Segurança em Sistemas Distribuídos

Ricardo Costa – rcosta@estg.ipp.pt

Segurança em Sistemas Distribuídos

Introdução

- **Diferenças essenciais entre os sistemas centralizados e os sistemas distribuídos**
- Canais seguros e seus requisitos

Criptografia simétrica

- Utilização da criptografia simétrica na comunicação e autenticação

Criptografia assimétrica

- Utilização da criptografia assimétrica na comunicação e autenticação

Sistemas Confiáveis (Dependable Systems)

- Sistemas confiáveis é a área da informática que estuda os problemas e as soluções inerentes à realização de sistemas seguros e fiáveis – "de confiança".

- Um sistema confiável, "do qual se pode depender", deverá exibir diversas propriedades:
  - disponibilidade, fiabilidade perante falhas -> redundância, replicação
  - seguro perante ataques

- Como?
  - Políticas (regras) de segurança, suportadas por mecanismos de segurança, implementadas sobre uma base de confiança estabelecida à partida

Da Necessidade da Segurança

- O **valor** inerente aos **dados** e às **operações** de um sistema fazem deles alvos de ataques;
- A proteção dos dados e das operações requer medidas específicas:
  - **Mecanismos de segurança** fornecem proteção;
  - **Políticas de segurança** definem como os mecanismos são usados afim de impor restrições aos acessos.

Segurança em Sistemas Distribuídos

- Sistemas distribuídos são suscetíveis a novos tipos ataques ...
- A separação física dos componentes (e a necessidade de comunicarem entre si, através de uma rede) introduz **vetores de ataque** adicionais.
- É necessário uma reavaliação da **base de confiança** -> **trusted computing base**.
- Para implementar políticas de segurança equivalentes, são necessários mecanismos de segurança mais sofisticados face a um sistema isolado.

Modelos de Segurança

- Num sistema existem entidades que do ponto de vista da segurança têm identidade própria, direitos e deveres - essas entidades podem ser utilizadores, componentes, processos, etc. e designam-se pelo termo **principal**.

- A segurança do sistema distribuído passa por:
  - autenticar os principais (**autenticação**);
  - verificar os seus direitos de acesso aos objetos (**controlo de acessos**);
  - da distribuição advém a necessidade de utilizar **canais seguros** para impedir o acesso, alteração ou destruição indevida de informação, incluindo, **proteger a privacidade**.

- Para fornecer segurança é necessário estabelecer que alguns componentes do sistema são seguros (**trusted computing base**) – caso contrário é impossível:
  - Por exemplo, num sistema isolado é normal assumir que o kernel do Sistema de Operação é seguro.

Elementos do Modelo de Segurança

- **Principal** - uma entidade (pessoa, processo, servidor, cliente, …) que é singular do ponto de vista dos direitos no sistema.

- **Autenticação** – processo de verificar que um principal P tem a identidade que diz ter – em geral, deve ser capaz de o provar.
  - Geralmente utiliza-se um método lógico do tipo segredo partilhado entre P e quem o autentica (de que uma palavra chave é o exemplo mais conhecido), mas também se pode basear na verificação de atributos físicos (identificação da voz, impressões digitais ou da retina por exemplo) ou na posse de algo que só P pode possuir (um cartão magnético por exemplo).

- **Controlo de acessos** - dada uma operação Op sobre um objecto O, é necessário decidir se o principal P pode aplicar Op a O.
  - Normalmente utilizam-se ACLs (Access Control Lists) ou Capacidades (Tickets).

Camadas de Segurança

- A segurança de um sistema pode ser endereçada por camadas
- O papel de cada camada é estabelecer uma linha de defesa e alargar a **trusted computing base** para as camadas superiores
- A **base de confiança (trusted computing base)** inicial deve ser tão minimalista quanto o possível
  - uma base de confiança reduzida à partida tem uma superfície de ataque mais reduzida

Vetores de ataque comuns

- **Adulterar** a **base de confiança computacional (trusted computing base - TCB)**
- **Explorar insuficiências** na trusted computing base
  - Falhas na especificação ou implementação da TCB podem dar ao atacante direitos ou identidades indevidas
- Violar os mecanismos de autenticação
- Obter segredos indevidos

Vetores de ataque comuns

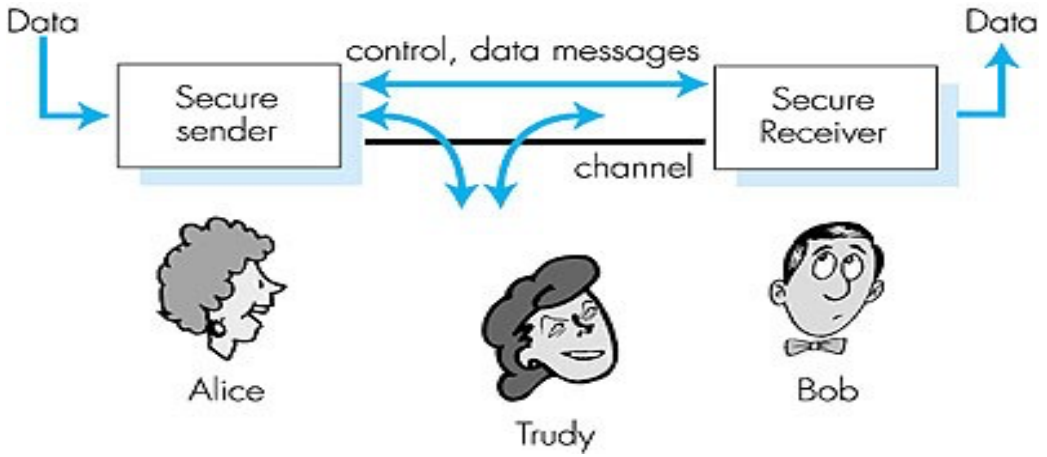
- Os ataques são muito diversos, podem ser muito engenhosos...
- Exploram bugs, insuficiências das implementações ou de planeamento ou simples fraquezas humanas...
- Na prática**, é impossível de enunciar todas as formas de ataque** e quase sempre estão-se a descobrir novas
- A tarefa de manter os sistemas seguros é um esforço contínuo que envolve todos
- Há programas de recompensam a descoberta de problemas de segurança em produtos de grande consumo

Bug bounty hunters
Google vulnerability program

Rewards for qualifying bugs range from $100 to $20,000. The following table outlines the usual rewards chosen for the most common classes of bugs:

Category	Examples	Applications that permit taking over a Google account [1]	Other highly sensitive applications [2]	Normal Google applications	Non-integrated acquisitions and other sandboxed or lower priority applications [3]
Vulnerabilities giving direct access to Google servers					
Remote code execution	*Command injection, deserialization bugs, sandbox escapes*	$20,000	$20,000	$20,000	$1,337 - $5,000
Unrestricted file system or database access	*Unsandboxed XXE, SQL injection*	$10,000	$10,000	$10,000	$1,337 - $5,000
Logic flaw bugs leaking or bypassing significant security controls	*Direct object reference, remote user impersonation*	$10,000	$7,500	$5,000	$500
Vulnerabilities giving access to client or authenticated session of the logged-in victim					
Execute code on the client	Web: *Cross-site scripting* Mobile: *Code execution*	$7,500	$5,000	$3,133.7	$100
Other valid security vulnerabilities	Web: *CSRF, Clickjacking* Mobile: *Information leak, privilege escalation*	$500 - $7,500	$500 - $5,000	$500 - $3,133.7	$100

Comunicação num sistema sem segurança



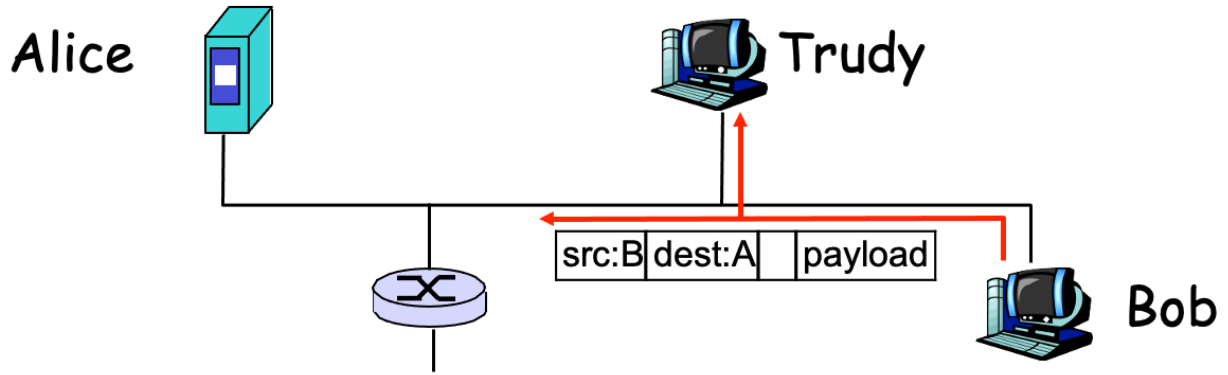
- Canal está acessível ao atacante
- Nomes usados na descrição dos protocolos de segurança:
  - **Alice**, **Bob**, **Carol**, **Dave** – participantes que querem comunicar;
  - **Eve** é usado para um atacante que lê mensagens – *eavesdropper*);
  - **Mallory**/**Trudy** – atacante que pode ler, intercetar, modificar, suprimir ou reintroduzir mensagens nos canais ou tentar passar por um dos participantes;
  - **Sara** – um servidor.

Ataques em sistemas distribuídos através da comunicação

- **Indiscrição** – obtenção de mensagens sem autorização (Eavesdropping)
- **Mascarar-se** ou **pretender** ser outro (Masquerading)
- **Reemissão** de mensagens prévias (Message replaying)
- **Adulteração** do conteúdo das mensagens (Message tampering)
- **Supressão** de mensagens (Message supression)
- **Vandalismo** por impedimento de prestação de serviço (Denial of service attacks)
- **Repúdio** de mensagens, negar a autoria de mensagens
- **Análise de tráfego** (Traffic analysis) procurando padrões que possam indiciar a natureza da comunicação, dos protocolos, etc.

Eavesdropping (exemplo): cópia dos pacotes que transitam na rede

- Packet sniffing:
  - **broadcast** media
  - modo promiscuous da placa **wireless** ou de rede permite capturar todos os pacotes
  - permite a leitura de dados não cifrados
    - (exemplo: passwords ftp, telnet, mas não ssh)
  - exemplo: *Trudy sniffs Bob's packets*

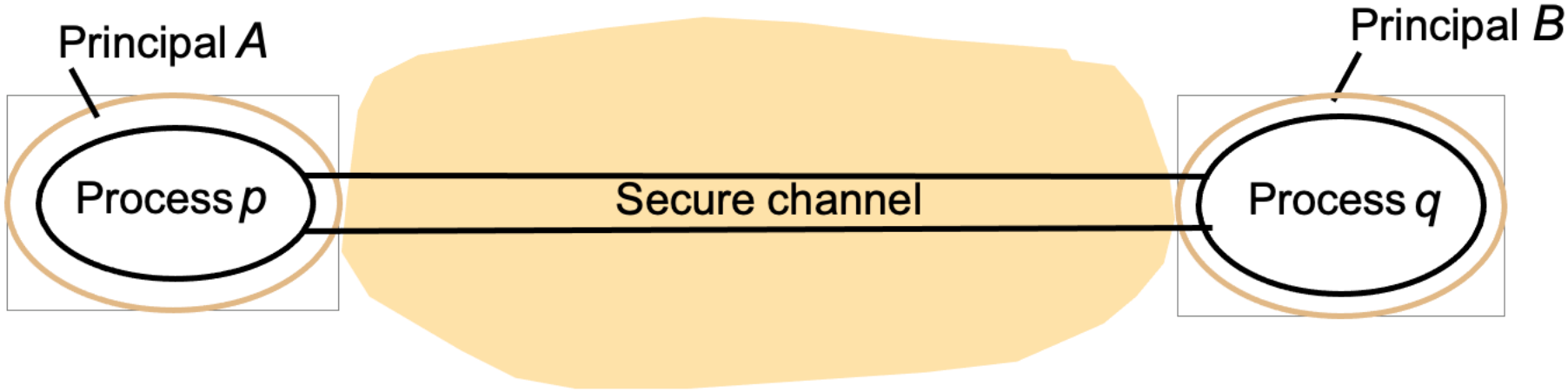


Defesa

- Mecanismos de segurança mais comuns
- Recurso à criptografia para:
  - obter **canais seguros**, imunes à repetição e violação da integridade dos dados e garantir confidencialidade
  - **autenticar** os principais (utilizadores, servidores, etc)
  - **certificar** conteúdos para garantir a sua autenticidade e não repúdio
- Evitar/Esconder padrões regulares na comunicação entre principais
  - (por via de mensagens falsas/inúteis, aleatoriedade)

Canais seguros

- **Objetivo**: trocar dados com **confidencialidade**, **integridade** e **autenticidade**
  - Num canal seguro os interlocutores (A e B) estão autenticados
  - O atacante **não pode** ler/copiar, alterar ou introduzir mensagens
  - O atacante **não pode** fazer replaying de mensagens (replaying = reenvio) O atacante **não pode** reordenar as mensagens



Canais seguros: Em concreto …

- **TLS –** Transport Layer Security (antigo Secure Socket Layer)

"The primary **goal** of the TLS protocol is to provide **privacy** and data **integrity** between two communicating computer applications. When secured by TLS, connections between a client (e.g., a web browser) and a server …"

- **HTTPS –** HTTP over TLS or HTTP Secure

"… HTTPS consists of communication over HTTP within a connection encrypted by Transport Layer Security or its predecessor, Secure Sockets Layer. The main motivation for HTTPS is **authentication of the visited website** and protection of the **privacy** and **integrity** of the exchanged data."

--from wikipedia.

Autenticação/autorização : OAuth

- OAuth um protocolo standard e aberto para a autorização de acesso a recursos partilhados (tipicamente, na web)
- Permite delegar a terceiros o acesso a recursos sensíveis, sem necessidade de lhes expor credenciais (passwords).
- Exemplo:
  - Fotografias - recursos de um utilizador (o dono) armazenadas num servidor (base da confiança) podem ser acedidas por uma aplicação móvel (o cliente) ou por um serviço de impressão, sem que estes tenham acesso às credenciais geridas pelo servidor.

Segurança em Sistemas Distribuídos

Introdução

- **Diferenças essenciais entre os sistemas centralizados e os sistemas distribuídos**
- **Canais seguros e seus requisitos**

Criptografia simétrica

- Utilização da criptografia simétrica na comunicação e autenticação

Criptografia assimétrica

- Utilização da criptografia assimétrica na comunicação e autenticação

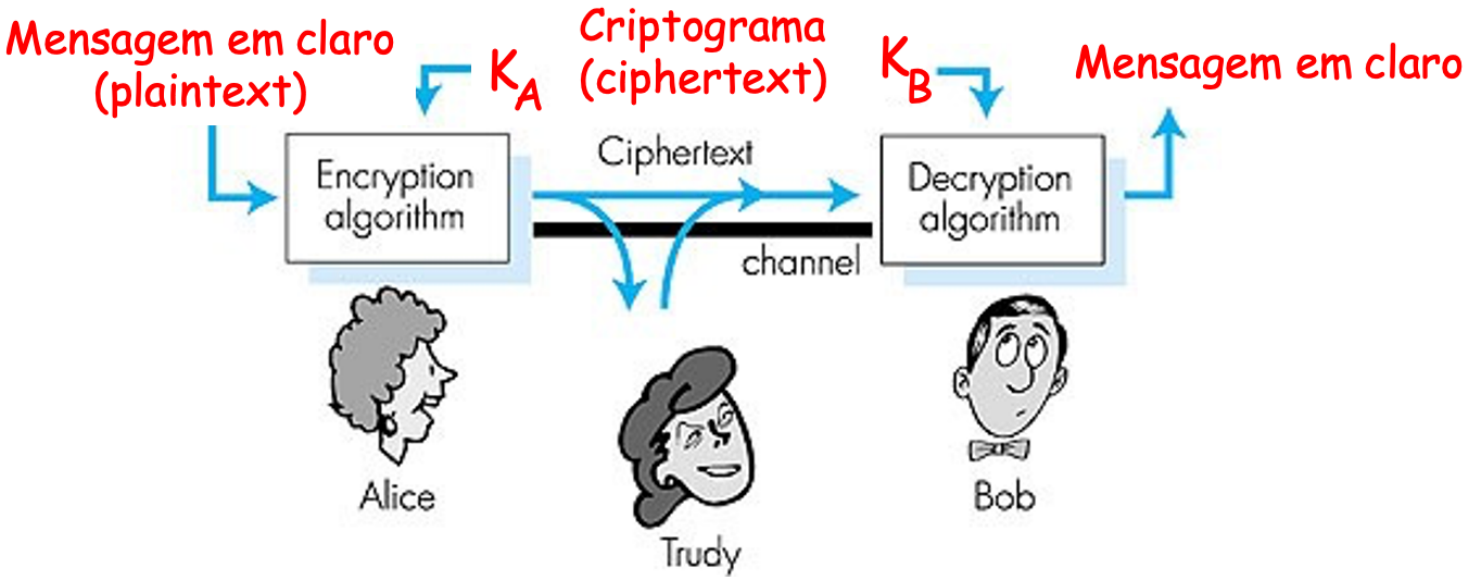
Criptografia

- **Criptografia** é a disciplina que inclui os princípios, meios e métodos de transformação dos dados com a finalidade de:
  - esconder o seu conteúdo semântico estabelecer a sua autenticidade
  - impedir que a sua alteração passe despercebida evitar o seu repúdio

Criptografia

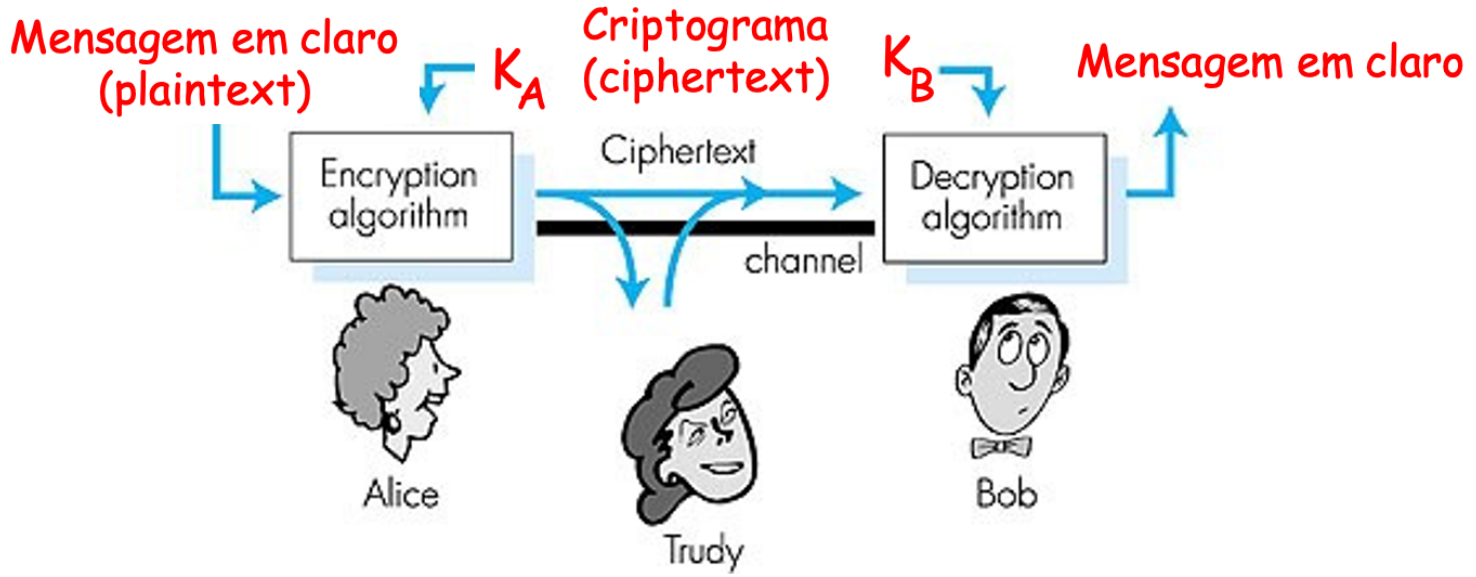
- A transformação dos dados por técnicas criptográficas pode ser:
  - **reversível**: criptografia simétrica ou assimétrica
  - **irreversível**: funções de síntese / hash seguras / message digests
- **chave criptográfica** é um parâmetro utilizado com um algoritmo criptográfico para transformar, validar, autenticar, cifrar ou decifrar dados

Criptografia: chave simétrica



- **Criptografia de chave simétrica**: a mesma chave cifra e decifra informação
- Base mais comum para ofuscar informação e oferecer privacidade de dados (em grande volume)

Criptografia: chave assimétrica



- **Criptografia de chaves assimétrica ou de chave pública**: cifra-se com a chave pública, decifra-se com a chave privada do recetor, ou vice-versa
- Permite autenticação, assinar, comunicação anónima segura.

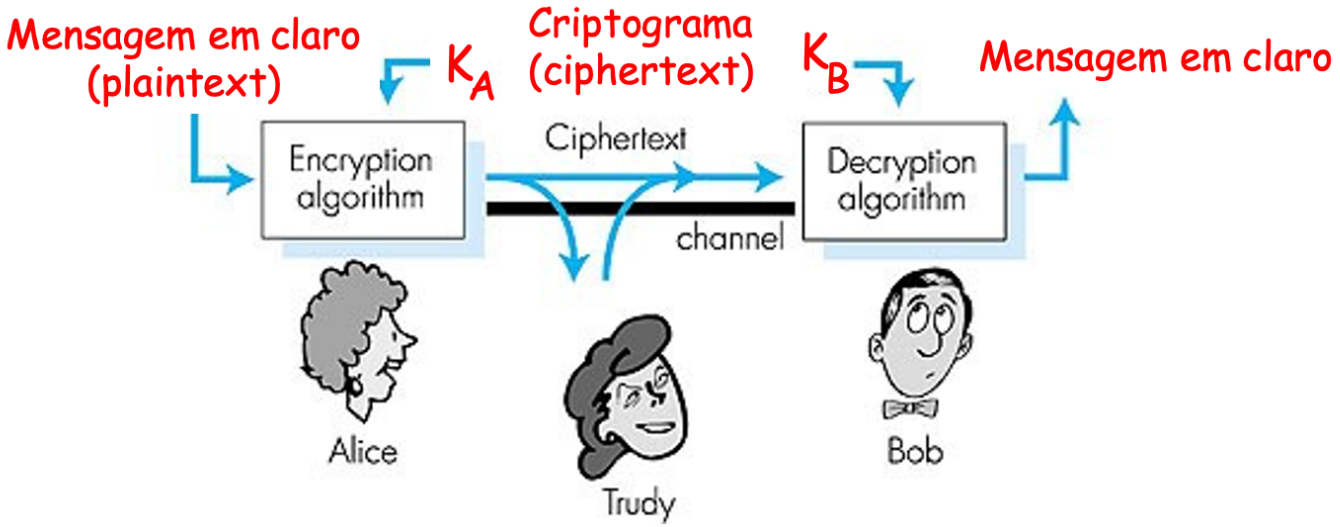
Criptografia: funções de síntese seguras

- As funções de síntese segura são funções de dispersão (hash) não invertíveis e com reduzida probabilidade de colisão.
  - Os dados de entrada são transformados (digeridos) de modo a obter um número reduzido de bits (o hash).
  - Além de não ser possível reconstruir o texto original, o algoritmo será seguro quando é computacionalmente caro produzir uma colisão, ou seja alterar o texto original de forma mínima e produzir o mesmo hash.
- Usam-se para garantir integridade e gerar assinaturas digitais de pequena dimensão.

Criptografia: derivados

- Por combinação das primitivas criptográficas anteriores é possível construir derivados com garantias mais elaboradas:
  - **Assinaturas digitais** – atestam a procedência e autenticidade de um documento, evitam o seu repúdio.
  - **Certificados** – permitem atestar a identidade de principais com base em informação pública. Utiliza entidades terceiras idóneas (provedor de certificados).

A linguagem criptográfica



- **Criptografia de chave simétrica**: as chaves de cifra e de decifra são idênticas
- **Criptografia de chaves assimétrica ou de chave pública**: cifra-se com a chave pública, decifra-se com a chave privada do recetor, ou vice-versa

Eficácia da Criptografia

- O algoritmo criptográfico será tanto melhor quanto mais difícil for obter o texto original a partir do texto cifrado, sem se conhecer a chave.
- A eficácia de um ataque depende de dois fatores:
  - Algoritmo de cifra
  - Cardinalidade do domínio da chave, isto é, número de bits da chave
- Métodos de ataque:
  - "**Força bruta**" - baseia-se na exploração sistemática de todas as chaves possíveis
  - **Criptoanalíticos** - baseia-se em explorar os métodos matemáticos utilizados em criptografia para descobrir como decifrar os dados (ou diminuir o número de possibilidades)

Eficácia da Criptografia

- Nenhum algoritmo criptográfico é inteiramente seguro se o número de bits da chave tornar um ataque "força bruta" realista no quadro de dois fatores:
  - O "valor" da informação
  - A capacidade computacional do atacante

Ataques de Força Bruta

- Supondo que é possível arranjar mil milhões (10^3 x 10^6) de computadores, capazes de testarem mil milhões de chaves por segundo cada um, então é possível testar 10^18 chaves por segundo.

- Uma chave de 128 bits tem cerca de 3.4 x 10^38 combinações (10 levantado a 38). Com o poder computacional indicado, seriam necessários 10^13 anos para completar o ataque
  - Curiosidade: estima-se que a idade do universo é de cerca de 10^10 anos.

- Uma chave de 54 bits tem cerca de 6 x 10^16 combinações. Com o poder computacional indicado, seria necessário menos de um segundo para descobrir a chave.

- **Conclusão**: chaves geradas de forma "totalmente" aleatória, e com um número de bits suficiente, são relativamente seguras quando sujeitas a ataques do tipo "força bruta"

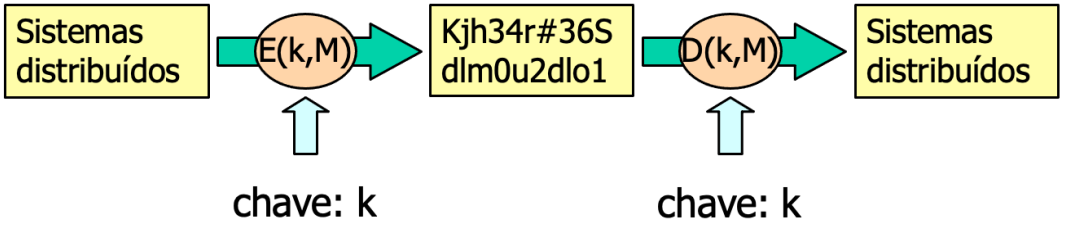
Confiança nos algoritmos criptográficos

- **Não é viável** a prova matemática que um algoritmo criptográfico não tenha falhas e fragilidades perante ataques criptoanalíticos.
- Consegue-se provar exatamente o contrário, através de exemplos.
- A segurança de um método é pois baseada em o mesmo ser público e ser sujeito ao escrutínio por especialistas.

- **A segurança deve ser resultado do segredo das chaves e não do segredo do algoritmo criptográfico**

Criptografia Simétrica

- Parceiros devem partilhar chave secreta
- Mensagem é cifrada e decifrada com a mesma chave
  - E(k,M) = X
  - D(k,X) = M
- Que garantias dá? (confidencialidade, autenticação?)
- Garante confidencialidade das mensagens
  - Dado X, deve ser computacionalmente impossível obter M sem saber K, mesmo conhecendo E e D



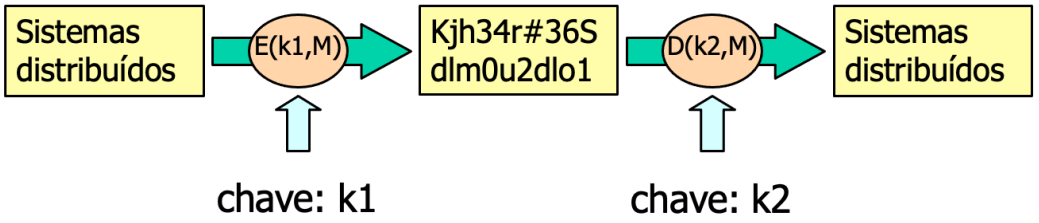
Algoritmos de Criptografia Simétrica mais comuns

- DES – Data Encryption Standard: chaves com 56 bits. OBSOLETO (< 1 dia) Triple DES - DES reforçado – 3 chaves DES : 168 bits. OBSOLETO em 2030
- IDEA – International Data Encryption Algorithm: chave com 128 bits, com origem na Europa. Após vários anos de utilização e divulgação pública não se conhecem ataques com êxito. (circa 2010)
  - 2016 – Atacado! Revelou fraquezas em certos grupos chaves. Patentes expiraram em 2012.
- AES/Rijndael – Advanced Encryption Standard: Nova norma U.S.A. Definida por concurso público, ganho por uma equipa belga. Admite chaves de 128, 192 ou 256 bits. Atacado!
- **IDEA & AES foram atacados!!! Perderam o equivalente a 2 bits na dimensão das chaves… (são seguros?)**

Criptografia Assimétrica

- Cada entidade tem duas chaves
  - Chave **privada** (K_{priv}) é conhecida apenas pelo seu dono Chave **pública** (K_{pub}) pode ser conhecida por **todos**
  - A partir de K_{pub} é impossível derivar K_{priv}

- Mensagem é cifrada com uma chave e decifrada com a outra
  - $E(K_{pub},M) = X$; $D(K_{priv},X) = M$
    - Garante o quê? (confidencialidade, ~~autenticação~~)
  - $E(K_{priv},M) = X$; $D(K_{pub},X) = M$
    - Garante o quê? (~~confidencialidade~~, autenticação)



Criptografia Assimétrica

- Cada entidade tem duas chaves
  - Chave **privada** (K_{priv}) é conhecida apenas pelo seu dono Chave **pública** (K_{pub}) pode ser conhecida por **todos**
  - A partir de K_{pub} é impossível derivar K_{priv}

- Mensagem é cifrada com uma chave e decifrada com a outra
  - $E(K_{pub},M) = X$; $D(K_{priv},X) = M$
    - Garante o quê? (confidencialidade, ~~autenticação~~)
    - Conhecendo K_{pub} e X deve ser computacionalmente impossível obter M sem saber K_{priv}. Só receptor conhece K_{priv}.
  - $E(K_{priv},M) = X$; $D(K_{pub},X) = M$
    - Garante o quê? (~~confidencialidade~~, autenticação)
    - A partir de K_{pub} deve ser computacionalmente impossível obter K_{priv}. Só quem possui K_{priv} pode gerar X.

Algoritmos de Criptografia Assimétrica mais comuns

- **RSA** – algoritmo mais usado. Chave recomendada: 2048 bits.
  - Patente RSA expirou em 2000
  - Baseado na fatorização de números primos
- **Algoritmos de curva elíptica** – método para gerar pares de chaves pública/privada baseado nas propriedades das curvas elípticas. Usa chaves de dimensão menor.
  - Baseado no cálculo de logaritmos discretos ($b^k = g$)
- **Em ambos os casos a segurança advém do recurso a problemas considerados matematicamente difíceis.**

Curiosidade: algoritmo RSA (Rivest, Shamir e Adelman)

RSA Challenge (768 bits quebrado)
https://en.wikipedia.org/wiki/RSA_numbers#RSA-768

Para gerar as chaves

- Escolhem-se dois números primos grandes , P e Q (P e $Q > 10^{100}$)
  - $N = P*Q$, $Z=(P-1) * (Q-1)$
- Escolhe-se d, que pode ser qualquer número menor que N que seja primo em relação a Z
- Calcula-se e, que é um número tal que e.d-1 é divisível por Z
- A chave privada é (N,e); a chave pública é (N,d)

A função de cifra é

- $E((e,N), M) = M^e \bmod N = c$

A função de decifra é

- $D((d,N), c) = c^d \bmod N = M$

Exemplo:

P=5
Q=11
N=55
Z=40

d=3
e=7

$E(2)= 2^7 \bmod 55$
$= 18$
$D(18)=18^3 \bmod 55$
$= 2$

NOTA: Segurança do método depende da dificuldade de factorizar N em P e Q

Funções de Síntese

- Objetivo: uma função de síntese segura H (Message Digest ou Secure Hash) deve produzir um sequência pequena de bits (128, 160, 512,...) que permita identificar uma mensagem de qualquer dimensão

- Propriedades:
  1. Calcular H(M) é fácil (computacionalmente)
  2. Dado H(M) é computacionalmente impossível calcular M
  3. Dado M é computacionalmente impossível descobrir M2 tal que H(M) = H(M2)

- As funções de síntese com estas propriedades dizem-se "Secure one-way hash functions" ou "funções de dispersão unidirecionais seguras".
  - Porquê unidirecional e seguro?

- Para que é que serve?
  - Usado para garantir integridade dos dados

Funções Seguras de Síntese

- Função segura de síntese MD5 (abandonada)
  - Calcula sínteses de 128 bits num processo em 4 fases
  - Uma das funções seguras de síntese computacionalmente mais eficientes
    - Conhecidos ataques que permitem gerar colisões.

- A função SHA-1 ainda muito usada, mas já declarada insegura
  - Norma USA. Conhecida publicamente
    - Conhecidos ataques que permitem diminuir o espaço de pesquisa para uma colisão para $O(2^{63})$ (em vez de (2^{80}))
    - Está-se a migrar para SHA-2 e SHA-3
  - Produz uma síntese de 160 bits

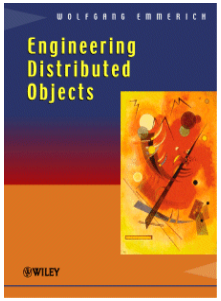
- Substitutos: SHA-2, SHA-3 (hashes de 224 a 512 bits)

O desempenho dos diferentes métodos

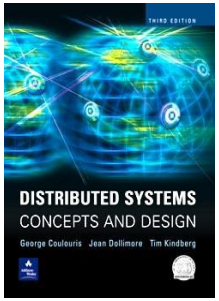
CIPHER	Débito (MB/s)
AES	< 100
Twofish	60
Serpent	32
IDEA	35
DES	32
RSA 1024 Cifrar	~3
RSA 1024 Decifrar	~0.2
RSA 2048 Cifrar	~1.5
RSA 2048 Decifrar	~0.04
MD5	255
SHA-1	153
SHA-256	111
SHA-512	99
CRC32	253
Adler32	920

https://www.cryptopp.com/benchmarks.html

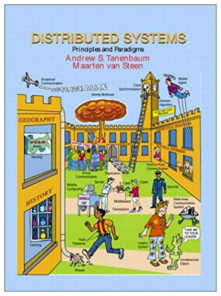
Bibliografia



From: Wolfgang Emmerich
Engineering Distributed Objects
John Wiley & Sons, Ltd 2000



From: Coulouris, Dollimore and Kindberg
Distributed Systems: Concepts and Design
Edition 4 © Addison-Wesley 2005



From: Andrew S., Tanembaum and Van Steen, Maarten
Distributed Systems: Principles and Paradigms
Edition 2 © Pearson 2013

Questões?