Sessão 6: Autenticação, autorização e certificação digital

1) Uso de criptografia simétrica em arquivos

Esta atividade será realizada nas máquinas virtuais *FWGW1-G* e *LinServer-G*.

1. Na máquina *FWGW1-G*, descubra quais cifras simétricas são suportadas pelo programa `gpg` (*GNU Privacy Guard*).
2. Crie um arquivo `teste.txt` com qualquer conteúdo. Criptografe-o usando a cifra simétrica AES256, com senha `rnpesr`. Em seguida, copie o arquivo cifrado resultante para o diretório *home* do usuário `aluno`, na máquina *LinServer-G*, usando o comando `scp`.
3. Na máquina *LinServer-G*, tente descriptografar o arquivo copiado. Seu conteúdo permanece o mesmo?

2) Uso de criptografia assimétrica em arquivos

Esta atividade será realizada nas máquinas virtuais *FWGW1-G* e *LinServer-G*.

1. Na máquina *FWGW1-G*, como usuário `root`, instale o pacote `rng-tools` e rode o comando `rngd -r /dev/urandom`:
2. Agora, como usuário `aluno`, descubra quais cifras assimétricas são suportadas pelo programa `gpg` (*GNU Privacy Guard*).
3. Vamos fazer um exercício de criptografia usando chaves assimétricas entre dois usuários fictícios, Alice (operando na máquina *FWGW1-G*) e Bobby (operando na máquina *LinServer-G*). Vamos começar por Alice — gere um par de chaves assimétricas RSA padrão, com 4096 bits e sem data de expiração para ela, usando o programa `gpg` com a opção `--full-generate-key`. O e-mail de Alice será alice@seg12.esr.rnp.br , e a senha de acesso à chave será `rnpesr123`.
4. Na máquina *LinServer-G*, como usuário `root`, instale o pacote `rng-tools` e rode o comando `rngd -r /dev/urandom`:
5. Como usuário `aluno`, gere a chave de Bobby na máquina *LinServer-G*. Repita o procedimento do passo (2), alterando o nome de usuário para Bobby e o email para bobby@seg12.esr.rnp.br .
6. Temos que exportar as chaves públicas de ambos os usuários, copiá-las para a máquina remota, e importá-las. Comece pela chave de Alice, exportando-a em formato *ASCII armored*; em seguida, copie-a para a máquina *LinServer-G* usando o `scp`, importe-a usando `gpg --import` e assine a chave.
7. Faça o procedimento reverso, exportando/copiando/importando e assinando a chave de Bobby na máquina de Alice. Lembre-se que o `ssh` para a máquina *FWGW1-G* é permitido apenas a partir da Intranet, então pode ser mais interessante iniciar o procedimento de cópia a partir do firewall, e não da máquina *LinServer-G*.
8. Agora, vamos fazer o teste de criptografia assimétrica propriamente dito. Na máquina *FWGW1-*

G, verifique que as chaves estão de fato disponíveis. Em seguida, criptografe um documento de texto com conteúdo qualquer com a chave pública de Bobby, envie para a máquina *LinServer-G*, e tente decriptá-lo usando a chave privada de Bobby.

9. Vamos agora testar a assinatura digital de arquivos. Começando a partir da máquina *LinServer-G*, crie um arquivo texto com conteúdo qualquer. Assine-o com a chave privada de Bobby, e copie o arquivo para a máquina *FWGW1-G*. Finalmente, verifique a assinatura usando o *keyring* de Alice.
10. Finalmente, vamos "juntar tudo". Da máquina *FWGW1-G*, crie um arquivo texto com conteúdo qualquer e (1) assine-o com a **chave privada de Alice**, e (2) criptografe-o com a **chave pública de Bobby**. Copie o arquivo para a máquina *LinServer-G*, decripte-o e verifique sua assinatura.

3) Uso de criptografia assimétrica em e-mails

Esta atividade será realizada em sua máquina física.

Vamos agora testar o procedimento de criptografia assimétrica usado na atividade (2) em um cenário mais prático: no envio e recebimento de e-mails.

1. Crie uma conta de e-mail gratuita no serviço GMail, do Google.
2. Em sua máquina física, instale o programa *gpg4win* (que pode ser baixado em https://www.gpg4win.org/download.html). Durante a instalação, aceite todas as opções padrão, e desmarque a caixa *Executar Kleopatra* ao final do processo de instalação.
3. Em sua máquina física, instale o cliente de e-mail *Mozilla Thunderbird* (que pode ser baixado em https://www.thunderbird.net/pt-BR/thunderbird/all/). Durante a instalação, aceite todas as opções padrão.
4. Ao abrir o Thunderbird, adicione a conta de e-mail criada no passo (1), como mostra a imagem a seguir:

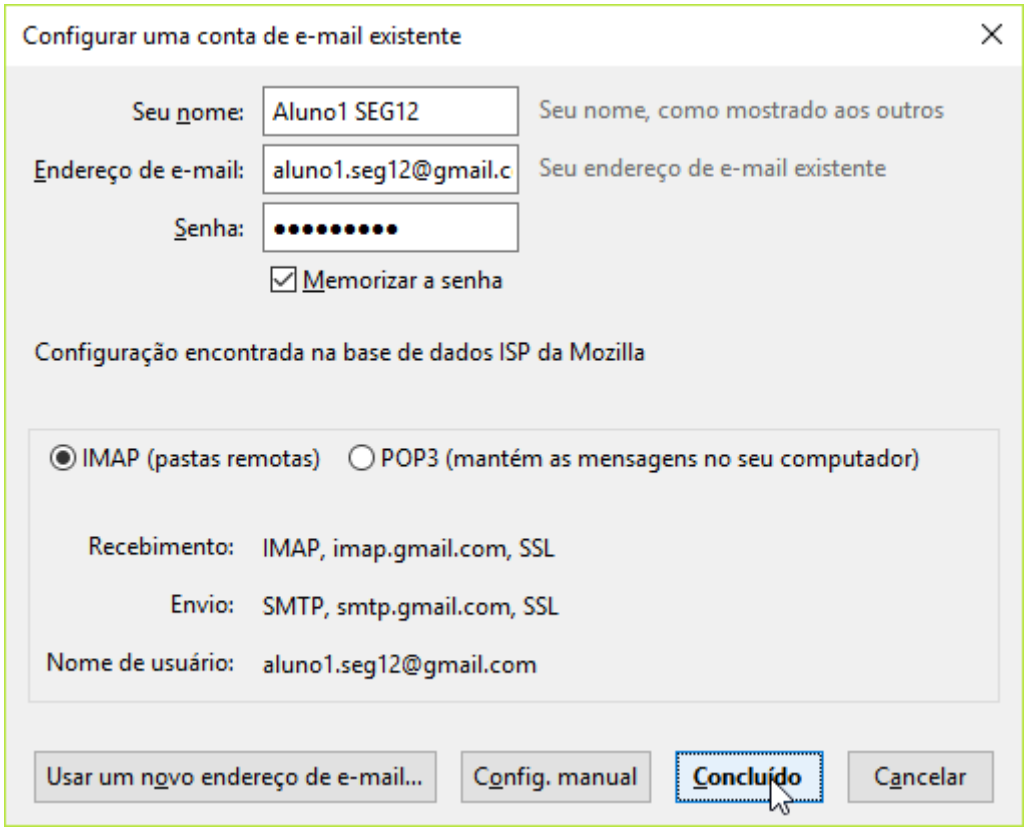

Figura 1. Adicionando uma conta de e-mail ao Thunderbird

Durante o processo de criação de conta, o *Thunderbird* irá abrir uma janela para autenticação no GMail, solicitando autorização para integração. Digite a senha da conta definida no passo (1) e garanta as permissões solicitadas.

5. No Thunderbird, navegue no menu localizado no canto superior direito. Clique em *Extensões* > *Extensões*. No canto superior da janela, pesquise por `enigmail` e pressione ENTER. O primeiro resultado, a extensão *Enigmail*, é o que queremos: clique no botão *Adicionar ao Thunderbird* > *Instalar agora*.

   Após a instalação do *Enigmail*, reinicie o *Thunderbird* (feche e reabra o programa).

6. Desde a versão 2.0.0 do `Enigmail`, lançada em março de 2018, o modo padrão de operação é o *Enigmail/PeP*. O *PeP* (*pretty Easy privacy* cujo website é https://www.pep.security/) é uma implementação de segurança para e-mails com o objetivo expresso de ser simples e de baixa configuração. Para o nosso cenário, isso significa:

   - Geração automática de pares de chaves assimétricas
   - Distribuição automática de chaves públicas via anexo ou *upload* para servidores de chaves (*keyservers*)
   - Criptografia e assinatura automática de mensagens

7. Vamos testar esses conceitos. Envie uma mensagem para o seu colega usando o Thunderbird. Caso o *Enigmail/PeP* esteja funcionando corretamente, o botão *Habilitar a Proteção* deverá estar marcado no centro da tela:

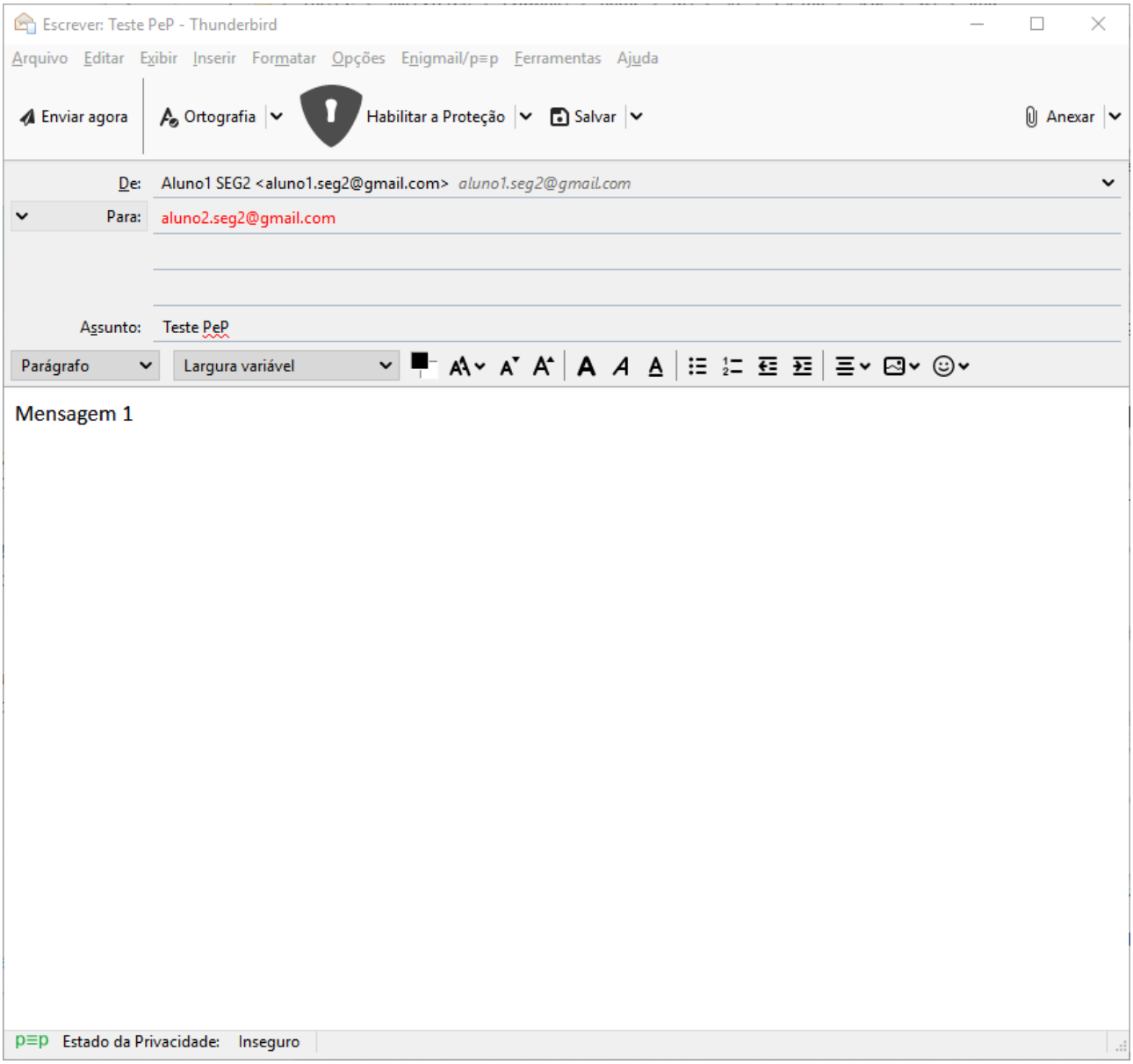

Figura 2. PeP habilitado no Thunderbird

Logo abaixo, você visualizará o estado da privacidade como "Inseguro". Isso se deve ao fato de que você e seu colega ainda não fizeram a troca de chaves. O *Enigmail/PeP* irá se encarregar de anexar sua chave pública automaticamente à mensagem. Envie o email.

8. Na máquina do seu colega, a mensagem deverá ser recebida normalmente. Note que a mensagem está legível — não há criptografia ainda, apenas assinatura. Clique no triângulo amarelo na lateral direita para observas as características de confiança da mensagem:

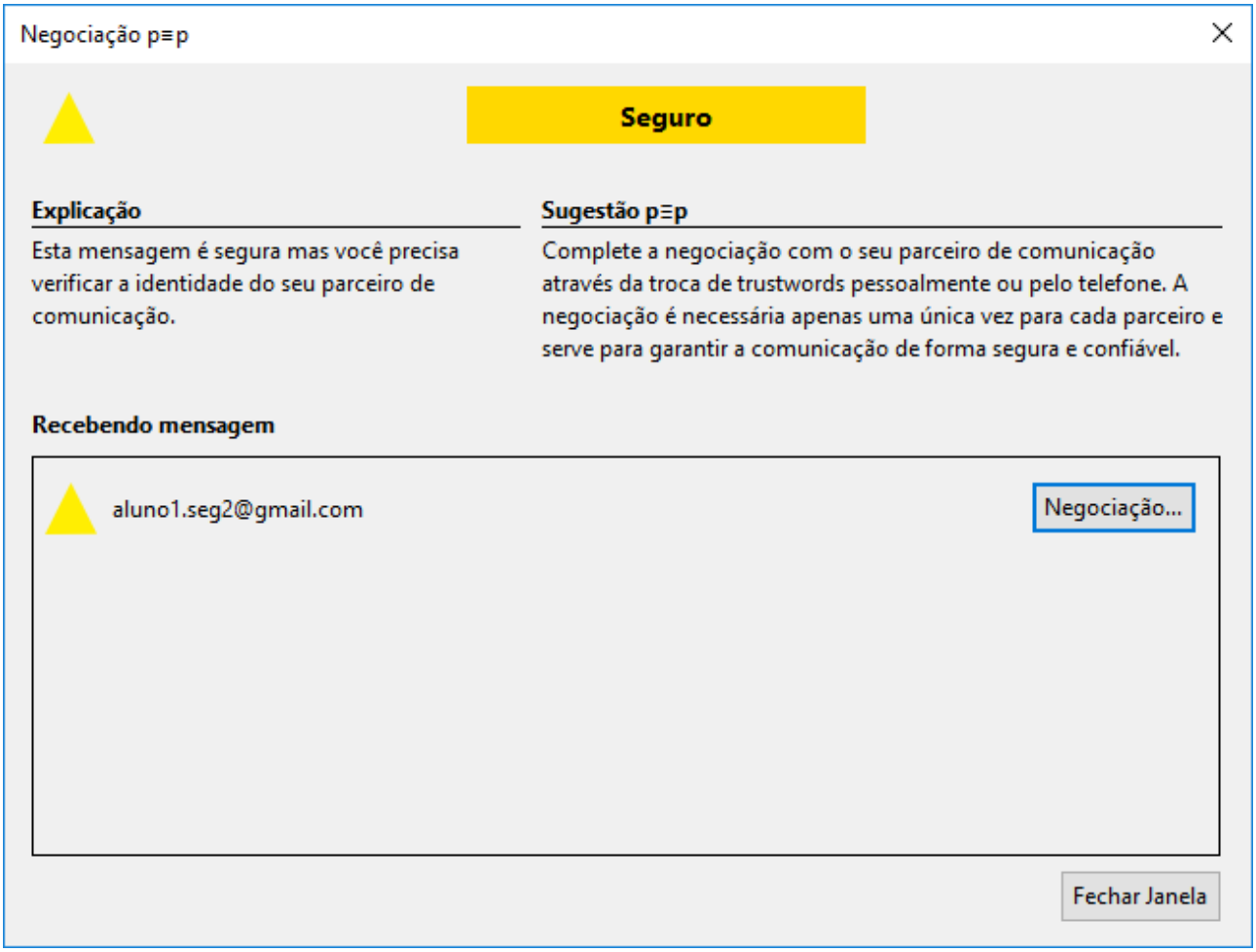

Figura 3. Mensagem segura, não confiável

9. Clique no botão *Negociação...* para confirmar a veracidade da chave. Você verá a tela a seguir:

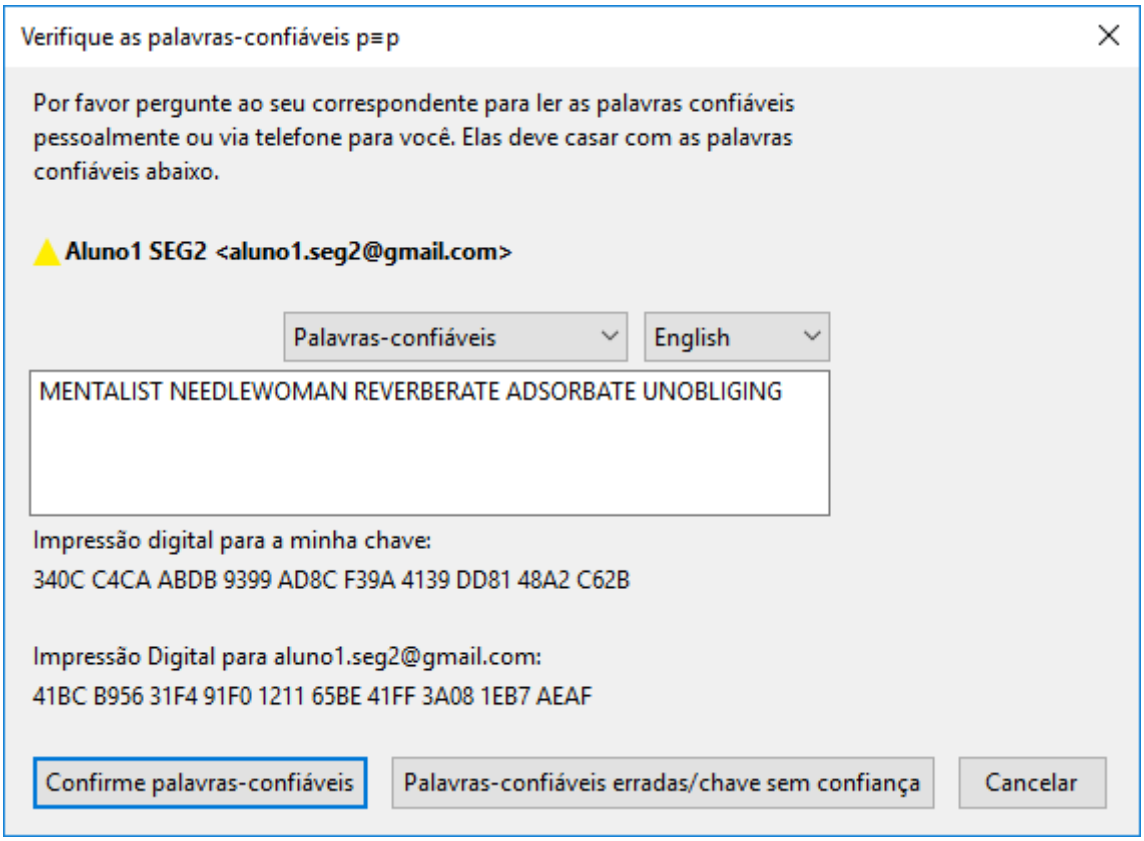

Figura 4. Verificação de palavras confiáveis

Serão mostradas cinco palavras confiáveis, que devem ser confirmadas com o seu colega. Se corretas, significa que a chave pública anexada à mensagem original é, de fato, correspondente à chave privada sob posso do seu colega. Clique no botão *Confirme palavras-confiáveis* para autenticar a chave.

10. Logo após, você verá a tela a seguir:

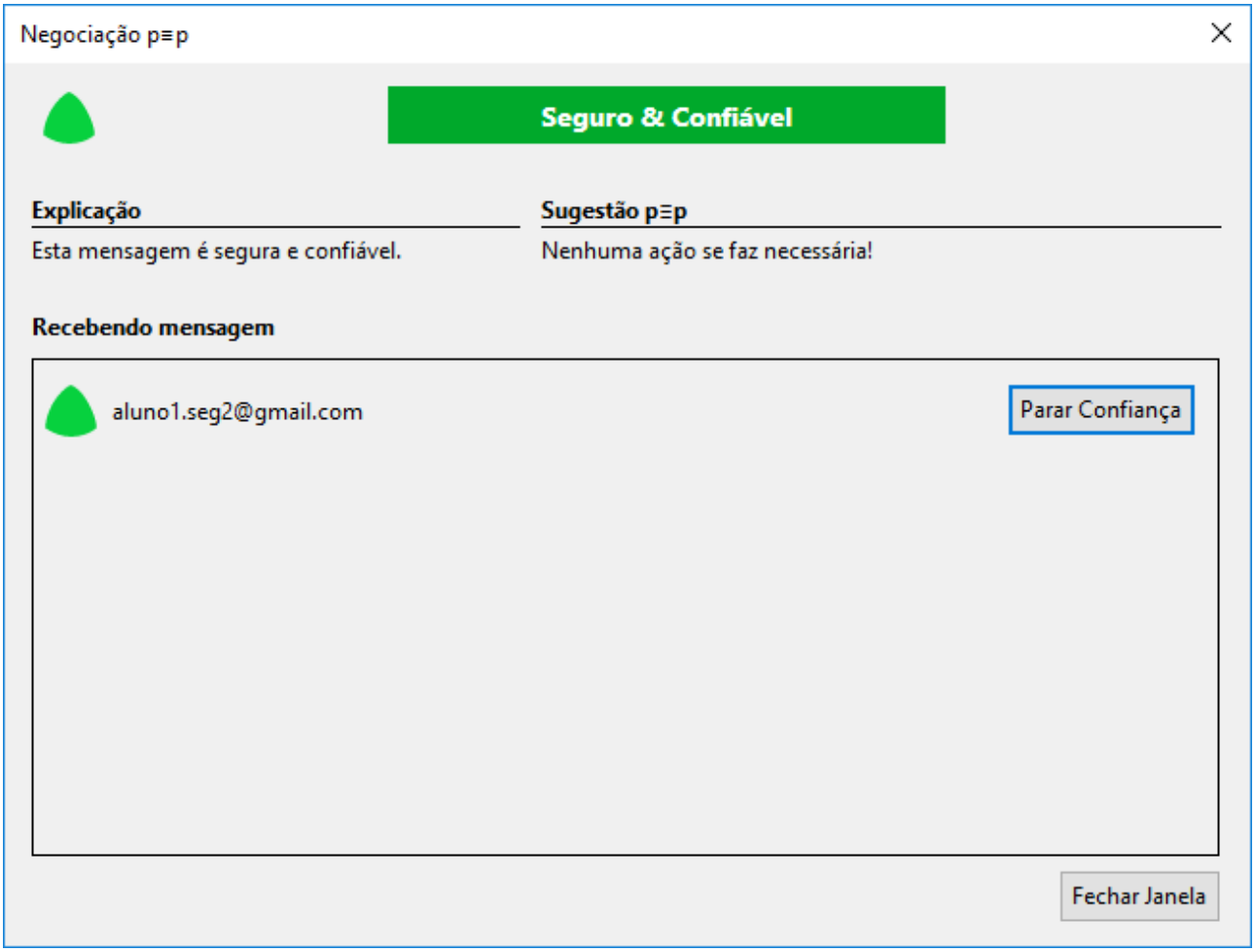

Figura 5. Mensagem segura e confiável

A partir desse momento, a chave pública do seu colega é conhecida pelo *Enigmail/PeP*, e é possível enviar mensagens criptografadas para ele e verificar mensagens assinadas. Envie um email-resposta para a mensagem original, e repita os passos (8) e (9) para autenticar a chave no sentido oposto. A partir desse momento, será possível trocar mensagens seguras entre as duas partes.

4) Criptografia de partições e volumes

Esta atividade será realizada em sua máquina física.

1. Instale o *VeraCrypt* (que pode ser baixado em https://www.veracrypt.fr/en/Downloads.html) em sua máquina física. Durante a instalação, aceite todas as opções padrão.
2. O VeraCrypt pode criptografar partições inteiras ou apenas criar um contêiner seguro. Com isso, podemos gravar arquivos sigilosos no contêiner e transportá-lo através de mídia física ou meio não confiável de forma bastante conveniente. Na tela principal do VeraCrypt, clique em *Create*

Volume.

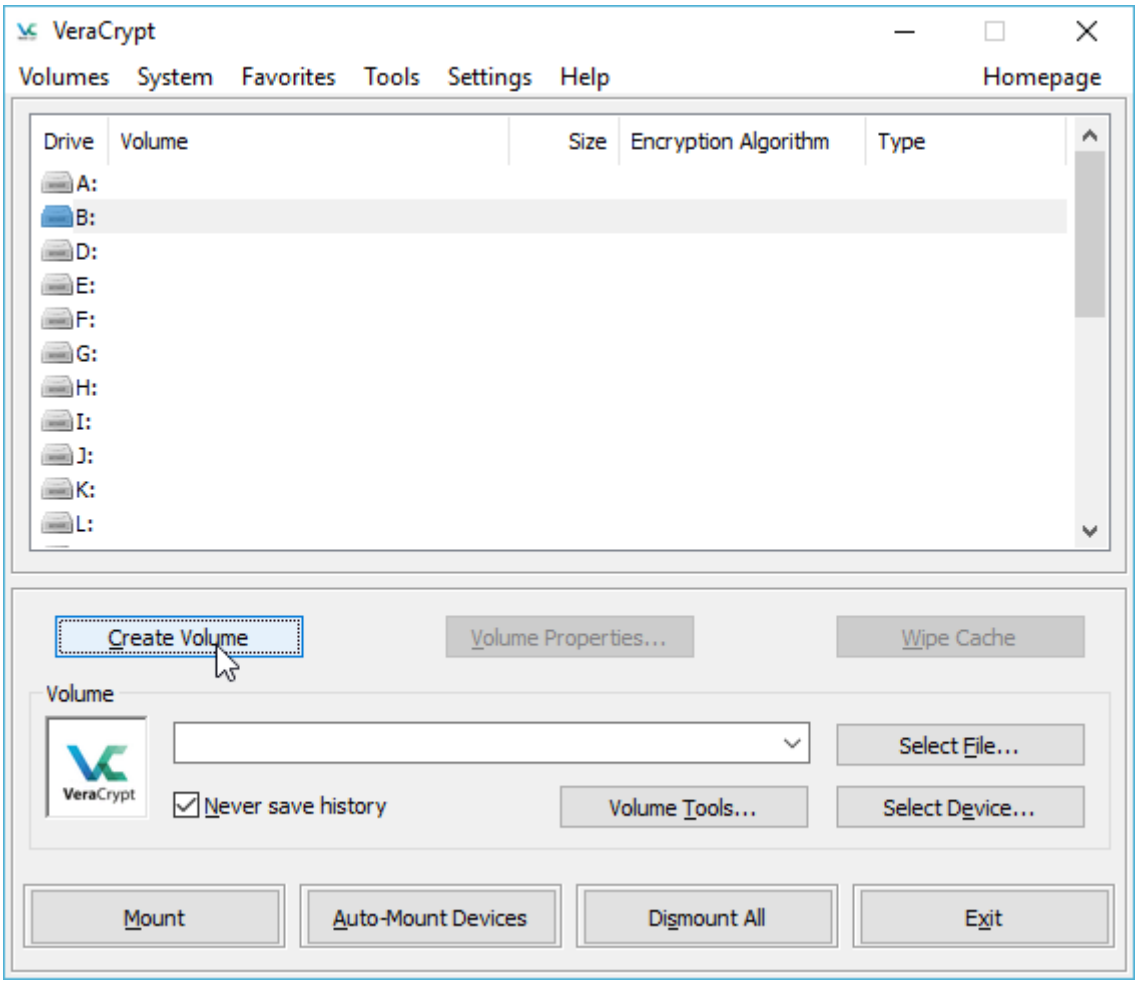

Figura 6. Criação de volumes no VeraCrypt, parte 1

3. Na tela seguinte, mantenha marcada a opção *Create an encrypted file container* e clique em *Next*.

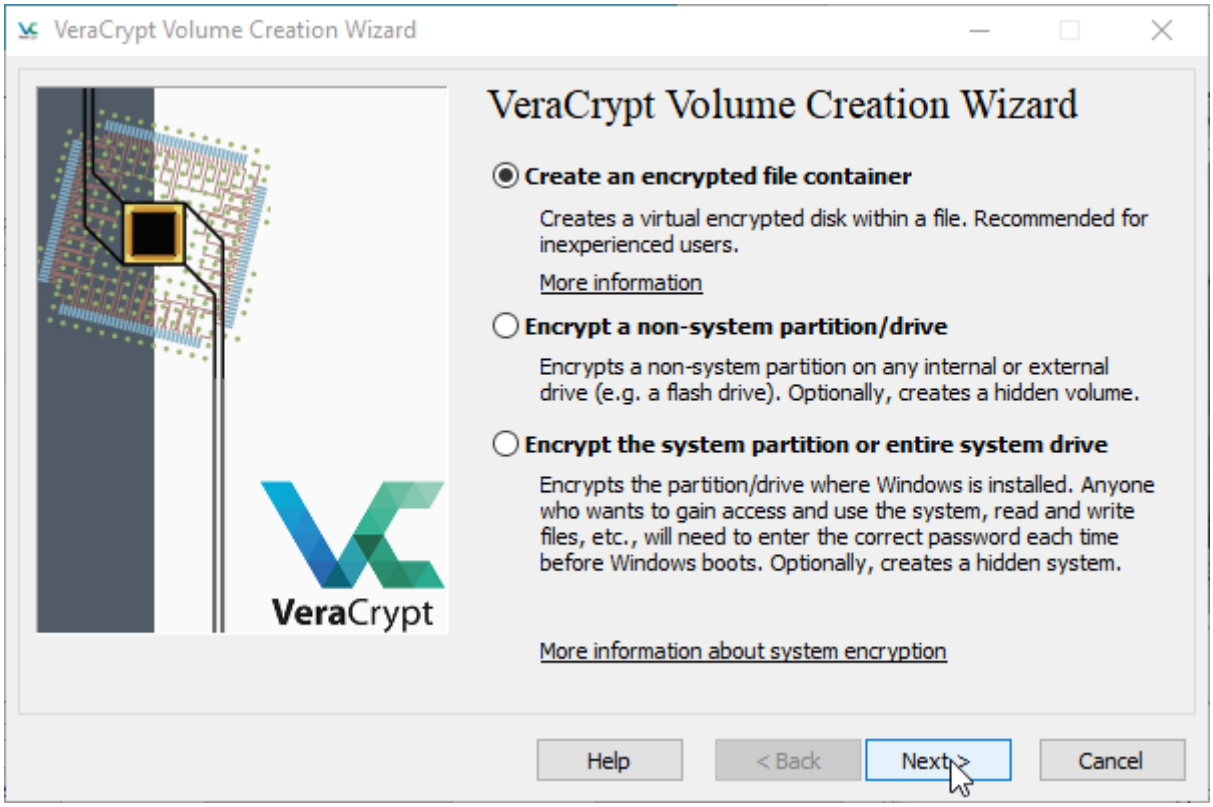

Figura 7. Criação de volumes no VeraCrypt, parte 2

4. Na tela subsequente, mantenha marcada a opção *Standard VeraCrypt volume* e clique em *Next*.
5. Em *Volume Location*, selecione uma pasta/arquivo destino para o contêiner e clique em *Next*.

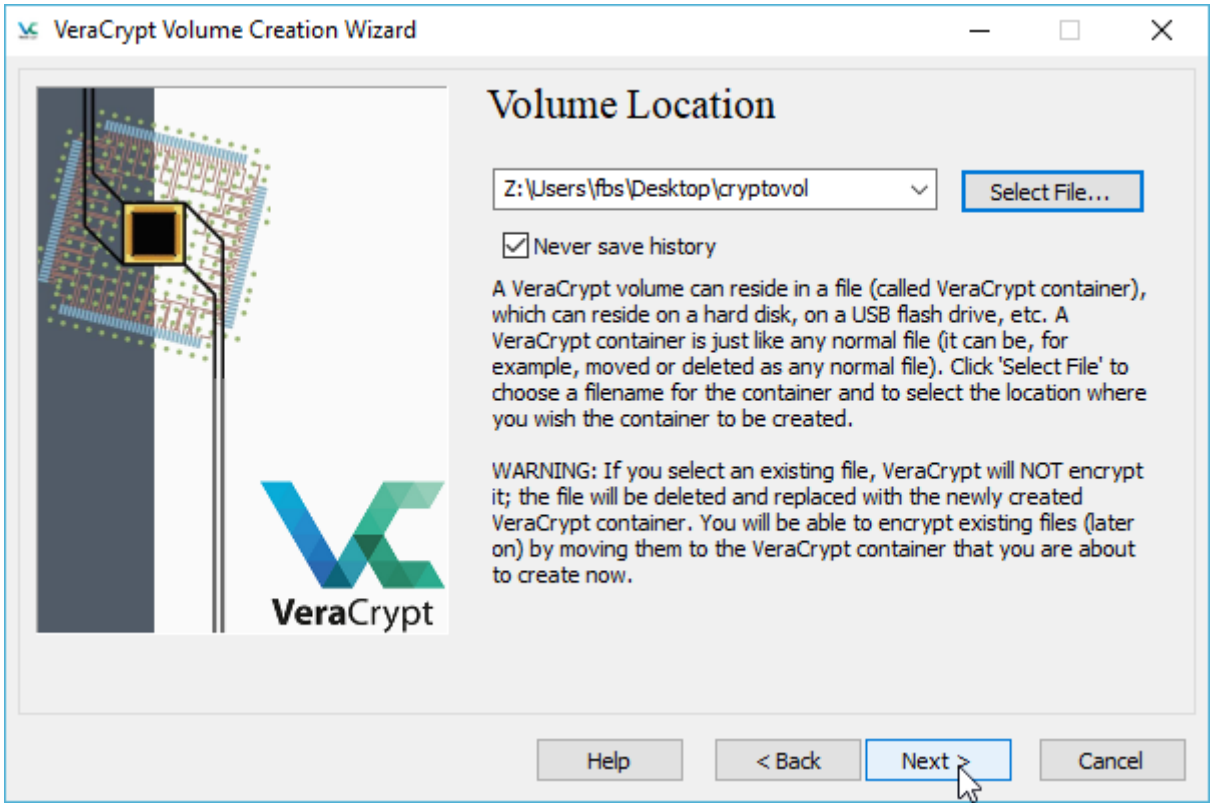

Figura 8. Criação de volumes no VeraCrypt, parte 3

6. Para as opções de criptografia, mantenha o algoritmo AES e hash SHA-512, e clique em *Next*.
7. Para o tamanho do volume, escolha 50MB, e clique em *Next*.

8. Para a senha do contêiner, é importante escolher uma senha forte que não seja facilmente descoberta. Para fins de teste, usaremos `rnpesr123`. Clique em *Next*.
9. Mantenha o *filesystem* em FAT, e mova o mouse para gerar entropia. Finalmente, clique em *Format*.
10. Para montar o volume, selecione uma letra vazia no seu sistema. A seguir, no quadro *Volume* da tela principal do VeraCrypt, clique em *Select File...* e selecione o arquivo indicado no passo (5). Depois, clique em *Mount* e digite a senha informada no passo (8).

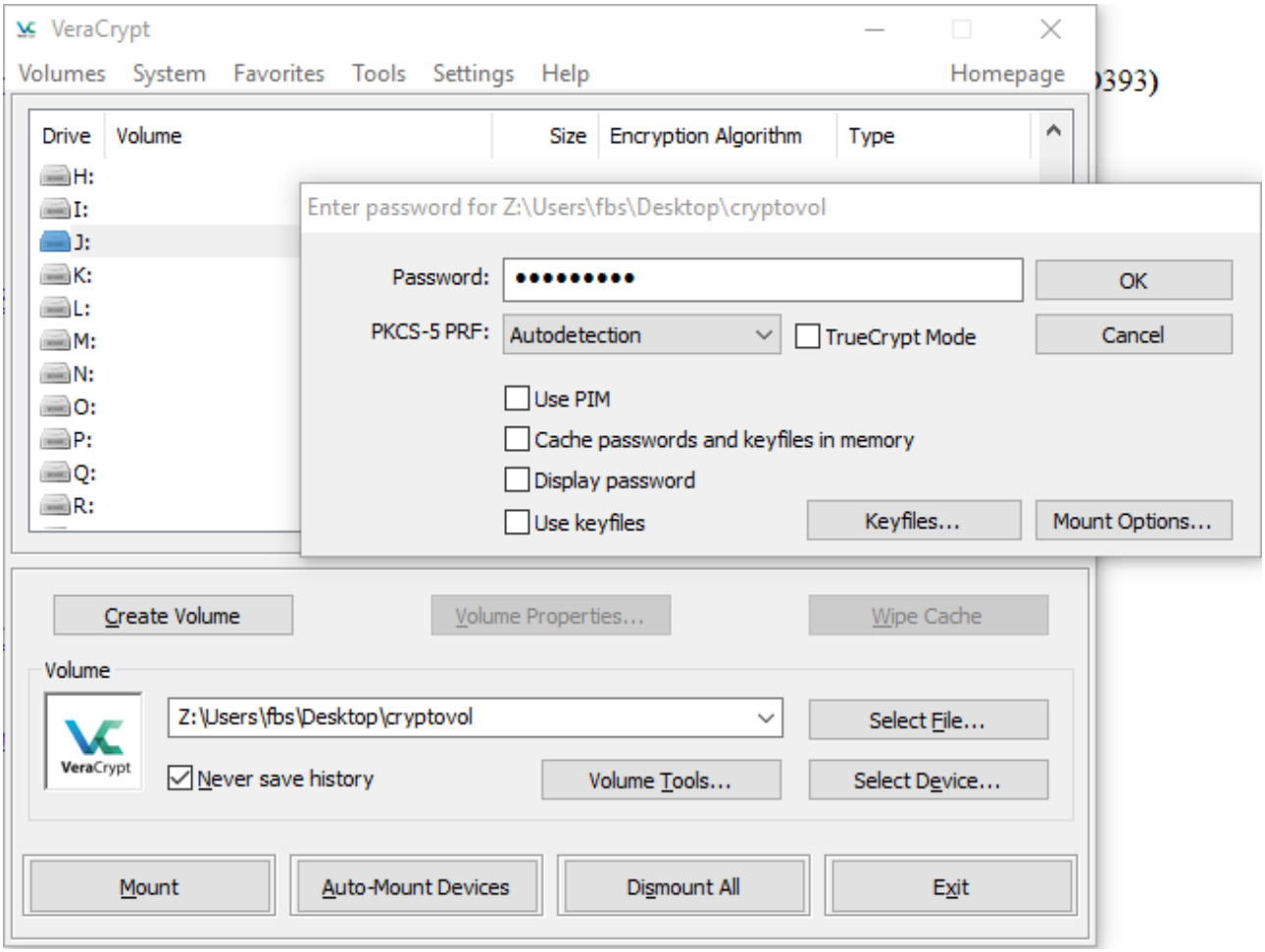

Figura 9. Criação de volumes no VeraCrypt, parte 4

11. Pronto, o volume criptografado está montado. Basta escrever arquivos como desejado e, ao final do processo, clicar em *Dismount* na janela principal do VeraCrypt. Caso queira mover o volume criptografado para outro local, copie-o em um *pendrive*, mídia removível ou mesmo através da Internet, e remonte-o no local de destino.

5) Autenticação usando sistema OTP

Esta atividade será realizada na máquina *LinServer-G*.

Nesta atividade iremos instalar e configurar um sistema TOTP (*time-based one-time password*) usando a ferramenta *Google Authenticator* na máquina *LinServer-G*. Essa autenticação de duplo fator irá prover mais segurança durante logins SSH na máquina-alvo.

1. Instale **em seu celular** o aplicativo *Google Authenticator*:

- Sistemas Android: https://play.google.com/store/apps/details?id=com.google.android.apps.authenticator2&hl=en
- Sistemas Apple: https://itunes.apple.com/us/app/google-authenticator/id388497605?mt=8

2. Para conseguir ler o *QR code* na tela, será necessário ter uma tela maior do que a console padrão do Virtualbox — faça login via `ssh` na máquina *LinServer-G* usando o PuTTY ou Cygwin e vire superusuário usando o comando `su`.

```
fbs@FBS-DESKTOP ~
$ hostname
FBS-DESKTOP
```

```
fbs@FBS-DESKTOP ~
$ ssh aluno@172.16.1.10
Password:
Last login: Thu Sep  6 09:31:40 2018 from 172.16.1.254
aluno@LinServer-A:~$
```

```
aluno@LinServer-A:~$ su -
Password:
root@LinServer-A:~#
```

3. Instale o pacote que implementa suporte ao Google Authenticator na biblioteca PAM:

```
# hostname
LinServer-A
```

```
# apt-get install libpam-google-authenticator
```

4. Depois, insira a linha `auth required pam_google_authenticator.so` imediatamente após a linha 4, `@include common-auth`, no arquivo `/etc/pam.d/sshd`:

```
# nano /etc/pam.d/sshd
(...)
```

```
# head -n5 /etc/pam.d/sshd | grep -v '^#' | sed '/^$/d'
@include common-auth
auth  required  pam_google_authenticator.so
```

5. Configure o `ssh` para permitir autenticação via *challenge-response*, alterando a diretiva `ChallengeResponseAuthentication` no arquivo `/etc/ssh/sshd_config` (linha 49). Feito isso, não

esqueça de reiniciar o daemon do `ssh`.

```
# nano /etc/ssh/sshd_config
(...)
```

```
# grep '^ChallengeResponseAuthentication' /etc/ssh/sshd_config
ChallengeResponseAuthentication yes
```

```
# systemctl restart ssh
```

6. Agora, na máquina *LinServer-G*, execute **como um usuário não-privilegiado** (como o usuário `aluno`) o comando `google-authenticator`.

   Tabela 1. Opções do google-authenticator

Pergunta	Opção
Do you want authentication tokens to be time-based?	y
Do you want me to update your "/home/aluno/.google_authenticator" file?	y
Do you want to disallow multiple uses of the same authentication token?	y
Increase token window from default size of 1:30min to about 4min?	y
Do you want to enable rate-limiting?	y

7. Abra o aplicativo *Google Authenticator* em seu celular e clique no `+` vermelho no canto inferior direito da tela. Em seguida, clique em *Scan a barcode* e leia o *QR code* gerado no passo (6). Na tela principal, deverá surgir uma nova linha com seis dígitos (que serão re-gerados a cada 30s) e o identificador `aluno@LinServer-G`.

8. Verifique que a hora atual do servidor está correta. Como configuramos o NTP na sessão 4, é provável que esteja tudo correto, mas a *timezone* pode estar desconfigurada, como mostrado abaixo:

   ```
   $ date
   Thu Sep  6 09:40:05 EDT 2018
   ```

   Se esse for o caso, rode o comando `dpkg-reconfigure tzdata` como usuário `root`. Escolha *America* > *Sao_Paulo* (ou outra *timezone*, se for esse o caso). Verifique que o relógio foi corrigido:

```
# dpkg-reconfigure tzdata

Current default time zone: 'America/Sao_Paulo'
Local time is now:      Thu Sep  6 10:42:27 BRT 2018.
Universal Time is now:  Thu Sep  6 13:42:27 UTC 2018.
```

```
# date
Thu Sep  6 10:42:36 BRT 2018
```

9. Perfeito, tudo pronto. **NÃO** feche a sessão `ssh` atual, pois em caso de erros poderá ser necessário verificar alguns arquivos. Em lugar disso, abra uma nova sessão `ssh`, como usuário `aluno`, para a máquina *LinServer-G*. No *prompt Verification code*, informe o código temporizado indicado pelo aplicativo instalado em seu celular.

```
fbs@FBS-DESKTOP ~
$ hostname
FBS-DESKTOP
```

```
fbs@FBS-DESKTOP ~
$ ssh aluno@172.16.1.10
Password:
Verification code:
You have mail.
Last login: Thu Sep  6 10:32:40 2018 from 172.16.1.254
aluno@LinServer-A:~$
```

```
aluno@LinServer-A:~$ hostname
LinServer-A
```

```
aluno@LinServer-A:~$ whoami
aluno
```