Blockchain, Criptomoedas & Tecnologias Descentralizadas

Fundamentos de Segurança

Prof. Dr. Marcos A. Simplicio Jr. – mjunior@larc.usp.br
Escola Politécnica, Universidade de São Paulo

Objetivos

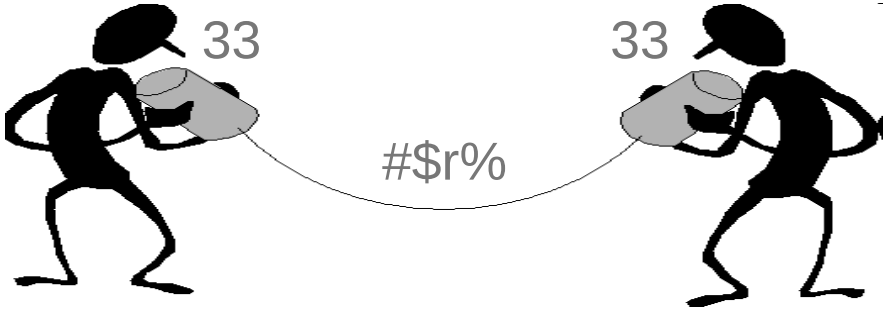
- Discutir aspectos fundamentais da segurança de sistemas computacionais.
  - Especificamente: *serviços de segurança*.
- **Serviços básicos de segurança**:
  - Disponibilidade
  - Confidencialidade
  - Integridade
  - Autenticidade
  - Irretratabilidade

Disponibilidade

- Garantia de que usuários legítimos *não sejam impedidos* indevidamente de acessarem as informações e os recursos do sistema.
- Serviço essencialmente extra-criptográfico (físico), e o mais arquitetural/empírico/ heurístico dentre os serviços básicos da segurança.
  - Exemplos de medidas: redundância, controle de acesso (físico), etc.

Confidencialidade

- **Confidencialidade de dados**: garantia de que qualquer *informação* armazenada num sistema de computação ou transmitida via rede seja *revelada somente a usuários devidamente autorizados*.
- Observação: *informação ≠ dado* (representação da informação).
  - Um dado pode estar acessível a qualquer entidade e mesmo assim não revelar a informação que ele contém.

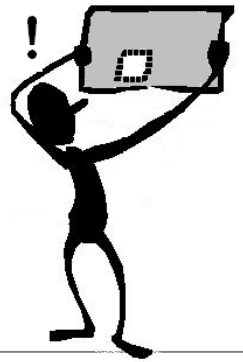


Confidencialidade

- **Privacidade**: Garantia de que os indivíduos *controlem* ou influenciem quais *informações sobre eles* podem ser coletadas e armazenadas e *por quem* e *para quem* tais informações podem ser reveladas
  - Tem relação direta com **confidencialidade de dados** (proteção da informação), mas também envolve políticas de **uso de dados** e **confidencialidade de identidades**.

Integridade

- Possibilidade de *verificar a consistência* da informação contida nos dados, *impedindo que seja alterada* indevidamente de *maneira imperceptível*.
- Detalhe: o serviço de integridade *não* garante que os dados não sejam alterados. A garantia efetiva é que, se os dados forem alterados sem autorização, a alteração será sempre *detectada*.



Autenticidade

- Garantia de que a *origem* ou o *originador* de uma mensagem seja *corretamente identificado* pelo seu destinatário.
- A *verificação de autenticidade* é necessária após todo processo de identificação, seja de um usuário para um sistema, de um sistema para o usuário ou de um sistema para outro sistema.

Irretratabilidade

- O *originador* e o *destinatário* das informações *não podem negar a sua transmissão, recepção ou posse*.
  - Obs.: ausência de serviço pode ser requisito de segurança (negação plausível)
- Relacionado a *assinaturas digitais*.
  - Conceito similar a assinaturas manuais, mas com garantias matemáticas...

Autenticidade vs. Irretratabilidade

- **Autenticidade**: destinatário **não consegue** necessariamente **provar para um terceiro** quem é o originador da mensagem
- Analogia com mundo real:
  - Os usuários Alice e Bob têm um mesmo “carimbo”
  - Se Alice recebe mensagem carimbada, então ela deve ter vindo de Bob
  - Porém, Alice não consegue provar para Carlos que foi Bob quem carimbou a mensagem (afinal, a própria Alice pode tê-lo feito!)

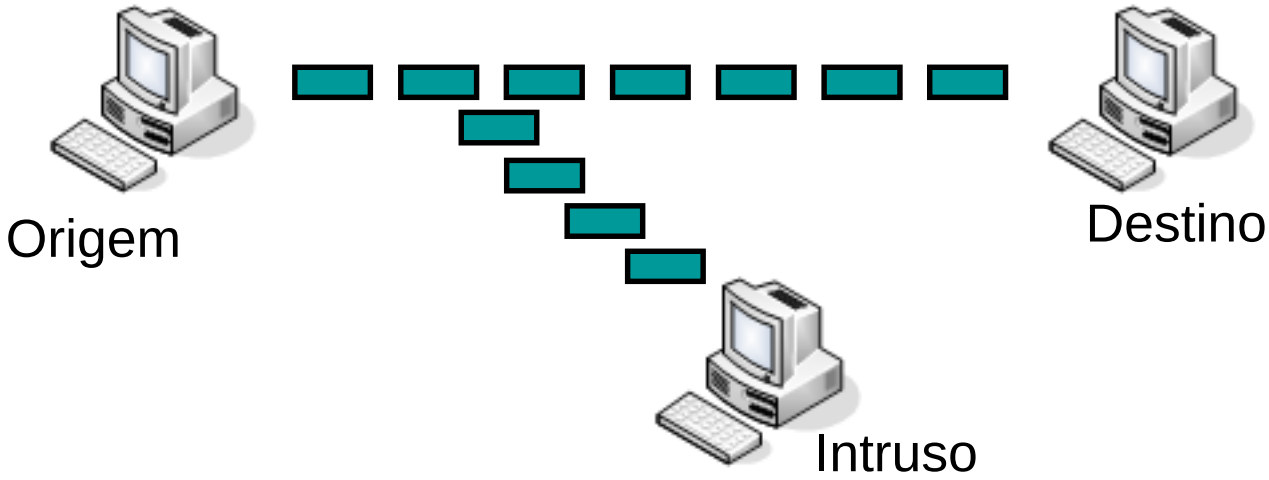
Autenticidade vs. Irretratabilidade

- **Irretratabilidade**: apenas originador da mensagem poderia tê-la gerado
- Analogia com mundo real:
  - Alice envia um documento a Bob usando sua **assinatura com firma reconhecida**
  - Bob pode apresentar o documento a Carlos, provando que Alice foi a originadora do documento
- Diferença importante: aparece nos algoritmos usados para prover tais serviços

Exemplos de Ataques vs. Serviços

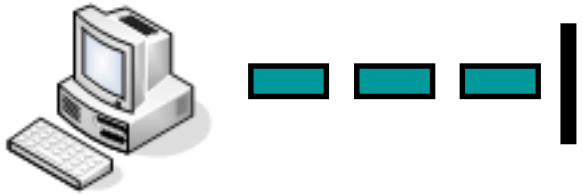
Interceptação

- Vazamento de informações (ex.: senhas)
- Para evitar que o intruso entenda o conteúdo das mensagens, é necessário cifrar os dados (***confidencialidade***)



Interrupção

- Dados nunca chegam ao destino
  - Ex.: “derrubar um site”



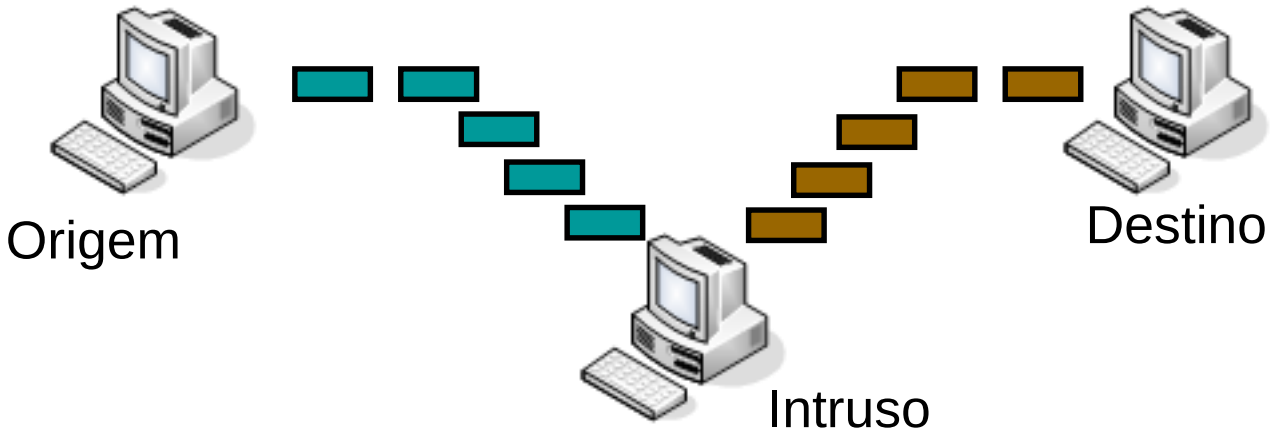
Origem

Destino

- É necessária a segurança física dos recursos de processamento e de comunicação de dados! (***disponibilidade***)

Modificação

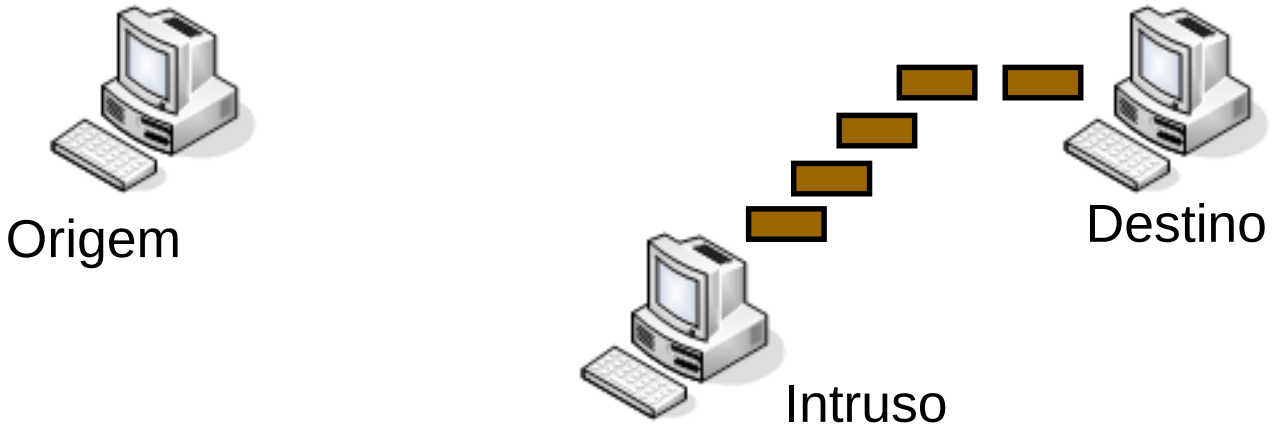
- Informações Corrompidas/falsas
  - Ex.: alterar destino de um pagamento bancário



- Para evitar tal ataque, é preciso garantir a ***integridade*** e a ***autenticidade*** dos dados

Fabricação

- Mensagens criadas por atacante
  - Ex.: gerar uma ordem de pagamento falsa



- Para evitar este tipo de ataque é preciso garantir a ***autenticidade*** dos dados

Criptografia: princípios básicos

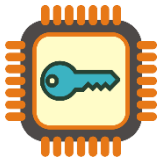
Para que serve criptografia?

- Serviços básicos da segurança:
  - Confidencialidade
  - Integridade
  - Autenticidade
  - Irretratabilidade
- **Não** é possível implementar disponibilidade
  - Mas é exatamente nesse quesito que **sistemas descentralizados** são excelentes!

➔ P2P+Criptografia: um par perfeito ➔

Algoritmos Criptográficos

- Existem dois tipos básicos de algoritmos criptográficos
  - **Simétricos:** uma mesma informação secreta (chave) é conhecida apenas por remetente e destinatário, mas não por atacantes
    - Esta categoria também inclui algoritmos auxiliares, que não usam chaves



  - **Assimétricos:** usam duas chaves distintas, porém relacionadas matematicamente. Uma chave é tornada pública (conhecida inclusive por atacantes), e a outra é conhecida apenas pelo seu dono.

- Se usada corretamente**, criptografia costuma ser a parte mais forte** de sistemas computacionais

Criptografia moderna

- **Quebra** de criptografia: **violação de seu serviço de segurança**
  - Ex.: contra algoritmos baseados em chave secreta simétrica, descobrir a chave secreta
  - Ex.: contra algoritmos baseados em chave secreta assimétrica, resolver o problema computacional que permite calcular a chave privada a partir da chave pública
- Criptografia moderna: algoritmos **computacionalmente inviáveis de se "quebrar"**
  - Mesmo com as técnicas mais modernas conhecidas (criptoanálise)
  - Mesmo com uma quantidade gigantesca de recursos computacionais disponíveis atualmente ou em futuro distante

- Exemplo de ataque genérico: **força bruta**
  - Capaz de **quebrar qualquer** sistema criptográfico baseado em chaves secretas: basta testar todas as chaves possíveis!
  - ➔ Qual seria o custo de executar tal ataque...?

Ataque de força bruta: exemplo

- Exemplo de complexidade vs. recurso:
  - Suponha que ataque a algoritmo envolva testar chaves de 128 bits: 2^{128} possibilidades
    - Esse é o nível de segurança mais usado atualmente
  - Suponha também que estejam disponíveis 1 milhão (10^6) de super-computadores, cada um capaz de realizar 1 peta (10^{15}) testes por segundo
  - Ainda assim seriam necessários ~2^{58} segundos para recuperar a chave...
    - Idade estimada do universo: ~2^{59} segundos

$$\frac{\#\text{testes}}{\#\text{computadores x capacidade}} = \frac{2^{128}}{10^6 \text{ x } 10^{15}} \approx \frac{2^{128}}{2^{20} \text{ x } 2^{50}} = 2^{58} \text{ segundos}$$

Blockchain, Criptomoedas & Tecnologias Descentralizadas

Fundamentos de Segurança

Prof. Dr. Marcos A. Simplicio Jr. – mjunior@larc.usp.br
Escola Politécnica, Universidade de São Paulo

Apoio financeiro: d24n.org <https://d24n.org>
ripple impact | university blockchain research initiative

USP

Referências

- W. Stallings, L. Brown “Computer Security Principles and Practice – 2nd/3rd/4th edition”. Prentice-Hall, ISBN: 0-13-277506-9. 2011/2015/2018.
  - Em português: W. Stallings, L. Brown. “Segurança de Computadores - Princípios e Práticas” (2ª Ed), Elsevier, 2014
- W. Stallings: “Cryptography and Network Security” (6th/7th Ed.), Prentice-Hall 2013/2016.
  - Em português: W. Stallings: “Criptografia e Segurança de Redes” (6ª Ed.), Pearson-Prentice-Hall (2014).
- M. Goodrich, R. Tamassia, “Introdução à Segurança de Computadores”. Bookman, 2013
- S. Wykes. Criptografia Essencial: A Jornada do Criptógrafo, 1a ed. Elsevier, 2016.