

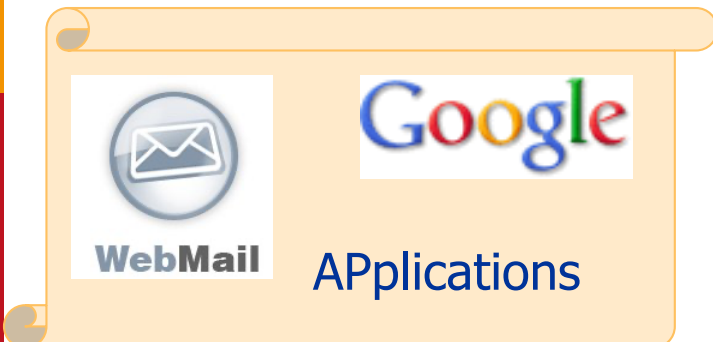
Chapitre AP_cnx

Connexions sécurisées à distance

ssh

SSL, TLS

Pare-feux

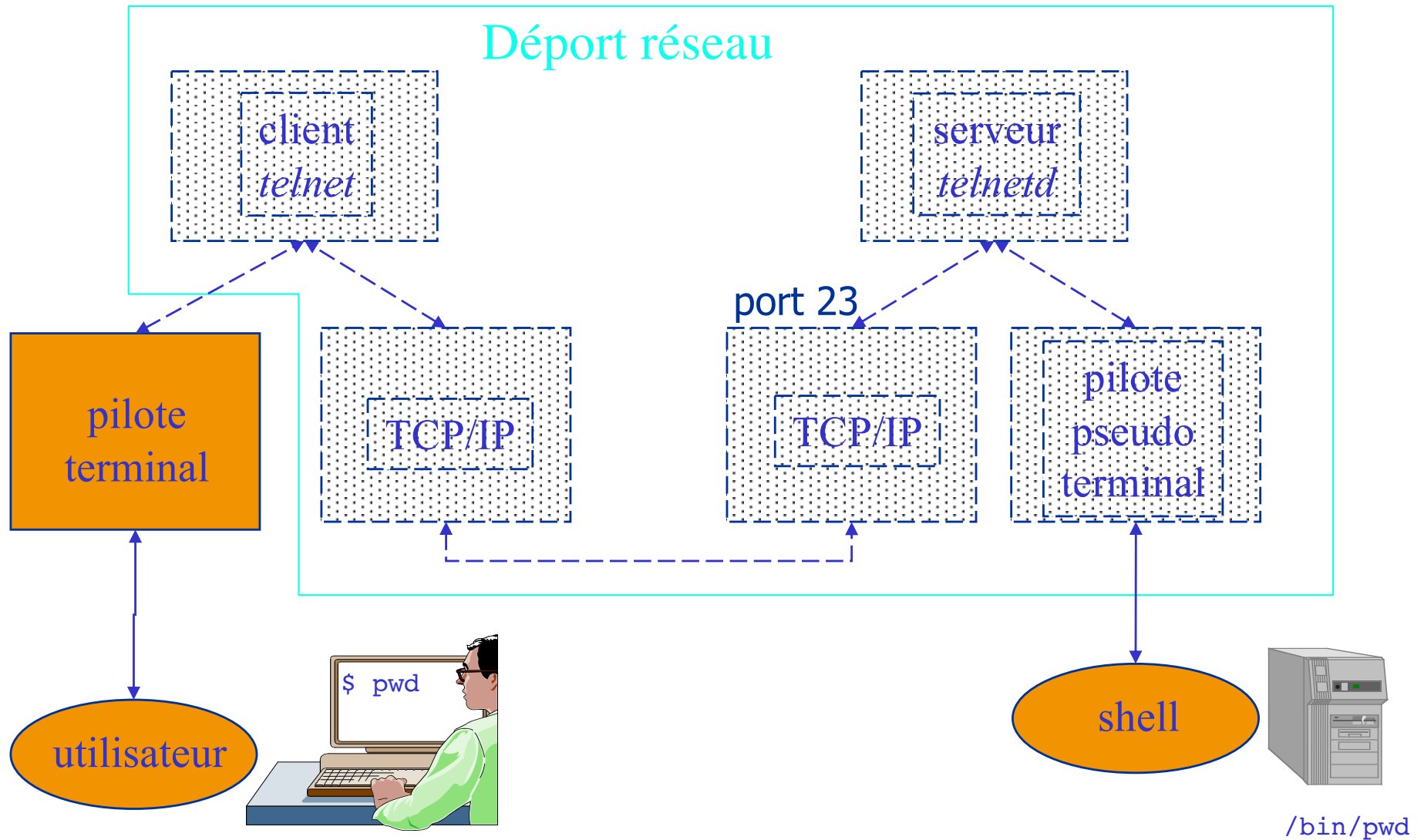




Contenu du chapitre AP_cnx

- Connexion à distance
 - Déport de terminal
- Ssh
- Partage de connexion ssh: « tunnel ssh »
- SSL/TLS
- Pare-feux

telnet, *ssh*: terminal à distance



telnet : terminal à distance

Principe: transmettre des caractères au shell distant et renvoyer les réponses du shell à l'utilisateur

- Connexion entre n'importe quels systèmes d'exploitation
 - NVT (Network Virtual Terminal) ASCII simple
 - 7 bits, fin de ligne CR, LF
 - utilisé par FTP, SMTP, finger, whois, HTTP...
- Mode ligne (traitée en local) par défaut, ou caractère
 - caractère: chaque caractère tapé provoque l'envoi immédiat d'un paquet
 - une correction (BackSpace) est donc transmise et traitée par le serveur
 - ligne: le client telnet n'envoie qu'une ligne complète (lors du « Enter »)
 - la correction est traitée en local, par le client
- Échappement - interprété par le client telnet
 - envoyer une ligne de commande au client (ex: couper la communication)
 - Control-] (par défaut, modifiable avec option -e)
- Port 23 (par défaut): service telnet
 - login distant, le mot de passe circule en clair
- `telnet <port>`: connexion TCP avec serveur sur ce port

Oscar



Menaces & solutions

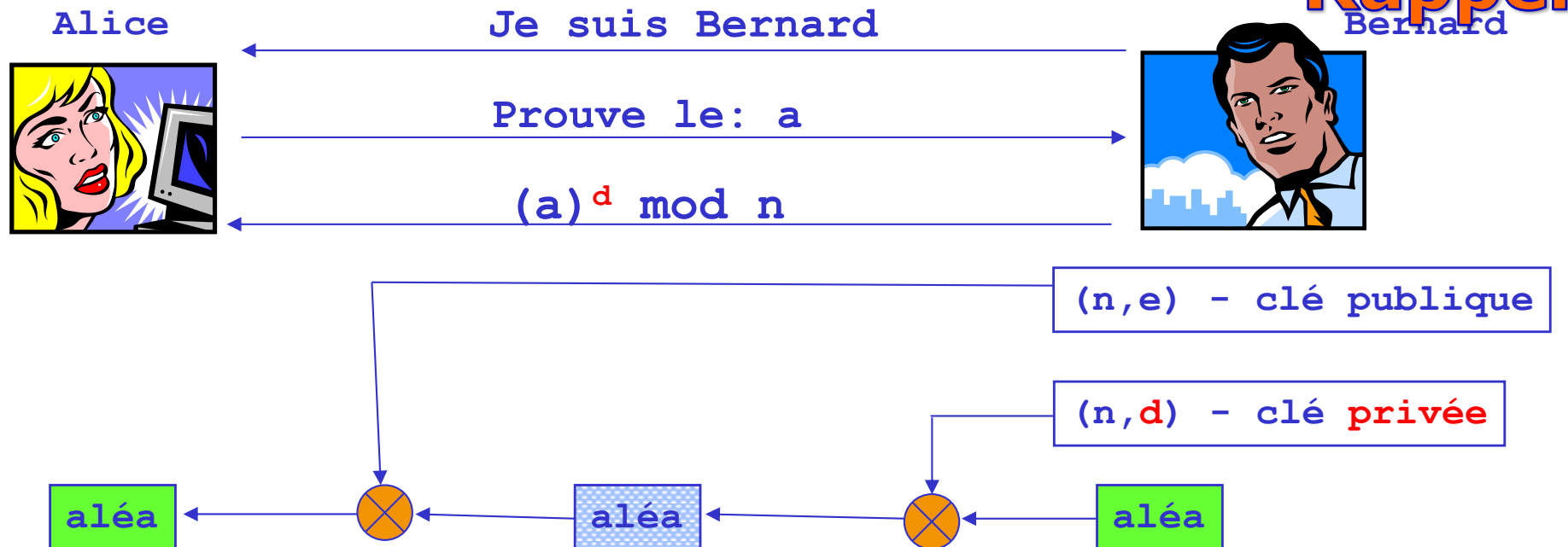
Oscar



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)



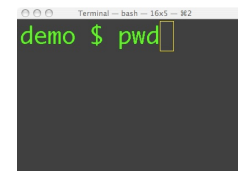
Sécurité dans ssh

`grozr@ensi-ens: ssh pcserveur`

1. ensi-ens authentifie pcserveur avec la clé publique de pcserveur (fic. `~/.ssh/known_hosts` sur ensi-ens)
2. Création de **clé secrète** entre ensi-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



```
ensi-ens% ssh pcserveur
```



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

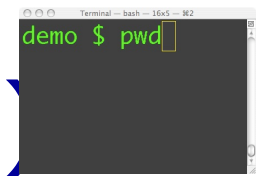
```
~/.ssh/known_hosts: pcserveur ssh-rsa  
AAAAB3NzaC1yc2EAAAADAQABAAQDLsv5  
XA+fMcJs...  
/YosCYGerlZenEBYucfy9pXeRsa7DQQvgV
```

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

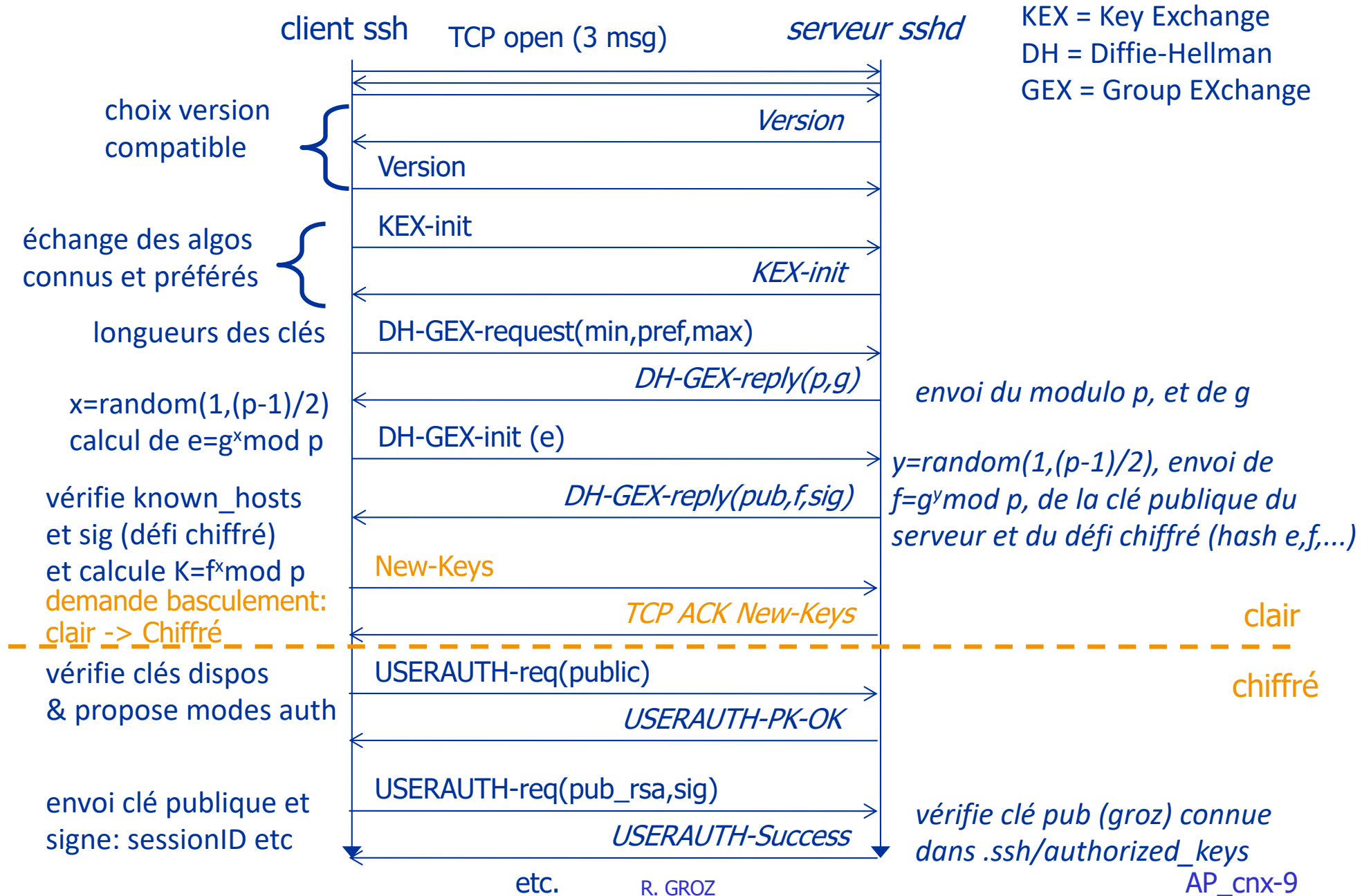
```
~/.ssh/id_rsa
```

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@ensi-ens:ssh--→`  `--→ pcserveur`

1. ensi-ens authentifie ^{Oscar}pcserveur avec la clé publique de pcserveur (fic. `known_hosts` sur ensi-ens)
Oscar laisse faire l'authentification, puis usurpe l'adresse IP de pcserveur
2. Création de clé secrète entre ensi-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 ensi-ens
3. Login de grozr sur pcserveur
*Oscar renvoie les informations à pcserveur pour transmettre en retour les «bonnes» réponses à ensi-ens. **Oscar voit tout passer!***

Attaque de l'homme au milieu (Man in the Middle)

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: ~

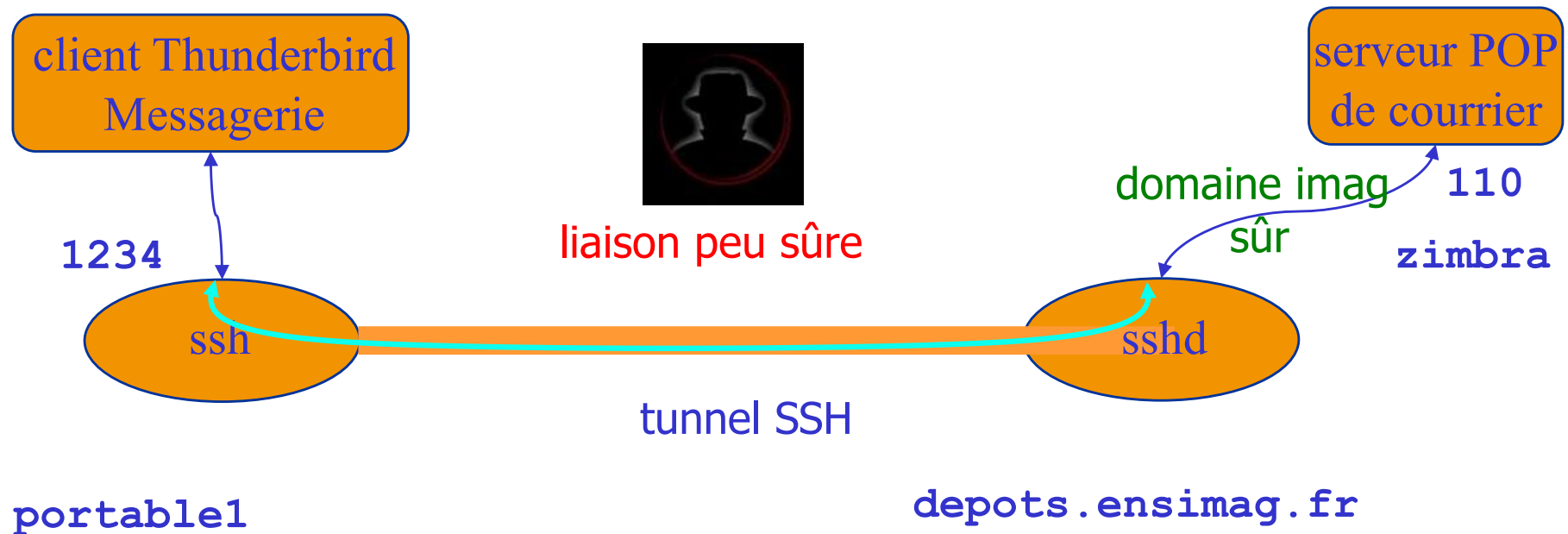
- On peut exécuter une commande (défaut=login)

- 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



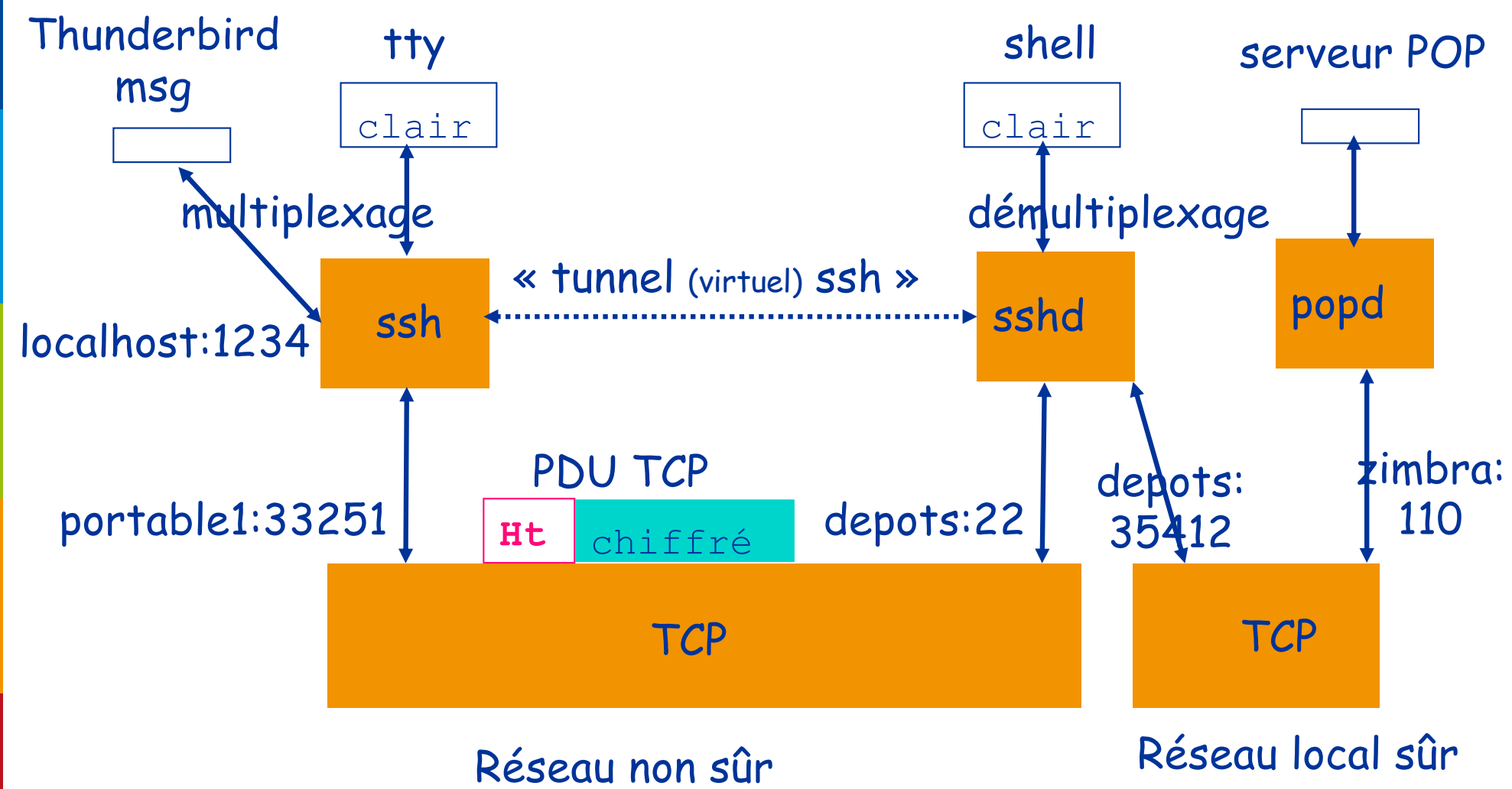
```
portable1% ssh -L 1234:zimbra.imag.fr:110 depots.ensimag.fr
```

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



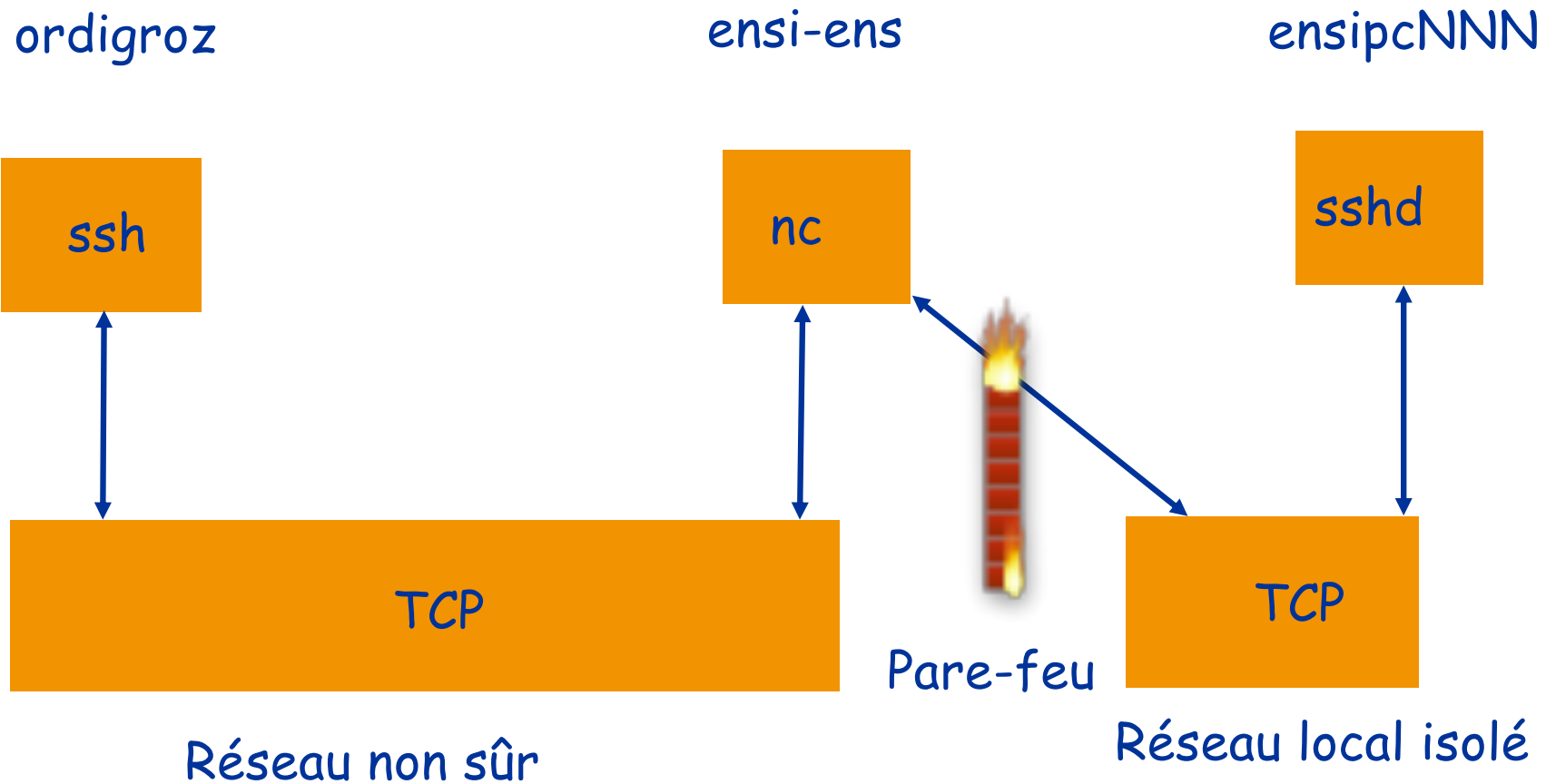
Accès via proxy (bastion)

`~/.ssh/config` (sur `ordigroz`):

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

```
ProxyCommand ssh -q ensi-ens.imag.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):

```
ensi-ens% scp fichier1 groz@bastet:repert/fic
```

Session X11 chiffrée:

```
ssh -X ensi-ens ou bien (plus sûr) ssh -Y ensi-ens
```

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

SSL-TLS: sessions chiffrées

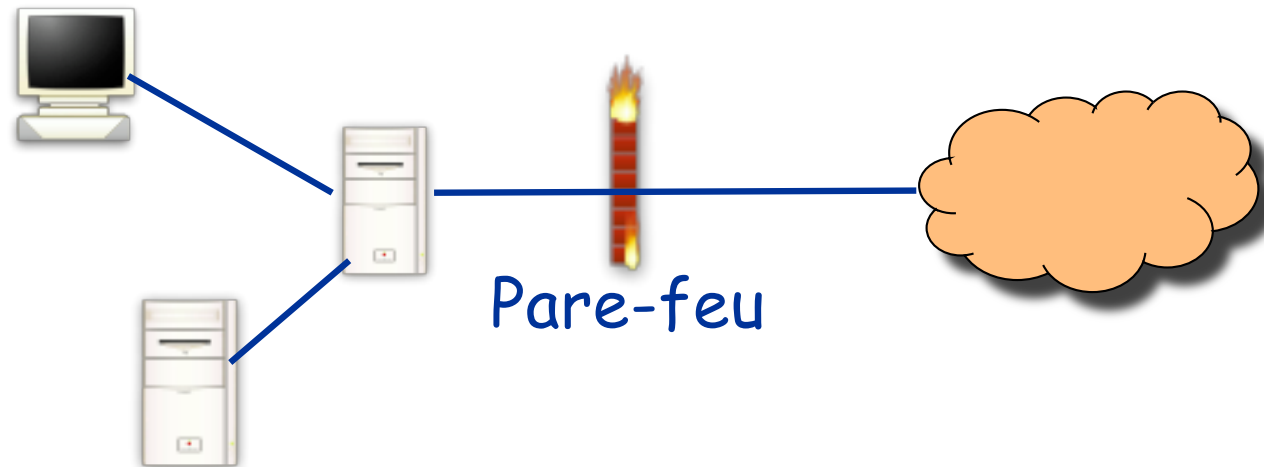
- Ssh: sécurisé lorsqu'on possède un compte sur machine distante
- Commerce électronique: comment sécuriser accès à site marchand ?
 - Nouveaux clients (humains) n'ont pas créé de compte
 - Clients ne sont pas experts pour faire du tunnel ssh
 - Garder accès Web (pages, navigateur etc)
- Solution: SSL (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 serveur

SSL (Secure Sockets Layer) - TLS

- Protocole de niveau session (OSI-5) s'intercalant entre l'application HTTP et TCP; http+ssl=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

Pare-feux

- Filtrage des flux de communication / réseau



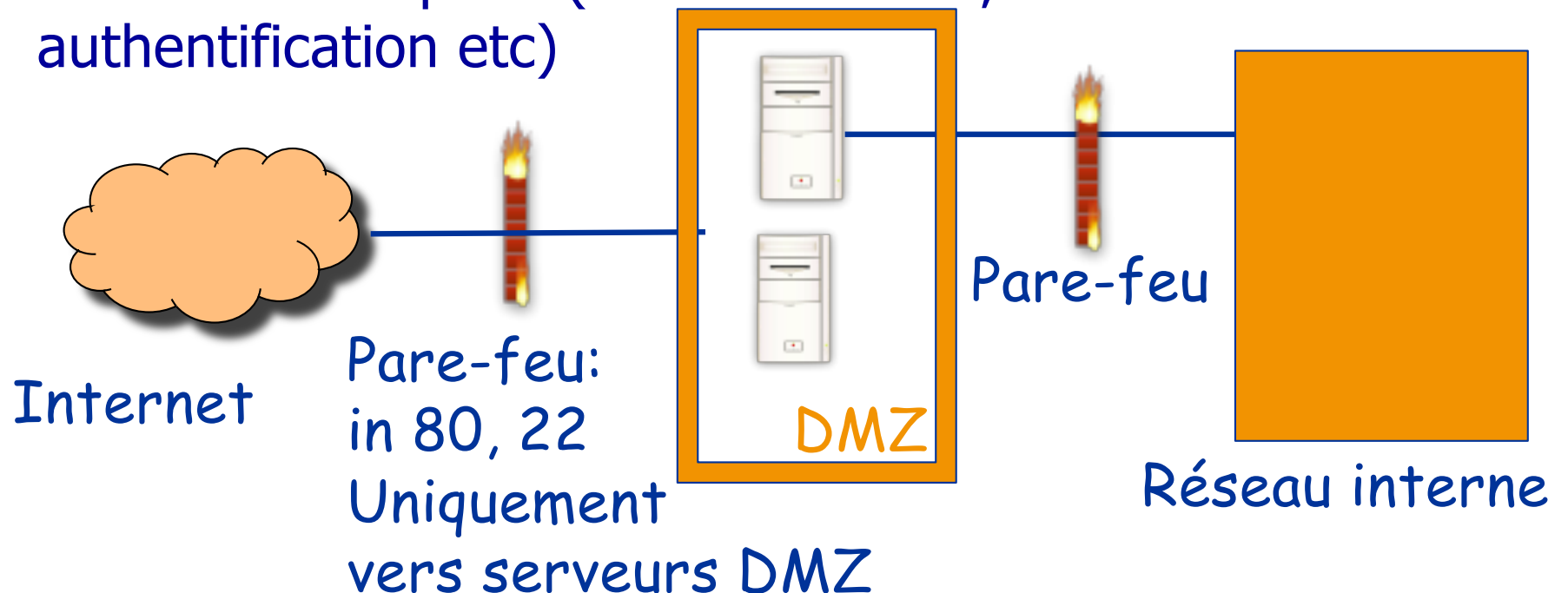
- Flux définis par adresses: niveau 3 (IP) ou 4 (ports) le plus souvent

```
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)



Bilan chapitre AP_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
- Principe de TLS
- Pare-feux