# Infrastructure de sécurité des réseaux

Rushed Kanawati
Département R&T, IUT de Villetaneuse
http://lipn.fr/∼kanawati
rushed.kanawati@lipn.univ-paris13.fr

Module M4210

February 18, 2015

# Plan

# Organisation du cours

**Module M4210** **30h**

- 4 séances de cours
- 3 TD
- 4 TP
- Un contrôle court (1h) / Un contrôle long (3h)
- **Contrôle continu**

# Problématique

## Problèmes de sécurité

- Problèmes de gestion
  - Perte de données suite à des mauvaises manipulations,
  - Conséquences des incendies, catastrophes naturelles, etc.
  - Duplication, repartition des données, Journalisation, sauvegarde automatique
- **Problèmes liés à des *attaques* d'un tiers malveillant**

# Objectifs d'un système de sécurité

- **Confidentialité**
  *Seuls les utilisateurs ayant droit ont accès aux données*
- **Intégrité**
  *Seuls les utilisateurs ayant droit peuvent modifier les données*
- **Disponibilité**
  *Garantir l'accessibilité des données aux utilisateurs ayant-droit*
- **Authenticité**
  *Garantir d'avoir l'identité exacte de la source d'une action*
- **Non-répudiation**
  *Garantir que le destinataire (resp. émetteur) d'un message ne peut pas prétendre ne pas l'avoir reçu (resp. envoyé).*

# Le cycle PPR

- **PPR** : Planification → Protection → Réaction
- **Planification**: Analyses des menaces (montant des dommages, Probabilité, coût de la protection, priorité) → règles & règlement
- **Protection**: Sélection et installation des outils, mise à jour, audit (test et mise à l'épreuve).
- **Réaction**: Détection de l'incident, procédure de réaction, restauration et sanction, **retour d'expérience** et réparation de failles.

# Attaques

## Vocabulaires

- **Attaque** : Une action qui vise à compromettre la sécurité d'un système.
- **Intrusion** : Prise de contrôle, partielle ou totale d'un système distant.
- **Usurpation** : (spoofing) la prise d'identité d'autrui (utilisateur ou système) afin de gagner une accès illégitime à un système.

# Types d'attaques

- **Attaques non ciblées**
  - Attaques contre des proies aléatoires.
  - Utilisation de **maliciels** (malware)
  - Vecteurs de propagation : e-mail, sites web contaminés, SMS, etc.
- **Attaques ciblées**
  - Attaque contre une cible identifiée afin de compromettre une propriété de base de la sécurité de systèmes (intégrité, disponibilité, confidentialité, . . . , etc.)

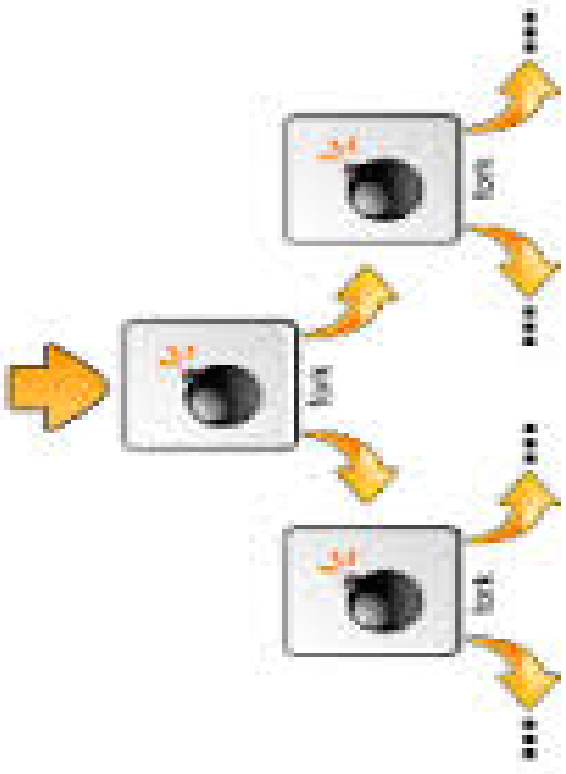
# MALICIELS : VIRUS

- Virus informatique : Un programme capable de se répliquer et conçu pour se propager à d'un ordinateur à un autre en s'insérant dans des logiciels légitimes, appelés **hôtes**.

# MALICIELS : VERS

- Vers : un programme capable de se répliquer et conçu pour se propager d'une manière autonome à travers les réseaux.

# MALICIELS : WABBIT



- Wabbit : un programme capable de se répliquer mais ne possède pas la capacité de propagation dans les réseaux.
- **:(){ :|: & };:**

# Maliciels : Cheval de Troie

- Cheval de Troie : Un maliciels qui prend l'apparence d'un logiciel légitime.
- Vecteur de propagation: sites de téléchargement !
- Souvent employé pour mettre en place d'autres maliciels : virus, vers, trappes, logiciels d'espionnage, etc.

# MALICIELS : TRAPPE (BACKDOOR)

- Trappe : un programme installé sur la machine victime et qui se connecte à la machine de l'attaquant pour lui donner un accès à la machine infectée.
- Installé par un développeur d'une application, ou par un autre maliciels (ex. cheval de Troie).

# MALICIELS : ESPIONS (SPYWARE)

- Logiciel espion : logiciel malveillant qui s'installe dans un ordinateur dans le but de collecter et transférer des informations sur l'environnement dans lequel il s'est installé, très souvent sans que l'utilisateur n'en ait connaissance.
- `Keylogger` : Matériel ou logiciel qui enregistre les touches frappées sur le clavier et les transmettre via les réseaux ou via des ondes électromagnétiques.

# MALICIELS : ATTAQUES PAR-E-MAIL

- **Hoax** (Canular) : Courrier électronique incitant le destinataire à retransmettre le message à ses contacts sous divers prétextes (bon sentiments, alarmes, légendes urbaines, etc.)

  `http://www.hoaxbuster.com/`
- **Spam** : courriers non sollicité qui encombrent le réseau.
- **hameçonnage** (phishing) : courrier dont l'expéditeur se fait généralement passé pour un organisme demandant au destinataire de fournir des informations confidentielles

# Attaques ciblées: buts 1/2

- **Attaques des systèmes**
  - **Interruption** : réduire la disponibilité d'un système

    *Déni de service (DoS)*
  - **Cartographie** : Reconnaitre les machines et les services actifs et les versions des systèmes d'exploitation

    *Balayage des machines et des ports*
  - **Modification de données**

    *DNS poisoning, ARP poisoning*

# ATTAQUES CIBLÉES: BUTS 2/2

- **Attaques des communications**
  - **Interception de messages**

    *Ecoute & analyse de traffic (snifing)*
  - **Modification, fabrication de messages**

    *Usurpation d'adresses IP*
  - **Vol de sessions**

# Attaques : Attaques DoS simples

## Ping de la mort

- Envoyer un paquet `ping` malformé (taille > taille max de 65 535)
- La plupart des systèmes jusqu'au 1998, ne peuvent pas traiter un tel paquet

## LAND

- Envoyer un paquet `SYN` malformé : mêmes adresses IP sources et destinations et mêmes ports
- Conséquence: blocage du système visé

## Teardrops

- Envoyer des fragments IP avec des informations erronées de décalage (offsets).

# ATTAQUES PAR SATURATION

## Rafales de SYN

- Envoyer une rafale de paquets `SYN`.
- La victime alloue une partie de ses ressources à chaque demande : on assiste à à un phénomène de saturation, voire de pannes

## Rafales d'Echo-Reply

- Envoyer un message `echo-request` avec @Source = @Victime, @Dest: mode diffusion
- Si la victime est reliée à un routeur opérant en mode diffusion, tous les ordinateurs du réseau répondent à la victime par des `echo-reply`.

# ARP POISONING

- Envoyer un paquet ARP avec @MAC: pirate et @IP: @victime
- Le trafic vers la victime sera d'abords adressée au pirate tant que la victime n'as pas communiqué avec la machine cible !
- Mise à jours sans relâche de la table ARP.

# Quelques outils

## nmap

- nmap: Network Mapper
- Logiciel de balayage de ports (tcp; udp, rcp)
- Interface graphique et API pour script
- Sera étudié en TP.

# Quelques outils

## scapy

- http://www.secdev.org/projects/scapy/
- API en Python pour la manipulation de paquets.
- Fonctions de construction, d'écoute et réactions.
- Logiciel de forage de paquets et de trames
- permet d'envoyer des paquets malformés, usurpation d'adresses (MAC, IP, ports)
- utilisé pour des tests et d'audits de politiques de sécurité
- Sera étudié en TP.

# Pare-feux

## Définition

Un équipement, logiciel ou matériel, chargé de contrôler l'échange de paquets entre le réseau protégé et l'Internet.

## Fonctions

- **Contrôle** : Gérer les connexions sortantes à partir du réseau local.
- **Sécurité** : Protéger le réseau interne des intrusions venant de l'extérieur.
- **Vigilance** : Surveiller/tracer le trafic entre le réseau local et internet.

# Pare-feux à filtrage statique

## Principe

- Utilisation d'un ensemble de règles de format : **SI-Alors**
- **Les règles sont évaluées d'une manière séquentielle** : La première règle applicable est exécutée !

  *Dans quelle ordre faut-il inscrire les deux règles : interdir les paquets* `SYN/FIN` *et accepter les connexion tcp à un serveur donné ?*
- Les règles de filtrage ne sont pas symétriques
- Le filtrage est basé sur les informations contenues dans les entêtes des paquets/trames

  Adresses src/dest, drapeaux, et numéros de port
- Nécessité de définir une politique par défaut

  *ex. Bloquer tout !*

# Pare-feux à filtrage statique

## Filtrage en sortie : règles communes

- Bloquer l'émission des paquets IP dont l'adresse source n'est pas une adresse de l'organisme !

  *Eviter à un pirate interne ou externe d'envoyer des paquets d'attaques*
- Filtrage des paquets ICMP (sauf ECHO-request)

  *Pourquoi ?*
- Filtrage de segments RST
- Filtrage de ports de serveurs
- Autorisation des connexions clientes

  *Numéro de ports source* $\in [49152, 65536]$

# Pare-feux à filtrage statique

## Limites

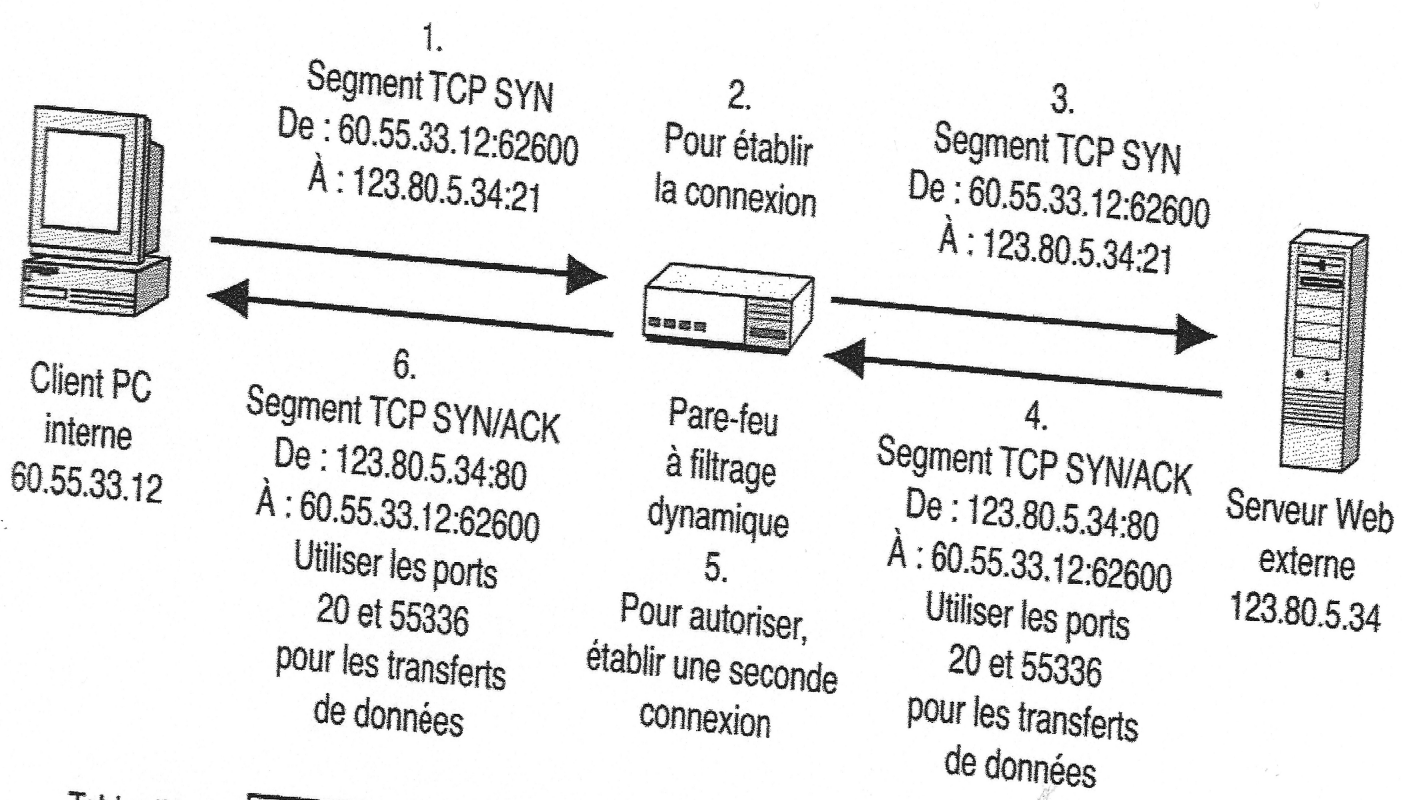
- Pas de prise en compte de l'état d'une session
- Faute rejeter un segment entrant `SYN/ACK` ?
- Difficile d'interpréter le trafic avec commutation de ports (cas de FTP par exemple).

# Pare-feux à filtrage dynamique

## Principe

- Le pare-feu maintiens une **table de connexions** ouvertes (TCP)
- Autorisation des passages de paquets pour les sessions ouvertes
- Extension du concept de session aux protocoles non-connectés (UDP, ICMP)
- Prise en compte de commutation de port dans certains cas (ex. ftp)

# Pare-feux à filtrage dynamique




| Type | Adresse IP interne | Numéro de port interne | Adresse IP externe | Numéro de port externe | État |
|---|---|---|---|---|---|
| TCP | 60.55.33.12 | 62600 | 123.80.5.34 | 21 | OK |
| TCP | 60.55.33.12 | 55336 | 123.80.5.34 | 20 | OK |

# Pare-feux: Translation d'adresses

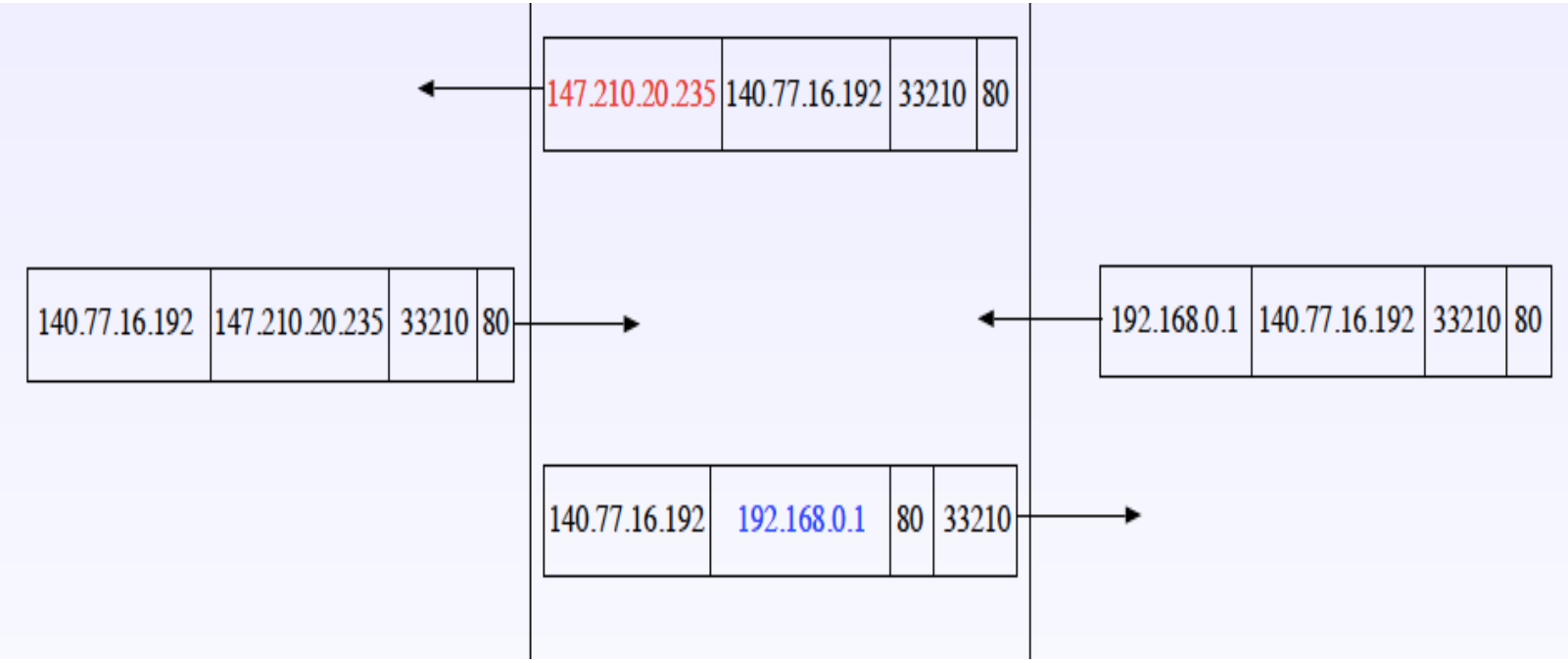
## NAT : Intérêt

- Gérer la pénurie d?adresses au sein d'un réseau
- Masquer l'intérieur du réseau par rapport à l'extérieur.
  *Changements d'adresses IP et des numéros de ports utilisés ← limiter l'intérêt d'écoute du trafic !*
- Faciliter la modification de l'architecture du réseau interne

## NAT : Types

- **Statique**: $n$ @-publiques $\leftrightarrow$ $n$ @ privées
- **Dynamique**: 1 @-publique $\leftrightarrow$ $n$ @ privées

# NAT STATIQUE



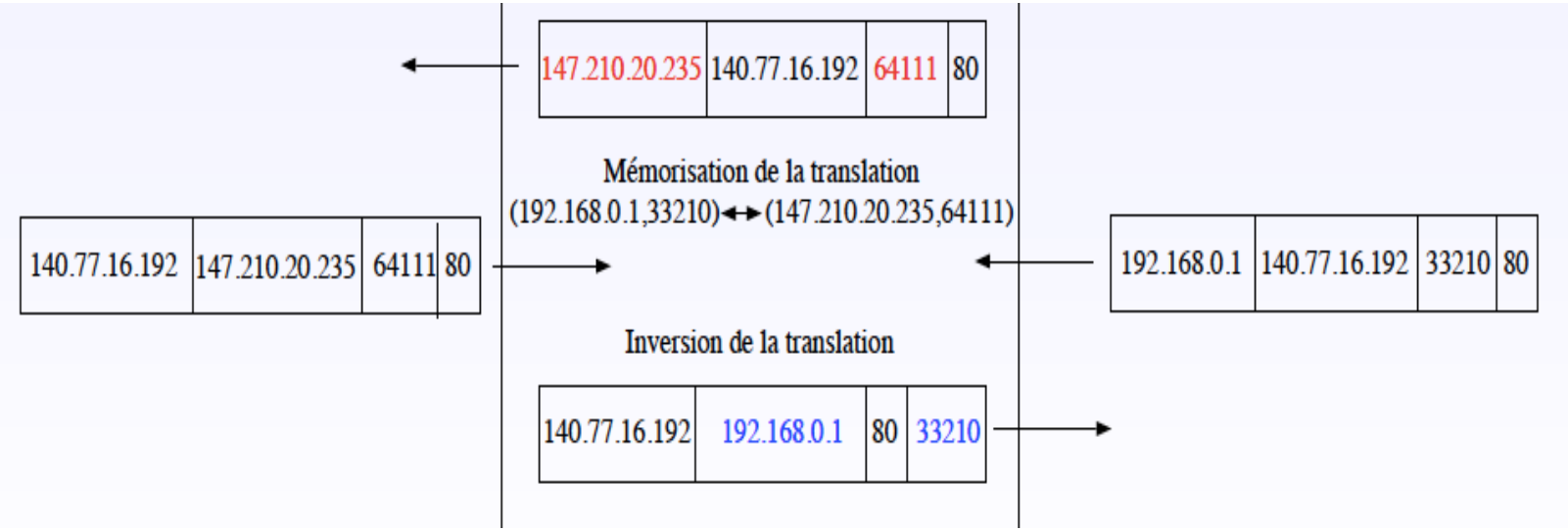

## NAT : Advantages/inconvenients

+ Facilité de mise en œuvre

+ Sécurité de l'échange

- Non résolution du problème de pénurie d'adresses.

# NAT DYNAMIQUE




## NAT : Advantages/inconvenient

Nécessité d'implémenter une méthode spécifique aux protocoles non-connectés (ex/ identifiant ICMP).

Nécessité de faire de la redirection de port pour rendre les machines internes joignables : Toutes les connexions entrantes sur un port donné sont redirigées vers une machine du réseau privé sur un port

# IPTABLES : LOGICIEL DE FILTRAGE DE PAQUETS

- Module du noyau Linux réalisant le filtrage de paquets
- Filtrage statique (sans prise en compte de contexte de session) ou dynamique
- Fonctionnement : La transition d'un paquets dans différentes *chaines*, A chaque chaine on peut exercer un contrôle en appliquant des règles contenues dans des tables
- Différent types de tables :

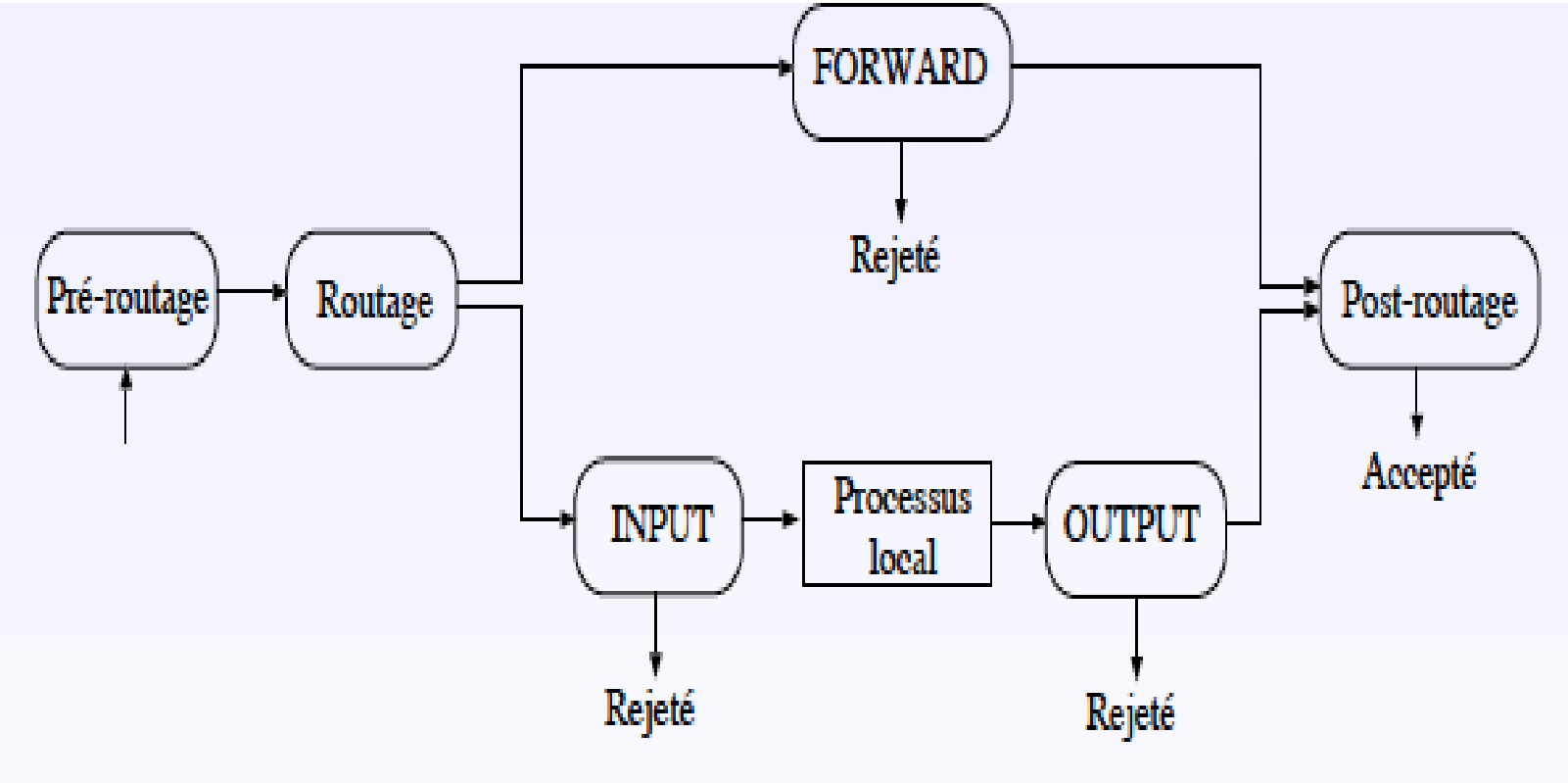
  `filter` : règles de filtrage de paquets

  `nat` : règles de translation d'adresse

  `mangle` : règles de modification des entêtes.

  `raw` (ou conntrack)

# IPTABLES




Tous les paquets émis par des processus locaux au routeur traversent la chaîne OUTPUT.

# Circuit des paquets générés par la machine

1. raw OUTPUT
2. mangle OUTPUT
3. nat OUTPUT
4. filter OUTPUT
5. mangle POSTROUTING
6. nat POSTROUTING

# Circuit des paquets en transit

1. raw PREROUTING
2. mangle PREROUTING
3. nat PREROUTING
4. mangle FORWARD
5. filter FORWARD
6. mangle POSTROUTING
7. nat POSTROUTING

# COMMANDES IPTABLES

## Syntaxe

```
iptables [-t table] command [match] [target/jump]
```

## Principales Commandes

- **-L** : affichage
  ```
  iptables -t nat -L PREROUTING
  ```
- **-P**: Politique par défaut
  ```
  iptables -P INPUT DROP
  ```
- **-A** : ajout d'une règle
  ```
  iptables -A INPUT -s 193.48.143.10 -j ACCEPT
  ```
- **-D** : effacer une règle
  ```
  iptables -D INPUT -s 193.48.143.10 -j ACCEPT
  ```
- **-F** effacer toutes les règles d'une table
  ```
  iptables -F
  ```

# IPTABLES : CRÉATION DE CHAINES

- **-N** : création d'une chaîne
- `iptables -N LOGACCEPT`
- `iptables -A LOGACCEPT -j LOG --log-prefix "LOGACCEPT : "`
- `iptables -A LOGACCEPT -j ACCEPT`

# IPTABLES: EXEMPLES

- Politique par défaut, jeter tous les paquets entrants
- `iptables -A INPUT -s 0/0 -i eth0 -d 192.168.1.1 -p TCP -j ACCEPT`
- Accepter tous les paquets destinés à l'adresse du routeur 192.168.1.1.
- `iptables -A FORWARD -s 0/0 -i eth0 -d 192.168.1.58 -o eth1 -p TCP -sport 1024:65535 -dport 80 -j ACCEPT`
- ?
- `iptables -A INPUT -p icmp -icmp-type echo-request -m limit -limit 1/s -i eth0 -j ACCEPT`
- Accepter un paquet `echo-request` par seconde

# IPTABLES: NAT

- `iptables -t nat -A POSTROUTING -o eth1 -s 192.168.0.0/24 -j MASQUERADE`
- Association entre toutes les adresses privées du sous-réseau 192.168.0.0/24 avec l'interface eth1.
- `iptables -t nat -A POSTROUTING -o eth2 -s 192.168.1.0/24 -j MASQUERADE`
- Association entre toutes les adresses privées du sous-réseau 192.168.1.0/24 avec l'interface eth2.
- `iptables -t nat -A PREROUTING -p tcp -i eth0 -d 140.77.13.2 -dport 80 -sport 1024:65535 -j DNAT -to 192.168.0.200:8080`
- Transférer les connexions sur le port 80 de l'adresse 140.77.13.2 sur la machine ayant l'adresse privée 192.168.0.200 sur le port 8080

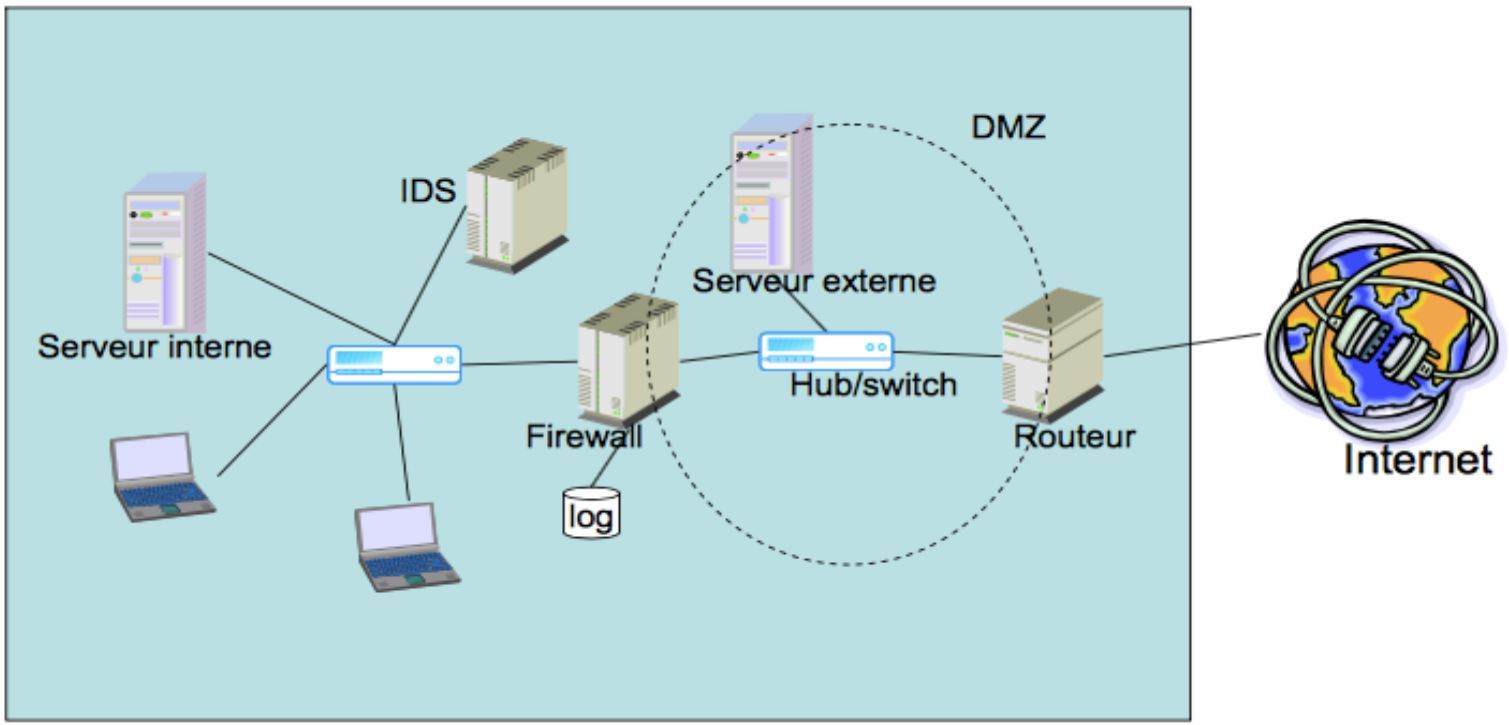
# Filtrage dynamique

- Suivi des connexions disponible (conntrack)
- 4 états possibles d'une connexion : NEW, ESTABLISHED, RELATED, INVALID
- `iptables -A OUTPUT -o eth0 -m state -state ESTABLISHED,RELATED -j ACCEPT`
- Autoriser tous les paquets émis par le routeur concernant des connexions déjà établies.

# DMZ

- Une zone démilitarisée (DMZ) est un sous-réseau se trouvant entre le réseau local et le réseau extérieur.
- Les connexions à la DMZ sont autorisées de n'importe où.
- Les connexions à partir de la DMZ ne sont autorisées que vers l'extérieur.
- intérêt : mettre en place des serveurs publiques : DNS, SMTP, WEB, etc

# DMZ


DMZ
IDS
Serveur externe
Serveur interne
Hub/switch
Firewall
Routeur
log
Internet

# IDS: Système de détection d'intrusions

**Définition**

Un équipement, logiciel ou matériel, qui automatise le processus d'analyse des événements d'un ordinateur/réseau pour y détecter des signes de problèmes de sécurité.

- un IDS détecte **mais n'empêche pas** les attaques
- Recherche de **signature d'attaque** : **motifs spécifiques** indiquant une intention suspecte.

# IDS : CLASSIFICATION

## Trois Critères

- **Architecture** :

  Centralisé
  Hiérarchique
  P2P

- **Mode de fonctionnement**

  Batch: analyse de log d'événements d'une manière périodiques ou programmé.
  Temps réel : analyse et détection en continue.

- **Type de détection**

  Détection basée sur des scénarios
  Détection d'anomalies

# Critères d'évaluation

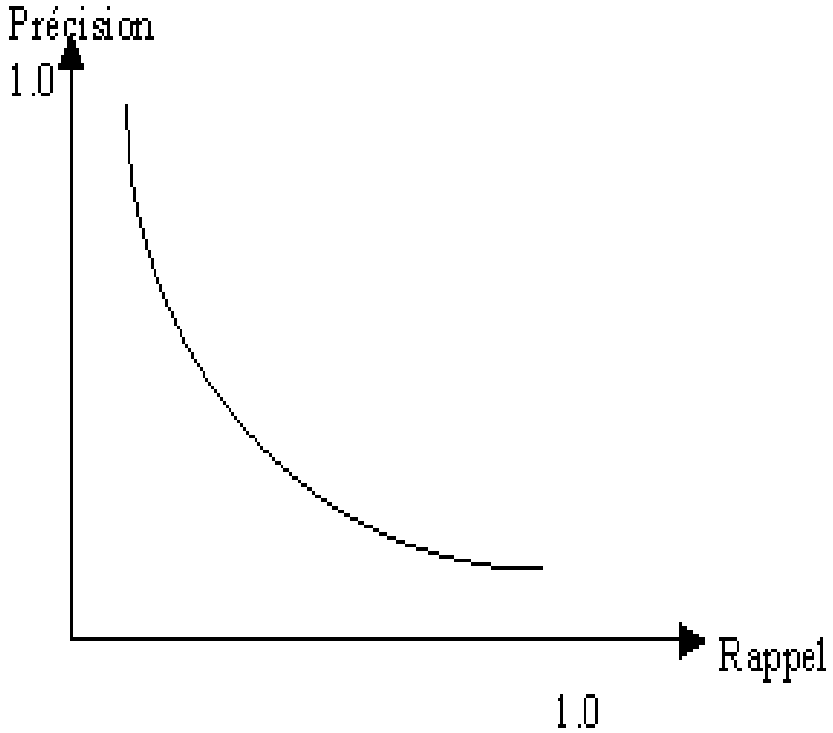
Table: Table de contingences

| | Attaque | Non-attaque |
|---|---|---|
| Attaque | VP | FP |
| Non-attaque | FN | VN |

## Trois Critères

- Précision : $P = \frac{VP}{VP+FP}$
- Rappel : $R = \frac{VP}{VP+FN}$
- F1 = $\frac{2 \times P \times R}{P+R}$

# Courbe Précision / Rappel



# Détection par scénarios

- Connaissance préalable des scénarios d'attaque, dont les occurrences sont détectées
- Fonctionnement : utiliser un ensemble de sondes pour générer des flus d'événements et recherche de motifs dans les événements à *matcher* avec une base de signatures d'attaques.
- Langage de description d'attaques: STATL (state-based Intrusion detection Language)

  Abstraction du détail de l'attaque
  Représentation d'attaques sous forme d'automates (états et transitions)
- Avantages: faible taux de faux négatif
- Inconvénients : Ne détecte que les attaques connues

# DÉTECTION D'ANOMALIES

- Principe : Apprendre un *profil* des comportements *normaux* et mesurerai la *déviation* du trafic observé par rapport au profil appris.
- Mesure et techniques utilisés : Seuils de détection, mesures statistiques, */dots*, etc.
- **Avantages** :

  Détecter les symptômes sans comprendre l'attaque.

  Alimenter une base de signature d'attaques
- **Inconvénients** :

  Génération de beaucoup de faux positives (fausses alarmes)

  Nécessite la disponibilité d'une base d'apprentissage fiable.

# NIDS: IDS orienté réseau

- Système d'coute et d'analyse du trafic sur un réseau
- Souvent relié à un switch
- cas de beaucoup de IDS commerciaux
- Fonctionnement : en trois phases

  Filtrage du traffic : éliminer les flux peu important

  Module de reconnaissance : signature, anomalie, etc.

  Module réaction : Notification, Alerts, **trap SNMP**, action

# NIDS: Avantages & Inconvénients

## Avantages

- Supervision de tout un réseau
  *pas besoin d'installer des IDS orienté hôte sur chaque machine*
- Déploiement sans perturbation de l'architecture réseau
- Indépendant des OS employés sur les différentes machines
- Détection en temps réel

## Inconvénients

- Faible performances en cas de traffic intense
- Non opérationnel avec les flux chiffrés
- Difficulté à traiter les fragments IP.

# HIDS: IDS orienté Hôte

- **Principe** : Analyse des logs des événements affectant la machine supervisée
- Source des données :

  System log

  Ecoute des ports de la machine → envoi d'allant si un port est utilisé

  Analyse de l'utilisation des ressources de la machine (base de registres, espace disque, mémoire, etc.)
- **Avantages** : Compatible flux crypté, pas d'ajout d'équipements, vérification de l'attaque
- **inconvénients** : OS dépendent, utilisation des ressource des machines, ps de détection en temps réel (analyse des log).

# Outil IDS


- Snort: outil logiciel libre compatible avec les principales OS (linux, UNIX, MAC OS X, windows)
- `https://www.snort.org`
- Type: NIDS
- détection en temps réel

# SNORT: FONCTIONS

- Opère au niveau 3 et 4 (IP, ICMP, UDP et TCP)
- Détection d'anomalies

  *paquets ICMP invalides*
- Pré-porcessuer HTTP
- Détection d'attaques de type dénis de service, saturation
- Langage de règles simple
- Règles parentérales (utilisation de variables de substitution)
- Importation de règles

# SNORT: LES RÈGLES

## Syntaxe

```
action protocole @IP-src sport direction @IP-dest sport options
```

- Action : alert, log, pass
- protocole : tcp udp icmp.
- Options : **msg, flags, ttl, offset, seq, ack, minfrag, content, . . . ,**

  `minfrag` : permet de fixer un seuil de taille minimale pour un fragment

  `content` : permet de rechercher un contenu spécifique dans le champ donnée

# SNORT: EXEMPLES DE RÈGLES

- `alert tcp any any -> 192.168.1.0/24 any (flags:SF; msg:"Scan SYN FIN");`
- `alert tcp any any -> 192.168.1.0/24 21 (content: "USER root"; msg: "Tentative d'accès au FTP pour l'utilisateur root";)`
- `log udp any any -> 192.168.1.0/24 any`
- `log 193.50.60.0/24 any <> 194.78.45.0/24 any`

# SNORT: RÈGLES AVANCÉS

- Variables de substitution :

  `var Mynetwork 192.168.1.0/24`

- Inclusion de fichiers de règles externs

  `include:<fichier>`

- Sites publiques pour règles snort.

  **voir** `www.snort.org`

# AUTHENTIFICATION

- **Définition** : Vérification de l'identité d'une *entité* afin de lui autoriser l'accès à des ressources.
- Entités à authentifier: utilisateurs, processus, machines
- Types d'authentification :

  Simple : Exiger un seul élément d'authentification (ex. mot de passe)

  Forte : Exiger au moins deux éléments (ex. carte à puce).

  Mutuelle : exiger une authentification dans le deux sens.

# Cryptographie

## Vocabulaires

- **Cryptographie** : science de secret
- **Message en clair** : message originale
- **Chiffrement** : transformation à l'aide d'une clé de chiffrement d'un message en clair en un message incompréhensible si on ne dispose pas d'une clé de déchiffrement
- **Cryptogramme** : message chiffré

# Cryptographie: principes

- Deux grandes approaches :

  **Cryptage symétrique** : utilisation d'un clé de cryptage partagé

  **Cryptage asymétrique** : Codage avec deux clés : une privée et l'autre publique.

- **Principe de Kerckhoffs**: (*Maxime de Shanon*) : L'ennemi connaît l'algorithme utilisé, donc le secret repose sur le secret de clé et non sur le secret de l'algorithme.

# Cryptographie : services

- Confidentialité
- Authentification de l'origine des données
- Intégrité
- Non-répudiation
- Non-rejeux
- Authenticité = Authentification + Intégrité

# Algorithmes

## Types

- Symétrique / Asymétrique
- En continu / par bloc

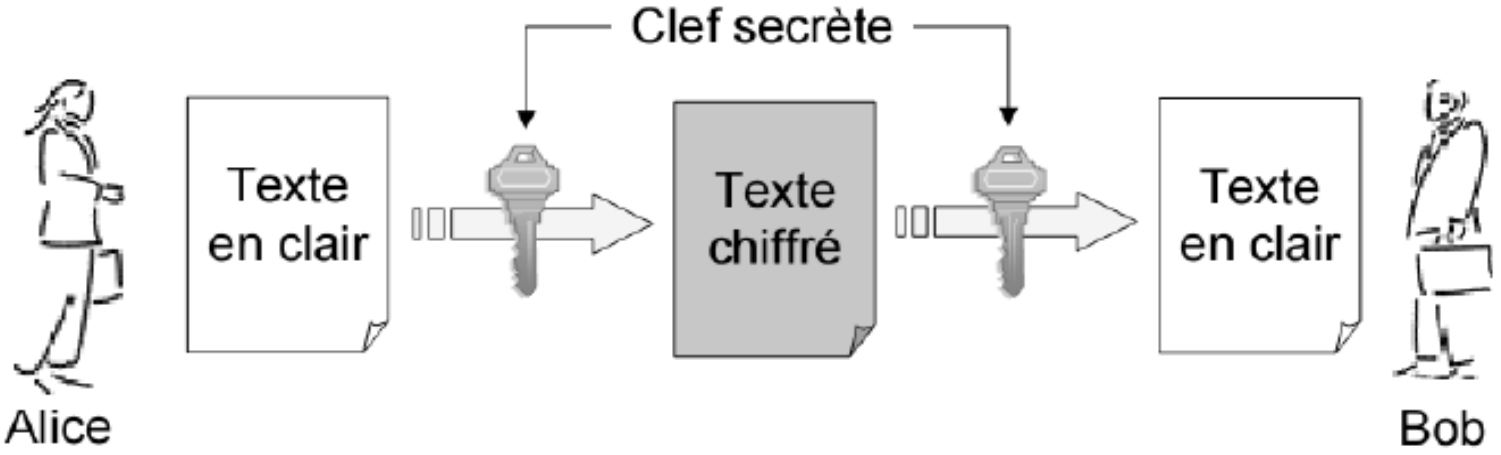
## Algorithme en continu

- Modification bit à bit
- Rapides et robustes aux erreurs de communications
- Exemple: `RC4`: (clé à 128 bits)

# Algorithme en blocs

- Le message est découpé en blocs $B_i$ de taille fixe.
- Chaque bloc est chiffré de manière corrélée avec le bloc précédent en utilisant l'opération `XOR` entre le bloc de message $B_i$ et le résultat du chiffrement de $B_{i-1}$.
- Initialisation : Utilisation d'une graine (i.e. seed) aléatoire pour le chiffrement de $B_1$.

# CHIFFREMENT SYMÉTRIQUE




*Noms des utilisateurs*

**A**lice, **B**ob & **C**harlie !

# Chiffrement symétrique

- Approches : Substitution et Permutation
- Avantages : rapidité, taille réduits des clés
- Inconvénients : la distribution de la clé doit être confidentielle
- Explosion de nombre de clés pour des communication à $n$ utilisateurs !
- Exemples : DES, Kerberos, IDEA.

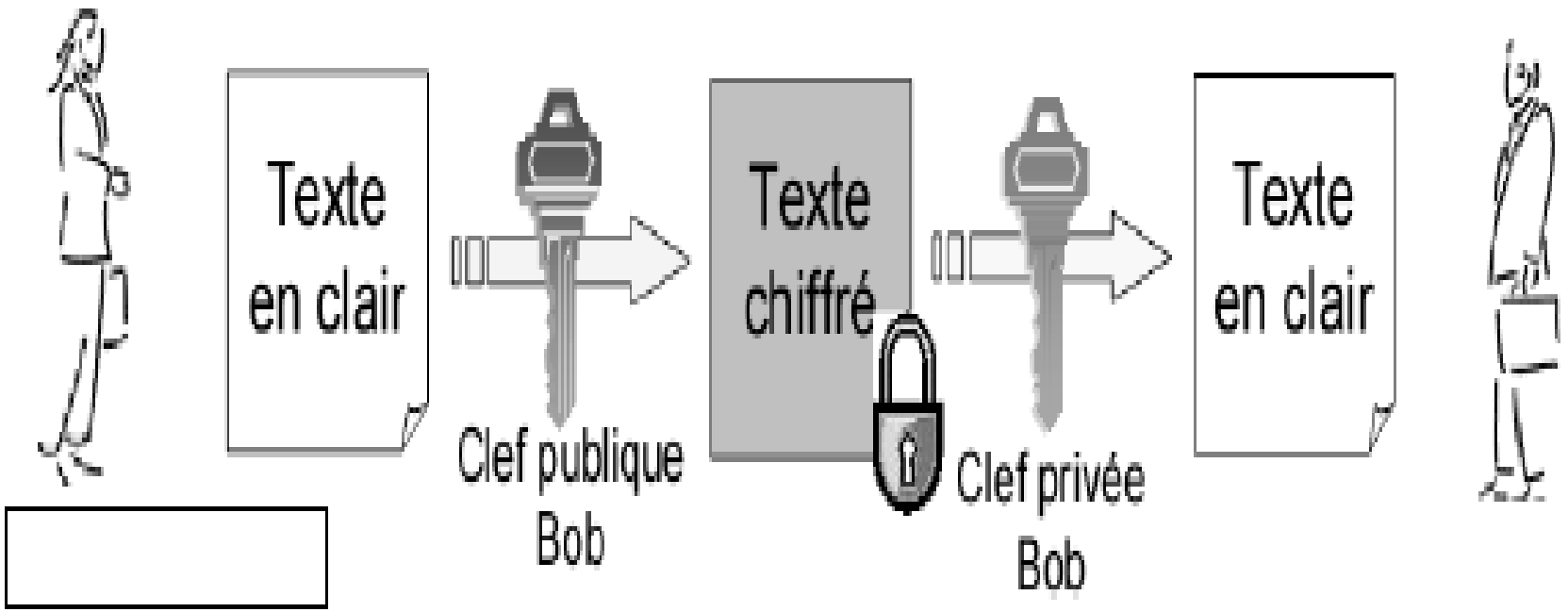
# Chiffrement asymétrique

- Principe : Clé de chiffrement (publique) est différentes du clé de déchiffrement (privée)
- La clé de déchiffrement est difficilement calculée à partir de la clé de chiffrement.
- Seul le détenteur de la clé de déchiffrement (clé privée) peut alors déchiffrer un message chiffré avec la clé publique correspondante

  *Confidentialité des échanges*
- Algorithmes lents et nécessitent beaucoup de ressources
- Exemple : RSA (Riverst-Shamir-Adleman), DSA (Digital Signature Algorithm)

# CHIFFREMENT ASYMÉTRIQUE : UTILISATION

## Confidentialité



# CHIFFREMENT ASYMÉTRIQUE : UTILISATION

## Authentification