Cryptographie

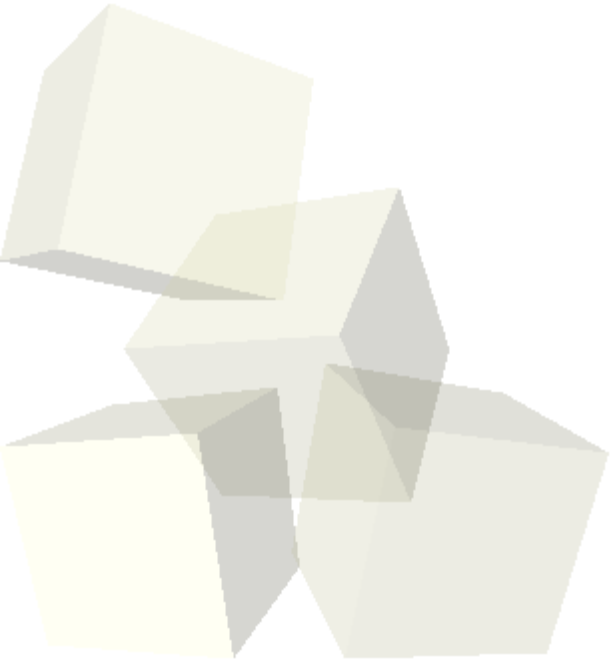
Cours 5

Authentification
Application
Echange de clef

Jérémy Briffaut
STI 2A

Plan

Authentification
&
SSL
&
Application



Introduction

- Apparition des systèmes distribués
- Réseaux à grande échelle

- préserver la *confidentialité* des données
- préserver *l'intégrité* des données
- *authentifier* le correspondant
- Assurer la *non-répudiation*

Authentification

Définition

- La personne à qui j'envoie un message crypté est-elle bien celle à laquelle je pense ?
- La personne qui m'envoie un message crypté est-elle bien celle à qui je pense ?

Authentification

Technique d'Identification

- Prouveur
  - Celui qui s'identifie, qui prétend être…
- Vérifieur
  - Fournisseur du service

- Challenge
  - Le Vérifieur va lancer un challenge au prouveur que ce dernier doit réaliser

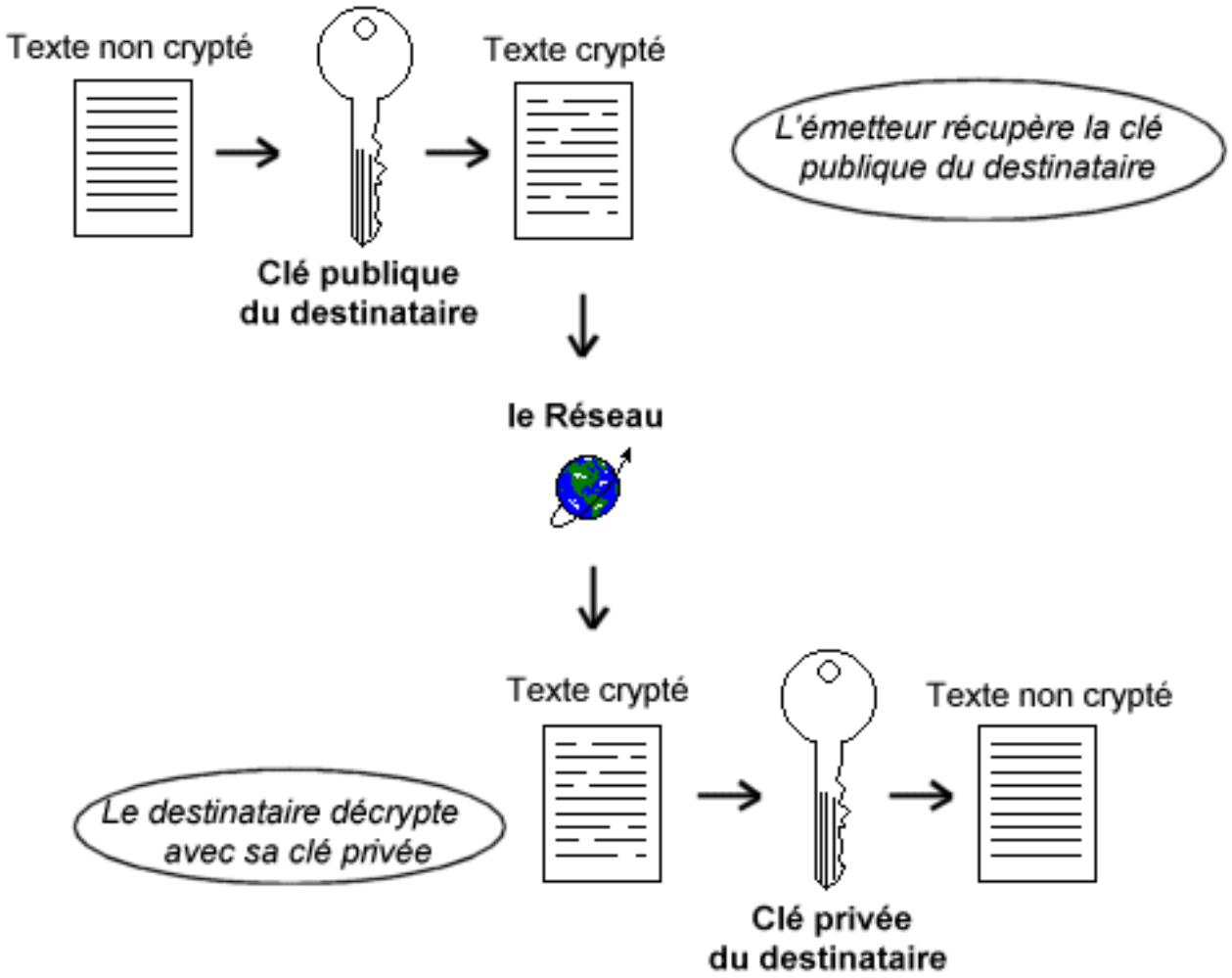
Technique A Clé Publique

Principe

- Algorithme RSA = Réversible
  - ((Mess)CPu)CPr = ((Mess)CPr)CPu
- Confidentialité
- Authentification

Technique A Clé Publique

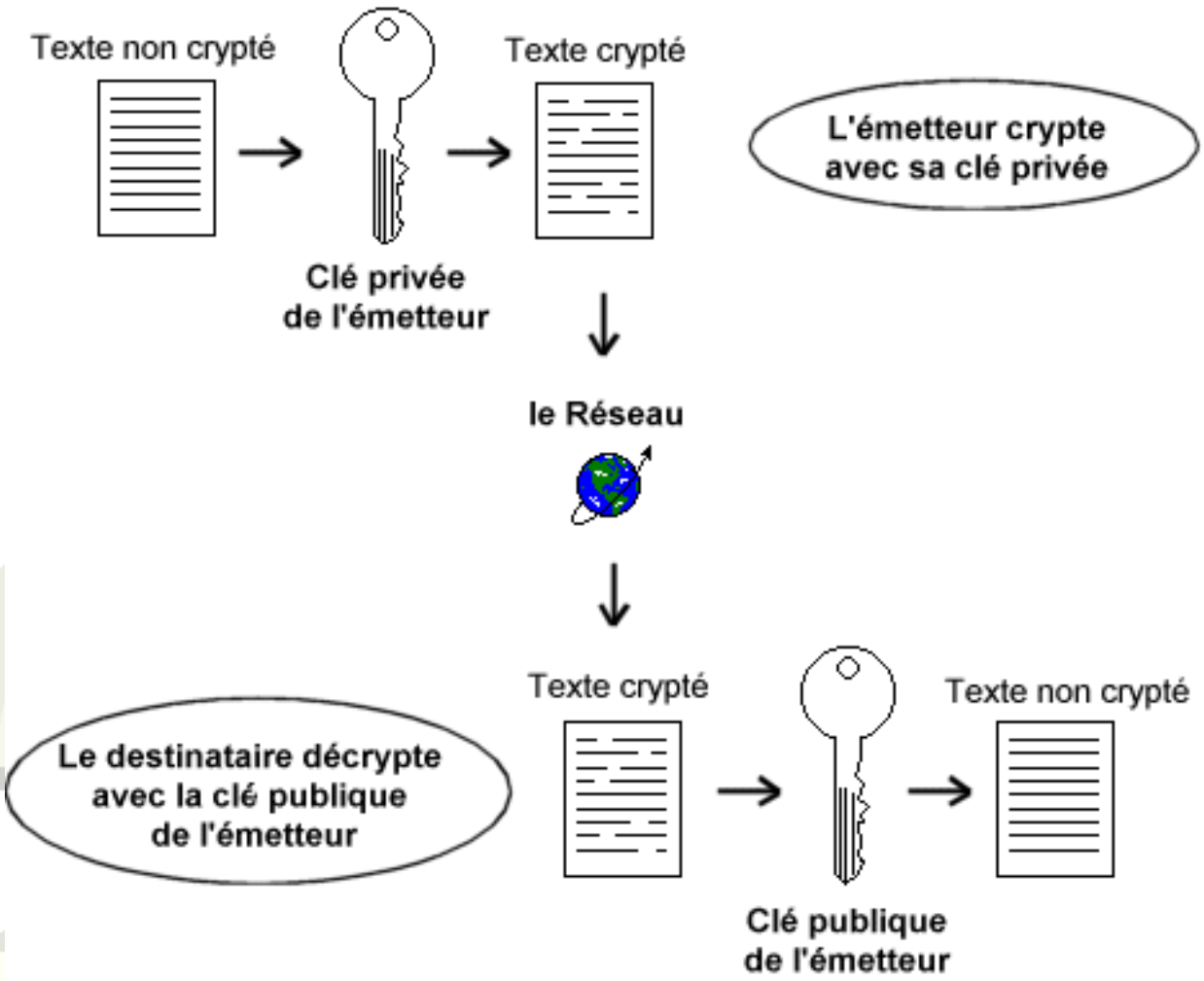
Confidentialité



Le Texte est totalement CONFIDENTIEL car le destinataire est le seul a avoir la CLÉ PRIVÉE

Technique A Clé Publique

Authentification



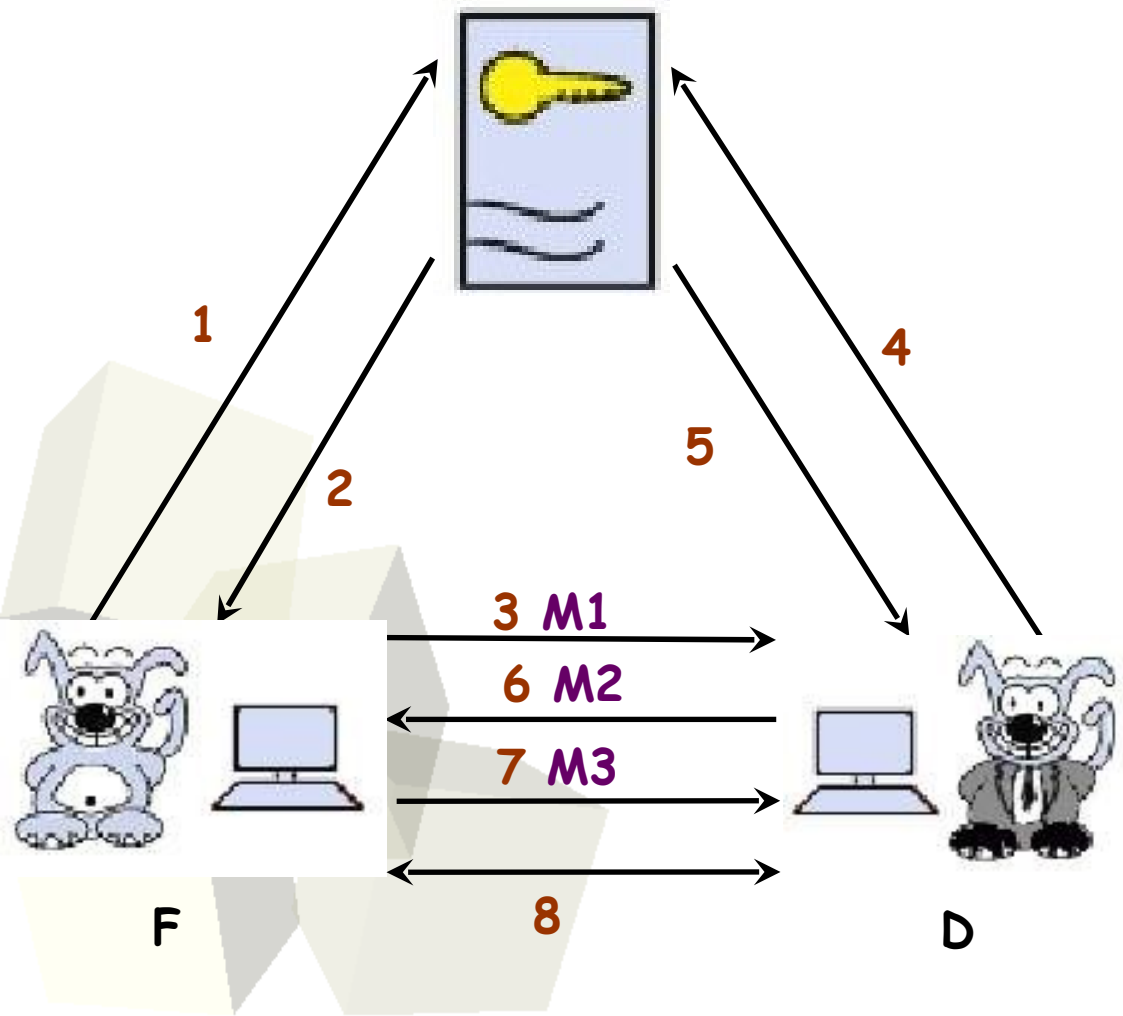
On est sûr de l'identité de l'émetteur car il est le seul à pouvoir chiffrer un message avec cette clé privée

Technique A Clé Publique

Protocole

Serveur d'authentification – Annuaire

(Clé Publiques de F & D…)



1) F demande la *Clé Publique* de D

2) S envoie la *Clé Publique* de D à F

3) F envoie le « challenge » à D:
Décrypte mon message M1(If) et renvoie mon If pour me le prouver!

4) D décrypte M1 et demande à S la *Clé Publique* de F

5) S envoie la *Clé Publique* de F à D

6) A son tour D envoie un « challenge » à F: Décrypte mon message M2(If, Id) et renvoie mon Id !

7) F décrypte M2 et renvoie M3(Id) à D pour lui montrer qu'il y est arrivé

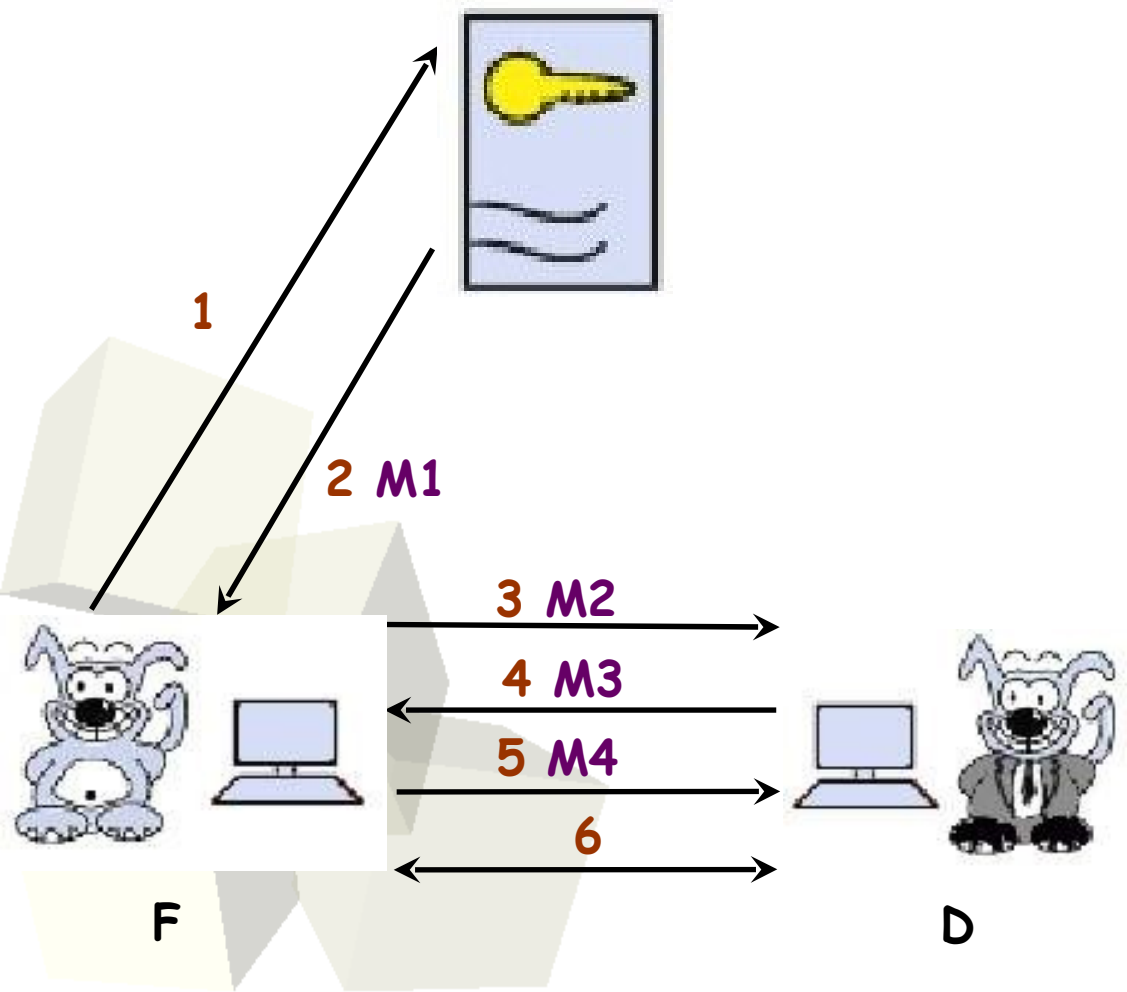
8) F & D peuvent maintenant par ex s'envoyer des messages en créant une *Clé Privée* à partir de (If,Id)

Technique A Clé Secrète

Protocole de Needham – Schroeder

Serveur d'authentification – Annuaire

(Clés Secrètes de F & D)



1) F demande une *Clé de Session* pour pouvoir parler avec D

2) S envoie à F M1 crypté par la *Clé Secrète* de F:

M1 = une *Clé de Session* CSfd en clair et une cryptée par la *Clé Secrète* de D (CSfd)CPd

3) F envoie le « challenge » à D: Décrypte mon message M2((CSfd)CPd) et renvoie un Id crypté par CSfd

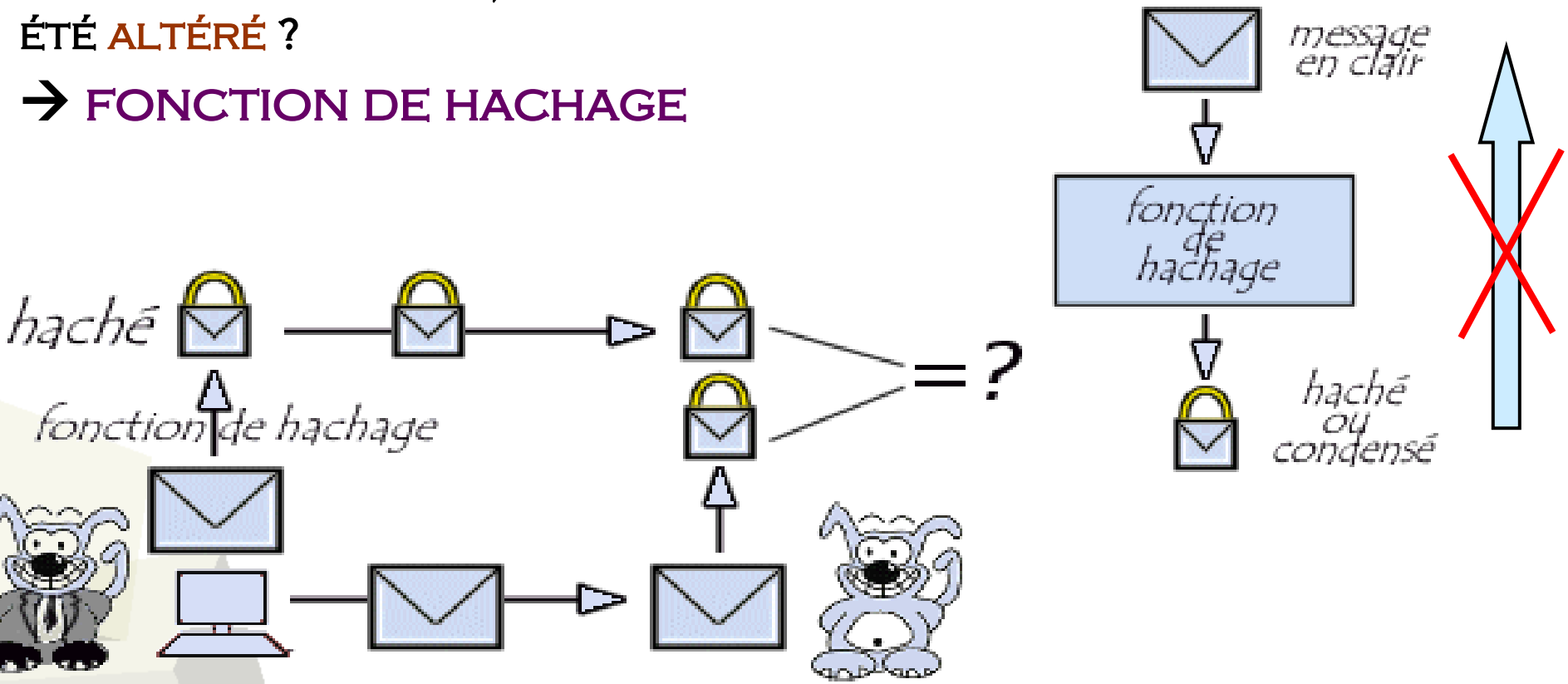
4) D décrypte M2 et envoie son « challenge » : Décrypte mon message M3((Id)CSfd) et renvoie Id-1

5) F décrypte M3 et renvoie M4((Id-1)CSfd)

6) F & D peuvent donc s'envoyer des messages avec la *Clé de Session* (MESSAGE)CSfd

— Comment savoir que le message n'a pas été altéré ?

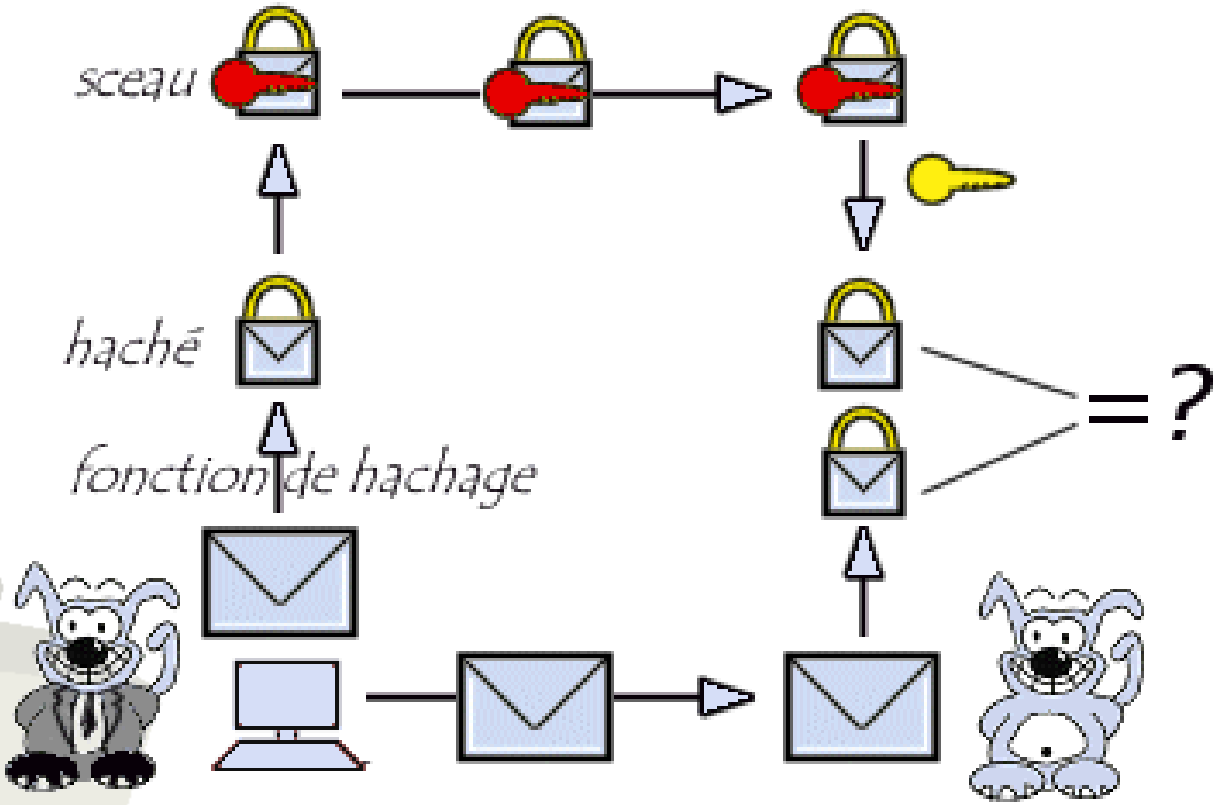
→ Fonction de hachage



— Algorithmes de hachage les plus utilisés: MD5 (128 bits) et SHA (160 bits)

– Pb du hachage : on est pas sur de l'expéditeur

→ Scellement des données



PGP

Pretty good privacy

PGP (Pretty Good Privacy)

Introduction

- PGP est un cryptosystème (système de chiffrement)
- inventé par Philip zimmermann, un analyste informaticien
- Il est très rapide et sûr ce qui le rend quasiment impossible à cryptanalyser

PGP

- Principes
  - Hybride = Repose sur la Combinaison de la cryptographie à clé publique et la cryptographie à clé secrète
- Étapes du chiffrement
  - PGP crée une clé secrète IDEA de manière aléatoire, et chiffre les données avec cette clé.
  - PGP chiffre la clé secrète IDEA précédemment créée au moyen de la clé RSA publique du destinataire
- Étapes du Déchiffrement
  - PGP déchiffre la clé secrète IDEA au moyen de la clé RSA privée.
  - PGP déchiffre les données avec la clé secrète IDEA précédemment obtenue.

PGP

Fonctionnalités

- Signature électronique et vérification d'intégrité de messages
- Chiffrement des fichiers locaux : fonction utilisant IDEA.
- Génération de clefs publiques et privées
- Gestion des clefs:
  - Distribution de la clé publique aux personnes voulant envoyer un message
- Certification de clefs:
  - Ajout d'un sceau numérique pour garantir l'authenticité des clés publiques
- Révocation, désactivation, enregistrement de clefs

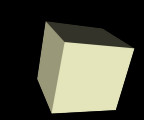
PGP

Format des certificats

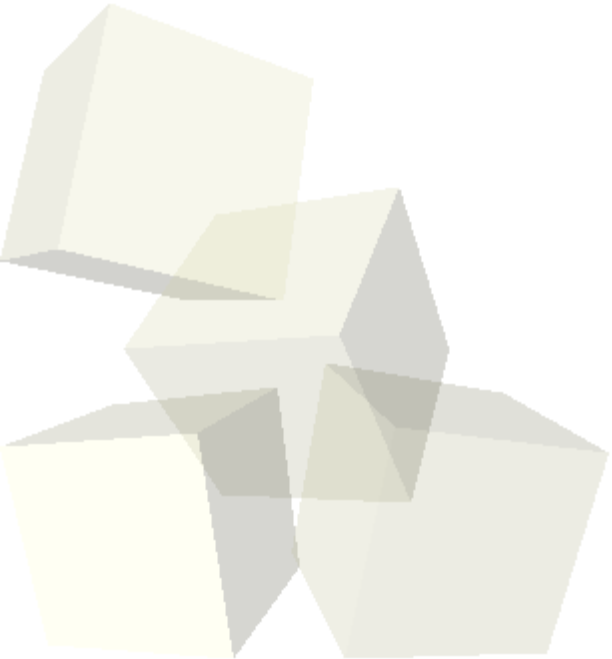
- Le numéro de version de PGP
  - Version de pgp avec lequel a été créé le certificat
- La clef publique du détenteur du certificat:
  - Partie publique de la bi-clé
- Les informations du détenteur du certificat
  - nom, ID utilisateur, photographie, etc.
- La signature numérique du détenteur du certificat :
  - = auto signature = signature effectuée avec la clef privée correspondant à la clef publique associée au certificat.
- La période de validité du certificat:
  - Dates/heures de début et d'expiration du certificat
- L'algorithme de chiffrement symétrique:
  - CAST, IDEA ou DES

PGP versus X509

	PGP	X509
Autorité de certification	Tous les utilisateurs	1 seule
Signature Numérique	Plusieurs	1 seule
Détenteur de clé	Plusieurs	1 seul
Révocation	Émetteur + ceux ajoutés par l'émetteur comme autorité de révocation	Émetteur Seul



Microsoft Passport



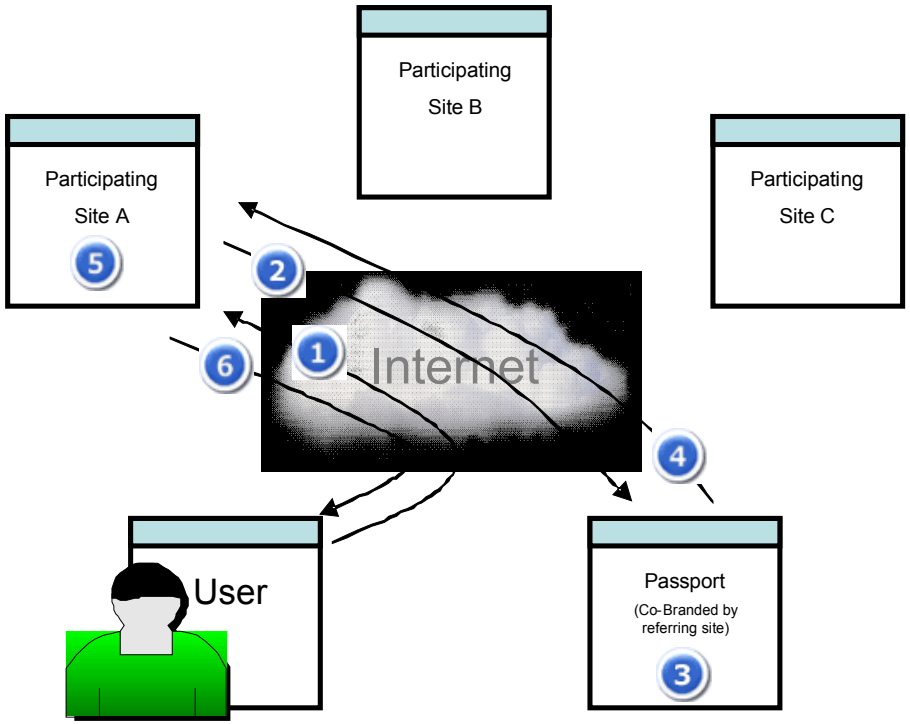
Microsoft .NET Passport

- service en ligne gratuit
- permet de se connecter (en toute sécurité ?) à n'importe quel service ou site Web Passport participant
- Utilisation d’une adresse de messagerie et d’un mot de passe unique

Microsoft .NET Passport

- Contenu obligatoire
  - Email (nom d'utilisateur)
  - Mot de passe
- Contenu optionnel
  - Phrase de rappel
  - Clé de sécurité
  - Numéro de mobile
  - Date de naissance, coordonnées
  - Informations bancaires

Microsoft .NET Passport



- L'utilisateur contacte un site
- L'utilisateur est redirigé sur le site Passport
- L'utilisateur s'authentifie et reçoit un cookie chiffré
- L'Utilisateur est redirigé vers le premier site qui lit le cookie
- L'utilisateur reste authentifié pour tout autre site

Authentification Réseau

Protocole SSL

SSL (Secure Sockets Layer)

- Définition
  - « Couche de Sockets Sécurisée »
  - Protocole d'échange de données au dessus de TCP/IP qui assure:
    - Confidentialité des échanges entre 2 applications
    - Authentification des serveurs
  - Indépendant du protocole Utilisé (HTTP, FTP, …)

SSL (Secure Sockets Layer)

- Principe
  - Utilise RSA (clé publique) pour s'échanger des clés DES (clé Secrète)
    - ➔ Protocole de négociation (choix clés)
    - ➔ Protocole d'échange (chiffré par DES)
  - Authentifie un navigateur, pas une personne

- Compatibilité
  - Presque Tous les Navigateurs
  - Affichage du cadenas en bas pour les sites Sécurisés
  - Un serveur sécurisé possède une URL commencant par https://

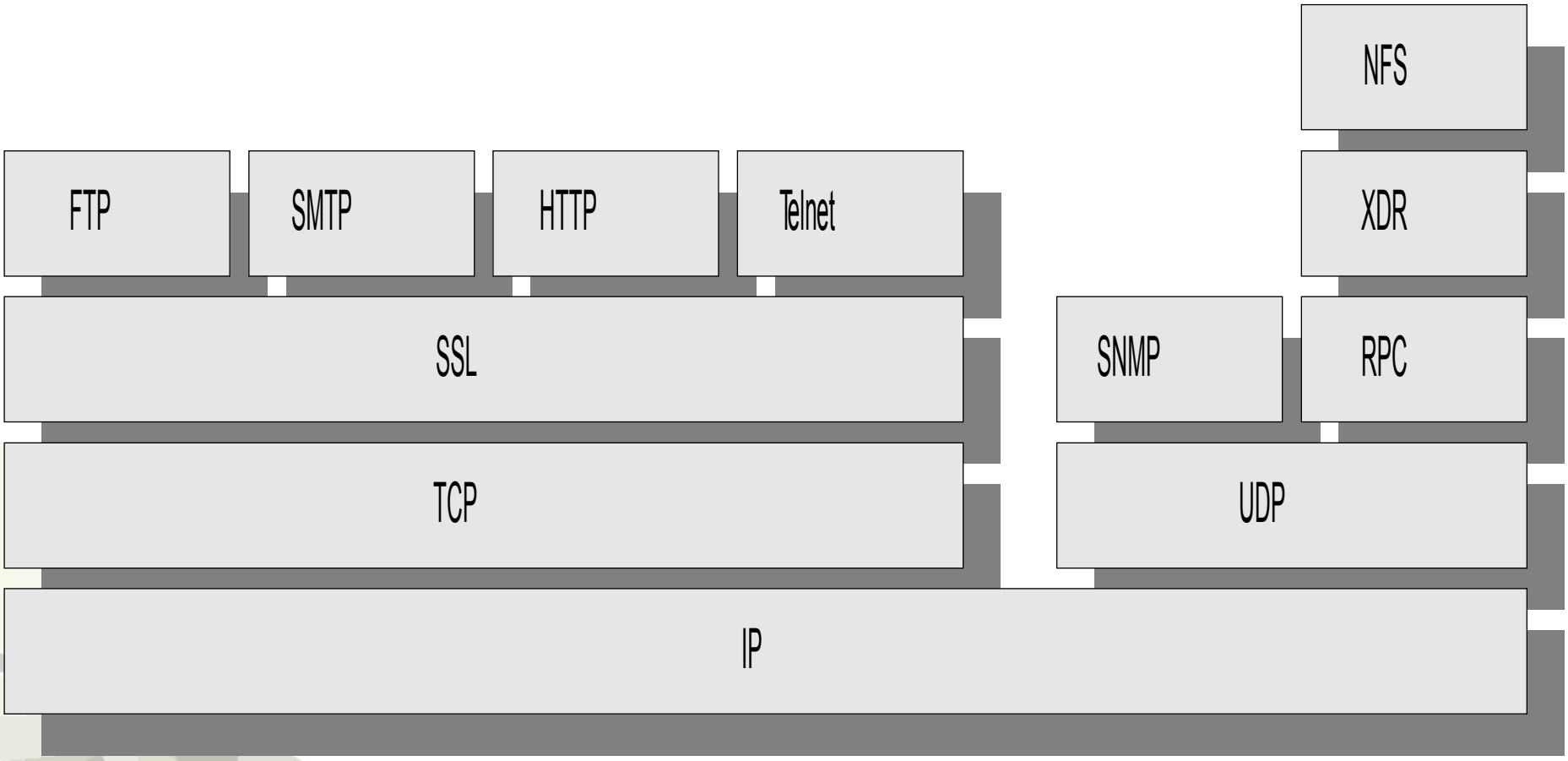
Phase de Négociation

- Authentification
  - Utilise des certificats émis par une autorité de certification
  - Authentifier le serveur vis à vis du client (navigateur)
  - Authentifier le navigateur vis à vis du serveur
- Génération des clés de session
  - Technique à clé publique vue précédemment
  - Création des clés de session
- Fin de négociation
  - Client & serveur sont authentifiés mutuellement
  - Ils ont leurs clés secrètes pour la phase d'échange

SSL : Introduction

- SSL défini par *netsacpe* et intégré au browser
- Première version de SSL testé en interne
  Première version de SSL diffusé : V2 (1994)
- Version actuelle V3
- Standard à l ’IETF au sein du groupe Transport Layer Security (TLS)
- Standard au sein du WAP Forum Wireless Transport Layer Security (WTLS)

SSL : Architecture



Ports au dessus de SSL (1/2)

Protocole sécurisé	Port	Protocole non sécurisé	Application
HTTPS	443	HTTP	Transactions requête-réponse sécurisées
SSMTP	465	SMTP	Messagerie électronique
SNNTP	563	NNTP	News sur le réseau Internet
SSL-LDAP	636	LDAP	Annuaire X.500 allégé
SPOP3	995	POP3	Accès distant à la boîte aux lettres avec rapatriement des messages

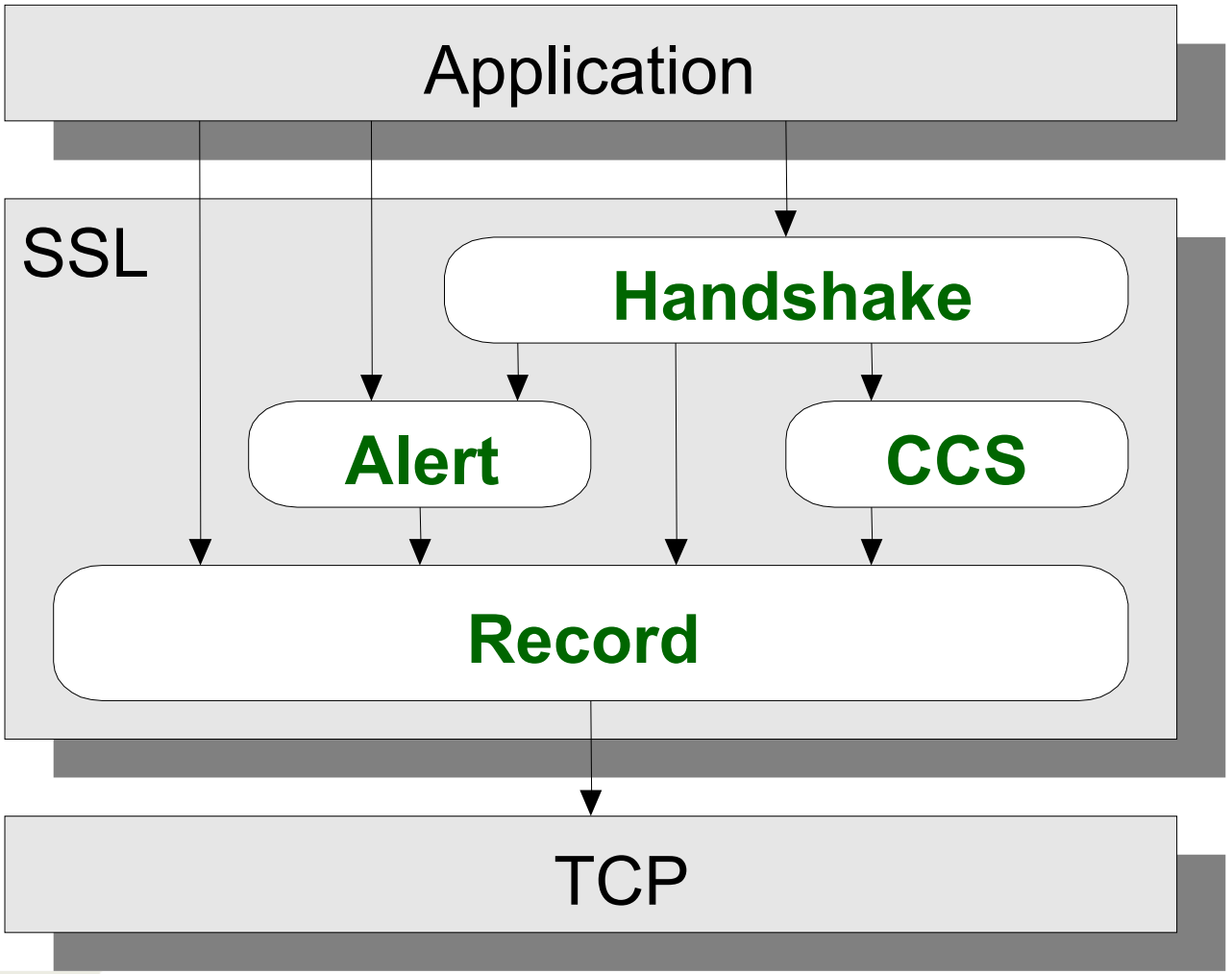
Ports au dessus de SSL (2/2)

Protocole sécurisé	Port	Protocole non sécurisé	Application
FTP-DATA	889	FTP	Transfert de fichiers
FTPS	990	FTP	Contrôle du transfert de fichiers
IMAPS	991	IMAP4	Accès distant à la boîte aux lettres avec ou sans rapatriement des messages
TELNETS	992	Telnet	Protocole d'accès distant à un système informatique
IRCS	993	IRC	Protocole de conférence par l'écrit

SSL : Services

- Authentification
  - Serveur (obligatoire), client (optionnel)
  - Utilisation de certificat X509 V3
  - A l'établissement de la session.
- Confidentialité
  - Algorithme de chiffrement symétrique négocié, clé généré à l'établissement de la session.
- Intégrité
  - Fonction de hachage avec clé secrète : hmac(clé secrète, h, Message)
- Non Rejeu
  - Numéro de séquence

SSL : Protocoles



Handshake (1/6)

- Authentification du serveur et éventuellement du client,
- Négociation des algorithmes de chiffrement et de hachage, échange d’un secret,
- Génération des clés.

Handshake (2/6)

Message	Type de message	Sens de transmission	Signification
HelloRequest	optionnel	serveur ~~client~~>	Ce message demande au client d'entamer le Handshake.
ClientHello	obligatoire	client ~~serveur~~>	Ce message contient : le numéro de version du protocole SSL ; le nombre aléatoire : client_random ; l'identificateur de session : session_ID ; la liste des suites de chiffrement choisies par le client ; la liste des méthodes de compression choisies par le client.
ServerHello	obligatoire	serveur ~~client~~>	Ce message contient : le numéro de version du protocole SSL ; un nombre aléatoire : serveur_random ; l'identificateur de session : session_ID ; une suite de chiffrement ; une méthode de compression.

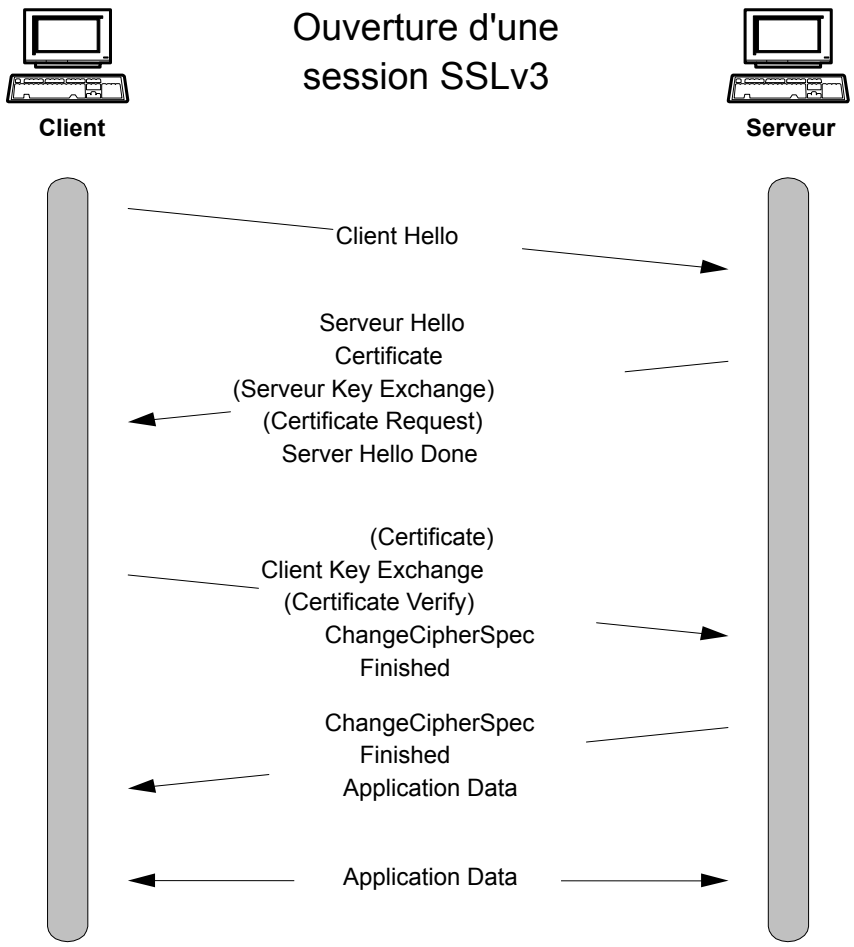
Handshake (3/6)

Certificate	Optionnel	serveur → client client → serveur	Ce message contient le certificat du serveur ou celui du client si le serveur le lui réclame et que le client en possède un.
ServerKeyExchange	Optionnel	serveur → client	Ce message est envoyé par le serveur que s'il ne possède aucun certificat, ou seulement un certificat de signature.
CertificateRequest	Optionnel	serveur → client	Par ce message, le serveur réclame un certificat au client.
ServerHelloDone	Obligatoire	serveur → client	Ce message signale la fin de l'envoi des messages ServerHello et subséquents.

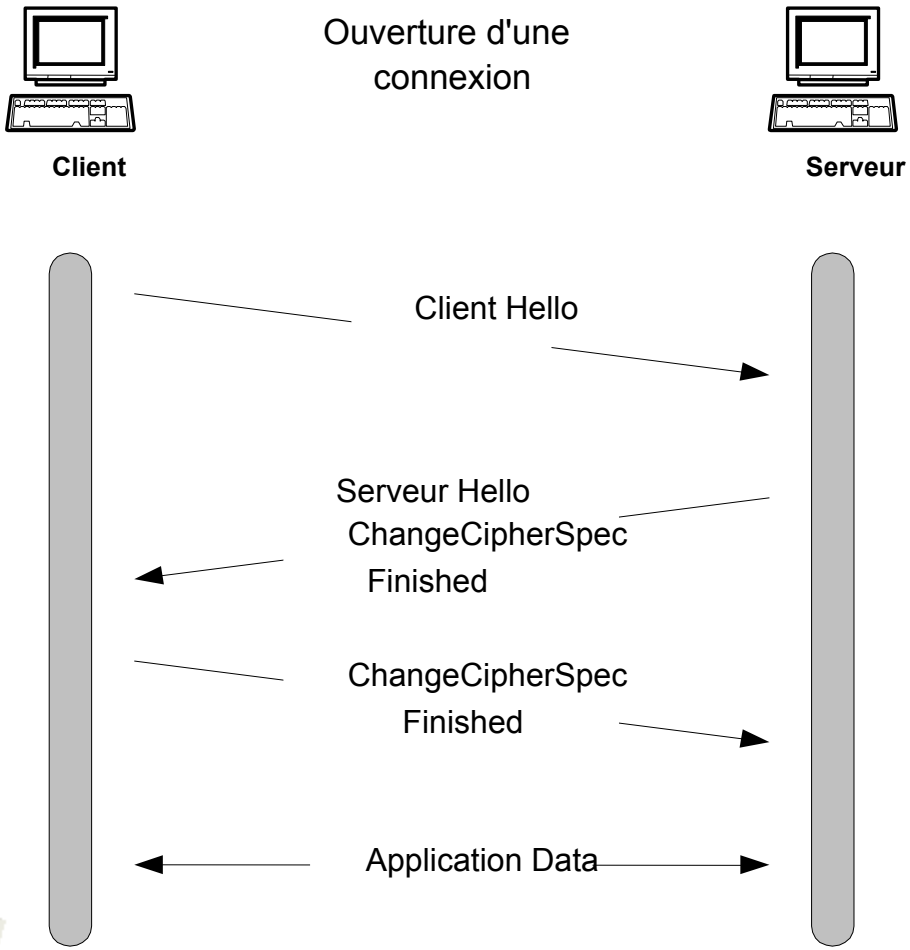
Handshake (4/6)

ClientKeyExchange	Obligatoire	client → serveur	Ce message contient le PreMasterSecret crypté à l'aide de la clé publique du serveur.
CertificateVerify	Optionnel	client → serveur	Ce message permet une vérification explicite du certificat du client.
Finished	obligatoire	serveur → client client → serveur	Ce message signale la fin du protocole Handshake et le début de l'émission des données protégées avec les nouveaux paramètres négociés.

Handshake (5/6)



Handshake (6/6)



ChangeCipherSpec (CCS)

- *ChangeCipherSpec* signale au *Record* toute modification des paramètres de sécurité,

- Constitué d'un message (1 octet)

Le protocole *Record*

- Reçoit les données des couches supérieures : (*Handshake*, *Alert*, *CCS*, HTTP, FTP ...), et les transmet au protocole TCP.

- Après application de :
  - la fragmentation des données en blocs de taille maximum de 2^{14} octets
  - la compression des données, fonction prévue mais non supportée actuellement
  - la génération d'un condensât pour assurer le service d'intégrité
  - le chiffrement des données pour assurer le service de confidentialité

Le protocole *Alert*

- Le protocole *Alert* peut être invoqué :
  - par l'application, par exemple pour signaler la fin d'une connexion
  - par le protocole *Handshake* suite à un problème survenu au cours de son déroulement
- par la couche *Record* directement, par exemple si l'intégrité d'un message est mise en doute

Le protocole *Alert* (2)

Message	Contexte	Type
bad_certificate	échec de vérification d'un certificat	fatal
bad_record_mac	réception d'un MAC erroné	fatal
certificate_expired	certificat périmé	fatal
certificate_revoked	certificat mis en opposition (révoqué)	fatal
certificate_unknown	certificat invalide pour d'autres motifs que ceux précisés précédemment	fatal
close_notify	interruption volontaire de session	fatal
decompression_failure	les données appliquées à la fonction de décompression sont invalides (par exemple, trop longues)	fatal
handshake_ failure	impossibilité de négocier des paramètres satisfaisants	fatal
illegal_parameter	un paramètre échangé au cours du protocole Handshake dépasse les bornes admises ou ne concorde pas avec les autres paramètres	fatal
no_certificate	réponse négative à une requête de certificat	avertissement ou fatal
unexpected_message	arrivée inopportune d'un message	fatal
unsupported_certificate	le certificat reçu n'est pas reconnu par le destinataire	avertissement ou fatal

SSL : charges (1/2)

- Les choix pour les calculs de la charge cryptographique de SSL:
  - algorithme de chiffrement du protocole record : DES 64 bits en mode CBC ;
  - algorithme de chiffrement asymétrique : RSA 1024 bits ;
  - fonction de hachage : MD5 ;
  - itinéraire de certification comprenant une seule étape ;
  - certificat du serveur : autorité de certification unique, déjà connue du client (un seul certificat dans le message *Certificate*) ;
  - taille des informations contenues, du message *Certificate* : 500 Koctets (notons que la taille des informations du certificat est dans la plupart des cas inférieure) ;
  - seule le serveur est certifié.

SSL : charges (2/2)

Opération	Temps de calcul pour le client (ms)	Temps de calcul pour le serveur (ms)	Total (ms)
Ouverture d'une nouvelle session (Handshake complet)	18,94	16,9	35,85
Rafraîchissement d'une session (Handshake simplifié)	0,11	0,11	0,22
Ouverture d'une nouvelle connexion	0,079	0,071	0,15
Temps de calcul pour 16Ko de données (chiffrement ou déchiffrement, élaboration et vérification du MAC)	5,5	5,3	10,8

SSL : liste non exhaustive de serveur

Nom de l'API	Fournisseur	Adresse
AOLserver 2.3	America Online Inc	http://www.aolserver.com
Alibaba 2.0	Computer Software Manufacturers	http://www.csm.co.at/alibaba/
Apache 1.3	The Apache Group	http://www.apache.org
Commerce Server/400 1.0C	I/NET, Inc.	http://www.inetmi.com
Enterprise Server 3.0	Novonyx	http://www.novonyx.com
Enterprise Web Secure/VM 1.1	Beyond-Software Incorporated	http://www.beyond-software.com
Internet Information Server 4.0	Microsoft Corp.	http://www.microsoft.com/iis
Java Server 1.1	Sun Microsystems	http://www.java.sun.com
Lotus Domino Go Webserver 4.6.1	IBM	http://www.ibm.com
Netscape Enterprise Server 3.5	Netscape Communications Co	http://www. netscape.com
Oracle Web Application Server 3.01	Oracle Corp.	http://www.oracle.com/products
Roxen Challenger 1.2b I	Idonex	http://www.roxen.com
SSLava	Phaos Technologies	http://www.phaos.com/main.htm
WebSite Professional 2.2	O'Reilly Software	http://www.website.oreilly.com/
WebTen 2.1	Tenon Intersystems	http://www.tenon.com/products/webten
Zeus Web Application Server 3	Zeus Technology	http://www.zeustech.net

SSL : liste de suite de chiffrement supportée par un serveur

Serveur et Version				Apache SSLLeay 08.0	Jigsaw 2.0 Beta 1	Microsoft IIS/4.0	Netscape Entreprise3.0L	Netscape Entreprise 3.0F	SSLava Beta 1
Suite		**Export**	**Code**						
RSA	RC4-40 MD5	✓	0x03	●	●	●	●	●	●
	RC4-128 MD5		0x04	●	●	●	●		●
	RC4- 128 SHA		0x05	●	●	●			●
	RC2 CBC-40 MD5	✓	0x06	●	●	●	●	●	
	IDEA CBC SHA		0x07	●	●				
	DES40 CBC SHA	✓	0x08	●	●	●			●
	DESCBC SHA		0x09	●	●	●	●		●
	3DES EDE CBC SHA		0x0A	●	●	●	●		
DH et DSA	DES40 CBC SHA	✓	0x0B		●				
	DES CBC SHA		0x0C		●				
	3DES EDE CBC SHA		0x0D		●				
DH et RSA	DES40 CBC SHA	✓	0x0E		●				
	DES CBC SHA		0x0F		●				
	3DES EDE CBC SHA		0x10		●				
DHE et DSA	DES40 CBC SHA	✓	0x11		●				
	DES CBC SHA		0x12		●				
	3DES EDE CBC SHA		0x13		●				
DHE et RSA	DES40 CBC SHA	✓	0x14		●	●			
	DES CBC SHA		0x15		●	●			
	3DES EDE CBC SHA		0x16		●	●			

SSL : liste non exhaustive d'APIs

Nom de l'API	Fournisseur	Adresse
AOLserver 2.3	America Online Inc	http://www.aolserver.com
Alibaba 2.0	Computer Software Manufacturers	http://www.csm.co.at/alibaba/
Apache 1.3	The Apache Group	http://www.apache.org
Commerce Server/400 1.0C	I/NET, Inc.	http://www.inetmi.com
Enterprise Server 3.0	Novonyx	http://www.novonyx.com
Enterprise Web Secure/VM 1.1	Beyond-Software Incorporated	http://www.beyond-software.com
Internet Information Server 4.0	Microsoft Corp.	http://www.microsoft.com/iis
Java Server 1.1	Sun Microsystems	http://www.java.sun.com
Lotus Domino Go Webserver 4.6.1	IBM	http://www.ibm.com
Netscape Enterprise Server 3.5	Netscape Communications Co	http://www. netscape.com
Oracle Web Application Server 3.01	Oracle Corp.	http://www.oracle.com/products
Roxen Challenger 1.2b I	Idonex	http://www.roxen.com
SSLava	Phaos Technologies	http://www.phaos.com/main.htm
WebSite Professional 2.2	O'Reilly Software	http://www.website.oreilly.com/
WebTen 2.1	Tenon Intersystems	http://www.tenon.com/products/webten
Zeus Web Application Server 3	Zeus Technology	http://www.zeustech.net

Attaques

- Pistes d’attaques classiques
- Vulnérabilités
- Scénarios d’attaque

Attaques classiques

- Pistes classiques
  - Casser les clefs
  - Attack replay
  - Man in the middle
  - Attaque à clair ouvert
- Parades de SSL
  - Taille des clefs
  - Nonces (connection id)
  - Certificats servent à passer les clefs
  - Clefs + Aléas

Sources de vulnérabilité

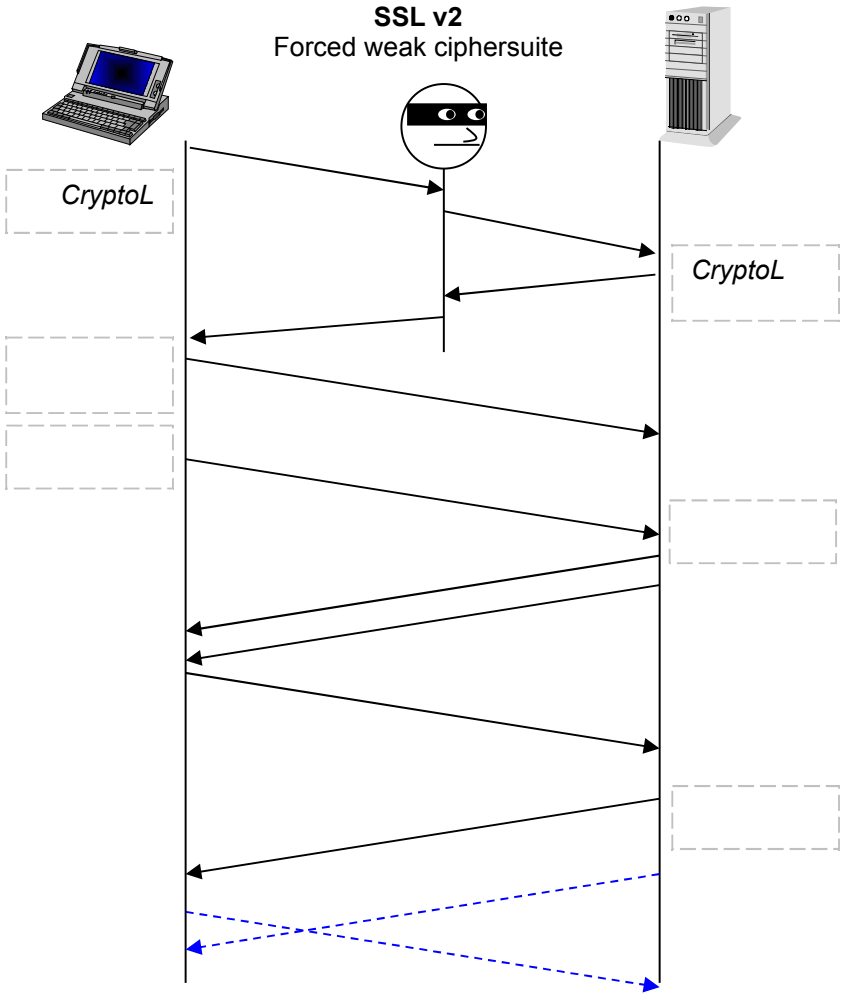
- Taille des clefs
- SSL v2
- Certificats
- Implémentations

Scénarios d'attaque

- SSL v2 : Forcer une faible taille de clef
- Tous : Diffie-Hellman anonyme
- SSL v3 : Accepte *Finished* avant *ChangeCipherSpec*
- SSL v3 : Envoi de données chiffrées avant réponse serveur au *Finished.*

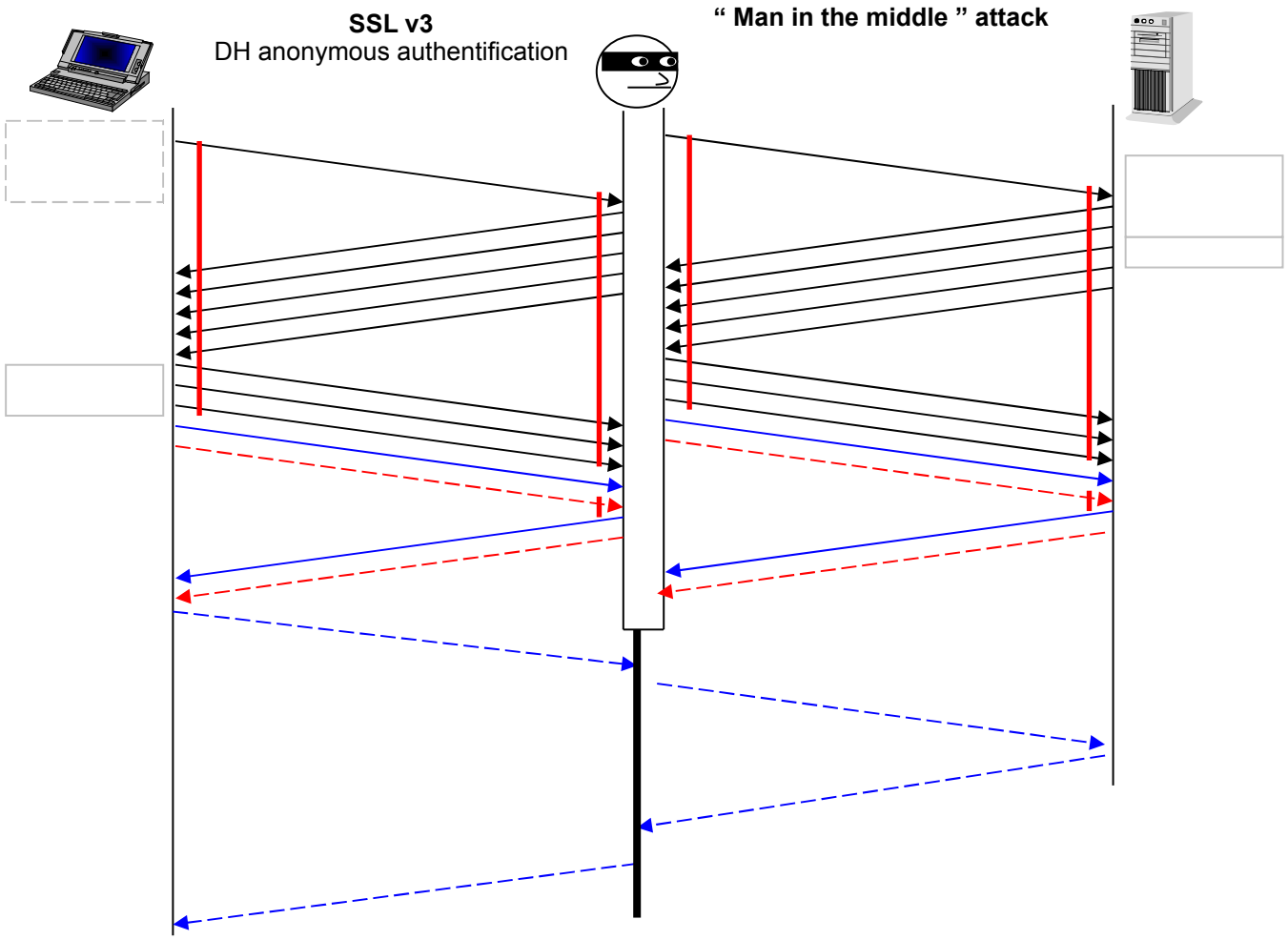
Exemple 1

- Faible clefs
- Pas de vérification d'intégrité

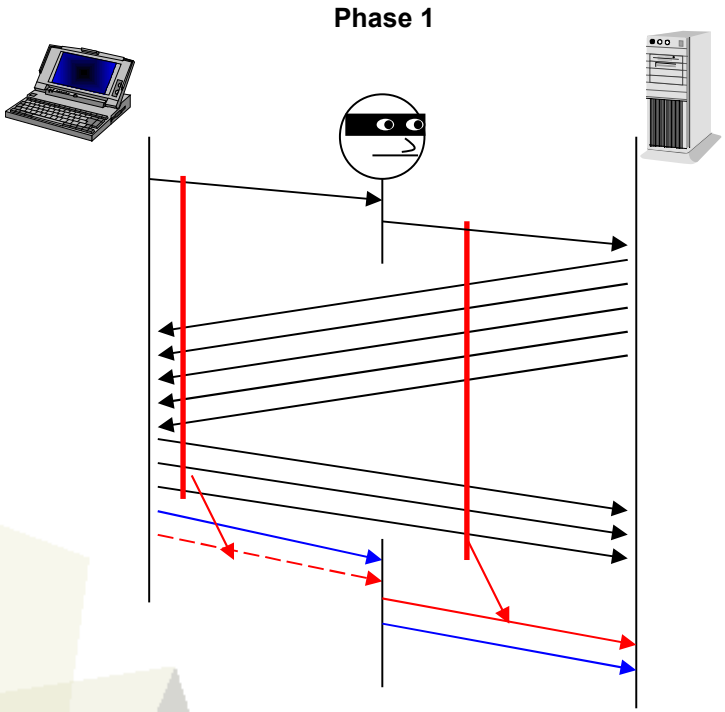


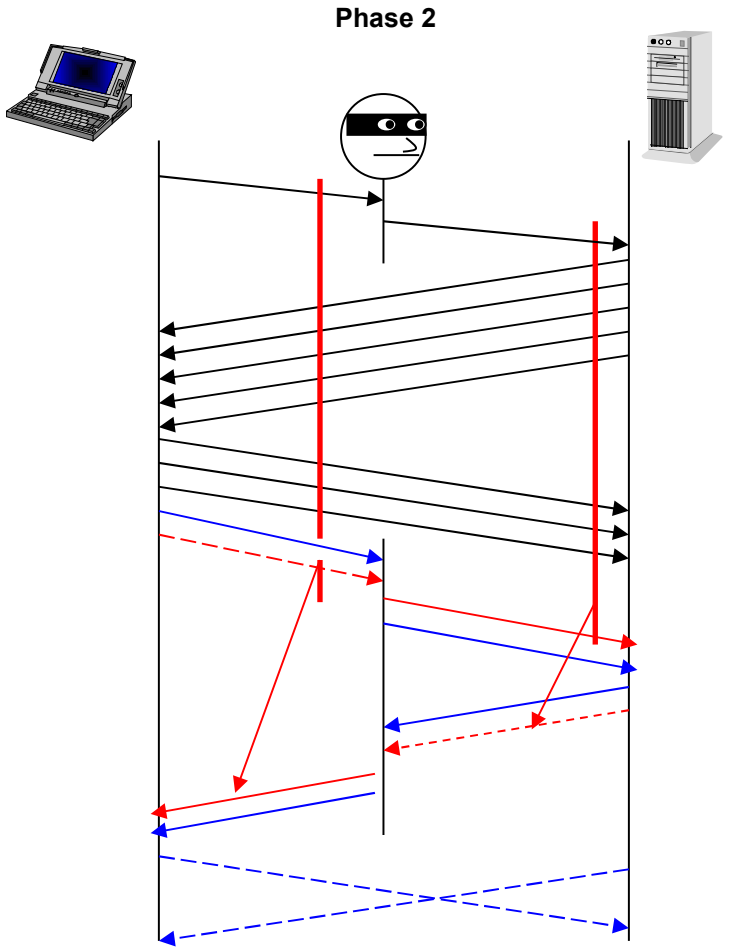
Exemple 2

- Tiers à l'écoute



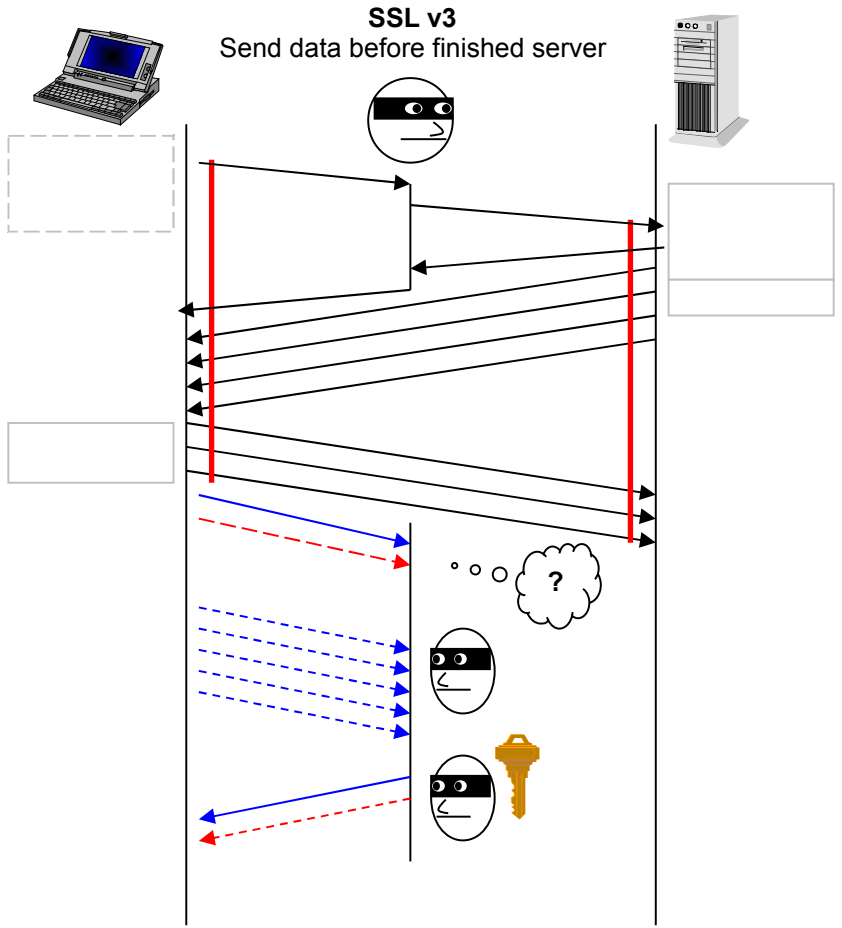
Exemple 3





Exemple 4

- Envoi de données chiffrées
- Permet de casser la clef,
  puis de répondre



Plan

I. Histoire, définition et objectifs de la cryptographie
   - Concepts et algorithmes de permutation et de substitution

II. Chiffrement Symétrique
   - DES, 3DES, AES, IDEA

III. Chiffrement Asymétrique
   - RSA, ElGamal

IV. Signature, Hachage et Scellement

V. Echange de clés
   - Algorithme Deffie-Hellman

VI. Hachage : MD5, SHA-1, SHA-2

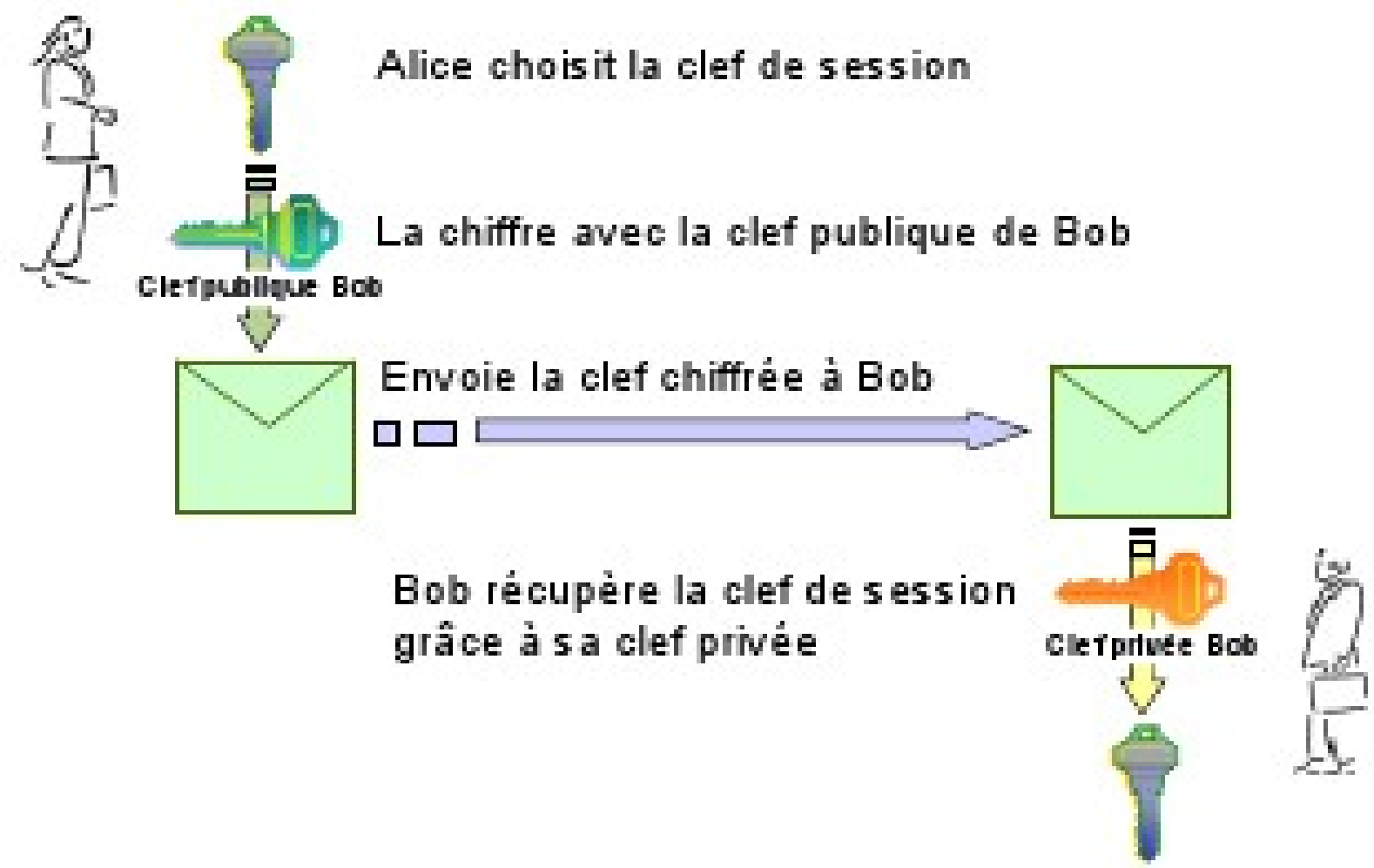
VII. Code d'Authentification & MAC

Authentification Mutuelle
Et
Echange de Clefs de session

- Relations entre échange de clefs et authentification mutuelle
  - L'échange de clefs doit être authentifié pour éviter les attaques
  - Une **clef de session** permet d'étendre l'authentification à l 'ensemble de la communication
  - Protocole d'authentification mutuelle avec échange de clefs
    - → fournit authentification mutuelle et un échange de clefs authentifié tout-en-un
- Types d'échange de clefs
  - Transport
    - → Exemple : transport RSA (utilisé par SSL)
  - Génération
    - → Exemple : Diffie-Hellman

- La première méthode d'échange de clés publiques à avoir été décrite fut celle de Diffie-Hellman (1976).
  - Cette méthode repose sur une fonction à sens unique du logarithme secret.
- Les échanges de clés publiques vont se faire,
  - en générant un secret partagé sur un réseau non sécurisé.
  - l'échange de clés publiques se fera alors de manière sécurisée permettant ainsi d'employer des chiffrements sysmétriques de manière évolutive.
- Cette méthode est tout de même sensible à l'attaque par interposition (**man in the middle**) néanmoins cet inconvénient peut être résolu en signant les échanges.
- Le protocole **IKE** (**Internet Key Exchange** en IPSec) est un système d'échange Diffie-Hellman avec authentification.
- **IPSec** exploite le chiffrement sysmétrique en mode **CBC** pour le chiffrement et les **HMAC** pour l'authentification des gros volumes de données.

- Transport de clef
  - Échange de clef de session
  - Exemple avec chiffrement asymétrique :



Diffie-Hellman : principe

- Qu'est-ce que le protocole DH ?
  - → Protocole cryptographique qui permet à deux tiers de générer un secret partagé sans informations préalables l'un sur l'autre
- Principe
  - → Échange de valeurs publiques
- Fonctionnement :

Algorithme Diffie-Hellman

Génération de clés de session

1- p un grand nombre premier ; α un générateur de Zp*.

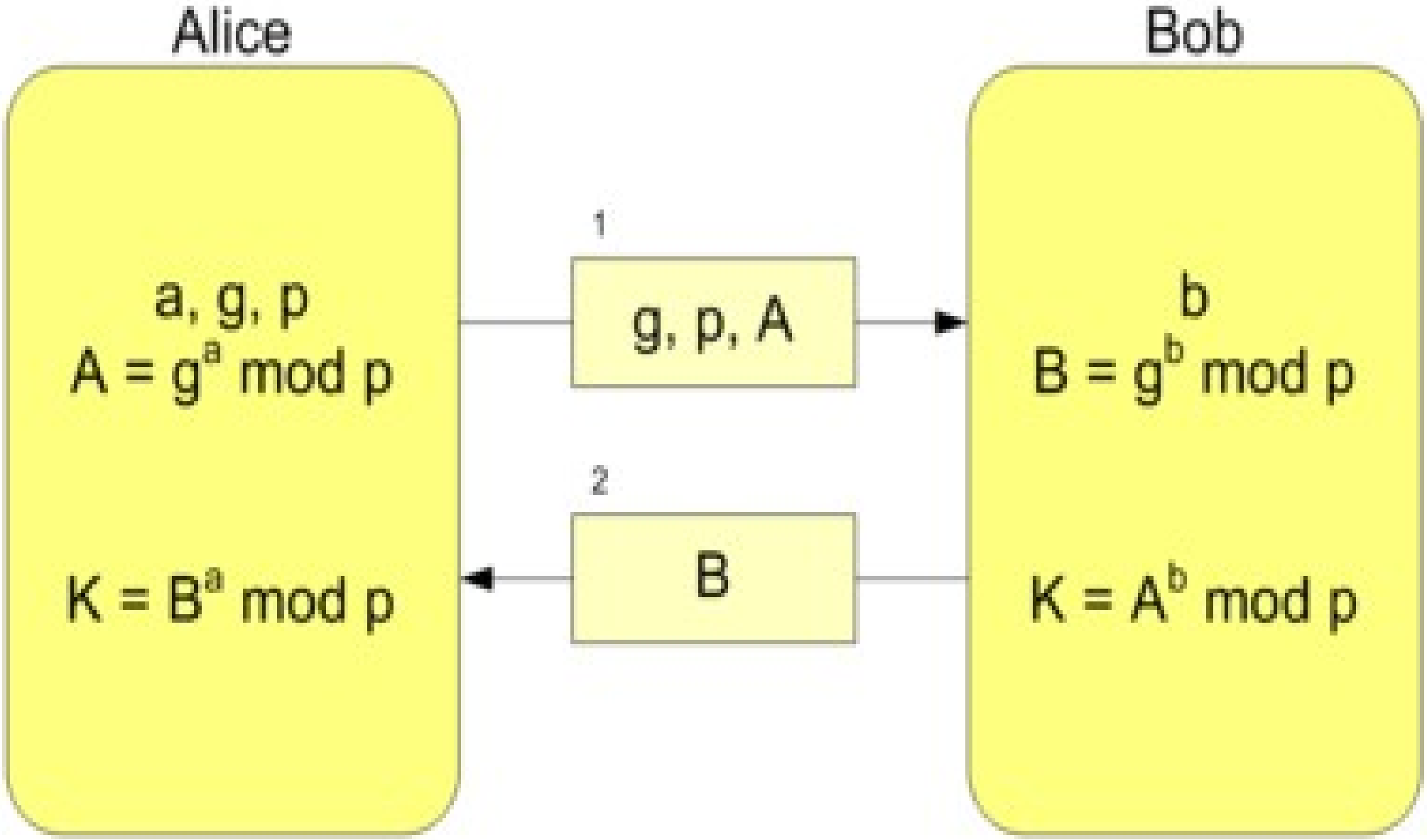
2- A choisit a, et calcule αa = α^a mod p.

3- B choisit b, et calcule αb = α^b mod p.

4- A et B s'échange αa et αb.

5- A calcule α^(ba) mod p et B calcule αab mod p.

IV. Authentification mutuelle et échange de clefs de session

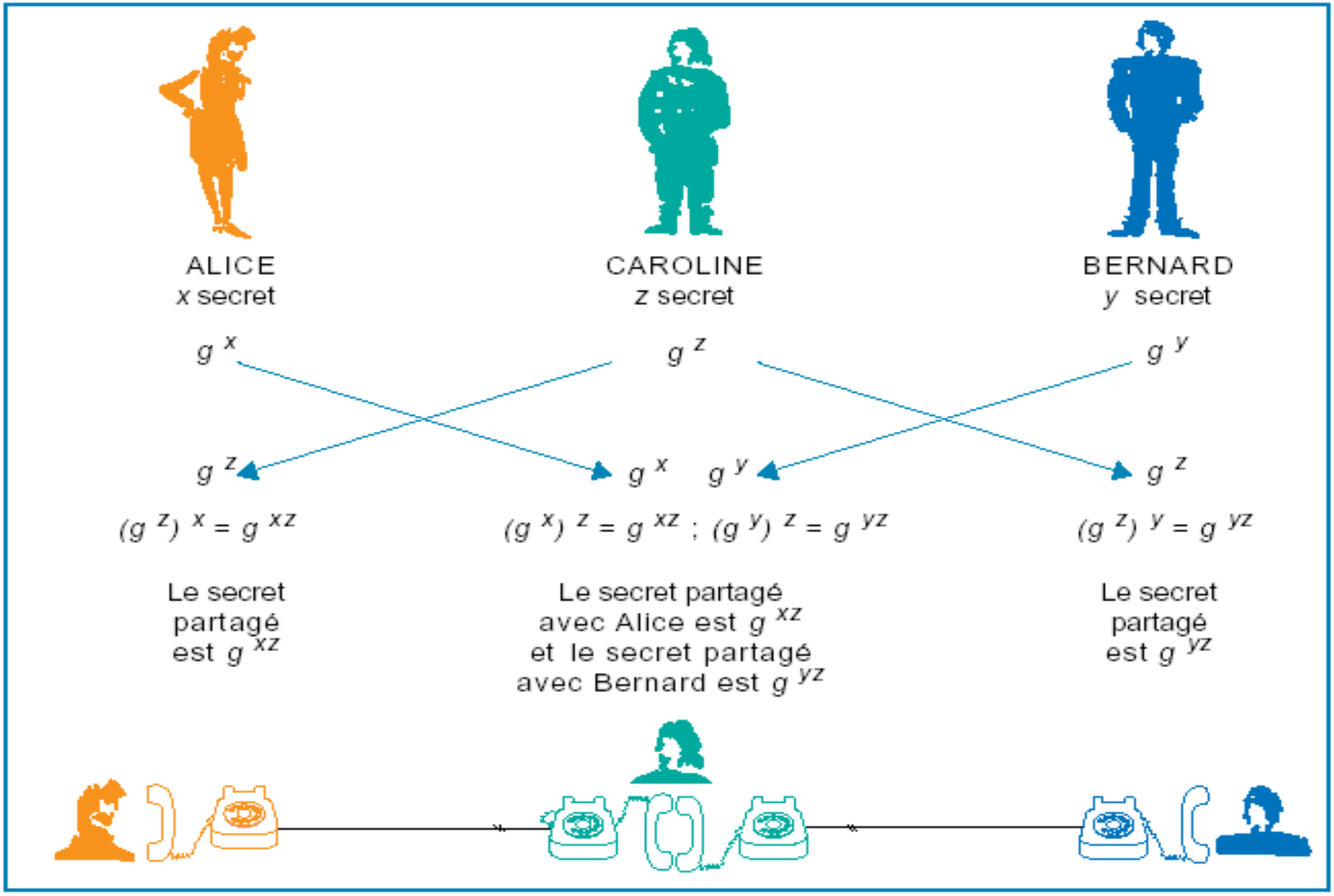


$$K = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p$$

Diffie-Hellman : propriétés

- Sensible à l'attaque de l'intercepteur
  - L'attaquant, Caroline, envoie sa valeur publique à la place d'Alice et de Bernard.
    - Elle partage ainsi un secret avec chaque tiers.
  - Solution: authentifier les valeurs publiques
    - le protocole résultant s'appelle Diffie-Hellman authentifié.
- Propriété de Perfect Forward Secrecy (PFS)
  - Principe
    - La découverte du secret à long terme ne compromet pas les clefs de session
    - Propriété fournie lorsque le secret à long terme n'intervient pas dans la génération ou la protection en confidentialité des clefs
  - DH authentifié fournit la PFS si les seules valeurs à long terme sont celles utilisées pour l'authentification (i.e. les valeurs privées/publiques sont à court terme)

- Logarithme secret et man in the middle



Chiffrement à clef publique

- Algorithme El-Gamal

El-Gamal fabrication de la clé

Inspiré de Diffie-Hellman

Destinataire

1. Choisit un nombre premier p et un générateur g de $\mathbb{Z}_p$
2. sélectionne aléatoirement un nombre $a \in \mathbb{Z}_p$
3. publie $(p, g, g^a \bmod p)$

El-Gamal protocole

Expéditeur:

1. soit m le message avec $m < p$ choisit **aléatoirement** un entier $b \in \mathbb{Z}_p^*$
2. calcule $\mathbf{c_1} = \mathbf{g^b} \bmod \mathbf{p}$ et $\mathbf{c_2} = \mathbf{m}.(\mathbf{g^a})^{\mathbf{b}} \bmod \mathbf{p}$
3. envoie $c = (c_1, c_2)$

Destinataire :

1. calcule $\mathbf{d_1} = (\mathbf{c_1})^{\mathbf{p-1-a}} \bmod \mathbf{p} = (\mathbf{c_1})^{\mathbf{-a}} \bmod \mathbf{p} = \mathbf{g^{-ab}} \bmod \mathbf{p}$
2. puis calcule $\mathbf{d_2} = \mathbf{d_1}.\mathbf{c_2} \bmod \mathbf{p} = \mathbf{g^{-ab}}.\mathbf{m}.(\mathbf{g^a})^{\mathbf{b}} \bmod \mathbf{p} = \mathbf{m}$

El-Gamal exemple

1. $p = 11111117$, $g = 111112$ et $a = 1234$
2. A calcule $k_a = 111112^{1234} \bmod 11111117 = 7218868$
3. A publie $p = 11111117, g = 111112, k_a = 7218868$

El-Gamal exemple

1. B choisit $b = 876$ et a pour message 99999
2. B calcule $c_1 = 111112^{876} \bmod 11111117 = 8671412$
3. et $c_2 = 99999.7218868^{876} \bmod 11111117 = 3205709$
4. B envoie $(8671412, 3205709)$

1. A calcule $d_1 = 8671412^{-1234} \bmod 11111117 = 5300581$
2. et $d_2 = 5300581 * 3205709 \bmod 11111117 = 99999$