

Chapitre Sécu_cnx

Connexions sécurisées à distance

ssh

SSL, TLS

Pare-feux

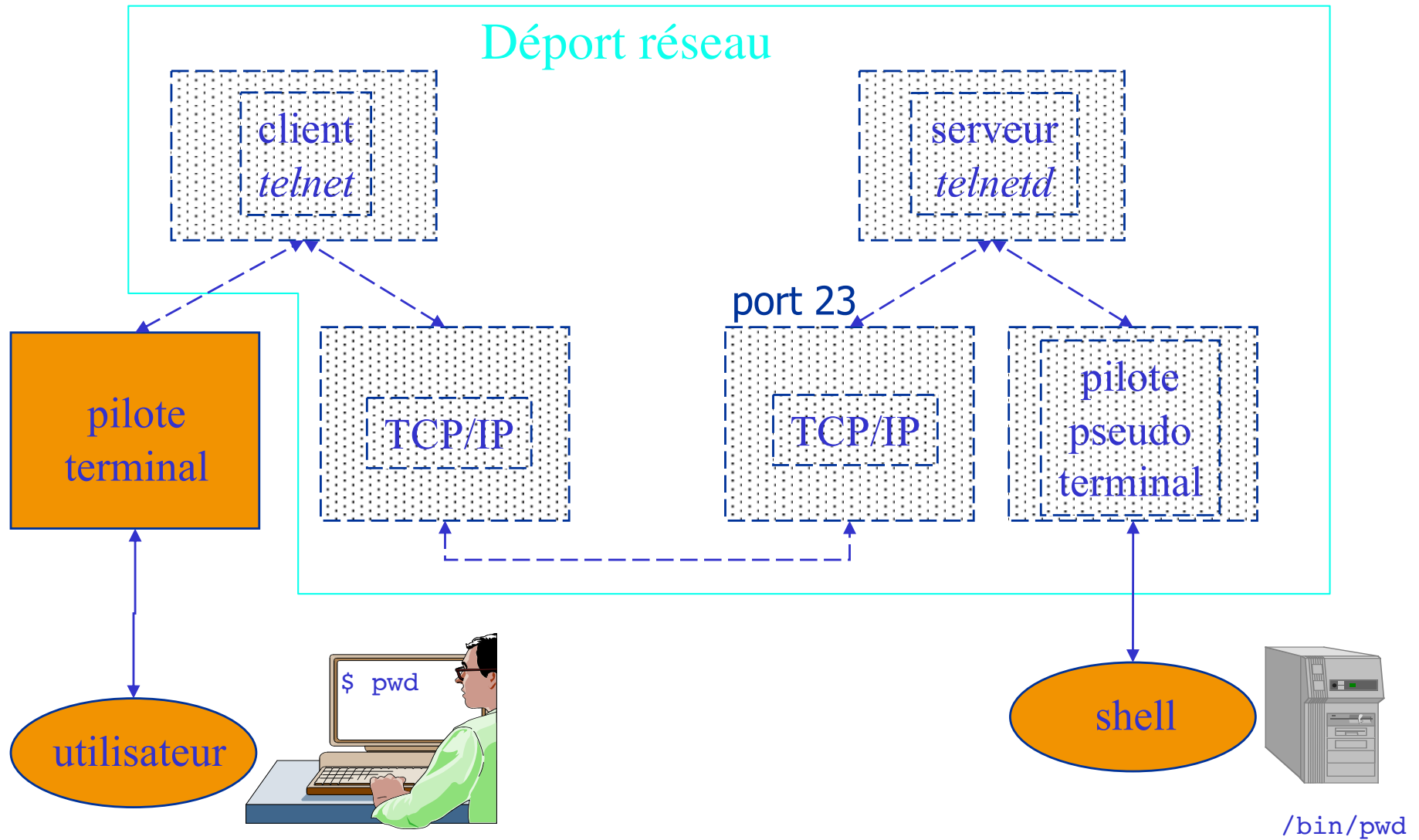
VPN



Contenu du chapitre Sécu_cnx

- Connexion à distance
 - Déport de terminal
- Méthode 1: ssh
 - Fonctionnement de ssh
 - Partage de connexion ssh: « tunnel ssh »
- Méthode 2: TLS
- Notion de pare-feu
- Méthode 3: VPN

telnet, *ssh*: terminal à distance



Menaces & solutions

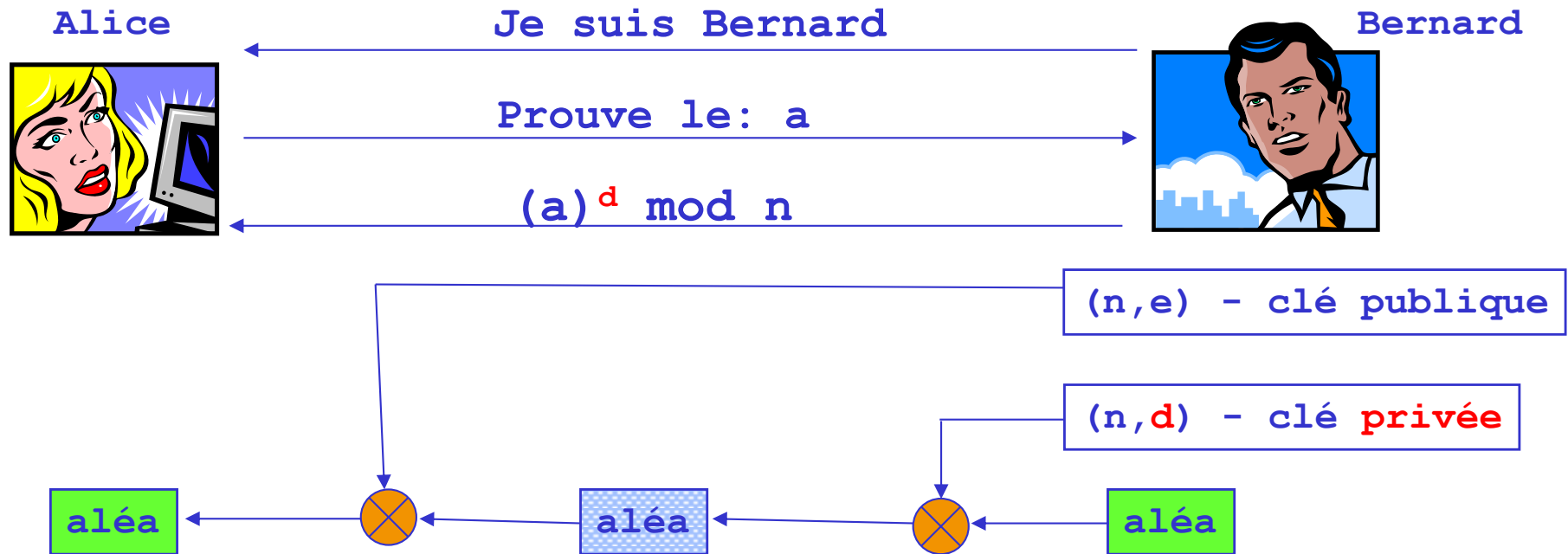
Oscar



R. Groz veut se connecter sur `pcserveur`. Oscar pourrait :

1. Se faire passer pour `pcserveur` vis à vis de groz
 2. Se faire passer pour groz vis à vis de `pcserveur`
 3. Observer l'envoi d'un mot de passe, ou toutes les commandes et réponses
- 3. « snoop » : observation du contenu des échanges
 - Chiffrer les informations sur la ligne
 - 1.&2. « spoof » : usurpation d'identité (de machine, d'utilisateur)
 - Authentifier : s'assurer de l'identité des (deux) interlocuteurs

Authentification à clé publique



- Alice envoie un défi aléatoire a , à usage unique
- Bernard le chiffre avec d : $(a)^d \bmod n$
- Alice vérifie que $(a^d)^e \bmod n = a$

CONTRAINTES: Alice doit connaître la clé publique de Bernard

- Enregistrée avant (ssh)
- Par certificat (SSL/TLS)



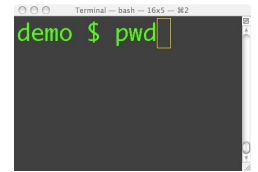
Sécurité dans ssh

`grozr@ens: ssh pcserveur`

1. ens authentifie pcserveur avec la clé publique de pcserveur (fic. `~/.ssh/known_hosts` sur ensi-ens)
2. Création de **clé secrète** entre ens et pcserveur par Diffie-Helman, pour chiffrer toute la suite de la session
3. Login de grozr sur pcserveur: authentifier grozr
 - authentification avec la clé publique de grozr si installée sur pcserveur `~/.ssh/authorized_keys`
N.B. Création clé publique par `ssh-keygen`
grozr s'authentifie en envoyant nom + id-session chiffré avec sa clé **privée** (`~/.ssh/id_rsa`)
 - Sinon, par **mot de passe** (chiffré cf 2.)

Cf TP sécurité

Connexion directe



```
ens% ssh pcserveur
```



ens.ensimag.fr

pcserveur

Clé publique de pcserveur,
stockée sur ens (codage
base64) :

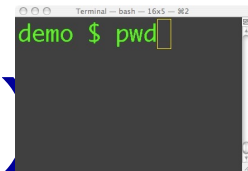
```
~/.ssh/known_hosts: pcserveur ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQDLsv5  
XA+fMcJs...  
/YosCYGerlZenEBYucfy9pXeRsa7DQQvgV
```

NB: la clé **privée** de grozr est dans:

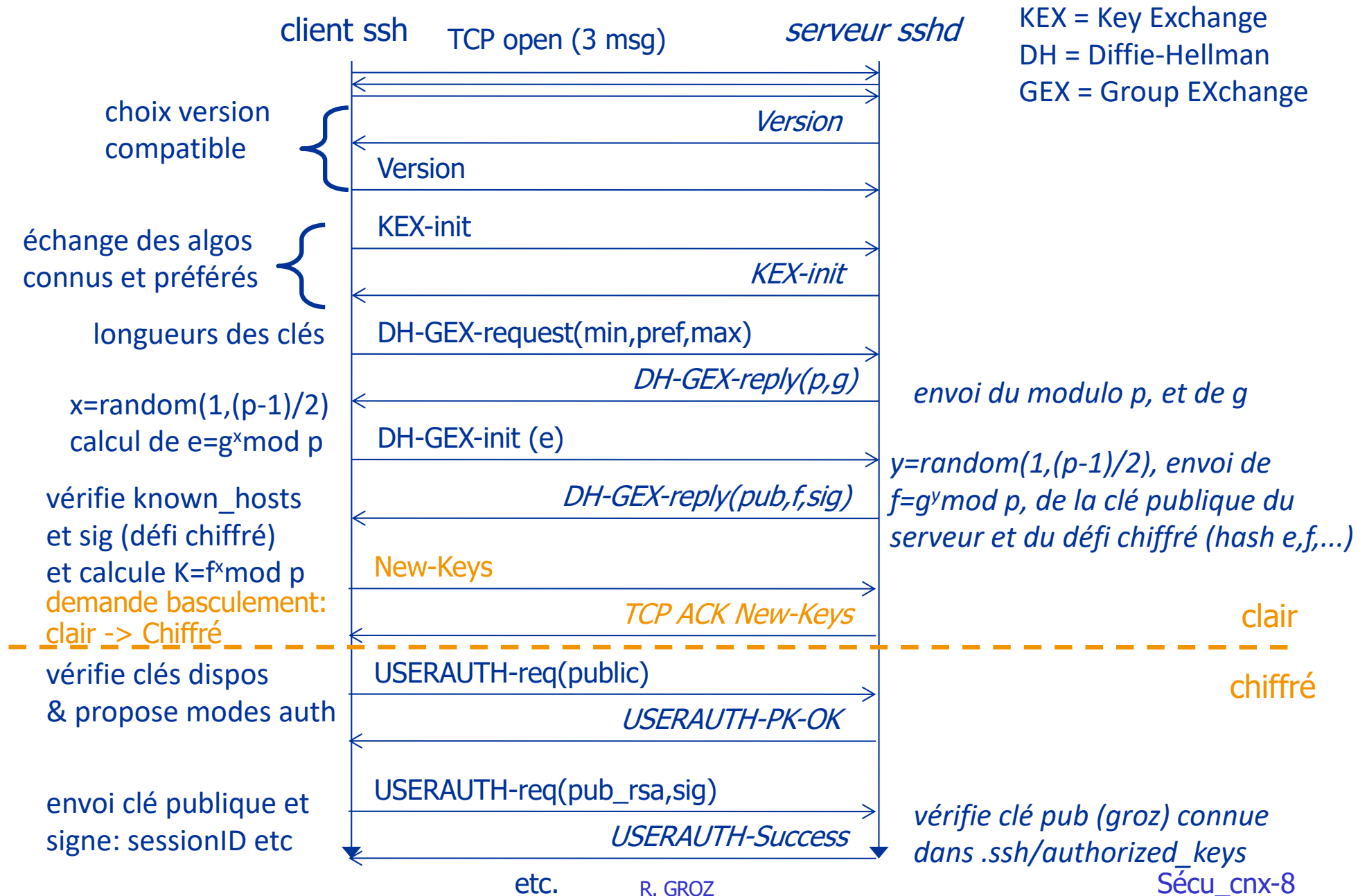
```
~/.ssh/id_rsa
```

Clé publique de grozr dans:

```
~/.ssh/authorized_keys: ssh-rsa  
AAAAB3NzaC1yc2EAAAABIwAAAIEAqW  
TsNmJr1EsRsHoB3+XP02/I7WcAml  
... RmMdyk7pZfeWCZe0=  
grozr@sixte.imag.fr
```




Echanges du protocole SSH (exemple)





Pourquoi n'y a-t-il pas de faille ?

`grozr@ens:ssh` -->  --> `pcserveur`

1. `ens` authentifie `pcserveur` avec la clé publique de `pcserveur` (fic. `known_hosts` sur `ens`)
Oscar laisse faire l'authentification, puis usurpe l'adresse IP de `pcserveur`
 2. Création de clé secrète entre `ens` et *Oscar* par Diffie-Helman, pour chiffrer toute la suite de la session
Oscar intercepte tous les messages, et en parallèle il fait un Diffie-Helman avec `pcserveur` en jouant le rôle de `ens`
 3. Login de `grozr` sur `pcserveur`
*Oscar renvoie les informations à `pcserveur` pour transmettre en retour les «bonnes» réponses à `ens`. **Oscar voit tout passer***
- Attaque de l'homme au milieu (Man in the Middle)
Mais elle échoue: pourquoi ?

ssh: offre 2 services

- Login à distance sécurisé (remplace telnet, rlogin)
 - Authentification de la machine distante et de l'utilisateur
 - chiffrement de la connexion à l'aide d'une clé de session secrète
 - port 22, échappement: ~

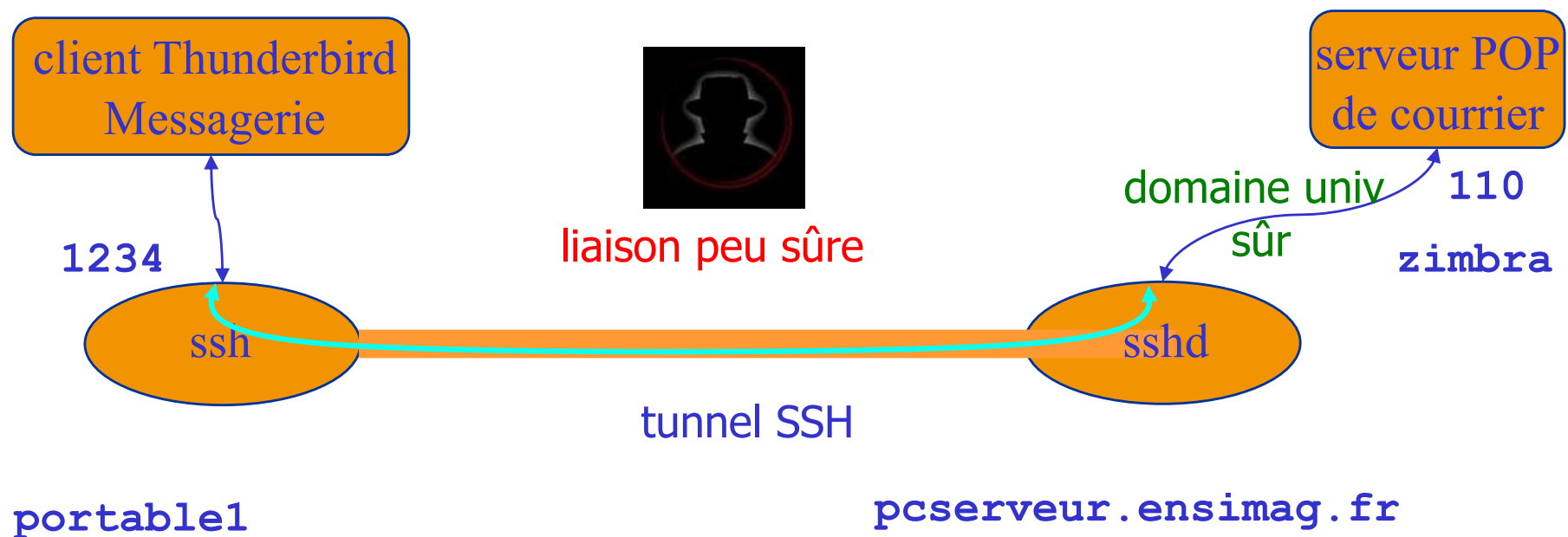
- On peut exécuter une commande (défaut=login)

- Ex: `ssh machine_distante date`

*Mais ssh offre plus qu'un simple terminal de login:
les mécanismes de sécurité peuvent être utilisés en parallèle (partagés)
pour acheminer des flux de communication pour d'autres
applications*

- « Tunnels » et redirection de connexions
 - ≈ connexion TCP sécurisée: port local <-> port distant
 - transfert chiffré pour d'autres applications:
 - courrier, fichier, etc
 - sessions X11 sécurisées
 - Possibilité de passer par un « bastion » d'entrée vers un réseau isolé par un pare-feu

Redirection d'un port local

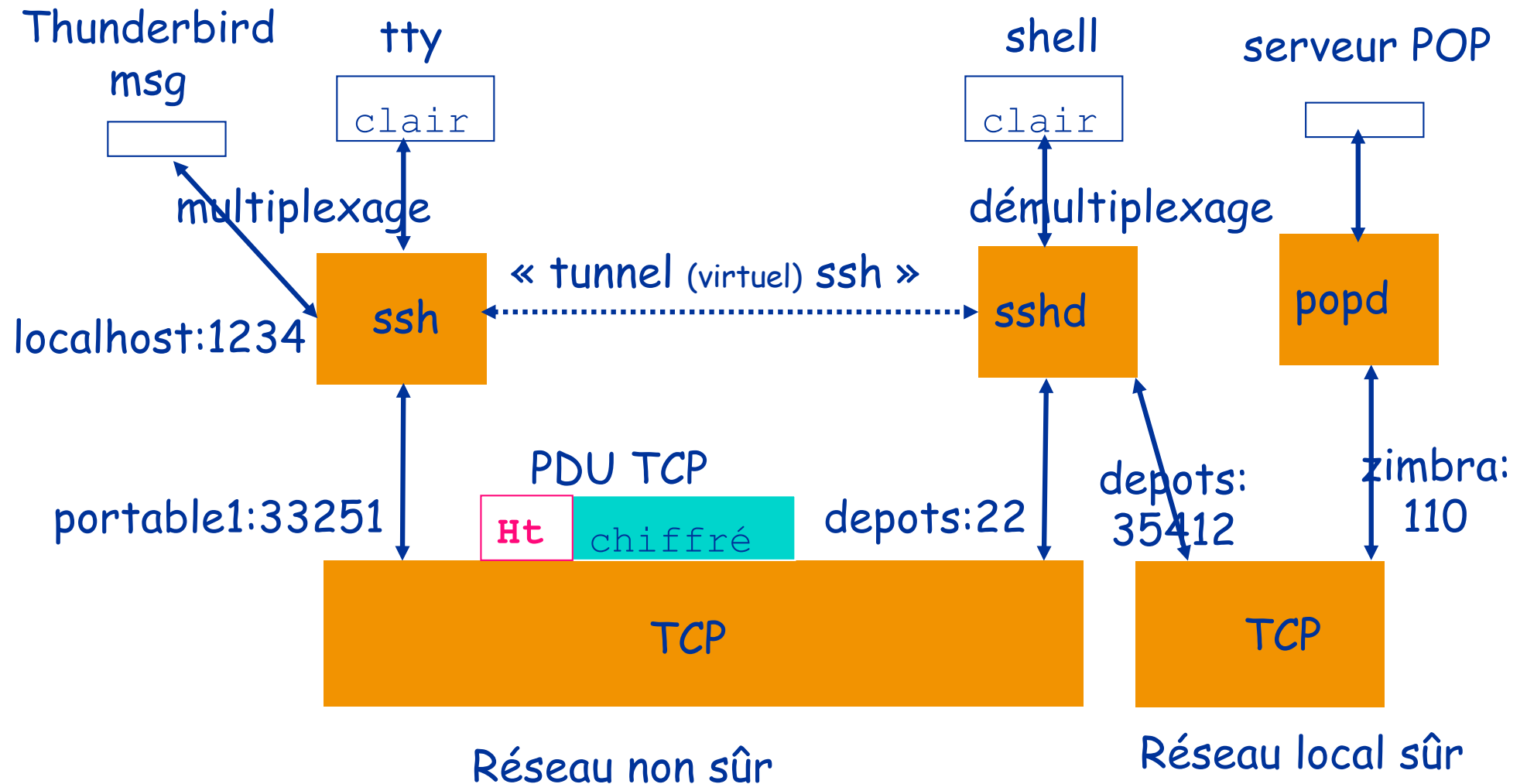


```
portable1% ssh -L 1234:zimbra.UGA.fr:110 pcserveur.ensimag.fr
```

Configurez Thunderbird sur `portable1` pour lire le courrier par POP sur:
`localhost`, port 1234

Il sera en fait lu sur `zimbra` mais chiffré lors de son passage sur le réseau via
ssh

Partage du « tunnel » chiffré



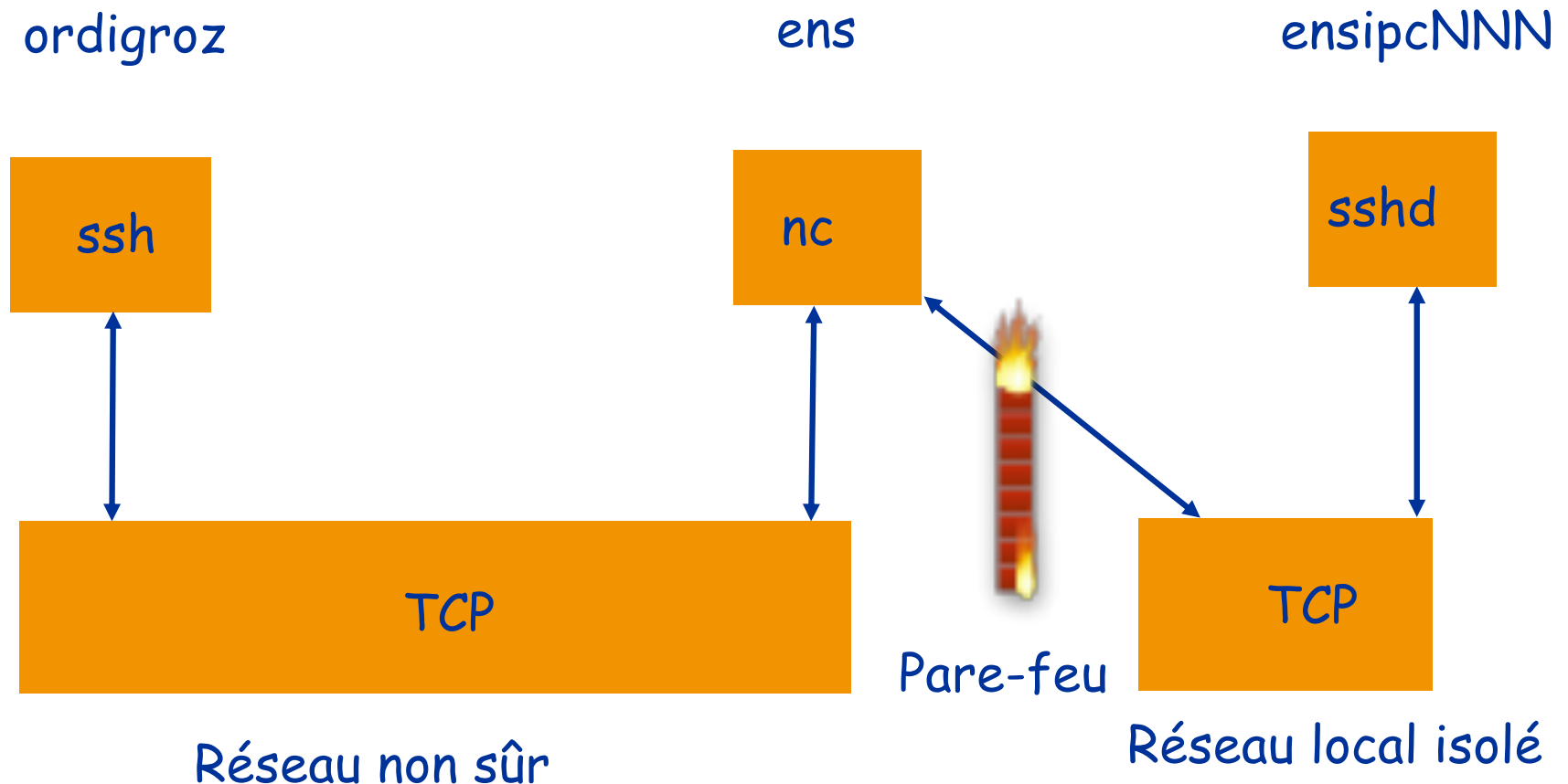
Accès ssh via proxy (bastion)

~/.ssh/config (sur ordigroz) :

```
Host ensipc* *.ensimag.fr
```

```
ProxyCommand ssh -q ens.ensimag.fr nc %h 22
```

nc = netcat: simple renvoi



ssh

- Excellente sécurité
 - chiffrement et authentification
 - a remplacé telnet/rlogin
- Intégration facile avec d'autres applications
 - courrier, X11 (par ssh -X ou ssh -Y ...)
 - Multiplexage de connexions chiffrées sur le même tunnel ssh

Autre commande utile (copie de fichier à distance):

```
ens% scp fichier1 groz@ligtwo:repert/fic
```

Session X11 chiffrée:

```
ssh -X ou bien (plus sûr) ssh -Y ens.ensimag.fr
```

«tout simplement », et ssh se charge de positionner le DISPLAY pour le shell de connexion, et de connecter les ports utilisés par X11



Contenu du chapitre Sécu_cnx

- Connexion à distance
 - Déport de terminal
- Méthode 1: ssh
 - Fonctionnement de ssh
 - Partage de connexion ssh: « tunnel ssh »
- Méthode 2: TLS
- Notion de pare-feu
- Méthode 3: VPN

SSL-TLS: sessions chiffrées

- Ssh: sécurisé lorsqu'on possède un compte sur machine distante
- Commerce électronique: comment sécuriser accès à site marchand ?
 - Nouveaux clients (humains) n'ont pas créé de compte
 - Clients ne sont pas experts pour faire du tunnel ssh
 - Garder accès Web (pages, navigateur etc)
- Solution: SSL (1995) devenu TLS (1999)
 - Couche 5 session au-dessus de TCP
 - Authentification du serveur par certificat
 - Puis chiffrement symétrique par clé calculée par client (processus TLS) puis partagée avec serveur

TLS – Transport Layer Security

- Protocole de niveau session (OSI-5) s'intercalant entre l'application HTTP et TCP; http+tls=https (port 443)
 - Ex: https://webmail.grenoble-inp.org
 - NB: utilisable par d'autres applis (pop, imap, smtp)
- Authentification du serveur
 - Les navigateurs connaissent les clés publiques de CA racines
 - Le navigateur demande au serveur un certificat
 - Le navigateur extrait du certificat la clé publique du serveur
 - Le navigateur authentifie le serveur par un défi de session
- Authentification du client: ad libitum (certif. ou login)
- Chiffrement et intégrité des données
 - Le client propose une clé préliminaire aléatoire de 384 bits, chiffrée avec la clé publique du serveur
 - le serveur (et le navigateur) calculent une clé symétrique à partir de cette clé préliminaire et du défi de l'authentification
 - les messages suivants sont chiffrés et signés

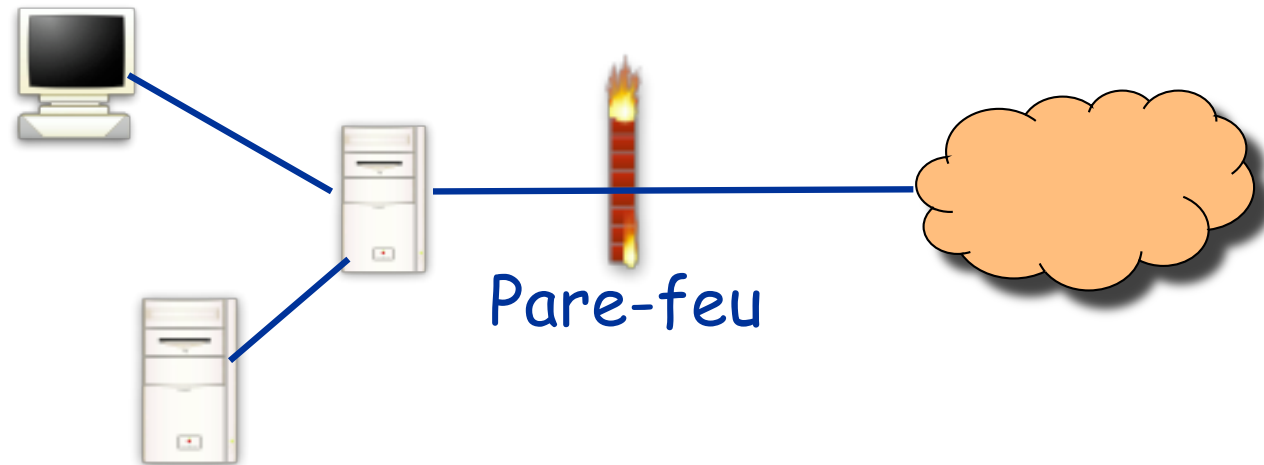


Contenu du chapitre Sécu_cnx

- Connexion à distance
 - Déport de terminal
- Méthode 1: ssh
 - Fonctionnement de ssh
 - Partage de connexion ssh: « tunnel ssh »
- Méthode 2: TLS
- Notion de pare-feu
- Méthode 3: VPN

Pare-feux

- Filtrage des flux de communication / réseau



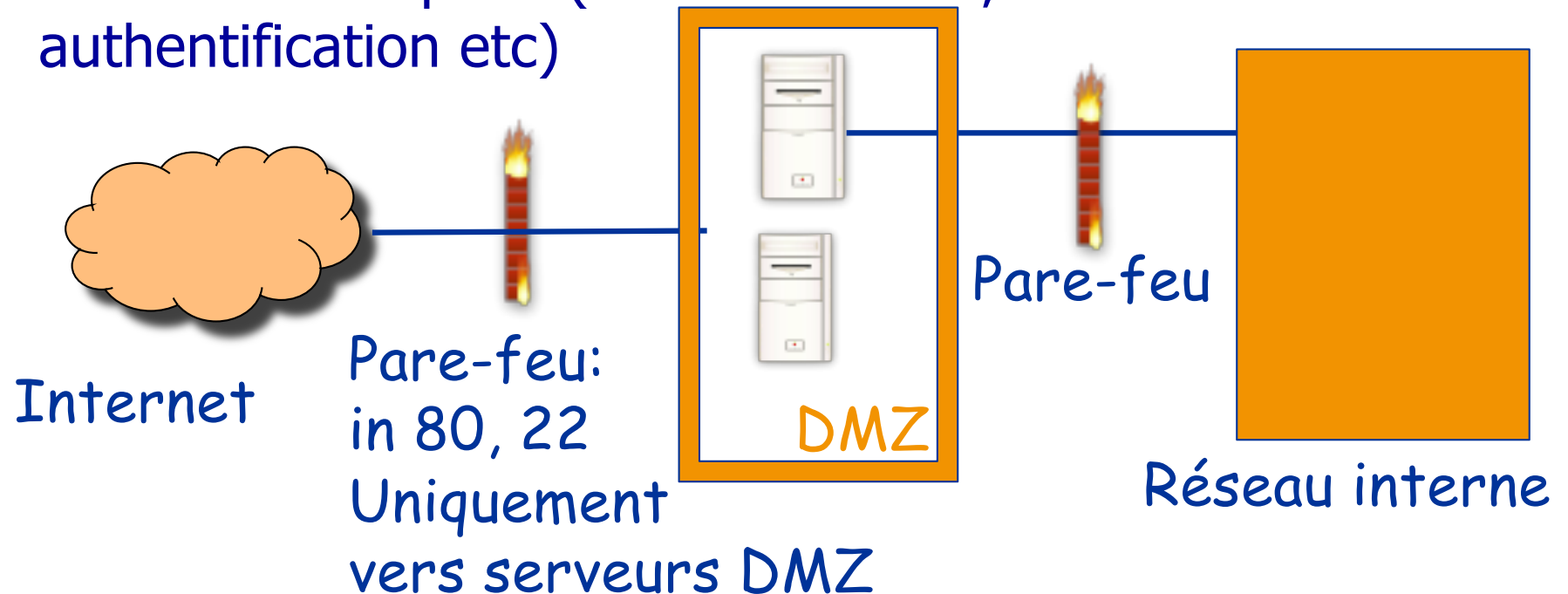
- Flux définis par adresses: niveau 3 (IP) ou 4 (ports) le plus souvent

```
deny src-ip 10.0.0.0/24,127.0.0.1/8  
allow in proto tcp to any port www
```

Pare-feux et DMZ

- DMZ: Zone DÉmilitarisée:

- Contient des serveurs accessibles de l'Internet pour certains services
- Peut contenir des passerelles sécurisées pour accès vers le réseau privé (ex: bastion ssh, serveur authentification etc)





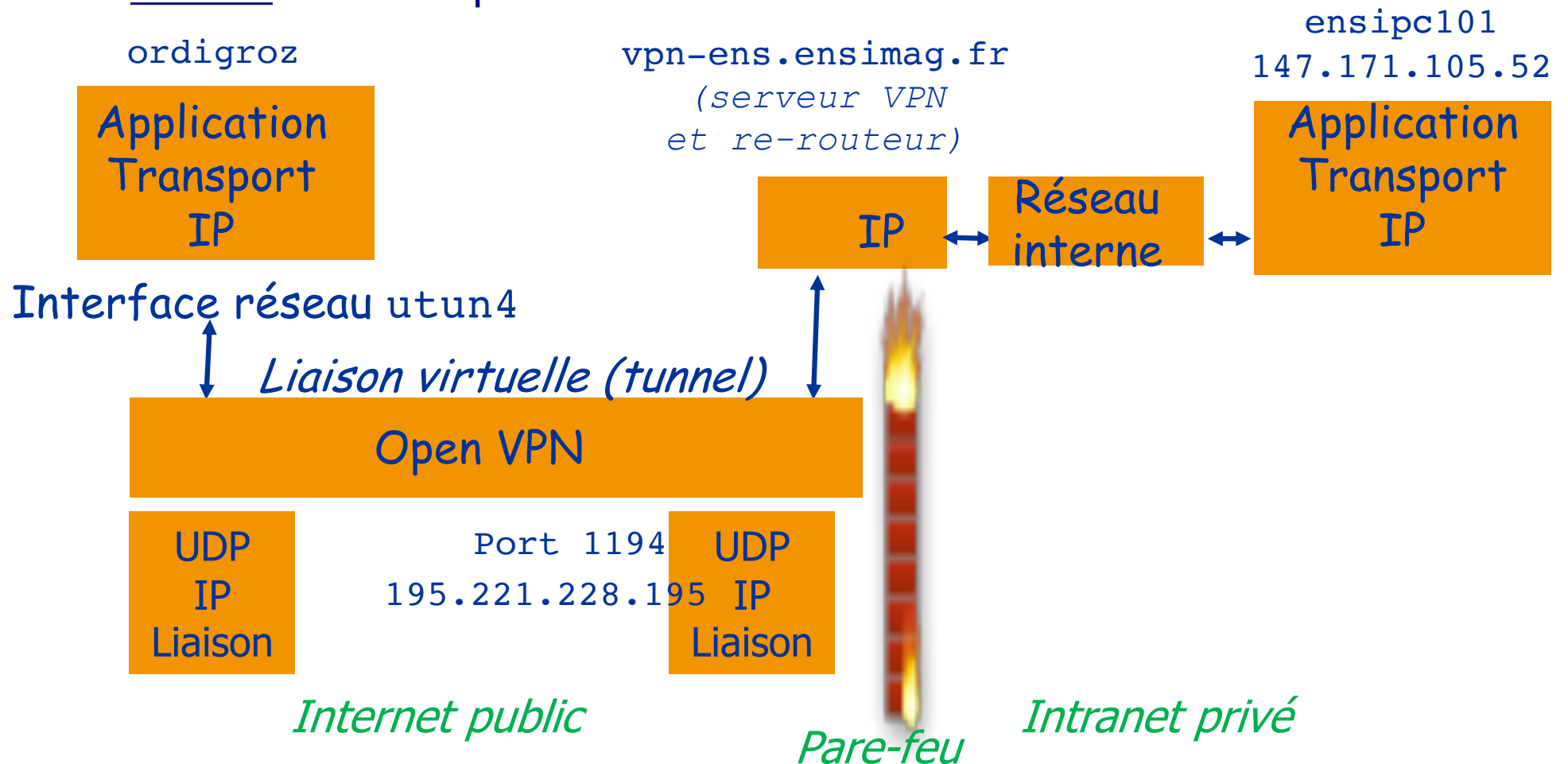
Contenu du chapitre Sécu_cnx

- Connexion à distance
 - Déport de terminal
- Méthode 1: ssh
 - Fonctionnement de ssh
 - Partage de connexion ssh: « tunnel ssh »
- Méthode 2: TLS
- Notion de pare-feu
- Méthode 3: VPN

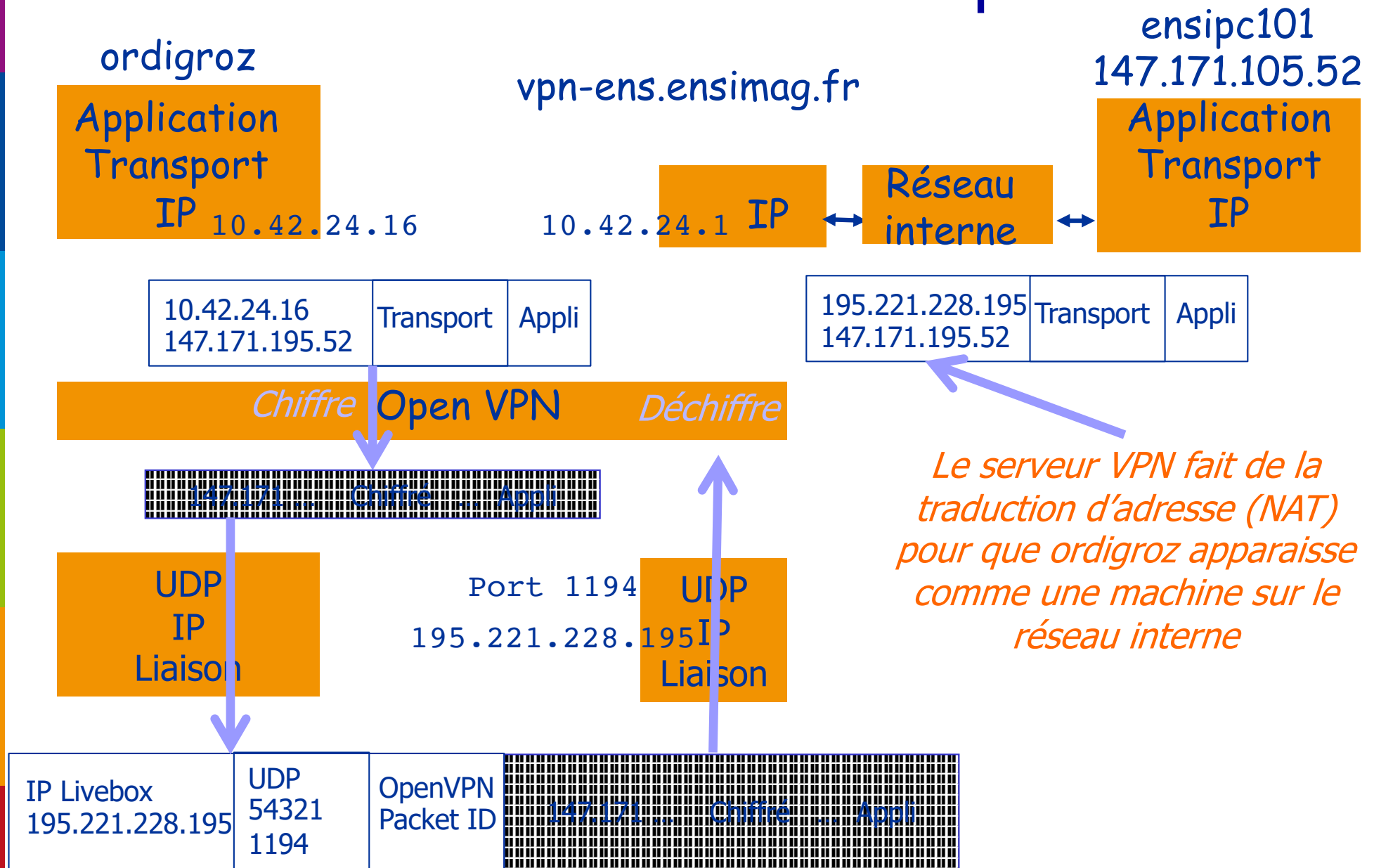
VPN=Virtual Private Network (fr: RPN)

- Au sens général, structure de réseau privé (multi-site) bâtie au-dessus d'un réseau public
 - Avec ou sans tunnel, mais l'opérateur garantit au moins l'isolation des flux
 - Alternative virtuelle moins chère que la solution physique de câbles et fibres physiques à longue distance
- Ne se limite donc pas à de la connexion chiffrée à distance=tunnel VPN (ce que vous utilisez)
- Même pour les tunnels VPN, il y a beaucoup de solutions différentes
 - OpenVPN sur UDP n'en est qu'une (Cisco en a d'autres, possibilité d'utiliser IPsec etc).

- La table de routage d'ordigroz est modifiée pour envoyer trafic vers intranet par `utun4`
- Tous les paquets échangés avec l'intranet sont chiffrés et réencapsulés dans UDP



Tunnel VPN: structure d'encapsulation



OpenVPN Ensimag: tables de routage

Table sans VPN

| Destination | Gateway | Netif |
|---------------------------|-------------------------|------------|
| default | livebox.home | en0 |
| 127 | localhost | lo0 |
| localhost | localhost | lo0 |
| 169.254 | link#4 | en0 |
| 192.168.1 | link#4 | en0 |
| 192.168.1.1/32 | link#4 box | en0 |
| livebox.home | 8c:fd:de:bc:1c:8 | en0 |
| 192.168.1.14/32 | link#4 ordigroz | en0 |
| 255.255.255.255/32 | link#4 | en0 |

Notes:

- Au départ, le client VPN contacte le serveur, et l'authentifie et négocie les clés par TLS (sur OpenVPN sur UDP)
- Puis il modifie les tables de routage
- Ordigroz a ensuite:
 - une interface réelle **en0** sur le réseau Wifi, avec adresse **192.168.1.14**
 - une interface virtuelle **utun4**, sur un autre réseau (privé) VPN, avec adresse **10.42.24.16**, pour le trafic vers l'intranet Ensimag

Table avec VPN

| Destination | Gateway | Netif |
|---------------------------|-----------------------------|--------------|
| 0/1 | livebox.home | en0 |
| default | livebox.home | en0 |
| 10.42.24/23 | 10.42.24.16 ordigroz | utun4 |
| 127 | localhost | lo0 |
| localhost | localhost | lo0 |
| 130.190.254/23 | 10.42.24.1 | utun4 |
| 147.171.104/21 | 10.42.24.1 vpn-ens | utun4 |
| 147.171.112/24 | 10.42.24.1 | utun4 |
| 169.254 | link#4 | en0 |
| 192.168.1 | link#4 | en0 |
| 192.168.1.1/32 | link#4 | en0 |
| livebox.home | 8c:fd:de:bc:1c:8c | en0 |
| 192.168.1.14/32 | link#4 | en0 |
| 195.220.30 | 10.42.24.1 | utun4 |
| 195.221.227 | 10.42.24.1 | utun4 |
| 195.221.228 | 10.42.24.1 | utun4 |
| 195.221.228.195/32 | livebox.home | en0 |
| 195.221.229 | 10.42.24.1 | utun4 |
| 195.221.230 | 10.42.24.1 | utun4 |
| 255.255.255.255/32 | link#4 | en0 |

- Seul le trafic vers les adresses Ensimag sera envoyé par **utun4**
- Tout autre trafic passera par défaut par la route normale (box)

Bilan chapitre Sécu_cnx: notions essentielles

- Sécurisation d' une connexion à distance
 - Savoir se servir de ssh-keygen, et gérer les fichiers de clés (known_hosts, authorized_keys): vu en TP
- Notion de tunnel chiffré (ssh ou VPN)
- Principe de TLS
- Pare-feux