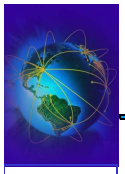
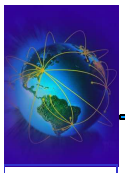


Legond-Aubry Fabrice
fabrice.legond-aubry@lip6.fr

Sécurité et
Administration des
Systèmes
Informatiques

Concepts de sécurité

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



Plan de cours

Concepts et Terminologie

Types d'attaques

Les politiques de sécurité

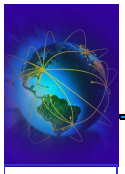
Les outils de la sécurité

Utilisation des CS symétriques

Utilisation des CS asymétriques

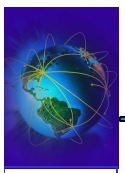
Les certificats

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.)



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

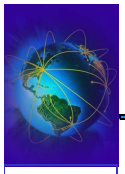
Authentification

Non répudiation

Intégrité

Confidentialité

Auditabilité

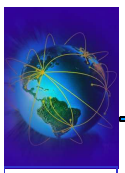


Propriété de sécurité: l'authentification

Concepts et Terminologie

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

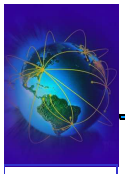


Propriété de sécurité: la non répudiation

Concepts et Terminologie

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

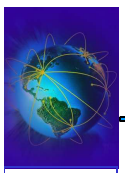
- Signature = Authentification+Non répudiation :
  - Seconde idée contenue dans la notion habituelle de signature
  - le signataire s'engage à honorer sa signature
  - engagement contractuel/juridique, on ne peut pas revenir en arrière
- Deux aspects spécifiques de la non répudiation dans les transactions 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)

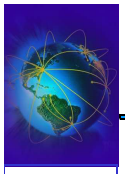
- Exemples :
  - Une modification intempestive (même très temporaire) est à interdire sur une écriture comptable validée
  - Le code binaire des programmes ne doit pas pouvoir être altéré
  - Les messages de l'ingénieur système doivent pouvoir être lus et non modifiés



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)

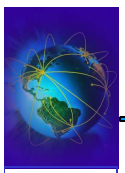
- Exemples :
  - Un mot de passe ne doit jamais pouvoir être lu par une autre personne que son possesseur
  - Un dossier médical ne doit pouvoir être consulté que par les malades et le personnel médical habilité
  - On ne doit pas pouvoir intercepter le contenu d'un courrier



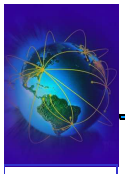
Propriété de sécurité: l'auditabilité

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.



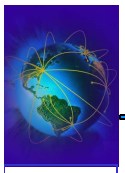
Plan de cours



Attaques sur l'authentification

Types d'attaques

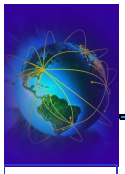
- Déguisement (Mascarade)
  - Pour rentrer dans un système, on essaye de piéger des usagers et de se faire passer pour quelqu'un d'autre (usurpation d'identité)
- Exemples:
  - simulation d'interface système sur écran,
  - simulation de terminal à carte bancaire



Attaques sur l'intégrité

Types d'attaques

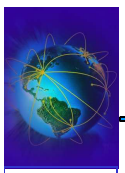
- **Intégrité des données : Modification de messages, de données**
  - Attribution par une entité non autorisée (usager , agent autorisé) d'avantages illicites
  - Comment ? En modifiant un fichier, un message …
  - Le plus souvent cette modification est réalisée par un programme et devient aussi une 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

Types d'attaques

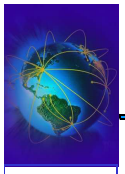
- **Intégrité des programmes**
  - Les modifications à caractère
    - *Frauduleux* : Pour s'attribuer par programme des avantages
    - *De sabotage* : Pour détruire avec plus ou moins de motivations des systèmes ou des données
  - Infections à caractère unique
    - Bombe logique ou cheval de Troie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



Attaques sur la confidentialité

Types d'attaques

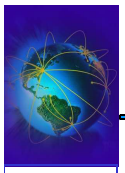
- Les attaques ayant pour but le vol d'informations
  - espionnage des transmissions de données
- Analyse de trafic : On observe le trafic de messages échangés pour en déduire des informations sur les décisions de quelqu'un.
  - Exemple: augmentation des transactions sur une place financière
- Inférence : 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

Types d'attaques

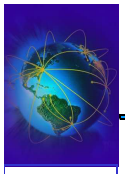
- **Attaque par violation de protocole**
  - Envoie de données non prévues (trames malformées, séquence non prévues)
- **Attaque par saturation**
  - Envoie de messages trop nombreux provoquant un écroulement des systèmes et réseaux
    - Exemple : « Distributed Denial Of Service »



Attaques sociales

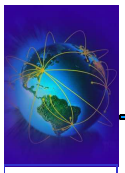
Types d'attaques

- 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

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
1. 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

3. 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 …)

3. Choix des contre mesures

Ex: Interdiction de boot sur des supports externes

Ex: Qui peut accéder physiquement aux machines ?

3. 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

3. 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

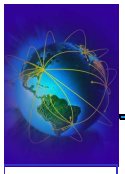
- Les moyens doivent être « complets »:
  - dans le cadre des hypothèses considérées, quoi qu'il arrive la politique est respectée
- Les moyens doivent être non contradictoires et raisonnablement contraignants
  - Ils ne doivent pas constituer un obstacle à la réalisation des fonctions opérationnelles 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**.

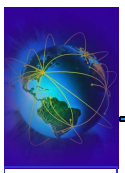
Dans le détail des politiques

- 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, puis ISO/CEI 27002.
- Il existe différents standards de sécurité
  - Défini par le ministère de la défense US
    - http://www.dtic.mil/whs/directives/corres/pdf/850001p.pdf (orange book)
    - http://csrc.nist.gov/publications/history/dod85.pdf (évaluation)

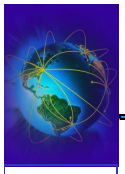


Dans le détail des politiques

- Il existe différents standards de sécurité (suite)
  - Défini par la NSA (Compartemented Mode Workstation)
    - http://www.nsa.gov/ia/mitigation_guidance/index.shtml
- Il existe différents standards de validation de politiques
  - ISO/IEC 15408-1:2009
  - http://niap.nist.gov/cc-scheme/index.html
  - http://www.iso.org/iso/home/store/catalogue_tc/catalogue_tc_browse.htm?commid=45306

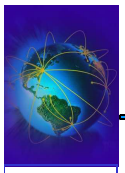


Plan de cours



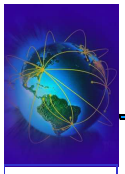
Quelques définitions

- **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 (ne) permet (pas) la cryptographie

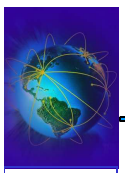
- Ce que ne peut pas faire la cryptographie
  - Empêcher l'effacement des données par un pirate
  - Protéger le programme de chiffrement et son exécution (traçage, debug,...)
  - Empêcher un décodage par hasard
  - Empêcher une attaque par force brute
  - Empêcher l'utilisation méthode inédite de décodage
  - 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.

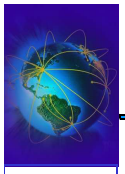
- Acteurs dans les futurs exemples :
  - Les acteurs : A(lice) et B(ob) – les gentils, Estelle (l'Espionne)
  - Bob échange des messages avec Alice
- un message M ∈ MESSAGES_A_ENVOYER.
- M est un message dit « en clair » (non chiffré).
- Estelle écoute la voie de communication pour connaître M.
- Bob, construit un texte chiffré C ∈ MESSAGES_CHIFFRES.
  - $C = E_k(M)$ ou $C = \{M\}^E_k$
- La fonction E_k dépend d'un paramètre k appelé clef (ou secret) 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

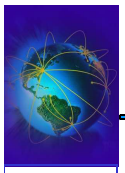
Le déchiffrement est l'opération inverse permettant de récupérer le texte en clair à partir du texte C chiffré.

- 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»
- $D_{k'}$ est donc une fonction inverse à gauche de E_k.
  - Note: il peut y avoir symétrie (D/crypter, E/décrypter)
  - Note: on ne suppose rien sur le lien entre k et k'
- 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.



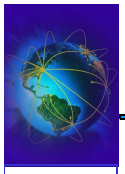
Efficacité du chiffrement

- L'efficacité général du cryptage dépend
  - De la confidentialité des « secrets » k, k'
  - De la difficulté à deviner les secrets ou a essayer toutes les possibilités (lié à la complexité des secrets)
  - De la difficulté de l'inversion de l'algorithme de (dé)chiffrement sans connaître la clé (*cassage*)
  - 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 n'offre pas d'alternative à l'essai de toutes les secrets possible de l'espace des secrets



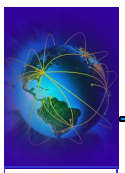
Un outil: Crypto-systèmes symétriques

- 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 ».
  - Certains « mauvais » CS symétriques sont utilisables !



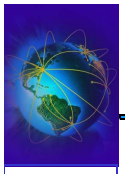
CS symétriques: exemples

- 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)



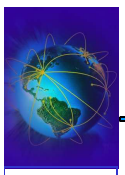
CS symétriques: l'ancêtre DES

- 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



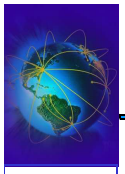
CS symétriques: Caractéristiques

- 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
    - Openssl i5 2,5Ghz : DES (64 Mb/s), aes256 (60Mb/s)
  - Une seule clef (donc partagée, « tout ou rien »)
  - Clef de taille relativement petite (~ 10^2 / 10^3 bits)



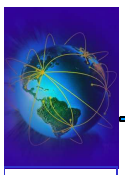
Un outil: Crypto-systèmes asymétriques

- 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'(k) à partir de k(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,



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).

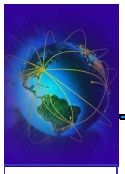
CS asymétrique RSA : Exemple

- **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**

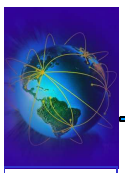
Un outil: Crypto-systèmes asymétriques

- Il existe RSA, DSA, ElGammal (ancien DSA), ECDSA
- Plus grosse clef RSA cassée
  - 768 bits, 232 décimales (décembre 2009)
  - Temps: 500 CPU (Intel Xeons 2.5Ghz) – Ans
  - 67Gb de RAM
- RSA 512 bits en 2012
  - Quelques douzaines de PC en quelques mois
- **Remarque (2012)** : Un utilisateur lamdba ne cassera jamais une clef RSA 1024 bits.
  - Une douzaine d'entité sur terre ont la puissance de calcul nécessaire pour avoir une petite chance de le faire.
  - Ils seront capables de casser une clef 1024bits dans un temps raisonnable d'ici 2020



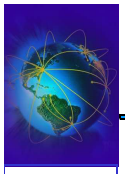
CS asymétrique RSA : performances

- **Utiliser des longueurs de clés de plus en plus importantes**
  - Valeurs utilisées : 512 bits, 640 bits
  - Non craqué en 2012 : 1024 bits (considéré comme sûr pour plusieurs années), 2048 bits (conseillé)
- **Utiliser des circuits intégrés de cryptage de plus en plus performants**
  - Actuellement une dizaine de circuits disponibles.
  - Vitesse de cryptage de base pour RSA 1024 bits
    Intel dual core i5 2,5Ghz: de 800Kb/s (private) et 1,6Mb/s (public)
- **Remarque 2** : Loi de Moore (empirique); La loi de Moore fixe un cycle de dix-huit mois pour les doublements de nombre de transistors.
- **Remarque 3** : Compte tenu de la complexité des traitements le CS Symétrique sont environ toujours 10 à 100 fois plus rapide que le CS asymétrique. Ceci est un ordre d'idée.



CS asymétrique RSA : Conseils

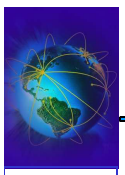
- **TOUT EST UNE QUESTION DE TEMPS**
- Ne jamais utiliser une valeur de n trop petite.
  - Actuellement un calcul en parallèle utilisant quelques milliers d'ordinateurs pendant quelques mois permet de factoriser des nombres d'une centaine de chiffres (768 bits, en 2009)
  - Utiliser des n=1024 ou 2048 bits selon que la protection recherchée est de plus ou moins de cinq ans.
- Ne pas chiffrer des blocs trop courts (les compléter toujours à n-1 bits), de façon à détruire toute structure syntaxique [padding, entropy]
- 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).

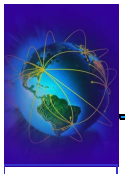
- Exemple:
  - Calcul modulo n (dans un anneau fini)
  - M^2 est facile à calculer modulo n (ou M^e)
  - *M* est difficile à calculer (log M)
  - CRC



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.

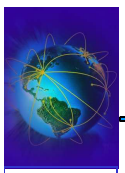
- L'intérêt d'une fonction de hachage est que M peut être arbitrairement grand alors que $\{M\}^H$ a une longueur donnée.
- Fonction **NON BIJECTIVE**
  - elle est destructrice (pas d'équivalence entre M et $\{M\}^H$)
  - Elle caractérise le bloc de données
  - Fonction a sens unique : difficulté pour 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



Les outils de la sécurité

Fonction de hachage sécurisé

- h(M) telle que f est une fonction de hachage par rapport à M
- h est à collision faible difficile: il est calculatoirement difficile de trouver M significatif tel que h(M)=K
  - Difficulté de trouver un bloc ayant un signature K
- h est à collision forte difficile: il est calculatoirement difficile de trouver M et M' tel que h(M)=h(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
  - Eviter MD5 (2009) et SHA1. Des pairs de fichiers à empruntes identiques ont été crées.
  - MD5 (2009) n'est plus sécurisé. SHA1 (2005) encore utilisé mais avec avertissement.
  - Non sécurisé : MD2, MD4, CRC32
  - Chiffre : 2012, l'attaque la plus efficace sur SHA-1 a un coût de 2,77M$ pour casser une seule valeur de hache.



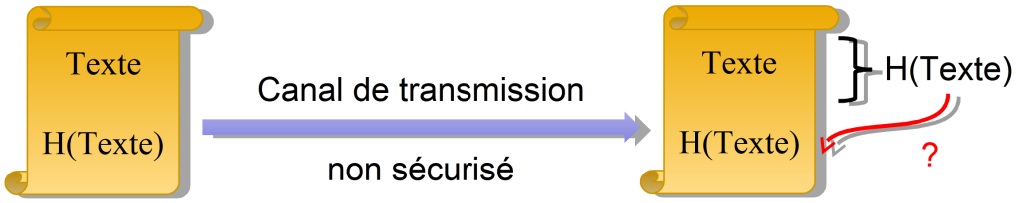
Utilisation des CS symétriques

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.**

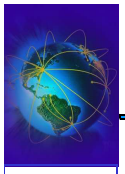
Fonction signature

- La signature numérique (signature électronique) est un mécanisme permettant de garantir l'intégrité d'un document électronique et d'en authentifier l'auteur.
  - analogie avec la signature manuscrite d'un document papier
  - En général : Non altération du document
  - Preuve d'origine du document / identité du créateur document signé
- En pratique :
  - Calcul d'une fonction de hachage sur le document (pour le caractériser)
  - Protection du résultat du calcul pour en éviter la modification
  - Lors de la réception du document
    - Vérification par comparaison de la nouvelle valeur calculée et de la valeur protégée stockée



Signature

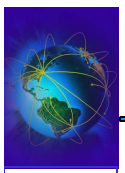
- Signature
  - Une signature manuscrite idéale est réputée posséder les propriétés suivantes:
  - La signature **ne peut-être imitée et authentifie** le signataire.
  - Elle prouve que le signataire a délibérément signé le document.
  - La signature appartient à un seul document (elle **n'est pas réutilisable**).
  - 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



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 »)
  - USA : limitation à 40 bits des clés des systèmes symétriques à l'exportation
- Mouvements sur Internet de démonstrations de craquages massivement parallèles pour pousser à l'autorisation de clés suffisamment grandes
- En France autorisation personnelle sans déclaration de clés sur 128 bits max (cryptage symétrique).
  - Loi n° 2004-575 (LCEN), Décret 2007-663 (cryptographie), Arrêté du 25 mai 2007 (cryptographie)
  - www.scssi.gouv.fr
- Pacte Ukusa / Echelon datant de 1948 sous contrôle de la NSA : USA + GB + Canada + Australie + Nouvelle-Zélande associés pour espionner les communications mondiales.

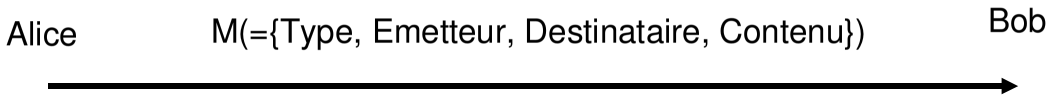
RENSEIGNEZ VOUS !!!



Plan de cours

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 :



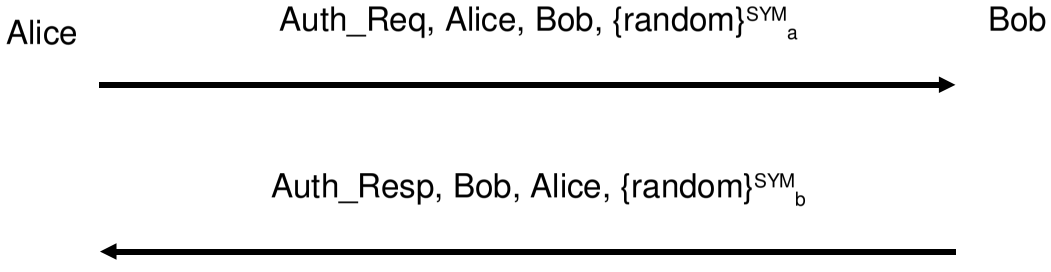
Notations

- 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)

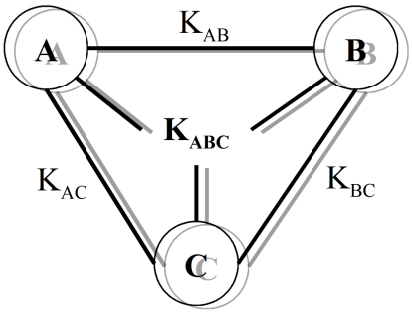
- Une première mauvaise solution
- Partage des clefs privées. Connaissances des participants :
  - Alice connait
    - sa propre clef A
    - Date de validité : Date début a / Date fin a
    - la clef B de Bob
  - Bob connait
    - sa propre clef B
    - Date de validité : Date début b / Date fin b
    - 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 « 1.5 »)



- Chaque canal à sa clef de cryptage
- Défi
  - Impossible d'utiliser un random identique cripter différemment
  - Ajouter « 1 » très simple défi
- PROBLEME : dissémination et de multiplications des clefs

Alice — Auth_Req, Alice, Bob, $\{random\}^{SYM}_{ab}$ → Bob

Auth_Resp, Bob, Alice, $\{random+1\}^{SYM}_{ab}$ ←

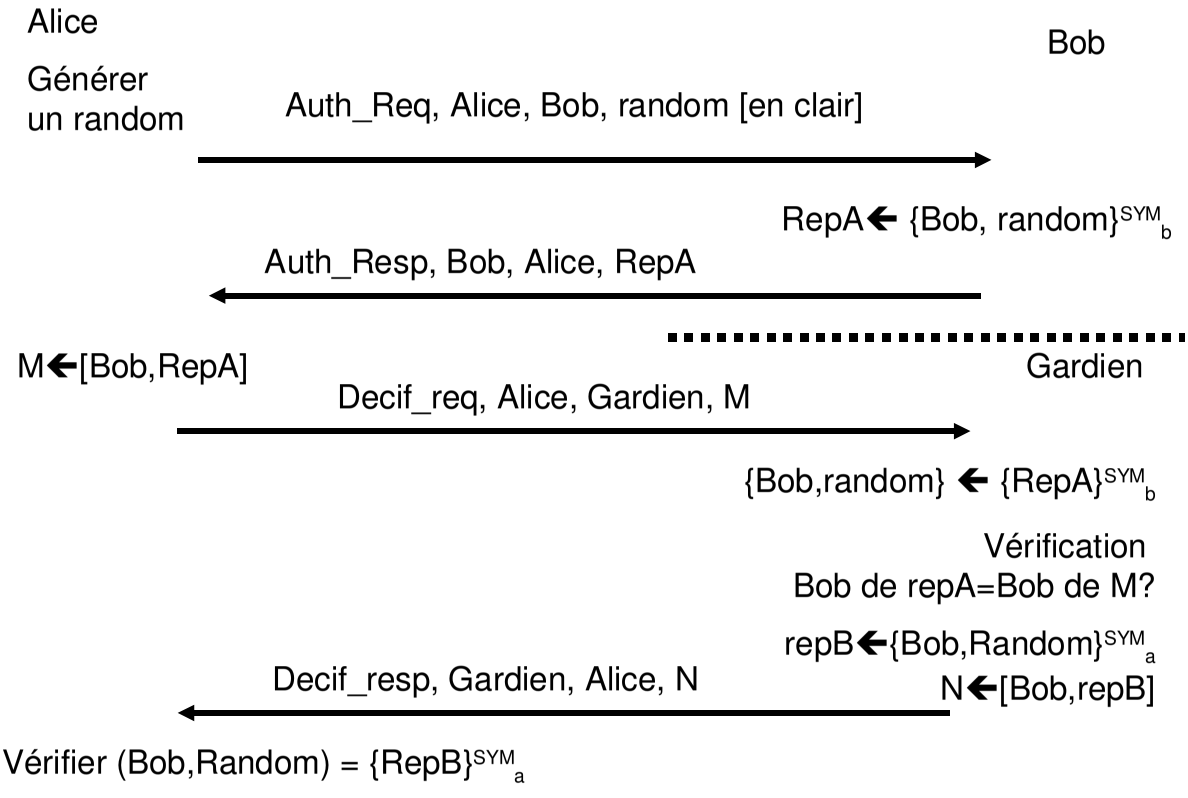
Authentification par un CS symétrique (solution 2)

- Utilisation d'un gardien des clefs
  - 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
  - Exemple dans quelques slides : kerberos [ie cerbere], le gardien des enfers

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

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**

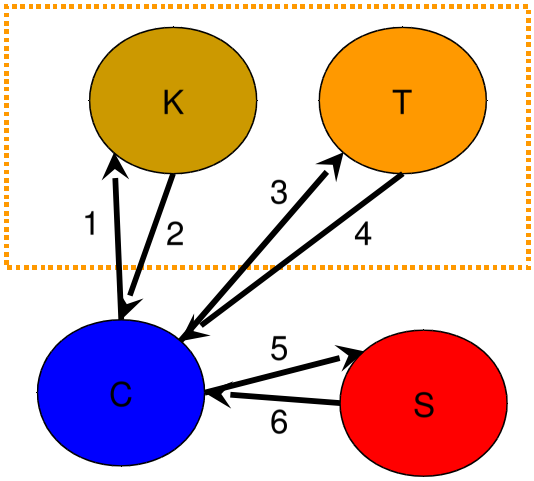
Authentification par un CS symétrique : Kerberos

- **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

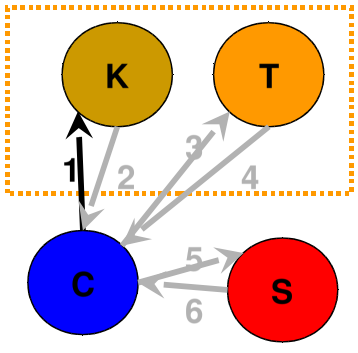
- **Architecture**

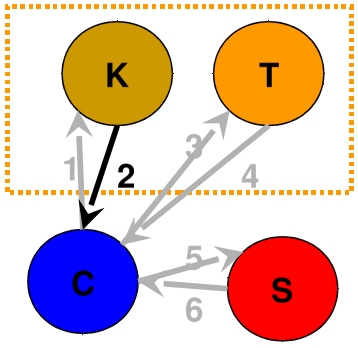
1. accès au serveur Kerberos K : authentification du client
2. Retour d'un ticket pour accéder au serveur de ticket
3. Accès au serveur de ticket T : contrôle d 'accès au serveur S
4. retour d 'un ticket pour accéder au serveur S
5. accès au serveur S



Authentification par un CS symétrique : Kerberos

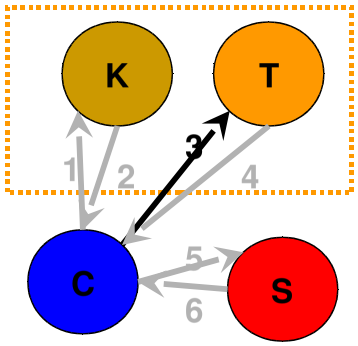
- (1) Demande par C d'un TGT (Ticket Granting Ticket) à K
  - Dé/Chiffré avec le mot de passe utilisateur
  - Message M1: tgt_req, C, K, {C, T}
  - K génère une clé de session KCT pour chiffrer le dialogue entre C et T
  - K génère un ticket TGTct pour autoriser l'accès du client C au serveur T
  - TGTct = {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)

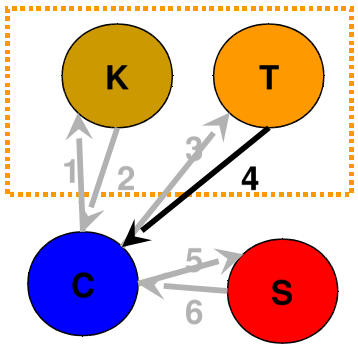




Authentification par un CS symétrique : Kerberos

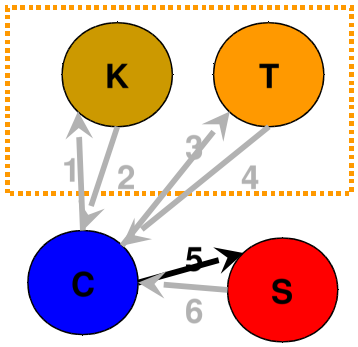
- (3) Demande d'un « *Service Ticket* » à T à l'instant **t1**
  - C construit un authentifieur : Act(t1) = {C, adr, t1}$^{SYM}_{KCT}$
  - message M3 : st_req, C, T, { Act(t1) ; TGTct ; S }
  - T déchiffre le ticket TGTct à l'aide de sa clé T, vérifie sa validité, et récupère ainsi la clé de session 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**}$^{SYM}_{KCT}$

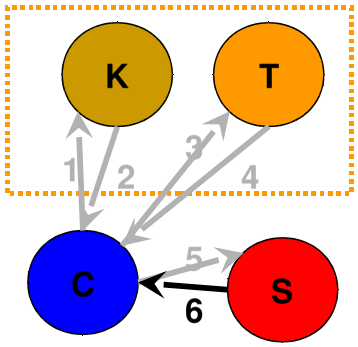




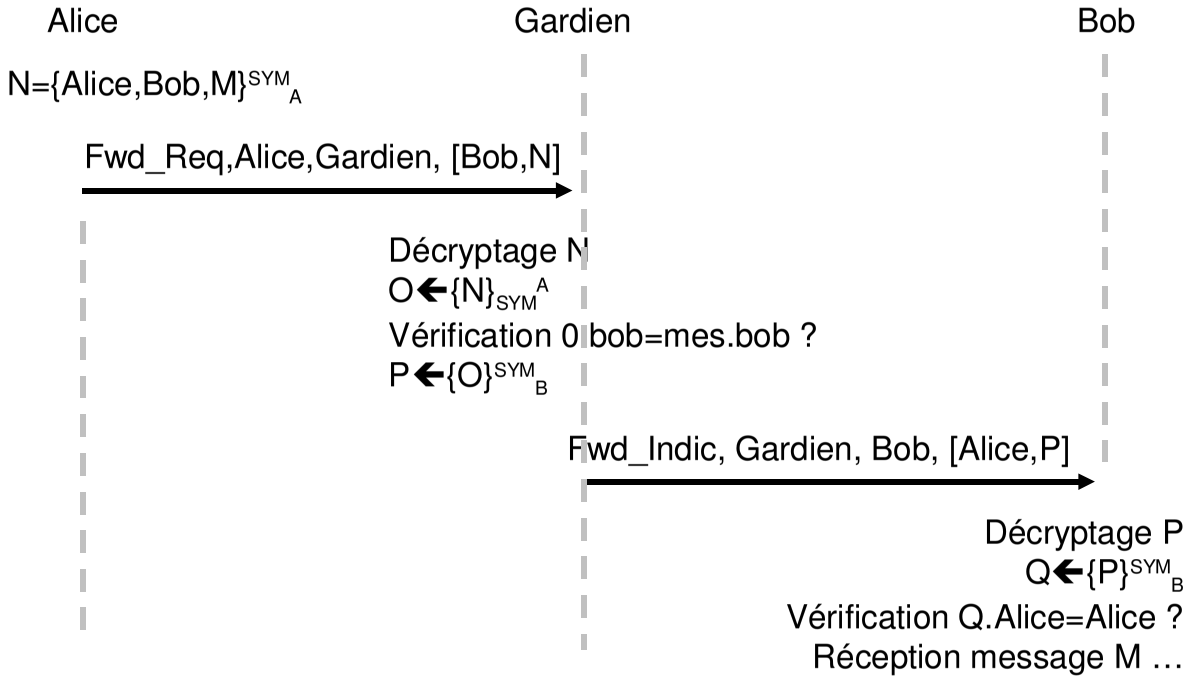
Authentification par un CS symétrique : Kerberos

- (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)



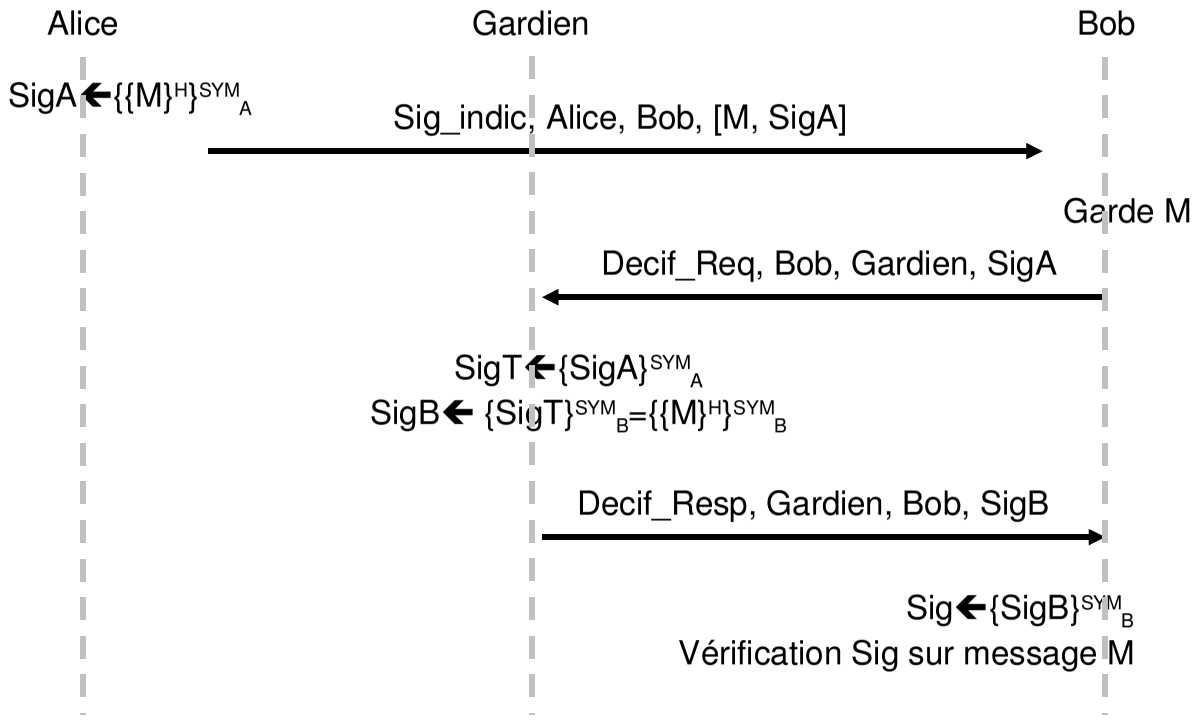


Confidentialité avec un CS symétrique



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

Intégrité et signature avec un CS symétrique



Plan de cours

Introduction

Concepts et Terminologie

Types d'attaques

Les politiques de sécurité

Les outils de la sécurité

Utilisation des CS symétriques

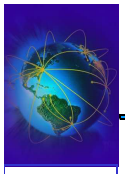
Utilisation des CS asymétriques

Les certificats

Authentification des personnes

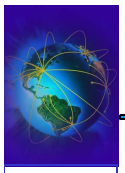
Notations

- 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

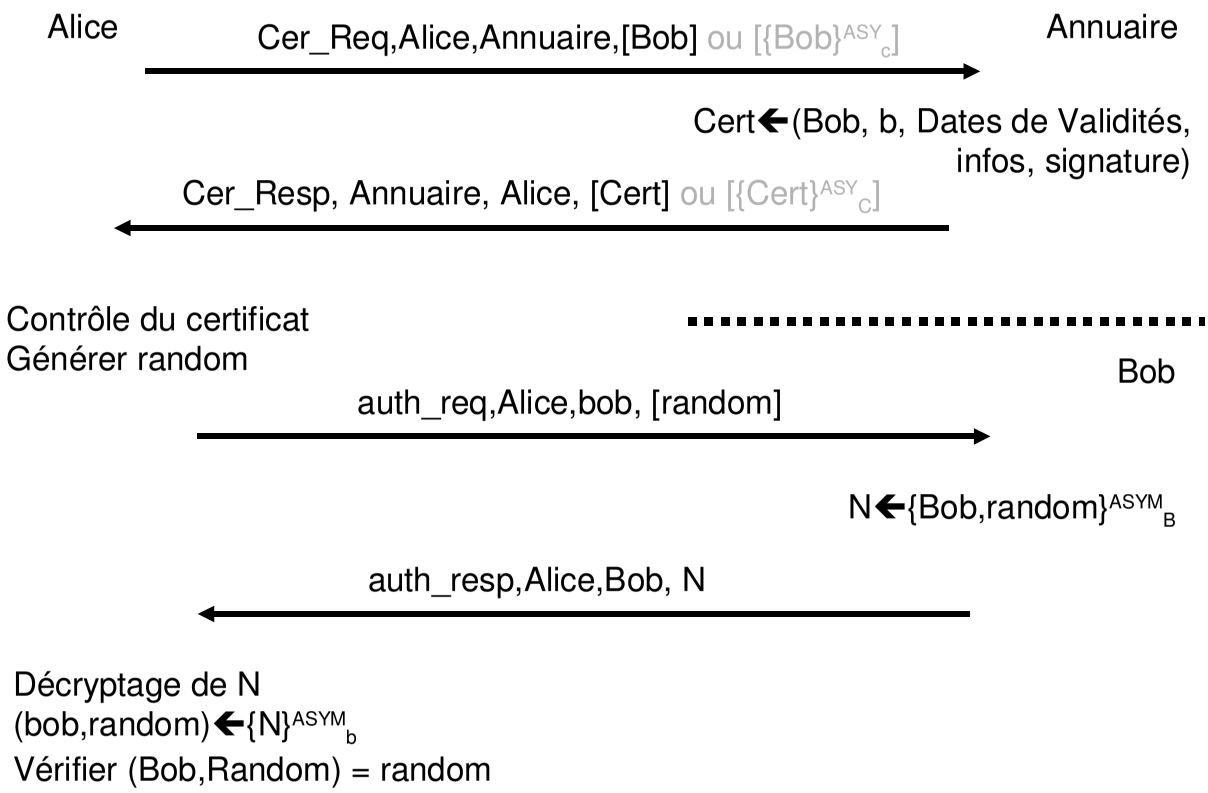
- Cryptage asymétrique
  - Si on crypte avec l'un, on décrypte avec l'autre
  - « clef » (minuscule) est la clef publique
  - « CLEF » (majuscule) est la clef privée
  - $\{M\}^{ASYM}_{clef}$ est le symétrique de $\{M\}^{ASYM}_{CLEF}$
- Fonctions de hachage et signature
  - $\{M\}^{H}_{meth}$ calculer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



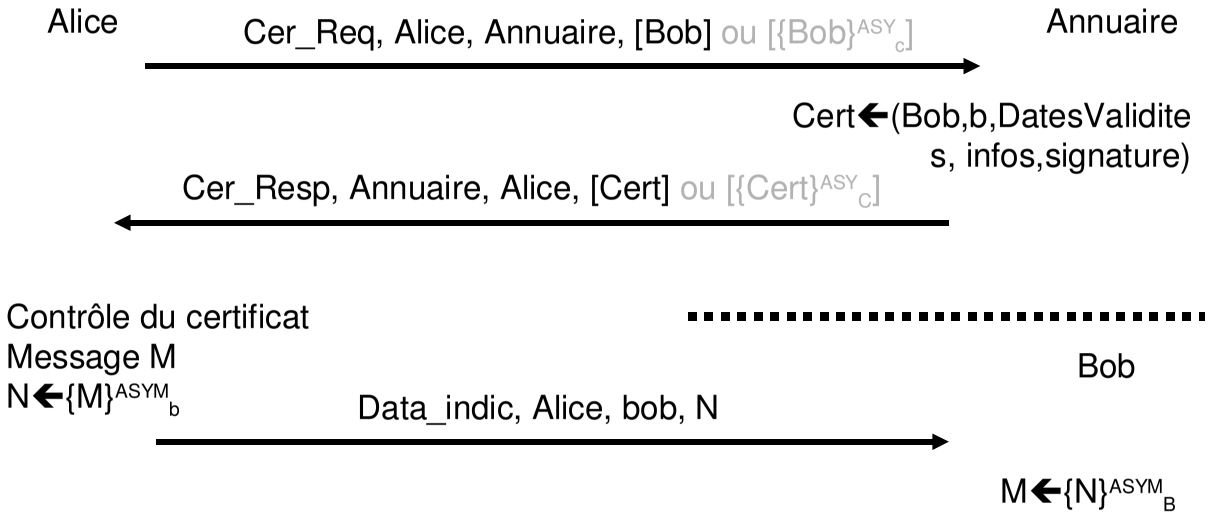
Authentification par un CS asymétrique

- 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)

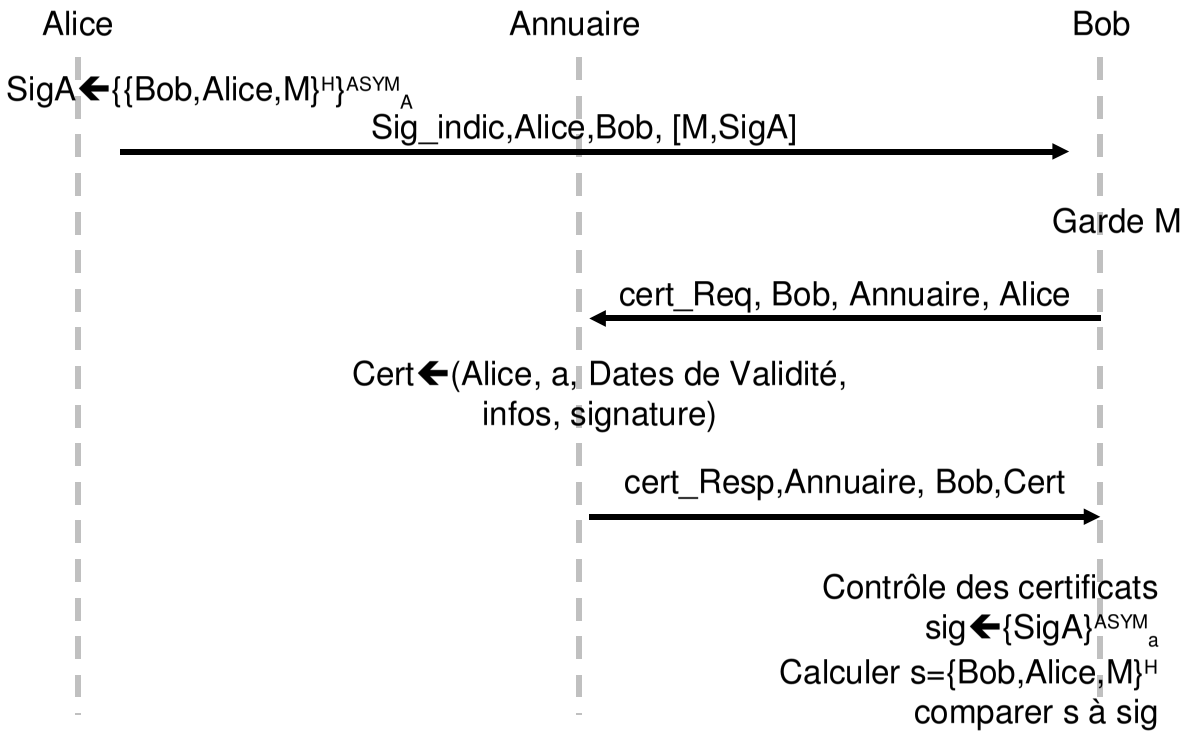


Confidentialité avec un CS asymétrique



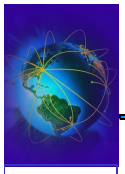
- Rappel: très peu utilisé car très lent !!
- On utilise le cryptage asymétrique pour échanger une clef de session
  - Principe de SSH

Intégrité et signature avec un CS asymétrique



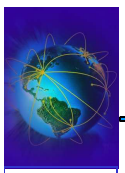
Intégrité de flots de messages

- 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



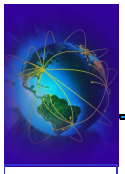
Considérations ad-hoc: Stockages des clefs

- 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

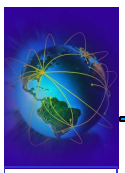
Les outils de la sécurité

Utilisation des CS symétriques

Utilisation des CS asymétriques

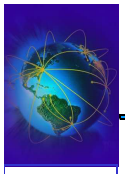
Les certificats

Authentification des personnes



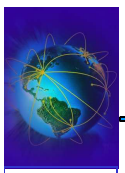
Certificats et cryptages asymétriques

- 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)



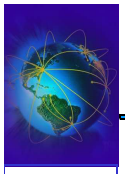
Les autorités de certifications (CA)

- 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)

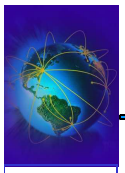
- Rôle :
  - Vérifie les demandes de certificats (Certificat Signing Request)
  - Génère les certificats et les publie
  - Génère les listes de certificats révoqués (Certificat Revocation List)
- Vérifie (par l'intermédiaire d'une autorité d'enregistrement) :
  - l'identité des demandeurs de certificats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



Les certificats

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)**

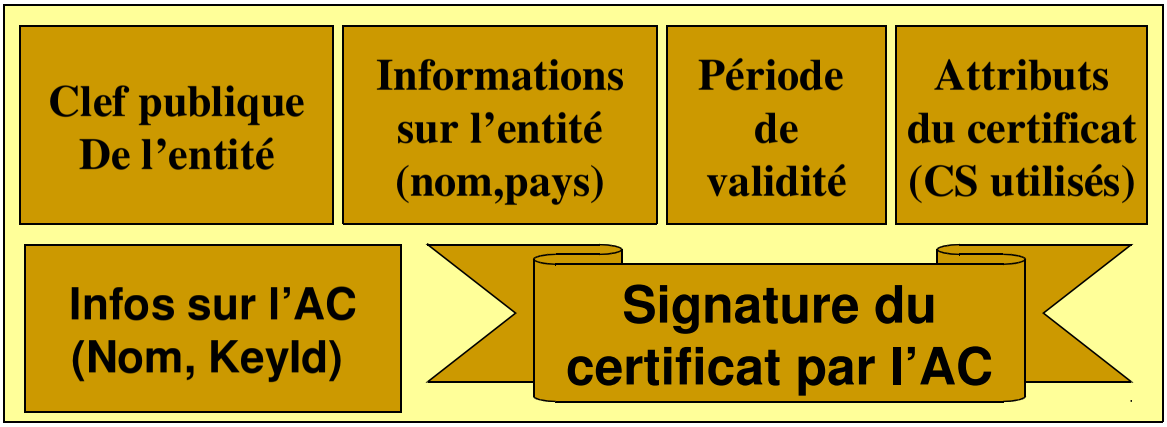


Utilisation des CS asymétriques

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 ←

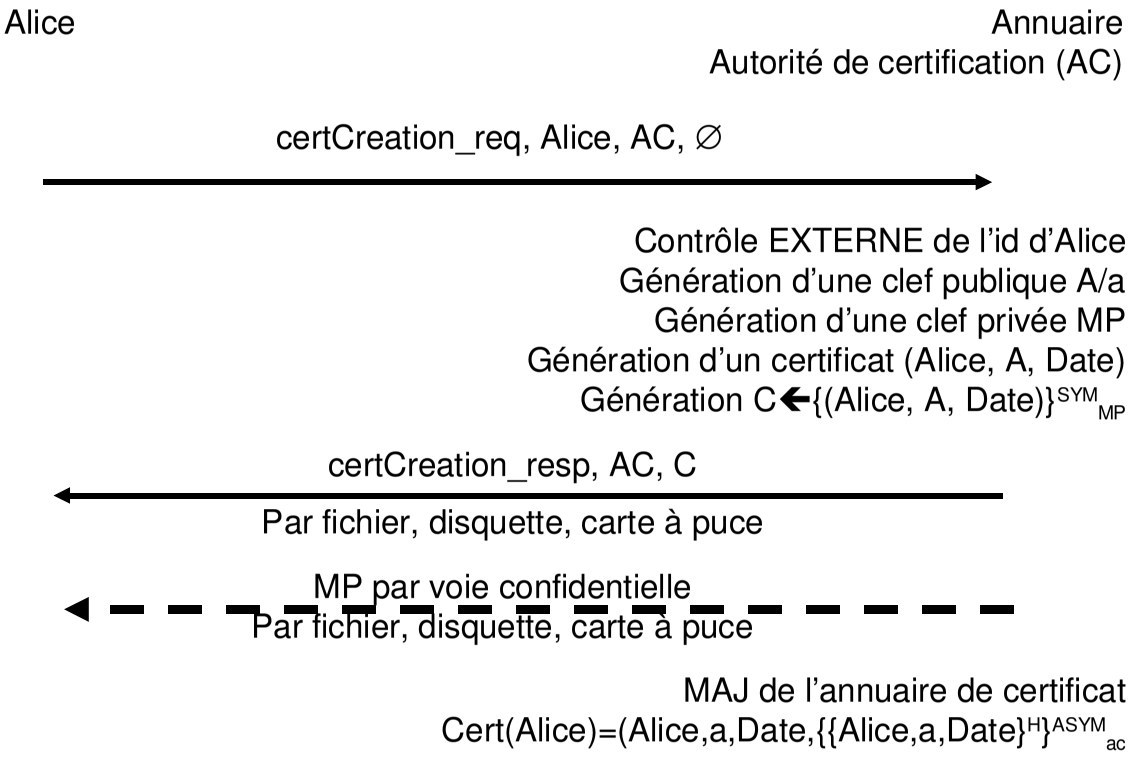
Génération d'une clef publique A/a
Génération d'une clef privée MP
Génération d'un certificat (Alice, A, Date)
Stockage (sûr) $\{(Alice, A, Date)\}^{SYM}_{MP}$
C← $\{(Alice, a, Date)\}^{ASYM}_{ac}$

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)

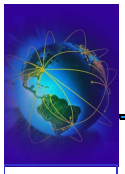


Les certificats

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

Les certificats



Plan de cours

Introduction

Concepts et Terminologie

Types d'attaques

Les politiques de sécurité

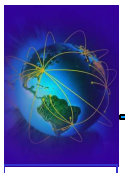
Les outils de la sécurité

Utilisation des CS symétriques

Utilisation des CS asymétriques

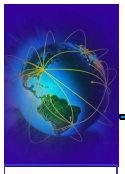
Les certificats

Authentification des personnes



Authentification: Rappels & généralités

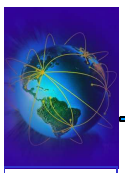
- Le contrôle d'accès est la base des mécanismes informatiques:
  - Il permet de spécifier la politique dans le domaine de l'informatique.
  - Il définit la façon dont le système contrôle ces droits.
  - Il devrait, en théorie, encapsuler toutes les autres techniques informatiques
  - Pour l'instant 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.



Authentification des personnes

Type d'authentification

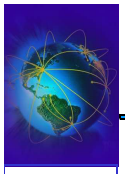
- L'authentification = vérification de l'identité d'une entité. (Voir def.)
- L'une des mesures les plus importantes de la sécurité:
  - Impossible d'assurer la confidentialité, l'intégrité, la non répudiation et l'auditabilité sans la garantie de l'identité de l'entité soumettant une requête.
- L'authentification peut-être
  - assurée en continue
  - ponctuelle. Par ex. à l'ouverture d'un objet ou en début de session
- Problème de l'authentification en continue
  - Couteux, contraignant
- Problème de l'authentification ponctuelle
  - plus faible qu'en continue
  - 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)



Authentification des personnes

Méthodes d'authentification des personnes

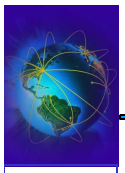
- L'authentification des personnes peut se faire par trois méthodes:
  - Ce que connaît l'utilisateur (secret)
    - Mot de passe
  - Ce que détient l'utilisateur (possession)
    - Carte, clef, liste, …
  - Ce qu'est l'utilisateur (être)
    - Méthode biométrique



Authentification par connaissance

Authentification des personnes

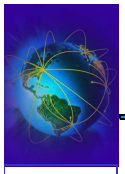
- Le mot de passe, le code confidentiel
  - Technique la plus simple et la plus répandue
- Problèmes bien connus:
  - Si le mot de passe est simple il peut être trouvé par une attaque par dictionnaire
  - Si le mot de passe est compliqué l'utilisateur le note pour s'en souvenir !
  - La frappe du mot 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: utilisation des fonctions de hachage

Authentification des personnes

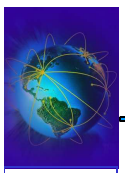
- On ne stocke pas les MDP en clair !
- Utilisation d'une fonction de hachage H à sens unique
  - Utilisation des propriétés de collisions faible/fort difficiles
    - Rappel : L'authentification est un défi !
    - Le système stocke H(password)
      → impossibilité de revenir au password (fonction à sens unique !)
    - Authentification :
      - L'utilisateur saisi son mot de passe p, le système calcul H(p)
      - Le système compare H(password) et H(p)
  - Utilisation de la collision faible difficile
    - Combinatoire importante: attaque par force brute difficile
    - Calculatoirement difficile de trouver p' tel que H(password)=H(p')
- Altération par un paramètre (SALT) pour introduire des différences entre les entités
  → éviter le pré-calcul de mot de passe simple (dictionnaire de haches)
- Habituellement, 5% à 15% des mots de passe sont devinables (parfois jusqu'à 30% suivant les personnes)



Authentification des personnes

Authentification par objet

- Rappel : L'authentification est un défi !
  - Le défi : posséder un 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 des personnes

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 (au cours du temps)
    - Stabilité, résistance
  - La variabilité inter-individuelle (différence entre individus)
    - Pouvoir discriminant

Authentification par l'utilisateur lui-même

- Conduit à deux types d'erreurs possibles:
  - Le rejet à tort d'un individu autorisé → False No-Match Rate
  - L'acceptation à tort d'une personne non autorisée. → False Match Rate
- Quelques autres problèmes soulevés :
  - Les éléments biométriques ne sont pas secrets
    - Ils peuvent se voler ; limites identification / authentification
  - Les éléments biométriques ne peuvent être révoqués / régénérés
    - Un individu change rarement de manière radicale mais il évolue !
  - Effet de bord de la biométrie : violation de la vie privée
  - Coercition des porteurs biométriques (personnes)
  - Un élément biométrique peut être corrompu / contaminé

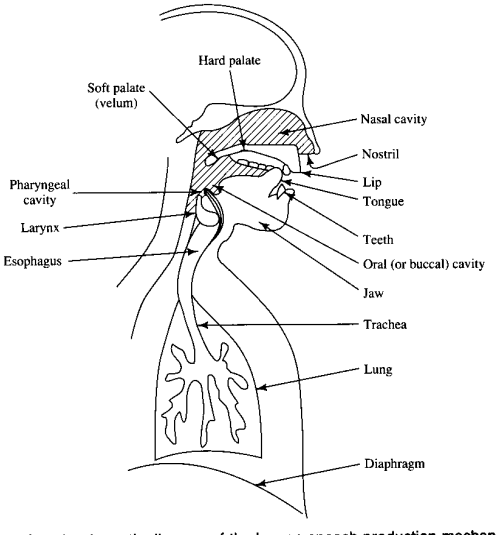
Authentification par l'utilisateur lui-même

- Critères de caractérisation des méthodes biométriques
  - Acceptation (Acceptability) : Les personnes acceptent-elles facilement de présenter cet élément biométrique ?
  - Permanence (Permanence) : Cet élément varie-t-il beaucoup au cours du temps ? Est-il modifiable ?
  - Universalité (Universality) : Toutes les personnes possèdent-elle cet élément biométrique ?
  - Quantifiabilité (Collectability) : Cet élément est-il aisément quantifiable / descriptible / accessible ?
  - Unicité (Uniqueness) : Plusieurs personnes peuvent-elles être confondues avec cet élément biométrique ?
  - Efficacité (Performance) : Cet élément est-il facile / rapide / peu couteux à collecter ? Est-il un bon discriminant ?
  - Incorruptibilité (Circumvention) : L'élément biométrique est-il falsifiable / copiable / corruptible ?

Quelques techniques biométriques

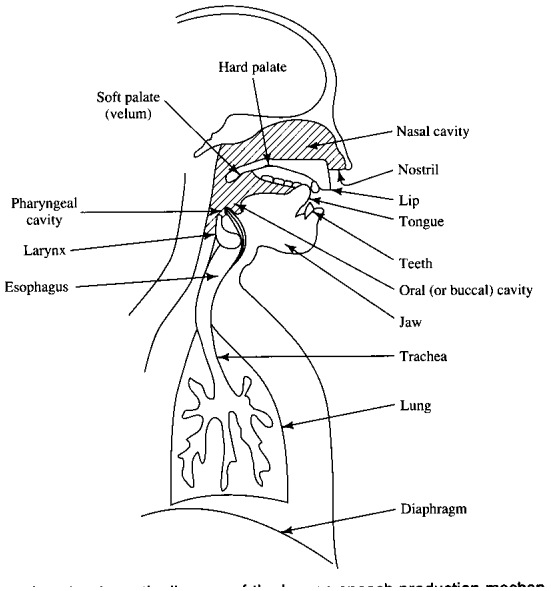
- L'empreinte digitale
- la vascularisation de la rétine
- l'iris
- la voix
- la géométrie de la main, du visage
- dynamique de la signature
- dynamique de la frappe clavier
- empreinte génétique
- Thermographie faciale

BIOMETRIQUE – Reconnaissance de voix



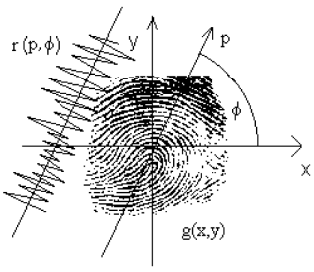
- Reconnaissance
  - Analyse de comportement
  - Par un code (dépendance au texte)
  - Par le timbre (indépendance au texte)
- Facile à gérer
- Vole aisé
- Sensibilité aux bruits parasites (environnement)
- Importance des erreurs, faible qualité, grande variabilité

BIOMETRIQUE – Reconnaissance de voix



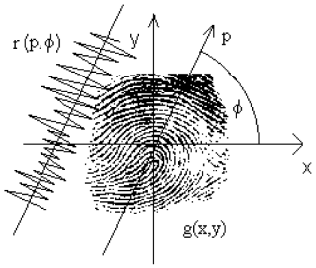
Emprunte vocale	Qualité
Universalité (Universility)	Moyenne
Unicité (Uniqueness)	Faible
Durabilité (Permanence)	Faible
Quantifiabilité (Collectability)	Moyenne
Efficacité (Performance)	Faible
Acceptation (Acceptability)	Haute
Incorruptibilité (Circumvention)	Faible

BIOMETRIQUE – Empruntes digitales



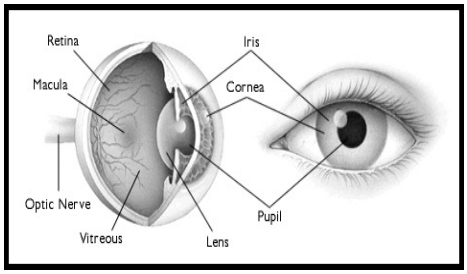
- Reconnaissance géométrique des empruntes des doigts
  - Très connue et exploité
    - depuis des milliers d'année (chinois)
  - Petite taille et faibles coûts des dispositifs
  - Analyse rapide, faible taux de rejet
    - On peut distinguer des jumeaux
  - Assez (trop) bien implantées
    - Mais associées à la criminalité (police)
    - Collecte importante par divers groupes (états)
    - ordinateurs portables (IBM)
    - Bientôt (déjà) carte d'identité à puce
  - Problèmes : Une petite partie des personnes n'ont pas d'empruntes exploitables
    - Doigts sales ou coupés, Génétique, Age, Poids

BIOMETRIQUE – Empruntes digitales



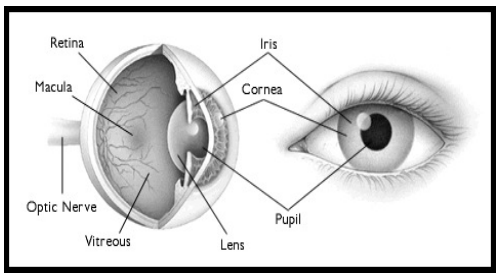
Empruntes digitales	Qualité
Universalité (Universility)	Moyenne
Unicité (Uniqueness)	Haute
Durabilité (Permanence)	Haute
Quantifiabilité (Collectability)	Moyenne
Efficacité (Performance)	Haute
Acceptation (Acceptability)	Moyenne
Incorruptibilité (Circumvention)	Haute

BIOMETRIQUE – Reconnaissance de l'iris



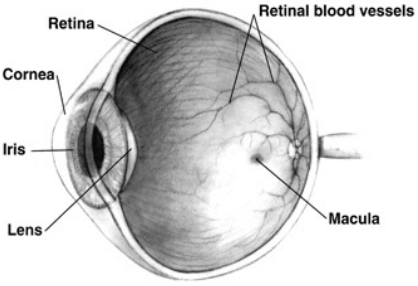
- Reconnaissance de la géométrie de l'iris
  - Anneau de l'œil entre la pupille et le blanc de l'œil
  - Stable à partir de 2 ans
  - Grande quantité d'information, informations distinctives
  - Reconnaît les vrais jumeaux
  - Reconnaissance à distance
  - Très peu de possibilité de modification ou contrefaçon
    - Détection des lentilles
  - Problèmes
    - Coût élevé
    - Accueil du public ? Trop récent

BIOMETRIQUE – Reconnaissance de l'iris



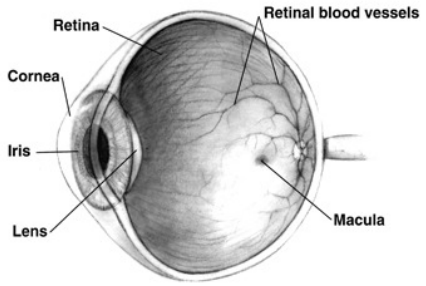
Iris	Qualité
Universalité (Universility)	Haute
Unicité (Uniqueness)	Haute
Durabilité (Permanence)	Haute
Quantifiabilité (Collectability)	Moyenne
Efficacité (Performance)	Haute
Acceptation (Acceptability)	Basse
Incorruptibilité (Circumvention)	Haute

BIOMETRIQUE – Reconnaissance rétinienne



- Reconnaissance rétinienne
  - Vaisseaux sanguins du fond de l'oeil
  - Peu de facteurs de variations (ie peu de maladies)
  - Très bons taux de réussite
  - Utiliser dans les prisons
  - Problèmes:
    - Très cher (toujours en 2012)
    - Intrusif donc peu populaire
      Qui veut coller son oeil dans l'objectif (lumière ou IR) ?
    - Devient moins efficace avec le temps (âge de la personne)

BIOMETRIQUE – Reconnaissance rétinienne



Reconnaissance rétinienne	Qualité
Universalité (Universility)	Haute
Unicité (Uniqueness)	Haute
Durabilité (Permanence)	Moyenne
Quantifiabilité (Collectability)	Basse
Efficacité (Performance)	Haute
Acceptation (Acceptability)	Basse
Incorruptibilité (Circumvention)	Haute

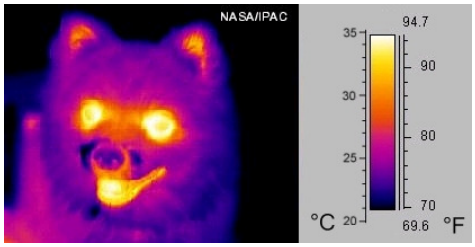
BIOMETRIQUE – Reconnaissance faciale

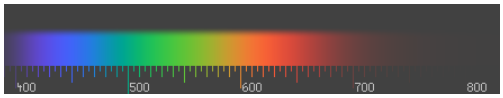
- Reconnaissance de la géométrie faciale
  - Le plus courant des éléments biométriques
  - Domaine de recherche actif
  - Basé sur la distance entre les yeux, la bouche, le nez, les sourcils, les lèvres, …
  - Facile à gérer, identification à distance
  - Utile pour l'analyse de foule mais faible qualité et facilement corruptible
  - Perturbations de l'environnement (position des prises de vues, …)
  - Problèmes :
    - Identification impossible des vrais jumeaux
    - Sensible aux problèmes du visages (maladies, accidents)
    - Sensible aux lunettes, piercing
    - Maquillage, masque, perruques ➔ échec

BIOMETRIQUE – Reconnaissance faciale

Reconnaissance faciale	Qualité
Universalité (Universility)	Haute
Unicité (Uniqueness)	Basse
Durabilité (Permanence)	Moyenne
Quantifiabilité (Collectability)	Haute
Efficacité (Performance)	Basse
Acceptation (Acceptability)	Haute
Incorruptibilité (Circumvention)	Basse

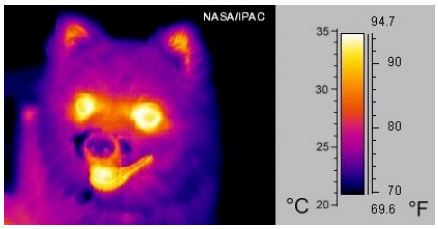
BIOMETRIQUE - Thermographie

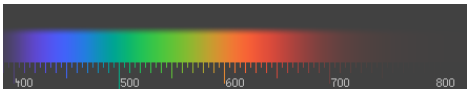




- Etude du spectre électromagnétique (IR)
  - cartographie de la chaleur du visage (vaisseaux sanguins)
  - Unique à chaque individu et non modifiable (volontairement ou non)
  - Bon taux de reconnaissance, non intrusif (sans contact)
  - Pas besoin du lumière
  - Reconnaît les vrais jumeaux
  - Problèmes:
    - Expérimentale
    - Sensible à l'état d'esprit (colère, …)

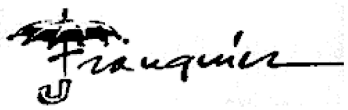
BIOMETRIQUE - Thermographie





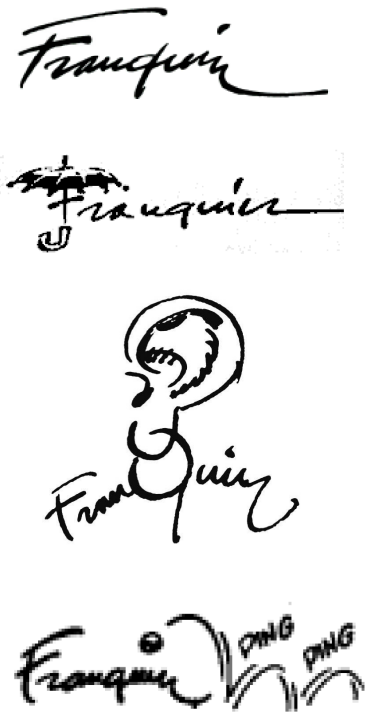
Thermographie	Qualité
Universalité (Universility)	Haute
Unicité (Uniqueness)	Haute
Durabilité (Permanence)	Basse
Quantifiabilité (Collectability)	Haute
Efficacité (Performance)	Moyenne
Acceptation (Acceptability)	Haute
Incorruptibilité (Circumvention)	Haute

BIOMETRIQUE - Signature



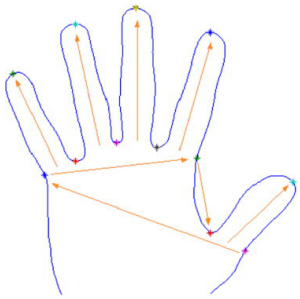
- Etude du dynamique de la signature
  - 2 approches : statique ou statique + dynamique
  - Etude sur la dynamique donc peu adaptée au traitement au traitement massif
  - Unique mais variabilité intra importante
  - Largement acceptée
- Problèmes:
  - Peu fiable en statique
  - Couteux en dynamique

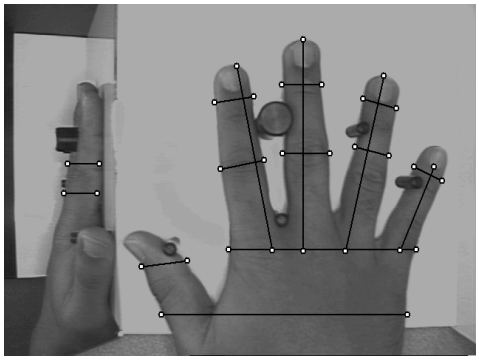
BIOMETRIQUE - Signature



Signature	Qualité
Universalité (Universility)	Basse
Unicité (Uniqueness)	Basse
Durabilité (Permanence)	Basse
Quantifiabilité (Collectability)	Haute
Efficacité (Performance)	Basse
Acceptation (Acceptability)	Haute
Incorruptibilité (Circumvention)	Basse

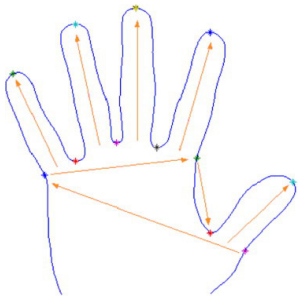
BIOMETRIQUE – Géométrie de la main

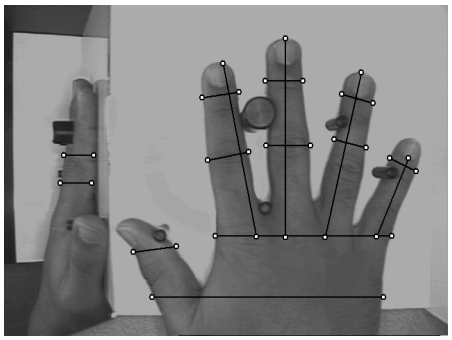




- Etude de la forme de la main
  - Longueur / Largeur de doigts, forme
  - Assez courant, simple, peu couteux
  - Peu affecté par problèmes de peau
  - Peu discriminatoire
  - Variabilité intra importante (enfance, …)
  - Gros capteur
  - Peut être combiné à la lecture des empruntes

BIOMETRIQUE - Géométrie de la main

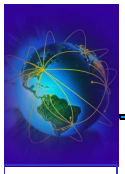




Main	Qualité
Universalité (Universility)	Moyenne
Unicité (Uniqueness)	Moyenne
Durabilité (Permanence)	Moyenne
Quantifiabilité (Collectability)	Haute
Efficacité (Performance)	Moyenne
Acceptation (Acceptability)	Moyenne
Incorruptibilité (Circumvention)	Moyenne

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 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