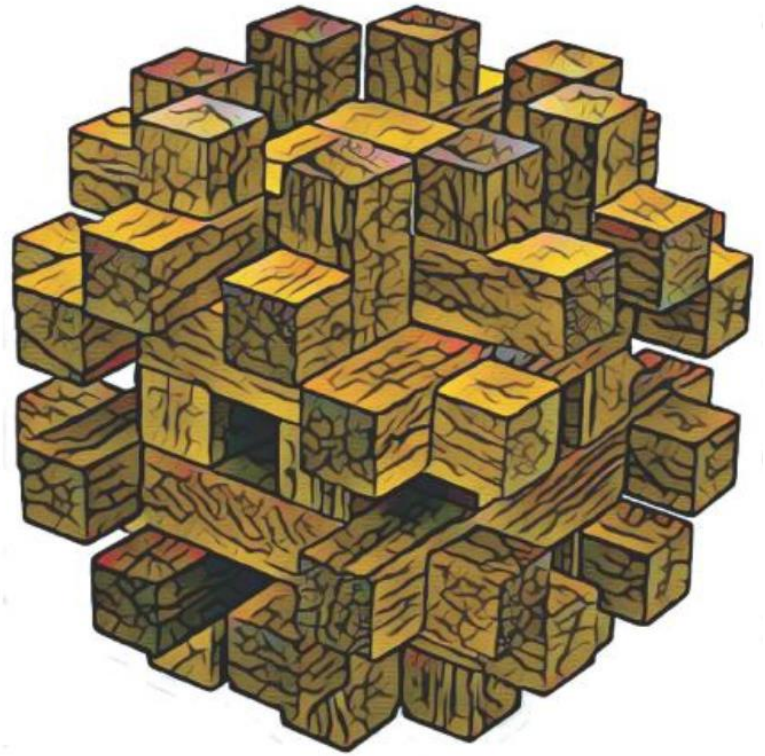
Distributed Systems

Maarten Van Steen & Andrew S. Tanenbaum



3th Edition – Version 3.03 - 2020

Capítulo 9

Segurança

15 de Setembro de 2022

SD - SEGURANÇA INTRODUÇÃO

REQUISITOS DE SEGURANÇA EM SD

- Compartilhamento seguro de recursos
- Interação autorizada entre processos
- Proteção contra acessos não autorizados
- Comunicação segura entre processos
- Inimigo capaz de:
  - Enviar mensagens para processos
  - Ler e copiar mensagens entre pares de processos
  - Realizar acessos autorizados / não autorizados (através de conta roubada)
  - Ameaças:
    - Interceptação (*sniffing / dumping*)
    - Interrupção (disrupção / negação de serviço)
    - Modificação (*tampering* / mudança de web site (*defacing*))
    - Fabricação (injeção / ataques replay)

SD - SEGURANÇA INTRODUÇÃO

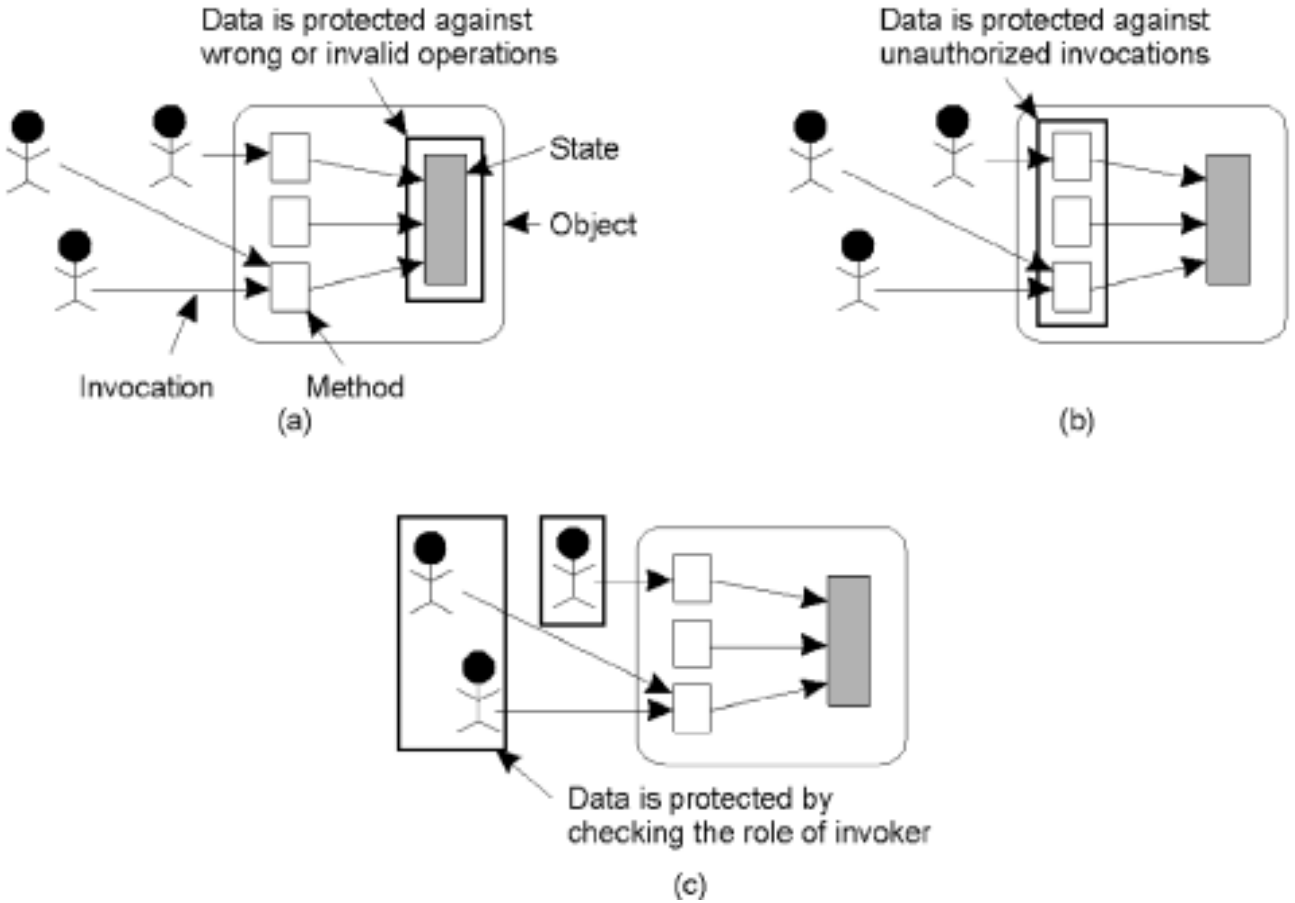
REQUISITOS DE SEGURANÇA EM SD

- Segurança para garantir que o sistema distribuído provê os seguintes requisitos:
  - Disponibilidade
  - Confiabilidade
- O sistema deve rodar por longos períodos
  - Segurança
- Manutenibilidade
- Confidencialidade
- Integridade
- Alterações somente de forma autorizada (interação correta, acesso autorizado)
- Políticas de segurança implementadas por:
  - Mecanismos de segurança
    - Encriptação (confidencialidade, verificação de integridade)
    - Autenticação (identificação das partes)
    - Autorização (controle de acesso a informação)
    - Auditoria

SD - SEGURANÇA INTRODUÇÃO

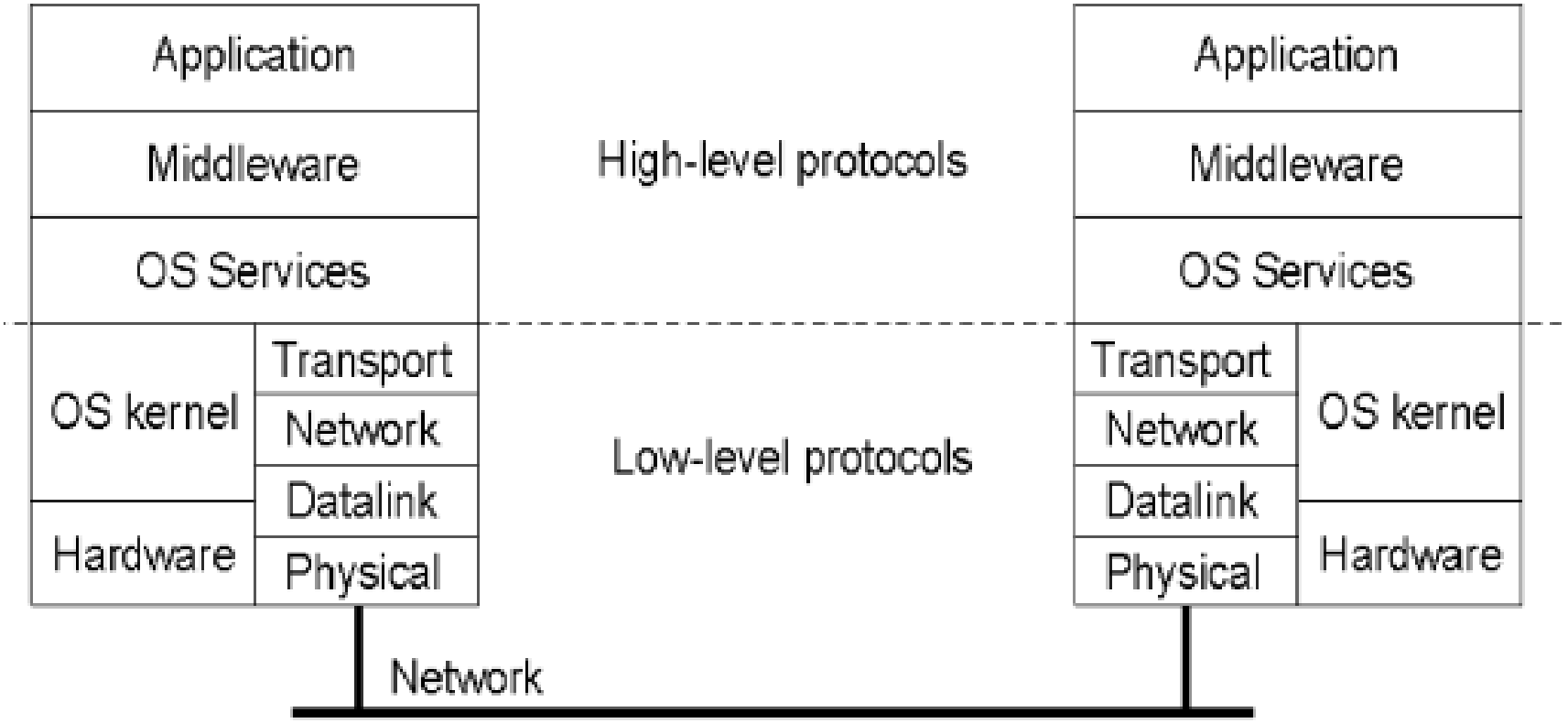
QUESTÕES DE PROJETO DE SEGURANÇA EM SD

- Proteção de dados, independente das diversas operações que podem ser feitas em itens de dados
- Foco em operações que podem ser invocadas em dados
- Foco em usuários: somente especifica pessoas tendo acesso a aplicação (papel do invocador)
- Foco de controle



QUESTÕES DE SEGURANÇA

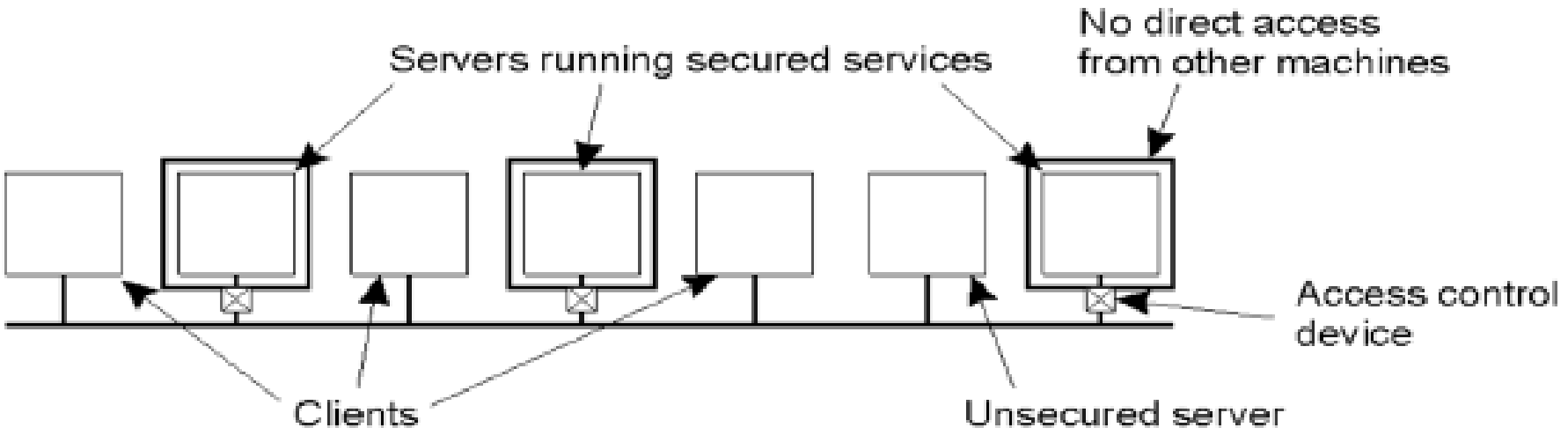
CAMADAS DE MECANISMOS DE SEGURANÇA



QUESTÕES DE SEGURANÇA

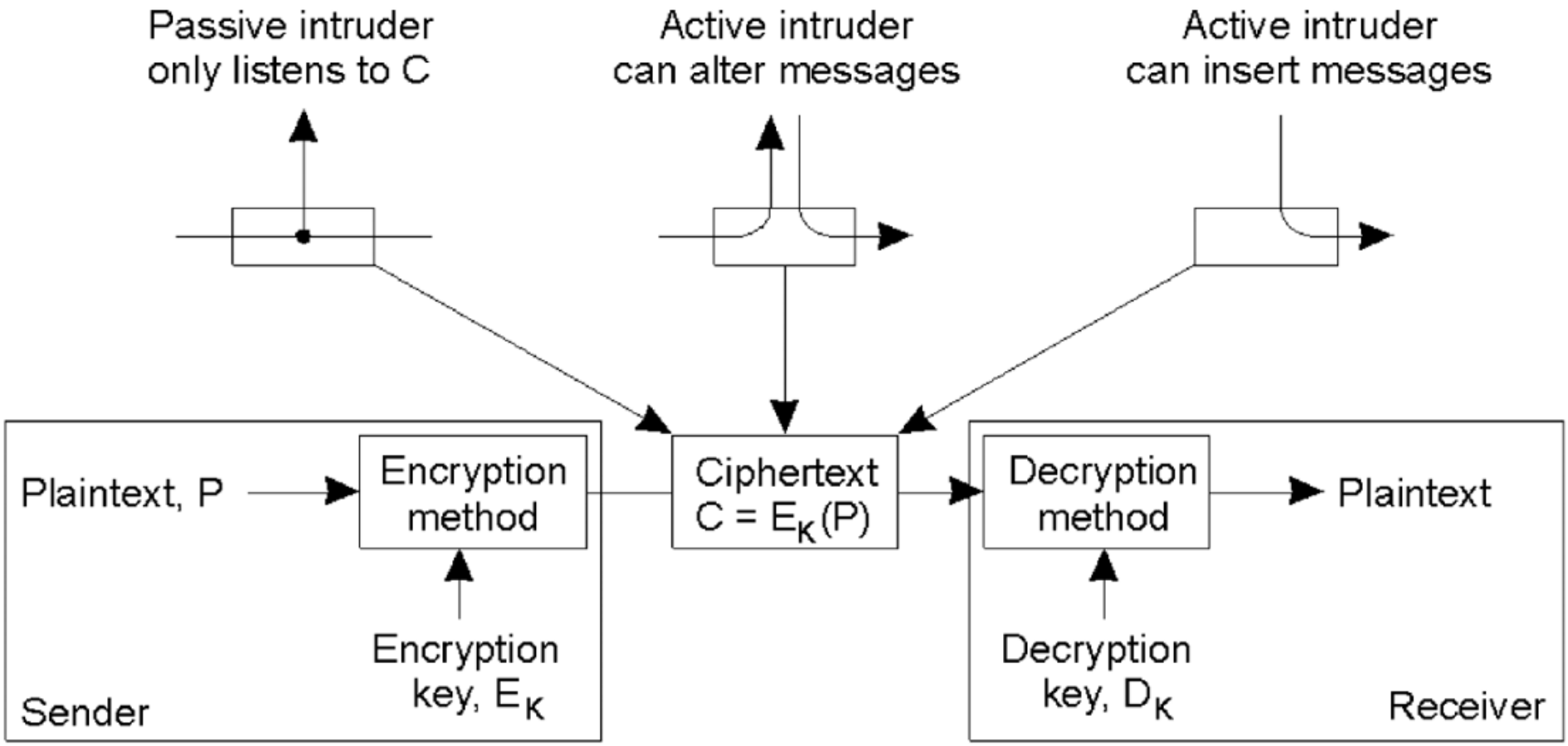
CAMADAS DE MECANISMOS DE SEGURANÇA

- Onde colocar mecanismos de segurança ?
- Se não podemos confiar nas camadas mais baixa (IPSec) podemos colocar nas mais altas (HTTPS) Ou na camada de transporte (SSL)
- TCB – Trusted Computing Base
  - Simplicidade (nem sempre é desejável)
  - SSO - single sign on (server)



CRIPTOGRAFIA

ENCRIPTAÇÃO



CRIPTOGRAFIA

ENCRIPTAÇÃO SIMÉTRICA

- Se $K_E = K_D$
- Usamos o símbolo $K_{A,B}$ para denotar uma chave compartilhada por A e B
- Para alcançar comunicação entre N participantes, precisamos de $O(N^2)$ chaves (keys) diferentes

CRIPTOGRAFIA

ENCRIPTAÇÃO ASSIMÉTRICA

- Se $K_E \neq K_D$
- K_E e K_D são unicamente amarrados um ao outro (formando um par)
  - K_D pode desencriptar somente de K_E
  - K_E pode encriptar somente para K_D
  - Computando K_E de K_D ou vice-versa deve ser computacionalmente inviável
  - Podemos distribuir uma chave sem perigo para outras
- Podemos distribuir de forma segura K_E para todos interessados em enviar mensagens para mim (mantendo K_D privado)

CRIPTOGRAFIA

ENCRIPTAÇÃO ASSIMÉTRICA

- Podemos também reverter as coisas: distribuir K_D e manter K_E privado
  - É assim que e-signing é feito

- Podemos chamar as duas chaves
  - K^+_A (chave pública de A)
  - K^-_A (chave privada de A)

CRIPTOGRAFIA

HASH

- Messagem m – funçaõ hash → Hash h
  - h = H(m)
- Função H co-domínio é menor que o domínio
  - H não é injetiva
  - Não temos uma função hash diferente para cada mensagem
- Colisões: m≠m’ mas H(m)=H(m’)
  - MD5 usa 128 bit (16 byte)
  - Mensagens podem ser de qualquer comprimento
  - Para 17 bytes temos 256 colisões, para 1Mbyte temos 250 Milhões de colisões, …

CRIPTOGRAFIA

HASH

- As propriedades desejadas de h dependem do uso pretendido para o hash (error codes, hash tables, …)
  - sem inversa
- Dado h tal que h=H(m) …
- … deve ser difícil achar m
  - resistência fraca a colisão
- Dado h e m tal que h=H(m) …

  … deve ser difícil de achar m'≠m tal que h=H(m') – resistência fraca a colisão
- Dado H() …… deve ser difícil de achar m'≠m tal que H(m)=H(m')

CRIPTOGRAFIA DE CHAVE PÚBLICA

RSA

- Baseado na dificuldade de fatoração de números grandes
- Geração de chaves
  - Escolha dois primos grandes, p e q, Compute n = p x q e z = (p – 1) x (q – 1)
  - Escolha um número d que é um primo relative a z
  - Compute o número e tal que e x d = 1 mod z (fácil)
- $K^-_B=(d,n)$ and $K^+_B=(e,n)$
- Alice encripta m: $c = m^e \pmod n$ e transmite c
- Bob desencripta c: $m = c^d \pmod n$ – pode ser provado que $m=m^{de} \pmod n$
- Algoritmos de chave simétricas são mais rápidos
  - RSA 100-1000 vezes mais lento que DES

CRIPTOGRAFIA DE CHAVE PÚBLICA

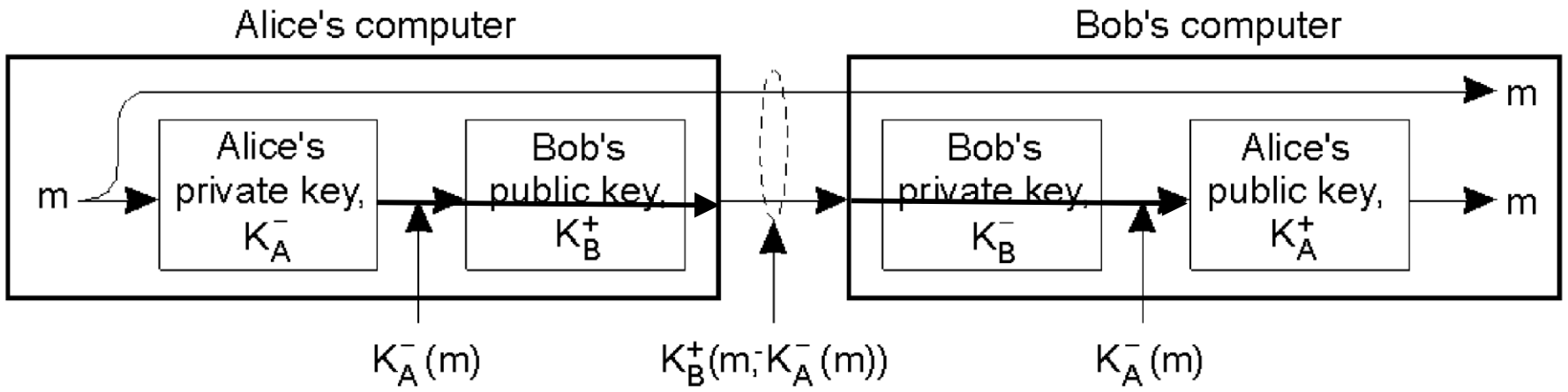
RSA - EXEMPLO

- Exemplo simples com números primos
  - p = 11
  - q = 5
  - n = p x q = 55
  - z = (p-1) x (q-1) = 40
  - Escolha um número d que é relativo ao primo z: d = 13
  - Compute o número e tal que (e x d) = 1 mod z: e = 37

- Alice quer encriptar uma mensagem m = 16 e enviá-la a Bob
  - Parte do segredo de Alice é (e, n) = (37, 55)
  - Alice encripta m : $c = m^e \pmod n = 36$
  - Alice envia c = 36 para Bob
  - Parte do segredo de Bob é (d, n) = (13, 55)
  - Bob desencripta m : $m = c^d \pmod n = 16$

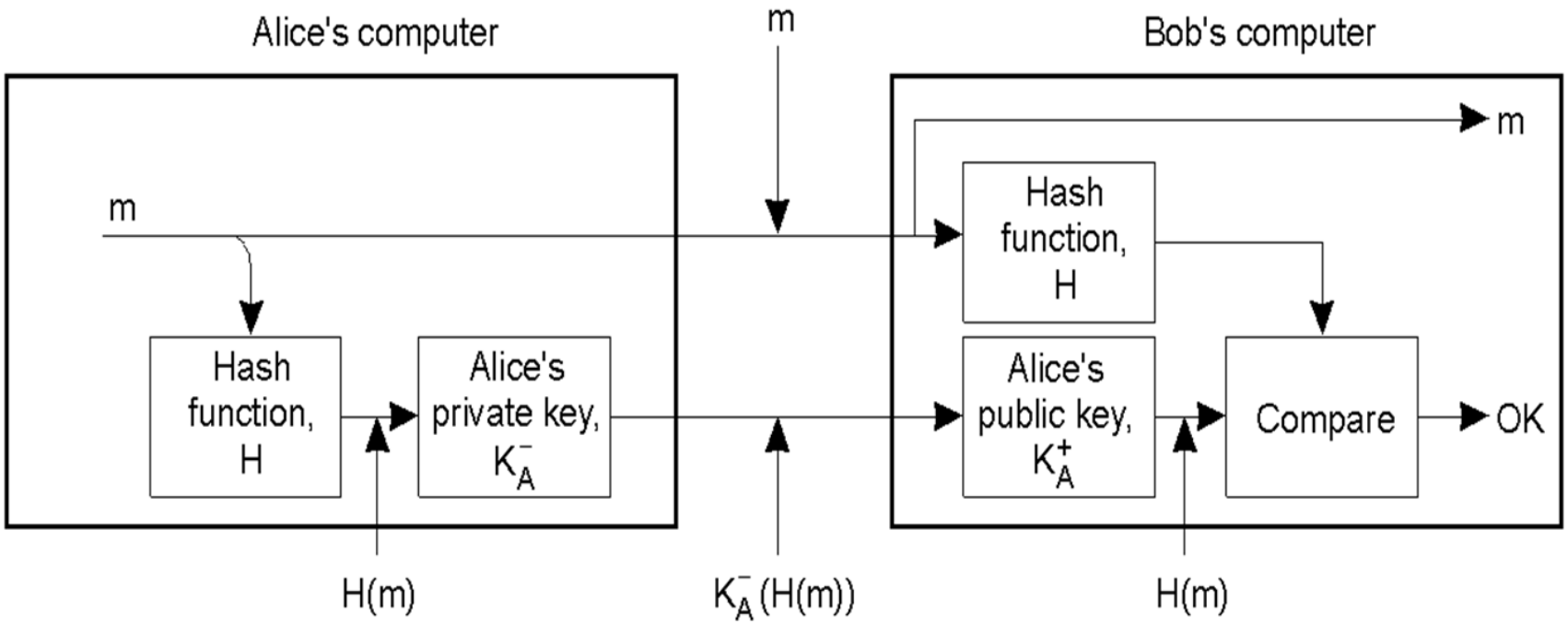
ASSINATURAS DIGITAIS

ASSINANDO - para garantir integridade da mensagem



ASSINATURAS DIGITAIS (MESSAGE DIGEST)

ASSINANDO DIGITALMENTE USANDO UMA "MESSAGE DIGEST" – garantia de origem



AUTORIDADES CERTIFICADORAS

COMO PODEMOS CONFIAR NA ASSOCIAÇÃO CHAVE PÚBLICA – PESSOA FÍSICA ?

- Através de certificados de chave pública
- Uma tupla
- Identidade
- Chave pública
- Assinado por uma autoridade de certificação (CA)
- A chave pública da CA é considerada bem conhecida
- A ideia básica: informações difusas são difíceis de alterar
  - Se a chave pública da CA estiver em todo lugar, é realmente difícil alterar cada cópia sem ser notado No entanto, a CA precisa autenticar chaves públicas antes de emitir um certificado
  - Nós podemos ter vários modelos de confiança –
    - Hierárquico: a CA raiz pertence à autoridade central (possivelmente governamental)
    - Existe uma hierarquia de CAs que se certificam mutuamente
    - Usuários de certificado de CAs de folha

AUTORIDADES CERTIFICADORAS

COMO PODEMOS CONFIAR NA ASSOCIAÇÃO CHAVE PÚBLICA – PESSOA FÍSICA ?

- Rede de confiança do PGP – *pretty good privacy*
- Os usuários podem autenticar outros usuários assinando sua chave pública com seus próprios
- Os usuários podem definir em quem eles confiam para autenticar os outros
- Os dois são ortogonais: posso ter certeza de que o $K^{+}A$ é a chave de Alice, mas posso não confiar em sua diligência em assinar a chave de outras pessoas.
- Certificados de vida útil
- Proteger do compromisso
- Um certificado associa *public-key* e o proprietário
- Chave secreta é comprometida. E agora?
- Revogação: Lista de revogação de certificados (CRL) publicada periodicamente pela CA
- Toda vez que um certificado é verificado, o CRL atual deve ser consultado Isso significa que um cliente deve entrar em contato com a CA sempre que uma nova CRL for publicada

CANAIS SEGUROS

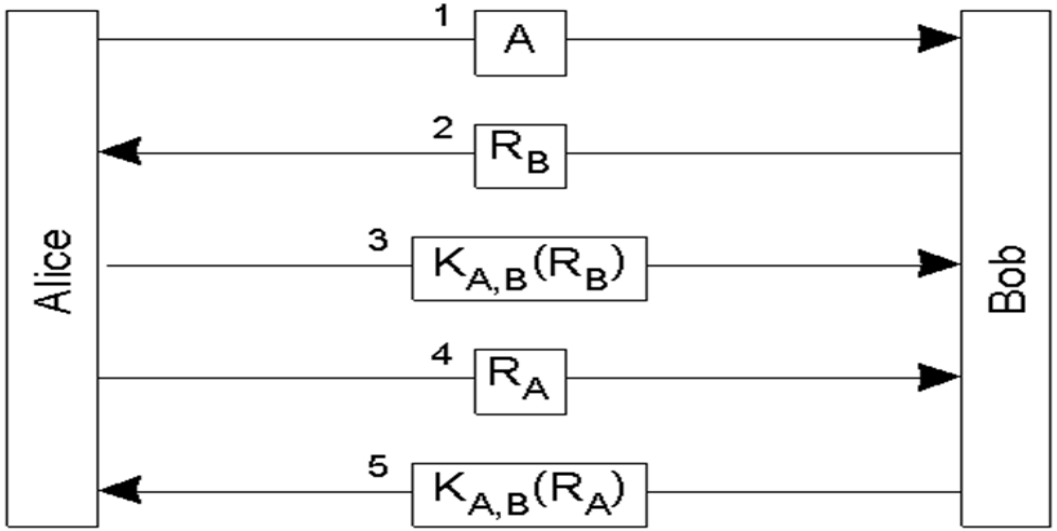
COMO PODEMOS CONFIAR NA ASSOCIAÇÃO CHAVE PÚBLICA – PESSOA FÍSICA ?

- Canal seguro fornece comunicação segura em sistemas distribuídos
- Canais seguros fornecem proteção contra Intercepção (através de criptografia)
- Modificação e Fabricação (através de autenticação e integridade de mensagens)
- Não proteja contra interrupção
    - Assumimos que os processos são seguros, enquanto todas as mensagens podem ser interceptadas, modificadas e falsificadas por um invasor
    - A autenticação e a integridade da mensagem devem estar juntas (o remetente e o conteúdo estão juntos)
    - Se uma mensagem for modificada, não será mais útil conhecer o remetente da mensagem original.
- Uma mensagem não modificada não é muito útil se eu não souber sua origem
- A autenticação precisa de informações compartilhadas entre o autenticador e a parte
- O próprio conceito de autenticação requer que (não a implementação em um protocolo específico)

CANAIS SEGUROS

COMO PODEMOS CONFIAR NA ASSOCIAÇÃO CHAVE PÚBLICA – PESSOA FÍSICA ?

- Nos seguintes protocolos, esta informação é a chave de autorização (simétrica ou assimétrica) trocada antecipadamente
- Como esta chave de autorização é trocada?
- Difícil: vamos tentar responder mais tarde
- Os protocolos de autenticação verificam essa informação comum sem divulgá-la no canal

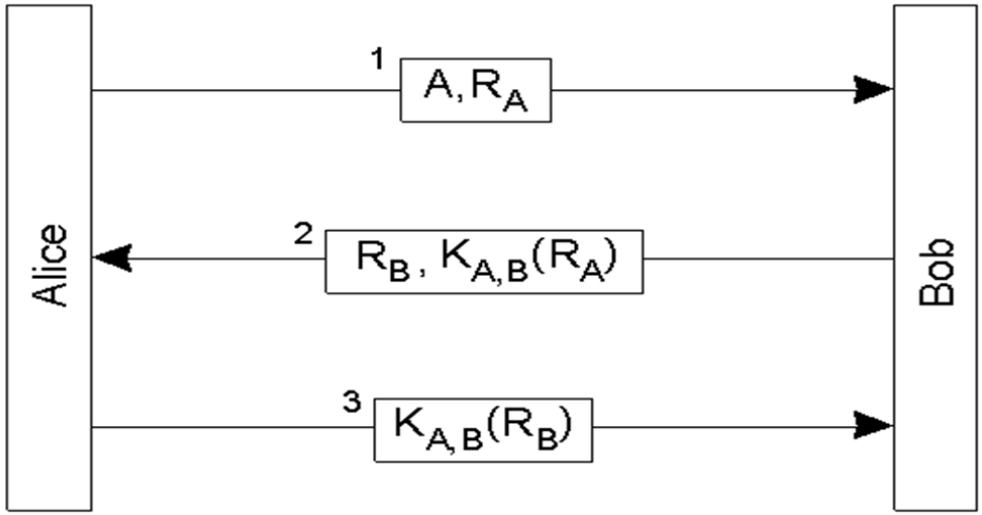


1- Alice contata Bob para estabelecer um canal de comunicação
2 – Bobo desafia Alice (ramdom number Rb)
3 – Alice encripta Rb com a chave comum Kab
4 – Alice desafia Bob com ramdom Ra
5 – Bob encripta Ra com chave comum Kab

CANAIS SEGUROS

AUTENTICAÇÃO COM CHAVE SECRETA COMPARTILHADA

- Desafio - resposta



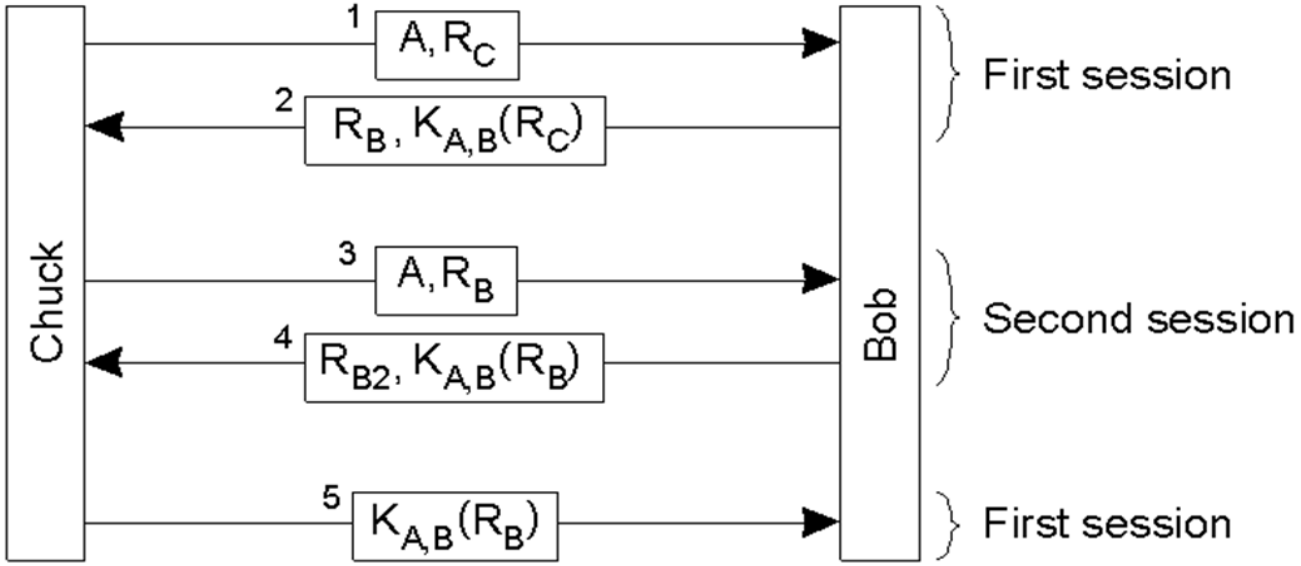
OBSERVAÇÃO

Parece ser uma troca longa .. Vamos tentar fazer piggyback em mensagens ..

CANAIS SEGUROS

OBSERVAÇÃO

- Uma vez que A vai autenticar B, vamos desafiar B na primeira rodada – não funciona !



REFLECTION ATTACK

- Um atacante pode requisitar uma resposta ao desafio
- Solução: use par para requisição e impar para resposta

CANAIS SEGUROS

DESGASTE DA CHAVE

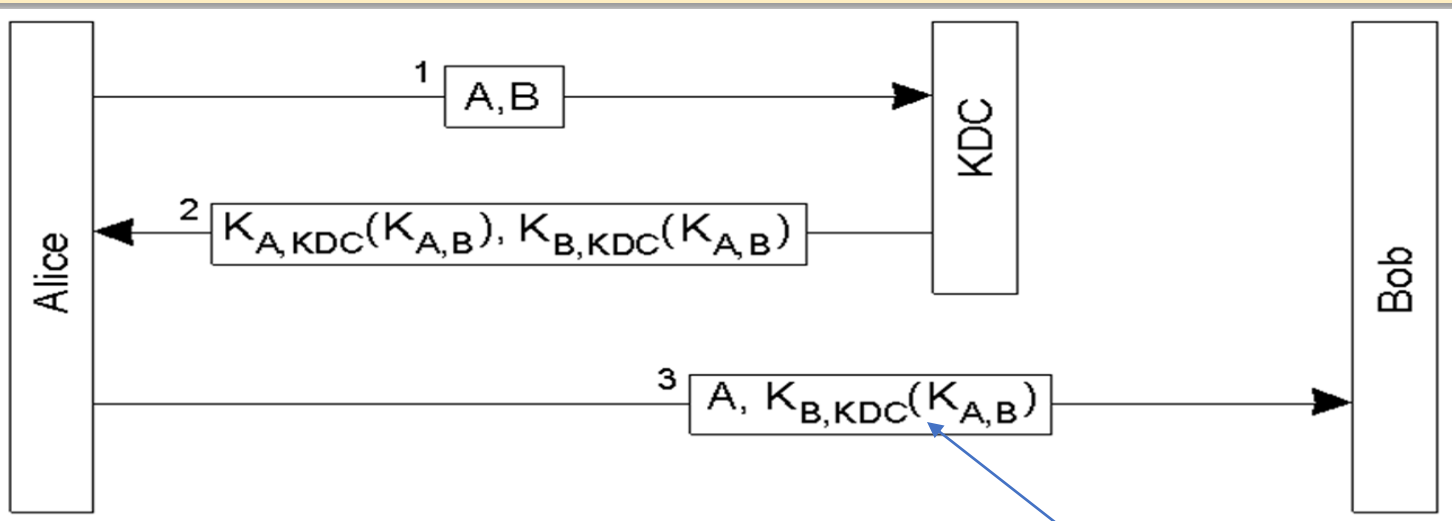
- Um invasor ativo pode solicitar o uso da chave (causando “uso de chave”)
- O que acontece em uma configuração de canal seguro após a autenticação?
- Normalmente, uma chave de sessão (simétrica) é trocada para fornecer integridade e, possivelmente, confiança das seguintes mensagens
- As chaves de sessão são úteis para limitar o uso da chave principal (usada para autenticação)
- Depois que a sessão é fechada, a chave da sessão deve ser destruída
- Um dos problemas com o uso de chave secreta compartilhada para autenticação é a escalabilidade
- Se um sistema distribuído contém N hosts, e cada host é necessário para compartilhar um segredo com cada um dos outros hosts N-1, nós temos N (N-1) / 2 chaves!

CANAIS SEGUROS

KDC

Uma alternativa é usar uma abordagem baseada no *Key Distribution Center (KDC)*

- KDC compartilha segredo com cada uma dos hospedeiros (N chaves) e gera tickets para permitir comunicação entre hospedeiros



Needham-Schroeder authentication protocol – tkt (simplificado)

AUTENTICAÇÃO COM KDC CONFIÁVEL

- $K_{B,KDC}(K_{A,B})$ é chamado ticket
- Este protocol não é seguro para alguns ataques