



LA SÉCURITÉ

G Florin, S Natkin

Introduction

Définition de base

La sécurité informatique c'est l'ensemble des moyens mis en oeuvre pour minimiser la vulnérabilité d'un système contre des menaces accidentelles ou intentionnelles.

Compléments de définition

1) Différences entre accidents et malveillances

En anglais : deux termes différents

a) Sécurité = "Safety"

Protection de systèmes informatiques contre les accidents dus à l'environnement, les défauts du système

Domaine d'élection : les systèmes informatiques contrôlant des procédés temps réels et mettant en danger des vies humaines (transports, énergie,..)

b) Sécurité = "Security"

Protection des systèmes informatiques contre des actions malveillantes intentionnelles.

**Domaine d'élection : les systèmes
informatiques réalisant des traitements
sensibles ou comprenant des données
sensibles**

2) Minimisation de risques

Différences entre une approche financière et une approche d'intolérance au risque.

Informatique d'entreprise classique :

Pour tout risque mise en balance du coût du risque et du coût de sa protection.

Informatique industrielle dite sécuritaire

Classification des pannes

Pannes catastrophiques vs pannes non catastrophiques

=> La panne catastrophique ne "devrait" pas se produire

- Techniques très sévères de validation/certification

- Mise en route d'un système si et seulement si une "confiance" très élevée lui est accordée.

3) Obligations légales en France

3.1 Validité juridique d'opérations informatiques

Certaines transactions informatiques entraînent des obligations légales de responsabilité => Elles sont considérées comme valides juridiquement par la loi ou la jurisprudence.

Ex:- ordres de virement informatique (par exemple deux fois le même ordre de paiement doit-être honoré)

- commandes télexées

3.2 Loi informatique et liberté

La Loi 78_17 du 6/1/1978

Définit la constitution et le rôle de la CNIL

Commission Nationale Informatique et Liberté.

Une entreprise ou une administration qui traite des fichiers administratifs nominatifs est responsable relativement à la non divulgation des informations qu'elle gère.

- . **Nécessité de formalités préalables à la mise en oeuvre des traitements automatisés pour**
- . **Collecte, enregistrement et conservation des informations nominatives**
- . **Exercice du droit d'accès**
- . **Dispositions pénales de non respect**

3.3 Loi no 85-660 du 3/7/1985

Décrit les règles relatives aux contrefaçons et au droit d'auteur

Par exemple la copie (autre que pour sauvegarde) est punissable de trois mois à deux ans de prison , d'une amende de 6000 à 12000 Francs.

3.4 Loi no 88-19 du 5/1/1988

Loi relative à la fraude informatique

Sont passibles de sanctions pénales pouvant atteindre 5 ans de prison, une amende de 2 millions les faits suivants:

- . **accès frauduleux** aux données.
- . **l'introduction** de données.
- . **l'entrave** au fonctionnement du système.
- . **la falsification** de documents informatisés.

Orientation du cours

Position en général des problèmes
(accident, malveillance)

Traitement des problèmes au sens des actions malveillantes

Présentation essentiellement de techniques informatiques

Utilisables dans les réseaux : les réseaux sont considérés en informatique comme le danger essentiel du point de vue de la sécurité

Techniques de cryptographie

PLAN DU COURS

I Aspects généraux de la sécurité

I.1 Les menaces

I.2 La politique de sécurité

II Les techniques informatiques

III Les protocoles de sécurité

IV La cryptographie

I**Aspects généraux de la
sécurité informatique**

I.1 Les menaces

L'ensemble des actions de l'environnement d'un système pouvant entraîner des pertes financières.

I.1.1) Menaces relevant de problèmes non spécifiques à l'informatique (hors du domaine de ce cours)

Risques matériels accidentels

Techniques de protection assez bien maîtrisées

Incendie , explosion
Inondation, tempête
Foudre

Vol et sabotage de matériels

Vol d'équipements matériels
Destruction d'équipements
Destruction de supports de sauvegarde

Autres risques

Tout ce qui peut entraîner des pertes financières dans une société (pertes plutôt

associées à l'organisation, à la gestion des personnels)

- Départ de personnels stratégiques
- Grèves
-

I.1.2) Les pannes et les erreurs (non intentionnelles)

Pannes/dysfonctionnements du matériel.

Pannes/dysfonctionnements du logiciel de base.

Erreurs d'exploitation.

oubli de sauvegarde
écrasement de fichiers

Erreurs de manipulation des informations.

erreur de saisie
erreur de transmission
erreur d'utilisation

Erreurs de conception des applications.

Erreurs d'implantation.

I.1.3) Les menaces intentionnelles

L'ensemble des actions malveillantes (qui constituent la plus grosse partie du risque).

Qui devraient être l'objet principal des mesures de protection.

Menaces passives

Détournement des données
(l'écoute, les indiscretions)

Exemples: espionnage industriel
espionnage commercial
violations déontologiques

Détournement des logiciels

Exemple: copies illicites

Menaces actives

Modifications des informations

Exemple : La fraude financière
informatique

Le sabotage des informations
(logique)

Modification des logiciels

Exemples: Bombes logiques, virus, ver

I.1.4) Pourcentages des différentes causes de pertes

- Actions malveillantes (en croissance) 61%
- Risques accidentels 24%
- Pannes et erreurs 12%
- Autres 3%

Explication de l'importance des actions malveillantes

- Développement de l'informatique
- Complexité croissante => plus grande vulnérabilité
- Dans une ambiance de non sensibilisation aux problèmes de sécurité

I.2 Exemples de menaces malveillantes à caractère informatique

I.2.1 Déguisement

Pour rentrer dans un système on essaye de piéger des usagers et de se faire prendre pour quelqu'un d'autre:

Exemple: simulation d'interface système sur écran
simulation de terminal à carte bancaire

I.2.2 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 créditement d'un compte bancaire.

I.2.3 Analyse de trafic

On observe le trafic de messages échangés pour en déduire des informations sur les décisions de quelqu'un.

Exemples:

Bourse : augmentation des transactions sur une place financière.

Militaire : le début de concentration entraîne un accroissement de trafic important.

I.2.4 Inférence

On obtient des informations confidentielles non divulguables à partir d'un faisceau de questions autorisées (et d'un raisonnement visant à faire ressortir l'information).

Exemple:

- Soit le fichier d'un hôpital la loi informatique et liberté interdit la divulgation d'informations personnelles (sur les maladies).

mais autorise des opérations statistiques (améliorer les connaissances épidémiologiques)

=> pas de possibilité de sélection sur le nom, le numéro de sec, l'adresse, ..etc. mais questions à caractère statistiques autorisées.

Pour obtenir des informations confidentielles poser des questions à caractère statistique comportant un faisceau de contraintes permettant en fait de filtrer une seule personne.

=> question sur les effectifs

**sexe = masculin, age = 30, arrêt
maladie, ...**

I.2.5 Répudiation (déli de service)

Un usager d'un service (informatique) affirme n'avoir pas :

émis un ordre qui le gêne a posteriori (commande, virement,)
reçu un ordre (idem)

I.2.6 Modification de messages, de données

Une personne non autorisée, un usager ou même un agent autorisé s'attribuent des avantages illicites en modifiant un fichier, un message (le plus souvent cette modification est réalisée par programme et entre dans la catégorie suivante)

I.2.7 Modification des programmes

I.2.7.1 Les modifications à caractère frauduleuses

Pour s'attribuer par programme des avantages.

Exemple: virement des centimes sur un compte

I.2.7.2 Les modifications à caractère de sabotage

Pour détruire avec plus ou moins de motivations des systèmes ou des données

Deux types de modifications

a) Infections informatiques à caractère unique

Bombe logique ou cheval de Troie

- Dans un programme normal on introduit un comportement illicite
- mis en action par une condition de déclenchement ou trappe (la condition, le moment ou l'on bascule d'un comportement normal à anormal)

Exemples: licenciement de l'auteur du programme

date quelconque

b) Infections auto reproductrices

Il s'agit d'une infection informatique simple (du type précédent) **qui contient de plus une partie de recopie** d'elle même afin d'en assurer la propagation

Virus : à action brutale

Ver : à action lente (détruisant progressivement les ressources d'un systèmes)

I.2 Politique de sécurité

I.2.1 Définition

Une politique de sécurité est un ensemble de règles qui fixent les actions autorisées et interdites dans le domaine de la sécurité.

I.2.2 Étapes types dans l'établissement d'une politique de sécurité

- Identification des vulnérabilités

. En mode fonctionnement normal (définir tous les points faibles)

. En cas d'apparition de défaillances un système fragilisé est en général vulnérable : c'est dans un de ces moments intermédiaires qu'une intrusion peut le plus facilement réussir

- *Évaluation des probabilités associées à chacune des menaces*

- *Évaluation du coût d'une intrusion réussie*

- *Choix des contre mesures*

- Évaluation des coûts des contre mesure

- Décision

I.2.4 Les méthodologies de sécurité

Réalisées par des grands utilisateurs de techniques de sécurité ou des groupes de travail elles sont applicables par des prestataires de service sous forme d'audit de sécurité

analyse de risques
propositions d'actions pour
améliorer la situation

I.2.4.1 Méthode M.A.R.I.O.N

Méthode d'Analyse des Risques Informatiques et Optimisation par Niveau. (à partir de 1984)

Norme :

CLUSIF: Club des Utilisateurs de La Sécurité Informatique Français

APSAD: Assemblée Pleinière des Sociétés d'Assurances Dommages

Objectif: Mettre en place le schéma directeur de la sécurité des systèmes d'information SDSSI

Trois approches selon le sujet traité:

- Marion-AP (avant-projet) (Applicable aux grands comptes et aux compagnies d'assurance)
- Marion-PME
- Marion-RSX (Applicable aux réseaux)

La Méthode Marion

Les six étapes d'élaboration du Schéma Directeur de Sécurité du Système d'Information

a) Analyse des risques

Établissement de scénarios de risques courus par l'entreprise.

b) Expression du risque maximum admissible

Calcul de la perte maximale subie par l'entreprise face à des événements mettant sa survie en péril

c) Analyse des moyens de la sécurité existants

Identifier et qualifier les moyens de la sécurité (organisation générale, physique et logique)

d) Évaluation des contraintes techniques et financières

Recensement des contraintes générales, techniques, humaines et détermination d'un budget pour la prévention et la protection

e) Choix des moyens de sécurité

Moyens à mettre en oeuvre ou à améliorer pour supprimer les risques en fonction des contraintes et du coût parade/risque

f) Plan d'orientation

Phase de bilan définissant le plan technique détaillé et rédaction finale du SDSSI.

I.2.4.2 Méthode M.E.L.I.S.A

Délégation générale à l'armement
1985.

MELISA S - Confidentialité des données
sensibles

MELISA P - Pérennité de
fonctionnement du système

MELISA M - Sécurité micro mini
informatique

MELISA. R - Sécurité réseau

II

Problèmes et techniques de la sécurité informatique

II.1 Les problèmes de la sécurité des données

II.1.1 Confidentialité des données

C'est la propriété qui assure que seuls les utilisateurs habilités, dans des conditions prédéfinies, ont accès aux informations.

Dans le domaine de l'entreprise cette garantie concerne

- le droit de propriété

 - .des secrets de fabrication

 - .des informations stratégiques

entreprise

 - niveau de production, de résultats

 - fichier clientèle

- le droit des individus

 - .défini par la loi informatique et liberté

II.1.2 Intégrité des données

C'est la propriété qui assure qu'une information n'est modifiée que dans des conditions pré définies (selon des contraintes précises)

Contraintes d'intégrité : l'ensemble des assertions qui définissent la cohérence du système d'information.

Exemples : -Toute règle de cohérence d'une base de données

-Une modification intempestive (même très temporaire) est à interdire (modification de bilan pour une journée).

I.1.3 L'authentification

C'est la propriété qui assure que seules les entités autorisées ont accès au système.

L'authentification protège de **l'usurpation d'identité** .

Signature (au sens classique) = **Authentification:**

La première idée contenue dans la notion habituelle de signature est que **le signataire est le seul** à pouvoir réaliser le graphisme (caractérisation psychomotrice)...

Entités à authentifier:

- une personne
- un processus en exécution
- une machine dans un réseau

Ne pas confondre **authentification** avec **confidentialité** ou **intégrité**.

L'authentification c'est un moyen clé de la sécurité pour assurer:

- **la confidentialité**

Celui qui lit une donnée est bien celui qui est autorisé à le faire (pour chaque fichier)

- l'intégrité

Idem : il ne suffit pas d'assurer l'intégrité des données. Celui qui a émis un message, un virement, est bien celui dont le nom figure dans le message, le virement, ..

II.1.4 Non répudiation

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

Signature (au sens habituel) = **Non répudiation** :

La seconde idée contenue dans la notion habituelle de signature est que le signataire s'engage à honorer sa signature: engagement contractuel, juridique, il ne peut plus 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 dénié par son émetteur.

b) La preuve de réception

Un récepteur ne peut ultérieurement dénier avoir reçu un ordre s'il ne lui a pas plu de l'exécuter alors qu'il le devait juridiquement.

Exécution d'ordre boursier, de commande, ..

II.1.5 Pérennité

Terminologie du milieu de la sécurité pour caractériser le bon fonctionnement du système informatique.

En termes de la sûreté de fonctionnement on parle de:

- *Disponibilité/Indisponibilité*

L'aptitude d'un système informatique à pouvoir être employé à un instant donné par les utilisateurs.

L'indisponibilité est une composante de la sécurité en ce sens qu'elle entraîne des pertes financières.

- *Fiabilité*

L'aptitude d'un système informatique à fonctionner correctement de manière continue pendant une période donnée (idée habituelle de pérennité).

II.2 Généralités concernant quelques techniques de sécurité des données en informatique

II.2.1 La protection (des interfaces) ou le contrôle de l'accès aux objets

A l'origine de la protection

- Idée de confinement des erreurs involontaires

Pour empêcher qu'un usager n'interfère avec un autre à la suite d'une erreur involontaire

Par exemple erreur de programmation détruisant l'espace mémoire d'un autre usager

Puis évolution vers le concept de contrôle d'accès

- Utilisation des mêmes moyens pour la validation des accès pour satisfaire les objectifs de confidentialité et d'intégrité

Techniques basées sur 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** .

Si l'on donne plus on risque de voir ces droits utilisés anormalement

- soit involontairement (aspect précédent)
- soit volontairement

Approche et moyens utilisés pour la protection

a) **Des mécanismes systèmes de désignation évolués**

- **Désignation des sujets**: entités opératoires comme les usagers, les processus, les procédures,...)

- **Désignation des objets** manipulés : comme les fichiers, les segments mémoire, les portes de communication....).

b) **Des mécanismes de gestion des droits d'accès (des sujets aux objets)**

c) **Une vérification des droits** (idéalement lors de toute opération, de manière moins stricte lors de l'ouverture de l'accès à un objet par un sujet).

d) **Une authentification correcte des sujets** (de façon à ce que les droits ne soient pas usurpés).

Objectif final poursuivi

L'ensemble des objets et des droits constitue pour chaque sujet **un domaine de protection dont l'interface** (la frontière) **est infranchissable** sauf autorisation explicite (dans tous les sens).

Les systèmes à capacités

On appelle **capacité** un droit d'accès que possède un sujet sur un objet.

Pour chaque sujet on gère au niveau système (dans des segments de mémoire particuliers) **l'ensemble des capacités** de ce sujet (liste de capacités ou **c-liste**).

Exemple: Dans le système VAX-VMS les primitives du noyau sont protégées par de très nombreux droits d'accès qui sont spécifiques de chaque utilisateur:

- droit de créer des processus
- droit de faire des opérations réseaux
- droit de modifier ses droits

....

A chaque fois qu'un sujet ouvre un objet

Ou a chaque fois qu'il accède à un objet

selon la nature de l'objet et le niveau de protection souhaitée on vérifie le droit du sujet

Problèmes à résoudre:

- **authentifier** les utilisateurs.

- **empêcher** la modification illégale des droits (passage par des guichets pour l'élévation des privilèges)

- **contrôler** le transfert des droits (existence d'un droit de transfert des droits).

- **assurer la révocation des droits** (si requête de l'utilisateur propriétaire ou à la destruction de l'objet).

Solutions implantées

- machines à anneaux

- machines à domaines

Les listes de droits

On définit pour chaque objet la liste des sujets et leurs droits sur l'objet.

Si l'on considère la matrice (sujet, objet) dont les éléments sont les droits d'accès du sujet sur l'objet:

Listes de capacités

- une approche de capacité consiste à stocker la matrice par lignes

Listes de contrôle d'accès

- une approche de liste de contrôle d'accès consiste à stocker la matrice par colonnes.

Exemple : Dans le système de fichier d'UNIX (et dans presque tous les systèmes de fichiers on a une approche analogue)

On associe à chaque fichier une liste de droits (lire, écrire, exécuter)

a) Pour un usager particulier :le propriétaire du fichier

b) Les membres du groupe du propriétaire

c) Tous les usagers du systèmes

NB Dans d'autres systèmes de fichiers les droits sont plus fins et l'on peut définir

des listes pour d'autres groupes ou même pour chaque utilisateur (véritables listes de contrôle d'accès pour les objets fichiers ex VAX-VMS).

Matrice de droits

	O1		Oj		Om
S1					
Si			Droits de Si sur Oj		
Sn					

(1)

(2)

(1) Liste de capacité

(2) Liste de contrôle d'accès

Choix entre les deux approches

Repose sur le coté plus ou moins pratique de la manipulation des droits:

1 Les listes de contrôle d'accès sont plus pratiques pour la révocation des droits (puisque'ils sont associés à l'objet).

2 Pour le stockage des droits et la rapidité de leur consultation cela dépend des objets:

- Les c-listes sont rangées avec les descriptifs des sujets.

Favorable pour ce qui concerne les droits des objets manipulés en mémoire centrale par le noyau (peu de sujets actifs)

- Les listes de contrôle d'accès doivent être rangées avec les descriptifs des objets

Favorable dans le cas des fichiers

II.2.2 Les techniques d'authentification

L'authentification = vérification de l'identité d'une entité.

L'une des mesures les plus importantes de la sécurité:

- Il est impossible d'assurer la confidentialité, l'intégrité, la non répudiation sans s'appuyer sur la garantie préalable indiscutable de l'identité de l'entité soumettant une requête.

- L'authentification devrait être assurée en continu. Il ne suffit pas de réaliser l'authentification une fois pour toute à l'ouverture d'un objet (en début de session)

. quand l'entité est une personne :

elle peut quitter son poste en le laissant ouvert => procédure de déconnexion automatique, procédure de réautorisation périodique.

. quand l'entité est informatique:

une substitution peut avoir lieu (surtout en réseau, nécessité de protocoles de sécurité)

Les moyens de l'authentification

Deux problèmes de nature différente
 : authentification des personnes,
 authentification des entités
 informatiques

L'authentification peut se faire par trois méthodes:

1 Ce que connaît l'utilisateur

Le mot de passe, le code confidentiel.

Technique:

la plus simple
 la plus répandue
 applicable aux entités informatiques

Problèmes bien connus:

- le secret peut-être découvert par un tiers
- le secret peut-être confié à un tiers

Quelques parades:

- **Obliger l'usager à changer régulièrement de mot de passe.**

- **Surveiller les tentatives d'accès illicite par comptage (les afficher).**

- **Prévenir l'utilisateur des connexions précédentes sur son compte en affichant la date et l'heure (par exemple du dernier accès).**

2 Ce que détient l'utilisateur

Un secret matérialisé physiquement

**La clé traditionnelle, la carte
(magnétique, à code barre, à puce)**

Technique simple, répandue.

Les problèmes :

- la perte, le vol du support
- la duplication (plus ou moins facile mais toujours possible)

3 Ce qu'est l'utilisateur: les méthodes biométriques

Une solution en rapide développement.

Qui peut-être très efficace (à la limite trop en cas de décès par exemple)

Le plus souvent onéreuse (dans l'état actuel des procédés)

Qui peut-être difficile à accepter dans certains cas par l'utilisateur

Techniques biométriques : Généralités

Elles sont issues:

- d'un **caractère morphologique** (biologique) a priori caractérisant de manière unique l'utilisateur mais dont le prélèvement peut ne pas suivre exactement un profil préenregistré (en raison même de son type).

- de la **classification automatique** des caractères d'un ensemble d'utilisateurs (analyse statistique, réseaux de neurones).

Nécessité d'études approfondies du caractère utilisé

- à l'intérieur du groupe humain des utilisateurs autorisés.

- ou dans une population quelconque (analyse de la variabilité du caractère retenu)

Incertitudes des techniques biométriques

- La variabilité intra-individuelle.
- La variabilité inter-individuelle.
conduisant 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 à l'étude

- L'empreinte digitale

Le sujet applique son doigt sur un prisme

La pression déclenche une analyse par balayage d'un faisceau infrarouge

Le signal reçu dépendant des sillons de l'empreinte (creux et bosses successives) est analysé et classifié).

- La vascularisation de la rétine

C'est une empreinte biométrique analogue à l'empreinte digitale qui est très stable.

L'image de fond de l'oeil est obtenue par un dispositif monoculaire utilisé dans les tests de vision médicaux.

La numérisation est effectuée par une caméra infrarouge

L'image est classifiée

- *La voix*

Le sujet prononce quelques mots

Le système numérise et classe le signal

- *La géométrie de la main*

Le sujet applique une main sur un support

Une caméra enregistre l'image.

La caractérisation est obtenue par mesure de la longueur et de la largeur de chaque doigts.

- *Dynamique de la signature*

Elle est obtenue par une tablette graphique et un stylo reliés à un ordinateur

La signature et surtout le mouvement réalisé par la main pour la fabriquer sont analysés en comparaison à plusieurs signatures de référence.

- *Dynamique de la frappe clavier*

Un clavier spécial permettant la mesure précise des intervalles dans les séquence de frappe ou la pression des doigts sur le clavier est utilisé

- *Empreinte génétique*

Analyse du code génétique de l'individu
Demande encore actuellement plusieurs heures.

II.2.3 Les protocoles de sécurité

a) **Caractère temporel de la sécurité:** une technique n'a réellement besoin d'être sûre que pour un laps de temps donné.

b) **Importance croissante des réseaux** (qui posent des problèmes graves de sécurité)

Deux facteurs qui amènent à définir des protocoles de sécurité (des suites d'échanges) permettant d'atteindre un niveau de sécurité suffisant sur une période donnée.

La technique essentielle est la **cryptographie**

Exemples de problèmes à résoudre:

- dans des systèmes à mots de passe (à clés secrètes): **protocoles pour l'échange sécurisé de clés .**

- dans des systèmes de transactions réparties : **protocoles assurant la confidentialité des informations,**

**protocoles assurant l'authentification
de l'émetteur d'une transaction.**

-

II.2.4 Les techniques de contrôle de sécurité informatique.

II.2.4.1 La validation formelle de sécurité

L'objectif est de démontrer formellement qu'un système ou sont implantées des techniques sécuritaires **réalise bien ses objectifs de sécurité.**

Une approche analogue à la preuve du logiciel mais une différence importante:

La preuve classique s'arrête le plus souvent à vérifier que **pour toutes les entrées correctes un programme (un système) produit des sorties attendues**

La validation formelle de sécurité doit prouver que le système est de sécurité: il revient toujours en un état de sécurité prédéfini.

- **Le système produit des résultats prédéfinis pour des entrées prédéfinies.**

- **Il ne fait que cela.** Il ne rajoute pas des effets de bord favorables (ce qui n'est pas grave) mais surtout mauvais (insécuritaires ou dégradant le système).

- Quelle que soit la configuration plus ou moins dégradée du fonctionnement.

II.2.4.2 L'audit de sécurité informatique

C'est l'opération d'évaluation et de contrôle des moyens de prévention et de protection des risques informatiques.

Nombreuses techniques à développer:

- Mise en place des scénarios d'attaques possibles.

- Analyse des moyens de stockage des données permettant de percevoir les attaques

- Analyse des moyens de détection en ligne utilisant ces données

- Analyse des moyens pour l'exploitation ultérieure hors ligne permettant de définir de nouvelles contre mesures.

Exemples:

- Concernant les tentatives d'intrusion (mise en défaut des mécanismes d'authentification à mots de passe)

- Enregistrement du maximum d'informations concernant tous les accès ayant réussis ou échoués.

III**LES PROTOCOLES DE
SÉCURITÉ**

III.1 Introduction à la cryptographie

III.1.1 Le chiffrement

$$M \xrightarrow{E_k} C$$

- Texte (message) **M** en clair :

Une information dans sa forme de base

- Texte (message) **C**
chiffré,
crypté,
codé,
brouillé,
ou cryptogramme :

l'information transformée de façon à ce que son sens soit caché

- L'opération de transformation **E_k** est appelée :

le chiffrement,
le cryptage,
l'encryptage,
le codage,
le brouillage

Un chiffre concerne plutôt une technique de cryptage portant sur des éléments de taille fixe (caractères alphabétiques par exemple).

Un code désigne plutôt un cryptage portant sur des éléments de taille variable (mots ou phrases)

- La possibilité de crypter repose sur la connaissance de:

la clé (algorithme E, secret k)

l'ensemble des paramètres permettant la réalisation des opérations de cryptage ou de chiffrement.

III.1.2 Le déchiffrement

$$D_{k'} \\ C \text{ -----} > M$$

- Déchiffrer un message chiffré C est l'opération qui permet de restituer un texte en clair M à partir d'une clé de déchiffrement $D_{k'}$ que l'on possède.

- Décrypter ou casser un code c'est parvenir au texte en clair sans posséder au départ les règles ou documents nécessaires au chiffrement.

- L'art de définir des codes est la cryptographie (cryptographe).

- L'art de casser des codes est appelé cryptanalyse ou cryptologie (cryptanalyste, cryptologue ou casseur de codes)

- Un cryptosystème est l'ensemble des deux méthodes de chiffrement et de déchiffrement utilisable en sécurité.

III.1.3 Propriétés des cryptosystèmes

- Propriétés de base indispensables

$$a) D_{k'} (C) = D_{k'} (E_k (M)) = M$$

$D_{k'}$ est la fonction inverse de E_k

b) E_k et $D_{k'}$ dépendent d'informations partiellement ou totalement secrètes

- Propriétés importantes générales

a) E_k et $D_{k'}$ doivent être de préférence simples à appliquer de manière à:

-atteindre des vitesses de chiffrement élevées

- éviter un encombrement important des clés pour tout k k' d'un domaine d'emploi

b) On estime que la sécurité ne doit pas dépendre du secret des algorithmes E et D mais uniquement du secret des clés k et k' .

- Propriétés importantes pour l'usage

a) Le code doit être très difficile à casser en partant uniquement des messages cryptés

b) Le code doit être très difficile à casser même si l'on dispose d'un échantillon de messages et des messages cryptés correspondants .

c) On ne doit pas pouvoir créer des textes C' qui aient l'apparence de textes cryptés

($D_{k'} (C')$ est un texte valide).

III.1.4 Les deux catégories de cryptographie

III.1.4.1 Les chiffres symétriques (à clé privée)

- $D_{k'}$ et E_k sont très liés du point de vue du secret.

- On peut déduire très facilement l'une des clés de la connaissance de l'autre : pratiquement $k = k'$.

Exemple : Décalage des lettres de l'alphabet de n positions.

La fonction directe (chiffrement) et son inverse (déchiffrement) se réalisent directement à partir de la connaissance de n .

III.1.4.2 Les chiffres asymétriques (à clé publique)

- On choisit deux méthodes D_k' et E_k qui sont telles qu'il est très difficile de déduire D_k' de la connaissance de E_k .

Il faut trouver une fonction dont la fonction inverse est difficile à déterminer)

- On peut donc rendre E_k publique (la clé publique) connue de tous dans un annuaire car c'est très pratique.

- Par contre la clé D_k' (la clé privée) doit rester secrète et particulariser chaque utilisateur.

- Propriété complémentaire (très utile) d'un système à clé publique: la commutativité.

$$D_{k'} (E_k (M)) = E_k (D_{k'} (M)) = M$$

III.1.5 Problème de la cryptographie

La sécurité d'un cryptosystème repose en fait sur l'analyse de la **complexité** des algorithmes définis et sur les puissances de calcul disponibles pour une attaque.

=> domaine évoluant en permanence avec la recherche

Exemples :

- Le crypte de Vigenere
Publié comme incassable au début du XX siècle => Décrypté 20 ans plus tard
- L'algorithme du sac à dos
Proposé comme une solution à clé publique => Rejeté en quelques années
- Le DES 56 bits => Déclassifié en 1988
- Une norme OSI propose un méthode de chiffrement => Version suivante : addendum déconseillant l'emploi du crypte comme non sécuritaire
- Quid du RSA ("meilleur" algorithme actuel) publié comme incassable en 1978 dans la même revue qui a publié le chiffre de Vigenere.

Le chiffrement absolu n'existe pas

III.2 UTILISATION DE LA CRYPTOGRAPHIE DANS LES PROTOCOLES DE SÉCURITÉ

III.2.1 Le problème de confidentialité

- Repose uniquement sur l'existence d'un algorithme de cryptage efficace
- Pour une information stockée I ou un message M.

III.2.1.1 Cryptographie à clé privée (E_k, D_k sont des secrets)

On stocke $E_k (I)$ On envoie $E_k (M)$

Personne d'autre que les détenteurs du secret (E_k, D_k) ne savent chiffrer ou déchiffrer

III.2.1.2 Cryptographie à clé publique E_k , clé privée D_k'

On stocke $E_k (I)$ On envoie $E_k (M)$

Tout le monde sait encrypter, donc tout le monde peut crypter des données ou envoyer des messages cryptés.

Seuls les détenteurs du secret D_k' peuvent retrouver le texte en clair

III.2.2 Le problème d'intégrité

III.2.2.1 En cryptographie à clé privée : E_k , D_k

On peut utiliser le même principe que précédemment basé sur la seule 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.

Le cryptage est relativement coûteux si les données sont longues .

Or il suffit de crypter une information courte comme un code polynomial ("CRC") caractéristique de tout un message pour s'apercevoir des modifications éventuelles.

On stocke I , $CRC (I)$, $E_k (CRC (I))$

On envoie M , $CRC (M)$, $E_k (CRC (M)$
)

- Tout le monde connaît la méthode de calcul des CRC et peut donc modifier les parties commençantes en leur donnant l'apparence de la cohérence.

- Seul un utilisateur autorisé peut générer la signature cryptée

correctement ou la vérifier pour détecter ainsi des modifications (atteintes à l'intégrité).

III.2.2.2 En Cryptographie à clé publique (E_k , $D_{k'}$)

On stocke I , $CRC(I)$, $D_{k'}(CRC(I))$
 On envoie M , $CRC(M)$, $D_{k'}(CRC(M))$
)

Solution analogue à la précédente

Il faut **générer le CRC crypté au moyen de la clé privée** pour que seuls les utilisateurs autorisés puissent générer des données (des messages) corrects.

Tout le monde pourra vérifier ensuite leur correction au moyen de la clé publique.

III.2.2.3 Intégrité d'un message dans un flot de messages

Un flot d'échanges de longue durée doit être caractérisé par une **connexion**.

Une **connexion dispose d'une référence**: un identifiant qui permet de distinguer les messages appartenant à des connexions différentes.

La référence de connexion est unique au moment où la connexion est ouverte, mais la référence est le plus souvent **réutilisée ultérieurement** (c'est généralement un numéro d'entrée dans la table des connexions ouvertes).

Chaque message d'une connexion est **numéroté** et donc est identifié, mais **les numéros sont réutilisés de manière circulaire** et donc réapparaissent ultérieurement (modulo 7 par exemple).

Problème posé par la répétition :

- Une tentative de répétition est possible

- Réutilisation d'un message crypté d'une connexion ancienne .

- Réutilisation d'un message ancien de la même connexion ayant un numéro cohérent dans le flot courant.

Un message dupliqué (mais correct du point de vue connexion, séquence et signature) peut être inséré dans un flot par un usager malveillant et menacer l'intégrité de l'application.

Solutions au problème de répétition:

- a) *Utiliser de très grands espaces de numérotation*

- . des connexions et des messages
- . par exemple à 32 ou 64 bits
- . afin de rendre la réutilisation excessivement problématique.

- b) *Utiliser un estampillage unique supplémentaire des messages*

- . par l'horloge physique de l'émetteur (datation des messages).
- . nécessite de disposer d'un **protocole de synchronisation d'horloge** (entre l'émetteur et le récepteur)
- . permet au récepteur de **vérifier la cohérence du message par sa date** et évite ainsi les répétitions non détectées.

III.2.3 Le problème d'authentification

III.2.3.1 L'authentification d'un usager d'un système informatique

III.2.3.1.1 Version de base par mot de passe

- *Chaque utilisateur dispose :*
 - d'un nom U
 - d'un mot de passe secret P

- *On pourrait stocker en fichier de mots de passe*
 - en clair U, P
 - mais protection des fichiers trop faible

- *On suppose l'existence d'un crypte E_k*
 - on stocke $U, E_k(P)$

- *Qualité de la fonction d'encryptage*
 - Etre employée dans un seul sens
 - On sait uniquement encrypter
 - Personne ne sait décrypter
 - Notion de "fonction univoque"
 - ("one way function")

Exemples de génération de fonctions univoques au moyen de cryptosystèmes

On dispose de l'un des deux types de cryptage.

a) *Un algorithme à clé privée*

- Utilisé avec comme clé le mot de passe lui même.

- Seul le détenteur du mot de passe peut crypter celui-ci et le vérifier.

b) *Un algorithme à clé publique*

- On génère un couple $E_k, D_{k'}$

- On détruit la clé de déchiffrement $D_{k'}$

On sait toujours chiffrer des données, mais personne ne sait les déchiffrer.

Seul le détenteur du mot de passe correct peut vérifier celui-ci.

Fonctionnement d'une authentification d'utilisateur à mot de passe

- A chaque ouverture de session l'utilisateur présente son mot de passe
 - Il est immédiatement crypté
 - On compare le crypte obtenu à celui enregistré dans le fichier des mots de passe
 - Vision théorique des choses: le fichier peut-être accessible à tous en lecture puisque personne ne sait décrypter
- Exemple du système UNIX
 - Le mot de passe est sur 8 caractères.
 - On en fait une clé de 56 bits pour chiffrer avec l'algorithme du DES.
 - On chiffre un texte composé de 64 bits à 0 avec la clé précédente.
 - On réitère 25 fois sur chaque crypte obtenu successivement.
 - Le résultat est traduit en 11 caractères imprimables placés dans un fichier (/etc/passwd) accessible en lecture par tous les usagers.

- Problèmes de l'approche UNIX

-A) Utilisation par les "pirates " de techniques de décryptage "force brute" .

1) *Un intrus essaye des mots de passe au hasard* (surtout si la longueur de P est faible)

Très facile surtout si l'intrus a pu recopier le fichier des mots de passe sur son ordinateur

Avec un ordinateur peu puissant le calcul du crypte UNIX prend de l'ordre de 1 seconde

Avec un circuit DES on peut le faire en 1ms.

2) *Meilleure technique de piratage*

Les mots essayés sont:

Tirés d'un dictionnaire de la langue de l'utilisateur.

Tirés d'une liste de prénoms

On peut ainsi casser jusqu'à 30% des mots de passe.

- B) **Espionnage de la ligne entre la console utilisateur et le ordinateur**

Le mot de passe y circule en clair
Par exemple sur ethernet (espion de ligne)

III.2.3.1.2 Mot de passe avec clé complémentaire aléatoire

Différentes solutions pour contrer les attaques force brute (utilisées indépendamment ou en conjonction)

- Interdiction de plus de n tentatives en échec
- Mise en protection du fichier des mots de passe (accès contrôlé par primitive système)
- Interdiction d'utilisation des mots de passe cassable facilement (mots du dictionnaire, ...).
- Adjonction au mot de passe P connu de l'utilisateur d'une clé complémentaire N très difficile à deviner (l'heure de création du mot de passe, une clé aléatoire)

On stocke dans le fichier des mots de passe protégé $U, N, E_k(P, N)$

A chaque ouverture on ajoute au passe P fourni par l'utilisateur la clé N avant de crypter.

Exemple :

Pour certains unix une clé complémentaire de 12 bits ("salt") est construite à partir de l'UID (code interne utilisateur) et de l'heure de génération du mot de passe.

On multiplie par 4096 le nombre d'essais à réaliser par mots de passe pour un pirate.

III.2.3.1.3 Procédures sécurisées d'authentification par mot de passe sur les réseaux

. Pour assurer la confidentialité du mot de passe:

- cryptage de celui ci

. Un intrus pourrait cependant

- enregistrer le mot de passe crypté

- le réutiliser ultérieurement sous sa forme cryptée (répétition, "replay")

. Solutions:

-Utilisation de la date précise d'ouverture pour rendre inopérantes les tentatives en répétition.

-Utilisation d'une synchronisation des horloges entre le site utilisateur et le site d'authentification pour vérifier la cohérence des dates et déjouer les tentatives de répétition.

La solution (à clé privée)

- Les deux entités ont en commun une clé secrète k_u
- Les deux sites ont des horloges synchronisées

Site Utilisateur
Authentificateur

Site

1 : Utilisateur U

Demande d'ouverture(U)

login(U)

2 :
fichier

Consultation

d'autorisation

3 :
 k_u

Lecture clé de U:

4 :
T

Lecture de la date

5 :
cryptée

Émission

T

$$X = E_{k_u} (T)$$

6 : **Déchiffrage** $T = D_{k_u} (X)$

7 : **Vérification** cohérence T

8 : U donne son mot de passe
et l'heure courante cryptés

$$Y = E_{k_u} (T, P)$$

9 : **Déchif T,P = Dku**
(Y)

10 : **Fin : vérification habituelle du passe**

Remarques :

- A envoie son heure courante T cryptée pour qu'un intrus ne puisse rejouer une séquence enregistrée à sa place. U vérifie la cohérence de T.

- U retransmet l'heure fournie par A en plus de son mot de passe pour la même raison.

- Le mot de passe ne peut-être obtenu par espionnage du réseau par un intrus.

- Solution encore plus sophistiquée: Kerberos.

III.2.3.1.4 Solution d'authentification d'un usager dans un réseau à clés publiques

- Les deux entités ont chacun un couple clé publique, clé privée:

(Eu , Du) pour l'utilisateur

(Ea , Da) pour l'authentifieur

- Tout le monde connaît les clés publiques Eu, Ea.

- Seul U connaît Du et seul A connaît Da

- On a la propriété de commutativité de l'algorithme à clé publique utilisé

$$E (D (M)) = D (E (M)) = M$$

- Les deux sites ont des horloges synchronisées

Site	Utilisateur	Site
Authentificateur		

1 : Demande d'ouverture(U)
 $\text{login}(U)$

----->

2 : Lecture de l'heure T

3 : Émission de T cryptée
 par la clé secrète de A
 $X = D_a (T)$

<-----

4 : Déchiffrage de T avec E_a
 clé publique de A : $T = E_a (X)$

5 : Vérification cohérence
 de l'heure transmise T

6 : U renvoie l'heure courante
 cryptée par sa clé secrète
 $Y = D_u (T)$

----->

7 : Déchiffrage $T = E_u (Y)$
 par la clé publique de
 U

8 : Fin d'ouverture

Remarques :

- En fait il n'y a plus de mot de passe au sens habituel: c'est la connaissance par l'utilisateur U de sa clé privée qui lui sert de mot de passe

- A envoie son heure courante T cryptée pour qu'un intrus ne puisse rejouer une séquence enregistrée à sa place. Seul A peut générer ce message puisque seul A connaît Da

- U vérifie la cohérence de T ce qui empêche la répétition d'un ancien échange.

- U peut décoder T (comme tout le monde d'ailleurs) car U connaît la clé publique de A.

- U retransmet l'heure fournie par A en la codant de sa clé privée. Lui seul peut le faire, ce qui pour A est équivalent à la fourniture d'un mot de passe.

- A peut décoder et vérifier la date ce qui évite à nouveau la répétition.

III.2.3.2 L'authentification d'un message (d'une donnée) d'un système informatique

III.2.3.2.1 Cas d'un algorithme de cryptage à clé privée

On stocke $E_k (I)$

On envoie $E_k (M)$

Personne d'autre ne sait encrypter pareillement, sauf ceux qui détiennent le secret D_k

L'émetteur comme le destinataire (d'un message) doivent donc partager une clé secrète unique de manière à assurer à la réception que seul l'émetteur autorisé a pu fabriquer le message.

Problème posé :

la fabrication puis la distribution de nombreuses clés de session secrètes (traité plus loin)

Remarques:

On obtient d'une seule opération la confidentialité, l'intégrité, l'authentification.

Comme précédemment on peut éviter de coder tout en ne signant que le CRC mais on perd la confidentialité.

III.2.3.2.2 Cas d'un algorithme de cryptage à clé publique

A Solution avec authentification seule

Comme un usager émetteur d'information E est le seul à connaître une clé privée D_e il suffit qu'il signe de cette clé:

On stocke $D_e (I)$

On envoie $D_e (M)$

Si le récepteur, en appliquant la clé publique E_e de l'émetteur arrive à décoder le message alors il est sûr de l'identité de l'émetteur

Personne d'autre ne sait encrypter pareillement, sauf ceux qui détiennent le secret D_e

Mais l'échange n'est absolument pas confidentiel puisque tout le monde peut décoder le message avec la clé publique E_e de l'émetteur.

B Authentication et confidentialité d'une transmission de message en cryptographie à clés publiques

- Les deux entités communicantes ont un couple clé publique, clé privée/

(Ee , De) pour l'émetteur

(Er , Dr) pour le récepteur

- Tout le monde connaît les clés publiques Ee, Er.

- Seul E connaît De et seul R connaît Dr

- On a $E (D (M)) = D (E (M)) = M$

Du récepteur vers l'émetteur : la preuve d'origine

- Authentication et confidentialité de l'émetteur vers le récepteur

Comme un usager émetteur d'information **E est le seul à connaître une clé privée** De il doit commencer par signer de cette clé.

De manière à autoriser seulement le récepteur à accéder aux informations **il doit ensuite signer de la clé Er**

Protocole d'authentification et de confidentialité à clé publique:

message M

signature $S = E_r (D_e (M))$

----->

S A la réception de

$$D_r(S) = D_r(E_r(D_e(M))) = D_e(M)$$

$$\text{Puis } E_e (D_e (M)) = M$$

Seul E a pu envoyer M car seul E peut appliquer D_e .

Seul R peut accéder à M car seul R peut appliquer D_r .

Du récepteur vers l'émetteur : la preuve de remise

Avec le protocole précédent le récepteur est certain que l'émetteur est bien l'émetteur autorisé mais l'émetteur n'est pas assuré de la remise de son message au destinataire autorisé.

On réalise le même protocole en retour pour transporter un acquittement noté ici ACQ.

acquittement ACQ (M)
signature $S = Dr (ACQ$
(M))

<-----

A la réception de S
 $Er (Dr (ACQ (M))) = ACQ (M)$

- Seul R a pu envoyer ACQ(M) car seul R peut appliquer Dr

Remarques:

- On peut aussi protéger la confidentialité de l'acquittement par $Ee(Dr(ACQ(M)))$.

- On ne résout pas le problème de non répudiation car l'émetteur comme le récepteur peuvent prétendre qu'ils ont perdu ou qu'on leur a dérobé leur clé.

III.2.4 Le problème de non répudiation

Il s'agit d'éviter un déni de responsabilité d'une entité communicante :

- Aussi bien du récepteur qui ne peut dénier avoir reçu un ordre
- que de l'émetteur qui ne peut dénier l'avoir donné.

Dans toutes les circonstances l'utilisateur peut prétendre que l'on a usurpé son identité.

C'est un problème principalement juridique mais qui doit être réglé dans le cadre de techniques informatiques.

Deux catégories de réponses:

- ***La responsabilité totale du secret des clés***
- ***La notarisation***

III.2.4.1 *La responsabilité totale du secret des clés*

Les émetteurs ou les récepteurs ne peuvent en aucun cas arguer de la perte de leur code s'ils n'en ont pas fait la déclaration immédiate.

Toute utilisation du secret hors déclaration de perte engage la responsabilité du détenteur (Ex: carte bancaire)

Dans cette hypothèse on peut utiliser des techniques d'authentification électronique et de preuve de remise.

- C'est la bonne foi de l'une des parties contre celle de l'autre (en général en faveur des banques pour la carte bancaire)

- Mais c'est également la qualité des appareils et du protocole de sécurité qui est en cause (des jugements ont été rendus contre les banques dans certains cas)

- Tous les conflits ne peuvent qu'être arbitrés en justice.

III.2.4.2 *Les techniques de notariation*

Les transactions sont effectuées par l'intermédiaire d'une entité juridiquement sûre (un notaire électronique).

C'est une tierce partie reconnue contractuellement par les entités qui communiquent.

Ex le réseau international inter bancaire SWIFT

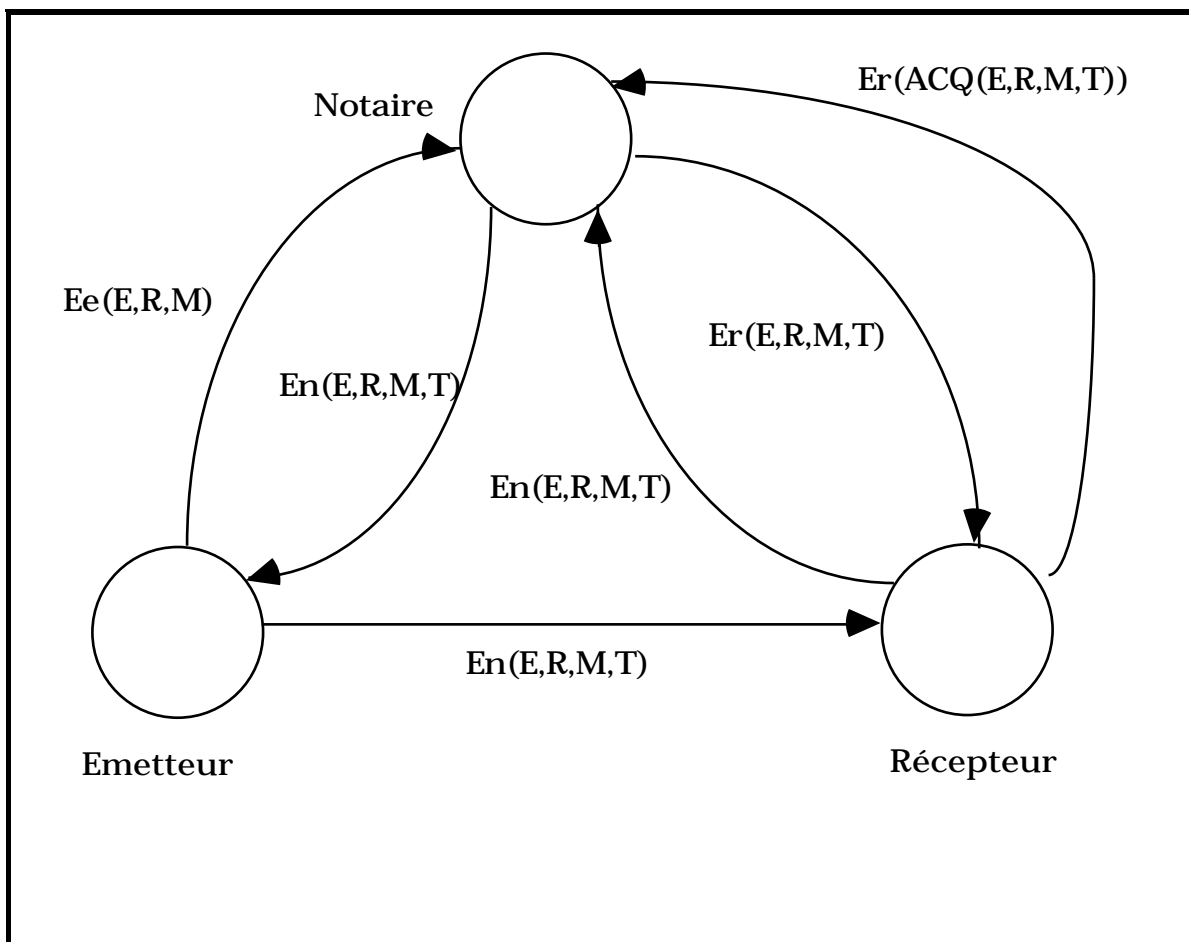
Le notaire garantit la sécurité d'une transaction:

- **confidentialité**
- **authentification**
- **non répudiation**
par la datation, la journalisation de la transaction.

de manière à réduire l'incertitude en cas de conflit.

Exemple - Un protocole de notariisation utilisant des clés privées.

Le notaire doit disposer des clés E_e , D_e et E_r , D_r de (émetteur, récepteur). et d'une clé secrète propre E_n , D_n



1 - L'émetteur E souhaite envoyer le message M au destinataire R de façon notarisée :

Il envoie au notaire $E_e (E , R , M)$

2 - Le notaire qui reçoit la transaction peut la décoder puisqu'il connaît D_e . Il date la transaction T et la journalise:

Enregistrement de E , R , M , T

La transaction ne pourra pas ensuite être reniée par E.

3 - Le notaire possède une clé secrète personnelle E_n qu'il utilise pour signer la transaction $S = E_n(E,R,M,T)$.

Il renvoie cette signature en réponse à E qui va la conserver pour preuve de la notariation effectuée.

4 - L'émetteur E envoie alors la transaction à son destinataire sous la forme S signée par le notaire.

De la sorte il ne peut avoir modifié celle ci entre temps

R ne peut encore interpréter les informations mais il enregistre S pour preuve de la requête de E.

5 - Pour connaître M, R demande au notaire le déchiffrement de $E_n(E, R, M, T)$

Le notaire envoie à R la transaction chiffrée avec la clé de R soit $E_r(E, R, M, T)$. Seul R peut la lire confidentialité, intégrité et authenticité.

6 - Pour terminer complètement le protocole il faut que le notaire dispose d'une preuve de remise à R soit une réponse:

$E_r(ACQ(E, R, M, T))$

que le notaire enregistre.

III.2.5 Le problème de distribution des clés

Avec les systèmes de cryptographie (à clés privées mais aussi à clés publiques) la sécurité dépend surtout du secret des clés et donc on doit changer souvent de clés.

III.2.5.1 Distribution manuelle des clés

Les clés sont fabriquées par un office central

Les clés sont distribuées par courrier postal ou tout autre procédé de livraison.

Solution assez médiocre:

- Vulnérable
- Très lente
- Peu pratique et coûteuse si l'on change souvent

III.2.5.2 Distribution par réseau des clés

Profiter des possibilités de transport des données offertes par le réseau

Sans mettre en jeu la sécurité.

III.2.5.2.1 Distribution hiérarchique

On utilise plusieurs niveaux de clés:

- **Un niveau supérieur** (par exemple un serveur national) sert à la fabrication et à la distribution de clés au niveau inférieur (par exemple régional).

- Pour distribuer des clés régionales, le niveau national dispose d'une clé unique qui n'est utilisée que pour la distribution (afin d'éviter d'offrir des messages longs cryptés aux cryptanalyste pour une attaque statistique)

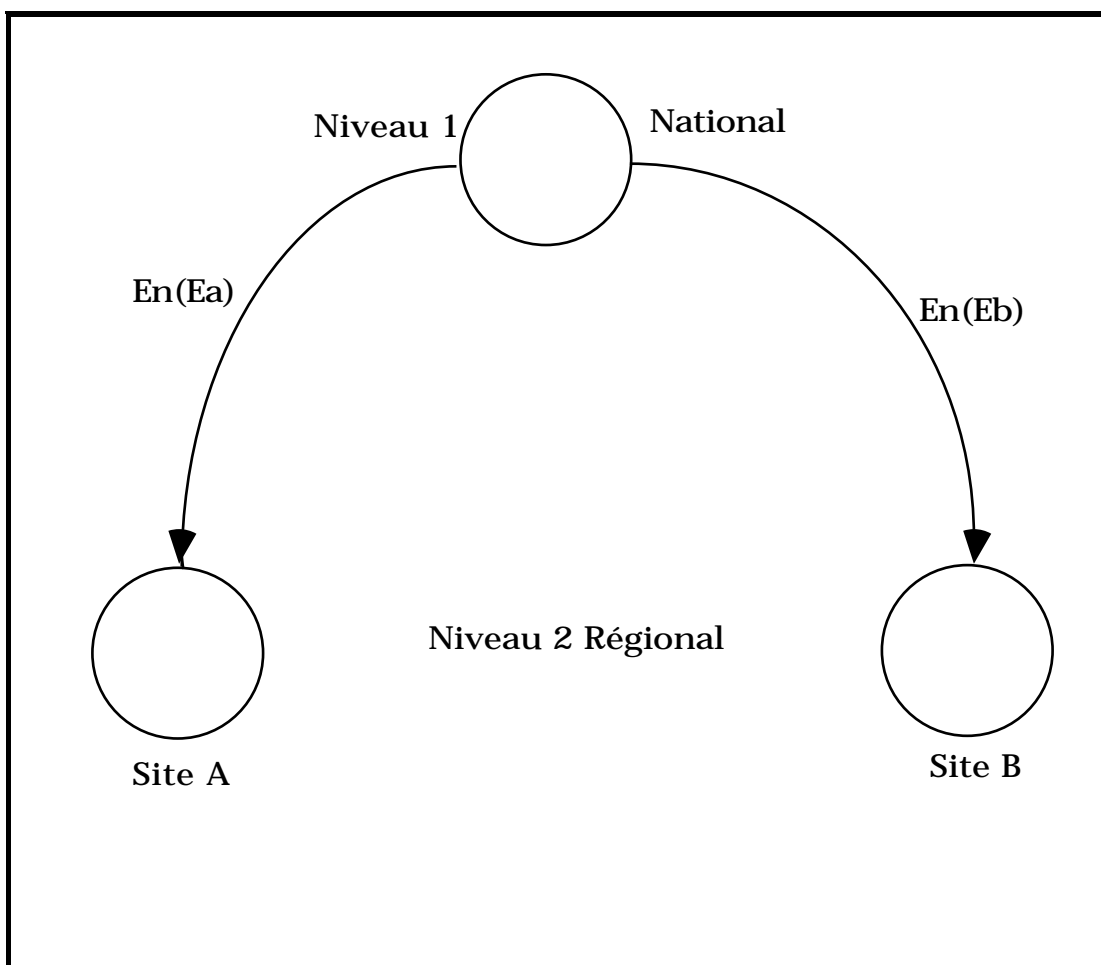
- La clé du plus haut niveau doit être distribuée manuellement.

Exemple de protocole de distribution de clés de session à deux niveaux:

- Chaque site du niveau 2 (régional) connaît la clé de distribution des clés du niveau 1 (national)

- Chaque site du niveau 2 reçoit tout d'abord du niveau national une clé secrète propre qui lui permet de dialoguer avec ce niveau sans utiliser en permanence la clé nationale qui serait vulnérabilisée

- Il déchiffre sa clé secrète avec la clé nationale.



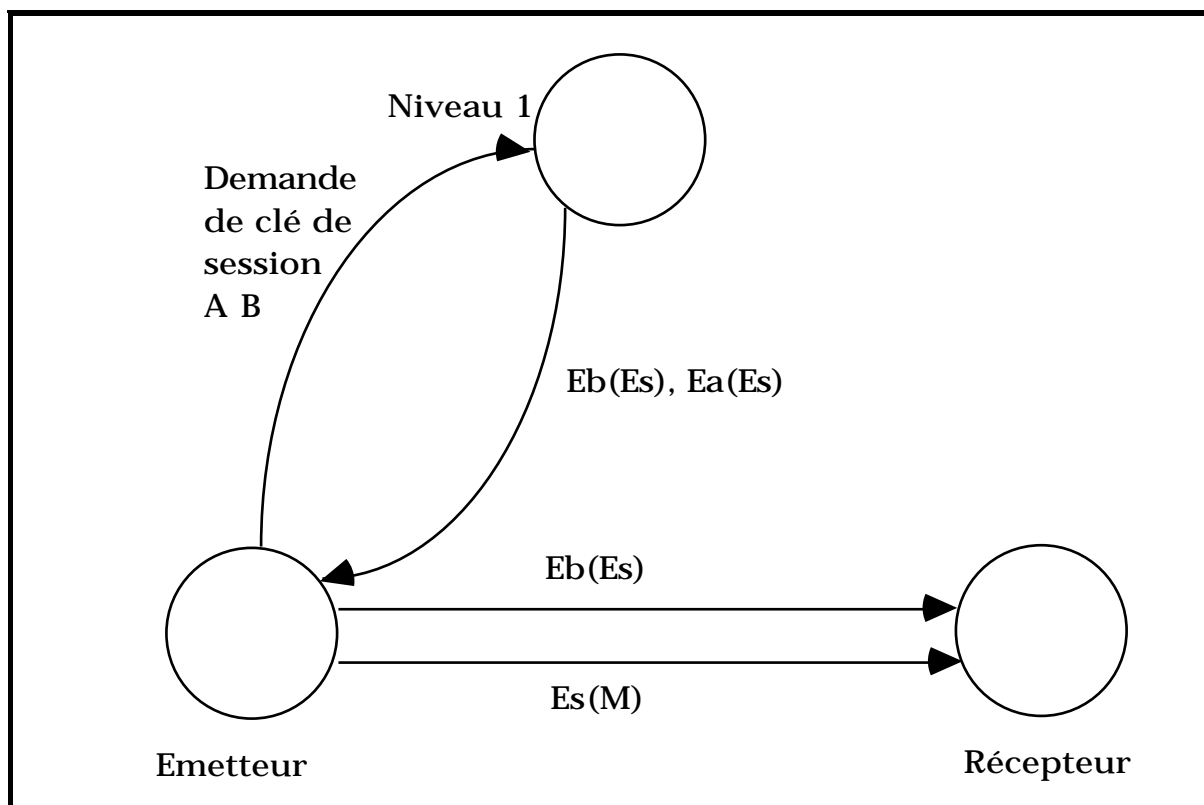
- Obtention d'une clé de session

L'initiateur de la communication (site régional A de clé secrète E_a) demande au niveau (national) une clé de session pour établir un dialogue privé avec un autre site régional B (de clé secrète E_b)

Le niveau national connaît les clés E_a et E_b . Il détermine une clé de session E_s et la transmet cryptée par E_a et aussi cryptée par E_b à A

A commence sa session avec B en lui envoyant la clé de session cryptée par E_b (que A ne serait pas capable d'interpréter) mais que B va décoder.

A et b disposent d'une clé commune secrète E_s



III.2.5.2 2 *Distribution non hiérarchique des clés*

Pour faire communiquer secrètement des entités qui n'ont aucune référence hiérarchique commune (comment échanger des clés dans un réseau sans avoir de clé et sans pouvoir être surpris).

La méthode des puzzles

L'initiateur

- **Il transmet au destinataire un grand nombre de possibilités**
par exemple 20000 possibilités de choisir une clé (les puzzles)

- **Un puzzle c'est :**
un identifiant unique d'une clé (numéro) une clé

- **Les puzzles sont tous cryptés**
de façon à ce que ni le numéro de séquence
ni la clé ne soient en clair.

- **Les puzzles sont cryptés**
tous de façon différente
et pas trop difficile à casser en force brute.

Par exemple la clé employée (d'un DES) fait 56 bits dont 24 ont été fixés. Il reste 2^{32} configurations à essayer (quelques heures de calcul pour casser un puzzle).

- **Les puzzles sont transmis dans le désordre**

Le destinataire

- Il choisit une clé au hasard et il la casse.

- Il retransmet à l'initiateur en clair le numéro de séquence de la clé qu'il a retenu:

- L'initiateur qui connaît tous les puzzles et le destinataire qui l'a décrypté ont en commun une clé de session.

- Un intrus connaît le numéro de séquence de la clé mais il ne connaît pas sa valeur.

- Pour obtenir le même résultat il doit en moyenne essayer 10000 décryptages.

en force brute (10000 fois 3 heures)

III.2.6 La protection des clés: le partage d'un secret

- 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.

- 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 parts ($t+1$ par exemple)

. 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.

- La solution de Shamir 1978

. V valeur numérique entière

. On génère aléatoirement t valeurs entières

$$a_1, a_2, \dots, a_t$$

. On leur associe un polynôme dont le terme constant est V :

$$P(x) = a_t x^t + a_{t-1} x^{t-1} + \dots + a_1 x + V$$

. Une part du secret est un couple

$$(x_i, P(x_i)) \quad x_i \text{ non nul}$$

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

. on fait tous les calculs en arithmétique modulo n

Fonctionnement de la méthode

Tout groupe d'au moins $t+1$ possesseurs de parts

- Peut résoudre le système linéaire de détermination des coefficients du polynôme

- Et ainsi déterminer V

$$\begin{array}{c}
 (a_t, a_{t-1}, \dots, a_1, V) \\
 \cdot \\
 \cdot \\
 \cdot \\
 x_1 \\
 1
 \end{array}
 \begin{array}{cc}
 x_1^{**t} & x_{t+1}^{**t} \\
 \cdot & \cdot \\
 \cdot & \cdot \\
 \cdot & \cdot \\
 x_1 & x_t \\
 1 & 1
 \end{array}
 = (P(x_1), \dots, P(x_{t+1}))$$

- Comme les x_i sont différents et non nuls la matrice est régulière

- Tout sous groupe de porteurs en nombre inférieur ou égal à t ne peut déterminer t qu'à des constantes multiplicatives près.

Annexe

**Les normes de sécurité
des systèmes**

Le livre orange

A) Norme du département de la défense américain (DOD)

Norme T.C.S.E.C (1985)
Trusted Computer Systems Evaluation
Criteria

Plus connue sous le nom de "livre orange"
"Orange book"

La référence actuelle pour l'évaluation de la sécurité d'un système informatique par rapport à une liste de critères conduisant à un classement.

Quelques critères utilisés:

- *Existence d'une politique de sécurité* définie de façon explicite (en particulier précisant comment sont octroyés les droits d'accès)

- *Existence d'un classement* selon le niveau de sécurité des informations.

- *Garantie du mécanisme de sécurité* au même niveau que les informations qu'il est censé garantir.

- *Identification des personnes* ayant accès

- **Accès uniquement aux informations** pour lesquelles il est habilité.

- **Enregistrement de toute action** pouvant affecter la sécurité (pour analyse ultérieure)

- **Évolution indépendante** du matériel et du logiciel du point de vue de la sécurité.

Les quatre divisions du livre orange

Après l'identification et l'évaluation des mécanismes sécuritaires implantés dans le système classification du système informatiques selon sept niveaux principaux de sécurité (avec des divisions et des sous divisions)

Chaque niveau inclut les mécanismes de sécurité du niveau inférieur

Quelques caractéristiques de chaque niveau

Division D : Sécurité très faible ou nulle

En l'absence d'une possibilité d'évaluation réelle

Ces systèmes ne peuvent être classés ailleurs.

L'accès physique ouvre l'accès à toutes les ressources

Exemple : Le PC sous MS-DOS
Les macintosh sous MAC-OS
(à coup sur pour les versions antérieures à V7)

Division C : La division des systèmes répandus

Niveau C1 *Sécurité dite discrétionnaire* (Sécurité de base des systèmes actuels)

- Les usagers sont identifiés
- Authentification par mot de passe
- Chaque usager contrôle les ressources dont il est propriétaire en définissant les droits d'accès en lecture ou écriture des autres.

Exemple: Les systèmes UNIX standards

Niveau C2 *Protection d'accès contrôlé* (C1 avec de l'audit)

- Tous les objets manipulés par un usager ne doivent contenir que des données autorisées
- On doit tracer tout événement touchant à la sécurité (audit des intrusions)

Exemple : Certains UNIX

Division B Systèmes très bien protégés (en pratique peu de systèmes grand public)

Niveau B1 *Protection avec étiquettes de sécurité (labels)*

Introduction d'une sécurité à plusieurs niveaux distinguant selon les ressources différentes étiquettes de sécurité (secret, top secret, ...)

Ex : Certifié ou en phase de certification B1

- SUN OS
- GOUD-UTX / 32S
- System v / MLS

Niveau B2 *Protection structurée*

- Tous les objets ont obligatoirement un label

- Les périphériques doivent respecter les labels

- Il existe un modèle de sécurité objet d'une preuve de validité avec des tests de pénétration

Ex : TRUSTED XENIX

Unix system laboratories System V r4 avec module ES "enhanced security"

Niveau B3 *Les domaines de sécurité*

- Isolation des domaines de protection renforcée à l'aide de dispositifs matériels.

Division A La conception vérifiée

- Existence d'un modèle formel de sécurité
- Preuve mathématique formelle de la validité du modèle de sécurité
- Analyse formelle des canaux de communication (apparents ou cachés)
- Distribution de confiance des accès

Exemple : Le système SCOMP de Honeywell

B Norme ITCSEC

- Le livre blanc
- Version européenne de la norme américaine
- Peu diffusée
- Postérieure à la norme américaine > corrige certains points