

# Canais de Comunicação Segura

March 20, 2019

# Sumário

Introdução

Protocolos de Autenticação com Chave Partilhada

Protocolos de Autenticação com Chave Pública

Chaves de Sessão

Camada de Implementação de Canais Seguros

Gestão de Chaves

Leitura Adicional

# Sumário

## Introdução

## Protocolos de Autenticação com Chave Partilhada

## Protocolos de Autenticação com Chave Pública

## Chaves de Sessão

## Camada de Implementação de Canais Seguros

## Gestão de Chaves

## Leitura Adicional

# Canais de Comunicação Seguros

- ▶ Muitos problemas de segurança em redes podem ser mitigados usando canais de comunicação seguros, os quais podem garantir:
  - Autenticidade** i.e. que as mensagens foram enviadas pelo remetente nelas indicado
  - Integridade** i.e. que as mensagens não são forjadas, ou mesmo modificadas em trânsito;
  - Confidencialidade** i.e. que o conteúdo das mensagens não pode ser observado em trânsito;
- ▶ Normalmente, integridade ou confidencialidade não fazem sentido sem autenticidade.
  - ▶ E autenticidade não faz muito sentido sem integridade.

# Autenticação

- ▶ Normalmente, durante a fase de estabelecimento do canal seguro, as duas entidades autenticam-se mutuamente.
  - ▶ Em alguns casos, como na Web, só uma das partes se autentica.
- ▶ Frequentemente, a fase de autenticação inclui também o estabelecimento duma **chave de sessão** que é usada para garantir integridade ou confidencialidade.
- ▶ *Passwords* não são apropriadas para autenticação no estabelecimento de canais seguros.
  - ▶ Normalmente usam-se protocolos de autenticação do tipo *challenge/response*.

# Sumário

Introdução

**Protocolos de Autenticação com Chave Partilhada**

Protocolos de Autenticação com Chave Pública

Chaves de Sessão

Camada de Implementação de Canais Seguros

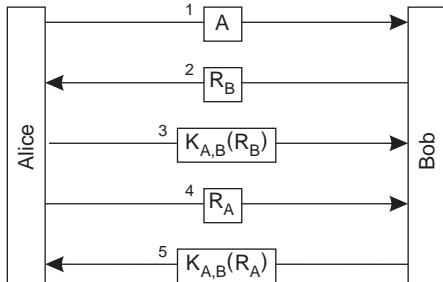
Gestão de Chaves

Leitura Adicional

# Prot. de Autenticação com Chave Partilhada

**Pressuposto** As entidades (Alice/A e Bob/B) nas duas extremidades partilham uma chave secreta ( $K_{A,B}$ )

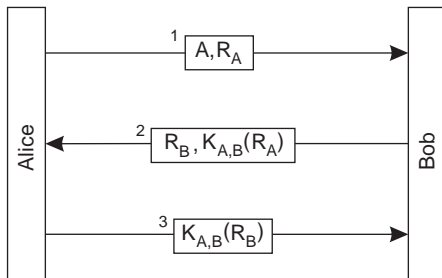
- ▶ Como se obtém a chave secreta, será descrito mais à frente.



- ▶ As mensagens 2 e 3 autenticam A perante B;
- ▶ As mensagens 4 e 5 autenticam B perante A.

# Prot. de Autenticação com Chave Partilhada

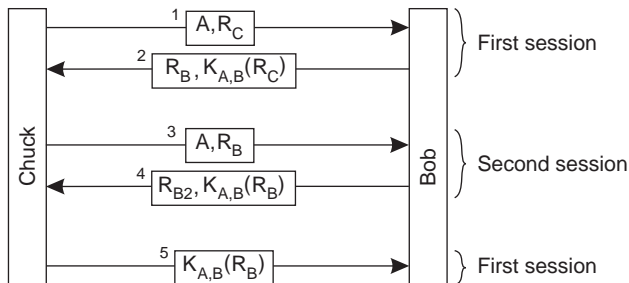
- Uma *versão otimizada* deste protocolo é:



- Mas é vulnerável a *ataques por reflexão (reflection attack)*.



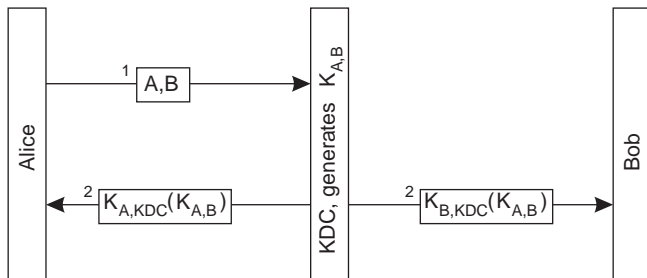
# Ataque por Reflexão



- ▶ O problema é que as 2 entidades usam o mesmo desafio em execuções diferentes.
- ▶ Um princípio para evitar este tipo de ataques é: **introduzir diferenças entre o que as 2 entidades têm que fazer**
  - ▶ P.ex. impôr que as duas partes usem sempre desafios diferentes, p.ex. um par outro ímpar

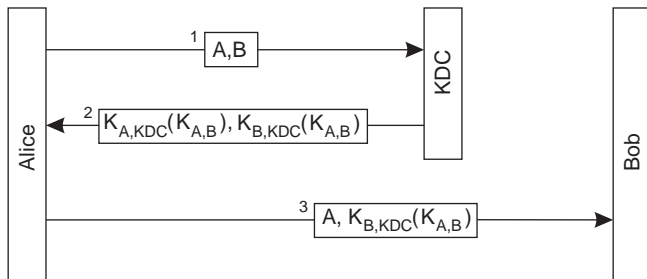
# Autenticação Mediada (c/ KDC) (1/2)

- ▶ A autenticação com chave partilhada é pouco escalável:
  - ▶ Cada par de entidades tem que partilhar uma chave.
- ▶ Uma solução é usar uma entidade mediadora (*Key Distribution Center (KDC)*), na qual todas as entidades **confiam**.
  - ▶ O KDC partilha uma chave com cada uma das entidades.
  - ▶ O KDC gera chaves partilhadas que são usadas para a comunicação entre 2 entidades.



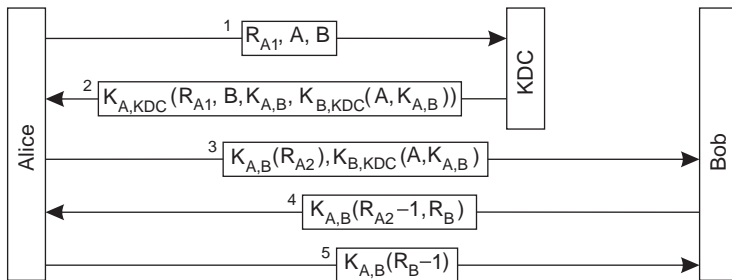
## Autenticação Mediada (c/ KDC) (2/2)

- ▶ E se A enviar uma mensagem para B, e esta chegar antes daquela do KDC?



- ▶ Este protocolo é incompleto:
  - ▶ A e B têm que provar mutuamente que conhecem  $K_{A,B}$

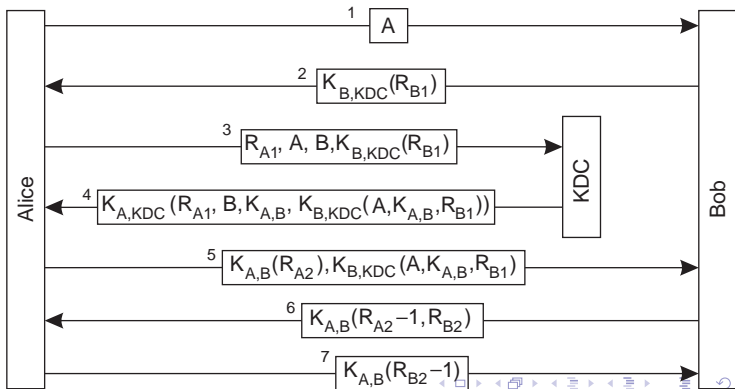
# Protocolo de Needham-Schroeder (1/2)



- ▶ O *nonce* ( $R_{A1}$ ) é usado por A, para garantir que comunica com o KDC (evita *replay attacks*).
- ▶ O KDC inclui B na resposta para impedir que C substitua B por C, na mensagem 1, e conseqüentemente ...
- ▶ O par de mensagens 3, 4 permite que A autentique B
- ▶ O par de mensagens 4, 5 permite que B autentique A
- ▶  $K_{B,KDC}(A, K_{A,B})$  na mensagem 2 designa-se por *ticket to Bob*

## Protocolo de Needham-Schroeder (2/2)

- ▶ Em 1981, Denning e Sacco descobriram uma vulnerabilidade, no caso de C descobrir a chave de A ( $K_{A,KDC}$ ), mesmo que esta tivesse sido substituída por A:
  - ▶ C poderia assumir a identidade de A, na comunicação com B.
- ▶ Em 1987, Needham e Schroeder publicaram uma versão do protocolo original resolvendo o problema.



# Sumário

Introdução

Protocolos de Autenticação com Chave Partilhada

**Protocolos de Autenticação com Chave Pública**

Chaves de Sessão

Camada de Implementação de Canais Seguros

Gestão de Chaves

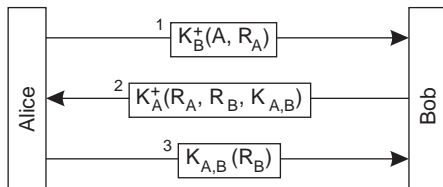
Leitura Adicional

# Autenticação com Chave Públicas

## Pressupostos

1. As entidades (Alice/A and Bob/B) conhecem as chaves públicas uma da outra.
  - ▶ Como se pode obter a chave pública duma entidade, será descrito mais à frente.
2. As chaves privadas de cada uma das entidades é de **seu conhecimento exclusivo**.

## Protocolo Básico



- ▶ Além de autenticação, este protocolo estabelece uma **chave de sessão**.

# Sumário

Introdução

Protocolos de Autenticação com Chave Partilhada

Protocolos de Autenticação com Chave Pública

**Chaves de Sessão**

Camada de Implementação de Canais Seguros

Gestão de Chaves

Leitura Adicional



# Integridade/Confidencialidade de Dados

- ▶ A autenticação das entidades nas extremidades do canal não é suficiente
  - ▶ Admitamos que depois de feita a autenticação, a comunicação é feita "às claras", i.e. sem recurso a criptografia ...
- ▶ Para garantir autenticidade/integridade/confidencialidade dos dados transferidos após a fase de autenticação, A e B poderão usar criptografia:
  - ▶ Criptografia reversível, quer para confidencialidade quer para integridade;
  - ▶ Criptografia não-reversível, para integridade.
- ▶ Encriptação usada para garantir confidencialidade nem sempre é suficiente para garantir integridade.
  - ▶ A menos que os dados cifrados tenham um conteúdo bem definido e a sua alteração possa ser facilmente detectada (p.ex. documento).

# Chave de Sessão

- ▶ Para garantir confidencialidade é conveniente usar uma chave secreta por conversação (**session key**) diferente das chaves usadas durante a autenticação:
  - ▶ Operações com chaves públicas são menos eficiente que operações com chaves partilhadas.
  - ▶ As chaves *desgastam-se* com o uso – quebrar uma chave é tanto mais fácil quanto mais informação cifrada se tiver.
  - ▶ O uso da mesma chave em múltiplas sessões, facilita *replay attacks*.
  - ▶ A descoberta de uma única chave não porá em risco mais do que uma sessão (mesmo que gravadas).
  - ▶ Se o programa usado para cifrar/decifrar fôr de menor confiança, não se põe em risco as chaves de autenticação.

# Chave de Sessão para Chave Partilhada

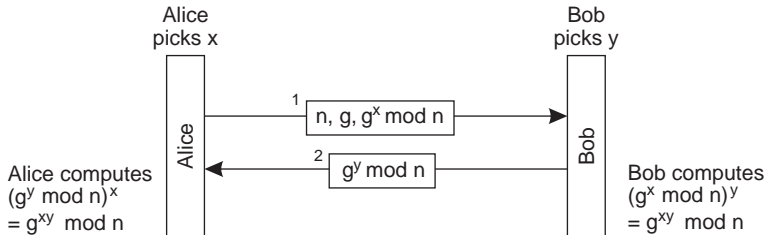
- ▶ Normalmente a chave de sessão pode ser obtida a partir da chave partilhada  $K_{A,B}$  e dos *nonces* usados na fase de autenticação.
- ▶ Características duma chave de sessão:
  - ▶ Deverá ser diferente em cada sessão.
  - ▶ Impossível de adivinhar, mesmo que se escute a comunicação.
  - ▶ **Não** deve ser uma quantidade  $X$  cifrada por  $K_{A,B}$ , sendo  $X$  um valor que prevísível – p.ex.  $R + 1$ .
    - ▶ Uma chave possível é:  $R$  cifrado pela chave  $K_{A,B} + 1$

# Chave de Sessão para Chave Pública

- ▶ Um dos participantes,  $A$  p.ex., escolhe um número aleatório e passa-o a  $B$  cifrado com a chave pública deste último.
  - ▶ Um atacante,  $C$ , pode substituir a chave escolhida por  $A$ , por outra, cifrada com a chave pública de  $B$ .
- ▶ Esta vulnerabilidade pode ser corrigida cifrando o valor acima com a chave privada de  $A$ .
  - ▶ Problema: se  $C$  registar a conversa entre  $A$  e  $B$  e posteriormente se *apossar* da chave de  $B$  e dos seus segredos, será capaz de decifrar a conversa.
  - ▶ Apossar-se da chave de  $A$  não serve:  $C$  precisa da chave privada de  $B$  para obter a chave de sessão.
- ▶  $A$  escolhe um número aleatório  $R_A$ , e envia-o para  $B$ , cifrado com a chave pública de  $B$ .  $B$  faz o mesmo. A chave de sessão pode ser p.ex.  $R_A R_B$ 
  - ▶ Para descobrir a chave de sessão,  $C$  terá que se *apossar* quer de  $A$  quer de  $B$ .

# Protocolo de Acordo de Chave de Diffie-Hellman

- ▶  $n$  e  $g$  são 2 grandes números, sujeitos a certas propriedades matemáticas:
  - ▶ Têm que ser acordados *a priori*, e podem ser públicos.
- ▶ Cada um dos lados escolhe um número privado muito grande,  $x$  e  $y$  respectivamente, e executa o protocolo:



- ▶ A chave de sessão pode ser  $g^{xy} \bmod n$ 
  - ▶ Só  $A$  e  $B$  podem calcular este valor.
  - ▶ Se  $A$  e  $B$  se esquecerem de  $x$  e de  $y$ , mesmo que  $C$  se apossar de  $A$  e de  $B$  não conseguirá decifrar a comunicação (*Perfect Forward Secrecy*).

# Chave de Sessão para Aut. Unidireccional

- ▶ Na Web, usa-se normalmente autenticação *unidireccional* com chave pública:
  - ▶ Os servidores autenticam-se usando chave pública.
  - ▶ Os clientes não
    - ▶ A administração de chaves públicas à escala da Internet não é fácil.
- ▶ Neste caso, a chave de sessão pode ser obtida:
  1.  $A$  pode escolher a chave de sessão e enviá-la a  $B$  cifrada na sua chave pública.
  2.  $A$  e  $B$  podem executar Diffie-Hellman, mas só  $B$  se autentica.
- ▶ Em qualquer caso:
  - ▶  $B$  **não** tem garantia de estar a comunicar com  $A$
  - ▶ Mas,  $B$  tem garantia de que comunica com uma **única** entidade.

# Sumário

Introdução

Protocolos de Autenticação com Chave Partilhada

Protocolos de Autenticação com Chave Pública

Chaves de Sessão

**Camada de Implementação de Canais Seguros**

Gestão de Chaves

Leitura Adicional

# Camada de Implementação de Canais Seguros

- ▶ Teoricamente um canal seguro pode ser implementado em qualquer camada:

**Ligação de dados** - p.ex. Wired Equivalent Privacy;

- ▶ Protege apenas a comunicação num segmento.

**Rede** - p.ex. IPSec

- ▶ Normalmente, implementado ao nível do *kernel* do SO;
- ▶ Normalmente, só usa endereços IP para identificação – e autenticação.

**Transporte** - p.ex. SSL/TLS

- ▶ Requer modificação das aplicações – (*sockets*);
- ▶ Vulnerável a ataques (*denial of service*) usando mensagens TCP forjadas.

**Aplicação** - p.ex. `ssh`, SMIME, PGP

- ▶ Protege os *objectos* da aplicação, p.ex. *mensagens de email* armazenadas em servidores.



# Sumário

Introdução

Protocolos de Autenticação com Chave Partilhada

Protocolos de Autenticação com Chave Pública

Chaves de Sessão

Camada de Implementação de Canais Seguros

**Gestão de Chaves**

Leitura Adicional

# Distribuição de Chaves

**Problema** Como é que se obtém as chaves necessárias para realizar a autenticação?

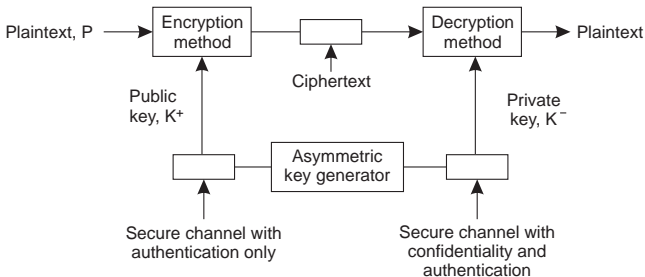
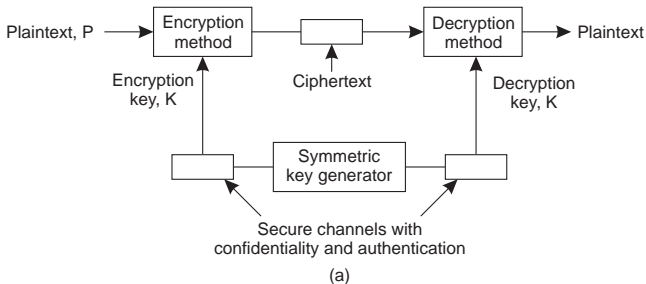
- ▶ Os protocolos apresentados pressupõem que cada entidade conhece uma chave relacionada com a outra entidade:

**Chave partilhada** no caso de sistemas de chave secreta;

**Chave pública** no caso de sistemas de chave pública.

**Solução** Depende uma vez mais se a chave a distribuir é uma chave secreta ou pública.

# Distribuição de Chaves: Solução



# Certificados de Chave Pública (1/3)

- ▶ O problema com chaves secretas é garantir que são secretas.
- ▶ O problema com chaves públicas é garantir a sua associação a uma entidade – na verdade, o máximo que se consegue é garantir a sua associação à chave privada dessa entidade.
  - ▶ Se C convencer A de que a chave pública de B é a chave pública de uma chave privada do seu (de C) conhecimento, então C poderá fazer passar-se por B.
- ▶ A solução adoptada são os ***certificados de chave pública*** (*public-key certificates*) os quais contêm:
  1. O nome da entidade.
  2. A chave pública correspondente à sua chave privada.
  3. A assinatura do conjunto por uma autoridade de certificação (*Certification Authority (CA)*).
  4. A identidade da CA.

## Certificados de Chave Pública (2/3)

- ▶ O pressuposto é que a chave pública das CAs são bem conhecidas
  - ▶ P.ex., muitos navegadores (*browsers*) da Web, incluem as chaves públicas de muitas CAs.
- ▶ Ao aceitar um certificado, o cliente **confia** que não é forjado.
  - ▶ Mas mesmo CAs como a Verisign e empresas como a Microsoft são por vezes *levadas*
- ▶ Na Web, o modelo de confiança associado aos certificados é hierárquico, mas há outros modelos
  - ▶ PGP baseia-se num modelo de confiança em cadeia, não hierárquico, e subjectivo.

## Certificados de Chave Pública (3/3)

- ▶ Um certificado inclui o prazo de validade, após o qual o certificado expira:
  - ▶ Os navegadores tipicamente avisam quando o certificado dum servidor visitado expirou.

**Problema** E se o certificado fôr *comprometido* antes de expirar?

**Solução** Revogar os certificados.

- ▶ As CAs publicam periodicamente listas de certificados revogados (*Certificate Revocation List (CRL)*).
- ▶ Qual deve ser o período de publicação duma CRL?
- ▶ Alguns *browsers* permitem o uso de protocolos para verificar certificados, e.g. o *Online Certificate Status Protocol (OCSL)* (RFC 2560)

# Sumário

Introdução

Protocolos de Autenticação com Chave Partilhada

Protocolos de Autenticação com Chave Pública

Chaves de Sessão

Camada de Implementação de Canais Seguros

Gestão de Chaves

**Leitura Adicional**

# Leitura Adicional

- ▶ Capítulo 9 de Tanenbaum e van Steen, *Distributed Systems, 2nd Ed.*
  - ▶ Secção 9.2: *Secure Channels*
  - ▶ Subsecção 9.4.1: *Key Management*