INTRODUCTION À LA SÉCURITÉ ET VPN

Anthony Busson

Plan du cours

- Séances 1et 2:
  - Introduction à la sécurité et chiffrement
  - VPN
- Travaux pratiques:
  - TP openssl
  - SSH, tunnels et VPNs

Plan du cours G6S4A

- Cours: 8h
  - Cours de sécurité: 4/6h
  - Firewall et DNS: 2-4h
- TP: 12h
  - Openssl : 4h ou 6h
  - SSH: 2h
  - Firewall: 2h
  - DNS: 2h
  - SNMP: 2h (à voir)

PRINCIPE DE BASE

La sécurité

- La sécurité est multidisciplinaire
  - Normative
  - Organisationnelle
  - Législative
  - Technique
  - Méthodologique
  - Etc.
- Ce cours n'est qu'une introduction aux aspects techniques

Vocabulaires

- Les algorithmes de sécurité assurent des services qui sont:
  - Intégrité: les données émises n'ont pas été détruites ou modifiées
  - Confidentialité: maintien du secret des informations (protection des données contre une divulgation non autorisée)
  - Identification et Authentification: s'assurer de l'origine du message.
  - Non répudiation: l'émetteur du message ne peut pas contester qu'il est l'émetteur du message.

Exercice 1: applications

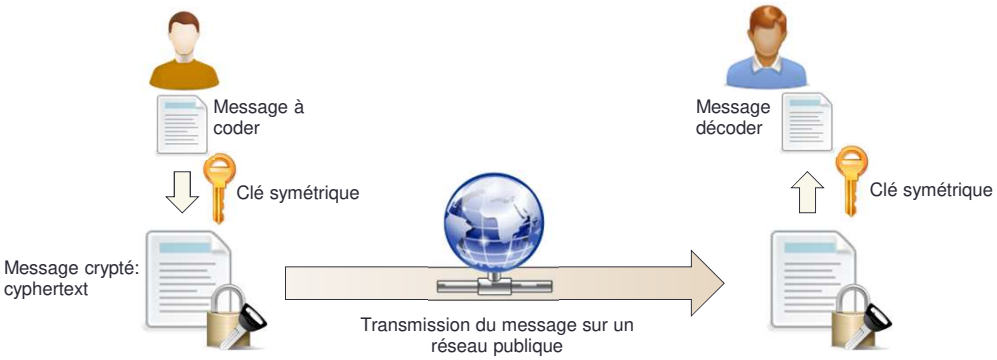
- Indiquez pour chacune des problématiques ci-dessous le service mis en jeu (intégrité, etc.).
  - Accès à un réseau Wi-Fi
  - Accès à mon serveur de mail d'entreprise (quand je suis dans l'entreprise)
  - Accès à mon serveur de mail d'entreprise (quand je suis à l'extérieur)
  - Je déclare mes impôts sur Internet
  - J'accède à un système d'accès (porte, etc.) par carte/système informatique
  - J'effectue un virement bancaire

CHIFFREMENT

Chiffrement symétrique

Chiffrement à clés symétriques

- Deux clés identiques (émetteur – récepteur)
- La même clé est utilisée pour coder et décoder



Exemple trivial: chiffrement

- XOR entre le mot de passe et les données à chiffrer
  - Mot de passe: toto
  - Texte à chiffrer: iut

Caractère	Ascii (décimal)	Ascii (binaire)
t	116	01110100
o	111	01101111
i	105	01101001
u	117	01110101

01110100 01101111 01110100 01101111 (toto)

XOR 01101001 01110101 01110100 (iut)

00011101 00011010 00000000

Exemple trivial: déchiffrement

- XOR entre le mot de passe et les données à chiffrer
  - Mot de passe: toto
  - Texte à chiffrer: iut

Caractère	Ascii (décimal)	Ascii (binaire)
t	116	01110100
o	111	01101111
i	105	01101001
u	117	01110101

00011101 00011010 00000000

XOR 01110100 01101111 01110100 01101111 (toto)

01101001 01110101 01110100 (iut)

Exercice 1: chiffrement symétrique

- On suppose un chiffrement utilisant un ou exclusif.
- Déchiffrer les mots suivants:

Mot chiffré	Mot de passe
060e1a0c1c	toto
1c0c171c0d	hier
0000051c11	titi
19041a1411	tata
0f00131917	beau
0819170c0f	jury
0b031b0f0d	iota
101a061d01	roux

Mot chiffré	Mot de passe
1e080a0217	rade
1f001b0617	sont
01001b0816	surf
000e160401	saxe
17151d0d01	dada
170e011700	date
06041c0102	grue

- En quoi à consister l'opération de chiffrement préalable?

Code ASCII

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(	72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29	)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-_	63	3F	?	95	5F	_	127	7F	DEL

Exercice 2: problème du chiffrement avec le mot de passe.

- Voici l'en-tête d'un fichier pdf: %PDF-1.4
- Voici les premiers champs d'un en-tête IP: Version (4bits), header length (4 bits), service type (8 bits)
- Quels problèmes cela pose-t-il avec l'algorithme précédent?
- Proposer une solution.

Exercice 3: Génération d'un codon statique

- Un codon est une suite d'octets générée à partir d'une clé.
- Par exemple:
  - X1= Clé*451 + 111 modulo (256)
  - X1=Clé*312 + 111 modulo (256)

  X2= X1*451 + 111 modulo (256)
  X2= X1*312 + 111 modulo (256)

1. Pour la clé = t (116 en décimal), donnez les 5 premiers éléments du codon pour les 2 algorithmes.
2. Chiffrer le mot « toto » avec le premier algorithme.
3. Quel est l'inconvénient de cette méthode?
4. Dans cet exemple, la clé est utilisée comme graine du générateur pseudo-aléatoire. Donnez un algorithme qui permettrait à partir de la même clé symétrique de générer des codons différents d'un transfert à l'autre. Vous donnerez les échanges réseau. On suppose que les clés ont déjà été échangées.

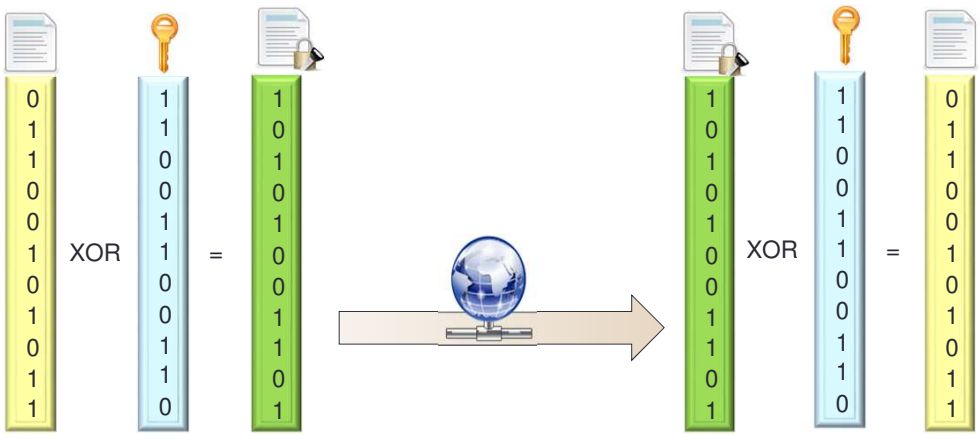
- Note : Une graine d'un générateur aléatoire est un paramètre d'initialisation de ce générateur.

Le chiffrement symétrique en pratique: deux approches

- Par blocs: les messages sont découpés en blocs de tailles fixes et codés:
  - DES (Data Encryption Standard)- 1977: Blocs de 64 bits codés avec une clé de 56 bits. Mis en œuvre en un mode chaînage de blocs (CBC: Cypher Block Chaining).
  - AES (Advanced Encryption Standard): Clés de 128, 192, ou 256 bits sur des blocs de 128 bits.
- Par flux: les messages sont traités bit par bit ou octet par octet
  - RC2-RC5

Exemple de chiffrement par flux

- Génération d'un codon (ou flux de clé) générer à partir de la clé symétrique.
- Un XOR est effectué pour coder et décoder.



Algorithme RC4

- K est la clé (compris entre 5 et 256 octets).
- K est écrit comme un tableau d'entier non signé de 1 octet (K est de l octets dans l'algo ci-dessous)
- En pratique: WEP et WPA (Wi-Fi), SSL, TLS, Oracle SQL, etc.

Initialisation d'un tableau S[] de 256 octets

```
for (i = 0;i<256;i++)
        S[i] = i; // permutation identité
j = 0;
for (i = 0;i<256;i++)
{
        j = (j + S[i] + K[i%l]) % 256;
        swap(S[i], S[j]);
}
```

Génération du codon (keystream)

```
i = 0;
j = 0;
while (données à coder)
{
      i = (i + 1) % 256;
      j = (j + S[i]) % 256;
      swap (S[i], S[j]);
      output S[ (S[i] + S[j]) %256 ];
}
```

Algorithme RC4: initialisation

Clé: IUTInfo → K

12	11	6	3	9

Tableau S[]

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

```
for (i = 0;i<16;i++)
    S[i ] = i; // permutation identité
```

i=0 → j = K[0] % 16 = 12%16=12
swap(S[0]; S[12]);

12	1	2	3	4	5	6	7	8	9	10	11	0	13	14	15

i=1 → j = (12 + 1 + K[1]) % 16 = 24%16=8
swap(S[1]; S[8]);

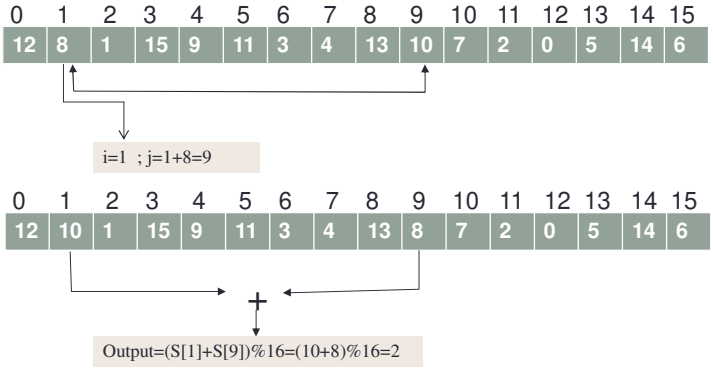
12	8	2	3	4	5	6	7	1	9	10	11	0	13	14	15

```
j = 0;
for (i = 0;i<16;i++)
{
      j = (j + S[i ] + K[i % 5]) % 16;
      swap(S[i ]; S[j ]);
}
```

Après 16 étapes…

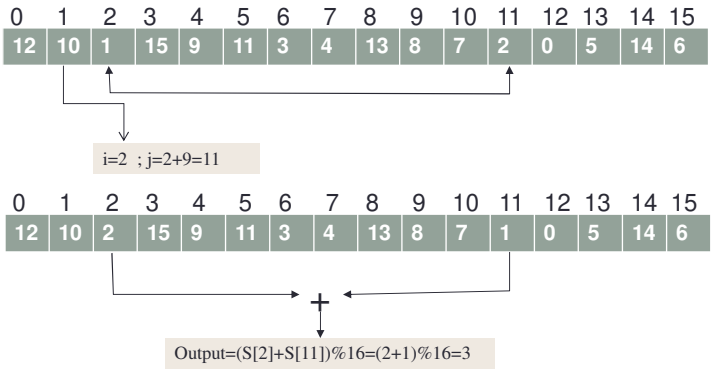
12	8	1	15	9	11	3	4	13	10	7	2	0	5	14	6

Algorithme RC4: génération




```
i = 0;
j = 0;
while (données à coder)
{
        i = (i + 1) % 16;
        j = (j + S[i]) % 16;
        swap (S[i], S[j]);
        output= S[ (S[i] + S[j]) %16 ];
}
```

Algorithme RC4: génération (2)




```
i = 0;
j = 0;
while (données à coder)
{
        i = (i + 1) % 16;
        j = (j + S[i]) % 16;
        swap (S[i], S[j]);
        output= S[ (S[i] + S[j]) %16 ];
}
```

A chaque étape on obtient un octet du codon

2 3 11 7 5 3 6 15 4 11 10 2 …

Avantage/inconvénients clés symétrique

- Avantage
  - Codage rapide des données (4 opérations pour RC-4 pour coder un octet)
  - Robustesse (pour certains algorithmes – en particulier aujourd'hui ceux utilisant un chiffrement par bloc).
- Inconvénients
  - Echange des clés

Exercice 4: échanges des clés

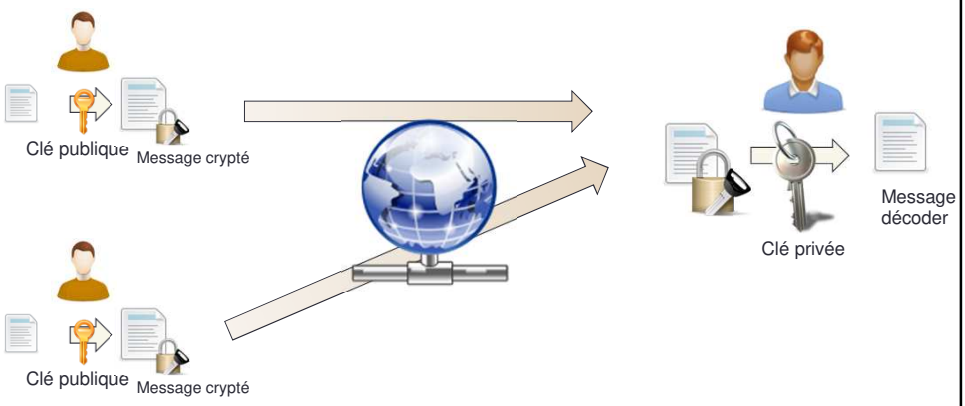
- Exercice 4 :
- Il y a 5 utilisateurs devant échanger des données entre eux.

1. Les 5 utilisateurs sont de confiance et peuvent utiliser la même clé. Combien de clés symétriques seront utilisés ?
2. Les utilisateurs ne doivent pas pouvoir déchiffrer les données des autres, et seulement les données qui leurs sont destinées. Combien de clés symétriques seront utilisées ?
3. Pour le premier ou le deuxième cas décrire une procédure utilisant le réseau permettant d'envoyer un fichier chiffré entre A et B (2 des 5 utilisateurs). Vous devrez y inclure l'échange des clés.

Chiffrement asymétrique

Algorithmes asymétriques

- Une clé pour chiffrer (clé publique)
- Une clé différente pour déchiffrer (clé privée)



Les algorithmes à clés asymétriques

- RSA: basé sur la factorisation en nombre premiers. Utiliser aujourd'hui avec des clés de 1024 et 2048 bits.
- Diffie Hellman: basés sur le calcul de logarithmes discrets.

RSA: Ronald Rivest, Adi Shamir et Leonard Adleman

- Basé sur la factorisation des nombre premiers
- Basé sur les principes suivants:
  - Les variables utilisées sont:

    n = p *q avec p et q deux nombres premiers
    e premier avec ϕ(n)=(p-1)*(q-1)
    d tel que e*d mod(ϕ (n)) = 1 (inverse de e dans Z / ϕ(n) Z)

  - Opération de cryptage

$$c = m^e \ mod(n)$$

  - Opération de décryptage

$$c^d \ mod(n) = m$$

Exercice 5: application simple.

- Soient p=3 et q=11

1. Donnez la valeur de n et ϕ (n).
2. Donnez les valeurs possibles de e (<ϕ (n))
3. Pour e=7, donnez la valeur de d.
4. Chiffrez, avec les valeurs de e et d trouvées à la question précédente, la valeur 6.
5. Déchiffrez cette valeur.

6. On souhaite chiffrer un message codé sur 1 octet (8 bits). e est codé sur 8 bits. Sur combien de bits est codé m^e? Quelle est la taille d'un entier classique?
7. Même question pour e codé sur 1024 bits. Quel problème cela pose t-il?
8. Quelle est la taille maximale du message que l'on peut coder?

Exercice 6: questions sur rsa

1. Avec l'algorithme RSA, quels sont les paramètres (parmi p, q, n, e, etc.) qui constituent la clé privé?

2. Même question pour la clé publique?

Exercice 7: échanges des clés

- Vous devez transmettre des fichiers provenant de plusieurs utilisateurs sur un même serveur. Ces fichiers doivent être chiffrés sur le réseau.
- Vous gérez les clés et leurs transmissions aux utilisateurs qui utilisent ces fichiers.

1. Utilisez vous la même clé pour l'ensemble des fichiers? Pour l'ensemble des utilisateurs?
2. Donnez les échanges réseaux permettant de chiffrer un message/fichier entre A et B. Cela doit inclure l'échange des clés.

Avantages et inconvénients

- Avantages:
  - Simplicité de la distribution des clés
  - Robustesse
- Inconvénients
  - Complexité / Temps de calculs pour le cryptage des données

Fonction de hachage

Fonction de hachage

- Rappel:
  - Une fonction de hachage est une fonction numérique (IN -> IN)
  - Pour ce qui concerne la sécurité, nous avons les propriétés suivantes:
    - Son espace d'état est borné (résultat sur un nombre de bits fixé).
      - Des arguments différents peuvent avoir le même résultat
    - Elle est non (ou très difficilement) inversible
  - Elle est numériquement facile à calculer
  - Exemple: fonction MD5 (Message Digest 5)
    - Résultat sur 128 bits / argument de taille quelconque
    - Calcul basé sur des décalages de bits / modulo / opérations logiques (xor /and /or)

MISE EN ŒUVRE DES ALGORITHMES

Identification / Authentification

Identification: principe de base

- Basé sur un mot de passe:
  - Afin d'authentifier l'origine de l'émetteur, on doit admettre l'axiome suivant:

    Seule la source légitime du message possède le mot de passe.
  - Cependant le mot de passe (la clé) ne doit pas passer en claire sur le réseau
- Basé sur un cryptage asymétrique
  - Afin d'authentifier l'émetteur, on suppose que le récepteur possède la clé publique de l'émetteur légitime

Exercice 8: authentification

- Basé sur un mot de passe:
  - Décrivez un moyen pour que le mot de passe ne passe pas en claire sur le réseau et ne puisse être rejoué.
- Basé sur un cryptage asymétrique
  - Décrivez un moyen pour authentifier la source du message.

Intégrité

Intégrité

- Des données doivent être transmises (en claire ici)
- On doit garantir que ces données n'ont pas été modifiées par un tiers
- On utilise une signature/condensat du document
- La signature est une valeur codée sur un nombre d'octets prédéterminé qui vient garantir l'intégrité du message/document

Exercice 9: intégrité

- Donnez deux algorithmes permettant de vérifier l'intégrité des données:
  - Basé sur des clés symétriques
  - Basé sur des clés asymétriques

Confidentialité

Confidentialité

- Cryptage des données avec un cryptage symétrique ou asymétrique
- Si l'échange est bidirectionnelle, nécessité d'échanger deux paires de clés (publiques, privées) dans le cas du cryptage asymétrique
- Mais
  - Cryptage symétrique : problème de distribution des clés
  - Cryptage asymétrique: problème de complexité
- Si des clés ont déjà été distribués, cryptage des données avec celles-ci
  - Généralement, la clé initiale servira à crypter des clés de sessions temporaires qui seront renouvelées régulièrement.
- Exemple: Wi-Fi

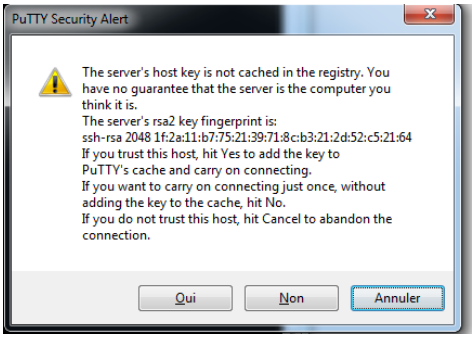
Exercice 10: questions

- Dans le cas où des clés n'ont pas été distribués (dans le cas du web par exemple):
  1. Trouver un algorithme combinant cryptage asymétrique et symétrique qui pallie aux inconvénients de ces deux méthodes
  2. Quel est l'inconvénient de cette méthode?

Exercice 11 : ssh

- ssh permet d'ouvrir un terminal sécurisé sur un poste (le serveur ssh) distant.
- L'authentification se fait au travers d'un système de clés publique/privée.
- Un utilisateur voulant se connecter devra donner son login et son mot de passe.
- Lorsque vous vous connecter la première fois au serveur, voici le message que vous recevez (sous windows ici)

1. Décrivez la raison de ce message
2. Décrivez l'ensemble de la procédure
3. Comment pourrait se faire une authentification sans mot de passe.



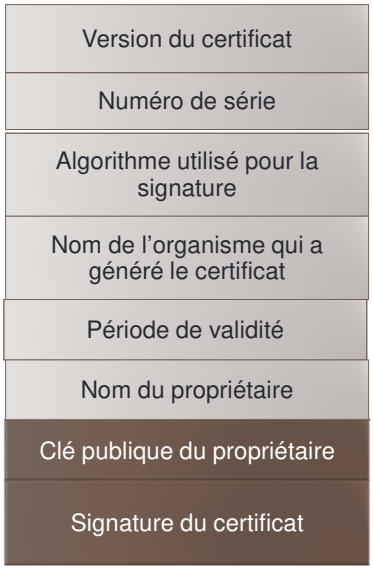
INFRASTRUCTURE DE GESTION DE CLÉS

Problématique

- Nécessité de s'échanger des clés lors d'une communication
  - Clés symétriques: problème de l'échange des clés
  - Clés asymétriques: comment garantir que celui qui envoi sa clé publique est bien celui qu'il prétend
- Tiers de confiance
  - Un tiers de confiance assure l'authentification et fournit la clé publique de l'entité avec laquelle on souhaite communiquer
  - Basé sur un système de certificat (norme X 509)

Norme X 509 (ITU)

- Les certificats sont échangés par les entités souhaitant communiquer.
- Les certificats ont un format normalisé (X 509)
- Ils sont signés avec la clé privé du tiers de confiance, autorité de certification.



Exercice 12: certificat

- Nous considérons les certificats ci-dessous.
- Remarque: en pratique, les résultats du hachage ne sont pas sur le certificat mais calculé au moment de la vérification de celui-ci.
- Quels certificats sont correctes? Et lesquels sont incorrects?

Certificat de l'autorité de certifications	Certificats fournies par mes sites webs		
Trust Company Clé publique (e,n) = (7,33)	**Version / N° de série** **www.site1.fr** **certification: Trust company** Clé publique (e,n) = (12,23) Le résultat du hachage des champs précédents = 20 Signature=15	**Version / N° de série** **www.site2.fr** **certification: Trust company** Clé publique (e,n) = (54,12) Le résultat du hachage des champs précédents = 8 Signature=17	**Version / N° de série** **www.site2.com** **certification: Trust company** Clé publique (e,n) = (54,12) Le résultat du hachage des champs précédents = 17 Signature=9

Exercice 13: résumé ssl

1. Décrire très précisément toutes les étapes pour une communication basée sur un système de certificat et ssl.

EXEMPLES D'IMPLEMENTATIONS

Exemple : SSL / TLS

- SSL (Secure Socket Layer)
- TLS version normalisée (par l'IETF) de SSL
- Sécurité au niveau applicatif (librairie/outil)
  - Correspondant exactement à l'exercice 13
  - Les certificats des autorités de certifications peuvent être téléchargé par l'utilisateur, où sont récupérer directement à partir de l'application lors du téléchargement de celle-ci.
- SMTPS / HTTPS / etc. sont basés sur SSL-TLS

Exemple: IPsec

- Assure des services de sécurité au niveau IP.
- Deux services
  - Authentification: sur le paquet + en-tête (excepté les champs susceptibles d'être modifiés).
  - Confidentialité: cryptage des données
  - Echanges de clés / certificats ou clés déjà connues (Pre-Shared Key)
  - Deux modes :
    - Mode tunnel : les paquets sont authentifiés/cryptés par des serveurs intermédiaires
    - Mode transport: les paquets sont authentifiés/cryptés de bout en bout (source-destination)

VPN : VIRTUAL PRIVATE NETWORK

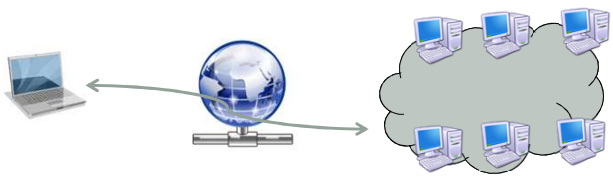
VPN: motivation

- Le réseau d'un site ne peut être accessible qu'en interne
- Les services ne sont accessibles qu'en interne
  - Mail
  - Intranet
  - Application métier
  - Etc.
- Des services externes ne sont accessibles que pour les adresses IP du site
  - Ressources bibliographiques
  - Abonnement à des services
  - Mises à jour logiciels
  - etc.

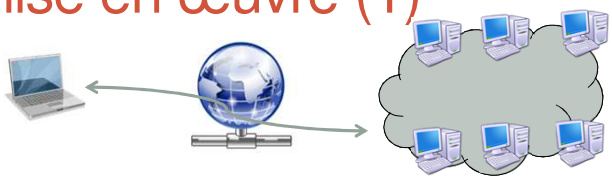
VPN: architecture

- VPN: service qui émule l'appartenance d'un site où d'une machine au réseau privée
- Les adresses IP peuvent être privées
- Deux catégories:
  - On émule des liens directes pour des sites physiquement distants

  - On émule l'appartenance d'un PC à un site

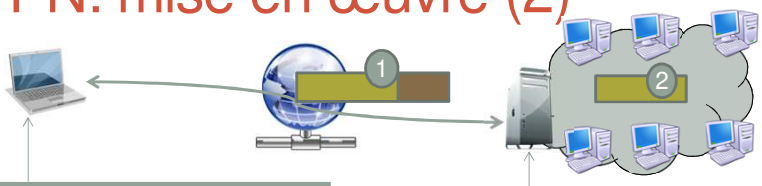


VPN: mise en œuvre (1)



- Le PC a une adresse sur Internet.
- L'idée est de créer un tunnel entre ce PC et le site
  - Connexion du PC sur une passerelle VPN sur le site
  - Obtention d'une adresse IP sur le site (le PC a deux adresses)
  - Les paquets IP d'adresses (site, site) sont encapsulés dans des paquets (@PC, passerelle)
- Le tunnel peut se faire à plusieurs niveaux:
  - IP
  - Point à point
  - Applicatif (SSL) – la passerelle est dans ce cas un proxy (on encapsule pas les paquets IP)
  - SSH – idem
- Les données peuvent être cryptées
  - SSL-TLS
  - IP-SEC

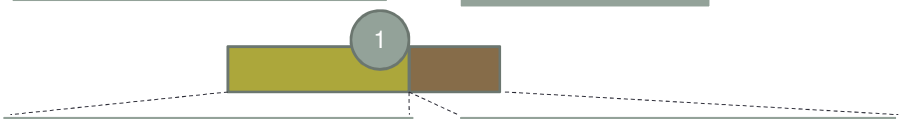
VPN: mise en œuvre (2)



Site privée:
78.0.0.0/24

PC
IP sur Internet: 129.13.45.89
IP sur le site : 78.0.0.4

Passerelle VPN
IP 25.0.0.1



Paquet IP
SRC: 78.0.0.4
DEST: 78.0.0.9 (un serveur du site)

En-tête IP
SRC: 129.13.45.89
DEST: 25.0.0.1 (la passerelle du site)

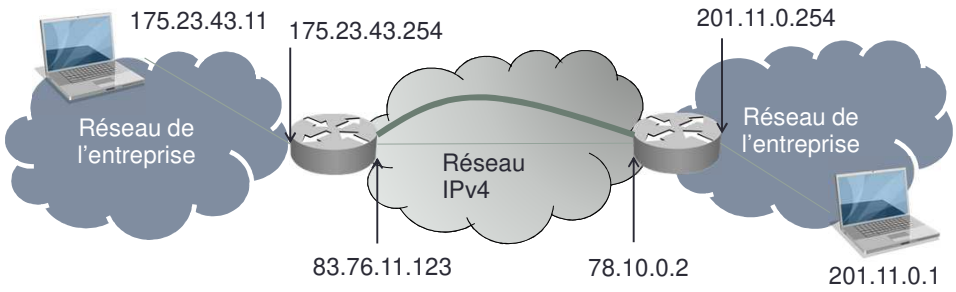
Paquet IP
SRC: 78.0.0.4
DEST: 78.0.0.9 (un serveur du site)

VPN: mise en œuvre (3)

- Le VPN inter-site peut être mise en œuvre par l'entreprise avec un système de tunnel (tunnel IPsec ou SSL)
- Service des opérateurs:
  - Tunnel IP-IP/GRE sécurisé ou non
  - Tunnel MPLS
  - Liaison de niveau 2 (liaison louée)
    - Technologie homogène le long du chemin entre les sites
  - Mise en œuvre de QoS, de réservation de ressources par l'opérateur (SLA, etc.)

Exercice 14: Tunnel

- Décrivez les adresses IP source et destination sur les 3 réseaux dans le cas du VPN/Tunnel suivant (mode tunnel)



Exercice 14bis: Tunnel

- Même question (mode transport)