Chat P2P Com autenticação

Segurança da informação

Alunos:

Leandro Schwab Dias Carneiro

Andre Felipe Tavares da Mota Monteiro

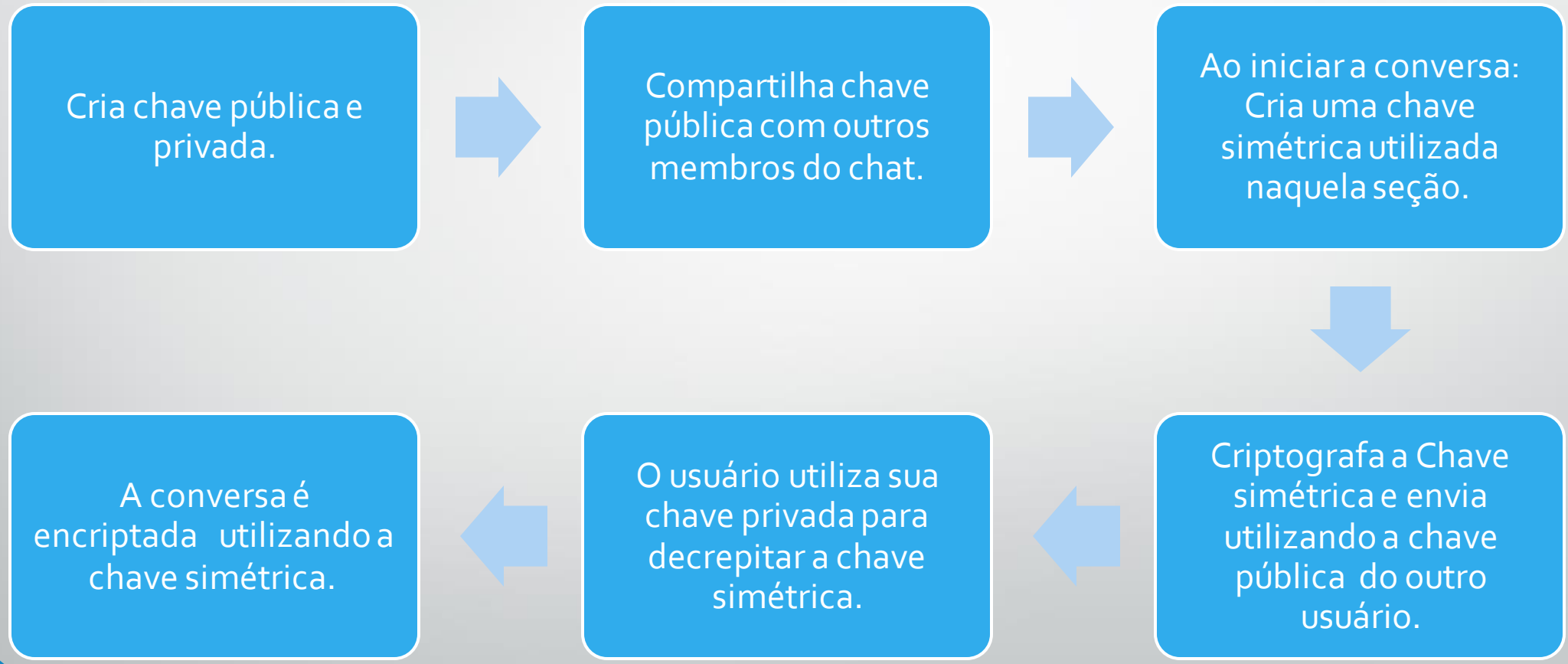
Relembrando

Desenvolver um Chat com criptografia.

Utilizamos a biblioteca Pycrypto

- Algoritimo de encriptação para chave assimétrica: RSA-1024
- Algoritimo de encriptação para chave simétrica: AES-128

Processo


Cria chave pública e privada.
Compartilha chave pública com outros membros do chat.
Ao iniciar a conversa: Cria uma chave simétrica utilizada naquela seção.
Criptografa a Chave simétrica e envia utilizando a chave pública do outro usuário.
O usuário utiliza sua chave privada para decrepitar a chave simétrica.
A conversa é encriptada utilizando a chave simétrica.

```
def createMyKeys(VarData):
    if os.path.exists('Data/' + str(VarData['porta']) + '/private.pem'):
        print 'Carregando chaves!'
        prv_file = open('Data/' + str(VarData['porta']) + "/private.pem", "r")
        pub_file = open('Data/' + str(VarData['porta']) + "/public.pem", "r")
        private_key = RSA.importKey(prv_file.read())
        public_key = RSA.importKey(pub_file.read())
        VarData['myprivatekey'] = private_key
        VarData['mypublickey'] = public_key

    else:
        print 'Criando chave privada!'
        private_key = RSA.generate(1024)
        public_key = private_key.publickey()
        VarData['myprivatekey'] = private_key
        VarData['mypublickey'] = public_key
        print VarData['mypublickey']
        prv_file = open('Data/' + str(VarData['porta']) + "/private.pem", "w")
        prv_file.write("{}".format(private_key.exportKey()))
        pub_file = open('Data/' + str(VarData['porta']) + "/public.pem", "w")
        pub_file.write("{}".format(public_key.exportKey()))
```

```
def createSession_key(VarData, userValor):
    print "createSession_key: started"
    session_key = Random.new().read(16)
    Clientepub_file = open('Data/' + str(VarData['porta']) + "/" + str(userValor['porta']) + "public.pem", "r")
    Clientpublic_key = RSA.importKey(Clientepub_file.read())
    cipher_rsa = PKCS1_OAEP.new(Clientpublic_key)
    enc_session_key = cipher_rsa.encrypt(session_key)
    mensagem = "ChatAES-+,+-" + str(VarData['porta']) + "-+,+-" + enc_session_key + "-+;+-"

    userValor['ChatAESKey'] = session_key

    connS = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  # qwl2IPv4,tipo de socket
    connS.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    connS.connect((userValor['ip'], int(userValor['porta'])))  # Abre uma conexão com IP e porta especificados
    connS.sendall(mensagem)
    connS.close()
```