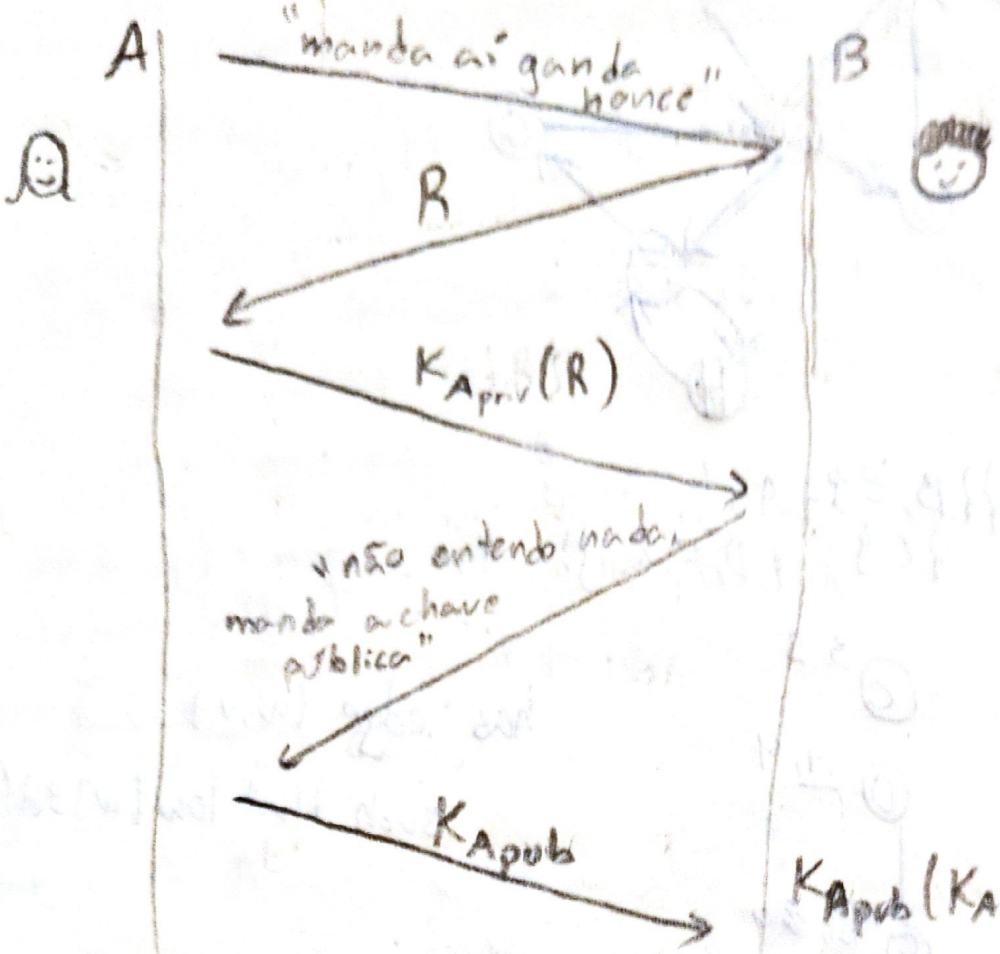
EN2022

(1)

R ≡ Nonce

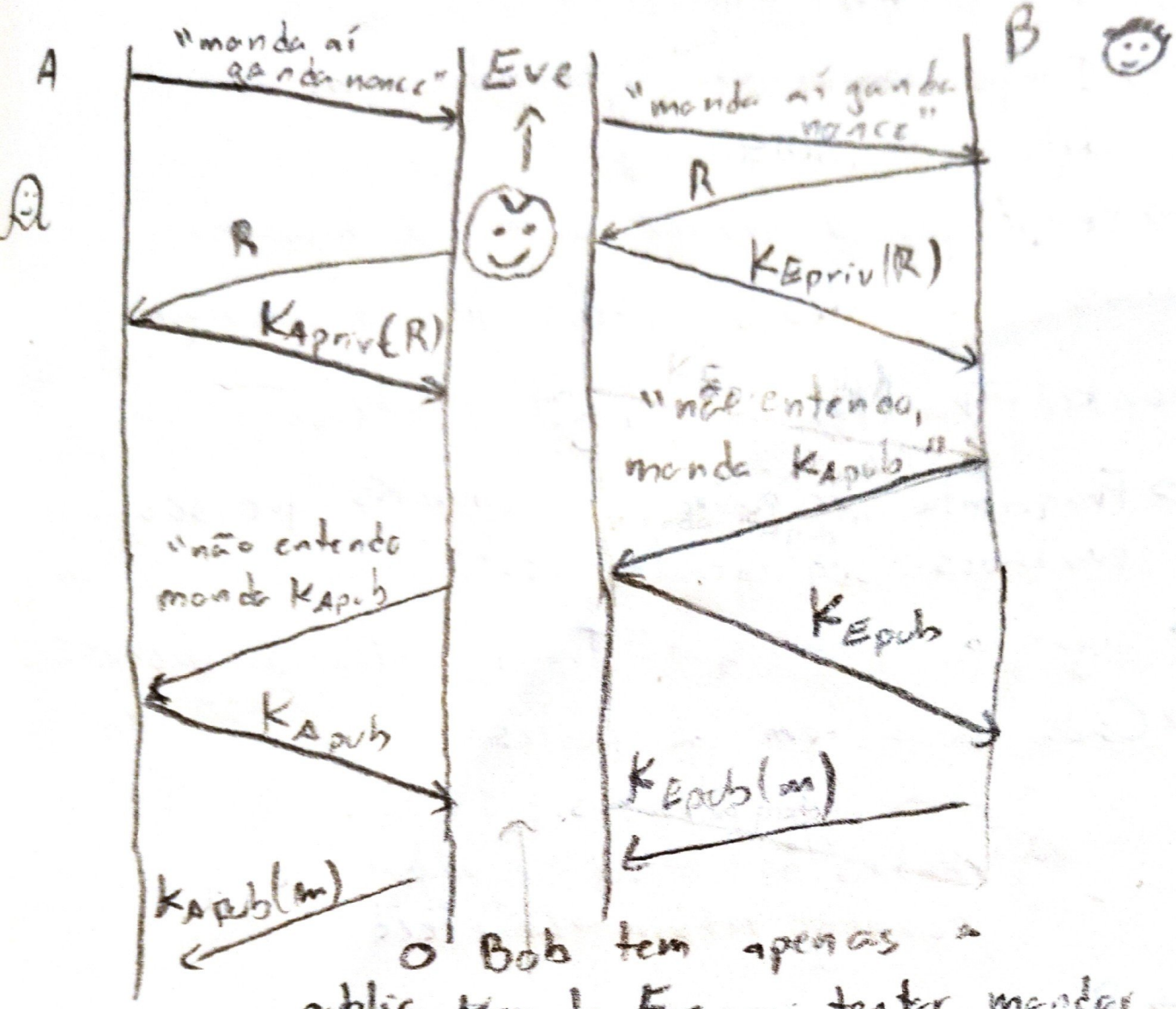


$K_{A_{pub}}(K_{A_{priv}}(R)) = R$

"chiça, este é precisamente o meu nonce, aposto que é a minha grande amiga Alice :3"

O que o burro do Bob acha que aconteceu

O que realmente acontecceu



O Bob tem apenas a public key da Eve, vai tentar mandar msgs à Alice com a public key da Eve, como tal, a Eve vai conseguir ler tudo!!

Como a Eve tb tem a public key da Alice, ao receber $K_{Epub}(m)$ de Bob, pode obter $m = K_{Epriv}(K_{Epub}(R))$ e enviar $K_{Apub}(m)$ a Alice!!

Suscetível a Man-in-the-middle!!!!

(2)

1- Content-length:

→ Campo no header que indica nº de bytes da mensagem.

→ Bom p/ qdo o server sabe o tamanho total do conteúdo antes de o enviar

2- Chunked Transfer-Encoding:

→ Fragmenta resposta em chunks que são enviados sequencialmente.

→ Ligar no header com: Transfer-Encoding: chunked

→ Cada bloco tem 2 partes:

1- Linha com block size

2- Dados do bloco seguidos de carriage return → Line feed

→ Bom para qdo server n/ sabe tamanho total da resposta

com header
↗ Connection: close

Nota: Se for HTTP/1.0 ou HTTP/1.1 a ligação TCP ao server é fechada no fim da resposta

③

1- Single-point of failure:
   - A consulta de índices é centralizada
2- Problemas de infração de copyright
3- Bottleneck de desempenho relacionado com a centralização de BD.

④

Após um pedido HTTP, o server mantem a ligação aberta, sem responder. Só responde qdo há eventos pl partilhar. Assim o server pode mandar respostas aos clientes sem estes fazerem requests.
(Método primitivo alternativo às websockets)

⑤

Sistema que permite que os programas armazenem e acedam a dados em ficheiros remotos.

Útil para:
- Partilha e armazenamento de dados
- Portabilidade entre OSs
- ets.

(6) Lightweight Directory Access Protocol
Serve para encontrar pessoas e recursos numa rede (directory service)

No fundo, é uma implementação lightweight do directory service X.500

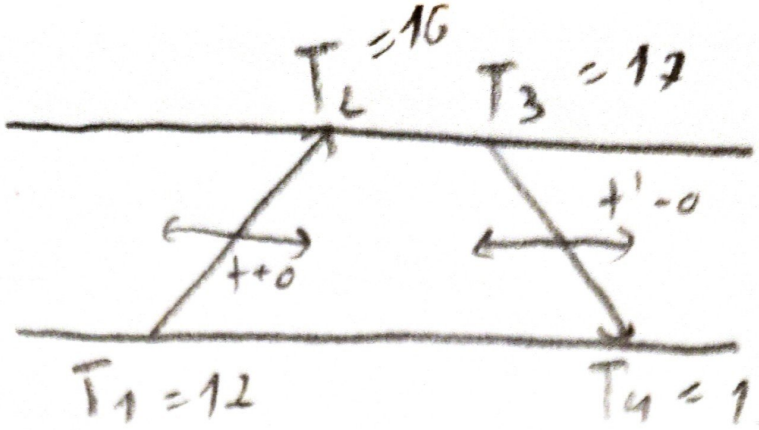
Benefícios:
- Seguro
- Estrutura flexível de atributos
- Transparência p/ o programador

(7)

Funcionalidades de Spring que evitam ser implementadas manualmente:

- Robustez e tratamento de falhas;
  - oferece semânticas fortes como at-most-once
  - filtra duplicados
  - etc
- Tratamento de concorrência e transações
- Separação da arquitetura em MVC
- Extração de dados
- Serviços RESTful
- Gestão do ciclo de vida dos componentes

⑧



$$o = \frac{(T_2 - T_1) + (T_4 - T_3)}{2} = \frac{4 + 2}{2} = 3$$

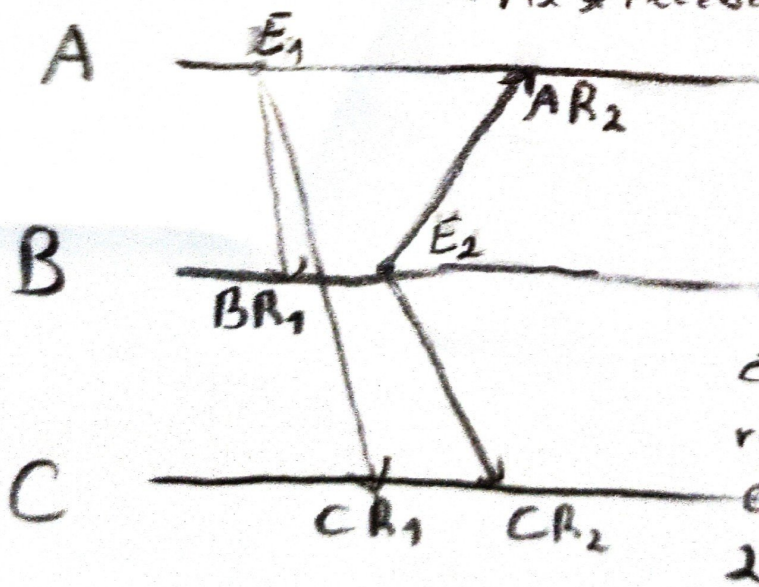
⑧

⑨ Não

Garante que eventos relacionados causalmente são processados pela ordem correta, Mas não impõe ordem única para eventos concorrentes ($x \| y$)

Contra-exemplo:

$E_x \equiv$ Enviar x
$PR_x \equiv$ Receber x no processo P



Suponhamos que

$E_1 \| E_2$

Com ordem causal é possível que C receba e processe 1 e depois 2 ou primeiro 2 e depois 1