando um algoritmo simétrico
* Porque só o A e B partilham a chave secreta, a mensagem só pode ter vinda de A e não pode ter sido alterada
* O hash code tem a estrutura e redundância necessária para ter autenticação
* Por que encriptação é aplicada à mensagem inteira e também ao hash code também garantimos confidencialidade

Uma imagem com texto, dispositivo

Descrição gerada automaticamente

Autenticação, integridade, mas não confidencialidade

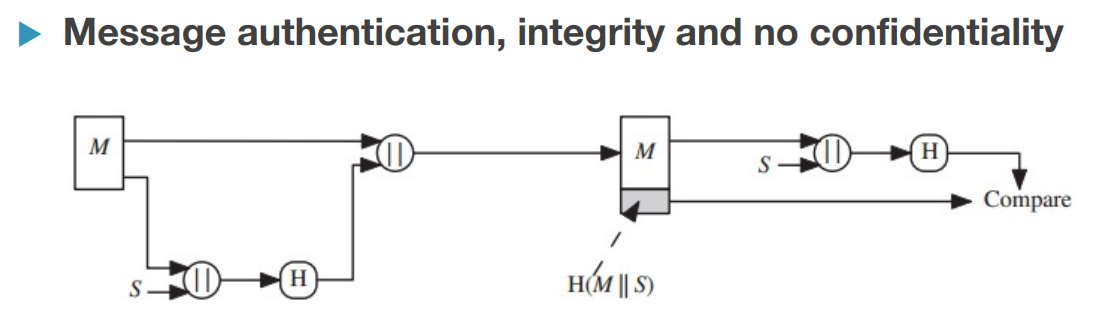
* Apenas o código hash é encriptado usando um algoritmo simétrico
* Reduz a carga de processamento para as aplicações que não necessitam de confidencialidade

Uma imagem com texto, relógio, dispositivo

Descrição gerada automaticamente

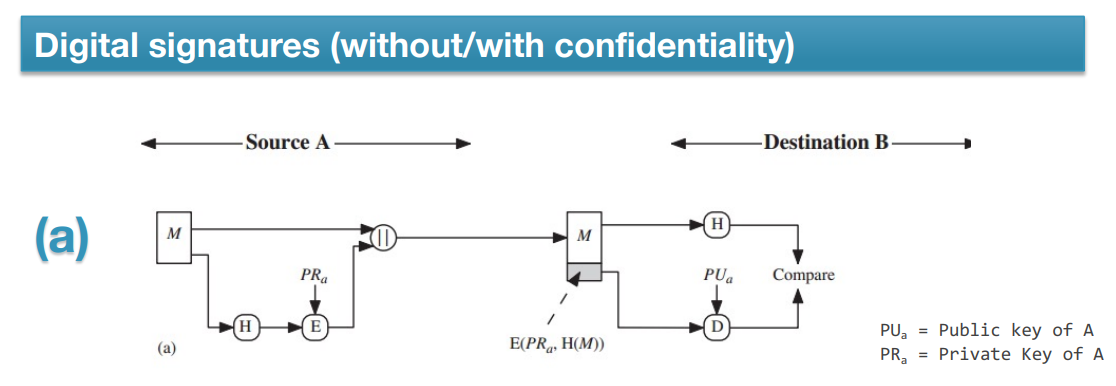
Autenticação de mensagens, integridade e sem confidencialidade

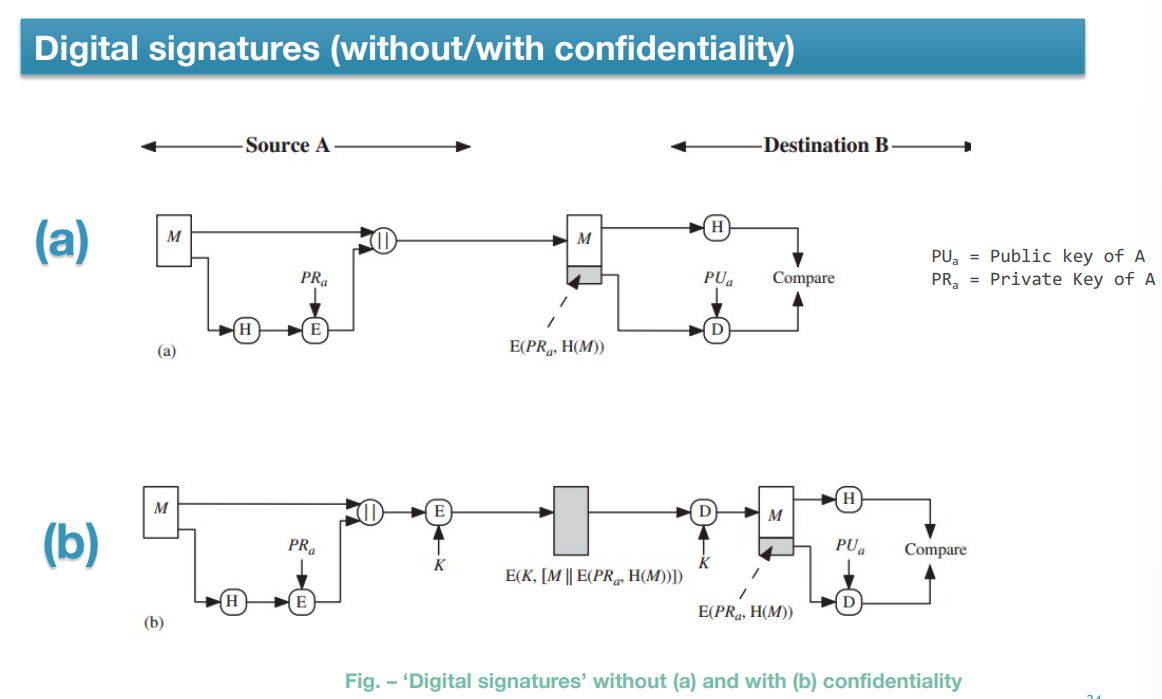
* É possível utilizar a função de hash mas não encriptação para a mensagem autenticada
* A técnica assume que ambas as entidades partilham a mesma chave secreta comum
* Como B contém a chave, pode recalcular o hash value para comparar e verificar
* Porque o valor secreto não é enviado, qualquer oponente não conseguirá alterar ou intercetar a mensagem e não pode gerar uma mensagem falsa



Assinaturas digitais

* O valor de hash é encriptada usando a chave privada do usuário
* Qualquer pessoa que também contenha a chave pública pode verificar a assinatura





* (a) e (b) fornece:
  + **Autenticação –** hash code é encriptado usando a chave privada do que envia
  + **Assinatura Digital –** Apenas o que envia conseguiria produzir o hash code encriptado
* Se confidencialidade e assinatura digital são requeridas
  + Mensagem + hash code encriptado com chave privada, podem ser encriptados utilizando uma chave secreta simétrica

**Requisitos e segurança**

1. Resistência à pré-imagem, reverter o hash para ter a mensagem inicial
2. Impossível encontrar a segunda imagem utilizando o mesmo hash value
3. Impossível encontrar 2 textos que tenham o mesmo hash value

**Existem 2 tipos de ataques às hash functions**

* **Força Bruta**
  + Apenas depende do tamanho do bit no hash value
* **Criptoanalise**
  + Baseado na fraqueza de um algoritmo de criptografia

**Estrutura Geral**

Uma imagem com texto

Descrição gerada automaticamente

**SHA1 –** já foi crackeado

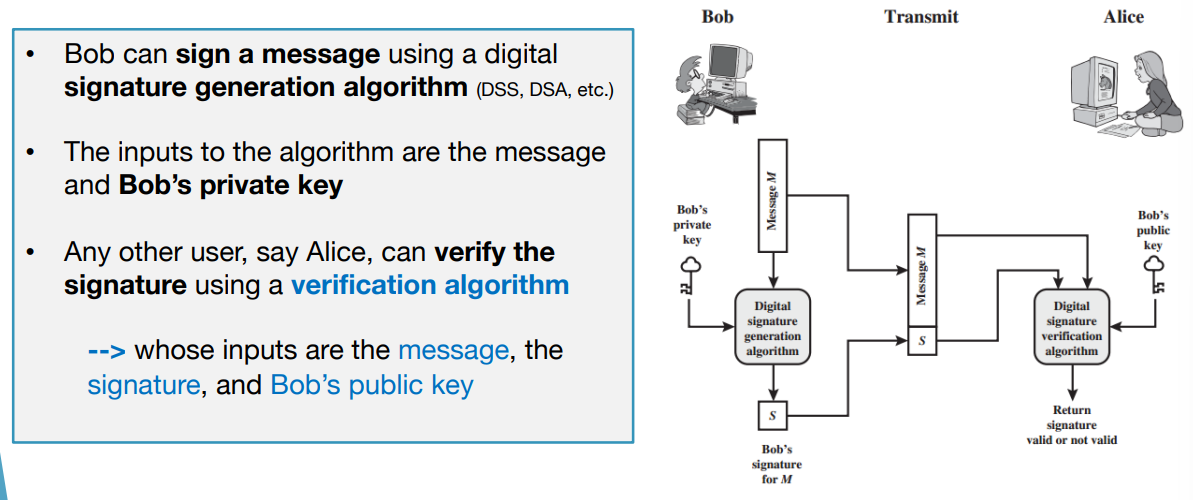
**Autenticação, Assinaturas Digitais e Certificados**

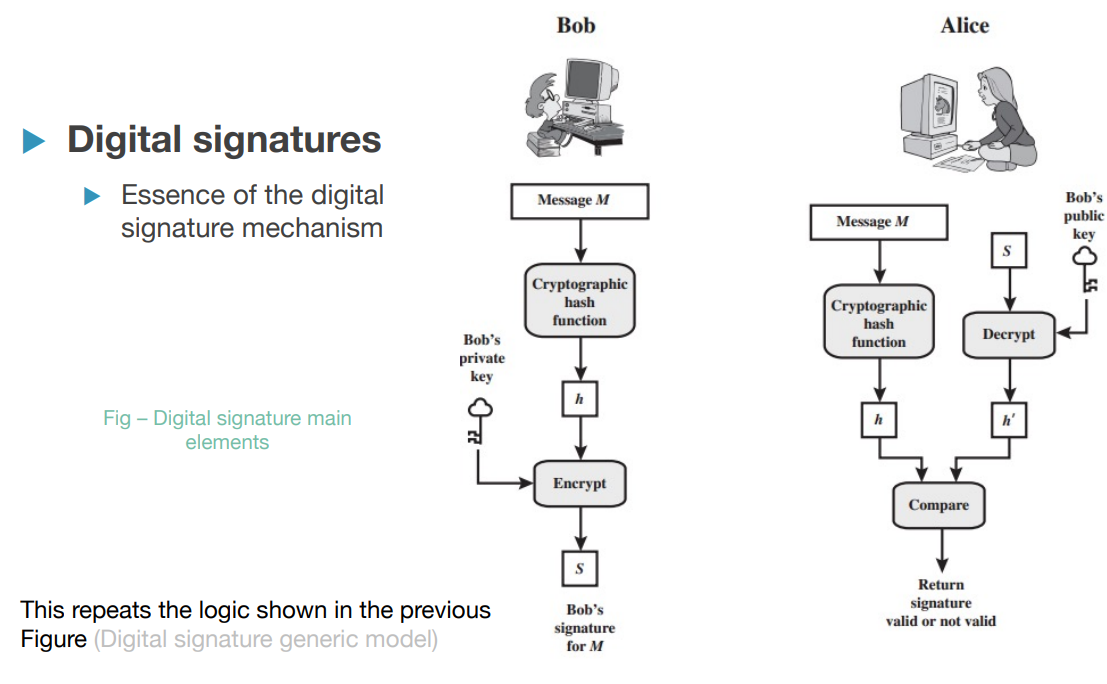
Quando **não existe confiança completa** entre a pessoa que **envia** e a que **recebe** é necessário algo mais que a **autenticação** 🡪 A melhor solução encontrada é **assinatura digitais**.

**Assinaturas Digitais** 🡪 É o mecanismo de **autenticação** que **habilita o criador da mensagem** de adicionar código que **funciona como assinatura** 🡪 a assinatura garante a **integridade** e **fonte** da mensagem

**Autenticação de Mensagem**:

* Garante que as mensagens não são lidas por um terceiro
* Mas não garante a proteção entre as duas pessoas~





**Assinatura Digital Propriedades:**

* Deve verificar o autor, data e horas da assinatura
* Tem de autenticar os conteúdos na altura da assinatura
* Deve de ser variável por third party, que resolver disputas

**Assinatura Digital Requisitos:**

* A assinatura deve de ser um padrão de bit’s que depende da mensagem que esteja a ser assinada
* A assinatura deve de usar alguma informação que seja única ao que envia para prevenir falsificação e negação (não repudio)
* Deve de ser relativamente fácil produzir uma assinatura digital
* Deve de ser relativamente fácil reconhecer e verificar assinatura digital
* Deve de ser computacionalmente complicado forjar uma assinatura digital

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

