Assinatura Digital

Autenticação e Integridade da Informação

Assinaturas digitais

- Identificar inequivocamente o autor de um documento (autenticidade)
- Impedir alterações do documento (integridade)
- Impedir que o autor repudie o conteúdo *a posteriori* (não-repudiação)

- As assinaturas não fazem sentido isoladas;
  só junto do texto a que se referem
  (ou com uma referência para o texto)

Assinaturas digitais: Técnica base de Autenticação

- Assinatura do documento T por A
  - $\{T\}_{\text{Kprivada A}}$
- Validação da assinatura:
  - T == $\{\text{assinatura}\}_{\text{Kpública A}}$

O sistema de cifra tem de ser de *chave assimétrica* para garantir a não repudiação da assinatura
(se A e B partilhassem uma chave simétrica, seria impossível provar quem assinou)

Assinaturas digitais: Técnica base de Integridade

- Garantir que o documento corresponde à assinatura
  - A calcula a assinatura e anexa-a ao documento T
    - T, {T}Kprivada A
  - B obtém a chave pública de A
  - Decifra a $\{\text{Assinatura de T}\}_{\text{Kpública A}}$
  - Verifica se o valor resultante é igual a T

- Este algoritmo simples tem o problema da cifra de documentos longos com chave assimétrica
- Solução: ***Em vez de assinar T assinar um resumo de T***

Funções de Resumo ou Dispersão (*Digest*/*Hash*)

Função **Resumo** (***digest***)

Recebe um texto (possivelmente longo) e devolve uma sequência de bits de comprimento fixo (e.g., 160 bits)

Propriedades das funções resumo

Eficiente – dado P é fácil calcular H(P)

Não-invertível – dado H(P) é impossível determinar P' tal que H(P') = H(P)

Resistente a colisões – difícil encontrar P1, P2 tais que H(P1) = H(P2)

A esta situação chama-se **colisão** dos resultados da função

Funções de *Hash* não invertíveis

- **As funções de *hash* não invertíveis têm como objetivo criar um resumo único semelhante a uma impressão digital de um conteúdo digital muito mais extenso**
  - São não invertíveis porque é computacionalmente impossível reconstruir o conteúdo original a partir do resumo
  - A probabilidade de colisão (dois textos diferentes produzirem a mesma assinatura) deve ser mínima
  - Mudanças pequenas no texto devem produzir resumo muito diferentes (valores de *hash* estão distribuídos uniformemente)

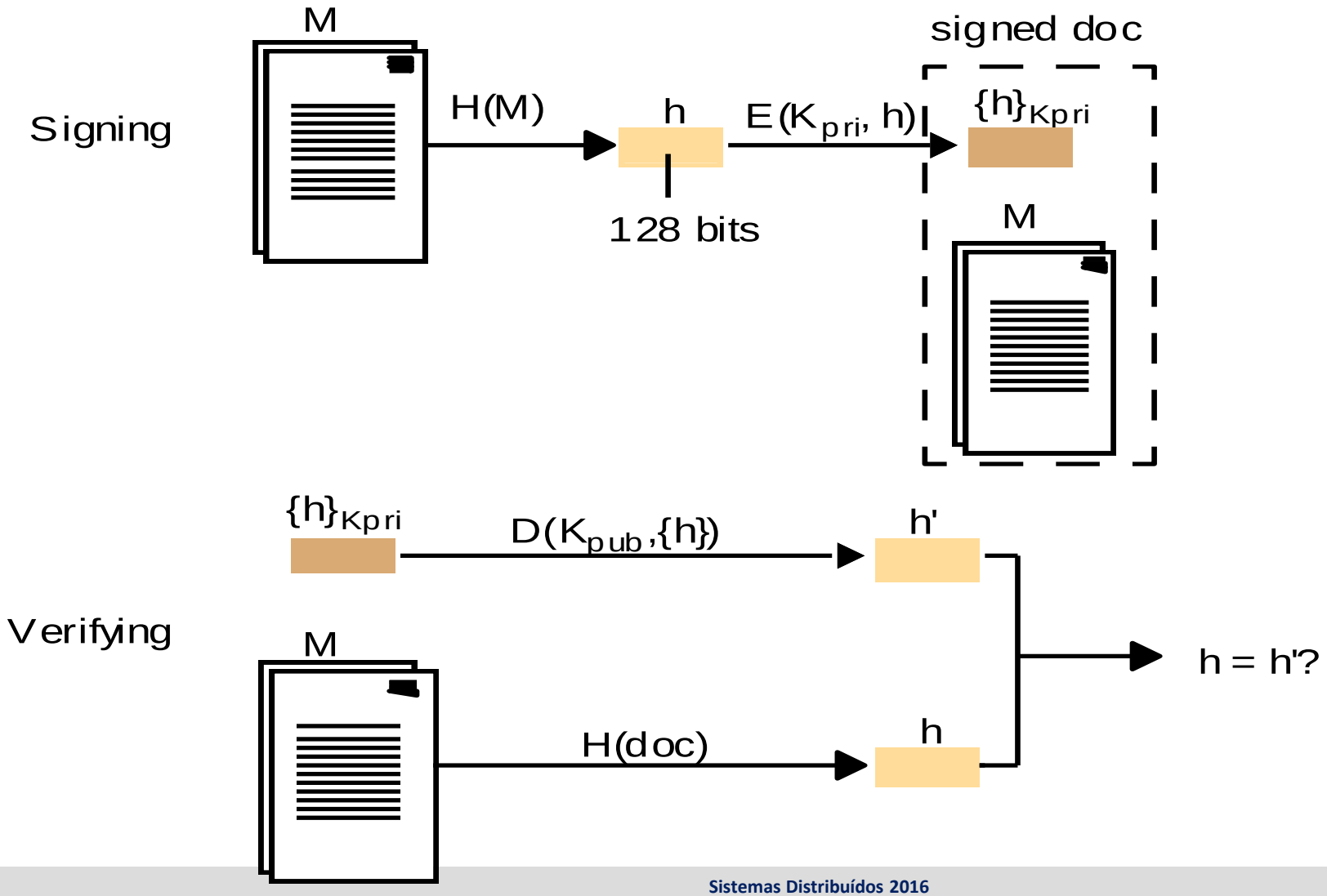
Funções Resumo (*Digest*)

- A função **MD5** [Rivest92].
  - A informação é processada em blocos de 512 bits (16 palavras de 32 bits) e o valor do resumo é uma palavra de 128 bits
  - Em cada etapa é calculado um novo valor de resumo baseado no valor anterior e no bloco seguinte de 512 bits da mensagem

Message	**MD5 Digest**
I need a raise of $10,000.	9i5nud5r2a9idskjs2tbuop2ildax
I need a raise of $100,000**.**	8m4ikijuelaidsfg8asyfnasdfgll
I need a raise of $1,000,000.	4M9i2t8c7h4361712t1h4e1d1otg7

- A função **SHA-1** é a norma dos EUA
  - Resumo de 160 bits
- A mais recente função **SHA-2** produz um resumo de 256 a 516 bits

Assinatura Digital



Protocolo de Assinatura Digital

A envia para B a informação T e a respectiva assinatura constituída pelo resumo da informação obtido pela função resumo D, cifrado com a chave privada de A

1. **A -> B:** **T, A, {D(T)} Kprivada A**

B pede ao servidor de autenticação a chave pública de A

2. **S_{AUT} -> B:** **A, Kpública A**

Com a chave pública de A (Kpública A), B decifra a assinatura

3. **B:** **calcula D(T)**
4. **B:** **decifra {D(T)} com Kpública A**

Se for idêntica, a mensagem não foi modificada, garante a integridade e tem a certeza que foi A que a enviou, garante a autenticação

5. **B:** **Compara os dois**

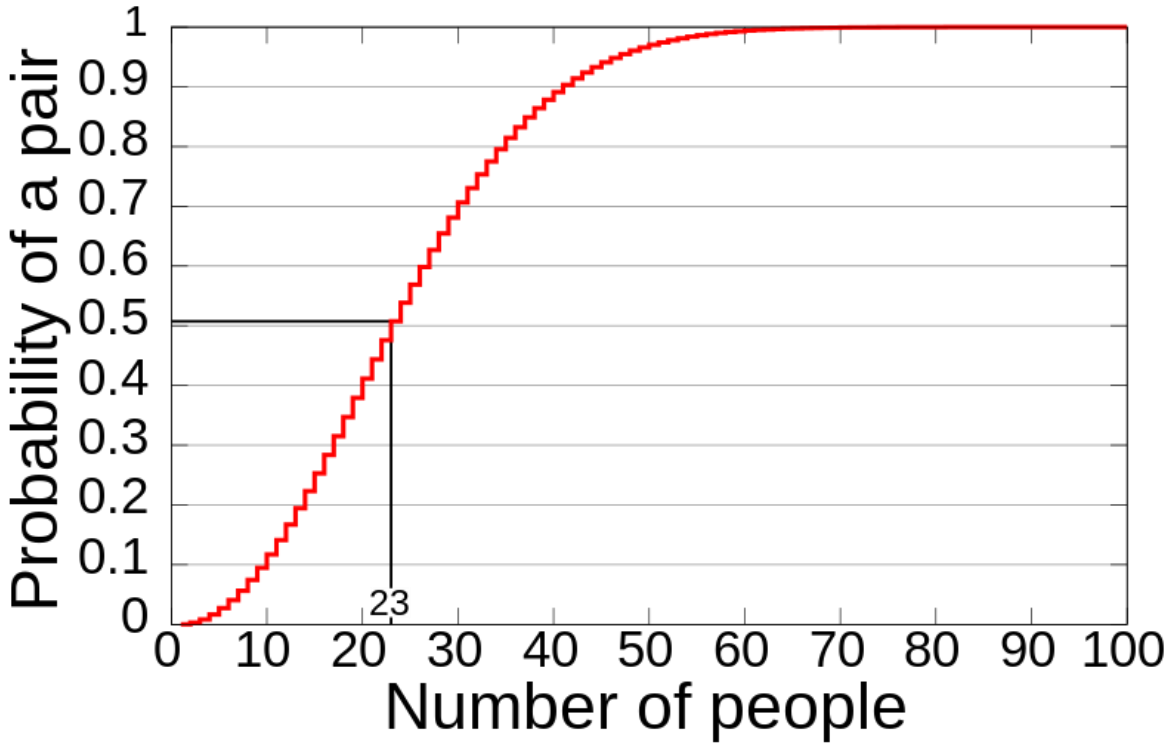
Porque é que deve ser difícil encontrar colisões?

Se não, seria fácil forjar assinaturas digitais

Como?

Paradoxo do Aniversário

- Qual a probabilidade de duas pessoas na aula terem o mesmo dia de aniversário?
  - Para n>=23, p>50%
  - Número de pares de aniversários = C(23,2) = 22 * 23 / 2 = 253 pares



Ataque do Aniversário

- Birthday Attack
  - Quantas operações são necessárias para encontrar uma colisão num resumo de m bits?
  - Resposta : $2^{m/2}$ (muito menos do que 2^{m})

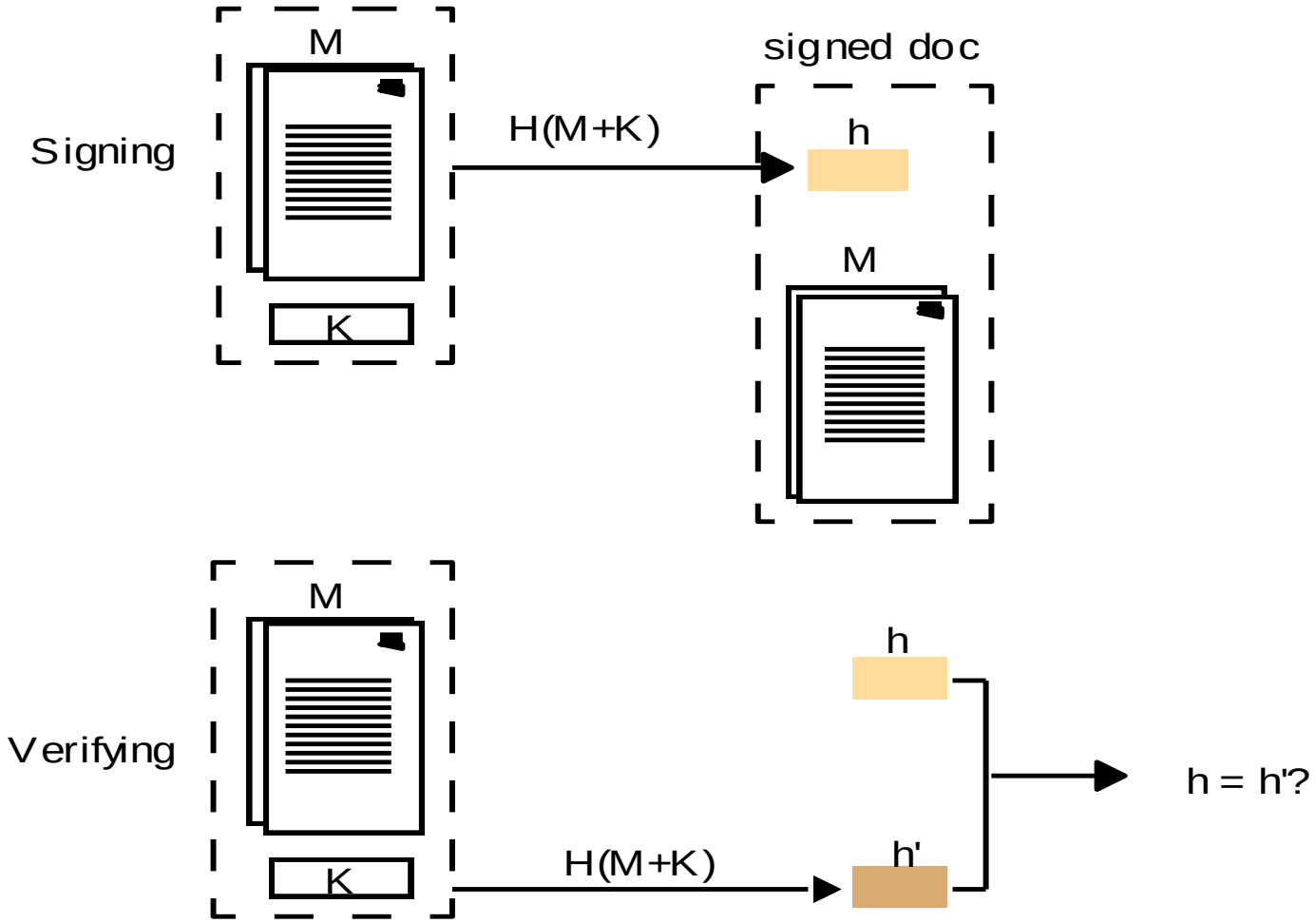
MACs: “Assinaturas” *low-cost*

- Funções de *hash* são muito mais rápidas que as funções de cifra
- Seria interessante ter método de assinatura digital que não implicasse cifra

…Como?

- Assumindo que interlocutores partilham segredo K é possível
  - Por exemplo, K pode ser chave de sessão em cifra híbrida

MAC – *Message Authentication Code*

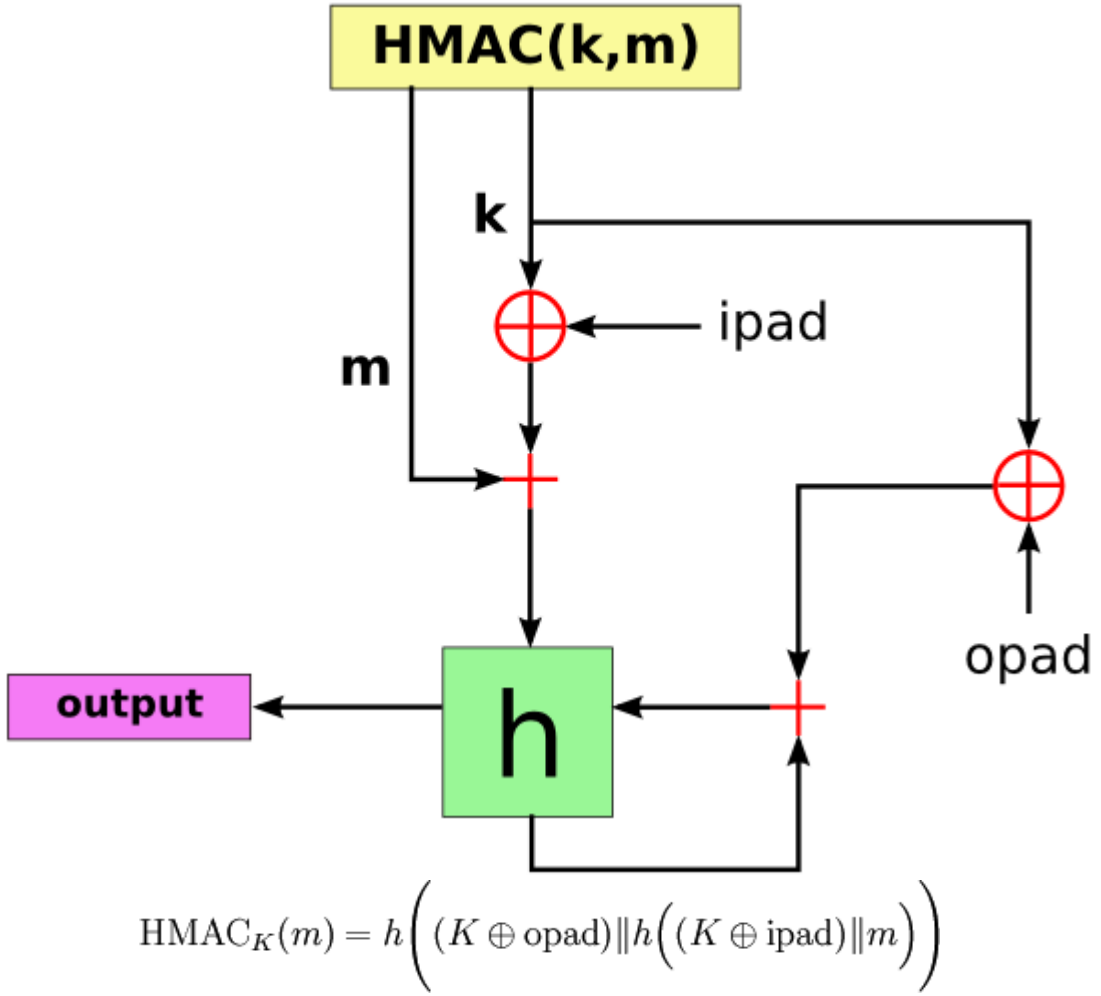


MACs: Discussão

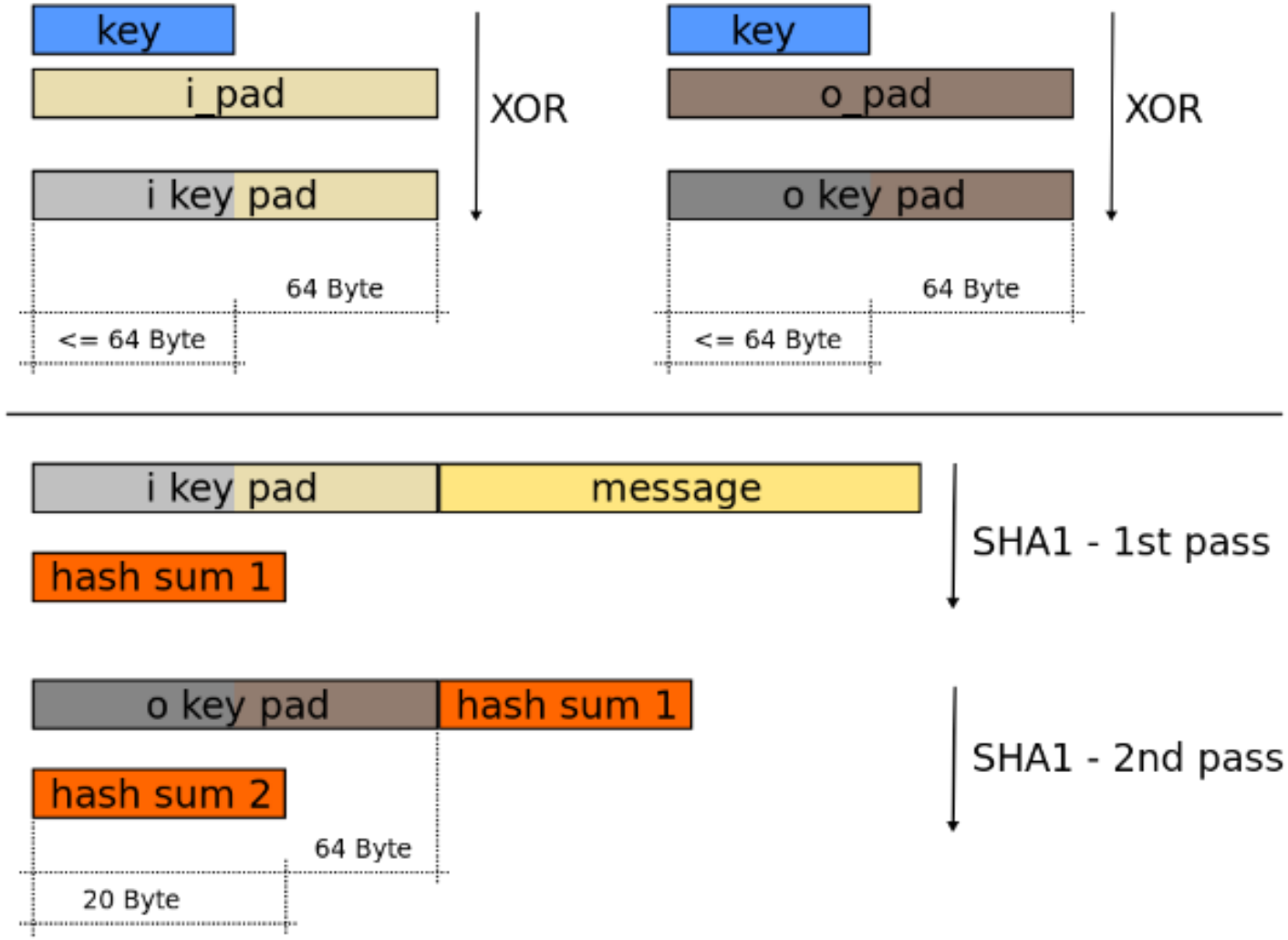
- Quem pode validar mensagens assinadas?
- Que requisitos são assegurados?
  - Autenticidade dentro do grupo conhecedor de K e Integridade

HMAC

- Garante integridade e autenticidade de mensagens
- Pressuposto: interlocutores partilham segredo (ou "chave secreta" K_{AB})
- Exemplo (HMAC):
  - m = mensagem
  - k = segredo K_{AB}
  - opad,ipad = padding fixo
  - h = função de *hash*



HMAC

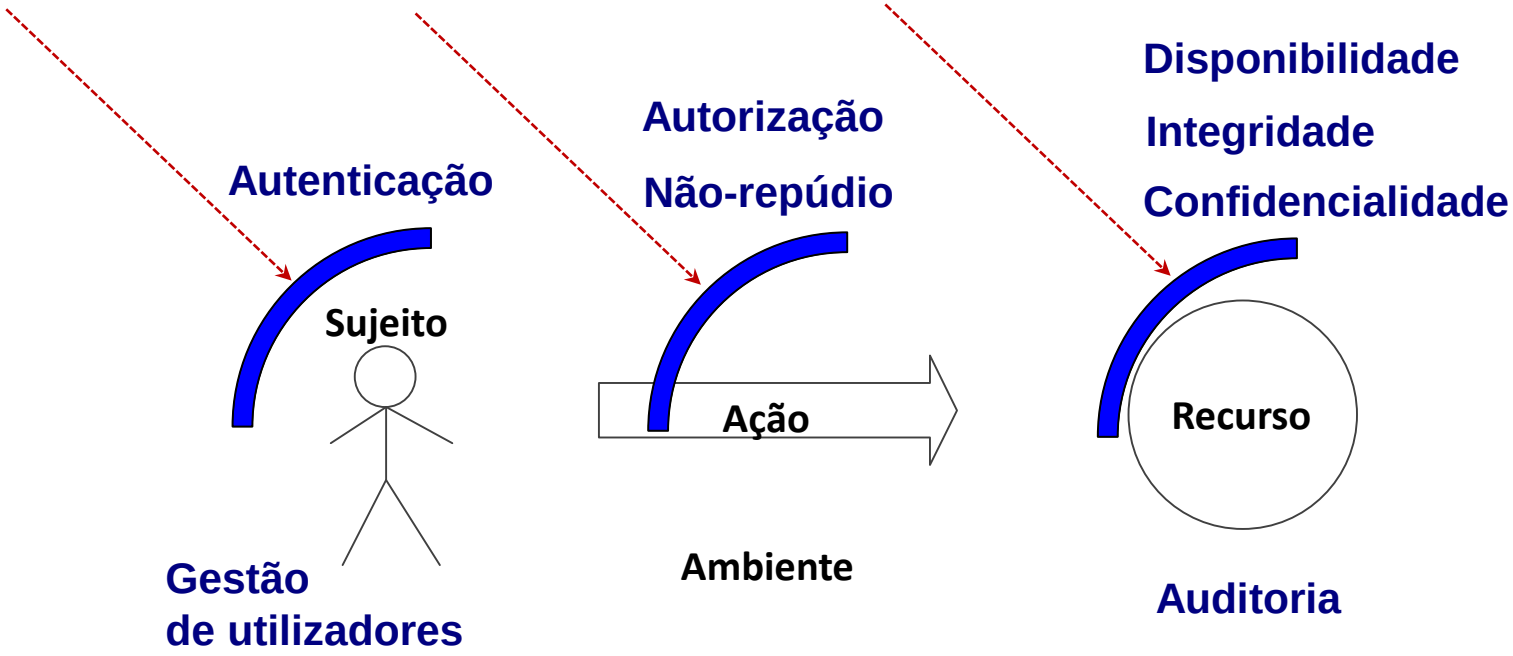

key
i_pad
XOR
i key pad
64 Byte
<= 64 Byte
key
o_pad
XOR
o key pad
64 Byte
<= 64 Byte
i key pad
message
SHA1 - 1st pass
hash sum 1
o key pad
hash sum 1
SHA1 - 2nd pass
hash sum 2
64 Byte
20 Byte

MAC (Código de autenticação de mensagem)

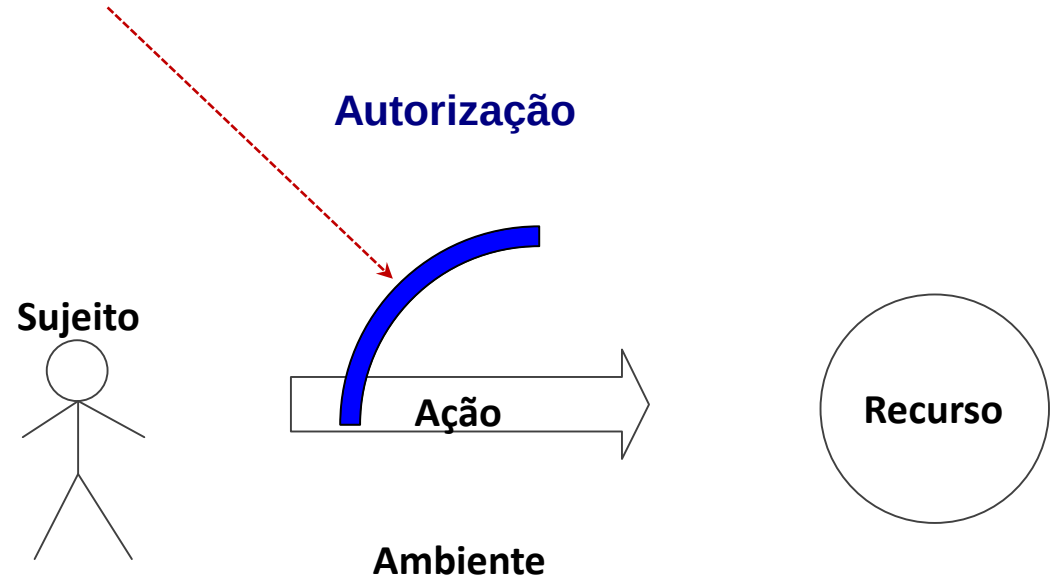
- Emissor envia <m, HMAC(m)>
- Receptor calcula HMAC da mensagem recebida e compara com HMAC recebido
- Se são iguais, há garantia de integridade e que foi produzida pelo (outro) detentor de K_{AB}

- Qual é a propriedade das assinaturas digitais não oferecida? Porquê?

Sistemas Distribuídos:
Políticas e Mecanismos de Segurança



Autorização



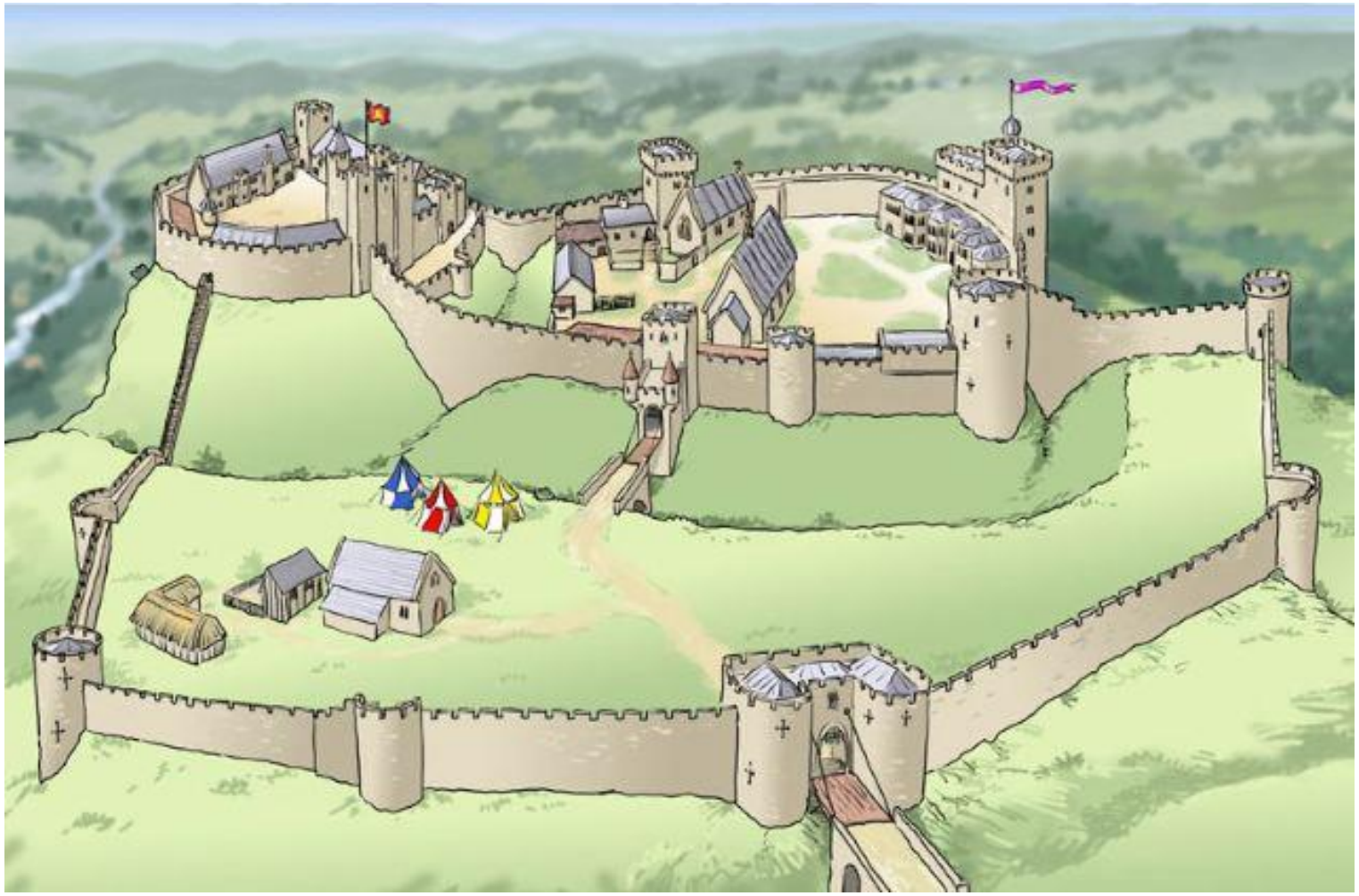
Controlo do Nível de Segurança da Informação

- Esta política considera um **controlo mandatório** sobre a segurança dos objetos, não permitindo aos agentes que a modifiquem
  - Política oriunda da visão militar da segurança
  - As políticas habituais consideram que o agente tem um controlo discricionário sobre os objetos
- Um agente só tem acesso a informação se realmente tiver necessidade de conhecer (*need to know*)
  - O sistema aplica regras estritas para determinar quem tem acesso a quê

Controlo do Nível de Segurança da Informação

- Os objetos são classificados de acordo como o seu nível de confidencialidade
  - Ex.: Muito Secreto, Secreto, Restrito, Não Classificado.
- A informação é também classificada em compartimentos de acordo com o assunto a que diz respeito
  - Ex.: NATO, Comunicações, etc.,
- Os agentes têm autorizações (*clearances*) que definem os compartimentos e o nível de segurança a que podem aceder
  - *Clearance level*
- Controlo de acesso dos agentes
  - Não podem ler informação classificada acima do seu nível
  - Não podem escrever informação classificada abaixo do seu nível

Controlo do Nível de Segurança da Informação



© selfless security

Controlo de direitos de acesso

- Modelo conceptual
  - Os objetos são protegidos por um monitor de controlo de referências
  - Cada agente, antes de poder efetuar uma ação sobre um objeto, tem que pedir autorização ao monitor
  - O monitor verifica se o agente está ou não autorizado através de uma matriz de direitos acesso

Controlo dos Direitos de Acesso

- Um **Monitor de Controlo de Referências** valida,
  quando uma operação é efetuada,
  se o agente tem direito de a executar
  - Os objetos só podem ser acedidos através do monitor de controlo de referências
  - Os objetos têm de ser univocamente identificados e
    o identificador não pode ser reutilizado sem precauções adicionais
  - Num sistema multiprogramado a informação relativa à matriz é mantida dentro
    do espaço de isolamento do núcleo
  - Esta situação é, obviamente, diferente numa rede
- Os ataques a esta política visam essencialmente subverter o isolamento entre os agentes
  - Mais do que procurar alterar a matriz ou
    eliminar a verificação do monitor de controlo de referências

Matriz de direitos de acesso

	Objectos			
Agentes	O1	O2	O3	O4
A1	R	RW	RX	---
A2	RX	---	RW	R

- Decomposição da tabela
  - Listas de controlo de acesso (*Access Control Lists*, **ACL**s)
    - Guardadas junto de cada objecto
  - Capacidades (*capabilities*)
    - Guardadas junto de cada agente
- A autenticação dos agentes é fulcral
  - Para determinar a parcela da ACL que lhe é aplicável
  - Para distribuir as capacidades correctas

ACLs vs Capacidades

- Capacidades permitem descentralizar autorização
  - Servidor analisa a capacidade enviada no pedido para determinar se cliente tem direito ao que pede
  - Não é necessário contactar nenhuma entidade centralizada que armazena ACLs
- Também suportam delegação facilmente
- Capacidade análoga a uma chave do mundo real
- E tem limitações análogas:
  - Pode ser roubada
  - Revogar acesso a alguém que tem a chave é difícil

Como lidar com isto?

Controlo dos Direitos de Acesso

- O Monitor de Controlo de Referência valida se o agente tem direito de executar a operação
- Duas opções:
  - A informação relativa à matriz é mantida dentro do espaço de endereçamento do servidor que se supõe seguro - ACL
  - É enviada uma capacidade de cada vez que o cliente pretende utilizar o objeto
    - Capacidade: Ticket ou Certificado de Autenticação + Direitos

Amoeba

- Sistema operativo distribuído baseado num micro-núcleo
- Capacidades para autenticação e autorização
- As capacidades são armazenadas no espaço de endereçamento dos utilizadores
- Cifra para proteger os campos de direitos
- Mecanismos para permitir revogar direitos

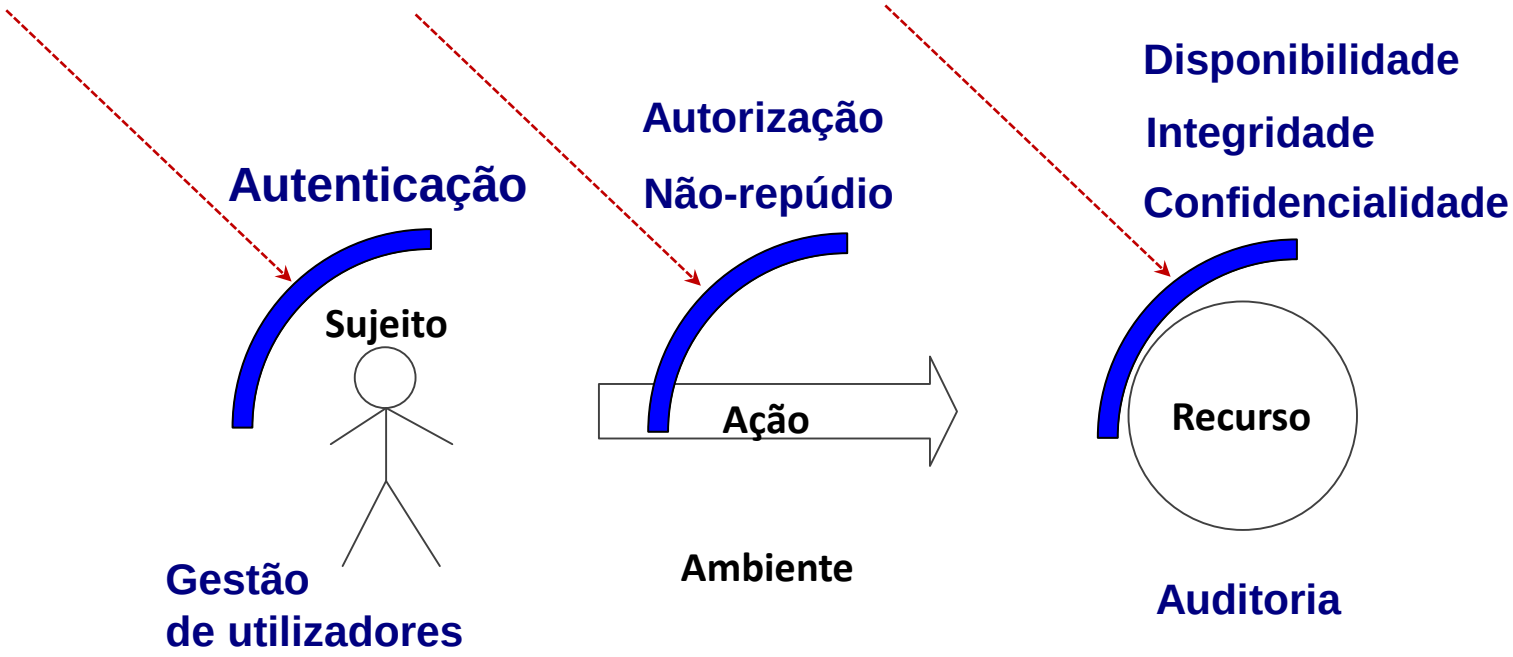
Amoeba: Estrutura das capacidades

48 bits	24 bits	8 bits	48 bits
Porto do Servidor	Número do Objeto	Direitos	Campo de verificação

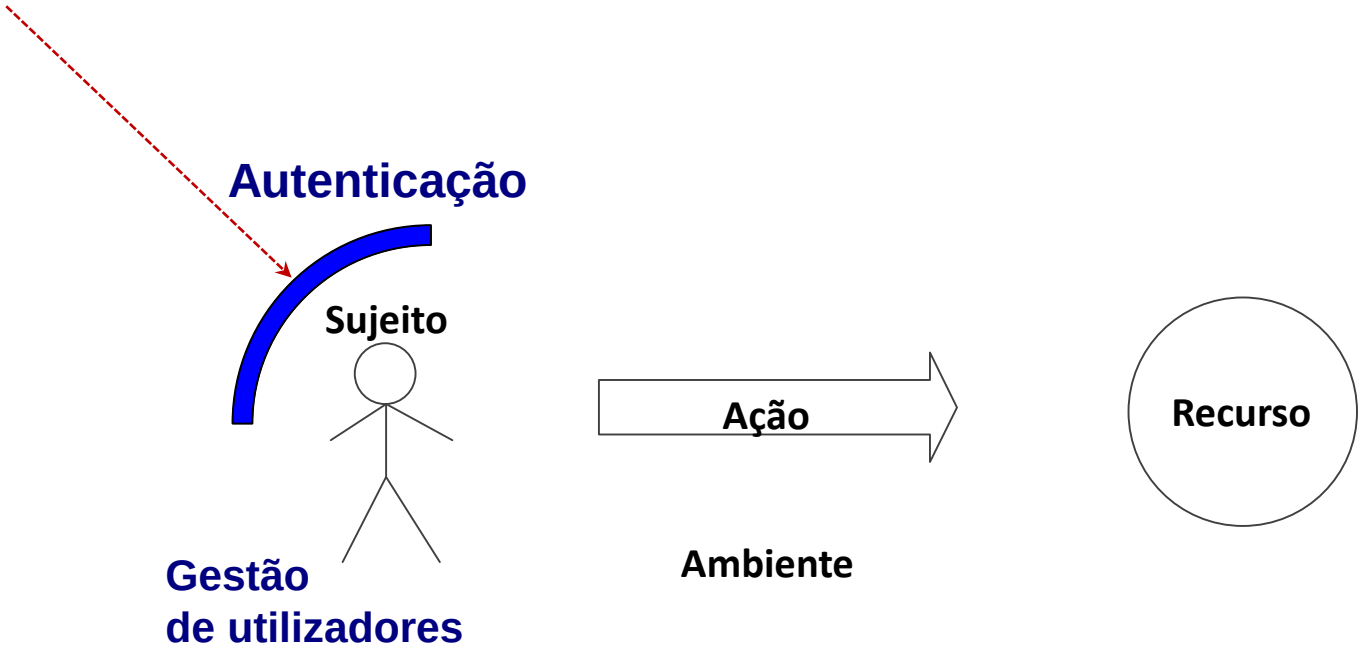
As capacidades são constituídas por quatro campos:
1 – Porto do servidor que gere o objeto: 48 bits
2 – Número do objeto (só com significado para o servidor): 24 bits
3 – Direitos sobre o objeto (1 bit por cada operação): 8 bits
4 – Número aleatório (usado para proteção das capacidades) : 48 bits

Cifra da capacidade para garantir que não é modificada. Problema: como modificar os direitos

Sistemas Distribuídos:
Políticas e Mecanismos de Segurança



Autenticação



Autenticação

- A autenticação baseia-se sempre em o sistema **apresentar um desafio** que o agente deve saber responder
- O desafio pode ser:
  - Fornecer uma informação que deve ser secreta
    - Senha
  - Apresentar um identificador físico
    - Cartão, Chave física
  - Fornecer informação biométrica
    - Impressões digitais, estrutura da íris

Autenticação com chaves simétricas
(chaves secretas)

Protocolo Simples de Autenticação

1) C ->S: "Iniciar Sessão"
2) S ->C: D
3) C ->S: $\{D\}_{Kcs}$

O segredo neste caso é a chave K_{cs}

- A chave poderia ser obtida por um *hash* da password, um segredo entre C e S

O protocolo tem vários problemas:

- É necessário estabelecer a chave secreta entre o cliente e o servidor
- Não é recíproco, só autentica o cliente
- O valor de D tem de variar, senão pode ser reutilizado

Frescura

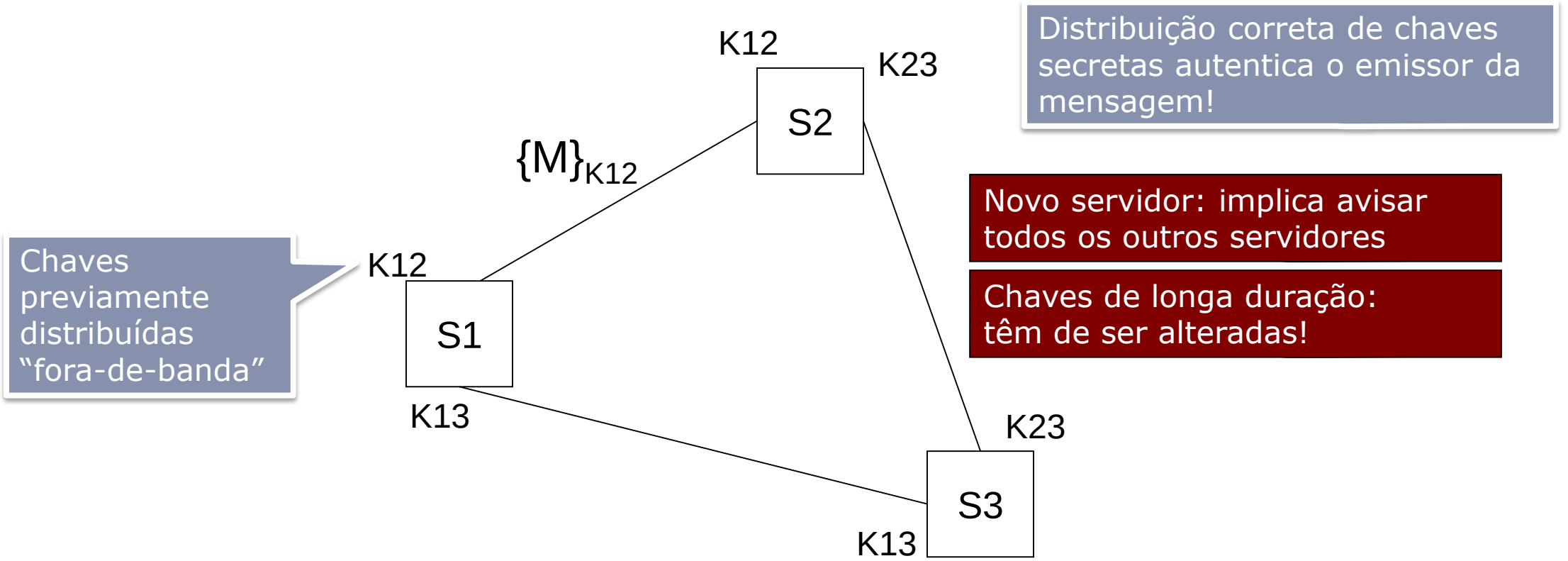
- Necessidade de *nounces* ou carimbos temporais para evitar a reutilização das mensagens

- Em cifra simétrica o problema principal é a partilha da chave

- Se o protocolo se basear em chaves apenas conhecidas do agente e de uma autoridade de distribuição de chaves (*KDS – Key Distribution Service*)
  podemos controlar a partilha do segredo

Distribuição manual de chaves simétricas

Distribuição de Chaves e Autenticação



Distribuição correta de chaves secretas autentica o emissor da mensagem!

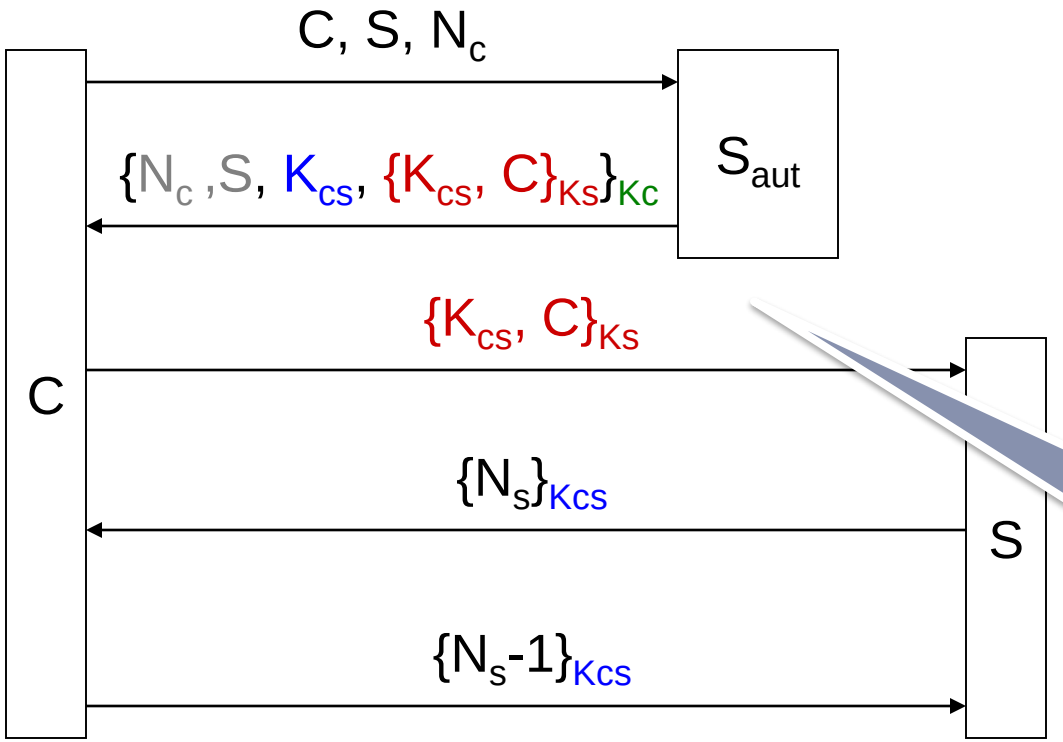
Novo servidor: implica avisar todos os outros servidores

Chaves de longa duração: têm de ser alteradas!

Chaves previamente distribuídas "fora-de-banda"

Distribuição de chaves simétricas com Entidade Terceira Confiada (*Trusted Third Party*)

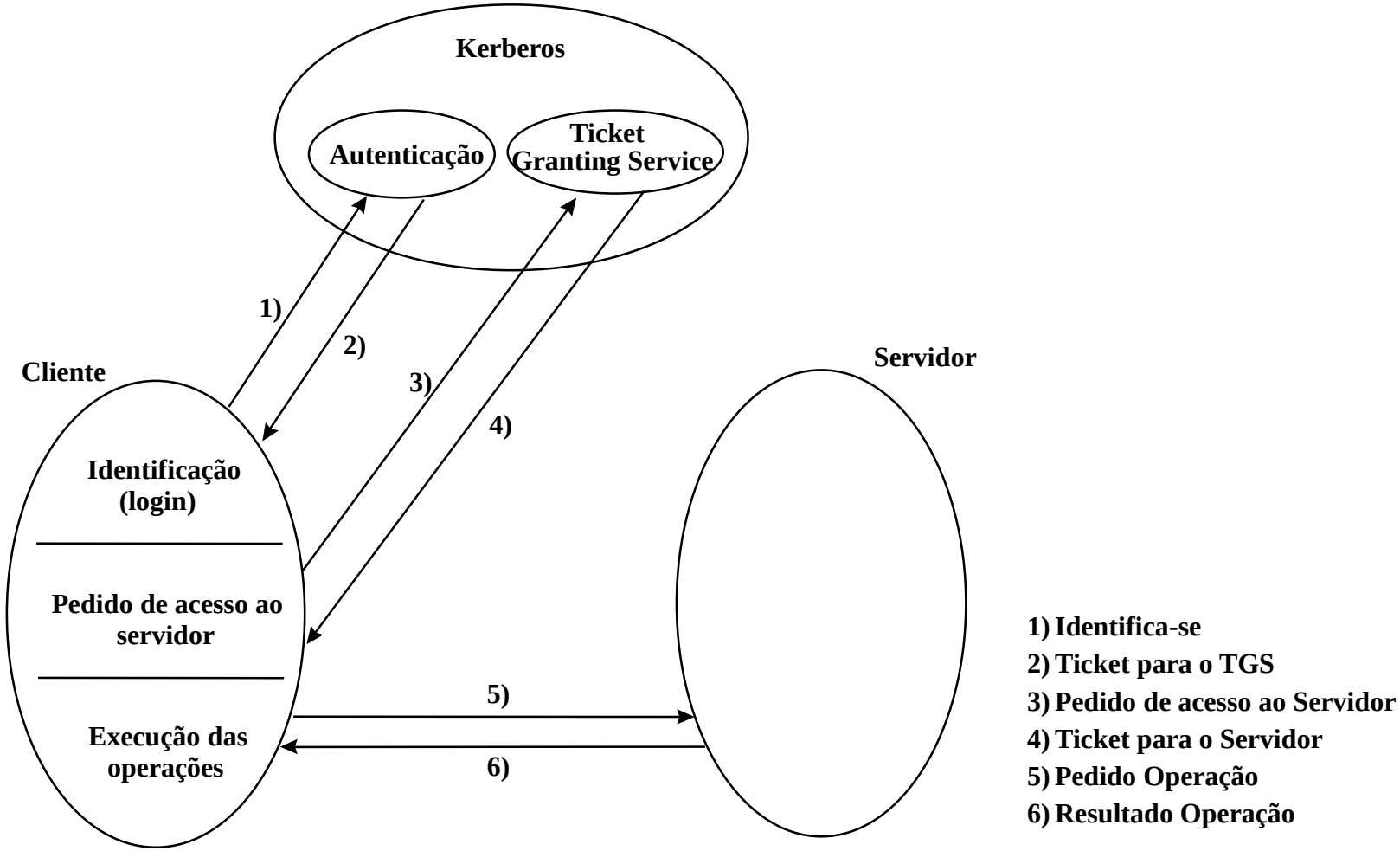
Protocolo de Needham-Schroeder – criptografia simétrica



	C	S	S_{aut}
C			
S	K_{cs}		
S_{aut}	K_c	K_s	

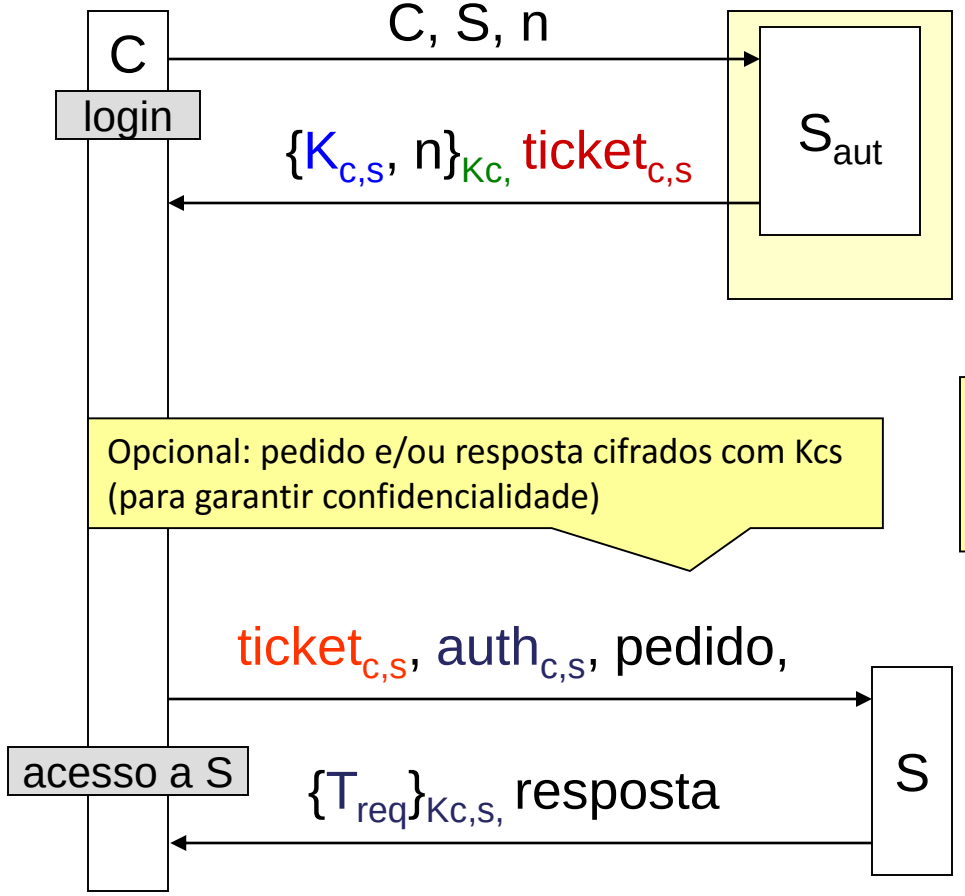
Pode ser alvo de *Replay Attack* se atacante descobrir K_{cs} e enviar esta mensagem para S

Autenticação: Kerberos



1) Identifica-se
2) Ticket para o TGS
3) Pedido de acesso ao Servidor
4) Ticket para o Servidor
5) Pedido Operação
6) Resultado Operação

Autenticação : Kerberos (Simplificado)



	C	S	S_{aut}
C			
S	$K_{c,s}$		
S_{aut}	K_c	K_S	

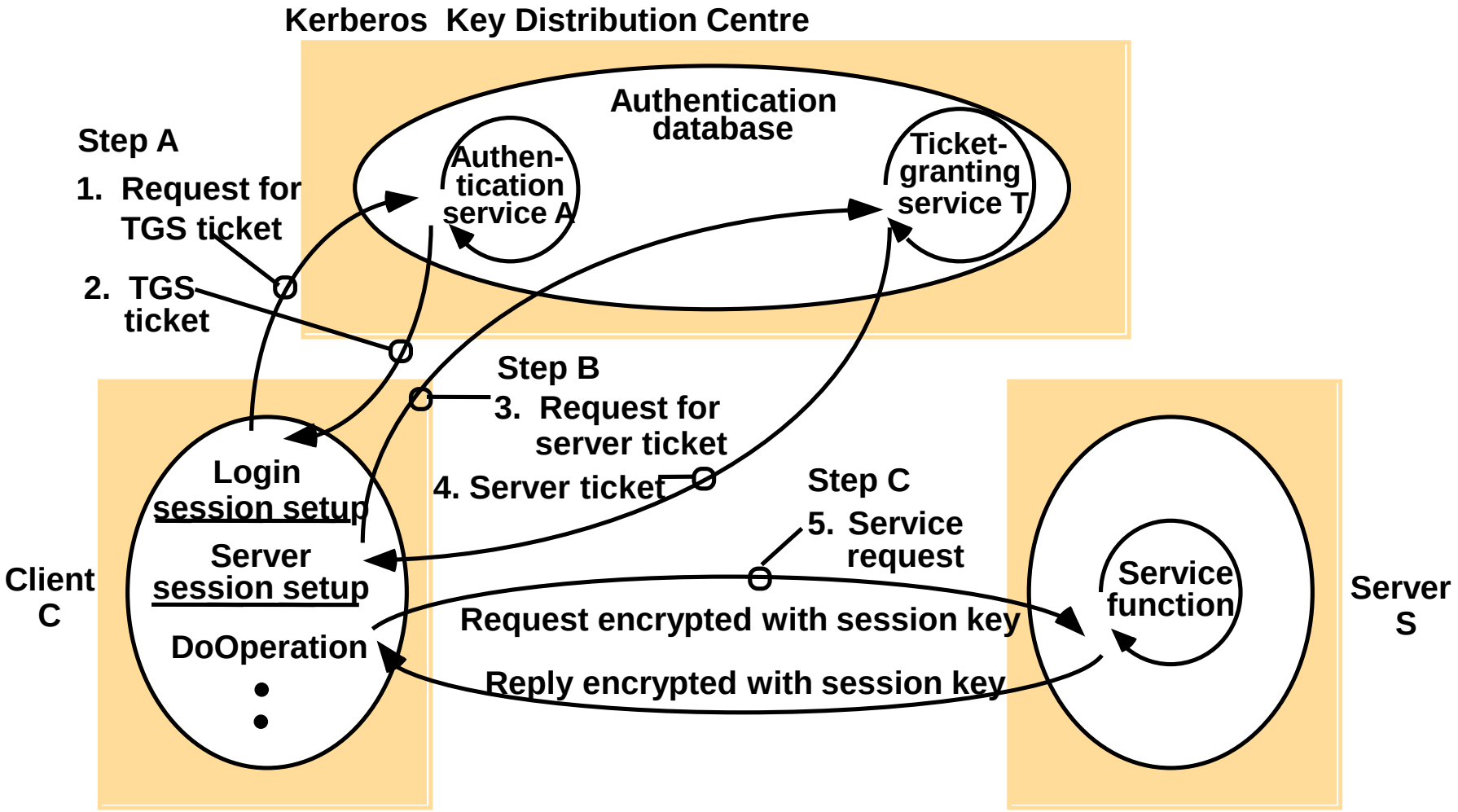
Timestamps reais: para evitar re-utilização de tickets antigos (implica relógios sincronizados)

$ticket_{x,y} = \{x, y, T_1, T_2, K_{x,y}\}_{Ky}$

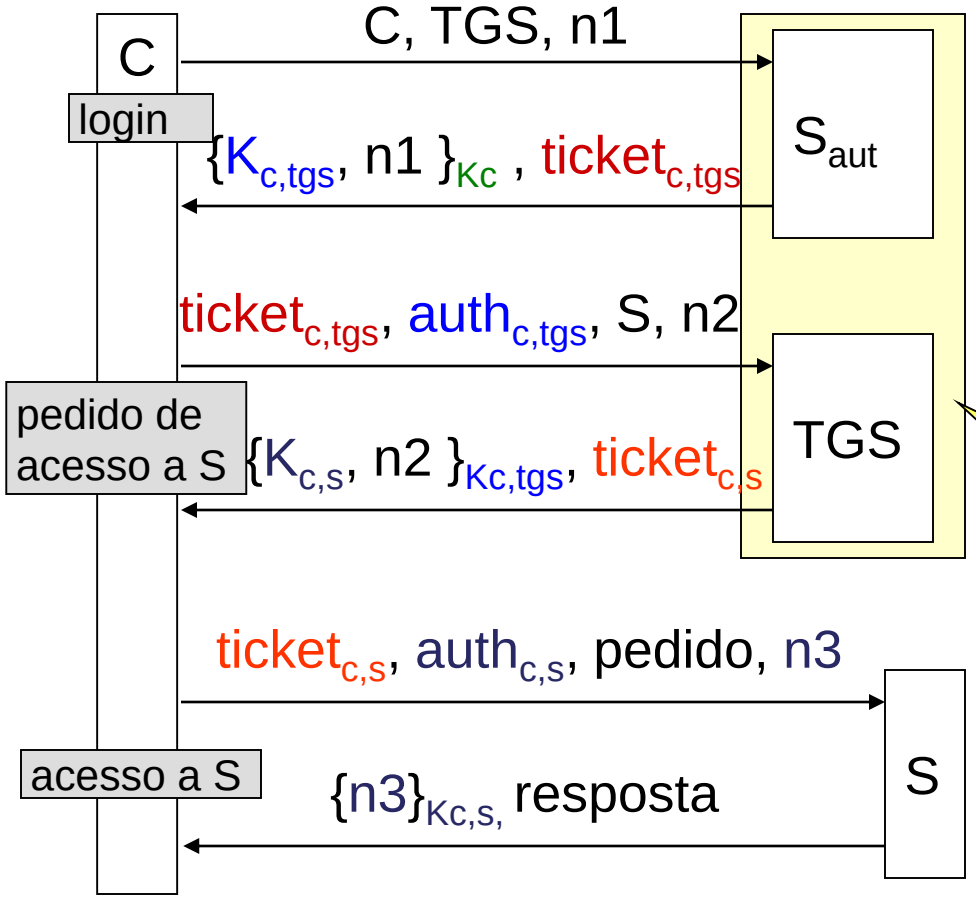
$auth_{x,y} = \{x, T_{req}\}_{Kx,y}$

Autenticador: para evitar re-envio de pedidos antigos (implica relógios sincronizados)

Arquitectura Kerberos (completo)



Autenticação : Kerberos (V5)



	C	S	TGS	S_{aut}
C				
S	$K_{c,s}$			
TGS	$K_{c,tgs}$	K_s		
S_{aut}	K_c		K_{tgs}	

Porquê a separação Saut/TGS?

Kerberos

- Escalabilidade
  - Subdivisão em *realms*
  - Cada *realm* possui um Saut e um TGS
  - Um *realm* pode aceitar autenticações feitas por outro
- Exploração
  - Segurança física dos servidores e das respetivas bases de dados
    - Saut e TGS
  - Relógios sincronizados
    - Para validar *tickets* e *authenticators*

Autenticação com chaves assimétricas

(chaves públicas e privadas)

Com chave assimétrica a posse da chave privada autentica o seu possuidor

C -> S: {Nc, C} Kps

- O cliente envia ao servidor uma mensagem cifrada com a chave pública do servidor (Kps) que contém o seu identificador e um carimbo
- Só o servidor, utilizando a sua chave privada, pode ver o conteúdo da mensagem

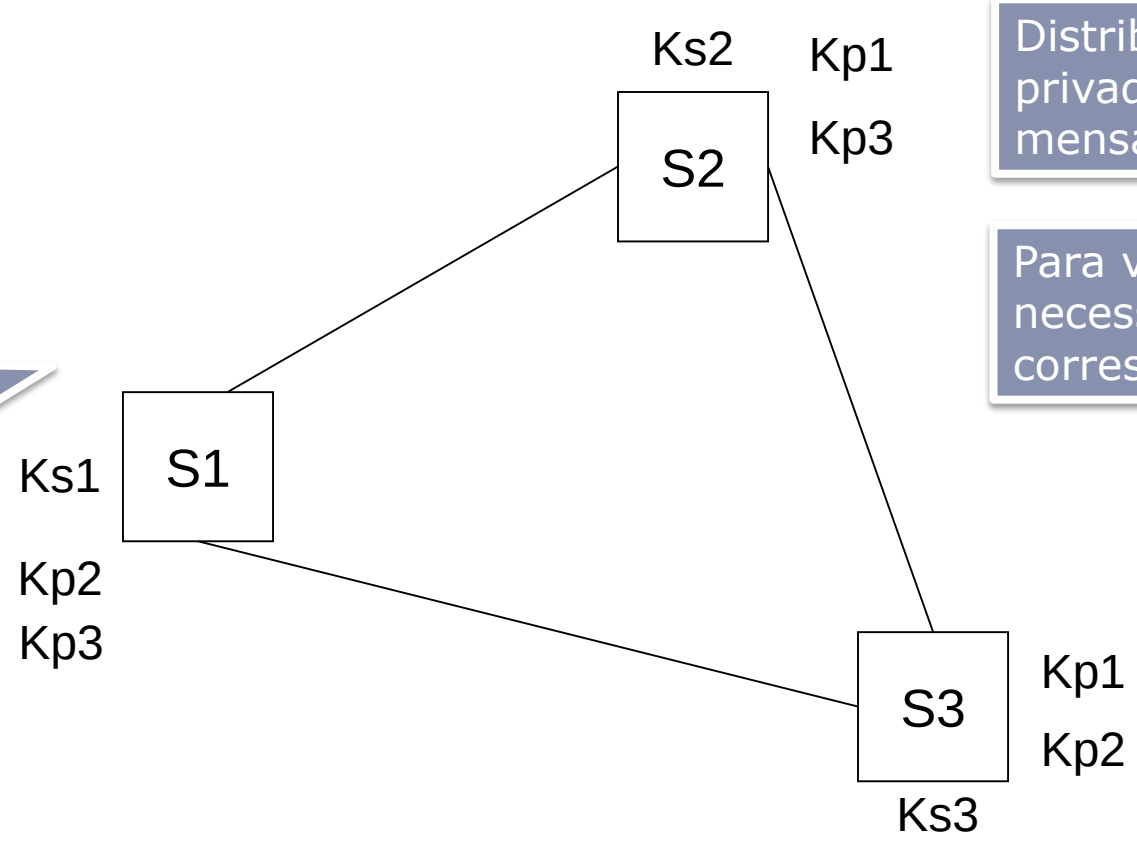
S -> C : { Nc, Ns }Kpc

C -> S: { Ns }Kps

Que ataque pode ser efetuado?

Distribuição manual de chaves assimétricas

Distribuição de Chaves e Autenticação



Chaves previamente distribuídas "fora-de-banda"

Distribuição correta de chaves privadas autentica o emissor da mensagem!

Para verificar autenticação é necessário ter a chave pública correspondente

Distribuição de chaves assimétricas com Entidade Terceira Confiada (*Trusted Third Party*)

Protocolo de Needham-Schroeder – criptografia assimétrica

1: C -> Saut: C, S

2: Saut -> C: $\{K_{ps}, S\}K_{sSaut}$

- O cliente pede ao servidor de autenticação a chave pública do servidor S
- O servidor de autenticação envia para o cliente a chave pública do servidor (K_{ps}), cifrada com a sua chave privada para garantir a autenticação da informação
- A mensagem é decifrada utilizando a chave pública do servidor de autenticação, conhecida de todos.

Protocolo de Needham-Schroeder – criptografia assimétrica

3: C -> S: {Nc, C} Kps

- O cliente envia ao servidor uma mensagem cifrada com a chave pública do servidor (Kps) que contém o seu identificador e um carimbo.
- Só o servidor, utilizando a sua chave privada, pode ver o conteúdo da mensagem

4: S -> Saut: C,S

5: Saut -> S: {Kpc, C} KsSaut

- As etapas 4 e 5 repetem o protocolo do lado do servidor.
  Este pede ao servidor de autenticação a chave pública do cliente

6: S -> C : { Nc, Ns }Kpc

7: C -> S: { Ns }Kps

Protocolo de Needham-Schroeder – criptografia assimétrica

3: C -> S: {Nc, C} Kps

- O cliente envia ao servidor uma mensagem cifrada com a chave pública do servidor (Kps) que contém o seu identificador e um carimbo.
- Só o servidor, utilizando a sua chave privada, pode ver o conteúdo da mensagem.

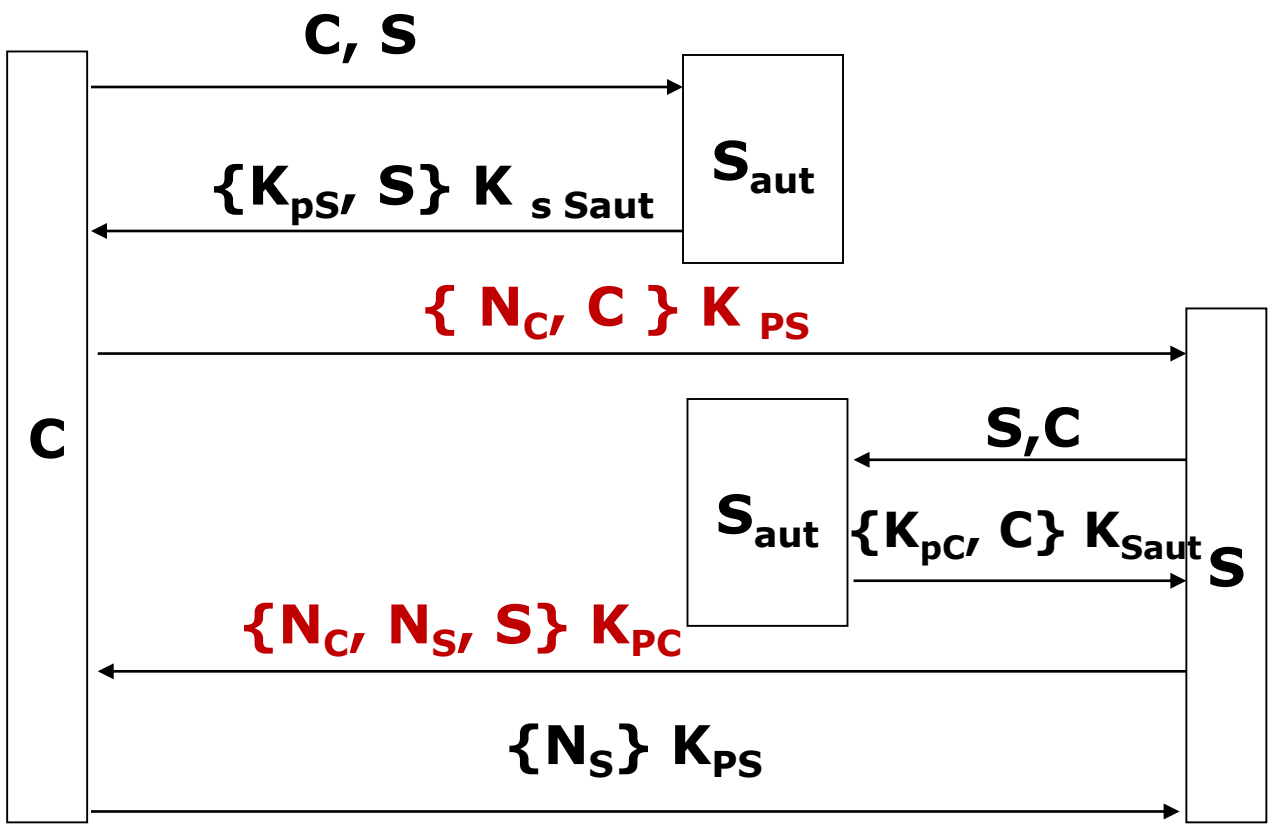
4: S -> Saut: C,S
5: Saut -> S: {Kpc, C} KsSaut

- As etapas 4 e 5 repetem o protocolo do lado do servidor.
  Este pede ao servidor de autenticação a chave pública do cliente.

6: S -> C : { Nc, Ns, **S** }Kpc
7: C -> S: { Ns }Kps

S evita o ataque *man-in-the-middle* foi sugerido como uma evolução por Lowe

Protocolo de Needham-Schroeder-Lowe criptografia assimétrica



Distribuição de chaves assimétricas com Entidade Terceira Confiada (*Trusted Third Party*) através de Certificados Digitais de Chave Pública

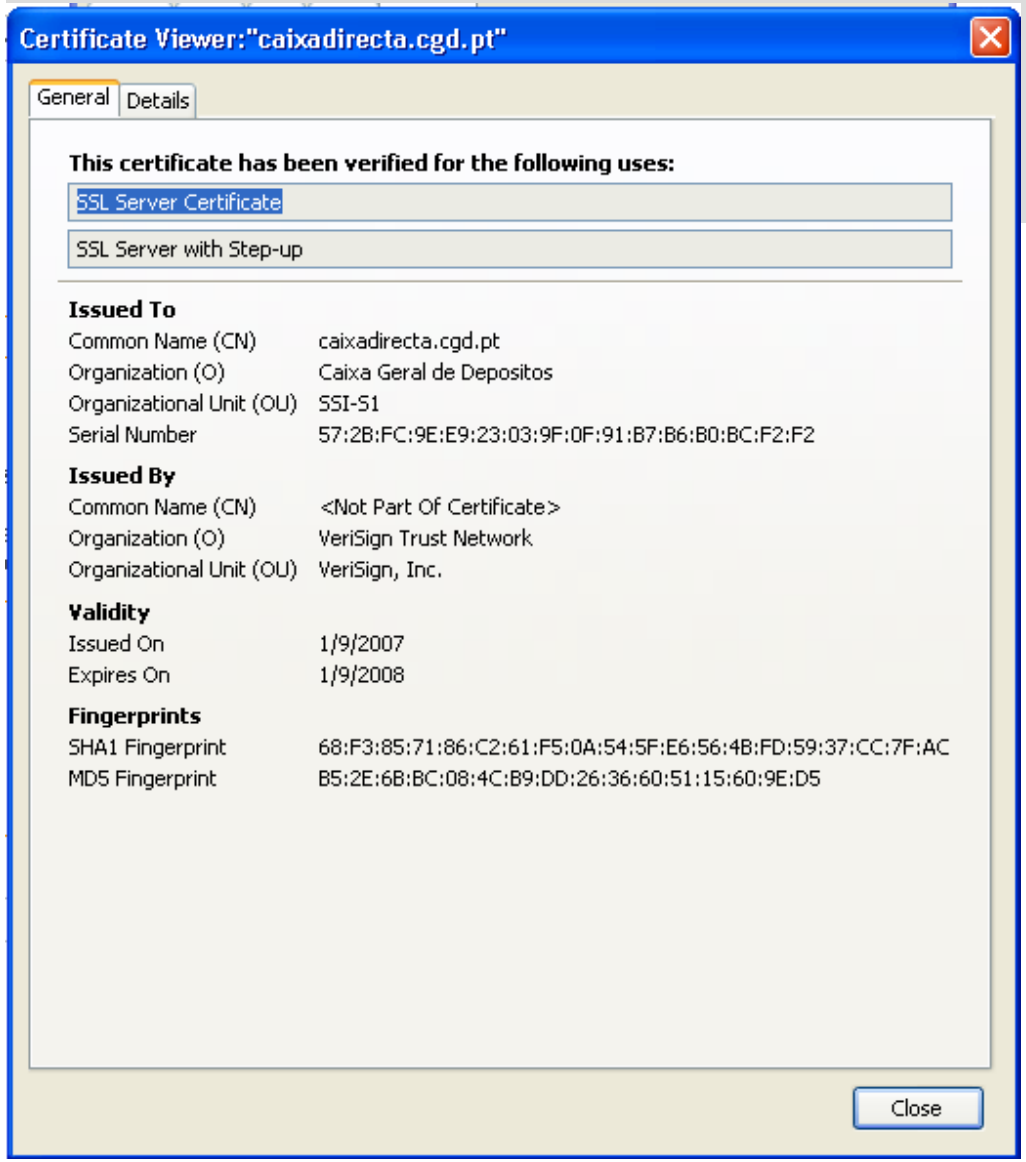
Certificados de chaves públicas

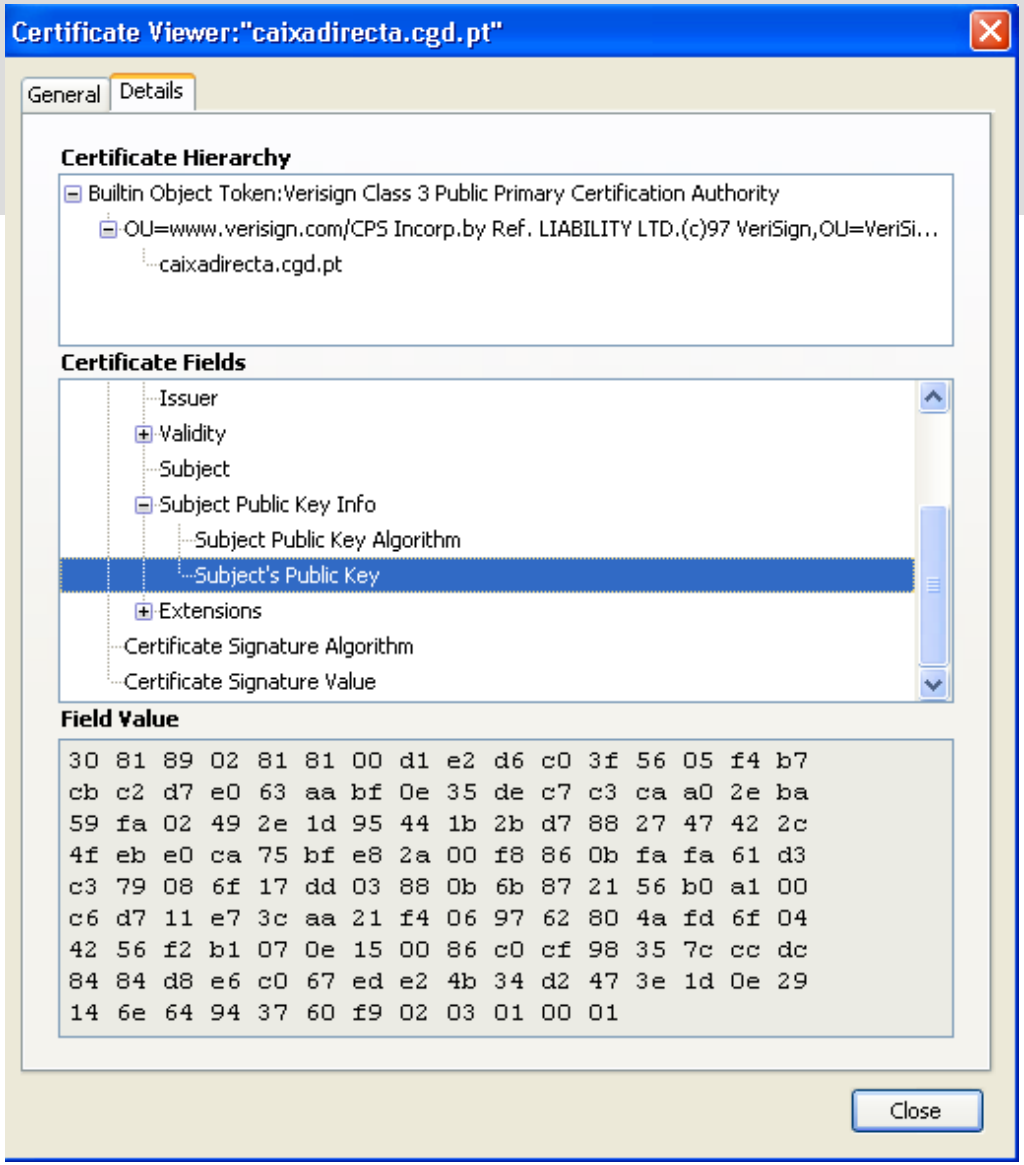
- Certificados de chaves públicas
  - Documento que associa uma chave pública a:
    - Um dono (nome, e-mail, etc)
    - Datas (de emissão, de validade)
    - Outra informação
  - Assinado por uma autoridade de certificação
    - Institucional ou não

- A **norma X.509** é a mais utilizada para formato de certificados

Formato do Certificado X509

Subject	Distinguished Name, Public Key
Issuer	Distinguished Name, Signature
Period of validity	Not Before Date, Not After Date
Administrative information	Version, Serial Number
Extended information	


Certificate Viewer:"caixadirecta.cgd.pt"
General
Details
This certificate has been verified for the following uses:
SSL Server Certificate
SSL Server with Step-up
Issued To
Common Name (CN) caixadirecta.cgd.pt
Organization (O) Caixa Geral de Depositos
Organizational Unit (OU) SSI-S1
Serial Number 57:2B:FC:9E:E9:23:03:9F:0F:91:B7:B6:B0:BC:F2:F2
Issued By
Common Name (CN) <Not Part Of Certificate>
Organization (O) VeriSign Trust Network
Organizational Unit (OU) VeriSign, Inc.
Validity
Issued On 1/9/2007
Expires On 1/9/2008
Fingerprints
SHA1 Fingerprint 68:F3:85:71:86:C2:61:F5:0A:54:5F:E6:56:4B:FD:59:37:CC:7F:AC
MD5 Fingerprint B5:2E:6B:BC:08:4C:B9:DD:26:36:60:51:15:60:9E:D5
Close


Certificate Viewer:"caixadirecta.cgd.pt"
General
Details
Certificate Hierarchy
Builtin Object Token:Verisign Class 3 Public Primary Certification Authority
OU=www.verisign.com/CPS Incorp.by Ref. LIABILITY LTD.(c)97 VeriSign,OU=VeriSi...
caixadirecta.cgd.pt
Certificate Fields
Issuer
Validity
Subject
Subject Public Key Info
Subject Public Key Algorithm
Subject's Public Key
Extensions
Certificate Signature Algorithm
Certificate Signature Value
Field Value
30 81 89 02 81 81 00 d1 e2 d6 c0 3f 56 05 f4 b7
cb c2 d7 e0 63 aa bf 0e 35 de c7 c3 ca a0 2e ba
59 fa 02 49 2e 1d 95 44 1b 2b d7 88 27 47 42 2c
4f eb e0 ca 75 bf e8 2a 00 f8 86 0b fa fa 61 d3
c3 79 08 6f 17 dd 03 88 0b 6b 87 21 56 b0 a1 00
c6 d7 11 e7 3c aa 21 f4 06 97 62 80 4a fd 6f 04
42 56 f2 b1 07 0e 15 00 86 c0 cf 98 35 7c cc dc
84 84 d8 e6 c0 67 ed e2 4b 34 d2 47 3e 1d 0e 29
14 6e 64 94 37 60 f9 02 03 01 00 01
Close

Public Key Infrastructure (PKI)

- Infraestrutura de apoio ao sistema de chaves públicas
  - Criação segura de pares de chaves assimétricas
  - Criação e distribuição de certificados de chaves públicas
  - Definição e uso das cadeias de certificação
  - Atualização, publicação e consulta da lista de certificados revogados
  - Revogação de certificados: qual o compromisso?

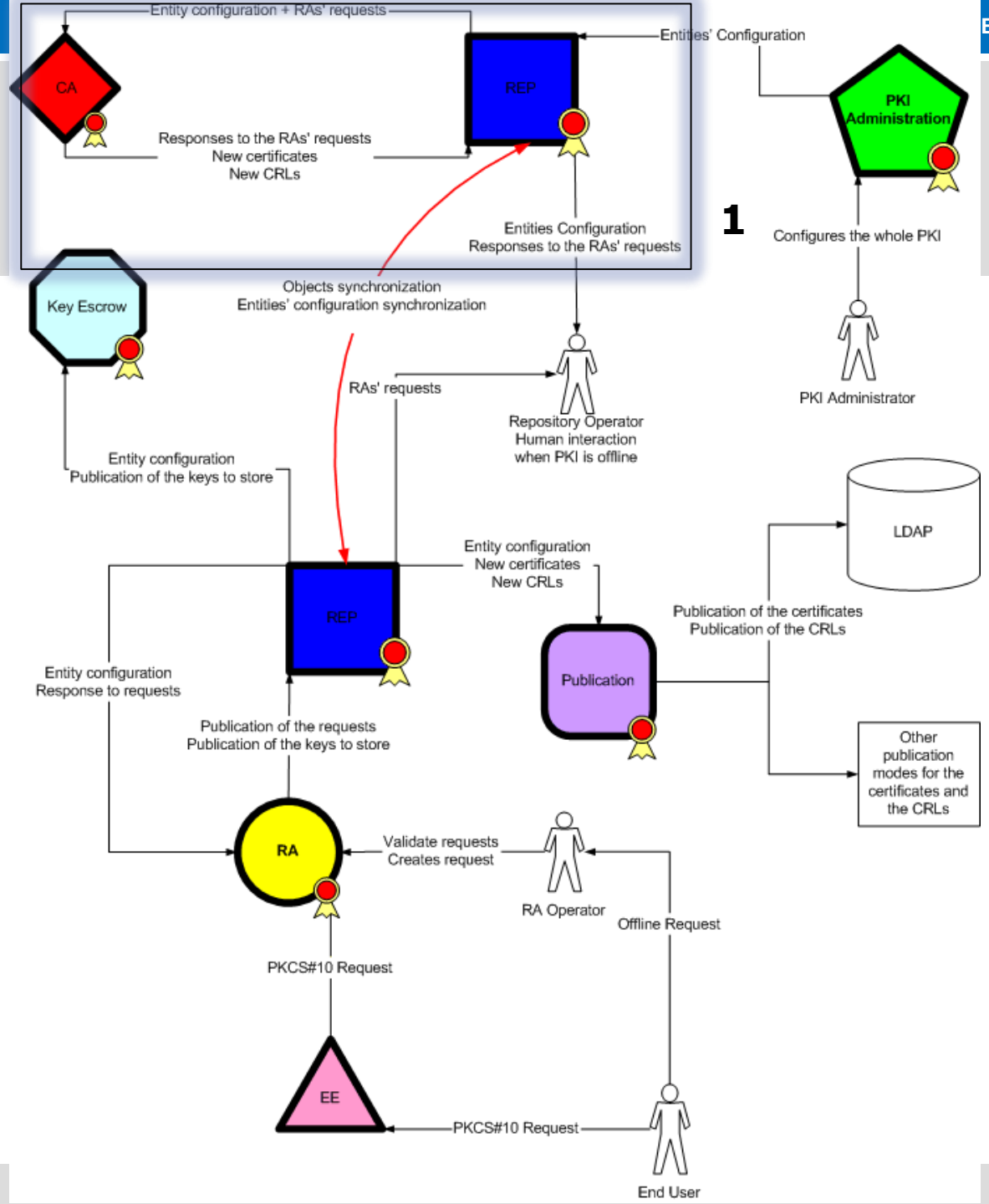
Certificados e Assinaturas Digitais

- Validação de assinaturas digitais
- Sensível à correção das chaves públicas
  - Têm de ser as obtidas de forma segura
    - Certificados
  - Têm que estar ainda em uso e não terem sido revogadas
    - *Black list*

Autoridades de certificação: Sistemas ad-hoc ou hierárquicos

- Certificação ad-hoc
  - Cada utilizador escolhe em quem confia como autoridade de certificação (ex. PGP)
- Certificação hierárquica
  - Existe uma hierarquia de certificação (institucional)
    - Árvore de Certification Authorities (CAs)
  - Cada CA emite certificados assinados com a sua chave privada
    - Que é distribuída em certificados assinados pela CA acima na hierarquia
    - A chave pública da raiz é bem conhecida (configurada manualmente, e.g., os browsers reconhecem a VeriSign)
  - Funções de uma CA
    - Emissão e distribuição de certificados
    - Gestão e distribuição de listas de certificados revogados

Funcionamento PKI



1 – funcionamento *offline*

Canais de comunicação seguros

Exemplos

Canais de comunicação seguros: propriedades

- Confidencialidade
  - Dos dados
    - Cifra dos dados enviados
  - Dos fluxos de informação
- Integridade
  - Das mensagens
    - Adição de valores de controlo não forjáveis
  - Dos fluxos de mensagens
    - Contextos de cifra e/ou controlo
- Autenticidade
  - Dos interlocutores
    - Cifra de valores pré-combinados e frescos
      - Com uma chave secreta partilhada por emissor e recetor
      - Com a chave privada do emissor
- Não Repudiação

Argumento “extremo-a-extremo” (*End-to-end principle*)

- As funcionalidades dos protocolos de comunicação devem ser implementadas pelos extremos do canal de comunicação (sempre que possível), pois…
  - Ao implementar nos níveis mais baixos,
    obrigam todos os canais a pagar o seu custo, mesmo que não queiram
  - Evitam redundâncias, quando as funcionalidades têm de ser repetidas extremo-a-extremo
- Princípio de desenho do IP (*Internet Protocol*)

Nível de Protocolo

- Nível de protocolo onde realizar o canal seguro
  - Ligação de dados
    - Podia ser eficientemente implementado no *hardware* do controlador de rede.
    - Não evita o ataque aos comutadores
  - Rede
    - ex.: IPsec – para *Virtual Private Networks*
    - Não vai até ao nível do transporte
  - Interfaces de Transporte
    - *Sockets* - ex.: SSL
  - Aplicação :
    - ex.: HTTPS, SSH, PGP, PEM, SET, WS-Security para os Web Services

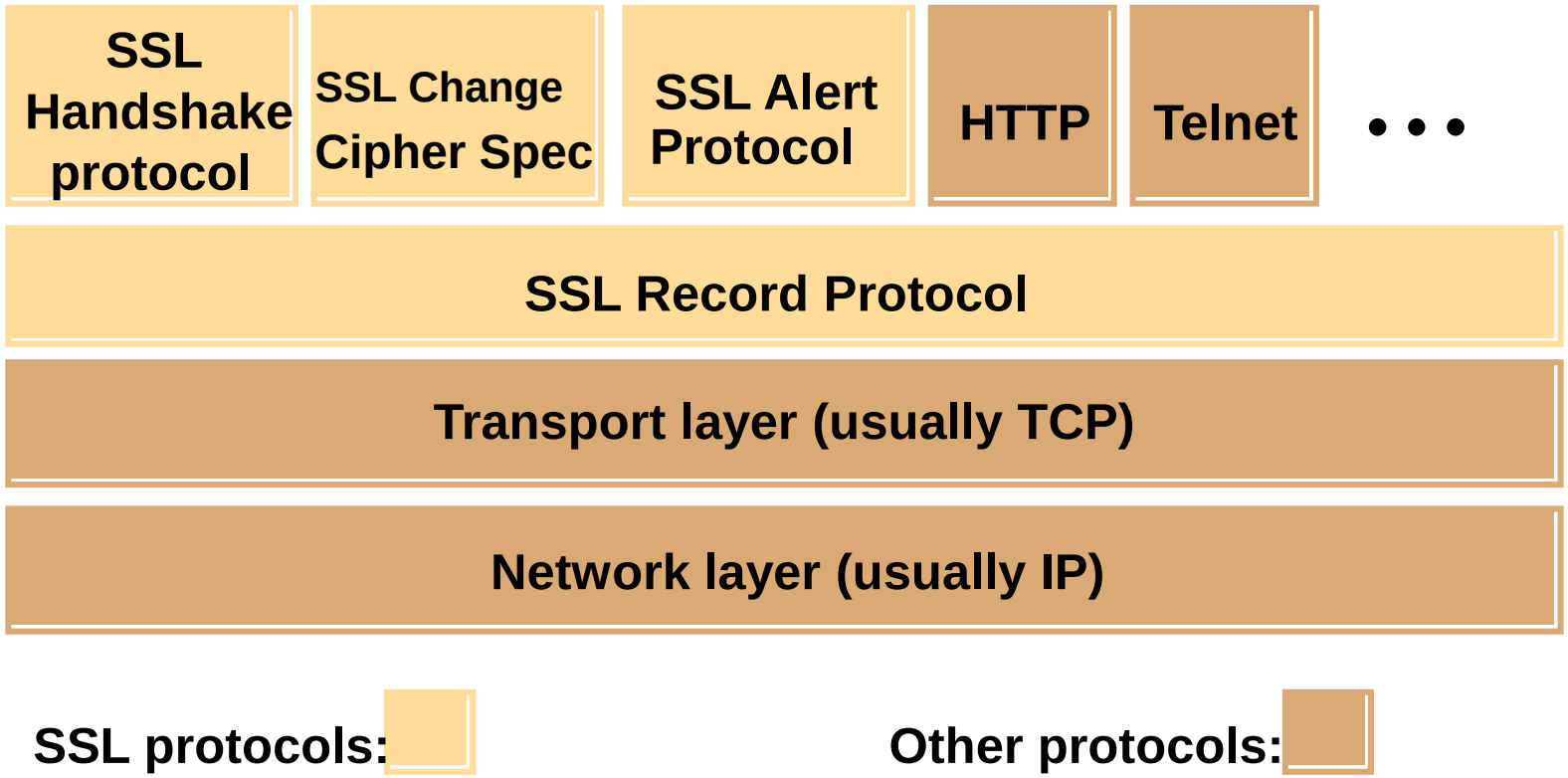
Canais de comunicação seguros: algumas soluções

- Nível Aplicação
  - E-mail: PGP, PEM
  - Aquisições electrónicas: SET
  - WS – Security nos Web Service
- Nível aplicação-transporte
  - HTTPS
  - SSH
  - SSL / PCT / TLS
- Nível rede
  - IPSEC (SKIP, Photuris, SKEME, ISAKMP, etc.)
    - Para IPv4 e IPv6

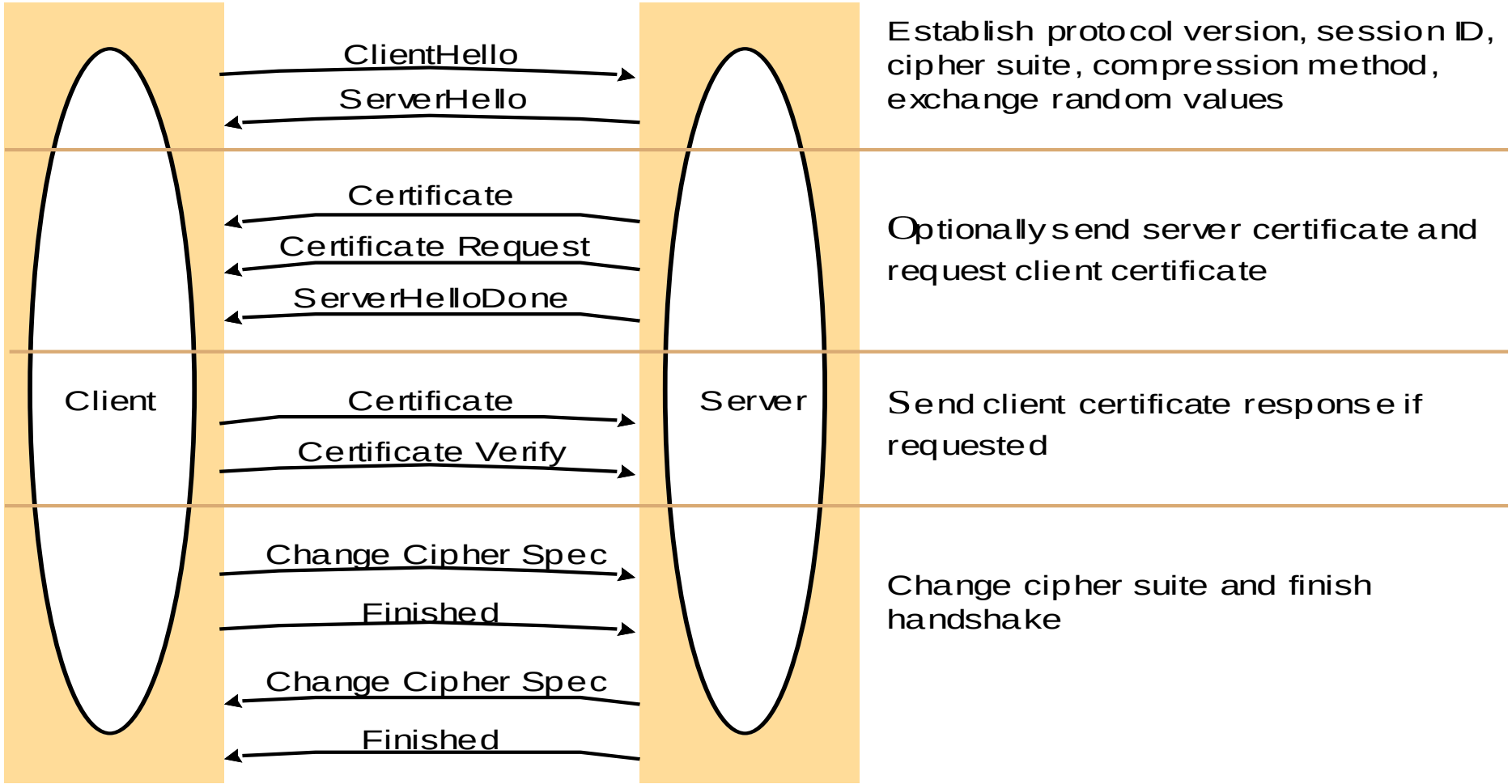
Caso de estudo: TLS/SSL

Base do HTTPS

Pilha de protocolos SSL



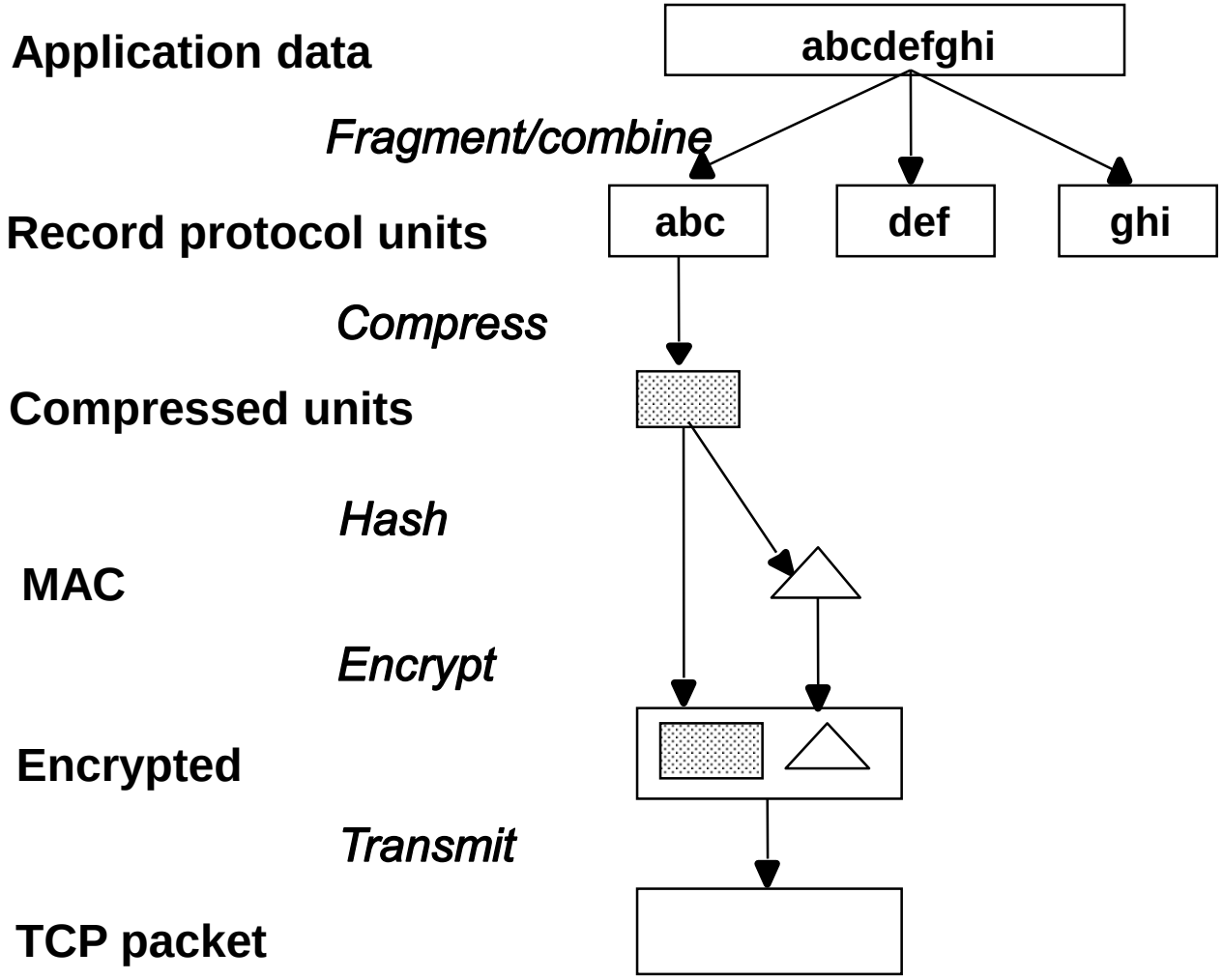
TLS handshake protocol



TLS handshake: opções

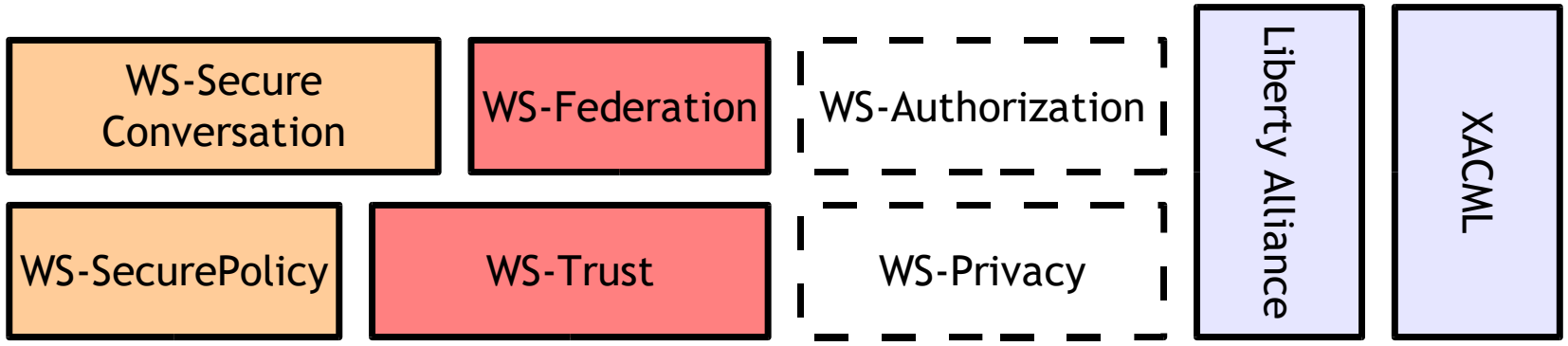
Component	*Description*	*Example*
Key exchange method	the method to be used for exchange of a session key	RSA with public-key certificates
Cipher for data transfer	the block or stream cipher to be used for data	IDEA
Message digest function	for creating message authentication codes (MACs)	SHA

TLS record protocol



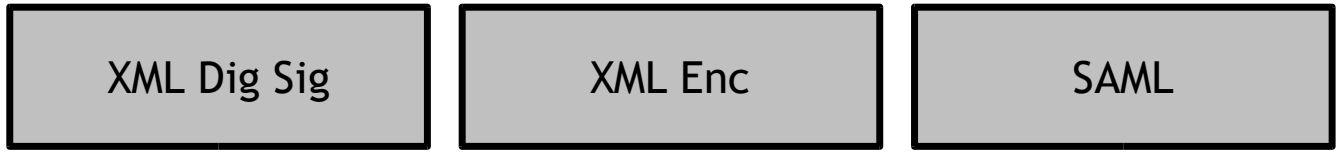
Segurança Web Services

Diagrama de Segurança dos Web Services



- Blocos da mensagem

- Blocos básicos



WS-Security Architecture