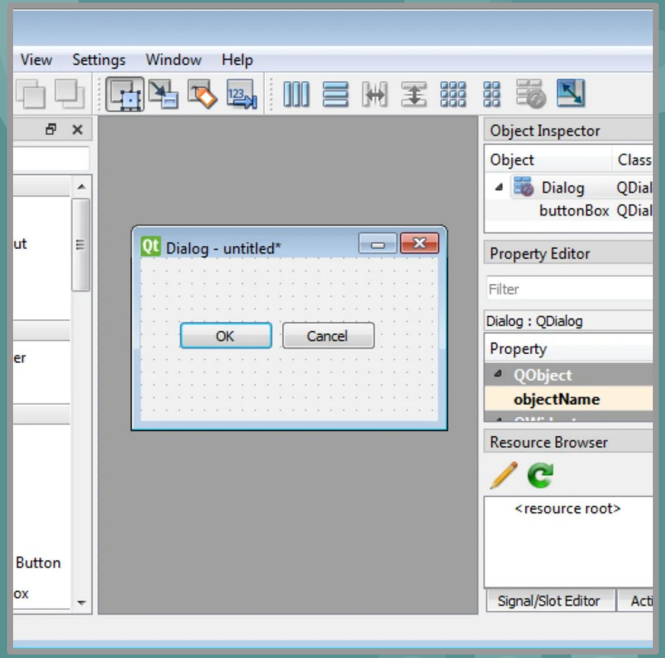
Chat em Python utilizando GUI, conexão tcp e criptografia RSA

Alunos: Daniel Carvalho e Augusto Alves

PyQt GUI (Graphical User Interface)

- O PyQt é um framework multiplataforma criado para desenvolver aplicações desktop.

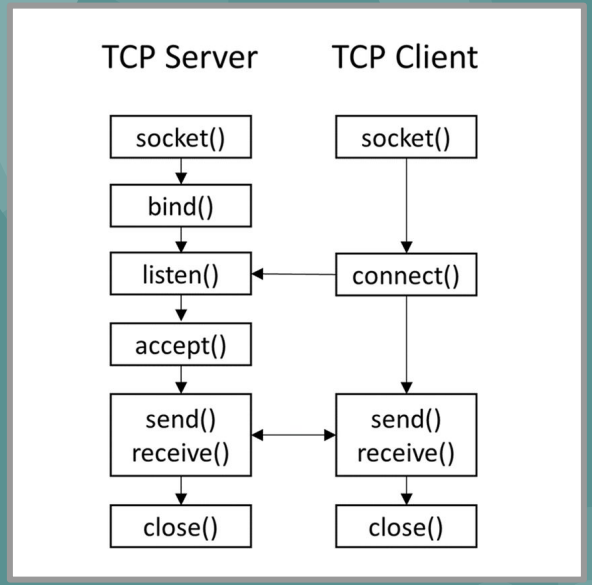
- O software desenvolvido possui um ambiente de interação no qual o usuário insere seu login (com opção de criar um novo usuário), a porta de envio e a porta de recebimento de mensagens, e o endereço de IP do outro usuário. Em seguida entra na tela principal do chat



Conexão TCP

- Para estabelecer uma conexão segura e garantir a totalidade da entrega de dados foi utilizado o protocolo TCP ao invés de UDP.

- Foi usada a biblioteca ***socket*** do Python e foi criado um esquema de cliente/servidor simultâneo para o chat, pois assim qualquer uma conta logada no chat pode iniciar uma conversa sem precisar previamente definir quem será o cliente e quem será o servidor.



Banco de dados descentralizado de usuários

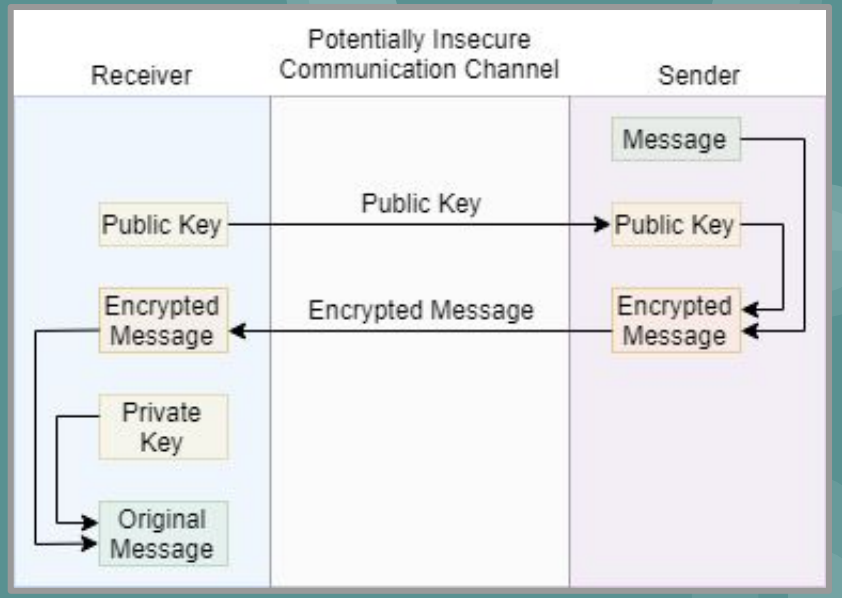
- O sistema de login de usuário foi criado baseado no sistema de carteiras descentralizadas do Bitcoin, no qual a chave pública é o ID de um usuário, que é criado a partir de uma chave privada escolhida aleatoriamente **de maneira segura**.

- Desse modo, como cada chave pública será única, basta associá-la com o usuário real que compartilhou essa sua chave pública, e como consequência pode se aproveitar essa mesma chave para autenticar, validar e criptografar as mensagens entre cada usuário graças a como funciona o algoritmo RSA.

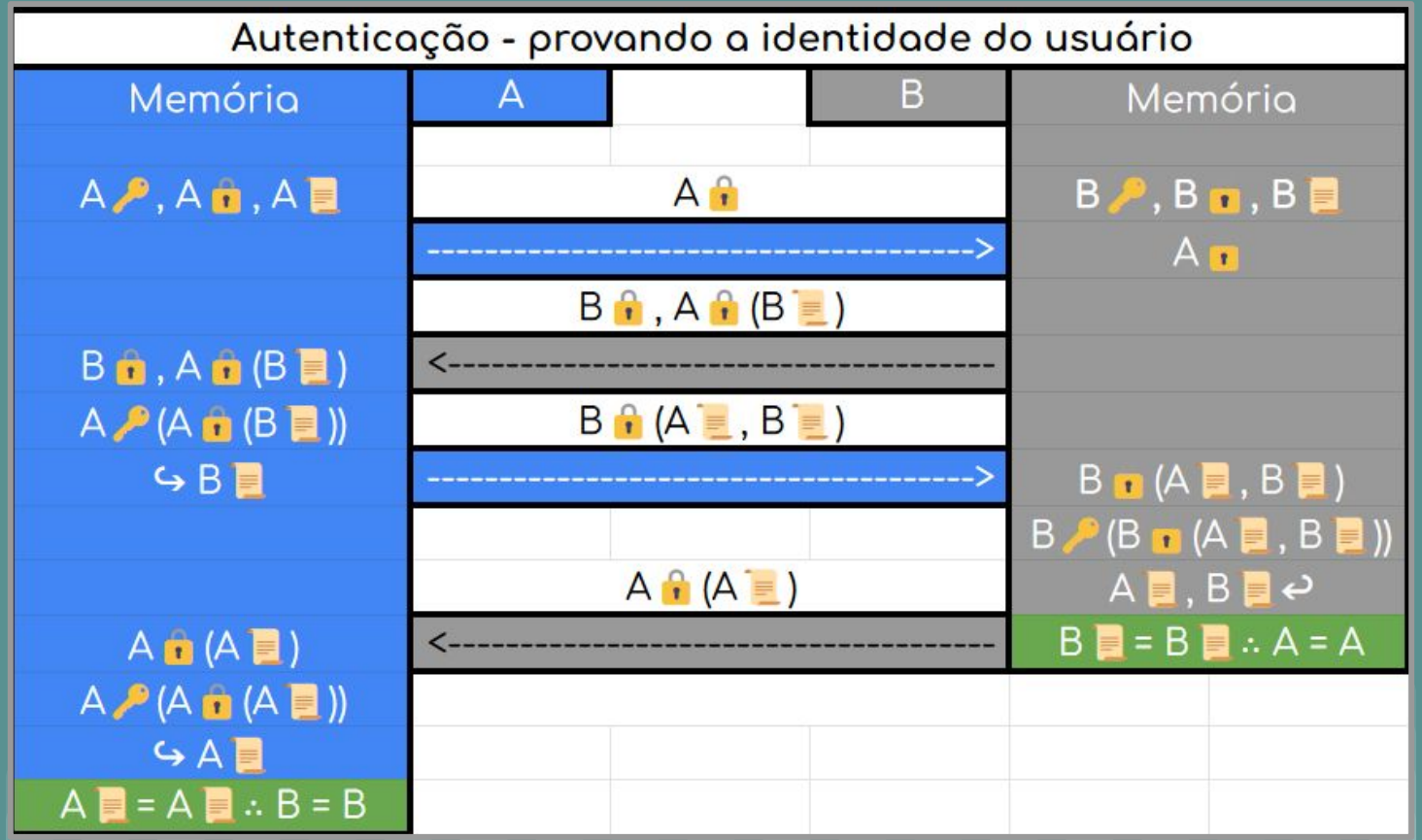
- A prova de que cada ID é único é baseado justamente na ideia do RSA: é possível criar uma chave pública a partir de uma chave privada, mas não o contrário. Logo, não é possível fraudar a autenticidade de outro usuário, pois o fraudador não terá a chave privada dele, cancelando a comunicação no momento do teste de autenticidade.

Criptografia RSA

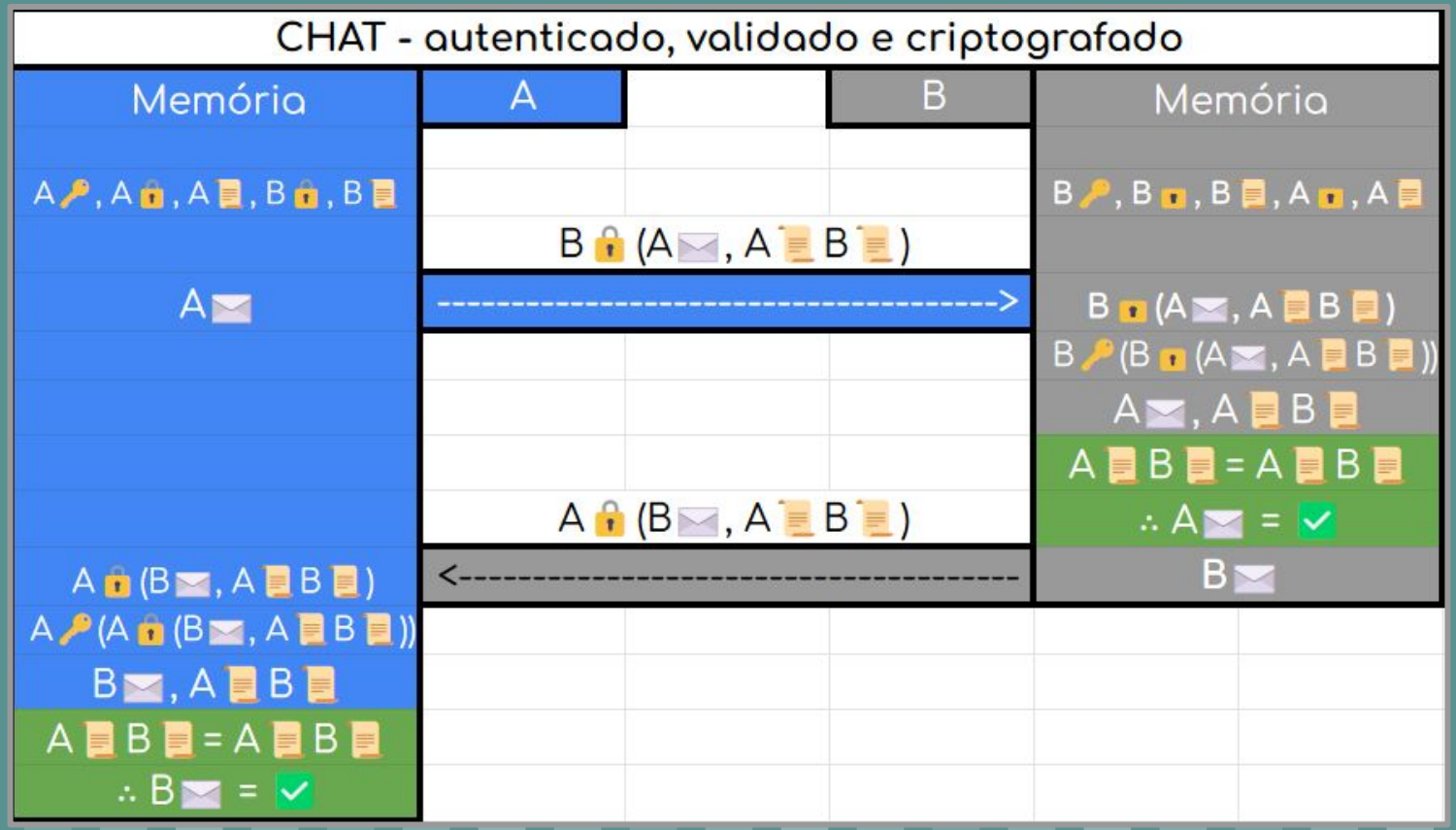
- Esquema de criptografia assimétrica com base em chaves públicas e privadas.
- A informação é criptografada com a chave pública do destinatário que irá receber a informação, e só pode ser descriptografada com a chave privada desse mesmo destinatário.



Esquema de funcionamento - Autenticação

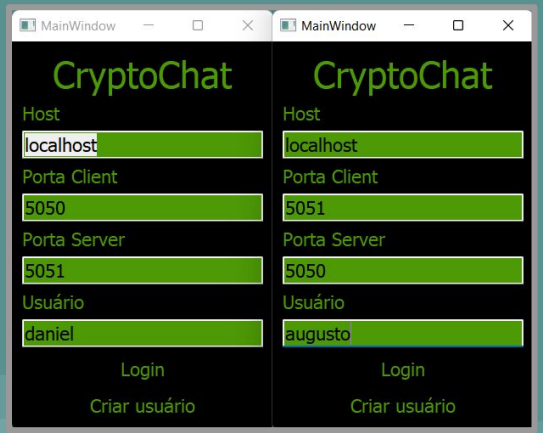


Esquema de funcionamento - Chat

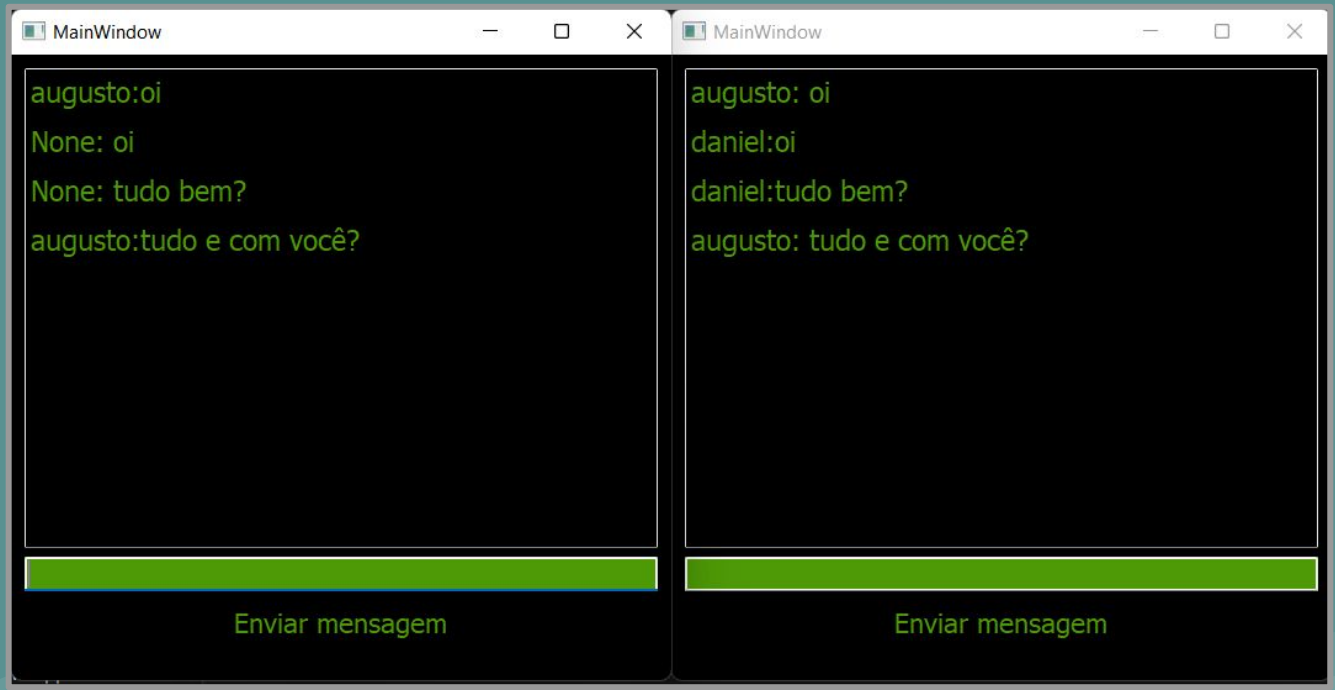


Chat em funcionamento

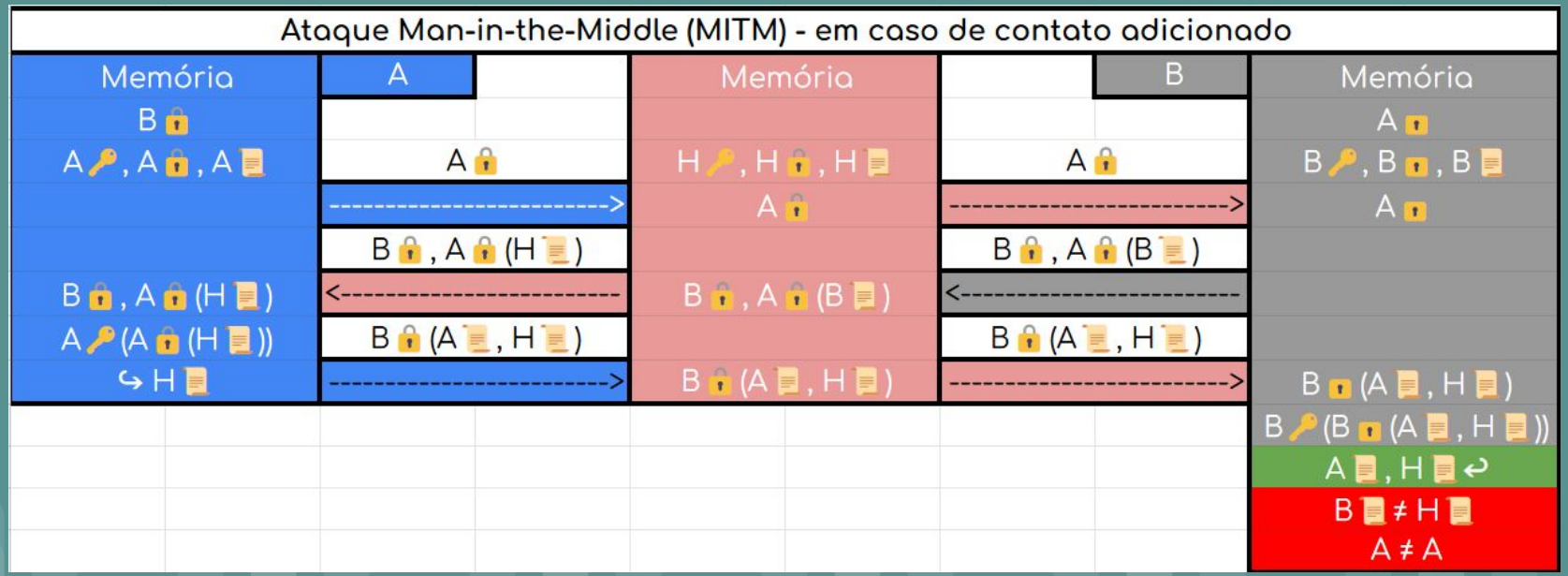
Configuração da conexão e login:



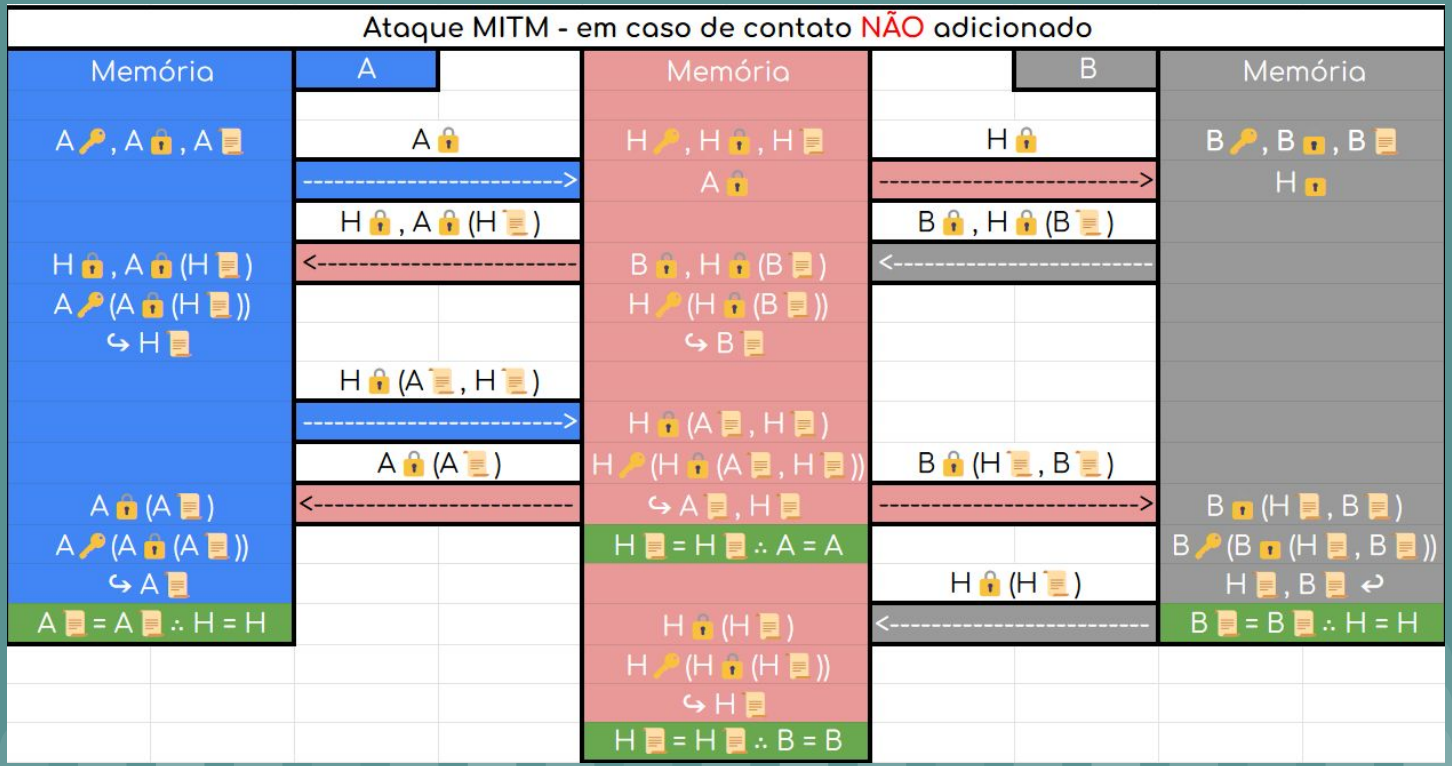
Chat em tempo real (daniel adicionou augusto, mas augusto não adicionou daniel):



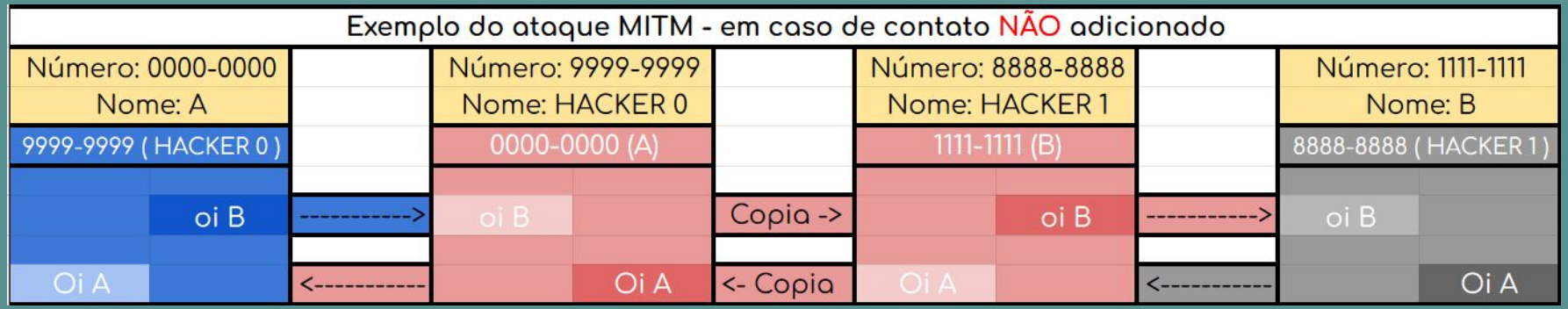
Simulação de ataque MITM teórico



Simulação de ataque MITM teórico



Simulação de ataque MITM na prática



Esse tipo de ataque é praticamente inevitável pois pode acontecer até em aplicativos de mensagem maiores como o WhatsApp, pois o hacker funciona como um nodo de repasse de mensagens entre os usuários, podendo interferir apenas quando necessário. Nesse ataque se cria a “falsa” sensação de autenticidade de usuários, e de fato a conversa pode ser “autêntica”, porém o hacker tem total controle da conversa quando lhe for conveniente.

O único modo de garantir que esse ataque não ocorra é sempre que se adicionar um contato novo, garantir (por outro meio externo, ex: pessoalmente) que aquela chave pública (ou número de telefone no caso do WhatsApp) foi cedida diretamente pelo próprio usuário. Então, sempre que um contato não adicionado iniciar uma conversa, para garantir sua autenticidade é necessário usar outro meio externo que valide sua autenticidade.