Sistemas Distribuídos

Aula 11 – Segurança

DCC/IM/UFRRJ
Marcel William Rocha da Silva

Objetivos da aula

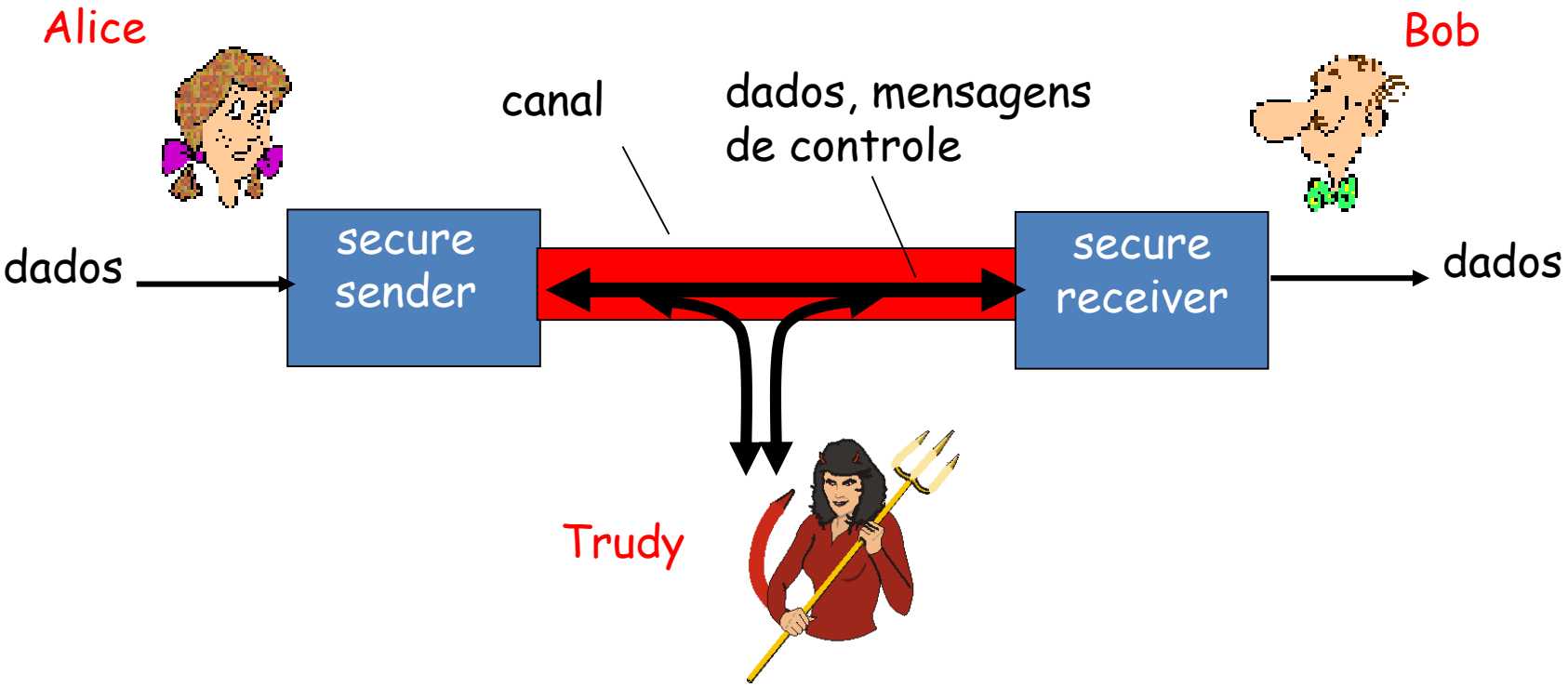
- **Aula anterior**
  - Tolerância à Falha
- **Aula de hoje**
  - Segurança
    - Criptografia
    - Canais seguros
    - Controle de acesso
    - Gerenciamento de segurança

Segurança em sistemas distribuídos

- A segurança de um sistema distribuído passa pelos seguintes pontos:
  - **Confidencialidade**: apenas o remetente e o destinatário são capazes de entender o conteúdo das mensagens
  - **Integridade**: remetente e destinatário querem impedir que ocorram mudanças nas mensagens (intencionais ou não)
  - **Autenticação**: remetente e destinatário querem confirmar a identidade um do outro
  - **Disponibilidade e Controle de acesso**: serviços precisam estar disponíveis e ser acessados apenas por usuários qualificados

Modelo para o estudo de segurança

- **Alice** e **Bob** querem se comunicar de forma segura
- **Trudy** (intruso) que deseja "interferir" na comunicação



Modelo para o estudo de segurança

- O que um intruso pode fazer?
  - Interceptar mensagens
  - Inserir novas mensagens da conexão
  - Forjar pacotes com outro endereço de origem
  - Sequestrar um conexão (remover Alice ou Bob)
  - Impedir que outros se comuniquem

Políticas e mecanismos de segurança

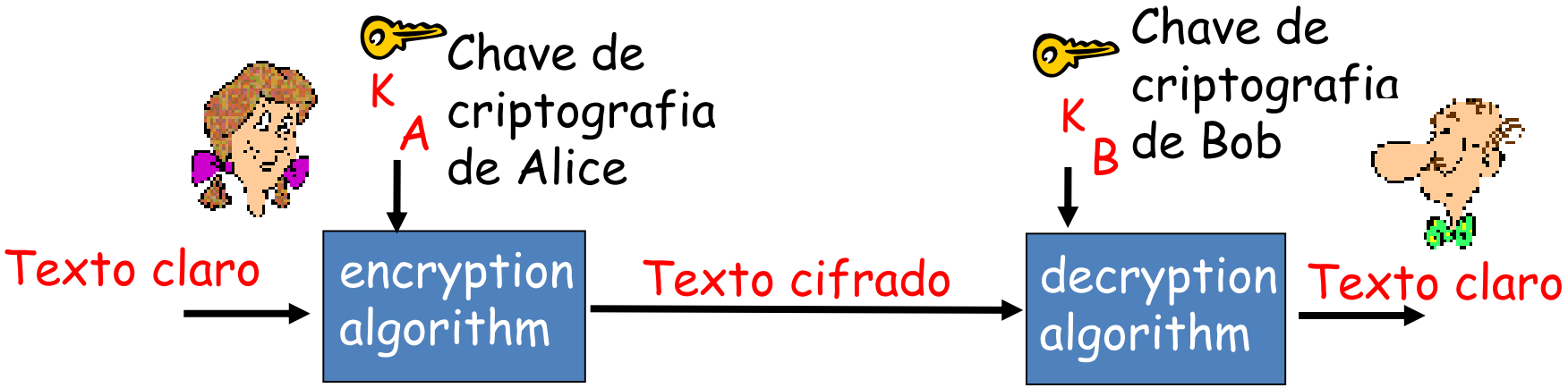
- Não basta declarar que um SD deva resolver todos os problemas anteriores para ter segurança

- Devemos definir:
  - **Políticas de segurança** → quais ações as entidades de um sistema têm permissão de realizar e quais são proibidas
  - **Mecanismos de segurança** → ferramentas para impor as políticas de segurança desejadas

Mecanismos de segurança

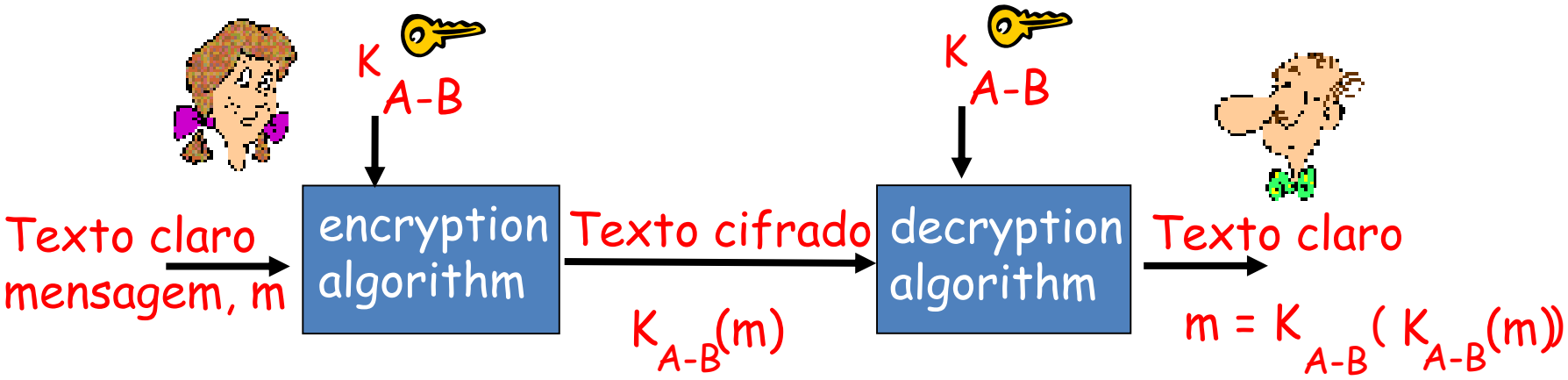
- **Criptografia**: transformação de dados em algo que um atacante não possa entender
- **Autenticação**: validação da identidade do par de comunicação (usuário, cliente, servidor ou outra entidade)
- **Autorização**: após autenticação, verifica se o par possui permissão de executar a ação
- **Auditoria**: rastrear quais recursos foram acessados por quais pares, e de que maneira

Criptografia



- Duas possibilidades:
  - **Chaves simétricas**: Alice e Bob possuem uma mesma chave ($K_A = K_B$)
  - **Chave pública (chaves assimétricas)**: criptografa com uma chave pública, decriptografa com uma chave privada ($K_A \neq K_B$)

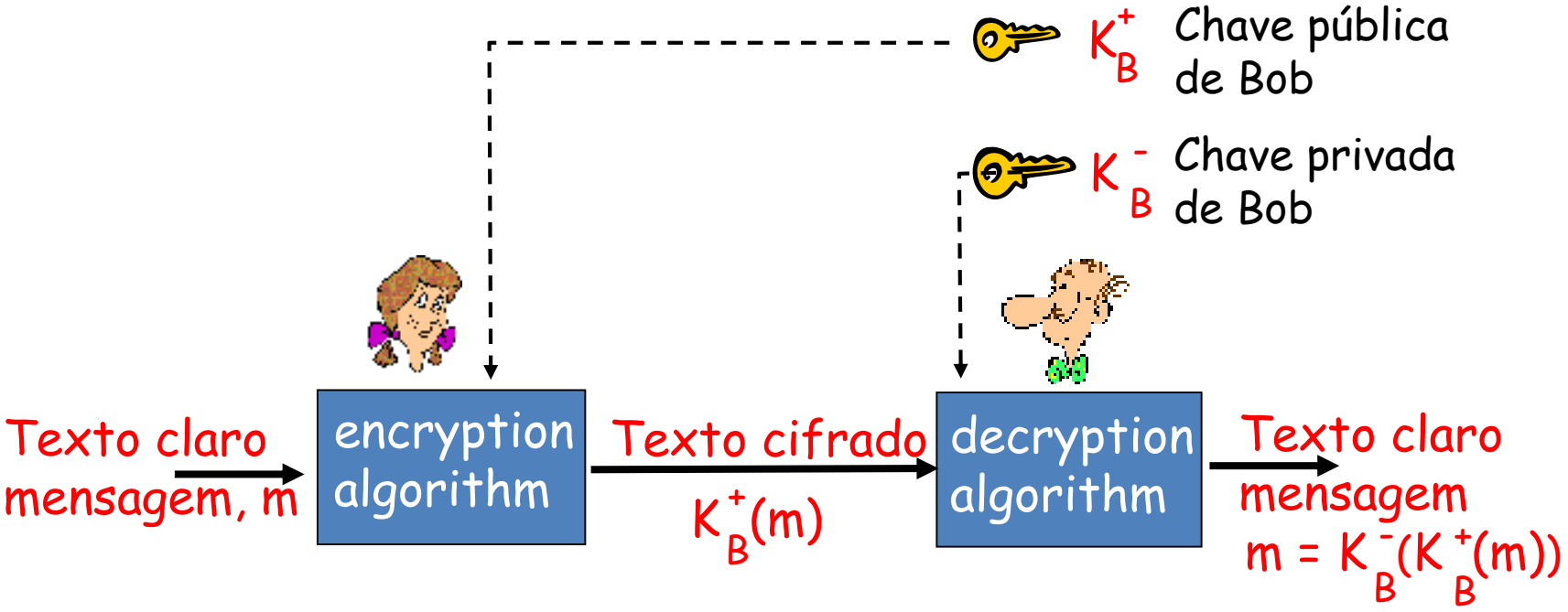
Criptografia com chaves simétricas



- Ambas as partes compartilham uma mesma chave
  - Serve para criptografar e decriptografar o texto
- Exemplos de algoritmos muito usados na prática
  - DES e AES

Criptografia com chaves públicas

- Utiliza um um par de chaves:
  - **Chave publica** → conhecida por todos
  - **Chave privada** → conhecida apenas pelo destinatário



Criptografia com chaves públicas

- **Algoritmo RSA**
  - Utiliza conceitos da teoria dos números
  - As chaves são escolhidas de tal forma que através da chave pública K^+_B é impossível obter a chave privada K^-_B
  - A seguinte equivalência também é garantida:

$$\underbrace{K^-_B(K^+_B(m))}_{\text{Decriptografar com a chave pública uma mensagem criptografada com a chave privada}} = m = \underbrace{K^+_B(K^-_B(m))}_{\text{Decriptografar com a chave privada uma mensagem que foi criptografada com a chave pública}}$$

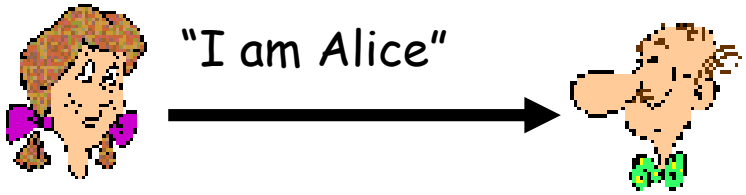
Canais seguros

- Um **canal seguro** protege remetente e destinatário de:
  - Interceptação de mensagens (confidencialidade)
    - Garantida com uso de criptografia
  - Modificação e "invenção" de mensagens (autenticação e integridade)
    - Garantidas com mecanismos de **autenticação** e **assinatura digital**

Autenticação

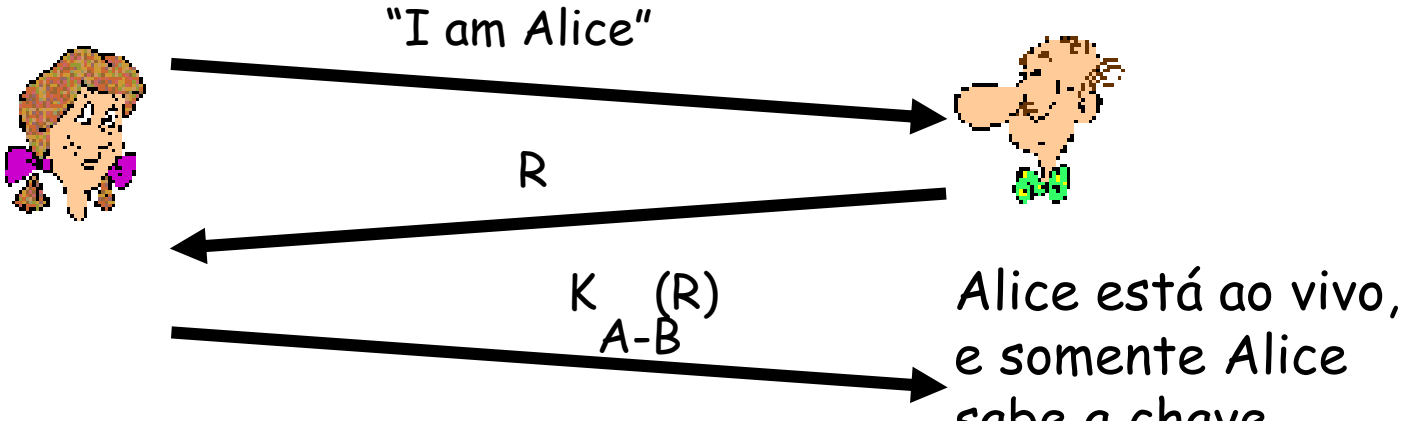
- Objetivo → Permitir que Alice prove sua identidade à Bob

- Duas possibilidades
  - Usando chaves **simétricas** ou **públicas**



Autenticação com chaves simétricas

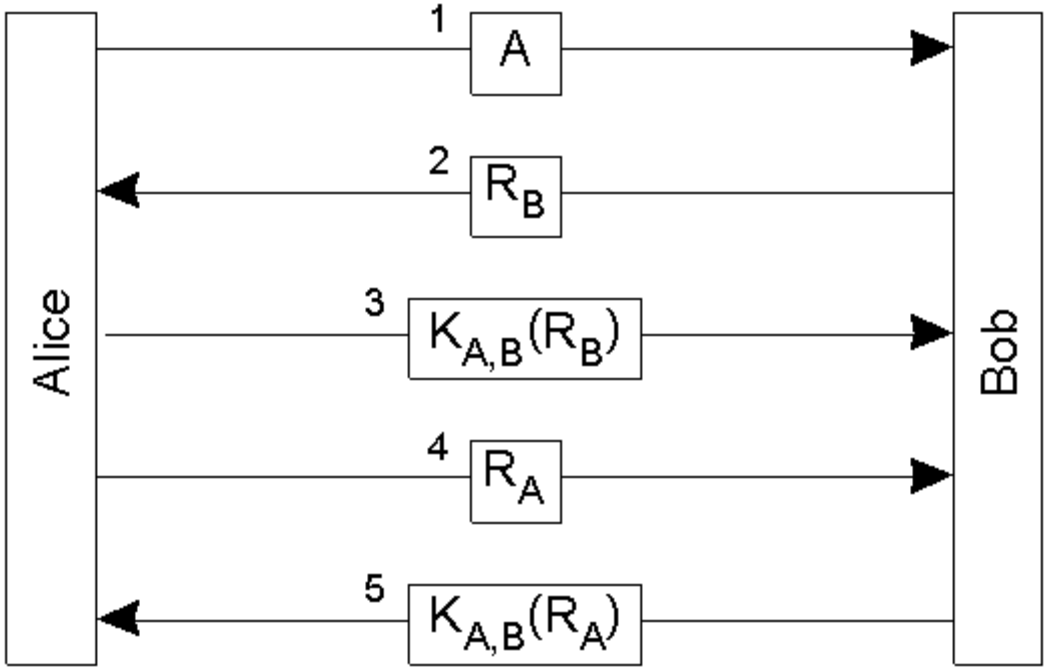
- Provar que Alice está “ao vivo” na outra ponta
- Bob envia um nonce (**R**) que será criptografado na hora por Alice com a chave simétrica
  - **nonce** → número que será usado apenas uma vez



Problema → Precisam compartilhar antecipadamente de alguma forma a chave simétrica

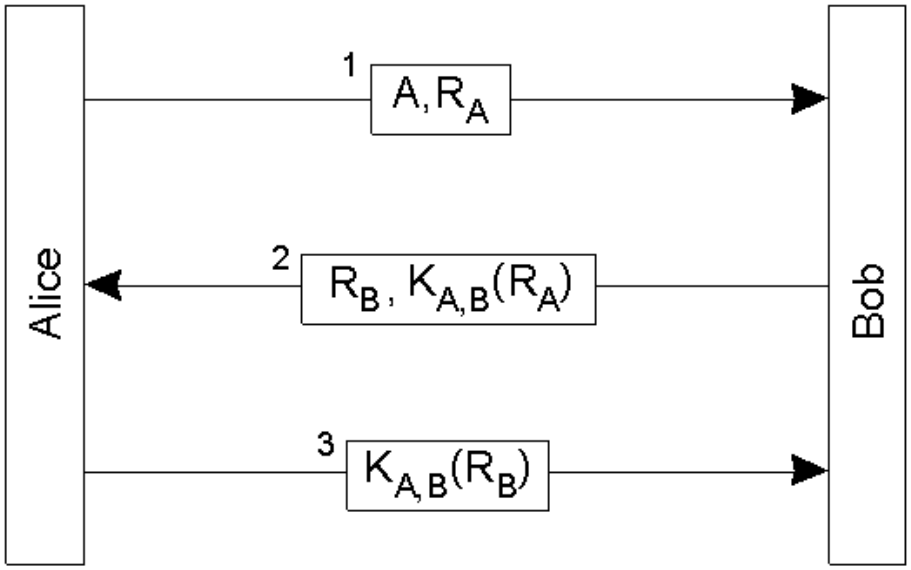
Autenticação com chaves simétricas

- Na prática → queremos autenticar ambas as partes



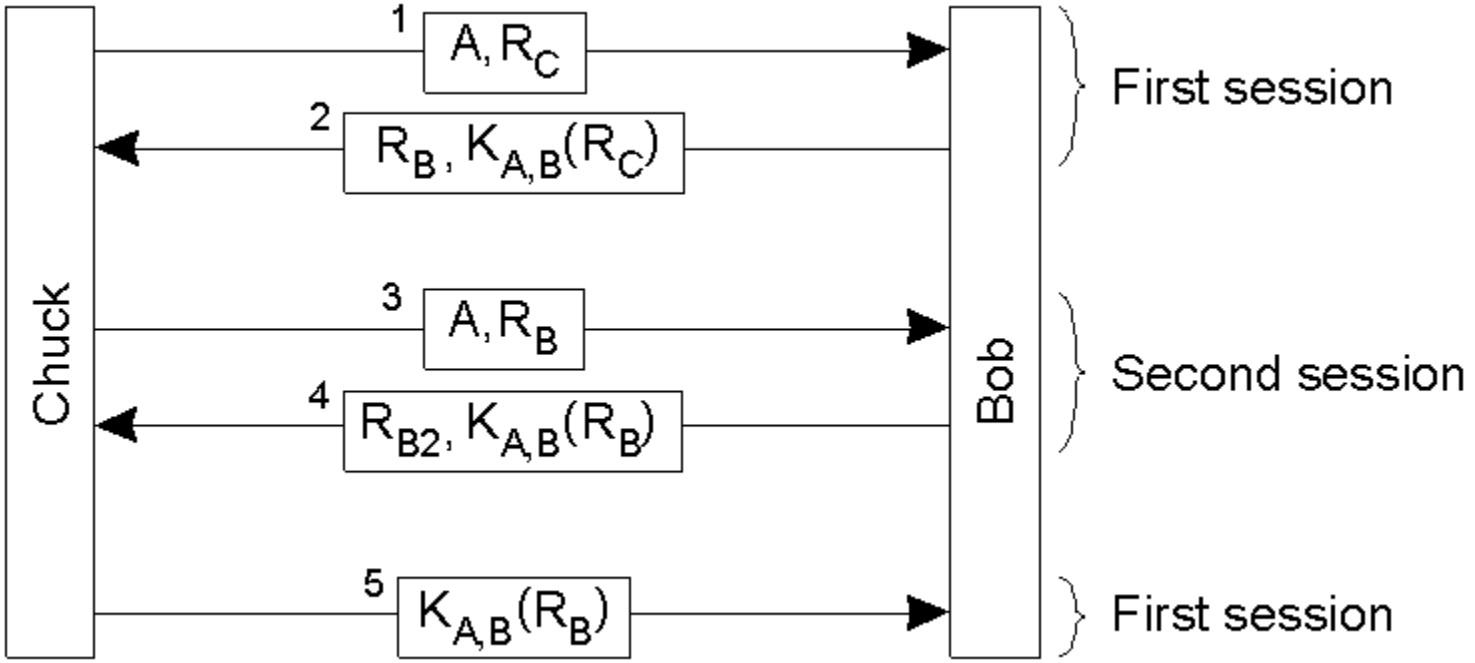
Autenticação com chaves simétricas

- Possível reduzir o número de etapas
- **Cuidado!** → Melhoria torna o protocolo sujeito a **ataques de reflexão**



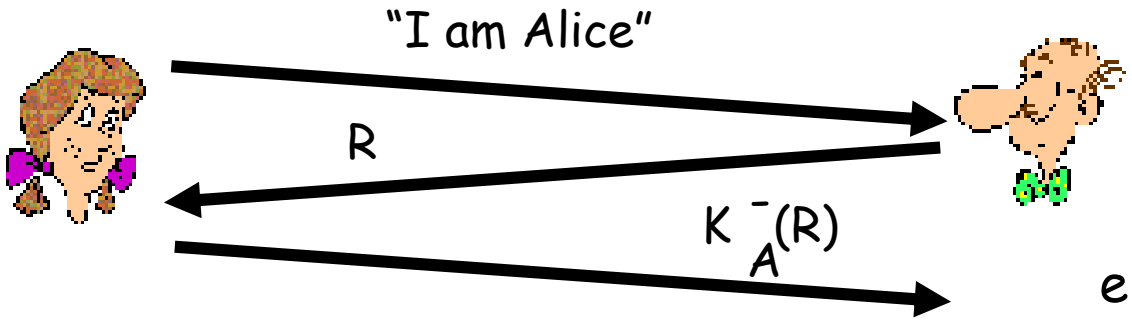
Autenticação com chaves simétricas

- Exemplo de ataque de reflexão



Autenticação com chave pública

- Idem ao anterior, mas usando chave pública



Bob computa:

$K_A^+(K_A^-(R)) = R$

e sabe que apenas Alice poderia ter esta chave privada que cifrou R, tal que:

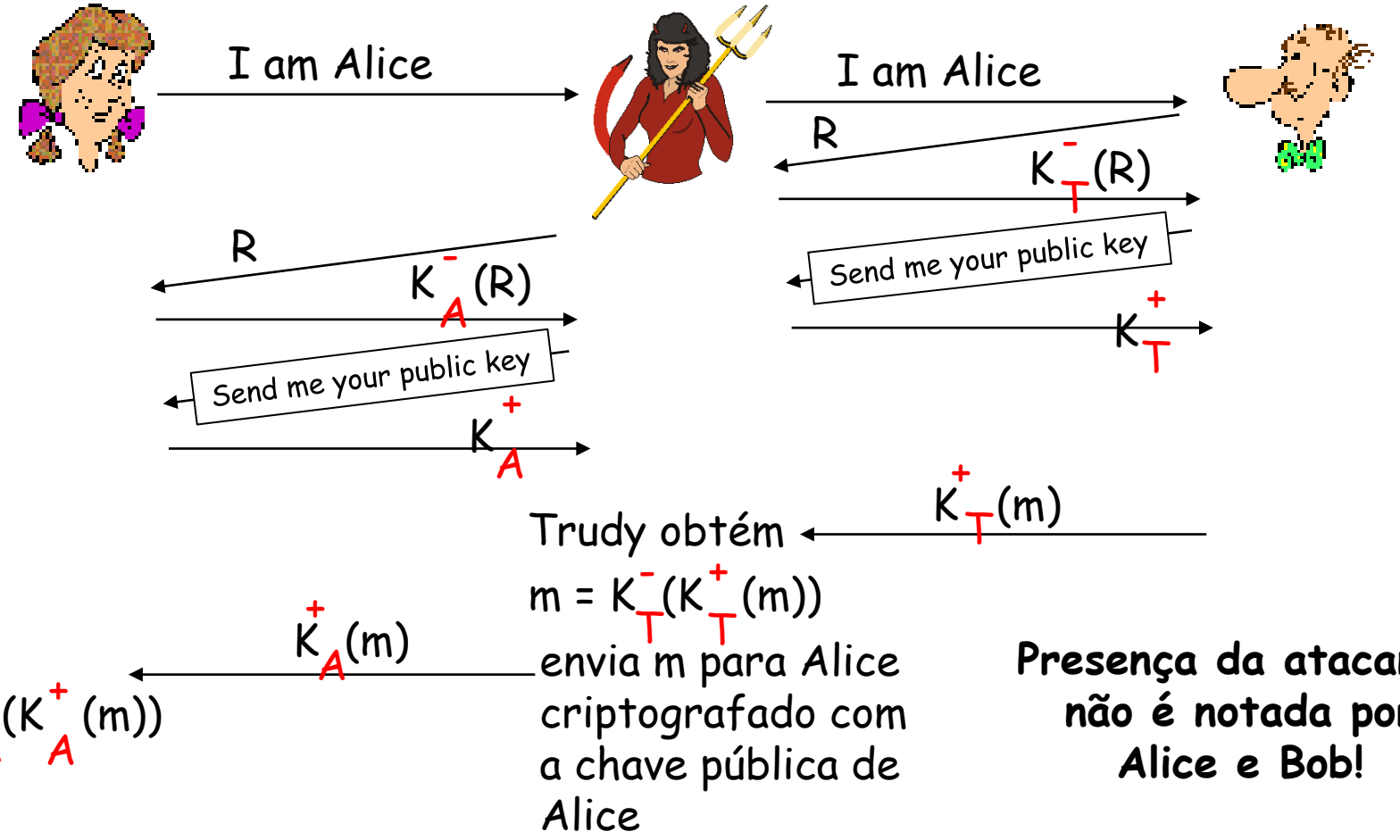
$K_A^+(K_A^-(R)) = R$

Problema → Como obter a chave pública da Alice?

Distribuição de chaves é um problema de segurança importante!

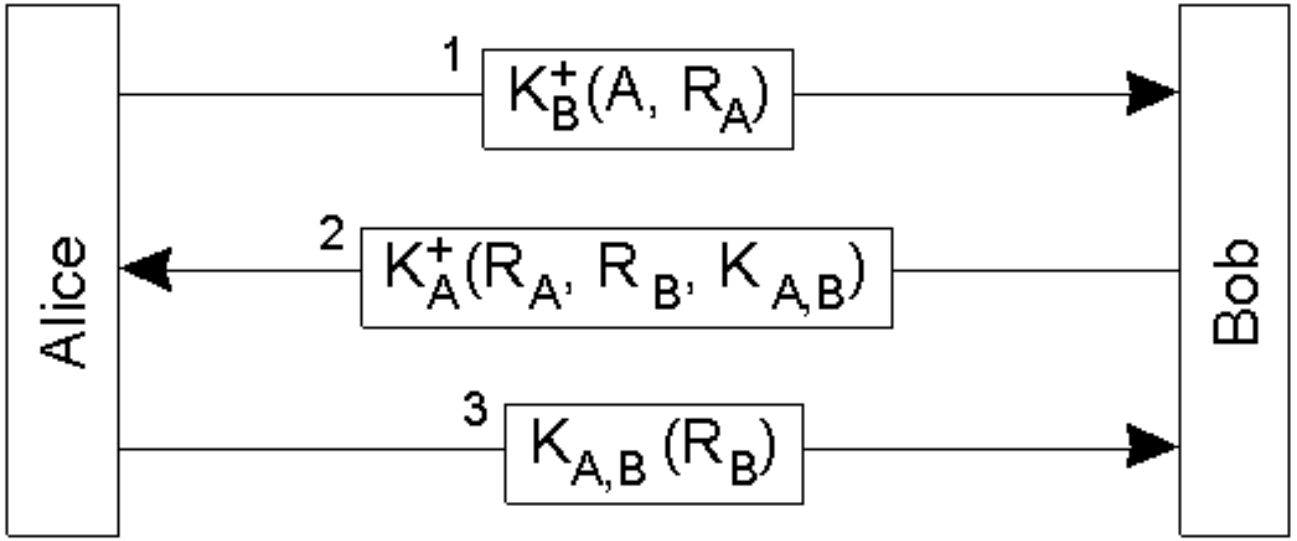
Autenticação com chave pública

- Distribuir a chave durante a autenticação
  - **Problema**: Ataque do **homem do meio** (*man in the middle*)



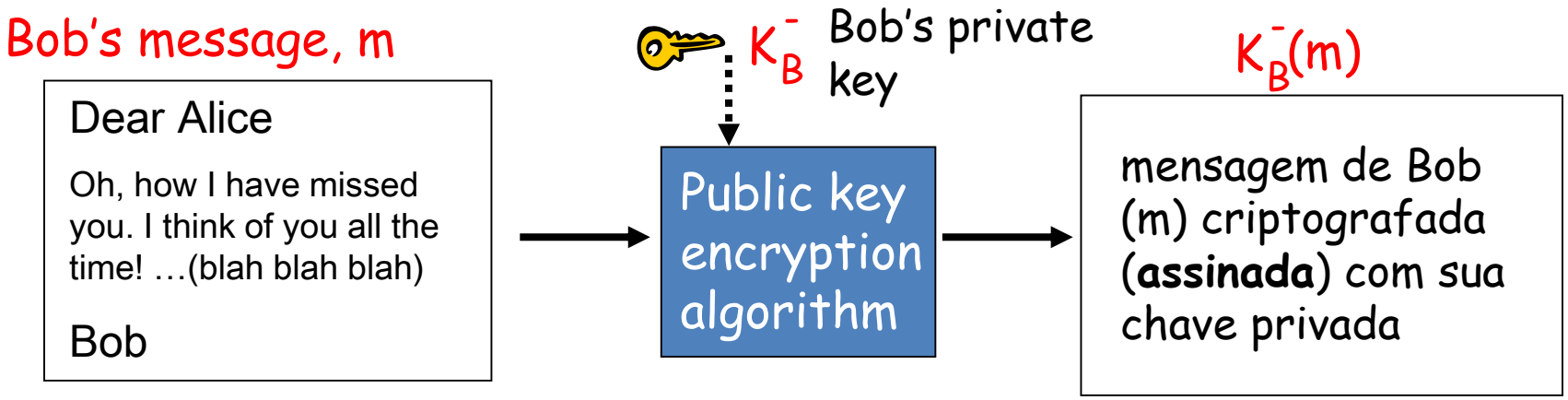
Autenticação com chave pública

- Variante melhor do protocolo anterior
  - Mas também tem problemas → seria necessário Alice obter a chave pública de Bob de uma maneira segura!



Integridade

- **Assinatura digital** → permite verificar que Bob é o único possível criador da mensagem
  - Também usa chave pública
- Bob cifra sua mensagem com sua chave privada
  - Integridade pode ser verificada fazendo: $\mathbf{K^+_B(K^-_B(m))=m}$



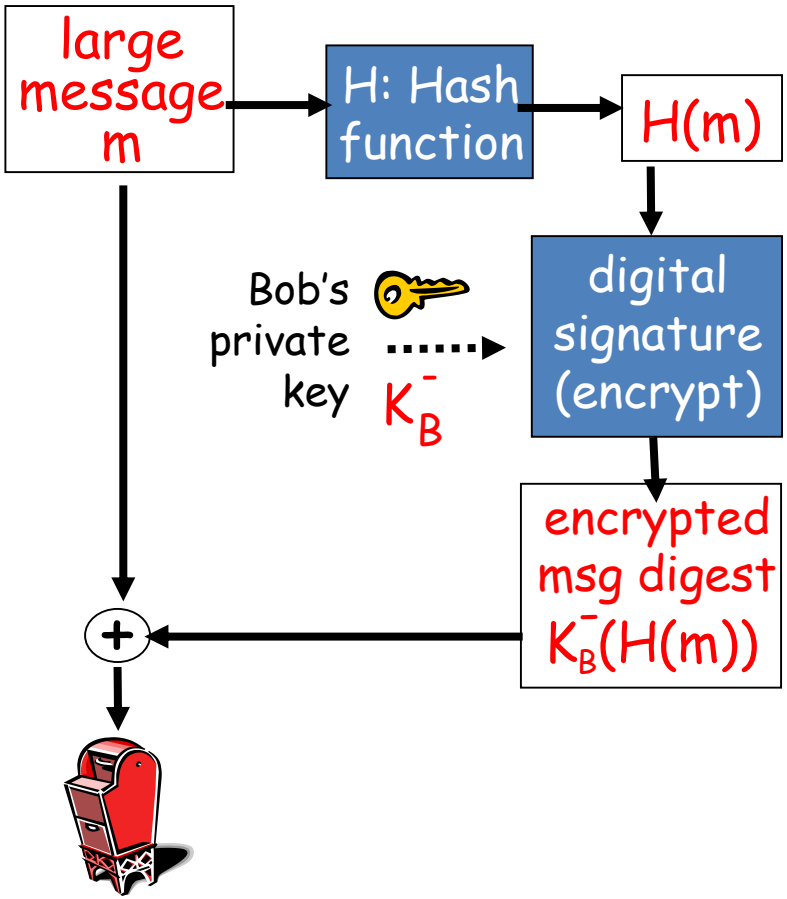
Integridade: Resumo de mensagem

- Pode ser muito custo cifrar a mensagem inteira
  - Principalmente com mensagens muito grandes

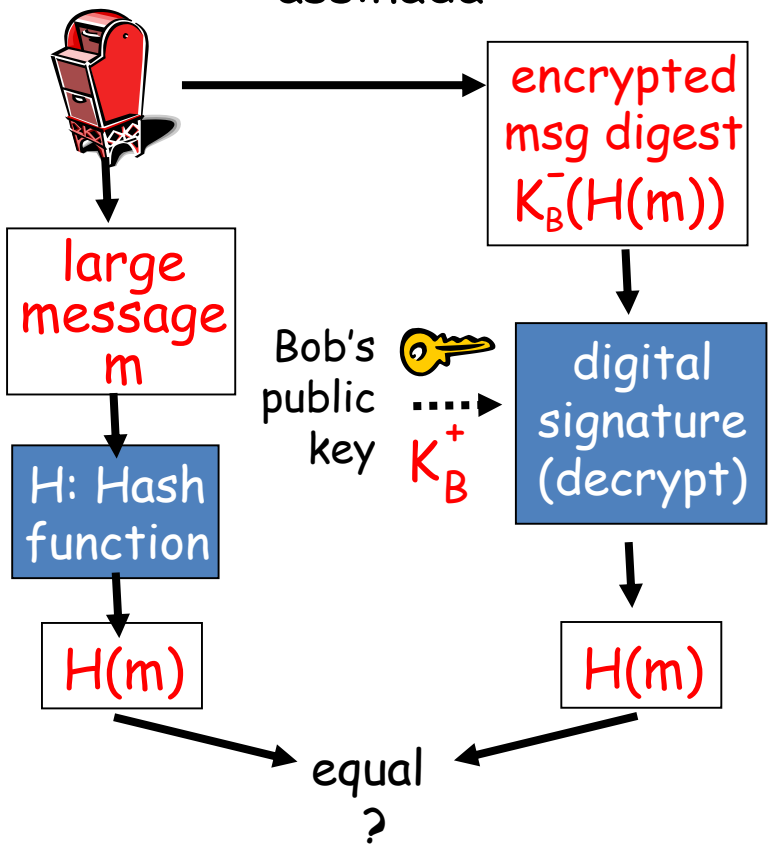
- Solução → apenas um **resumo da mensagem** é cifrado
  - Aplicar uma função ***hash*** para reduzir o tamanho da mensagem e tornar mais rápido sua criptografia
  - Função ***H(.)*** que mapeia os bits de uma mensagem ***m*** em um resumo com poucos bits
  - Exemplos de algoritmos de *hash* muito usados
    - MD5 e SHA1

Integridade: Resumo de mensagem

Bob envia mensagem assinada digitalmente:



Alice verifica a assinatura e a integridade da mensagem assinada:



Comunicação segura entre grupos

- Conjunto de N processos que desejam se comunicar de maneira segura

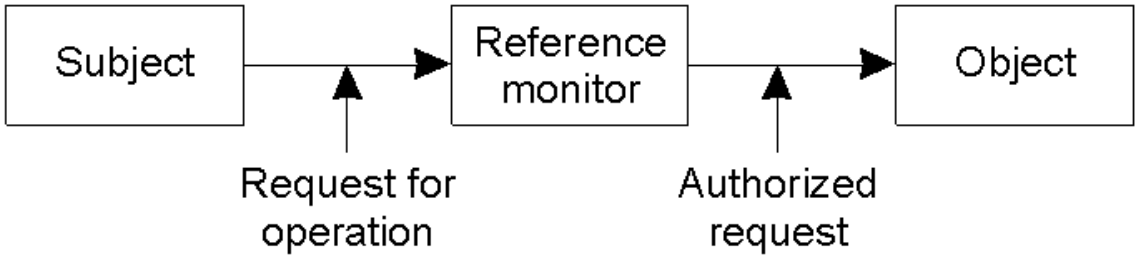
- Esquema simples → todos compartilharem a mesma chave secreta
  - Todos devem ser de confiança
  - Sistema fica mais vulnerável a ataques quando comparados com canais seguros ponto-a-ponto

Comunicação segura entre grupos

- Alternativas...
  - Usar uma chave secreta compartilhada a cada par de membros do grupo
    - Se um membro torna-se não confiável, basta parar de mandar mensagens para ele → outras comunicações podem prosseguir
    - Problema → seria necessário manter um número grande de chaves (N(N-1)/2)
  - Usar chaves pública/privada
    - Cada membro tem seu par de chaves
    - Necessárias N chaves ao todo

Controle de acesso

- Após estabelecer um canal seguro → cliente pode enviar requisições a um servidor
- Entretanto, servidor pode estabelecer regras sobre quais requisições podem ser executadas por cada cliente sobre quais objetos
  - **Controle de acesso**
- Modelo utilizado para o estudo do controle de acesso:



Controle de acesso

- Uma maneira de modelar as regras é através de uma **matriz de controle de acesso**
  - Linhas representam os sujeitos
  - Colunas representam os objetos
  - Posições da matriz contém uma lista de operações permitidas

- Problema → matriz pode ser muito grande e esparsa
  - Solução: atribuir a cada objeto uma **lista de controle de acesso** (ACL)

Controle de acesso

- ACLs também podem ficar extensas
  - Se muitos sujeitos no sistema

- Solução → criar **domínios de proteção**
  - **Grupos** de usuários → conjuntos de usuários com as mesmas permissões
  - **Certificados** → deixar o usuário carregar sua própria ACL (deve ser impossível o usuário modificar seu certificado)
  - Definir **papéis** → permissões são atribuídas aos papéis definidos e cada usuário possui um ou mais papéis

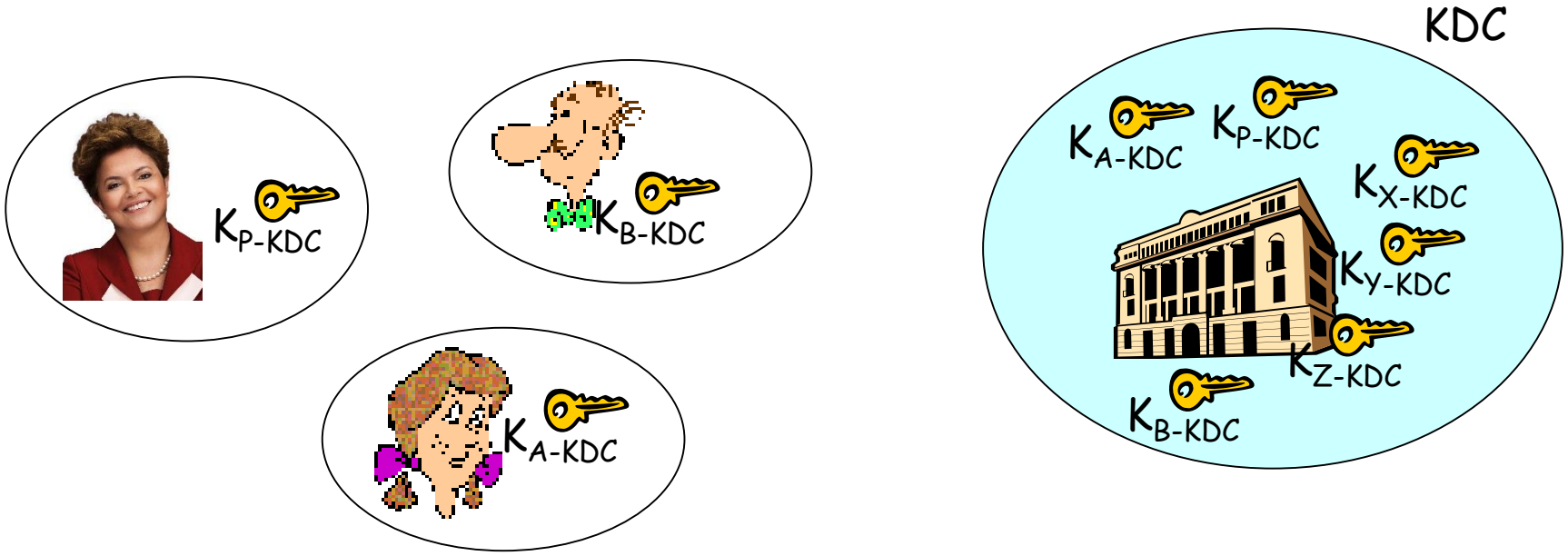
Gerenciamento de segurança

- Problema com chaves simétricas ou chave pública
  - Como distribuir em segredo um chave simétrica?
  - Como saber que a chave publica é realmente de Bob?

- Necessita de um **intermediário de confiança**
  - **Central de distribuição de chaves** no caso de chaves simétricas
  - **Autoridade certificadora** no caso de chaves públicas

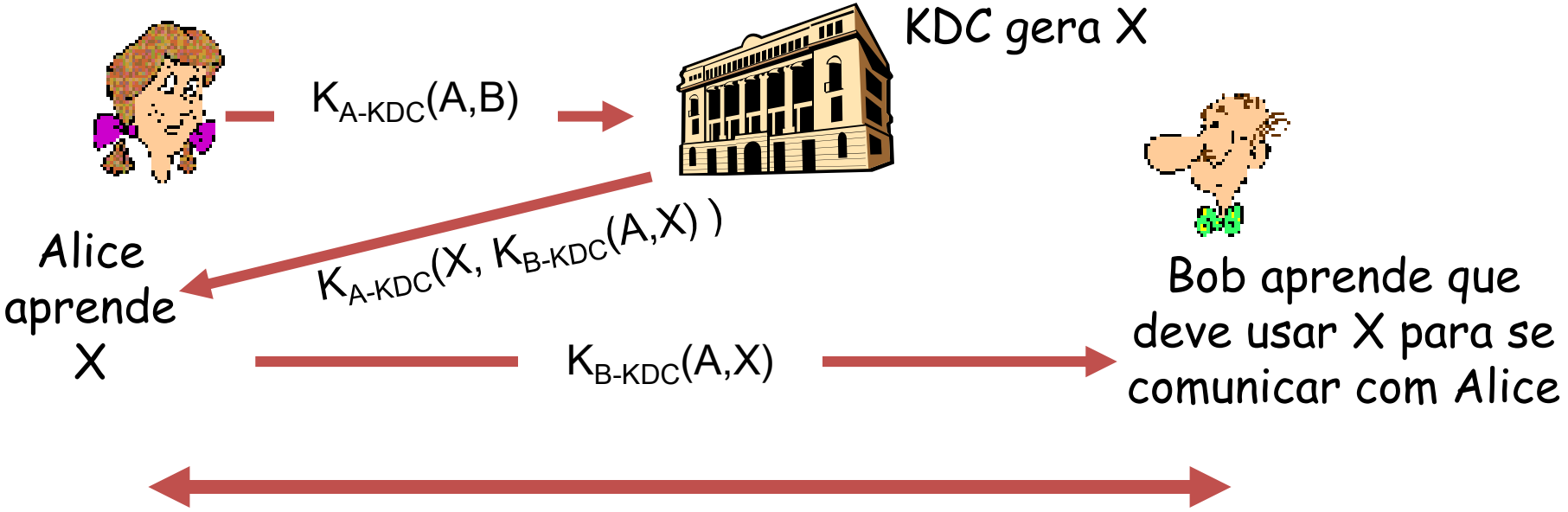
Central de distribuição de chaves

- ***Key distribution center*** (**KDC**)
- Entidade que compartilha chaves simétricas com seus vários usuários
  - Cada usuário possui uma chave simétrica única que usa para se comunicar com o KDC



Central de distribuição de chaves

- Como Alice e Bob obtém do KDC uma chave simétrica secreta para se comunicarem?



Autoridade certificadora

- ***Certification authority*** (**CA**)
- CA é responsável por emitir **certificados**
  - Associação entre uma entidade e sua chave pública
  - Entidade fornece "provas de identidade" para a CA
- Certificados emitidos possuem uma assinatura digital com a chave privada da CA
  - Chave pública da CA poderia ser distribuída por algum outro meio confiável