FICHE DE RÉVISION — INTER-CONNEXION ET SÉCURISATION

TSSR — PKI, chiffrement, HTTPS/TLS, hachage, pare-feu et VPN

1. Définitions essentielles

Notion	Définition
PKI	**Public Key Infrastructure** : infrastructure qui permet de **créer des clés publiques**. Les clés publiques interviennent dans ce qu'on appelle le **chiffrement**.
Chiffrement	Cryptage, encryption : rendre les données **non lisibles** sans la bonne infrastructure. En clair « Bonjour », résultat chiffré « dgkuzegfizue ».
Algorithme de chiffrement	La **méthode mathématique** utilisée pour chiffrer / déchiffrer.
Clé de chiffrement	La **valeur secrète (ou publique)** passée en entrée de l'algorithme pour produire un résultat unique.
Taille de clé	Une propriété **de la clé**, pas la clé elle-même : AES-128 → 128 bits, AES-256 → 256 bits, RSA-2048 → 2048 bits. Plus la clé est longue, **plus la casser par brute force est coûteux**.

2. Le chiffrement symétrique

2.1 Algorithme à clé fixe — Jules César

On décale chaque lettre à la lettre suivante de l'alphabet. **Clé : +1**

```
B O N J O U R   devient   C P O K P V S
```

2.2 Le problème des séquences répétées

Quand on envoie des données, on envoie des **données chiffrées**. Ce que les hackers cherchent, ce sont **des séquences qui se répètent**, afin de décrypter plus vite. Pour éviter ça, on utilise un **algorithme variable avec un mot code**, qui oblige à éviter la redondance de certains caractères.

2.3 Algorithme à mot code

L'algorithme décale la lettre de l'alphabet **en fonction du mot code**. **Clé : PSG**

```
B O N J O U R          <= Mot clair
P S G P S G P S G      <= Mot code
R H U Z H B H          <= Chiffrage
```

On prend la position de la lettre `P` et on l'ajoute à `B`, on obtient `R`, et ainsi de suite.

Le plus important dans le chiffrement, c'est LA CLÉ. Il est extrêmement important d'avoir une clé solide.

3. Le chiffrement asymétrique (clé privée + clé publique)

Le point de départ : si on veut chiffrer des données, il faut un **échange de clé**. Et cet échange ne peut pas prendre le risque d'être intercepté — d'où la création d'un autre type de clé.

3.1 Concrètement — Bob veut envoyer un message secret à Alice

Étape	Ce qui se passe
1	Alice **génère sa paire de clés**
2	Alice envoie sa **clé publique** à Bob — elle peut se balader sur le réseau, **aucun risque**
3	Bob chiffre son message avec la **clé publique d'Alice**
4	Alice déchiffre avec sa **clé privée** — **elle seule** peut le faire

```
Bob ──[clé publique Alice]──► chiffre ──► réseau ──►
          déchiffre ◄──[clé privée Alice]── Alice
```

> *Même si un hacker intercepte le message chiffré ET la clé publique, il ne peut rien faire : il lui manque la clé privée.*

3.2 La faille : la contrefaçon de clés

On peut créer des clés publiques **et les contrefaire**. Un attaquant intercepte les clés sur le réseau et les contrefait en se fabriquant lui-même une clé publique **et** la clé privée correspondante, ce qui lui permet de lire et donc de **contrefaire**.

Il nous faut donc un TIERS DE CONFIANCE, et ce tiers de confiance, c'est le CA.

4. CA — Certificate Authority (Autorité de certification)

Son boulot, c'est de **créer les clés publiques**. Quand un demandeur sollicite la CA, celle-ci fait **d'abord une enquête d'identité** avant de créer la clé. La clé est ensuite **validée par l'autorité de certification**.

4.1 Les deux formats de fichiers

Format	Contenu
`.CER`	**certificat numérique** : contient la **clé publique** + l'identité + la **signature de la CA**
`.PFX`	**certificat + clé privée**, les deux ensemble, protégé par **mot de passe**

On envoie uniquement le .CER. JAMAIS le .PFX.

> *Le .PFX contient la clé privée. Si quelqu'un met la main sur ce fichier, il peut déchiffrer toutes les communications et usurper l'identité du propriétaire. Le mot de passe est la dernière ligne de défense si le fichier est volé. On ne l'exporte que dans des cas très précis (migration de serveur, sauvegarde), et jamais par email.*

4.2 Ce que l'ordinateur vérifie sur un certificat

Vérification	Question posée
La provenance	qui a émis ce certificat ?
Le nom	la correspondance du nom est-elle exacte ?
La date de validité	le certificat est-il encore dans sa période de validité ?
La non-révocation	est-ce que la clé publique est encore valable ?

> *Moyen mnémotechnique : clé privée = nombre premier, clé publique = résultat des nombres premiers.*

5. HTTPS et TLS

Le chiffrement **asymétrique** est **beaucoup plus gourmand** que le chiffrement symétrique. HTTPS s'appuie sur le chiffrement asymétrique, et se fait également **par échange de clés** :

```
La clé publique (www.fff.fr)(CA).CER
La clé privée  (www.fff.fr)(CA).PFX
```

Le protocole HTTPS utilise le chiffrement ASYMÉTRIQUE pour chiffrer la clé de chiffrement SYMÉTRIQUE. C'est du double chiffrement.

5.1 Les deux étapes

- **Étape 1** — télécharger la **clé publique** du site internet
- **Étape 2** — utiliser cette clé publique pour **chiffrer la clé de chiffrement symétrique** (le mot clé `FFFCHAMPION3ETOILES`)

```
1. Asymétrique ⟶ échange sécurisé de la clé symétrique  (la SÉCURITÉ)
2. Symétrique  ⟶ chiffrement de toutes les données      (la VITESSE)
```

5.2 SSL ou TLS ?

Tout ce protocole s'appuie sur **SSL (obsolète) / TLS**. Le **TLS (Transport Layer Security)** est le successeur de SSL, et c'est lui qui est utilisé aujourd'hui. En pratique tout le monde dit encore « SSL » par abus de langage — même les certificats s'appellent encore « certificats SSL » dans le commerce — mais **techniquement c'est TLS qui tourne**.

Protocole	Statut
SSL 2.0 / 3.0	❌ Obsolète
TLS 1.0 / 1.1	❌ Déprécié
TLS 1.2	☑ Encore utilisé
TLS 1.3	☑ Standard actuel

6. PKI — résumé complet

Composant	Rôle
Clé privée (`.PFX`)	certificat + clé privée, **déchiffre**, ne quitte jamais le serveur
Certificat (`.CER`)	contient la **clé publique** + identité + signature CA
CA (Certificate Authority)	**génère, signe et révoque** les certificats
Enquête d'identité	vérifie que le demandeur est **bien qui il prétend être**
Révocation	mécanisme pour invalider un certificat compromis ou expiré (**CRL / OCSP**)
TLS (ex-SSL)	protocole qui **combine asymétrique + symétrique** pour sécuriser les échanges

6.1 Cycle de vie du certificat

```
Demande ⟶ Enquête d'identité ⟶ Génération ⟶
Distribution ⟶ Vérification ⟶ Révocation
```

7. Le hachage — à ne pas confondre avec le chiffrement

HACHAGE ≠ CHIFFREMENT. Le hachage est un algorithme mathématique À SENS UNIQUE.

Il transforme une donnée de **taille arbitraire** en une **empreinte numérique de taille fixe** (le **hash** ou **digest**), dont la longueur dépend de l'algorithme choisi.

Entrée (SHA256)	Empreinte produite
un fichier de 100 Mo	**256 bits**
un fichier de 1 To	**256 bits**

Pourquoi le hachage : pour **vérifier qu'une donnée n'a pas été altérée**, sans avoir besoin de connaître la donnée elle-même. Dans les 256 bits, si un seul bit `0` est transformé en `1`, la donnée ne sera **plus du tout la même**.

> *La longueur de la clé après l'algorithme est une des raisons pour lesquelles on doit mettre à jour nos navigateurs (dans notre exemple : RSA-2048).*

8. Demander un certificat depuis IIS — les trois voies

Méthode	Quand l'utiliser
Créer un certificat de domaine	l'AC est **membre du domaine AD**. IIS lui envoie la demande directement et récupère le certificat signé **automatiquement** — tout se fait en interne, de façon transparente
Créer une demande + Terminer la demande	l'AC est **externe au domaine** (AC publique type DigiCert / Let's Encrypt, ou AC interne isolée). Deux temps : la demande génère un fichier **CSR** (Certificate Signing Request) envoyé à l'AC, puis « Terminer la demande » importe le certificat signé renvoyé par l'AC
Certificat auto-signé	IIS génère lui-même son certificat, **sans AC**. Personne ne le reconnaît comme de confiance : les navigateurs affichent une alerte de sécurité, **et à raison**

Le certificat auto-signé n'est utilisable QU'EN LABO. En production, aucun tiers de confiance n'a validé l'identité du serveur : c'est non.

8.1 Correspondance exacte du nom

La correspondance du nom doit être PARFAITE : www.dompa.lan doit être IDENTIQUE au nom du site web.

Après création, penser à `gpupdate /force` pour faire remonter la PKI dans nos certificats.

8.2 Repérer et sauvegarder la clé privée

Dans la MMC du serveur web, sous **Personnel**, le certificat apparaît **avec une petite clé** : c'est le signe qu'il embarque une **clé privée**.

> *Les clés privées doivent être sauvegardées. Pour ce faire : clic droit > Toutes les tâches > Exporter.*

9. Key Usage — restreindre ce que la clé a le droit de faire

Un certificat contient une clé publique qui peut **techniquement tout faire**. Le **Key Usage** est ce qui **restreint ses permissions** à ce qu'elle a le droit de faire — exactement comme des droits NTFS.

Sans ça, un certificat serveur HTTPS pourrait théoriquement **signer d'autres certificats**, ou faire des choses qu'il ne devrait pas — ce qui est une **faille de sécurité**. Pour un serveur exposé à l'extérieur, on ajoute **deux options** :

Option	Signification
Signature numérique	prouver son identité — pendant le handshake TLS, le serveur **signe des données** pour prouver qu'il détient bien la **clé privée** associée au certificat
Chiffrement de la clé	chiffrer l'échange de clé

On coche UNIQUEMENT ce dont on a besoin. Principe de MOINDRE PRIVILÈGE, comme partout en sécurité.

Usage	Objectif
Le chiffrement des données	rendre les données **illisibles**
La signature	identifier la personne — la **garantie de l'identité**

10. Certificat pour un serveur web hors du domaine

Deux méthodes vues en TP :

- **Depuis la CA du domaine, puis import** : sur SRVPA02, créer un certificat de domaine (`www.pacorp.fr`) → exporter ce certificat → le copier sur SRVPA05 → l'importer.
- **Depuis le serveur hors domaine (SRVPA05)** : créer une demande de certificat → enregistrer la demande → ouvrir un navigateur sur `http://AdresseIPDuServeurCA/certsrv` et se connecter en **administrateur du domaine**.

La seconde méthode, c'est comme ça que ça se passe dans la vraie vie.

> *Via la MMC, on peut faire une demande de certificat complète et élégante, qui fonctionne avec les navigateurs récents.*

11. Le pare-feu

L'idée de base est de **bloquer le trafic entrant**. Pour bloquer tout le trafic entrant, le pare-feu **bloque tous les ports** :

```
65 536 ports TCP
65 536 ports UDP
```

On doit donc créer des **règles de pare-feu** (des **exceptions**) : identifier le protocole, identifier le port utilisé par ce protocole, puis **ouvrir le port** (autoriser le flux entrant).
Sur les machines Windows, beaucoup de règles sont **déjà prédéfinies**. Les règles **actives** sont matérialisées par l'**icône verte**.

Quand on fait des ajouts de rôles et de fonctionnalités, certaines règles du pare-feu peuvent être activées ou désactivées TOUTES SEULES.

11.1 Le cas du ping

Le ping n'utilise PAS de port dédié. Il utilise ICMP, qui fait partie de la pile TCP/IP au même niveau que IP, TCP et UDP.

Deux solutions au choix :

- Créer une règle qui autorise la réponse **ICMPv4**

- Autoriser la règle déjà existante : **Partage de fichiers et d'imprimantes (Demande d'écho - Trafic entrant ICMPv4)**

> *Sur l'autel de la sécurité : les RSSI désactivent tous les pings.*

12. VPN (Virtual Private Network)

12.1 VPN Client — Serveur

Le VPN, c'est **établir une connexion réseau entre le serveur VPN et le client VPN** : c'est le **tunnel VPN**. L'intérêt du VPN, c'est qu'il **passe par Internet**. On a besoin de deux choses :

- le **serveur VPN** a accès à Internet
- le **client VPN** a accès à Internet

12.2 Le DHCP des connexions VPN

Choix	Conséquence
Utiliser un serveur DHCP existant	le VPN s'appuie sur le DHCP déjà en place dans le domaine
À partir d'une plage d'adresses spécifiées	c'est **le serveur VPN qui devient serveur DHCP** pour les connexions VPN. On aura donc **deux DHCP** dans le réseau : un pour le domaine, l'autre uniquement pour le VPN

> *En vérifiant dans le serveur DHCP, on constate que le serveur VPN a pré-réservé des adresses IP pour pouvoir les attribuer directement aux connexions VPN.*

12.3 Autoriser l'utilisateur dans l'AD

Il faut activer, dans la base des comptes AD, l'**Appel entrant** → **Autoriser l'accès**.

12.4 Le protocole de tunnel

Côté client, on crée une **nouvelle connexion VPN**, et pour cela il faut le **bon protocole de tunnel**. Il existe plein de protocoles différents.

Le serveur VPN et le client VPN doivent utiliser LES MÊMES PROTOCOLES.

Une fois le nom d'utilisateur et le mot de passe saisis, on peut se connecter au VPN.

12.5 Ce qui se passe réellement

Le tunnel **encapsule et chiffre** le trafic :

- **Chez le client** se crée une **nouvelle interface réseau virtuelle**, visible par l'OS, avec sa propre IP
- La **passerelle par défaut** est redirigée vers cette interface, pour faire passer tout le trafic dans le tunnel
- Tout le trafic IP sort donc **chiffré** vers le serveur VPN, qui le **déchiffre et le renvoie** sur le réseau interne

12.6 VPN Site à Site

Deux réseaux d'entreprise situés à deux endroits géographiquement différents : **deux sites distants**, deux réseaux locaux distincts, chacun avec son **serveur VPN en bordure de réseau**. Le tunnel est établi **entre les deux serveurs VPN**. Ce sont eux qui gèrent l'encapsulation, de manière **totalement transparente** pour les machines des deux côtés.

Tous les ordinateurs de l'entreprise A peuvent pinguer tous ceux de l'entreprise B : c'est comme si les serveurs VPN devenaient des ROUTEURS.

> *Dans la maquette, la connexion site à site a été montée via ZeroShell.*

13. Pièges et points à retenir pour l'ECF

HTTPS = double chiffrement. L'asymétrique sert UNIQUEMENT à transporter la clé symétrique ; le symétrique chiffre les données.

- **Algorithme** = la méthode mathématique. **Clé** = la valeur passée en entrée. Ne pas confondre les deux.
- Symétrique = **une seule clé**, rapide. Asymétrique = **paire publique/privée**, sûr mais **gourmand**.
- `.CER` = certificat + clé publique. `.PFX` = certificat + **clé privée**. On n'envoie jamais le `.PFX`.
- La CA fait une **enquête d'identité avant** de générer la clé — c'est ce qui en fait un **tiers de confiance**.
- L'ordinateur vérifie **4 choses** : provenance, nom, date de validité, non-révocation.
- **Hachage ≠ chiffrement** : sens unique, taille de sortie **fixe** quelle que soit la taille d'entrée. Sert à vérifier l'**intégrité**.
- Le **CN du certificat doit correspondre exactement** au nom du site (`www.dompa.lan`).
- Auto-signé = labo uniquement. En prod, alerte navigateur garantie.
- **Key Usage** = moindre privilège appliqué aux certificats : Signature numérique + Chiffrement de la clé, et rien d'autre.
- Pare-feu : **65 536 ports TCP + 65 536 ports UDP**, tout est bloqué en entrant, les règles sont des **exceptions**.
- Le **ping n'a pas de port** : c'est **ICMP**, au même niveau qu'IP, TCP et UDP dans la pile.
- VPN : le serveur **et** le client doivent parler **le même protocole de tunnel**.
- VPN client-serveur : nouvelle **interface virtuelle** côté client + **passerelle par défaut redirigée** dedans.
- VPN site à site : le tunnel est **entre les deux serveurs VPN**, transparent pour les postes.