Legond-Aubry Fabrice
fabrice.legond-aubry@lip6.fr

SSI

Sécurité des Systèmes Informatiques

Concepts

Note: une partie des slides est extrait du cours de sécurité de S. Naktin du CNAM

Plan de cours

Utilisation des outils

Introduction

Concepts et Terminologie

Types d'attaques

Les politiques de sécurité

Les outils de la sécurité

Utilisation des CS symétriques

Utilisation des CS asymétriques

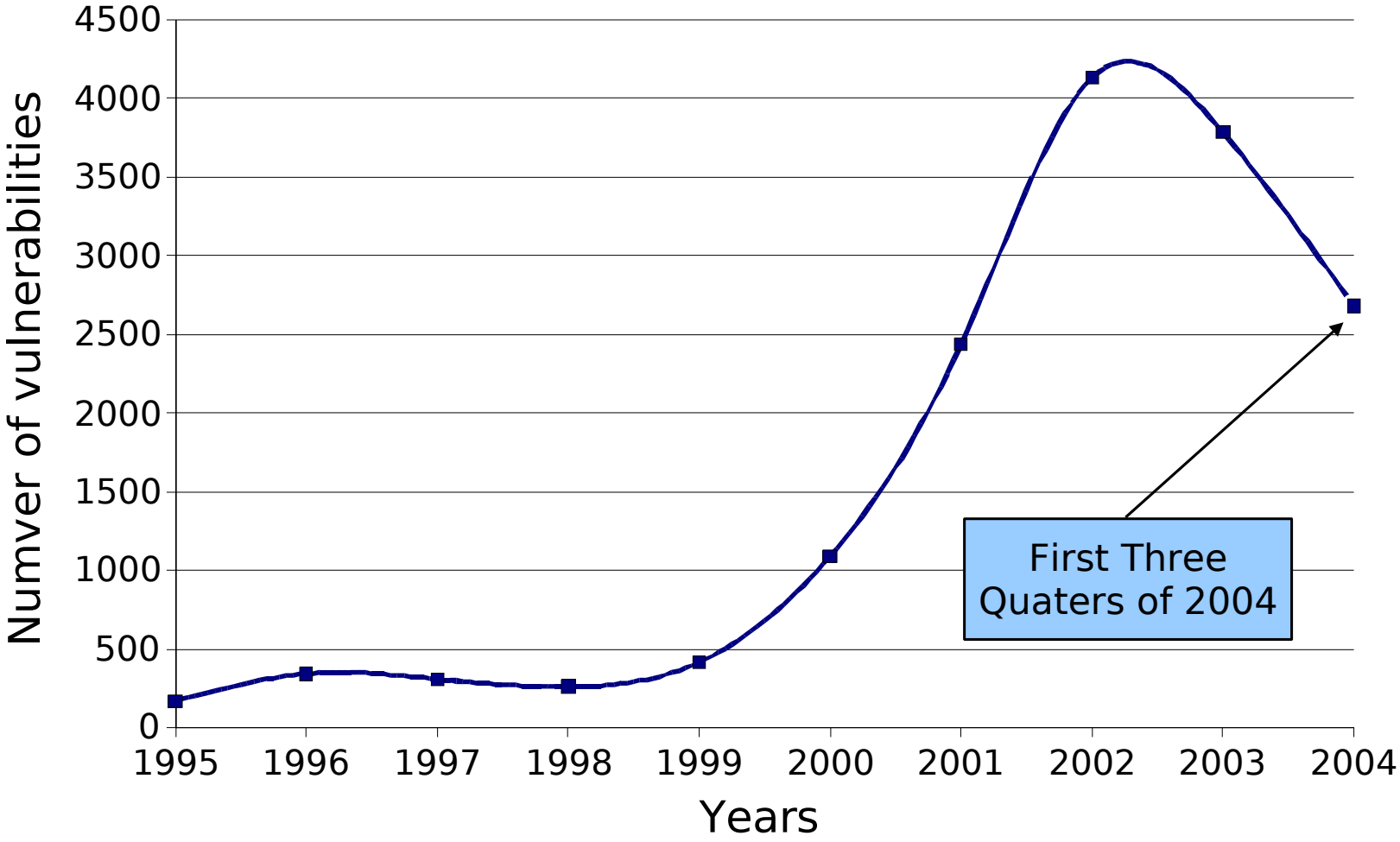
Autres Utilisations des CS

Les certificats

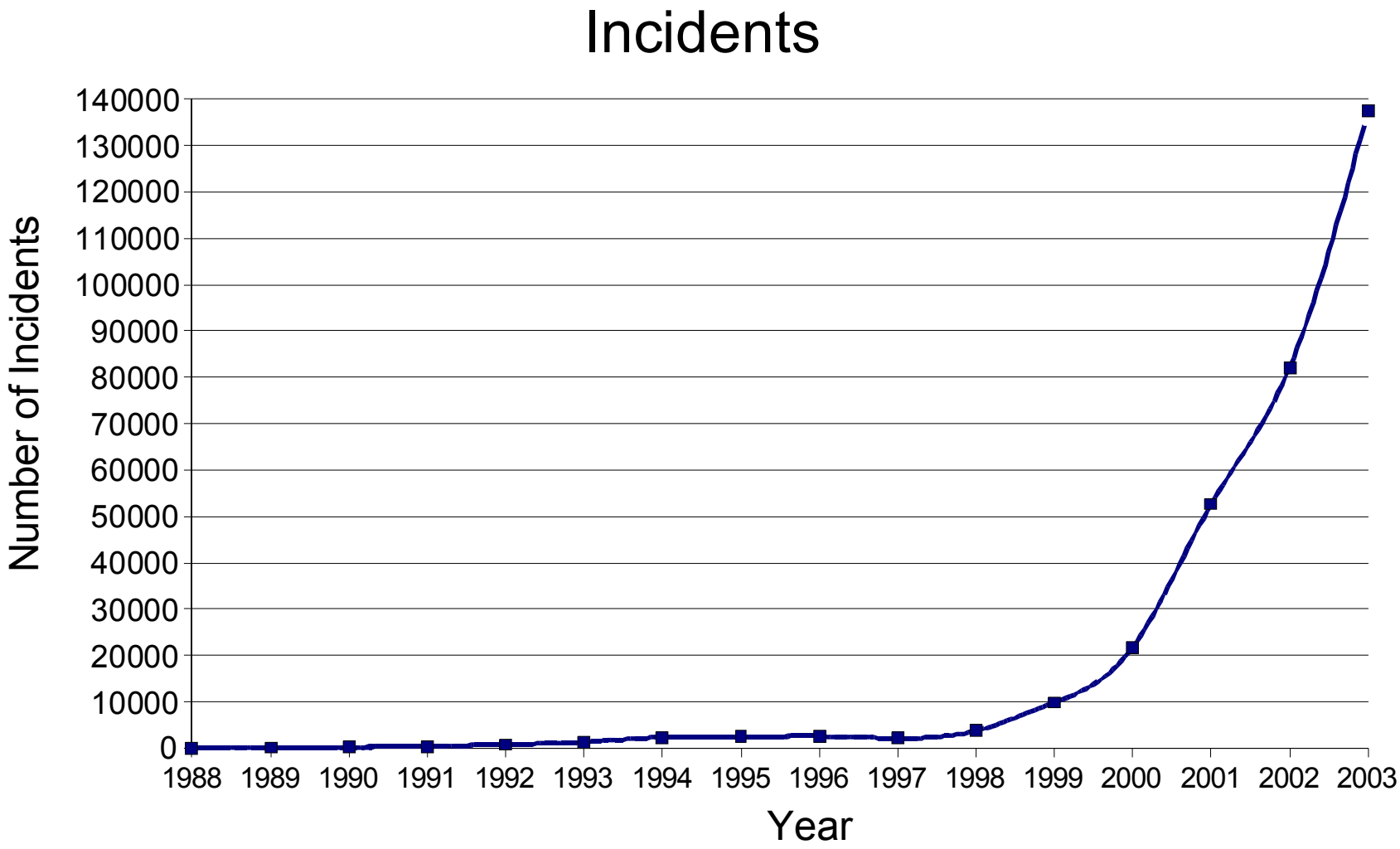
Authentification des personnes

Statistiques

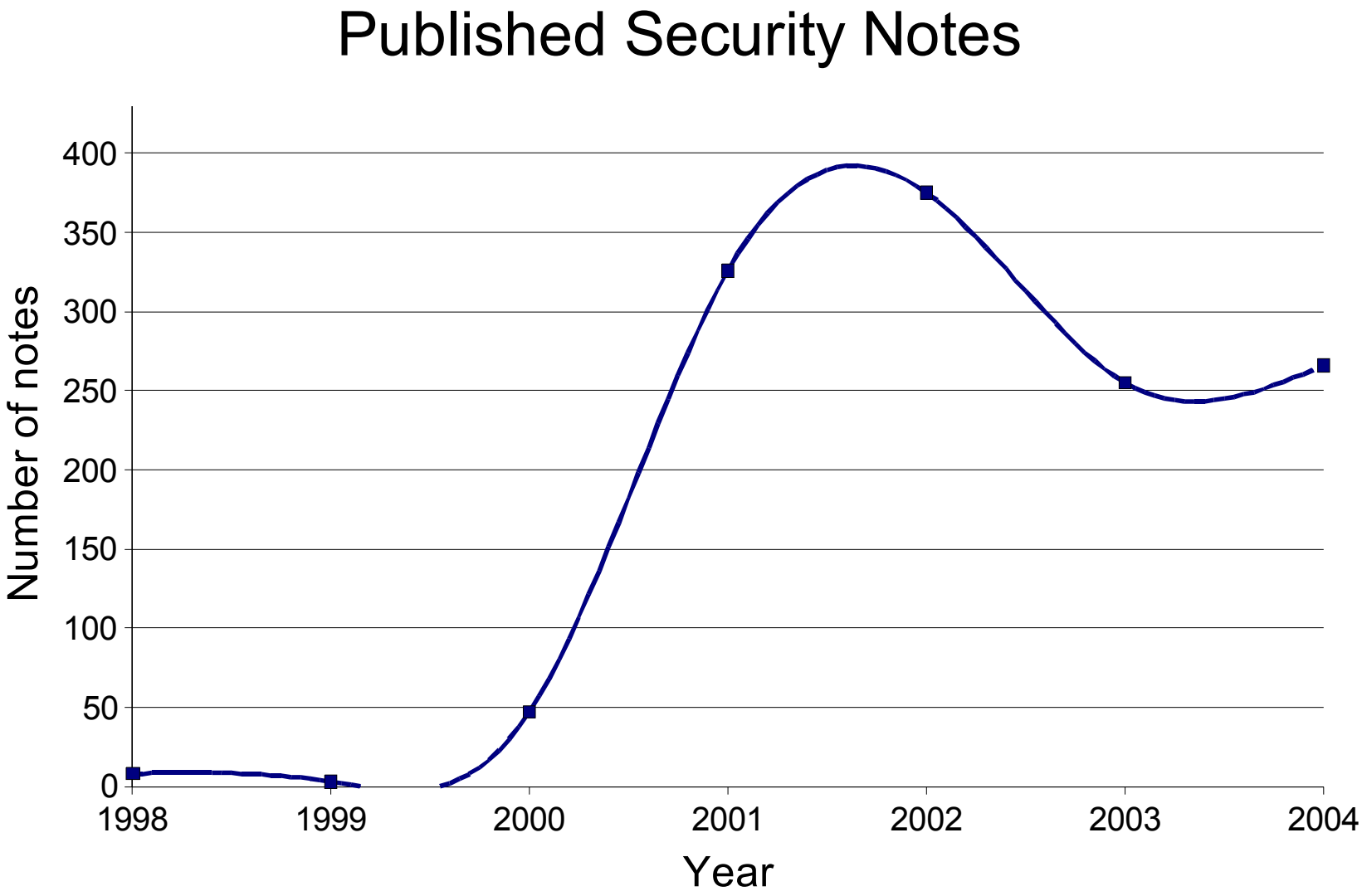
Vulnerabilities



Statistiques



Statistiques



L'homme et l'ordinateur

- Importance croissante du rôle de l'ordinateur
  - Dans la diffusion de l'information
  - via des systèmes techniques de plus en plus complexes, dans des domaines de plus en plus variés.
- Dans le passé, l'informatique était concentrée sur
  - les effets possibles d'une erreur de programmation
  - la validation de processus critique (transport, énergie...)
- Actuellement, on se concentre sur
  - le détournement possible des nouvelles technologies de l'information
  - la communication par soit des pirates / groupes étatiques / groupes industriels d'informations
- Faut-il craindre avec raison les effets pervers d'une informatisation trop rapide de la société?
  - OUI !!!!

Ce qui peut arriver à votre machine

- Refuser de faire quoi que ce soit (plantage)
- Faire trop tôt ou trop tard ce qu'il devait faire (mauvaise réactivité)
- Accomplir des actions différentes de celles attendues (« bug »)
- Etre attaquer par un pirate avec pour conséquence (attaque)
  - La destruction de vos données
  - La transformation de votre écran en une œuvre d'art minimaliste
  - L'inondation de la planète de messages pornographiques
  - L'espionnage de votre comportement et la vente de ces informations
  - L'utilisation de votre machine comme relai d'attaque
  - L'implantation d'un module de surveillance étatique

Conséquences

- Dans la majorité des cas, des conséquences assez bénignes:
  - "retaper" deux ou trois fois la même chose, suite à la « perte d'un fichier »
  - Perdre des emails, des photos, des textes
  - Subir de la publicité
  - Réinstallation d'une machine chez le particulier
- Mais pour une entreprise, des conséquences considérables:
  - la paralysie des serveurs Web,
  - le vol de sommes considérables,
  - la faillite d'une entreprise qui ne peut plus facturer
- Dans le futur ?
  - l'échec d'un tir de fusée
  - la création d'embouteillages monstrueux,
  - une panne de courant paralysant une métropole
  - une panne paralysant les transports ferroviaires d'un grande capitale

Pourquoi est-ce si important ?

- Dommages potentiels importants
  - Indisponibilités des systèmes
  - Manipulation(s) des données/systèmes
  - Destruction des données/systèmes
  - Si la puce de votre carte d'identité crash existerez-vous encore ?
- Coûts importants de remise en marche
  - Financiers (remise en marche, ventes manquées)
  - Temporels (remise en marche, configuration)
  - Humains (juristes, informaticiens)
  - Matériels (serveurs de sauvegarde, redondance, sécurité)
- Coûts importants de maintient en état
  - Analyses des données de surveillance
  - Analyses des machines

La sécurité : un problème critique !

- La sécurité des machines et des réseaux
  - Doit être la première préoccupation de toute entité utilisant l'informatique
    - Entité = gouvernement, entreprise, particulier, associations, …
    - Laisseriez-vous ouverte votre porte d'appartement ?
  - Si ignoré, des nombreux problèmes financiers
  - Doit dépendre d'ingénieurs spécialistes
- Nécessite une définition
  - Des risques de sécurité
  - D'un plan et d'une politique de sécurisation
    - architectures, droits, organisations, acteurs
  - D'un plan de réaction aux attaques

Sécurité : centres de préoccupations

- Sécurité locale (sur une machine)
  - Sécurisé l'OS
  - Sécurisé les services
  - Sécurisé l'accès à la machine (boot)
- Sécurisé réseau (services réseaux)
  - Topologie et architectures des serveurs
  - Les protocoles exploités
  - Le contrôle d'accès par firewall
- **Les utilisateurs !!!!**
  - Education comportementale
    - Peur de l'ordinateur (ligne 14, régulateurs vitesses, …)
    - La non-compréhension des outils informatiques
  - Identifications des maillons faibles (personnes à risques)
- Le cadre juridique et sociale (politique ?)

Conclusion

- Besoin en spécialistes en sécurité pour définir :
  - Quels types d'attaques l'entreprise peut subir
  - Quels moyens de protections sont à mettre en place
  - Quels mesures (contre-mesures) utiliser en cas d'attaque
  - Quels sont les contrôles sont à effectuer et à quelle fréquence
- Il n'y a aucun système sûr à 100%
- Il faut savoir choisir son degré de protection en fonction de ses besoins
  - ADEQUATION DES MOYENS ET DES OBJECTIFS
  - C'EST UNE OBLIGATION JURIDIQUE

Conclusion

Il existe donc une probabilité raisonnable de pouvoir cohabiter et même collaborer avec les ordinateurs.

Il suffit de prendre le temps de savoir ce que nous voulons en faire et comment.

Lorsque le problème est bien posé, les solutions techniques existent déjà souvent et, dans le cas contraire, seront inventées.

Bibliographie

- S. Natkin, « *Protocoles de Sécurité de l'Internet* », Dunod, 2002
- B. Schneier, « *Cryptologie appliquée* », Thomson publishing, 2001
- J. Stern ,« *La science du secret* », Odile Jacob Ed, 1998
- D. Stinson, « *Cryptologie: théorie et pratique* », Thomson publishing, 2003
- D. B. Chapman & E. D. Zwicky, « *La sécurité sur Internet — Firewalls* », O'Reilly , 1996
- Microsoft Security Team, « *Sécurité Windows* », Microsoft Press, 2005
- Les livres de poches d'Isaac Asimov (« *les robots* »)
- S. Garfinkel & G. Spafford , « *Practical UNIX & Internet Security* », seconde Édition, 1996
- B. Schneier, « *Secrets and Lies, Digital Security in a Networked World* », J. Wiley and sons ed, 2000

Sites WEB

- Commentcamarche : introduction
- Le site du cnam (G. Florin, S. Natkin) pour le module sécurité

  (de nombreuses informations sont extraites de ce cours)
- http://www.ssi.gouv.fr/ Serveur thématique sur la sécurité des systèmes d'information du Secrétariat Général de la Défense Nationale
- http://fr.wikipedia.org/ section sécurité
- http://www.cru.fr/ Comité Réseau des Universités
- http://www.urec.fr/ Unité Réseau du CNRS
- http://www.hsc.fr/ Hervé Schauer Consultants
- https://www.clusif.asso.fr/index.asp Club de la Sécurité des Systèmes d'Information Français
- www.renater.fr, www.cert.org, www.securite.org, www.ouah.org, www.securityfocus.org
- Cours sur le web :
  - Michel Riguidel (http://perso.enst.fr/~riguidel/UESecur)
  - Stephane Naktin au CNAM

Plan de cours

Utilisation des outils

Introduction

Concepts et Terminologie

Types d'attaques

Les politiques de sécurité

Les outils de la sécurité

Utilisation des CS symétriques

Utilisation des CS asymétriques

Autres Utilisations des CS

Les certificats

Authentification des personnes

Concepts et Terminologie

- La **sûreté** de fonctionnement d'un SI correspond au degré de confiance que peut accorder les utilisateurs dans le service délivré
  - **Disponibilité (avaibility)** : capacité à être prêt à délivrer le service (dans les meilleures conditions)
  - **Fiabilité (reliability)** : continuité de service (pas d'arrêt)
- La **sécurité** de fonctionnement d'un SI se décompose en deux thèmes
  - **Sécurité (safety)** : évitement des situations catastrophiques
    - Celles qui sont considérées comme inacceptables pour les utilisateurs
  - **Sécurité (security)** : préservation de la confidentialité et de l'intégrité des informations
    - la lutte contre les fautes intentionnelles (virus, bombes logiques, chevaux de Troie, etc.)

Concepts de sûreté

- Défaillance : Service délivré ≠ Service spécifié
- Erreur : état du système susceptible d'entrainer une défaillance
- Faute : Cause de l'erreur

- Relations erreurs/fautes/défaillances:
  - L'erreur est la manifestation de la faute sur le système
  - La défaillance est l'effet d'une erreur sur le service

Concept de sûreté : Faute

- Une faute devient active lorsqu'elle produit une erreur
- Une faute active est :
  - soit une faute dormante activée par le traitement
  - soit une faute externe
- Une faute interne peut passer cycliquement de l'état dormant à actif, …

Concept de sûreté : Erreur

- Temporaire par nature
- Latente ou détectée
  - Latente, tant qu'elle n'a pas été reconnue
  - Détectée, soit par les mécanismes de détection et
- traitement d'erreurs, soit par son effet sur le service (défaillance)

1 erreur ➠ propagation d'autres erreurs dans d'autres parties du système (effet papillon)

Concept de sûreté : Défaillance

- Une défaillance survient lorsqu'une erreur traverse l'interface Système/Utilisateur et altère le service délivré par le système
- Dans un système constitué d'un ensemble de composants, la conséquence de la défaillance d'un composant est :
  - une faute interne pour le composant englobant,
  - une faute externe pour les composants avec lesquels il interagit
- Type de défaillance
  - Fautes franches (*fail stop*) : arrêt pur et simple
  - Omissions : perte de messages
  - Temporaires : déviations temporelles / spécifications
  - Byzantin : comportement aléatoire ou malveillant

De l'erreur à la défaillance

... ➠ défaillance ➠ faute ➠ erreur ➠ défaillance ➠ ...

- Une erreur est susceptible de provoquer une défaillance, mais ne la provoque pas nécessairement (ou pas immédiatement)
  - Parce qu'il y a une redondance interne suffisante pour que le système continue de fournir le service
  - Parce que la partie erronée de l'état n'est pas utilisée pour telle ou telle fonction
- Une erreur est latente tant qu'elle n'a pas provoqué de défaillance
- Le temps entre l'apparition de l'état d'erreur et la défaillance est le délai de latence
  - plus le délai de latence est long, plus la recherche des causes d'une défaillance est difficile

Propriétés de sécurité

Avant de pouvoir effectivement développer des applications sécurisées, vous devez comprendre les concepts fondamentaux de la sécurité.

Les 5 piliers de la sécurité sont

Authentification

Non répudiation

Intégrité

Confidentialité

Auditabilité

Propriété de sécurité: l'authentification

C'est la propriété qui assure la reconnaissance sûre de l'identité d'une entité

- L'authentification protège de l'usurpation d'identité
- Signature = Authentification
  - Authentification: Première idée contenue dans la notion habituelle de signature
  - le signataire est le seul à pouvoir réaliser le graphisme (caractérisation psychomotrice)
- Entités à authentifier:
  - une personne
  - un programme qui s 'exécute (processus)
  - une machine dans un réseau

Propriété de sécurité: la non répudiation

C'est la propriété qui assure que l'auteur d'un acte ne peut ensuite dénier l'avoir effectué

- Signature = Authentification+Non répudiation :
  - Seconde idée contenue dans la notion habituelle de signature
  - le signataire s'engage à honorer sa signature
  - engagement contractuel/juridique, on ne peut pas revenir en arrière
- Deux aspects spécifiques de la non répudiation dans les transactions électroniques:
  - ***a) La preuve d'origine :*** Un message (une transaction) ne peut être nié par son émetteur.
  - ***b) La preuve de réception :*** Un récepteur ne peut ultérieurement nier avoir reçu un ordre s'il ne lui a pas plu de l'exécuter alors qu'il le devait juridiquement.
- Exemple: Exécution d'ordre boursier, de commande, …

Propriété de sécurité: l'intégrité

C'est la propriété qui assure qu'une information n'est modifiée que par des entités habilitées (selon des contraintes précises)

- Exemples :
  - Une modification intempestive (même très temporaire) est à interdire sur une écriture comptable validée
  - Le code binaire des programmes ne doit pas pouvoir être altéré
  - Les messages de l'ingénieur système doivent pouvoir être lus et non modifiés

Propriété de sécurité: la confidentialité

C'est la propriété qui assure qu'une information ne peut être lue que par des entités habilitées (selon des contraintes précises)

- Exemples :
  - Un mot de passe ne doit jamais pouvoir être lu par une autre personne que son possesseur
  - Un dossier médical ne doit pouvoir être consulté que par les malades et le personnel médical habilité
  - On ne doit pas pouvoir intercepter le contenu d'un courrier

Propriété de sécurité: l'auditabilité

C'est la propriété qui assure la capacité à détecter et à enregistrer de façon infalsifiable les tentatives de violation de la politique de sécurité.

- **Audit** : Examen méthodique d'une situation relative à un produit, un processus, une organisation, réalisé en coopération avec les intéressés en vue de vérifier la conformité de cette situation aux dispositions préétablies, et l'adéquation de ces dernières à l'objectif recherché [définition ISO, d'après la norme AFNOR Z61-102]
- **Auditabilité** : Garantir une maîtrise complète et permanente sur le système et en particulier pouvoir retracer tous les événements au cours d'une certaine période.

Plan de cours

Introduction

Concepts et Terminologie

Types d'attaques

Les politiques de sécurité

Les outils de la sécurité

Utilisation des CS symétriques

Utilisation des CS asymétriques

Autres Utilisations des CS

Les certificats

Authentification des personnes

Attaques sur l'authentification

- Déguisement (Mascarade)
  - Pour rentrer dans un système, on essaye de piéger des usagers et de se faire passer pour quelqu'un d'autre (usurpation d'identité)
- Exemple:
  - simulation d'interface système sur écran,
  - simulation de terminal à carte bancaire

Attaques sur l'intégrité

Types d'attaques

- **Intégrité des données : Modification de messages, de données**
  - Attribution par une personne non autorisée (usager , agent autorisé) d'avantages illicites
  - Comment ? En modifiant un fichier, un message, ...
  - Le plus souvent cette modification est réalisée par un programme et devient aussi une attaque sur l'intégrité des programmes
  - Ex : modification des données sur un serveur Web
- **Intégrité des protocoles : Répétition ("replay")**
  - Espionnage d'une interface, d'une voie de communication (téléphonique, réseau local) pour capter des opérations (même cryptées elles peuvent être utilisables)
  - Répétition de l'opération pour obtenir une fraude.
  - Exemple: Plusieurs fois la même opération de crédit d'un compte bancaire.

Attaques sur l'intégrité

- **Intégrité des programmes**
  - Les modifications à caractère
    - *Frauduleux* : Pour s'attribuer par programme des avantages (virement des centimes sur un compte)
    - *De sabotage* : Pour détruire avec plus ou moins de motivations des systèmes ou des données
  - Type de modifications
    - Infections informatiques à caractère unique
      - Bombe logique ou cheval de Troie
      - Introduction d'un comportement illicite avec un trigger
    - Infections auto reproductrices
      - Virus (reproduction rapide), Ver (reproduction lente, dormant)
      - Vecteur d'infection : secteur amorçage, infection fichier, macros virus, mutation

Attaques sur la confidentialité

- Les attaques ayant pour but le vol d'informations via un réseau par espionnage des transmissions de données (espion de ligne, accès aux données dans des routeurs et des serveurs Internet)
- Analyse de trafic : On observe le trafic de messages échangés pour en déduire des informations sur les décisions de quelqu'un.
  - Exemples: augmentation des transactions sur une place financière
  - Exemple: le début de concentration militaire entraîne un accroissement de trafic important.
- Inférence : On obtient des informations confidentielles à partir d'un faisceau de questions autorisées (et d'un raisonnement visant à faire ressortir l'information).

Attaques sur la disponibilité

- **Attaque par violation de protocole**
  - Erreur très rare en fonctionnement normal et non supportées par le protocole
  - Envoie de données non prévues (trames malformées, séquence non prévues)
- **Attaque par saturation**
  - Envoi de messages trop nombreux provoquant un écroulement des systèmes et réseaux
  - Exemple : « Distributed Denial Of Service »

Attaques sociales

- Dans la majeure partie des cas le maillon faible est l'utilisateur lui-même !
- Par méconnaissance ou duperie, l'utilisateur va ouvrir une brèche dans le système.
- Comment ?
  - En donnant des informations (mot de passe par exemple) au pirate informatique
  - En exécutant une pièce jointe
  - En discutant sur du chat
  - En ramassant une disquette/CD et en l'insérant dans un lecteur
- Aucun dispositif de protection ne peut protéger l'utilisateur contre les arnaques
  - seuls bon sens, raison et un peu d'information sur les différentes pratiques peuvent lui éviter de tomber dans le piège !

Attaques sociales

- Déroulement :
  - Une phase d'approche (ou d'accroche ☺)
    - permettant de mettre l'utilisateur en confiance
      - en se faisant passer pour une personne de sa hiérarchie, de l'entreprise, de son entourage ou pour un client, un fournisseur, sa banque, …
  - Une mise en alerte
    - afin de le déstabiliser et de s'assurer de la rapidité de sa réaction
      - prétexte de sécurité, d'une situation d'urgence
  - Une diversion (une situation permettant de rassurer l'utilisateur et d'éviter qu'il se focalise sur l'alerte)
    - Phase optionnelle
    - Ex: un remerciement, un courrier électronique, un site web, une redirection vers le site web de l'entreprise
- http://www.securityfocus.com/infocus/1527
- http://www.cert.org/incident_notes/IN-2002-03.html

Plan de cours

Introduction

Concepts et Terminologie

Types d'attaques

Les politiques de sécurité

Les outils de la sécurité

Utilisation des CS symétriques

Utilisation des CS asymétriques

Autres Utilisations des CS

Les certificats

Authentification des personnes

Etapes pour une politique de sécurité

1. Définition de la politique
   - Règles concernant les ressources informatiques

     Ressources immatériels / données
   - Règles concernant les ressources physiques

     Documents papiers, accès aux bâtiments
2. Identification des vulnérabilités
   - En mode fonctionnement normal (définir tous les points faibles)

     Ex: Arrivé/Départ de personnel, sortie de documents, entrée d'appareils électroniques, les passes des « TECHNICIENS DE SURFACES » !
   - En cas d'apparition de défaillances un système fragilisé est en général vulnérable

     Ex: Lecteur de carte HS, Contrôle biométrique inopérant, Serveur Kerberos HS
   - C'est dans un de ces moments intermédiaires qu'une intrusion peut le plus facilement réussir

     Ex: arrêt d'un firewall pour effectuer un transfert de données

Etapes pour une politique de sécurité

3. Évaluation des probabilités associées à chacune des menaces

   Ex: Danger des CDs de musique, des clefs USB

   Ex: Autoriser les accès aux sites de cracks

4. Évaluation du coût d'une intrusion réussie

   Ex: Le coût d'un vol d'information grâce à un iPod 30go ?

   (beaucoup de base de données tiennent sur 30go ...)

5. Choix des contre mesures

   Ex: Interdiction de boot sur des supports externes

   Ex: Qui peut accéder physiquement aux machines ?

6. Évaluation des coûts et de **l'adéquation** des contres mesures

   Ex: Mettre des contrôles rétiniens en dehors d'entité confidentiel défense est hors la loi.

   Ex: Pénétrer une machine à l'origine de l'attaque est interdit

7. Décision

   Application et mis en place des solutions

Cohérences des moyens

- La réalisation d'une politique de sécurité résulte de la mise en œuvre cohérente de:
  - Moyens physiques
    - architecture des bâtiments, systèmes de contrôle d 'accès, destructeurs de documents…
  - Moyens informatiques
    - Contrôles de services et des machines
  - Règles d'organisation et moyens procéduraux
    - règles de fonctionnement qui doivent être respectées

Cohérences des moyens

- Les moyens doivent être « complets »:
  - dans le cadre des hypothèses considérées, quoi qu'il arrive la politique est respectée
- Les moyens doivent être non contradictoires et raisonnablement contraignants
  - Ils ne doivent pas constituer un obstacle à la réalisation des fonctions opérationnelles de l 'organisation considérée
  - Ex: les procédures trop complexes sont souvent contournées
- Les moyens doivent être homogènes par rapport aux risques et aux attaques considérés
  - Ex: il est inutile de chiffrer tous les documents informatiques s'ils partent en clair dans les poubelles
- Le respect des procédures est un des points essentiels de l'efficacité
  - Elles doivent donc être comprises et acceptées par toutes les personnes concernées.

Principe de mise en œuvre

- Assurer la mise en œuvre d'une politique de sécurité consiste à garantir que, à chaque instant, toutes les opérations sur les objets (ressources) ne sont réalisables et réalisées que par les entités (physique ou informatique) habilitées.
- La base de la réalisation de la sécurité sont
  - **le confinement**: L'ensemble des objets sont maintenus dans des domaines étanches, l'accès ce fait via un guichet protégé.
  - **le principe du moindre privilège:** Pour qu'un système fonctionne en sécurité il faut donner à ses utilisateurs exactement les droits dont il ont besoin pour s'exécuter, **ni plus ni moins**.

Dans le détail des politiques

- Il existe différentes méthodes pour créer des politiques de sécurités
  - **MEHARI** (*MEthode Harmonisée d'Analyse de RIsques*), CLUSIF, https://www.clusif.asso.fr/fr/production/mehari/
  - **EBIOS** (*Expression des Besoins et Identification des Objectifs de Sécurité*), DCSSI, http://www.ssi.gouv.fr/fr/confiance/ebios.html
  - La norme ISO 17799
- Il existe différents standards de sécurité
  - Défini par le ministère de la défense US
    - http://www.dia.mil, http://www.radium.ncsc.mil/ (orange book)
  - Défini par la NSA (Compartemented Mode Workstation)
    - http://www.nsa.gov/
- Il existe différents standards de validation de politiques
  - CC-EAL-5 (Common Criteria Evaluation Assurance Level 5), ISO 15408
    - http://niap.nist.gov/cc-scheme/index.html

Plan de cours

Introduction

Concepts et Terminologie

Types d'attaques

Les politiques de sécurité

Les outils de la sécurité

Utilisation des CS symétriques

Utilisation des CS asymétriques

Autres Utilisations des CS

Les certificats

Authentification des personnes

Quelques définitions

- **Décrypter ou casser un code** c'est parvenir au texte en clair sans posséder au départ ces informations secrètes. C'est l'opération que doit réaliser Estelle pour retrouver M.
- **L'art de définir des codes (de chiffrement) est la cryptographie.** Un spécialiste en cryptographie est appelé cryptographe.
- **L'art de casser des codes est appelé cryptanalyse ou cryptologie**. Un spécialiste en cryptanalyse est appelé cryptanalyste.
- **Un crypto-système** est l'ensemble des deux méthodes de chiffrement et de déchiffrement utilisable en sécurité.

Ce que permet la cryptographie

- Ce que ne peut pas faire la cryptographie
  - Empêcher l'effacement des données par un pirate
  - Protéger le programme de chiffrement et son exécution (traçage, debug,...)
  - Empêcher un décodage par hasard
  - Empêcher une attaque par force brute
  - Empêcher l'utilisation méthode inédite de décodage
  - Empêcher la lecture avant codage ou après décodage
- Ne pas sous-estimer les autres méthodes de sécurité sous prétexte que la cryptographie est omnipotente

Chiffrement

Le chiffrement est donc une transformation d'un texte pour en cacher le sens.

- Les acteurs : A(lice) et B(ob) – les gentils, Estelle (l'Espionne)
- Bob, doit transmettre à Alice, un message
  $M \in$ MESSAGES_A_ENVOYER.
- M est un message dit « en clair » (non chiffré).
- Estelle écoute la voie de communication pour connaître M.
- Bob, construit un texte chiffré $C \in$ MESSAGES_CHIFFRES.
- $C = E_k(M)$ ou $C = \{M\}^E_k$
- La fonction E_k dépend d'un paramètre k appelé clef de chiffrement.
- La possibilité de chiffrer repose donc sur la connaissance de l'algorithme de chiffrement E et de la clef k de chiffrement.

Déchiffrement

Le déchiffrement est l'opération inverse permettant de récupérer le texte en clair à partir du texte C chiffré.

- Il repose sur la fonction $D_{K'}$ de MESSAGES_CHIFFRES dans MESSAGES_A_ENVOYER telle que
  - $M = D_{K'}(C)$ ou $C = \{M\}^{D}_{K'}$
- On doit avoir l'idempotence !! ➔ $D_{K'}(E_k(M)) = M$
  - K, K' sont des secrets, D et E des «algorithmes» publics
- $D_{K'}$ est donc une fonction inverse à gauche de E_k.
  - Note: il peut y avoir symétrie (D=crypter, E=décrypter)
- Pour que ces opérations assurent la confidentialité du transfert entre Alice et Bob, il est nécessaire qu'au moins une partie des informations E, D, k, K soit ignorée du reste du monde.

Efficacité du chiffrement

- L'efficacité dépend
  - Du Secret de la clé
  - De la difficulté à deviner la clé ou a les essayer toutes : lié à la taille de la clé
  - De la difficulté de l'inversion de l'algorithme de chiffrement sans connaître la clé (*cassage*)
  - De la longueur de la clé
  - De l'existence de portes par derrière (pour les gouvernements…) ou d'autres moyens plus faciles de déchiffrement
    - D'où l'importance de l'accès ou non au code source
  - Possibilité de déchiffrement par attaque à texte (partiellement) connu
- Bon système : résiste à tous les points précédents et n'offre pas d'alternative à l'essai de toutes les clés

Un outil: Crypto-systèmes symétriques

- Tels que soit **k=K**, soit la connaissance d'une des deux clefs permet d'en déduire facilement l'autre.
- Conséquences :
  - Dichotomie du monde : les bons et les mauvais
  - Multiplication des clefs (un secret n'est partagé que par 2 interlocuteurs), donc pour N interlocuteurs N*(N-1)/2 couples
- La qualité d'un crypto système symétrique s'analyse par rapport à des propriétés statistiques des textes chiffrés et la résistance aux classes d'attaques connues.
- En pratique tant qu'un crypto système symétrique n'a pas été cassé, il est bon, après il est mauvais.

CS symétriques: exemples

- Substitution mono alphabétique
  - Pour chaque lettre, on associe une lettre de substitution
  - Attaquable par la connaissance du % d'utilisation des lettres
  - Attaquable par la connaissance de la structure du message
- Substitutions de polygrammes
  - Pour chaque lettre, on associe des groupes de lettres de substitution
- Par transposition
  - On écrit le texte dans un tableau de n colonnes puis on écrit les colonnes
- CS symétriques modernes :
  - Combinaison complexe d'opérations de transposition et de substitution sur des chaines de bits (opérateurs arithmétiques) prenant comme paramètre tout ou partie de la clef.
  - Fonctionne par blocs (ECB) ou en chaîne (CBC)

CS symétriques: l'ancêtre DES

- Créé en 1978 par IBM
- Caractéristiques
  - Cryptage par bloc de 64 bits (0/1)
  - Utilise une clef de 56 bits (0/1)
  - 19 étages (étapes) d'opérations de logique combinatoire
  - Chaque étape est son propre inverse
- Performances excellentes (car basé sur des opérations logiques simples).
- Peu sécurisé car il existe des algorithmes de cassage efficace.
- Il existe des faiblesses dans le DES connu depuis longtemps par la NSA : il ne résiste pas à la cryptanalise différentielle

CS symétriques: Caractéristiques

- Les nouveaux
  - 3DES : succession de 3 DES en cascade avec 2 clefs K1 et K2 de 56bits ➔ DES_{K1},DES^{-1}_{K2},DES_{K1}
  - IDEA : longueur de clef élevé (128bits)
  - Blowfish, SAFER, AES, AES256, TWOFISH, CAST5
  - RC2, RC3, RC4, RC5, Shipjack (secret)
- Caractéristiques :
  - Débits importants
    - IDEA: 880 Kb/s (logiciel sur 386 33Mhz), 55Mb/s (circuits)
  - Une seule clef (donc partagée)
  - Clef de « petite taille »

Un outil: Crypto-systèmes asymétriques

- Tels que la connaissance de k (la clef de chiffrement) ne permet pas d'en déduire celle de K(la clef de déchiffrement). [K' ≠ k]
- Un tel crypto-système est dit asymétrique, la clef k est appelée la **clef publique**, la clef K est appelée la **clef privée**.
- Fondement théorique : montrer que la recherche de K à partir de k revient à résoudre un problème mathématique notoirement très compliqué, c'est à dire demandant un grand nombre d'opérations et beaucoup de mémoire pour effectuer les calculs.
- **RSA** (l'algorithme le plus utilisé à l'heure actuel) la déduction de K à partir de k revient à résoudre le problème de factorisation d'un grand nombre. Un problème sur lequel travaille les mathématiciens depuis plus de 2000 ans,
- On estime que le plus rapide ordinateur que l'on puisse construire utilisant la meilleure méthode connue met plus de 1000 ans pour retrouver la clef privée d'un système RSA utilisant un modulo de 1024 bits (ordre de grandeur de la taille des clefs).
- Il existe RSA, DSA, ElGammal (ancien DSA), ELG

CS asymétrique : « l'ancêtre RSA »

- **Chiffrement**
  - La clé publique est un couple d'entiers: **K = (e, n)**
  - Le chiffrement se fait au moyen de l'élévation à la puissance e modulo n: $\mathbf{E_K (M) = M^e \bmod n}$
- **Déchiffrement**
  - La clé secrète est un couple d'entiers: **k = (d, n)**
  - Le déchiffrement se fait au moyen de l'élévation à la puissance d modulo n: $\mathbf{D_k (M) = M^d \bmod n}$
- Utilisation intensive des grands nombres
  - Il existe des algorithmes pour calculer les puissances avec modulo sur des grands nombres

CS asymétrique RSA : Les clefs

- **Détermination de n**
  - Trouver **deux entiers premiers** p et q très grands
  - **Calculer n = p q**
  - p et q doivent rester secrets: La sécurité du système repose sur la difficulté de factoriser un grand entier n en deux entiers premiers p et q.
  - n doit avoir une longueur supérieure à 512 bits.
- **Détermination de e**
  - Calculer **z = (p-1) (q-1)**
  - Choisir un entier **e premier avec z.**
- **Détermination de d**
  - Choisir un entier d tel que : **e * d =1 mod z**
  - d inverse de e dans l'arithmétique mod z

La clé privée est (d,n). La clé publique est (e,n).

CS asymétrique RSA : Exemple

- **Exemple:**
  - P=7, Q=3 ➔ N = P*Q = 21
  - Z= (P – 1) * (Q – 1) = 6 * 2 = 12
  - Choisir e premier avec z: e=5
  - Choisir d tel que d*e=1 [z]

    d*5=1 mod 12 ➔ d=17
- Clef publique (d=17,n=21)
- Clef privée (e=5,n=21)
- Cryptage, décryptage du nombre 19 (M)
  - **E_K (19) = M^e [n] = 19^5 [21] = 2 476 099 [21]**
    = 117909 * 21 + 10 [21] = 10 [21]
  - **D_k (10) = 10^{17} [21] = 100 000 000 000 000 000 [21]**
    = 4 761 904 761 904 761 * 21 + 19 [21] = 19

CS asymétrique RSA : performances

- L'algorithme précédent est en O(3t) multiplications
- Multiplications sur 512 Bits= 64 multiplication en moyenne sur 32 bits. 192 multiplications pour l'élévation à la puissance.
- **Utiliser des longueurs de clés de plus en plus importantes**
  - Valeurs utilisées 512 bits, 640 bits, 1024 bits (considéré comme sûr pour plusieurs années), 2048 bits
- **Utiliser des circuits intégrés de cryptage de plus en plus performants**
  - Actuellement une dizaine de circuits disponibles.
  - Vitesse de cryptage de base pour 512 bits: de 10 à 100 Kb/s
  - Évolution en cours de l'ordre de 1 Mb/s
- **Remarque**: Compte tenu de la complexité des traitements le DES est environ toujours 10 à 100 fois plus rapide que le RSA.

CS asymétrique RSA : Conseils

- Ne jamais utiliser une valeur de n trop petite.
  - Actuellement un calcul en parallèle utilisant quelques milliers d'ordinateurs pendant quelques mois permet de factoriser des nombres d'une centaine de chiffres (400 bits)
  - Utiliser des n=1024 ou 2048 bits selon que la protection recherchée est de plus ou moins de cinq ans.
- Ne pas utiliser une clef secrète trop courte.
- N'utiliser que des clefs fortes, c'est à dire telles que p-1 et q-1 ont un grand facteur premier.
- Ne pas chiffrer des blocs trop courts (les compléter toujours a n-1 bits), de façon à détruire toute structure syntaxique [padding]
- Ne pas utiliser un n commun à plusieurs clefs si ces clefs peuvent être utilisées pour chiffrer un même message.
- Si une clef secrète (d,n) est compromise, ne plus utiliser les autres clefs utilisant n comme modulo.
- Ne jamais chiffrer ou authentifier un message provenant d'un tiers sans le modifier (ajouter quelques octets aléatoires par exemple).

Fonction à sens unique

C'est une fonction f(M) facile à calculer mais telle qu'il est extrêmement difficile de déduire M de f(M).

- Exemple:
  - Calcul modulo n (dans un anneau fini)
  - M^2 est facile à calculer modulo n (ou M^e)
  - *M* est difficile à calculer (log M)
  - CRC

Fonction de hachage

Une fonction de hachage h est une fonction qui à un message M de longueur quelconque fait correspondre un message H(M) (notée aussi $\{M\}^H$) de longueur constante.

- L'intérêt d'une fonction de hachage est que M peut être arbitrairement grand alors que $\{M\}^H$ a une longueur donnée.
- Fonction **NON BIJECTIVE**
  - elle est destructrice si taille M > longueur de $\{M\}^H$
  - Elle caractérise le bloc de données
  - Fonction a sens unique : difficulté pour retrouver M à partir de $\{M\}^H$
- Terminologie
  - Résumé, fonction de contraction, digest, empreinte digitale, ...
- Exemple:
  - « Hash codes » des systèmes de fichiers
  - codes détecteurs d'erreurs

Fonction de hachage : création

- On appelle **Fonction de hachage** :

Une fonction qui détermine la place d'une entité uniquement d'après sa clé

- H **:** clés ➔ [0..m-1] avec [0..m-1] = Espace des empruntes
- C'est la composition de deux fonctions :
  - 1) **Fonction de codage**
    - Clés ➔ entiers
  - 2) **Fonction d'adressage**
    - entiers ➔ [0..m-1]
- Caractéristiques :
  - Une bonne fonction de hachage doit faire intervenir tous les bits de la clé.
  - Une bonne fonction de hachage doit briser en sous chaînes des bits de la clé.

Fonction de hachage sécurisé

- f(M) telle que f est une fonction de hachage par rapport à M
- f est à collision faible difficile: il est calculatoirement difficile de trouver M significatif tel que f(M)=K
  - Difficulté de trouver un bloc ayant un signature K
- f est à collision forte difficile: il est calculatoirement difficile de trouver M et M' tel que f(M)=f(M')
  - Difficulté de trouver deux blocs ayant la même signature
- Elle est avec clef si son calcul dépend d'une information secrète (la clef K)
- Les algorithmes de hachages
  - Sécurisé : MD5 (emprunte 128bits), SHA1, SHA2 (256/384/512), TIGER
  - Ne plus utiliser MD5 et SHA1. Des pairs de fichiers a empruntes identiques ont été crées. Ils ne sont plus sécurisés.
  - Non sécurisé : MD2, MD4, CRC32

Les outils et la politique

- Pour des raisons d´état, besoin de décoder certaines données/messages
- Limitation par exemple de la taille des clés pour permettre une attaque force brute avec de gros moyens de calcul (officiellement ils « font de l'algèbre »)
- Mouvements sur Internet de démonstrations de craquages massivement parallèles pour pousser à l'autorisation de clés suffisamment grandes
- USA : limitation à 40 bits des clés des systèmes symétriques à l'exportation
- En France autorisation personnelle sans déclaration de clés sur 128 bits max (cryptage symétrique). Décret n99-199 du 17 mars 1999
  - www.scssi.gouv.fr
- Pacte Ukusa datant de 1948 sous contrôle de la NSA : USA + GB + Canada + Australie + Nouvelle-Zélande associés pour espionner les communications mondiales.

RENSEIGNEZ VOUS !!!

Plan de cours

Introduction

Concepts et Terminologie

Types d'attaques

Les politiques de sécurité

Les outils de la sécurité

Utilisation des CS symétriques

Utilisation des CS asymétriques

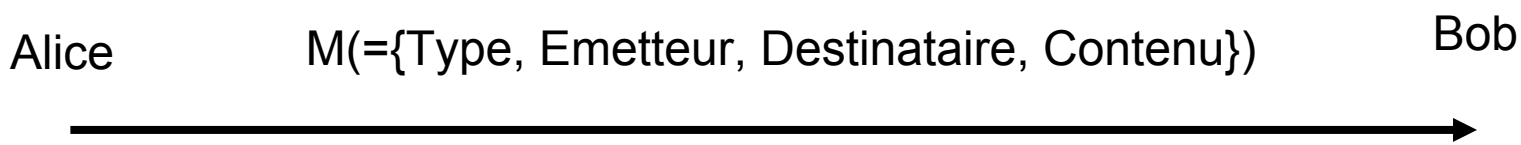
Autres Utilisations des CS

Les certificats

Authentification des personnes

Notations

- Pour chaque échange de messages, on a:
  - Type, Emetteur, Destinataire, Contenu
  - Type ➔Sémantique du message (but)
  - Emetteur ➔ expéditeur du message (identifié par @IP)
  - Destinataire ➔ récepteur du message (identifié par @IP)
  - Contenu ➔ Informations nécessaires au message
- Alice envoie a Bob le message M :



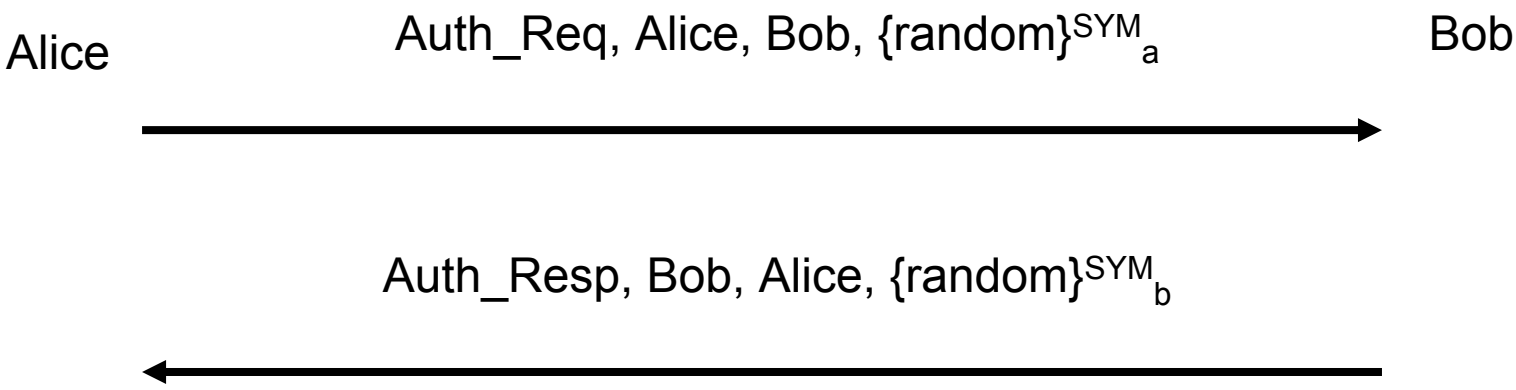
Notations

- Cryptage symétrique
  - $\{M\}^{SYM}_{clef}$ pour crypter et décrypter.
- Fonctions de hachage et signature
  - $\{M\}^{H}_{meth}$ calculer le résumé avec la méthode « meth »
- Signature d'un bloc d'informations M par Alice :
  - $\{M\}^{SIG}_{alice} = \{\{M\}^{H}\}^{SYM}_{CLEF}$

Authentification par un CS symétrique (solution 1)

- Partage des clefs privées. Connaissances des participants :
  - Alice connait
    - sa propre clef A
    - Date de validité : Date début a / Date fin a
    - la clef B de Bob
  - Bob connait
    - sa propre clef B
    - Date de validité : Date début b / Date fin b
    - la clef A de Alice
- Protocole permettant à Alice d'authentifier Bob

Authentification par un CS symétrique (solution 1)



- Assure aussi la confidentialité et l'intégrité
- Possibilité d'utiliser une clef unique pour identifier le couple Alice/Bob
- Peu extensible ➔ trop de clefs à échanger entre les acteurs !

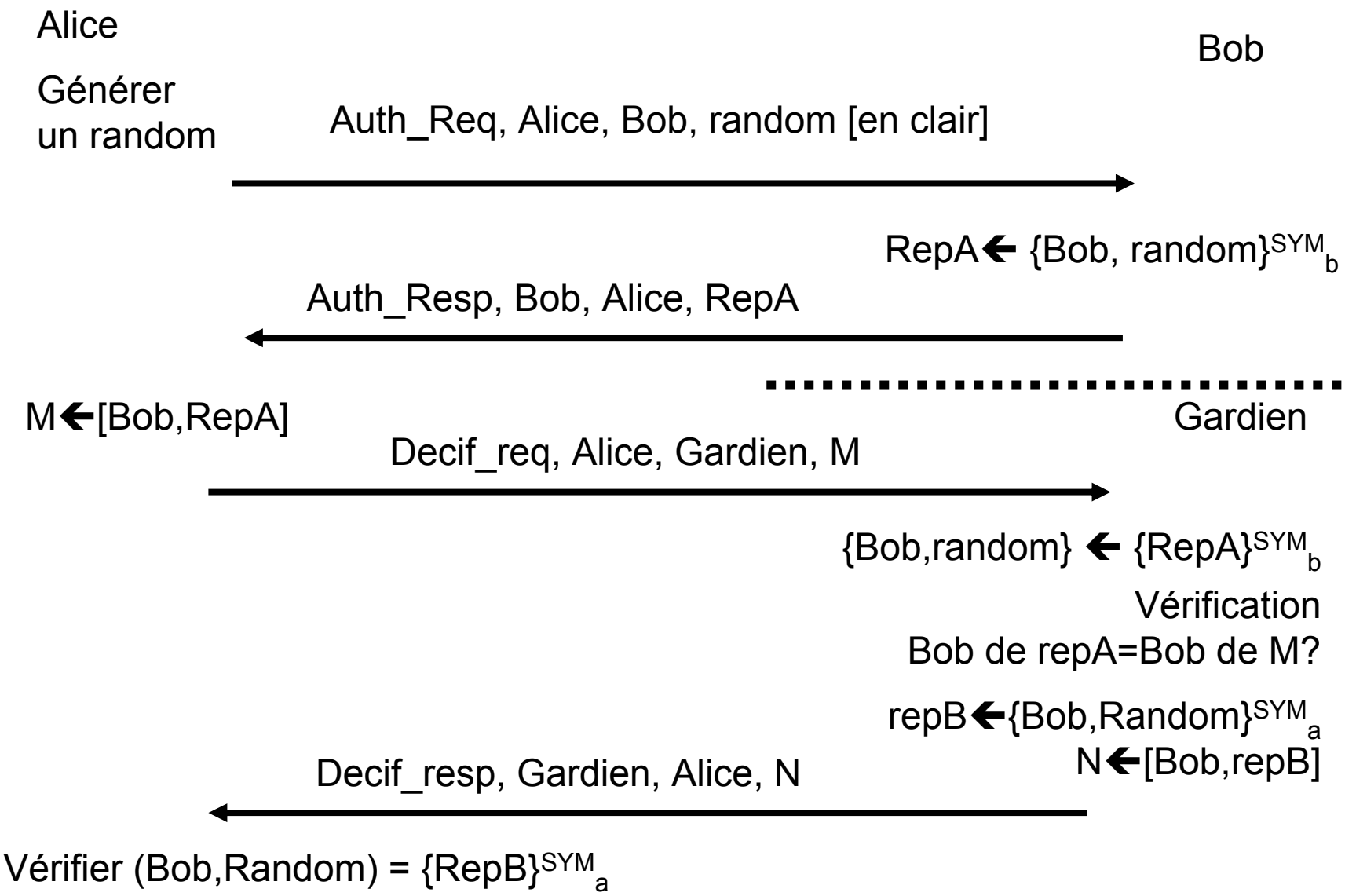
Authentification par un CS symétrique (solution 2)

- Utilisation d'un gardien des clefs
  - Exemple : kerberos [ie cerbere], le gardien des enfers
  - Chaque participant connaît sa clef et celle du gardien G
  - La gardien connait toutes les clefs (a,b,c)
  - Les informations (clefs) sont protégées en intégrité et confidentialité
  - Protection par une clef connue du gardien seulement (en général)
- Dans un système à clef privée:
  - Le gardien est un point faible
  - Il doit conserver toutes les clefs !!!
  - S'il est compromis, tout le monde l'est !!!
- Alice connait sa clef A, date de validité : date début a / Date fin a
- Bob connait sa clef B, date de validité : date début b / Date fin b
- Rappel: Le gardien connait toutes les clefs

Authentification par un CS symétrique (solution 2)

- But de l'authentification par gardien des clefs :
  - Protocole permettant à Bob de prouver à Alice qu'il est Bob
  - Bob détient un secret sur lequel repose l'authentification
  - Bob ne doit pas révéler le secret à Alice
  - Il existe un tiers fiable qui a authentifié Bob (gardien des clefs ou annuaire de certificats)

Authentification par un CS symétrique (solution 2)



Authentification par un CS symétrique : Kerberos

- **Motivations :**
  - développé au MIT pour le projet Athena
  - protéger les serveurs partagés des accès non autorisés depuis les stations de travail (plusieurs milliers)
- **Principes directeurs**
  - mode d'exécution client-serveur
  - vérification de l'identité d'un « client » (utilisateur sur une station)
  - contrôle du droit d'accès à un serveur pour le client
  - fournit au client une clé d'accès (*ticket*) pour le serveur
- Gestions des clefs:
  - Utilisation du cryptage symétrique
  - **clef différente pour chaque serveur**
  - **clef valide pour une période de temps finie**

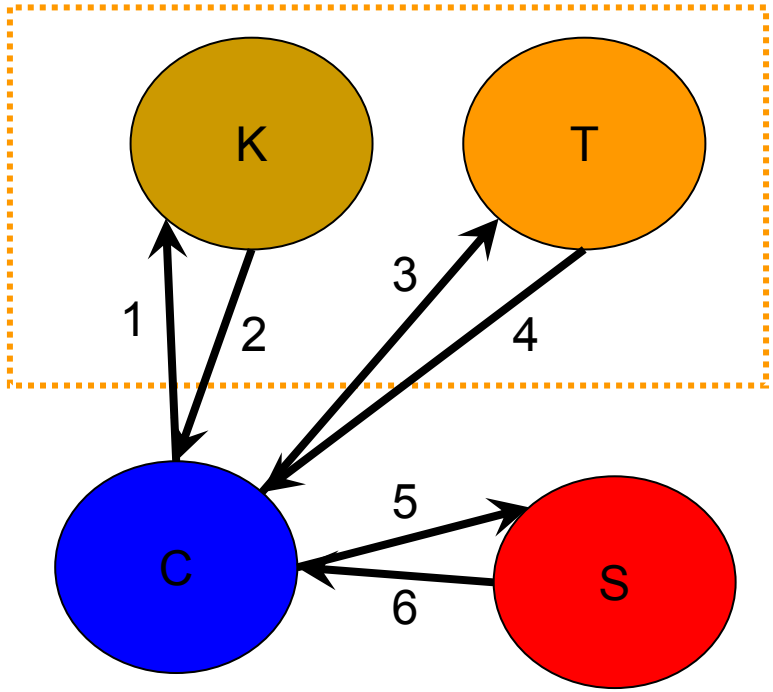
Authentification par un CS symétrique : Kerberos

- **Principe de fonctionnement : certificats « infalsifiables »**
- **Ticket** : caractérise une session entre un client C et un serveur S
  - **$Tcs = \{S, C, adr, Td, life, Kcs\}_{Ks}$**
  - **adr : adresse IP du client**
  - **Td : heure de début de session**
  - **life : durée maximale de vie la de session**
  - **Kcs : clé de session partagée par C et S**
  - **Ks : clé permanente du serveur S**
- **Authentifieur** : caractérise une autorisation pour le client à un instant t
  - **$Acs(t) = \{C, adr, t\}_{Kcs}$**
  - **généré par le client**
  - **permet une authentification « permanente » par le serveur**

Authentification par un CS symétrique : Kerberos

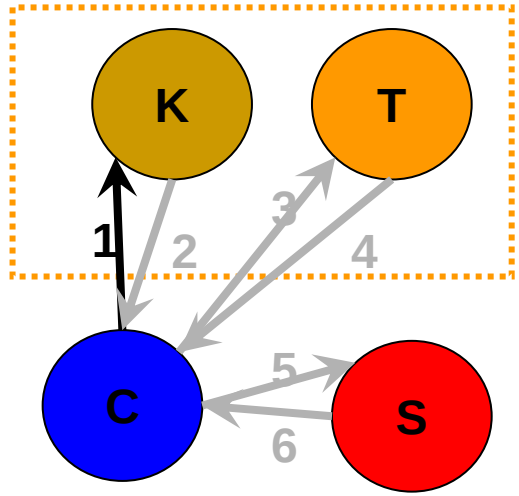
- **Architecture**

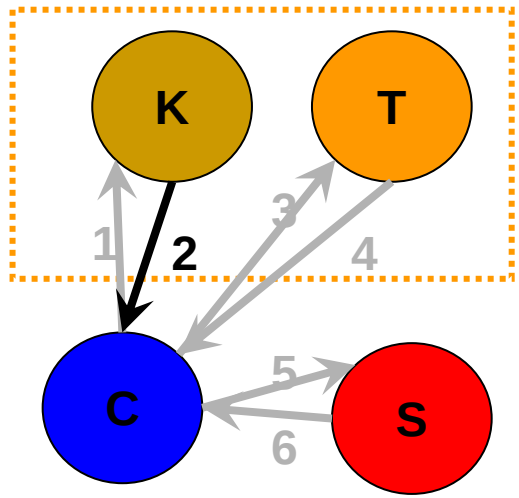
1. accès au serveur Kerberos K : authentification du client
2. Retour d'un ticket pour accéder au serveur de ticket
3. Accès au serveur de ticket T : contrôle d 'accès au serveur S
4. retour d 'un ticket pour accéder au serveur S
5. accès au serveur S



Authentification par un CS symétrique : Kerberos

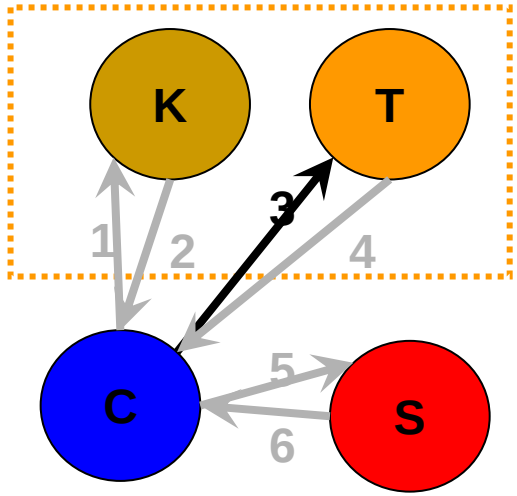
- (1) Demande par C d'un TGT (Ticket Granting Ticket) à K
  - Dé/Chiffré avec le mot de passe utilisateur
  - Message M1: tgt_req, C, K, {C, T}
  - K génère une clé de session KCT pour chiffrer le dialogue entre C et T
  - K génère un ticket TGTct pour autoriser l'accès du client C au serveur T
  - TGTct = $\{T, C, adr, td, life, KCT\}^{SYM}_{T}$
  - K connaît la clé T (de T)
- (2) Récupération du TGT par C
  - Message M2: tgt_resp, K, C, $\{\{KCT\}^{SYM}_{C}, TGTct\}$
  - C déchiffre $\{ KCT \}^{SYM}_{C}$ à l 'aide de sa clef C et mémorise la clé KCT
  - C mémorise le ticket TGTct (sans pouvoir le déchiffrer)

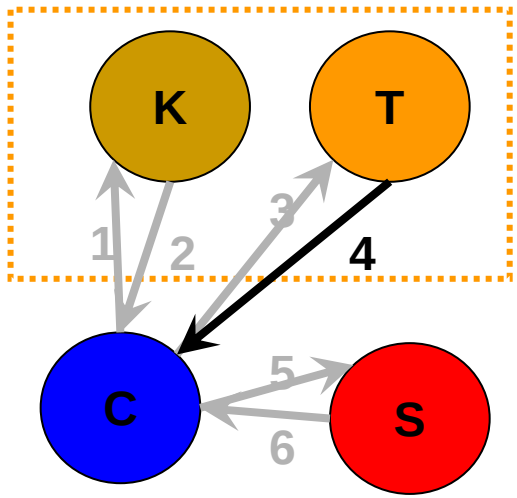




Authentification par un CS symétrique : Kerberos

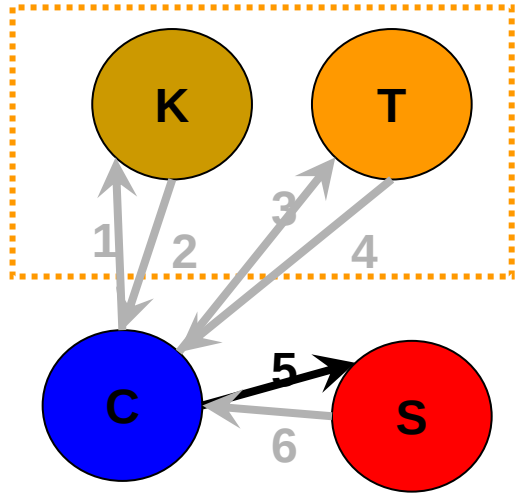
- (3) Demande d'un « *Service Ticket* » à T à l'instant **t1**
  - C construit un authentifieur : Act(t1) = {C, adr, t1}$^{SYM}_{KCT}$
  - message M3 : st_req, C, T, { Act(t1) ; TGTct ; S }
  - T déchiffre le ticket TGTct à l'aide de sa clé T, vérifie sa validité, et récupère ainsi la clé de session KCT
  - T déchiffre l'authentifieur Act à l 'aide de la clé de session KCT (obtenue dans TGTct) et récupère l'identification du client
  - T contrôle le droit d 'accès du client C au serveur S
  - T génère une clé de session **KCS** pour chiffrer le dialogue entre C et S et génère un ticket **STcs** pour autoriser l'accès du client C au serveur S
  - **STcs** = {S, C, adr, td, life, **KCS**}$^{SYM}_{S}$
  - T connaît la clef S du serveur S
- (4) Obtention du ticket
  - message M4 : st_resp, T, C, {KCS, **STcs**}$^{SYM}_{KCT}$

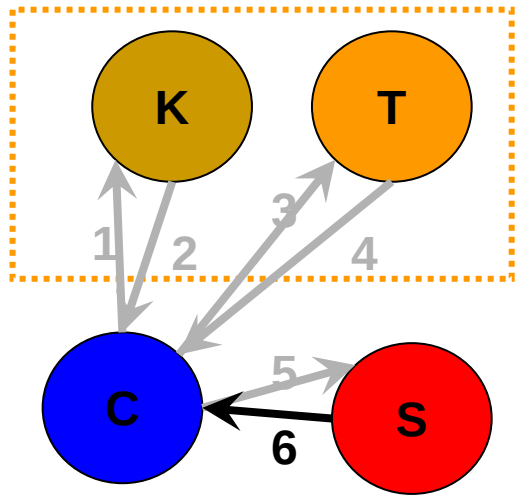




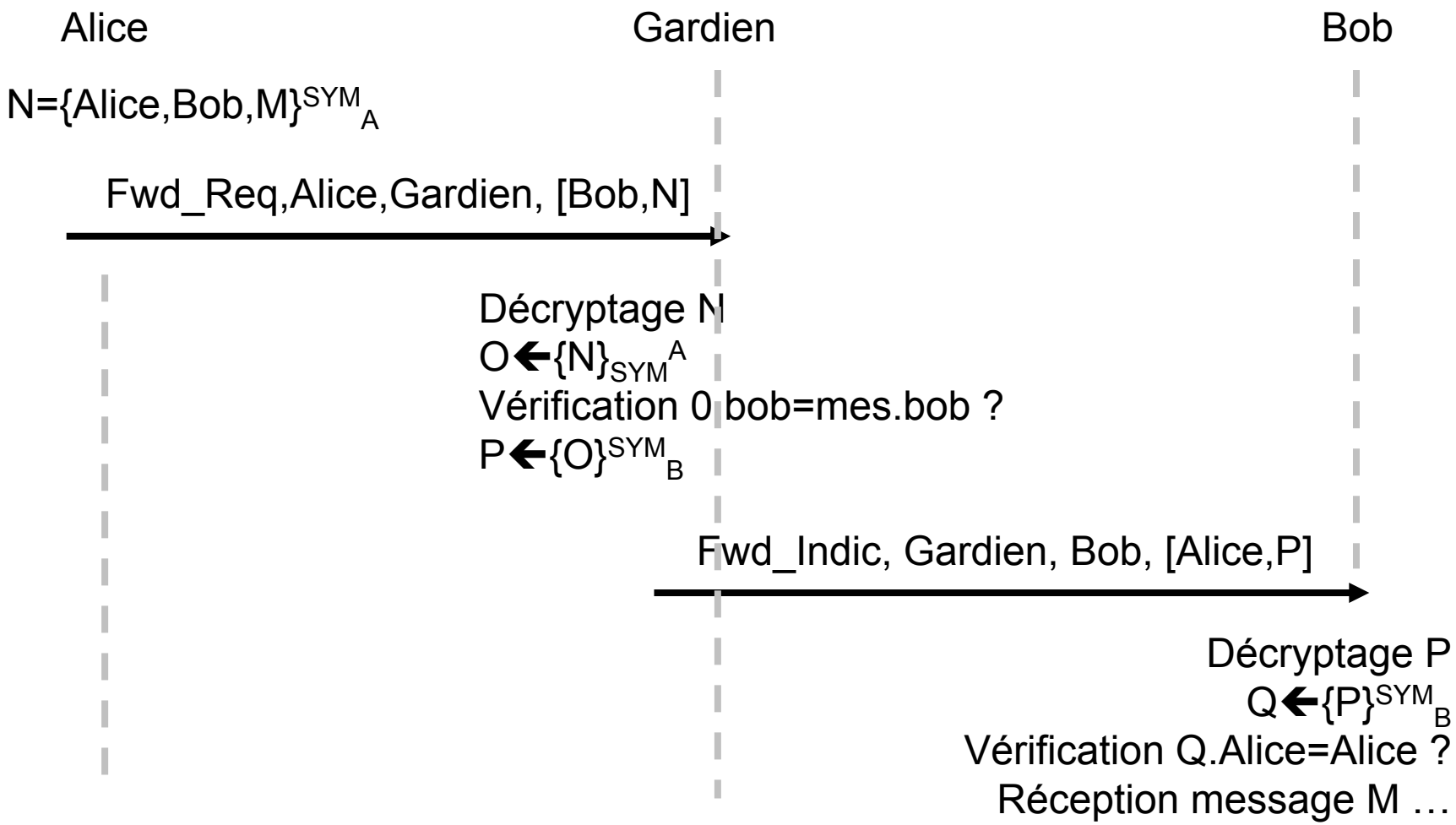
Authentification par un CS symétrique : Kerberos

- (5) Requête au serveur S
  - Rappel : message M4
    st_resp, T, C, $\{KCS, STcs\}^{SYM}_{KCT}$
  - C déchiffre le message M4 à l 'aide de la clé KCT
  - C mémorise le ticket STcs (sans pouvoir le déchiffrer) et KCS
  - C construit un authentifieur :
    $Acs(t2) = \{C, adr, t2\}^{SYM}_{KCS}$
  - Message M5: serv_req, C, S, {requête, STcs, Acs}
- (6) Traitement de la requête
  - S déchiffre le ticket STcs à l 'aide de sa clef S, vérifie sa validité (temporelle, authentification,...)
  - S récupère la clé de session KCS
  - S déchiffre l'authentifieur Acs(t2) à l 'aide de la clé de session KCS et vérifie sa validité (temporelle)





Confidentialité avec un CS symétrique



- Le message M peut être une clef privée (symétrique) de session
- On évite ensuite le gardien comme intermédiaire (maillon faible, goulot)

Intégrité et signature

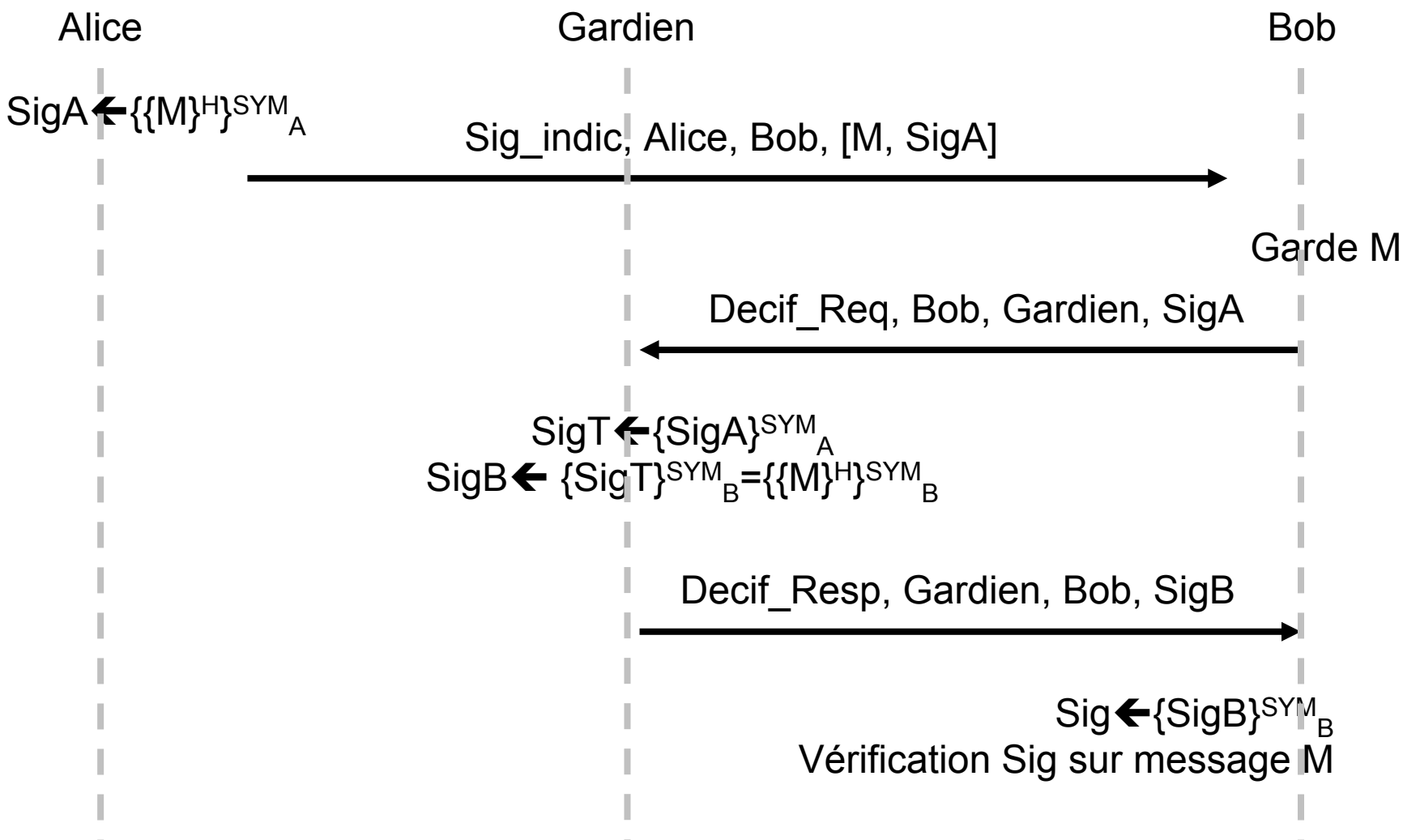
- Alice doit envoyer à Bob un message, tel que Bob puisse contrôler que le message n'a pas été modifié et a bien été créé par Alice.
- On pourrait utiliser le principe de confidentialité basé sur **la possibilité de générer des données correctes par les usagers autorisés** détenteurs du secret.
  - L'intégrité ne peut être mise en cause que par les détenteurs du secret.
  - **Problème**: La vérification de l'intégrité est alors **coûteuse** si les données sont **longues [cryptage nécessaire].**
- **Solution**: Chiffrer une information **courte** caractéristique du message grâce à une **fonction de hachage à sens unique.**

Signature

- Signature
  - Une signature manuscrite idéale est réputée posséder les propriétés suivantes:
  - La signature **ne peut-être imitée et authentifie** le signataire.
  - Elle prouve que le signataire a délibérément signé le document.
  - La signature appartient à un seul document (elle **n'est pas réutilisable**).
  - Le document signé ne peut être partiellement ou totalement **modifié**
  - La signature peut être **contrôlée et** ne peut-être **reniée**.
- Base de la signature numérique: une fonctions de hachage
  - H sécuritaire et d 'une fonction à sens unique f avec brèche.
  - La signature est composée de $f^{-1}(\{M\}^H)$
  - Seul le signataire sait calculer f^{-1}
  - Tout le monde peut calculer H et f et donc pour M donné vérifier la signature
  - Si H est a collision faible, on ne pourra pas coller une fausse signature sur un document à créer
  - Si H est à collision forte difficile Estelle ne pourra pas fabriquer 2 documents, un que Bob peut signer, l'autre pas, ayant le même résumé donc la même signature

Intégrité et signature avec un CS symétrique

Utilisation des CS symétriques



Plan de cours

Introduction

Concepts et Terminologie

Types d'attaques

Les politiques de sécurité

Les outils de la sécurité

Utilisation des CS symétriques

Utilisation des CS asymétriques

Autres Utilisations des CS

Les certificats

Authentification des personnes

Notations

- Pour chaque échange de messages, on a:
  - Type, Emetteur, Destinataire, Contenu
  - Type ➔Sémantique du message (but)
  - Emetteur ➔ expéditeur du message (identifié par @IP)
  - Destinataire ➔ récepteur du message (identifié par @IP)
  - Contenu ➔ Informations nécessaires au message
- Alice envoie a Bob le message M :

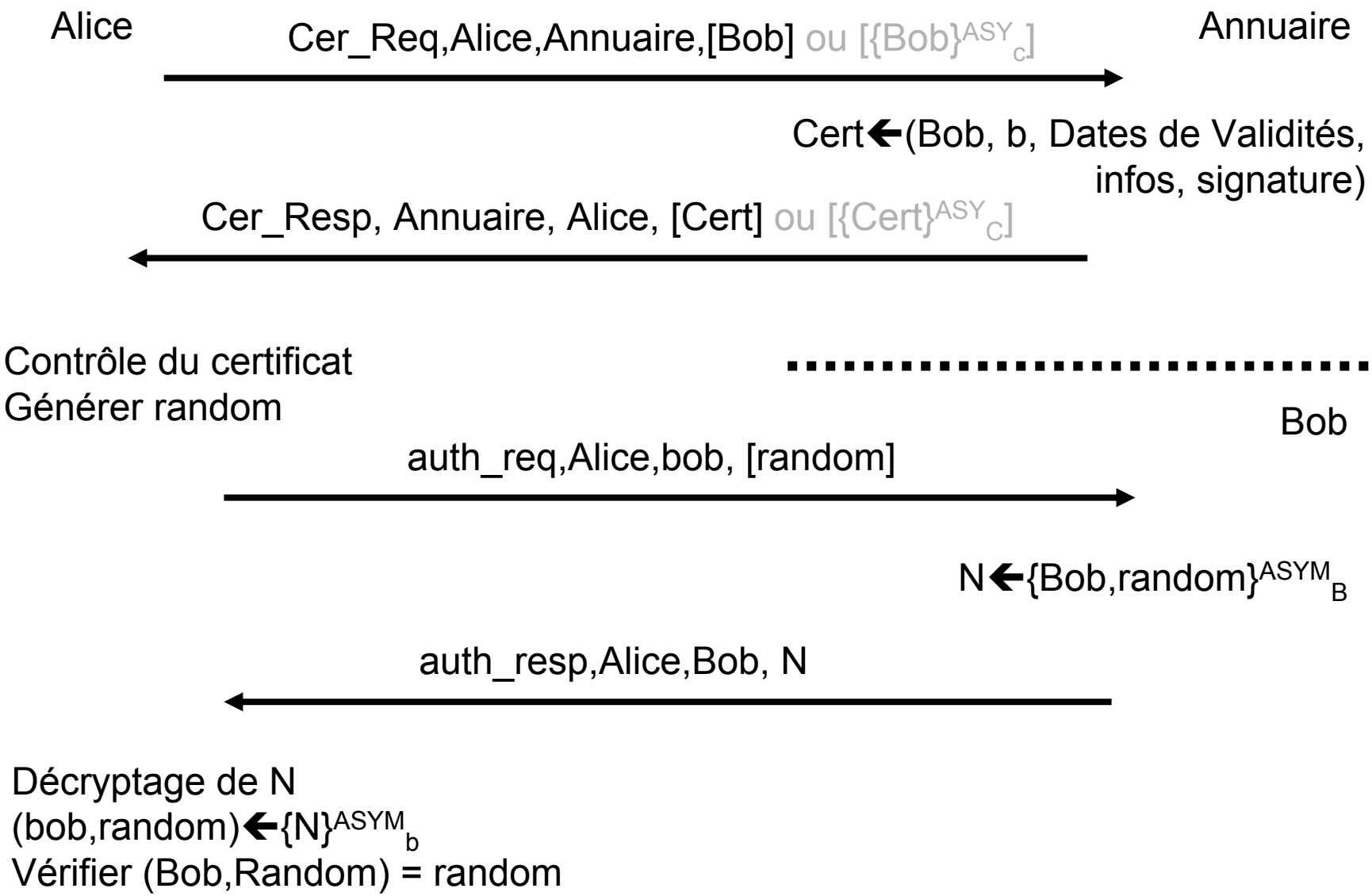
Notations

- Cryptage asymétrique
  - Si on crypte avec l'un, on décrypte avec l'autre
  - « clef » (minuscule) est la clef publique
  - « CLEF » (majuscule) est la clef privée
  - $\{M\}^{ASYM}_{clef}$ est le symétrique de $\{M\}^{ASYM}_{CLEF}$
- Fonctions de hachage et signature
  - $\{M\}^{H}_{meth}$ calculer le résumé avec la méthode « meth »
- Signature d'un bloc d'informations M par Alice :
  - $\{M\}^{SIG}_{alice} = \{\{M\}^{H}\}^{ASYM}_{CLEF}$

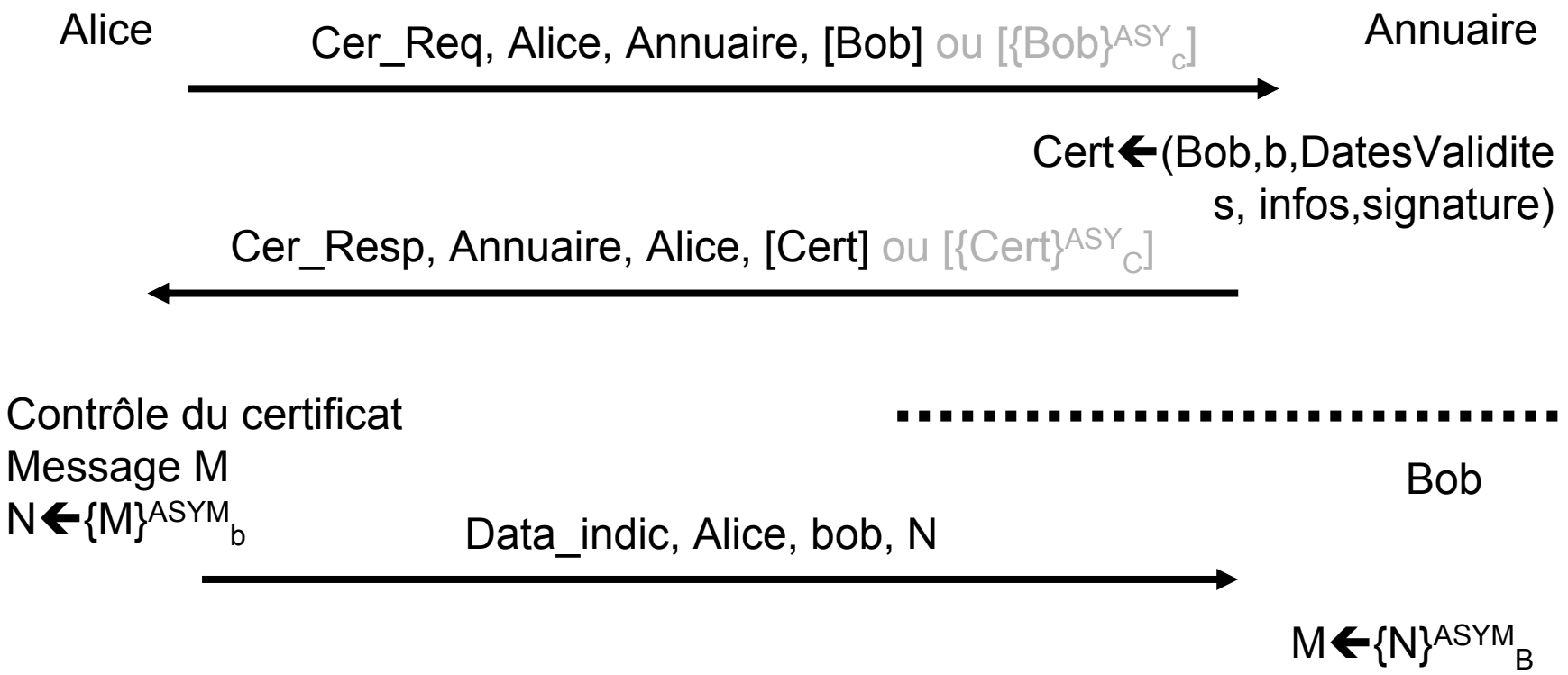
Authentification par un CS asymétrique

- Systèmes à clefs publiques: Annuaire de clefs
  - L'annuaire possède les clefs publiques des membres
  - L'annuaire a sa clef (partie publique « c » et privée « C »)
- Les informations de l'annuaire sont protégées en intégrité
- Chaque participant connaît sa clef privée, sa clef publique et la clef publique de l'annuaire
  - Alice connaît sa clef privée A / publique a, et c
  - Bob connaît sa clef privée B / publique b, et c
  - L'annuaire connaît C / c, le certificat de a (donc clef a) et de b (donc la clef b)

Authentification par un CS asymétrique (solution 2)

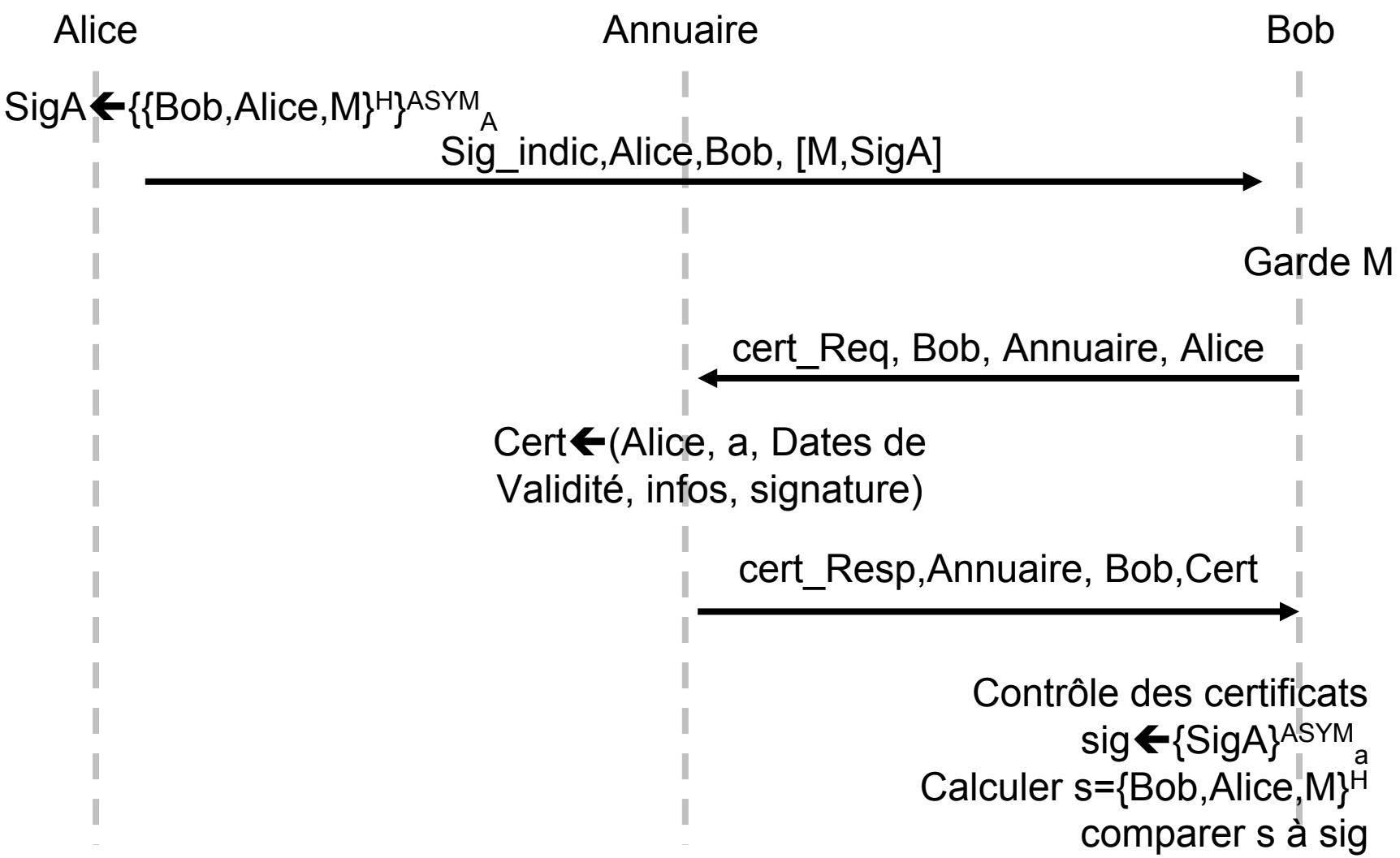


Confidentialité avec un CS asymétrique



- Rappel: très peu utilisé car très lent !!
- On utilise le cryptage asymétrique pour échanger une clef de session
  - Principe de SSH

Intégrité et signature avec un CS asymétrique



Intégrité de flots de messages

- Un flot d'échanges de longue durée doit être caractérisé par une **connexion**.
  - Problème de rejeu ("replay")
  - Un message dupliqué **peut être inséré dans un flot par un usager malveillant**
  - Il peut être correct du point de vue de la connexion, séquence et signature mais menacer l'intégrité de l'application
- Rejeu possible d'un message
  - D'une ancienne connexion
  - De la connexion courante
- Intégrité du flot de message
- Utilisation d'un **Nonce** (Used Only Once), qui distingue chaque message:
  - Numéro de séquence sur un modulo grand (sur 32 ou 64 bits)
  - Estampillage par horloge (horodatage)
  - Nombre aléatoire

Considérations ad-hoc: Stockages des clefs

- Clef publique de l'autorité, ne doit pas pouvoir être modifiée
  - Dans le code en dur
  - sur un support fiable (carte à puce)
- Clef privée de l'utilisateur, ne doit pas pouvoir être lue:
  - sur un support confidentiel (carte à puce) ou un fichier chiffré avec un mot de passe (local au poste ou sur disquette)
  - SSH ➔ clef privée droit 700, clef sur stick USB
- Certificat de l'utilisateur:
  - Annuaire+support local ou carte ou disquette
  - Annuaire central+version locales (cache, annuaire privé)

Considérations ad-hoc: Utilisation des clefs

- Plus on utilise une clef plus elle est vulnérable !
  - Clef utilisée pour chiffrer une suite de transfert de fichier
  - Clef utilisée pour chiffrer un numéro de carte bleue
- Plus elle sert a protéger des données pérennes, plus elle doit être fiable
  - Signature électronique d'un article de presse
  - Signature électronique d'un testament
- *On peut utiliser des canaux très lents mais très fiables pour véhiculer des clefs qui seront utilisées sur des voies plus rapides et moins fiables*

Plan de cours

Introduction

Concepts et Terminologie

Types d'attaques

Les politiques de sécurité

Les outils de la sécurité

Utilisation des CS symétriques

Utilisation des CS asymétriques

Autres Utilisations des CS

Les certificats

Authentification des personnes

Partage de secret: protocoles à seuil

- Certaines opérations sont suffisamment sensibles pour devoir **engager la responsabilité de plusieurs** personnes.
- On peut faire **vérifier l'identité** de plusieurs usagers simultanément possesseurs d'un **mot de passe** pour engager une action. [ex: attaque nucléaire]
- Mais cette approche peut ensuite être encore raffinée en souhaitant **donner une part de responsabilité plus importante** selon un grade:
  - Ex : Il suffit de la présence du responsable financier pour ouvrir le coffre ou de trois chefs de service ou ...
- **Le problème du partage d'un secret:**
  - Comment diviser une clé d'accès représentée par **une valeur numérique V en k parts**
  - De telle façon qu'un groupe de porteurs de t+1 parts peuvent reconstituer la clé alors qu'un groupe de porteurs de t parts ne le peuvent pas.
  - Les porteurs de parts **doivent pouvoir reconstituer V** dans un système informatique d'autorisation sans jamais connaître V.

Partage de secret: protocoles à seuil (Shamir)

- V **valeur numérique** entière
  - On **génère aléatoirement t valeurs entières** a_1 , a_2 , ... , a_t
  - **On leur associe un polynôme** dont le terme constant est **V** :
- $P(x) = a_t.x^t + a_{t-1}.x^{t-1} + ... + a_1 x + V$
- Une part du **secret est un couple (x_i,P(x_i))** x_i non nul
  - les parts sont générées par des x_i différents
- Pour éviter une possible **attaque force brute** par un groupe de porteurs agissant par essais et erreurs pour compléter leur connaissance
  - on choisit un entier premier N grand
  - les calculs sont faits en arithmétique modulo N
- Pour retrouver le secret, niveau 2^{nd} ➔ résolution d'un système à (t+1) valeurs inconnues (a_1 , a_2 , ... , a_t ,V) avec (t+1) équations [parts]

Notarisation par un CS asymétrique

- But: non répudiation d'une action (commande) entre A et B
- Moyen: Utilisation d'un tiers de confiance (N)
- A connait la clef publique de N (kn), la clef publique de B (kb), sa clef KA/ka
- B connait la clef publique de N (kn), la clef publique de A (ka), sa clef KB/kb
- N connait la clef publique de A (ka), la clef publique de B (kb), sa clef KN/kn

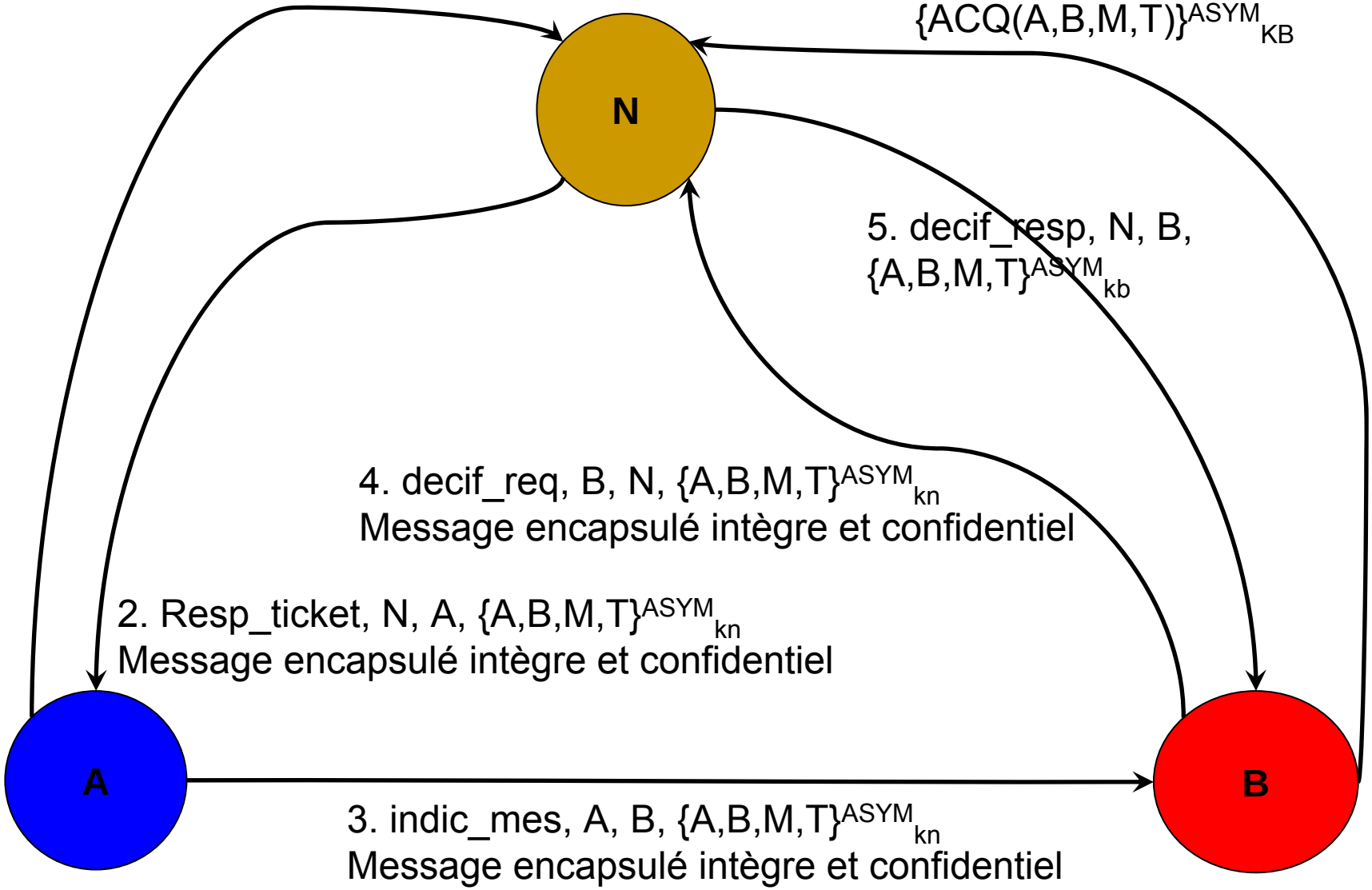
Notarisation par un CS asymétrique

1. **A souhaite envoyer le message M** au destinataire B de façon notarisée (notariée ?)
   - il envoie au notaire $\{A,B,M\}^{ASYM}_{KA}$
2. **Le notaire qui reçoit la transaction** peut la décoder puisqu'il connaît **ka**.
   - Il date la transaction T et la journalise
   - Enregistrement de A , B , M , T
   - La transaction ne pourra pas ensuite être reniée par A.
3. **Le notaire possède une clé secrète personnelle KN** qu'il utilise pour signer la transaction
   - $M= \{A,B,M,T\}^{ASYM}_{KN}$, $T=\{A,B,M,date\}^{H}$
   - Il renvoie le message M en réponse à A qui va la conserver pour preuve de la notarisation effectuée.

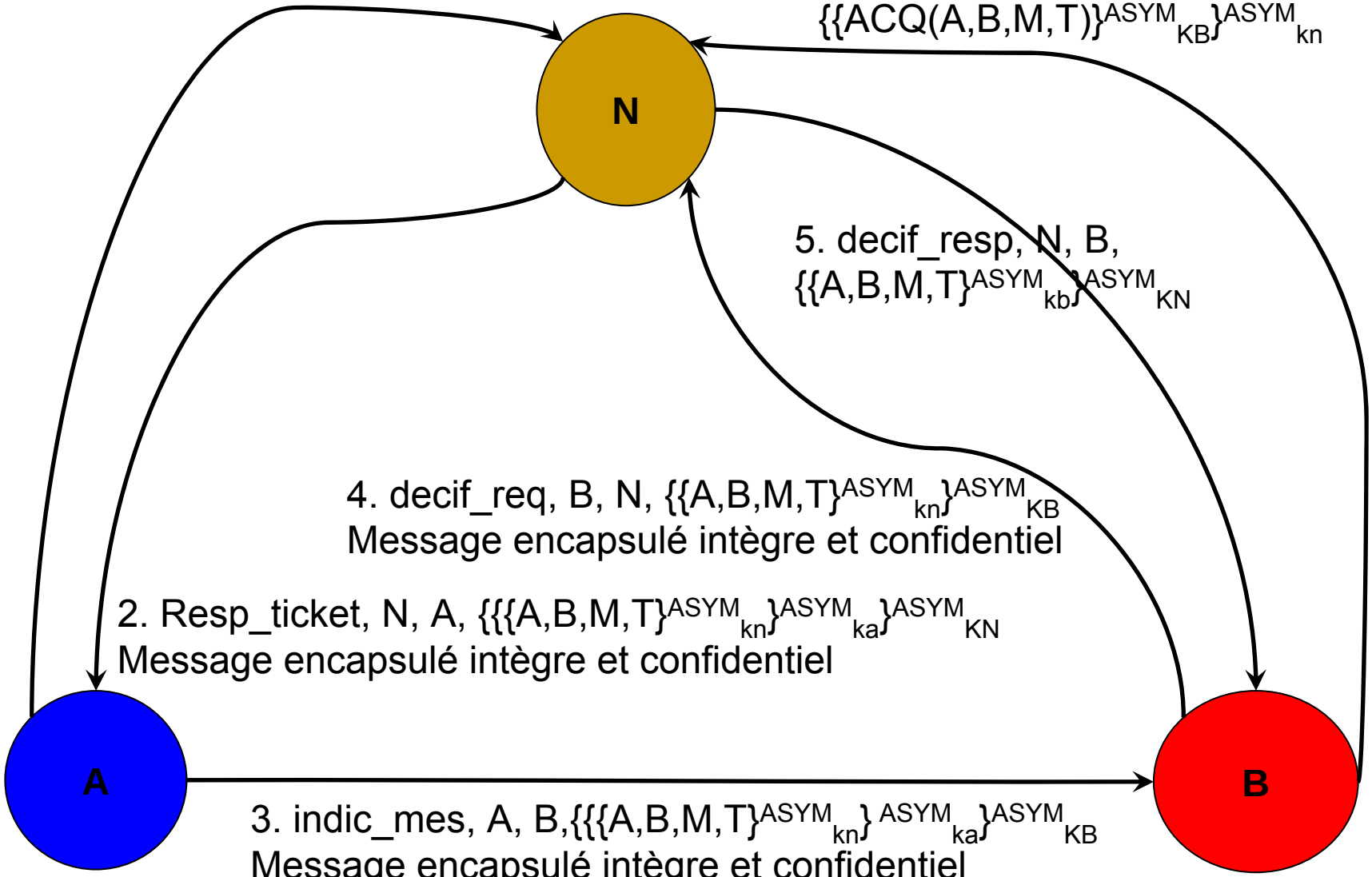
Notarisation par un CS asymétrique

4. **L'émetteur A envoie alors la transaction** à son destinataire B sous la forme M signée par le notaire (il ne peut avoir modifié celle ci entre temps).
   - B ne peut encore interpréter les informations mais il enregistre M pour preuve de la requête de A.
5. **Pour connaître M, B demande au notaire** le déchiffrement de M
   - Le notaire envoie à B la transaction chiffrée avec la clé de kb soit $\{A,B,M,T\}^{ASYM}_{kb}$.
   - Seul B peut la lire ➔ confidentialité l'intégrité
   - Pour l'authenticité, il faudrait $\{\{A,B,M,T\}^{ASYM}_{kb}\}^{ASYM}_{KC}$
6. **Pour terminer complètement le protocole** il faut que le notaire dispose d'une preuve de remise à B soit une réponse $\{ACQ(A,B,M,T)\}^{ASYM}_{KB}$ que le notaire enregistre.

Notarisation par un CS asymétrique



Notarisation par un CS asymétrique



Plan de cours

Introduction

Concepts et Terminologie

Types d'attaques

Les politiques de sécurité

Les outils de la sécurité

Utilisation des CS symétriques

Utilisation des CS asymétriques

Autres Utilisations des CS

Les certificats

Authentification des personnes

Certificats et cryptages asymétriques

- Rappel : Cryptage asymétrique
  - PB : Tout repose sur la confiance dans la provenance de la clef publique.
  - Attaque du type Man-In-The-Middle :
    Celui qui souhaite écouter les messages de votre correspondant vous remet une fausse clef publique pour cette personne.
  - Exemple SSH ne gère pas les certificats
- Solution : une autorité est chargée de signer les clefs publiques
  - Cette autorité s'appelle l'autorité de certification (« Certificate Authority » ou CA)

Les autorités de certifications (CA)

- AC: autorité de certification
  - Norme de représentation des certificats X509
  - Norme de protocole d'accès: LDAP
  - Elle chiffre (avec sa clef privée) une empreinte de
    - L'identité de son titulaire, personne, serveur ou application (*Distinguished Name of Subject*)
    - Sa (celle du titulaire) clef publique
    - Les informations relatives à l'usage de cette clef : période de validité, type des opérations possibles, etc …
  - L'ensemble est appelé certificat X509.
  - Les certificats X509 font l'objet d'une norme : ITU-T X509 international standard V3 1996, RFC2459

Les autorités de certifications (CA)

- Rôle :
  - Vérifie les demandes de certificats (Certificat Signing Request)
  - Génère les certificats et les publie
  - Génère les listes de certificats révoqués (Certificat Revocation List)
- Vérifie (par l'intermédiaire d'une autorité d'enregistrement) :
  - l'identité des demandeurs de certificats et les éléments de la demande
  - Recueille et vérifie les demandes de révocation

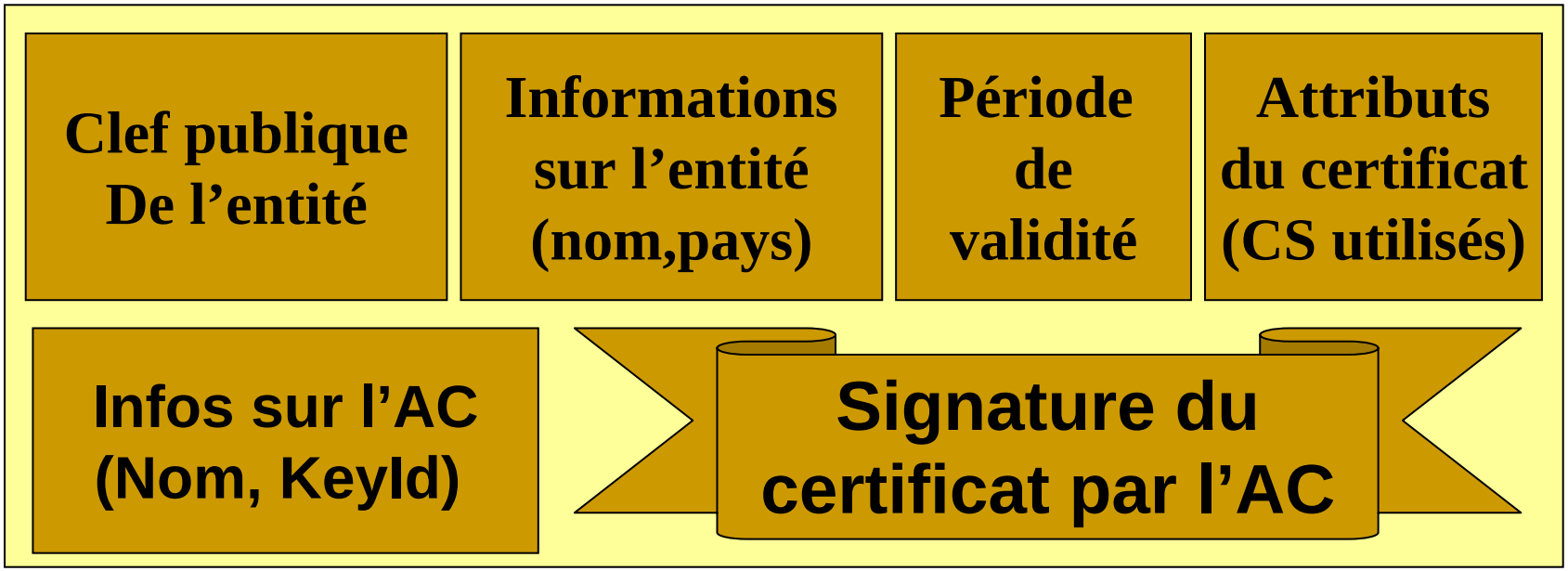
Les autorités de certifications (CA)

- Contrôle des certificats
  - Toutes entités impliquées dans un schéma à clef publique doit détenir la clef publique de l'autorité de certification.
  - Tout accès à un certificat doit être contrôlé:
    - Vérifier que la signature est valide
    - Vérifier que la date courante est dans la période de validité
    - Vérifier la clef publique du certificat en vérifiant la signature qui y a été apposée à l'aide de la clef publique de l'autorité de certification (CA).
  - Pour éviter les rejeux de certificats invalidés le serveur d'annuaire doit :
    - Soit s'authentifier
    - Soit dater et signer sa réponse
    - Soit transmettre périodiquement des listes de révocation datée et signées
- **On ne fait confiance qu'aux clefs signées !!!**
- **Attention, il existe des certificats auto-signés (sans CA officiel)**

Utilisation des certificats

- Les certificats sont composés de deux éléments :
  - d'informations sur l'entité propriétaire du certificat
  - d'informations sur l'entité émettrice du certificat (l'annuaire, l'autorité de certification, ...)
- Les informations sur l'entité propriétaire du certificat sont
  - le nom,
  - la clef publique de l'entité à identifier,
  - des informations supplémentaires
- Les informations sur l'entité émettrice du certificat
  - Date de validité du certificat,
  - Le but de la clef
  - L'emprunte du certificat (faite par l'annuaire)
  - Le nom de l'entité émettrice (annuaire)
  - Des informations concernant les algorithmes utilisés

La structure des certificats



- Le certificat établit un lien fort entre le nom (DN) de son titulaire et sa clé publique
  ➔ AUTHENTIFICATION FORTE
- Protocoles : TLS/SSL, S/MIME, VPN, Java, …
- Usages : Horodatage, Signature, E-commerce, E-vote, E-Administration, …

Exemple de certificat

Certificate:

Data:

Version: 3 (0x2)

Serial Number: 13805 (0x35ed)

Signature Algorithm: sha1WithRSAEncryption

Issuer: C=FR, O=CNRS, CN=CNRS-Standard

Validity

Not Before: Apr 24 14:09:48 2006 GMT

Not After : Apr 24 14:09:48 2008 GMT

Subject: C=FR, O=CNRS, OU=UMR7606, CN=src.lip6.fr/emailAddress=postmaster@lip6.fr

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

00:ec:29:c5:24:d6:4d:e4:b5:31:71:46:2f:15:64:

…

a6:ee:85:31:22:de:74:d8:d1:5f:8a:32:e0:b3:d7:

84:e4:8f:ab:66:92:ad:f8:eb

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Basic Constraints: critical

CA:FALSE

Netscape Cert Type:

SSL Client, SSL Server

X509v3 Key Usage: critical

Digital Signature, Non Repudiation, Key Encipherment

X509v3 Extended Key Usage:

TLS Web Server Authentication, TLS Web Client Authentication

Netscape Comment:

Certificat serveur CNRS-Standard

X509v3 Subject Key Identifier:

79:F7:B4:D3:D8:E9:B8:ED:3C:A1:85:A6:DD:FA:68:CC:74:8C:82:1F

X509v3 Authority Key Identifier:

keyid:67:59:A5:E5:07:74:49:03:EF:05:CF:CC:2E:A4:18:D5:10:C8:9E:3C

DirName:/C=FR/O=CNRS/CN=CNRS

serial:02

X509v3 Subject Alternative Name:

DNS:src.lip6.fr

X509v3 CRL Distribution Points:

URI:http://crls.services.cnrs.fr/CNRS-Standard/getder.crl

Signature Algorithm: sha1WithRSAEncryption

54:a4:1c:c2:21:fd:06:9b:df:bd:50:4b:d2:ae:e0:3f:46:64:

Les certificats: Aspects Juridiques

- Usages : Horodatage, Signature, E-commerce, E-vote, …
- Validité de l'écrit électronique
  - Reconnaissance juridique de la signature électronique
  - Obligation de dématérialisation des procédures
  - E-Administration (déclaration d'impôts)
- Le cadre est défini par
  - La loi du 13 mars 2000
  - Le décret du 30 mars 2001
  - Le décret du 18 avril 2002
  - L'arrêté du 31 mai 2002.

Création de certificat (Annuaire publique)

Alice

Annuaire
Autorité de certification (AC)

Obtention du certificat de l'AC ←

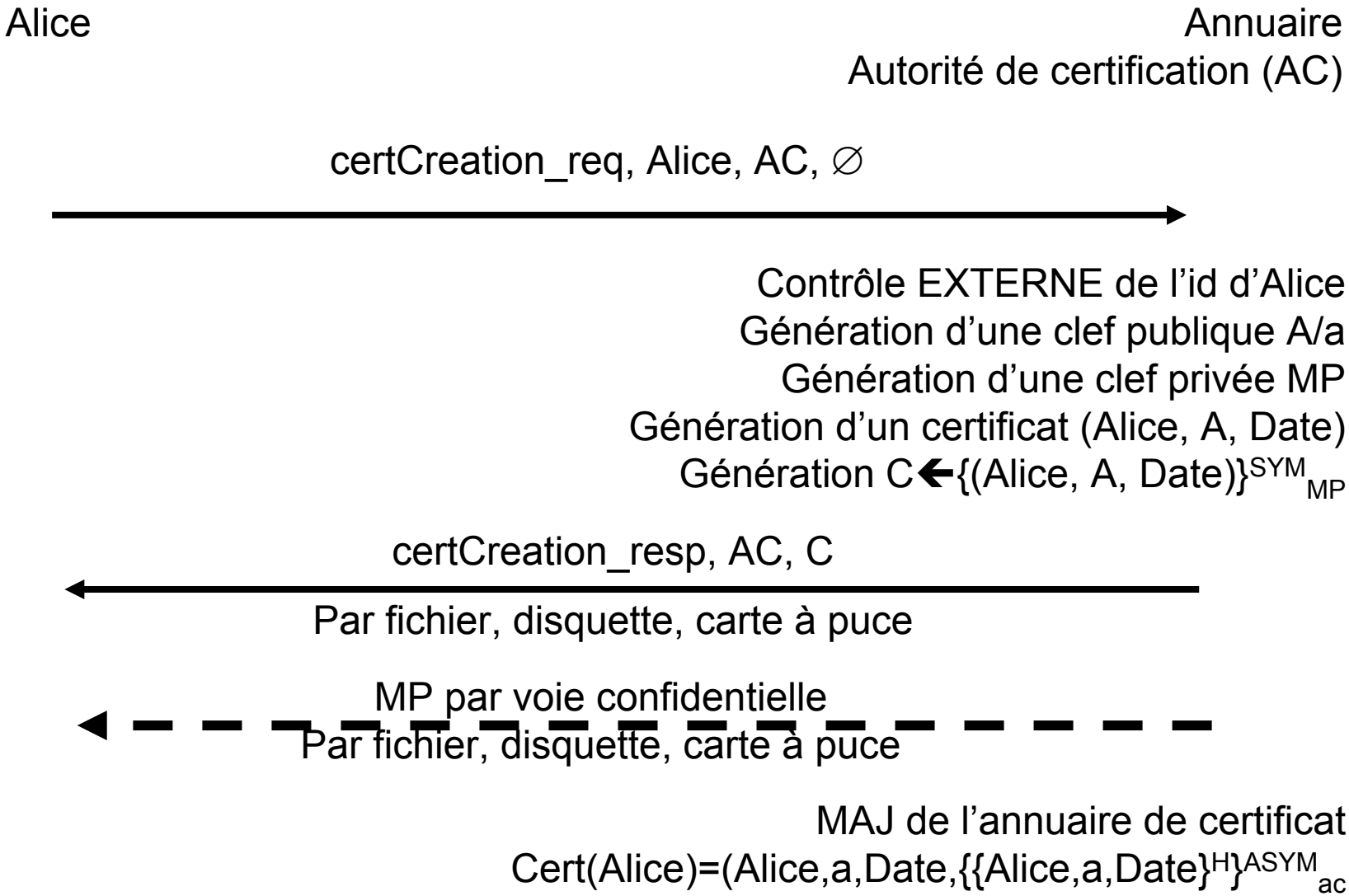
Génération d'une clef publique A/a
Génération d'une clef privée MP
Génération d'un certificat (Alice, A, Date)
Stockage (sûr) $\{(Alice, A, Date)\}^{SYM}_{MP}$
C← $\{(Alice, a, Date)\}^{ASYM}_{ac}$

certCreation_req, Alice, AC, C →

Décryptage de C
D← $\{(Alice, a, Date)\}^{ASYM}_{AC}$
Contrôle EXTERNE de l'id d'Alice
MAJ de l'annuaire de certificat
Cert(Alice)=$(Alice,a,Date,\{\{Alice,a,Date\}^{H}\}^{ASYM}_{ac}$

certCreation_resp, AC, Alice, [Cert(Alice)] ←

Création de certificat (Annuaire privé)



Révocations de certificats

- CRL = « certificat revocation list »
- Les CRL : la liste des certificats révoqués, liste signée par la CA
  - Similaire à l'opposition des CB/chèque en cas de vol
  - Pas encore de CRL incrémentale (le certificat contient une url du fichier de crl)
  - La révocation est une limite théorique au modèle des PKIs.
- Les navigateurs doivent vérifier par eux-même les CRL
  - Mal implémenté ➔ souvent non vérifié

Plan de cours

Introduction

Concepts et Terminologie

Types d'attaques

Les politiques de sécurité

Les outils de la sécurité

Utilisation des CS symétriques

Utilisation des CS asymétriques

Autres Utilisations des CS

Les certificats

Authentification des personnes

Rappels & généralités

- Le contrôle d'accès est la base des mécanismes informatiques:
  - Il permet de spécifier la politique dans le domaine de l'informatique.
  - Il définit la façon dont le système contrôle ces droits.
  - Il devrait, en théorie, encapsuler toutes les autres techniques informatiques
  - Pour l'instant ce n'est pas le cas.
- Principe du **moindre privilège** :
  - Un objet ne doit disposer que des droits qui lui sont strictement nécessaires pour réaliser les tâches qui lui sont dévolues.
- Utilisation de politique obligatoire :
  - La politique doit le moins possible dépendre des utilisateurs en tant que personne, mais reposer sur les rôles de la politique de sécurité du système d'information.

Méthodes d'authentification des personnes

- L'**authentification** = **vérification de l'identité** d'une entité.
- L'une des mesures les plus importantes de la sécurité:
  - Impossible d'assurer la confidentialité, l'intégrité, la non répudiation sans la garantie de l'identité de l'entité soumettant une requête.
- L'authentification devrait être assurée en continue (pas une fois pour toute à l'ouverture d'un objet ou en début de session)
- Un personne peut quitter son poste en le laissant ouvert :
  - procédure de déconnexion automatique
  - procédure d 'authentification périodique
- Une entité informatique peut être corrompue
  - une substitution peut avoir lieu (surtout en réseau, nécessité de protocoles de sécurité)
- L'authentification des personnes peut se faire par trois méthodes:
  - Ce que connaît l'utilisateur (Mot de passe),
  - Ce que détient l'utilisateur (carte...),
  - Ce qu'est l'utilisateur (Méthode biométrique)

Authentification par connaissance

- Le mot de passe, le code confidentiel
  - Technique la plus simple et la plus répandue
- Problèmes bien connus:
  - Si le mot de passe est simple il peut être trouvé par une attaque par dictionnaire
  - Si le mot de passe est compliqué l'utilisateur le note pour s'en souvenir !
  - La frappe du mot de passe peut être publique
  - Les mots de passe doivent être stockés (point sensible)
- Quelques parades:
  - **Ne jamais utiliser** son login, son nom, le nom de son chien, son n°de tél., un mot d'un dictionnaire...
  - Utiliser chiffres et lettres avec des caractères spéciaux au moins 6 à 7 caractères, mais trouver un moyen mémotechnique
  - **Obliger l'usager à changer** régulièrement de mot de passe.
  - **Surveiller les tentatives d'accès** illicite par comptage (les afficher).
  - **Prévenir l'usager des connexions** précédentes sur son compte en affichant la date et l'heure (par exemple du dernier accès).

MDP: l'exemple d'Unix/Linux crypt()

- On ne stocke pas les MDP dans /etc/passwd ou /etc/shadow
  - Utilisation d'une fonction à sens unique crypt()
    - L'inverse n'existe pas.
    - Propriété : Si p=p' alors crypt(p)=crypt(p')
    - Propriété : Si crypt(p')=crypt(p) alors p=p'
    - Combinatoire importante: attaque par force brute difficile
    - Alteration pat un paramètre (SALT) pour introduire des différences entre les entités
      - crypt(p, Salt)
- Habituellement, 30% des mots de passe sont devinables

Authentification par objet

- Un secret matérialisé physiquement
  - La clé traditionnelle
  - Une carte magnétique, à code barre, à puce
  - Un stick USB
  - Un porte-clefs générateur de clef temporaire
- Technique simple, répandue.
- Les problèmes :
  - la perte, le vol du support
  - la duplication (plus ou moins facile mais toujours possible)
  - Nécessite souvent l'intervention humaine

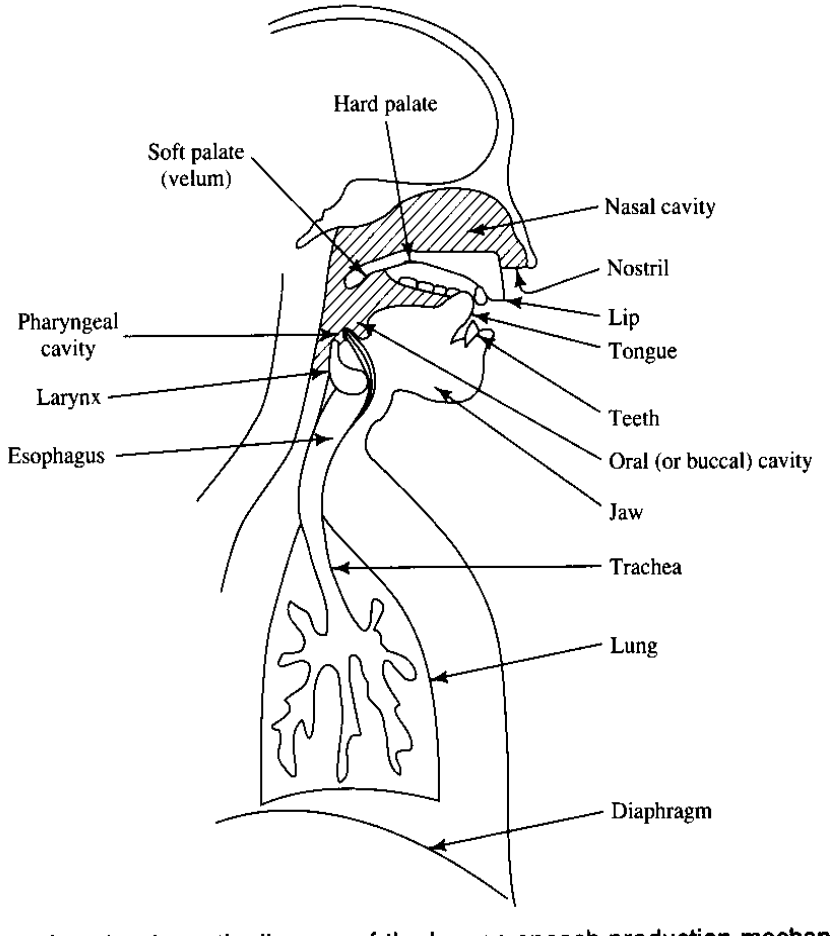
Authentification par l'utilisateur lui-même

- **Les méthodes bio métriques**
- Une solution en rapide développement
  - peut-être très efficace
  - souvent onéreuse,
  - peut-être difficile à accepter dans certains cas par l'utilisateur
- Nécessité d'études approfondies (analyse de la variabilité) du caractère utilisé
  - à l'intérieur du groupe humain des usagers autorisés
  - ou dans une population quelconque
- Incertitudes des techniques bio métriques
  - La variabilité intra-individuelle
  - La variabilité inter-individuelle
- Conduit à deux types d'erreurs possibles:
  - Le rejet à tort d'un individu autorisé
  - L'acceptation à tort d'une personne non autorisée.

Quelques techniques biométriques

- L'empreinte digitale
- la vascularisation de la rétine, de l'iris
- la voix
- la géométrie de la main, du visage
- dynamique de la signature
- dynamique de la frappe clavier
- empreinte génétique
- Thermographie faciale

BIOMETRIQUE – Reconnaissance de voix



Reconnaissance

- The speaker
- Par un code (dépendance au texte)
- Par le timbre (indépendance au texte)

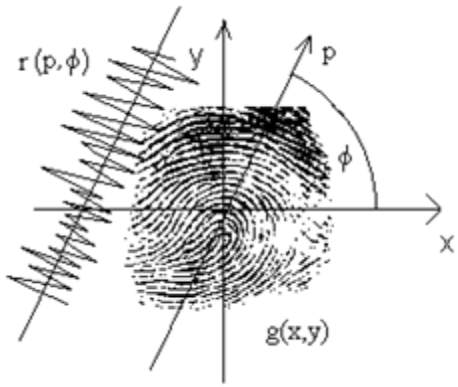
Facile à gérer

Enregistrement possible

Sensibilité aux bruits parasites

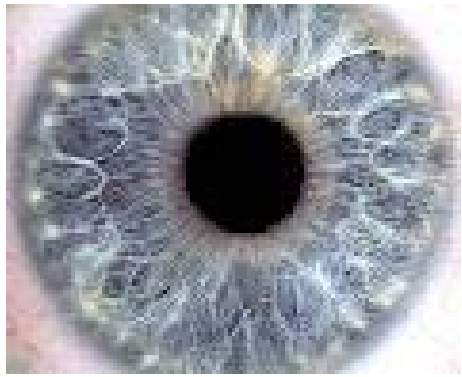
Importance des erreurs (toux)

BIOMETRIQUE – Empruntes digitales

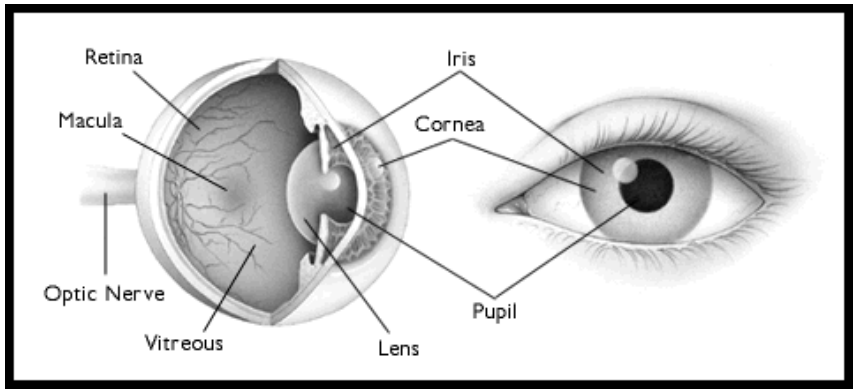


- Reconnaissance géométrique du doigt
  - Très connue et exploité
  - Petite taille des dispositifs
  - Faibles coûts des dispositifs
  - Analyse rapide, faible taux de rejet
  - Assez bien implanté
    - police
    - ordinateurs portables (IBM)
    - Bientôt (déjà) carte d'identité à puce
  - Problèmes
    - Doigts sales ou coupés
    - Besoin de la coopération de l'utilisateur

BIOMETRIQUE – Reconnaissance de l'iris

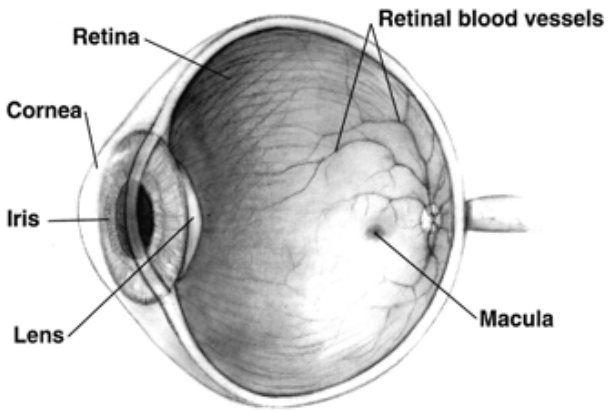


- Reconnaissance de la géométrie de l'iris
  - Grande quantité d'information
  - Reconnaît les vrais jumeaux
  - Très peu de rejet
  - Reconnaissance à distance



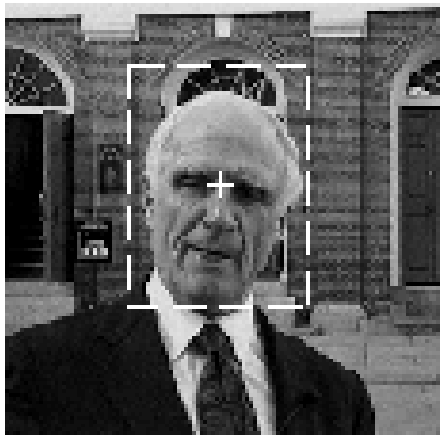
  - Problèmes
    - Non différenciation photo/humain
    - Non différenciation fausse iris/humain
    - Coût élevé

BIOMETRIQUE – Reconnaissance rétinienne



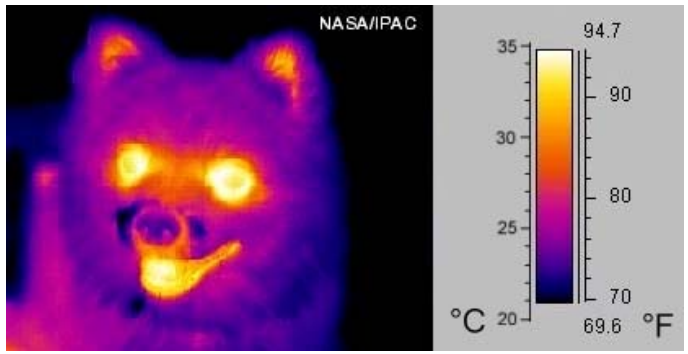
- Reconnaissance rétinienne
  - Vaisseaux sanguins
  - Peu de facteurs de variations (ie peu de maladies)
  - Meilleur taux de réussite
  - Problèmes:
    - Très cher
    - Intrusif donc peu populaire
      Qui veut coller son oeil dans l'objectif ?
    - Devient moins efficace avec le temps (âge de la personne)

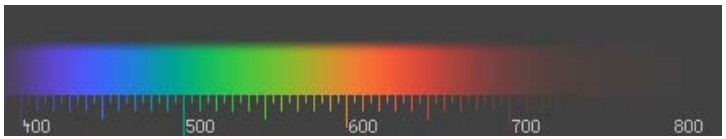
BIOMETRIQUE – Reconnaissance faciale



- Reconnaissance de la géométrie faciale
  - Distance entre les yeux, la bouche, le nez, ...
  - Facile à gérer
  - Identification à distance
  - Utile pour l'analyse de foule
  - Problème :
    - Identification impossible des vrais jumeaux
    - Sensible aux problèmes du visages (maladies, accidents)
    - Sensible aux lunettes, piercing
    - Maquillage, masque, perruques ➔ échec

BIOMETRIQUE - Thermographie





- Etude du spectre électromagnétique
  - IR
  - cartographie de la chaleur du visage
  - Bon taux de reconnaissance
  - Reconnaît les vrais jumeaux
  - Problèmes:
    - Coût très élevé
    - Expérimentale

MULTIMETRIQUES

- Biométriques seules sont insuffisantes
- Combinaison de différentes techniques
  - Biométriques
    - Reconnaissance par emprunte digitale
  - Méthodes traditionnelles
    - Mot de passe, Smart Cards avec des informations
- Forte taux de réussite en authentification
- Très résistant
  - Nécessite le cassage des n méthodes
- Coût raisonnable

Authentification – “One Time Passwords”

- Principes :
  - Utilisation de plusieurs mots de passe
  - Une fois utilisé, le mot de passe n’est plus utilisable (Utilisable une et une seule fois)
  - L’utilisateur a une clef privée (le secret) connue par le serveur qui permet de générer le mot de passe
  - L’utilisateur et le serveur on un calculateur logiciel ou matériel
- Génération du mot de passe
  - Le serveur génère un défi et une solution (le mot de passe)
  - Le serveur transmet le défi à l’utilisateur
  - L’utilisateur génère la propre solution à partir du défi
  - L’utilisateur la donne au serveur
  - Si les solutions correspondent, l’utilisateur est authentifié
- Peut être implanté sur des systèmes embarqués
  - Smart Cards, PDAs, portables, Keys Holders, etc...

Authentification – “One Time Passwords”

- Désavantages
  - L’utilisateur doit
    - posséder l’algorithme ou une appareil capable de le faire
    - un liste pré-imprimée de mots de passe
  - Dangereux en cas d’utilisation de la liste !
- Exemple
  - Challenge: 5241
    - Rechercher du MDP 5241 dans la liste
    - Rappel : La liste est générée à partir de la clef secrète
  - Réponse: CLAD ROY TOP BAD CAKE MATH
    - Quand la liste est expirée, on change de clef et on régénère une liste
- Difficile à gérer
- Outil les plus connus: S/Key (Neil Haller, “The S/KEY one-time password system”, 1994), OPIE, SecurID, CryptoCard

Authentification bi-factuelle

- Comme pour l'authentification multimétriques
- On combine deux méthodes :
  - "Quelque chose que vous connaissez" (secret)
    - ie un mot de passe
  - "Quelque chose que vous avez" (appareil)
- L'appareil stocke ou fait tourner un algorithme de génération de clefs uniques configurer pour l'utilisateur
- La plus part des «systèmes OPT » utilise la double identification