Sécurité des Systèmes d'Information

Eric Alata	eric.alata@laas.fr
Yves Deswarte	yves.deswarte@laas.fr
Vincent Nicomette	vincent.nicomette@laas.fr
Benoît Morgan	benoit.morgan@enseeiht.fr

INSA de Toulouse - ENSEEIHT

13 septembre 2022

Premières définitions

Système d'information

Un système d'information est l'ensemble des éléments participant au traitement, à la gestion et à la transmission d'informations entre les membres d'une communauté.

Sécurité

La sécurité des systèmes d'information est l'ensemble des moyens permettant d'assurer les propriétés de confidentialité, d'intégrité et de disponibilité des informations.

Les propriétés de la sécurité

Les attaques

Les défenses

La protection des systèmes informatiques

Sommaire

Sommaire

La sûreté de fonctionnement informatique

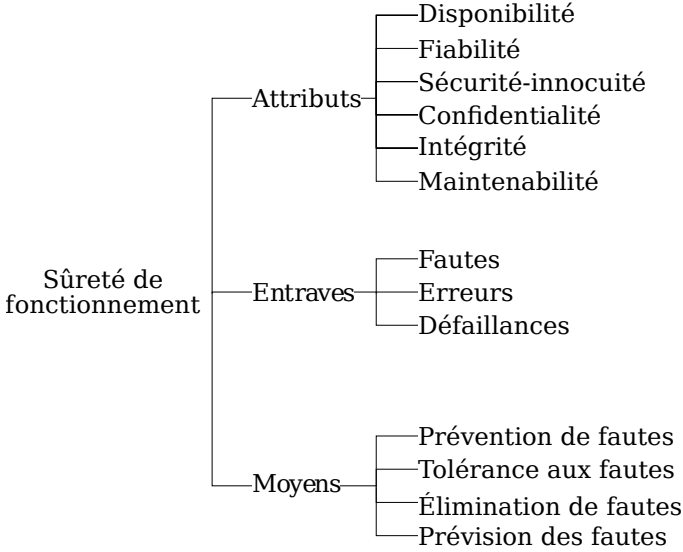
Définition [7]

La sûreté de fonctionnement d'un système informatique est la propriété qui permet de placer une confiance justifiée dans le service qu'il délivre

Attributs [8]

- le fait d'être prêt à l'utilisation conduit à la **disponibilité**
- la continuité de service conduit à la **fiabilité**
- la non-occurrence de conséquences catastrophiques conduit à la **sécurité-innocuité**
- la non-occurrence de divulgations non-autorisées de l'information conduit à la **confidentialité**
- la non-occurrence d'altérations inappropriées du système conduit à l'**intégrité**
- l'aptitude aux réparations et aux évolutions conduit à la **maintenabilité**

La sûreté de fonctionnement informatique



Entraves à la sûreté de fonctionnement

- Une **défaillance** survient lorsque le service délivré dévie de l'accomplissement de la fonction du système
- Une **erreur** est la partie de l'état du système qui est susceptible d'entraîner une défaillance
- Une **faute** est la cause adjugée ou supposée d'une erreur

Chaîne fondamentale

. . .faute → erreur → défaillance → faute → . . .

Les moyens pour la sûreté de fonctionnement informatique

Éviter les fautes

Prévention des fautes

Comment empêcher que des fautes surviennent ou soient introduites

Élimination des fautes

Comment réduire la présence (en nombre ou en gravité) des fautes

Accepter les fautes

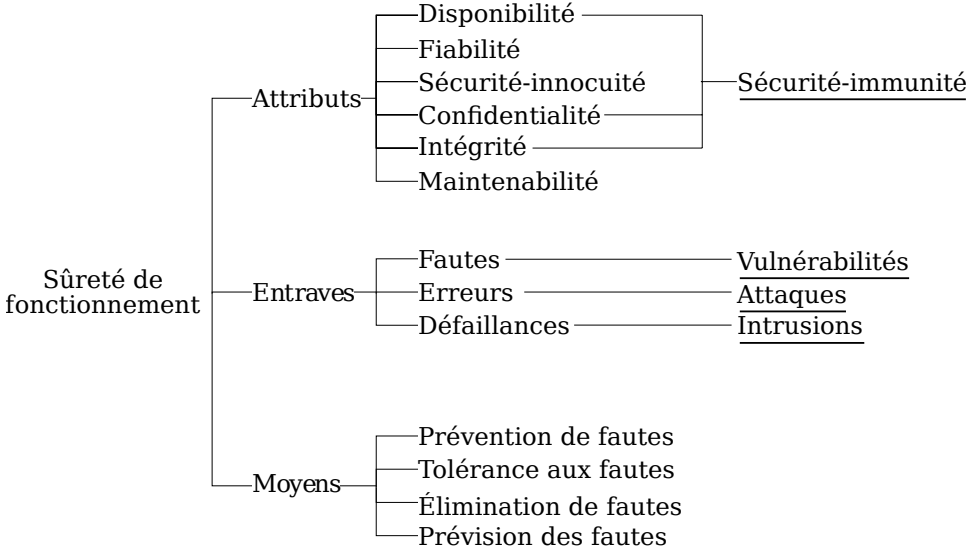
Tolérance aux fautes

Comment fournir un service conforme à la fonction en dépit des fautes

Prévision des fautes

Comment estimer la présence, la création et les conséquences des fautes

La sûreté de sécurité-immunité



Définition 1/2

Definition

Sécurité(-immunité) = confidentialité + intégrité + disponibilité
Vis-à-vis des fautes intentionnelles dites malveillances
Malveillances = logiques malignes + intrusions

- Perte de confidentialité = divulgation non autorisée d'information
- Perte d'intégrité = altération non autorisée de l'information
- Perte de disponibilité = incapacité d'un système à être prêt à l'utilisation

vis-à-vis de malveillances

Définition 2/2

For most distributed systems, the security objectives of confidentiality, integrity, and availability of information apply. A loss of confidentiality is the unauthorized disclosure of information. A loss of integrity is the unauthorized modification or destruction of information. A loss of availability is the disruption of access to or use of information or an information system.[1]

Entraves à la sécurité-immunité

- **Une attaque** est une faute d'interaction externe au système, dont le but est de violer un ou plusieurs des attributs de sécurité. Elle peut être aussi définie comme une tentative d'intrusion.
- **Une vulnérabilité** est une faute qui peut être accidentelle, intentionnelle malveillante ou non malveillante placée dans les exigences, la spécification, la conception ou la configuration du système, ou dans la manière dont il est utilisé.
- Une vulnérabilité peut être exploitée avec une attaque pour créer une **intrusion**. Une intrusion est donc une faute malveillante, initiée depuis l'extérieur pendant l'utilisation du système.

Chaîne fondamentale

. . .vulnérabilité → attaque → intrusion → vulnérabilité → . . .

Les moyens pour la sécurité-immunité

Éviter les fautes intentionnelles

Prévention des fautes

Prévention des vulnérabilités ; prévention des attaques ; prévention d'intrusion

Élimination des fautes

Élimination des vulnérabilités

Accepter les fautes intentionnelles

Tolérance aux fautes

Tolérance aux intrusions

Prévision des fautes

Prévisions des vulnérabilités ; prévision des attaques ; prévision des intrusions

L'information

Definition

Une information est composée de données et méta-données.

- Données : captées ou générées, traitées, stockées, transmises, affichées
- Méta-données : créées et utilisées par les services sous-jacents

Une méta-donnée est une donnée à un niveau inférieur

Autres propriétés

Anonymat

Confidentialité de (identité de l'utilisateur)

Protection de la vie privée

Confidentialité de (identité de l'utilisateur + données personnelles)

Authenticité d'un message

Intégrité de (contenu + identité de l'émetteur + date + . . .)

Authenticité d'un document

Intégrité de (contenu + identité du créateur + date + . . .)

Authenticité d'un utilisateur

Intégrité de (identité)

Autres propriétés

Imputabilité

Disponibilité de (qui + quoi + quand + où + . . .) d'une action

Non-répudiation d'origine

Disponibilité de (identité de l'émetteur + . . .) +
intégrité du (contenu)

Non-répudiation de réception

Disponibilité de (identité du récepteur + . . .) +
intégrité du (contenu)

Protection de la propritété intellectuelle

Confidentialité de (contenu) +
intégrité du (contenant)

Besoins de sécurité selon les secteurs

- Défense, gouvernement
  Confidentialité ≫ intégrité, disponibilité
- Finance
  Intégrité ≫ disponibilité > confidentialité
- Autres (industrie, administrations, médecine, ...)
  Ça dépend

⇒ Besoin de définir les spécificité de l'application
⇒ Politique de sécurité

Sommaire

Sommaire

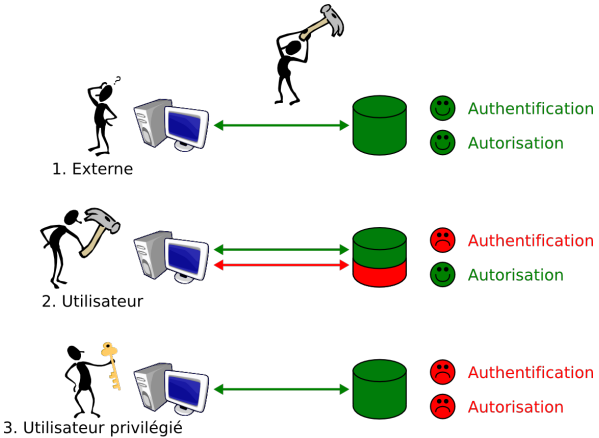
Sommaire

Les attaquants et leurs motivations

- Jeu : explorer les limites, éprouver et étendre ses connaissances, découvrir de nouvelles failles, améliorer la sécurité : "hackers"
- Emulation, sectarisme : groupe de hackers : "exploits"
- Vandalisme : montrer sa force, punir : "web defacing", virus, vers, . . .
- Politique, idéologie : ex. CCC, 600 sites danois "défigurés" en février 2006
- Vengeance : ex. SCORES
- Profit : espionnage, extorsion de fonds : concurrence déloyale, crime organisé, espionnage international (attaques probablement chinoises contre des sites gouvernementaux des USA, GB, Allemagne, France, . . .)
- Guerre informatique, terrorisme : 2007 DDoS contre des sites estoniens, 2008 contre des sites georgiens, . . .
- Sensibilisation, lobbying
- Protection abusive : ex. SONY

Les attaquants et leurs motivations

Qui sont les “intrus” ?

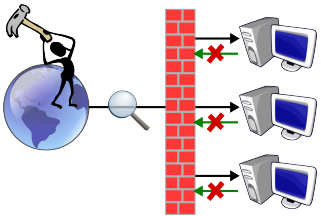


80% des fraudes sont "autorisées"

Les attaquants et leurs motivations

- Organisation
  - Seul ? Groupe ?
- Compétence
  - Novice ? Averti ? Expert ?
- Comportement
  - Discret ? Ostensible ?

⇒ Utilisation de “pots de miel”



Les attaquants et leurs motivations

Vidéo x2

Sommaire

Classification des attaques – Générale

- **Interception** (intégrité) : modification d'informations transmises
- **Cryptanalyse** (confidentialité) : obtenir des informations secrètes (messages en clair, clés, algorithme de chiffrement) à partir des informations publiques (cryptogrammes)

  Collisions dans MD5 en 2004 [15]
- **Répudiation** (intégrité) : refuser de reconnaître une opération qu'on a effectuée (répudiation d'origine, de réception)
- **Déduction par inférence, furetage** (confidentialité) : obtenir des informations secrètes (par exemple, des données personnelles) à partir des informations auxquelles on a accès (par exemple, statistiques)

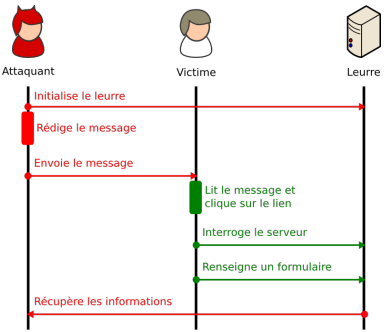
Classification des attaques – Générale

- **Déguisement,** *masquerade* (intégrité) : se faire passer pour quelqu'un d'autre (tromper l'authentification, s'il y en a ...)
- **Canaux cachés** *covert channels* (confidentialité) : communiquer par des moyens non-surveillés
- **Canaux de fuite** *side channels* (confidentialité) : obtenir des informations cachées de façon détournée

Classification des attaques – Ingénierie sociale

- **Déguisement* – phishing** :
  obtenir des renseignements personnels (hameçonnage de mots de passe)
  Téléphone [14], courriel, blog, news, IRC, MSN Messenger, SMS [17], . . .
  Thèmes utilisés liés à l'actualité, pour marquer plus les esprits
  Statistiques nombreuses

Track Usenix dédié



Classification des attaques – Ingénierie sociale

Techniques pour cacher la véritable url malveillante : *Open Redirect*, Pharming, DNS poisoning, attaques homographes, attaques sur les erreurs typographiques

```
http://cgi4.ebay.com/ws/eBayISAPI.dll?
MfcISAPICommand=RedirectToDomain&
DomainUrl=http%3A%2F%2F%32%31%31%2E%31%37
%32%2E%39%36%2E%37%2FUpdateCenter
%2FLogin%2F%3FMfcISAPISession%3DAAJbaQqze
HAAeMWZlHhlWXS2AlBXVShqAhQRfhgTDrferHCUR
stpAisNRqAhQRfhgTDrferHCURstpAisNRpAisNR
qAhQRfhgTDrferHCUQRfqzeHAAeMWZlHhlWXh
```

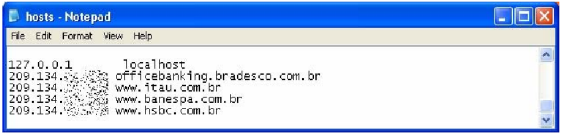
http://ryanair.com ≠ http://ryamair.com

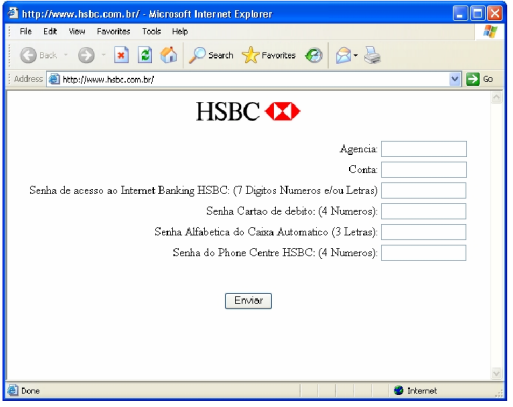
Bing.com

Bing.com

Bing.com

Classification des attaques – Ingénierie sociale





Classification des attaques – Ingénierie sociale

From remboursement@impots.gouv.fr Mon Oct 5 09:17:46 2009
Return-Path: <remboursement@impots.gouv.fr>
Reply-To: <remboursement@impots.gouv.fr>
From: "L'administration Fiscale"<remboursement@impots.gouv.fr>
Subject: Notification d'impôt
Date: Mon, 5 Oct 2009 02:55:50 -0400
Content-Type: text/html; charset="Windows-1251"
X-Spam-Status: Yes
X-Spam-Score: 9.706 (*********)

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

DIRECTION GENERALE DES FINANCES PUBLIQUES 05/10/2009

Notification d'impôt - Remboursement

Après les derniers calculs annuels de l'exercice de votre activité, nous avons déterminé que vous êtes admissible à recevoir un remboursement d'impôt de € 178,80.

S'il vous plaît soumettre la demande de remboursement d'impôt et nous permettre de 10 jours ouvrables pour le traitement.

Pour accéder au formulaire pour votre remboursement d'impôt, cliquez ici

Un remboursement peut être retardé pour diverses raisons. Par exemple la soumission des dossiers non valides ou inscrivez après la date limite.

Le Conciliateur fiscal adjoint

Philippe BERGER

Classification des attaques – Ingénierie sociale

- **Déguisement* – Scam** (Fraude 4-1-9) : escroquerie
  419 : article du code nigérian sanctionnant ce type de fraude

Date: Thu, 24 Sep 2009 11:31:29 -0700 (PDT)
From: Linda Fastus <ms_lindafastus@info2link.biz>
Subject: FROM MRS LINDA FASTUS SCHWARZ
X-Spam-Status: Yes
X-Spam-Score: 12.088 (**********)

FROM MRS LINDA FASTUS SCHWARZ
ABIDJAN COTE D'IVOIRE
PLEASE EMAIL BACK

DEAREST IN CHRIST,

KNOW THAT THIS MAIL MAY REACH YOU BY SURPRISE AS WE DONT KNOW OURSELF PREVIOUSLY.I AM THE ABOVE NAME PERSON FROM INDIA. I AM MARRIED TO MR FASTUS SCHWARZ ,WHO WAS THE AMBASSADOR OF JAMAICA FOR NINETEN YEARS IN COTE DIVOIRE.WE WERE MARRIED FOR FIFTEN YEARS WITHOUT A CHILD. HE DIED IN DECEMBER 27TH 2004 AFTER A BRIEF ILLNESS THAT LASTED FOR ONLY TWO WEEKS

BEFORE HIS DEATH WE ARE HAPPY HUSBAND AND WIFE CHRISTIAN FAMILY. SINCE HIS DEATH IDECIDED NOT TO REMARRY OR GET A CHILD OUTSIDE MY MATRIMONIAL HOME WHICH THE BIBLE IS AGAINST. WHEN MY LATE HUSBAND WAS ALIVE, HE DEPOSITED THE SUM OF (USD $12.7MILLION) TWELVE MILLION SEVEN HUNDRED THOUSAND U.S.DOLLARS INTO A BOX FOR SECURITY REASON AND THE MONEY STILL WITH THE SECURITY COMPANY HERE IN ABIDJAN COTE D'IVOIRE.

MEANWHILE, I HAVE NOT TELL ANY BODY THE CONTENT OF THIS DEPOSIT IN THE SECURITY COMPANY.I AM TELLING YOU THE CONTENT REASON THAT I WANT YOU TO ASSIST ME USE THE FUND FOR THE WORK OF GOD. EVEN DO YOU ARE NOT A CHRISTAIN.THAT IS NOT A PROBLEM .WHAT I WANT IS FOR YOU TO USE IT AND HELP THE HELPLESS PEOPLE AROUND YOU ,TO HELP THE ORPHANAGES, WIDOWS, AND MOTHERLES CHILDRENS

RECENTLY, MY DOCTOR TOLD ME THAT I HAVE SERIOUS SICKNESS WHICH IS CADIAC PROBLEM.THE ONE THAT DISTURBS ME MOST IS MY STROKE SICKNESS HAVING KNOWN MY CONDITION I DECIDED TO DONATE HIS FUND TO YOU TO UTILIZE THIS MONEY ACCORDING TO MY DIRECTION AND THE WILL OF GOD.

PLEASE DO GIVE URGENT RESPONSE TO THIS MAIL WITHOUT ANY DELAY.

I WANT TO GIVE YOU NUMBER TO CALL ME BUT I DONT WANT IN A WAY MY HUSBAND RELATIONS WILL KNOW THAT I AM GIVING YOU THIS MONEY. I HAVE SISTER NURSE WHO IS FEARFUL TO THE LORD THAT WILL BE HELPING GIVING YOU INFORMATION OF THIS DEPOSIT.HER NAME ID SISTER (CHANTAL KONE)

SO PLEASE I AM WAITING FOR YOUR URGENT REPLY SO THAT I CAN GIVE YOU ALL THE INFORMATION ABOUT THIS MONEY AND THE SECURITY COMPANY WERE IT WAS DEPOSITED BY MY LATE HUSBAND ,

REMAIN BLESSED ALWAYS
YOURS SISTER IN CHRIST
MRS LINDA FASTUS SCHWARZ

Classification des attaques – Ingénierie sociale

- **Déguisement*** – arnaque au président : escroquerie aux faux ordres de virement (FOVI)
  Police nationale

Vidéo x2

Classification des attaques – Matérielles

- **Écoute passive** (confidentialité) : accès sans modification à des informations générées, transmises, stockées ou affichées sur des composants vulnérables

  sniffing, *snooping*, *eavesdroping*, *wiretapping*, réutilisation de mémoire (buffers, fichiers temporaires, supports magnétiques), analyse de trafic, effet Van Eck (*Van Eck phreaking* – TEMPEST), *key logger*, . . .

  Câbles de claviers
  Câbles de contrôlleurs d'affichage

Effet Van Heck

Classification des attaques – Matérielles

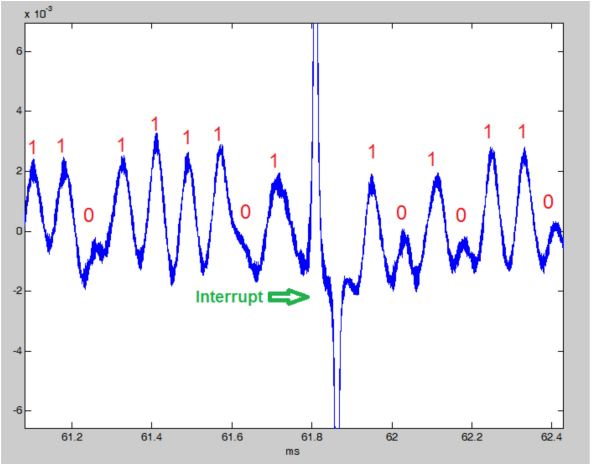
Vidéo x2

Classification des attaques – Matérielles

- **Canaux de fuite***
  - Exemples avec les cartes à puce : analyse de la consommation de courant d'alimentation (simple SPA, différentielle DPA)
  - Captation : microscope à balayage, micro-sondes, rayonnement électromagnétique, ...
  - Injection de faute : micro-sondes, impulsions électro-magnétiques (y compris lumineuses), rayonnement nucléaire, ...

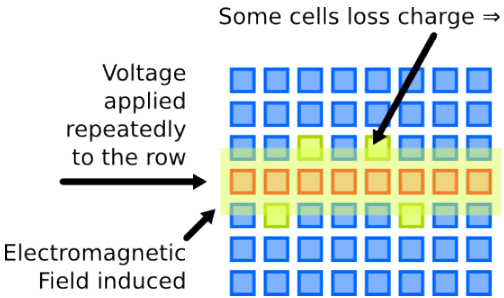
Classification des attaques – Matérielles

Get Your Hands Off My Laptop :
Physical Side-Channel Key-Extraction Attacks on PCs [5]



Classification des attaques – Matérielles

Row-hammer : exploitation du couplage dans les structures DRAM
Implementation en javascript [6]




```
code1a:
        // read from
        // address X
        mov (X), %eax
        // read from
        // address Y
        mov (Y), %ebx
        // flush cache
        / for address X
        clflush (X)
        // flush cache
        // for address Y
        clflush (Y)
        jmp code1a
```

Classification des attaques – Matérielles

SonarSnoop : active acoustic side-channel attacks [3]

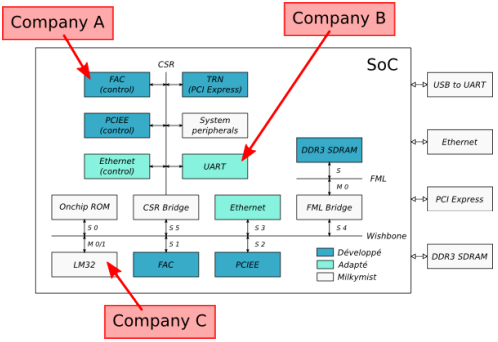
Lamphone : Real-Time Passive Sound Recovery from Light Bulb Vibrations [11]

Vidéo

Classification des attaques – Matérielles

- **Porte dérobée** – *Trapdoor* / *Backdoor* (confidentialité, intégrité, disponibilité) : contourner les mécanismes de protection

Hardware Trojan : Threats and emerging solutions [2]



Accusations d'espionnage contre Huawei

Classification des attaques – Matérielles

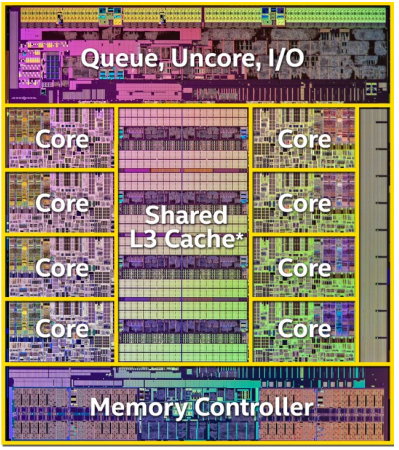
Trojan basé sur un *Don't care* bit

```
entity crypto_alu is
port (clk : in std_logic;
      a, key : in signed(7 downto 0);
      op : in unsigned(1 downto 0);
      r : out signed(7 downto 0));
end crypto_alu;
architecture Behavioral of crypto_alu is begin
process(clk) begin
    if (rising_edge(clk)) then
        if op = "00" then r <= a ^ key else
        if op = "01" then r <= a else
        if op = "10" then r <= ~ a else
        -- if op = "11" then r <= key else -- trojan!
        r <= "zzzzzzzz";
    end if;
end process;
end Behavioral;
```

Classification des attaques – Bas-niveau

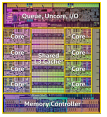
- **Canaux cachés*** et **Canaux de fuite***

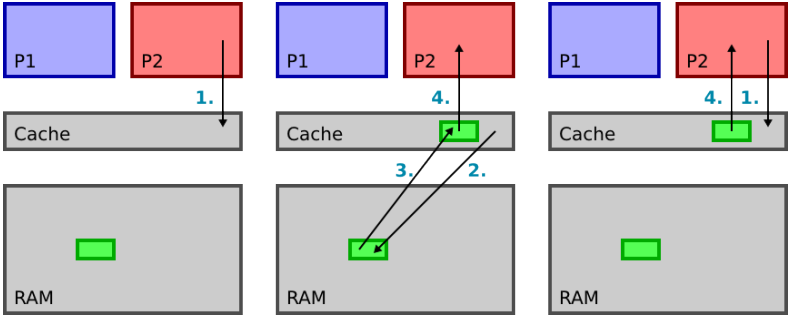
Cache missing for fun and profit [12]



Classification des attaques – Bas-niveau

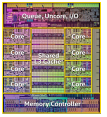
- Cache attack
- Principe de déduplication
- Implementation dans le Cloud (virtualization)

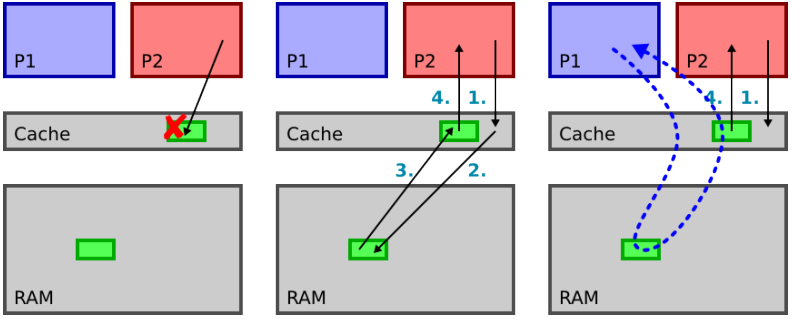




Classification des attaques – Bas-niveau

- Cache attack
- Principe de déduplication
- Implementation dans le Cloud (virtualization)





Classification des attaques – Bas-niveau

Démo Canal de communication caché ?

Classification des attaques – Réseau

- **Canal caché***
- **Déguisement***
  - *WiPhishing* : hotspots WiFi ouverts
  - *IP spoofing* (contre-mesure : *ingress filtering*)
  - *DNS poisoning* : URL ↔ @IP
  - *ARP poisoning* : @MAC ↔ @IP
  - *Pharming*

Classification des attaques – Réseau

- **Déni de service** – *DoS* : *denial of service* (disponibilité) : empêcher les utilisateurs légitimes d'accéder aux informations ou aux services auxquels ils ont droit
  - *flooding*, *smurfing* (*ICMP echo requests*), DDoS (par *botnets*)
    - Ver de Morris (novembre 1988)
    - DDoS – *The February 2018 GitHub DDoS attack*
    - DDoS – *The 2016 Dyn attack*
    - DDoS – *The 2007 Estonia attack*

Classification des attaques – Réseau

- **Interception*** – *Man-in-the-middle*

Présentation + Vidéo

Classification des attaques – Logiciel

- **Logiciels malveillants** (confidentialité, intégrité, disponibilité) : (*malware* / maliciels : *rootkits*, *zombies*, . . .)
  - Furtivité (*stealth*)
  - Escalade de privilèges (jusqu'à *root*)
  - Installation de portes dérobées, de bombes logiques, de *spyware*, . . .
- **Porte dérobée** – *Trapdoor* / *Backdoor* (confidentialité, intégrité, disponibilité) : contourner les mécanismes de protection
  - Authentification (Turing Award de Ken Thompson), autorisation
  - Exemple : *le nid du coucou*, Clifford Stoll, 1986
  - *Rootkits*
    - Utilisation d'une porte dérobée pour devenir *root* (escalade de privilège)
    - Modification du noyau, appels systèmes ou commandes (ps, w, netstat, . . .)
    - Installation d'une porte dérobée pour un accès plus facile (ex. à distance)
    - Installation de logiciels malveillants, invisibles au niveau utilisateur

Classification des attaques – Logiciel

- **Bombe logique** (confidentialité, intégrité, disponibilité) : déclencher des dégâts sur un événement particulier
- **Cheval de Troie** (confidentialité, intégrité, disponibilité) : fonction illicite cachée dans un programme apparemment bénin
- **Virus** (confidentialité, intégrité, disponibilité) : segment de code qui, lorsqu'il est exécuté, se reproduit en s'attachant à un autre programme (système ou application), éventuellement porteur d'une bombe logique
- **Ver** – *worm* (confidentialité, intégrité, disponibilité) : programme autonome, capable de se répliquer et de se propager, éventuellement porteur d'une bombe logique

Classification des attaques – Logiciel

Vers MIRAI

Code source – Morris

```
// Set up passwords
add_auth_entry("\x50\x4D\x4D\x56", "\x5A\x41\x11\x17\x13\x13", 10);               // root     xc3511
add_auth_entry("\x50\x4D\x4D\x56", "\x54\x4B\x58\x5A\x54", 9);                    // root     vizxv
add_auth_entry("\x50\x4D\x4D\x56", "\x43\x46\x4F\x4B\x4C", 8);                    // root     admin
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x43\x46\x4F\x4B\x4C", 7);                // admin    admin
add_auth_entry("\x50\x4D\x4D\x56", "\x1A\x1A\x1A\x1A\x1A\x1A", 6);                // root     888888
add_auth_entry("\x50\x4D\x4D\x56", "\x5A\x4F\x4A\x46\x4B\x52\x41", 5);            // root     xmhdipc
add_auth_entry("\x50\x4D\x4D\x56", "\x46\x47\x44\x43\x57\x4E\x56", 5);            // root     default
add_auth_entry("\x50\x4D\x4D\x56", "\x48\x57\x43\x4C\x56\x47\x41\x4A", 5);        // root     juantech
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x10\x11\x16\x17\x14", 5);                // root     123456
add_auth_entry("\x50\x4D\x4D\x56", "\x17\x16\x11\x10\x13", 5);                    // root     54321
add_auth_entry("\x51\x57\x52\x52\x4D\x50\x56", "\x51\x57\x52\x52\x4D\x50\x56", 5);// support  support
add_auth_entry("\x50\x4D\x4D\x56", "", 4);                                        // root     (none)
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x52\x43\x51\x51\x55\x4D\x50\x46", 4);    // admin    password
add_auth_entry("\x50\x4D\x4D\x56", "\x50\x4D\x4D\x56", 4);                        // root     root
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x10\x11\x16\x17", 4);                    // root     12345
add_auth_entry("\x57\x51\x47\x50", "\x57\x51\x47\x50", 3);                        // user     user
add_auth_entry("\x43\x46\x4F\x4B\x4C", "", 3);                                    // admin    (none)
add_auth_entry("\x50\x4D\x4D\x56", "\x52\x43\x51\x51", 3);                        // root     pass
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x43\x46\x4F\x4B\x4C\x13\x10\x11\x16", 3);// admin    admin1234
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x13\x13\x13", 3);                        // root     1111
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x51\x4F\x41\x43\x46\x4F\x4B\x4C", 3);    // admin    smcadmin
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x13\x13\x13\x13", 2);                    // admin    1111
add_auth_entry("\x50\x4D\x4D\x56", "\x14\x14\x14\x14\x14\x14", 2);                // root     666666
```

Classification des attaques – Logiciel

Username/Password	Manufacturer	Link to supporting evidence
admin/123456	ACTi IP Camera	https://ipvm.com/reports/ip-cameras-default-passwords-directory
root/anko	ANKO Products DVR	http://www.cctvforum.com/viewtopic.php?f=3&t=44250
root/pass	Axis IP Camera, et. al	http://www.cleancss.com/router-default/Axis/0543-001
root/vizxv	Dahua Camera	http://www.cam-it.org/index.php?topic=5192.0
root/888888	Dahua DVR	http://www.cam-it.org/index.php?topic=5035.0
root/666666	Dahua DVR	http://www.cam-it.org/index.php?topic=5035.0
root/7ujMko0vizxv	Dahua IP Camera	http://www.cam-it.org/index.php?topic=9396.0
root/7ujMko0admin	Dahua IP Camera	http://www.cam-it.org/index.php?topic=9396.0
666666/666666	Dahua IP Camera	http://www.cleancss.com/router-default/Dahua/DH-IPC-HDW4300C
root/dreambox	Dreambox TV receiver	https://www.satellites.co.uk/forums/threads/reset-root-password-plugin.101146/
root/ztlxx	EV ZLX Two-way Speaker?	?
root/juantech	Guangzhou Juan Optical	https://news.ycombinator.com/item?id=11114012
root/xc3511	H.264 - Chinese DVR	http://www.cctvforum.com/viewtopic.php?f=56&t=34930&start=15
root/hi3518	HiSilicon IP Camera	https://acassis.wordpress.com/2014/08/10/i-got-a-new-hi3518-ip-camera-modules/
root/klv123	HiSilicon IP Camera	https://gist.github.com/gabonator/74cdd6ab4f733ff047356198c781f27d
root/klv1234	HiSilicon IP Camera	https://gist.github.com/gabonator/74cdd6ab4f733ff047356198c781f27d
root/jvbzd	HiSilicon IP Camera	https://gist.github.com/gabonator/74cdd6ab4f733ff047356198c781f27d
root/admin	IPX-DDK Network Camera	http://www.ipxinc.com/products/cameras-and-video-servers/network-cameras/
root/system	IQinVision Cameras, et. al	https://ipvm.com/reports/ip-cameras-default-passwords-directory
admin/meinsm	Mobotix Network Camera	http://www.forum.use-ip.co.uk/threads/mobotix-default-password.76/
root/54321	Packet8 VOIP Phone, et. al	http://webcache.googleusercontent.com/search?q=cache:W1phozQZURUJ:community.freepbx.org/t/packet8-atas-phones/411
root/00000000	Panasonic Printer	https://www.experts-exchange.com/questions/26194395/Default-User-Password-for-Panasonic-DP-C405-Web-Interface.html
root/realtek	RealTek Routers	
admin/1111111	Samsung IP Camera	https://ipvm.com/reports/ip-cameras-default-passwords-directory
root/xmhdipc	Shenzhen Anran Security Camera	https://www.amazon.com/MegaPixel-Wireless-Network-Surveillance-Camera/product-reviews/B00EB6FNDI
admin/smcadmin	SMC Routers	http://www.cleancss.com/router-default/SMC/ROUTER
root/ikwb	Toshiba Network Camera	http://faq.surveillixdvrsupport.com/index.php?action=artikel&cat=4&id=8&artlang=en
ubnt/ubnt	Ubiquiti AirOS Router	http://setuprouter.com/router/ubiquiti/airos-airgrid-m5hp/login.htm
supervisor/supervisor	VideoIQ	https://ipvm.com/reports/ip-cameras-default-passwords-directory
root/<none>	Vivotek IP Camera	https://ipvm.com/reports/ip-cameras-default-passwords-directory
admin/1111	Xerox printers, et. al	https://atyourservice.blogs.xerox.com/2012/08/28/logging-in-as-system-administrator-on-your-xerox-printer/
root/Zte521	ZTE Router	http://www.ironbugs.com/2016/02/hack-and-patch-your-zte-f660-routers.html

https ://krebsonsecurity.com/2016/10/who-makes-the-iot-things-under-attack

Classification des attaques – Logiciel

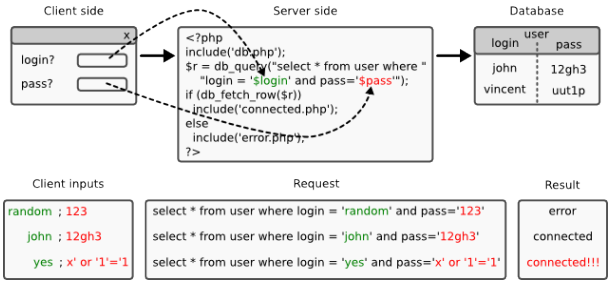
- **Race condition** (intégrité)

Démo

Classification des attaques – Web

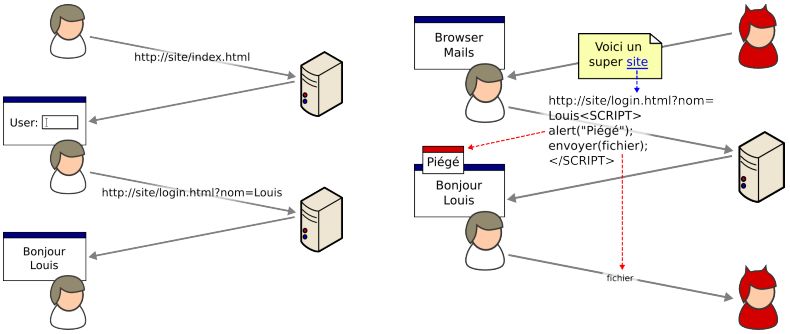
- **Injection** – SQL

  - Envoi de données *malveillante*, mal vérifiées par server 1, pour changer la sémantique de la requête au serveur 2
  - XSS, Sql Injection, CSRF, NoSql Injection, Xpath injection, ...



Classification des attaques – Web

- **Injection** – XSS



`www.cert.org/advisories/CA-2000-02.html`

- Le pirate peut communiquer le lien soit par *phishing* soit indirectement via, par exemple, un *blog* ou *forum* d'un serveur innocent
- La victime lit le message avec un navigateur configuré pour permettre l'exécution de scripts

Classification des attaques

Cette classification est incomplète !

- RFID zapper
- Spyware
- Spamming
- Page web minées
- botnets
- ML abus
- vehicules autonomes

Classification des attaques

Gains financiers pour les pirates qui contrôlent un ordinateur

- Utilisation de numéros de cartes de crédit
- Chantage, extorsion de fonds, espionnage industriel, etc.
- Spéculation en bourse : *pump and dump scams* (*spam*, VoIP), exemple : `www.investopedia.com/ask/answers/05/061205.asp`
- Connexion à des lignes téléphoniques payantes
- Accès à des comptes (banques, retraites, paypal, e-Bay, FAI, opérateurs téléphoniques, hotspots, etc.)
- Vente d'adresses e-mails
- Services payants (exemples : porno, films piratés, etc.) + spammers, etc.
- *Click fraud* (relais de publicité)
- Location de botnets, spammers, etc.

Classification des attaques

Les principales failles exploitables :

- API : débordement de buffers, heap, stack, entiers, etc.
- API : contrôle ou vérification insuffisants (ex. injection)
- Utilisation non-prévue → Fuzzing
- Race conditions
- Contrôle d'origine insuffisant (ex. applets, plug-ins, certificats, etc.)

`cwe.mitre.org/top25`

Sommaire

Sommaire

Différents volets de la sécurité

- Sécurité physique
  Protection des locaux contre incendie, inondation, etc.
  Contrôle des accès physiques
- Sécurité du personnel (pas la CHS)
  Règles liées aux conditions de travail pour les personnels internes (employés, intérimaires, stagiaires, etc.) et externes (visiteurs, maintenance, sous-traitants, etc.), y compris dans des circonstances particulières : embauche, départ, grève, etc.

Différents volets de la sécurité

- Sécurité procédurale
  Procédures pour la gestion du SIC : enregistrement (et effacement) des utilisateurs, sauvegardes, maintenance, installation et mises à jour de matériels et de logiciels, etc.
- Sécurité technique
  C'est tout ce qu'on va voir maintenant.

Sommaire

Terminologie

- Cryptologie = cryptographie + cryptanalyse
  - Cryptographie, du grec *kruptos* (caché) et *graphein* (écrire)
    Ecrire des messages incompréhensibles par des tiers
  - Cryptanalyse
    Découvrir le(s) secret(s), décrypter
- A ne pas confondre avec stéganographie
  - Du grec *stegano* (dissimuler)
  - Encre sympathique
  - Filigranes (tatouages)
- Chiffre, chiffrement (pas chiffrage ni cryptage), déchiffrement, clair, cryptogramme

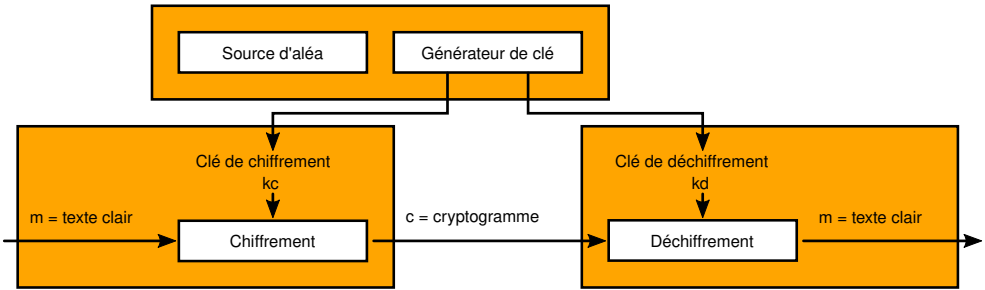
Propriétés couvertes par la cryptographie

- **Confidentialité** de l'information
  Exemple : écoute passive
- **Integrité** / **authenticité** de l'information
  Exemple : homme dans le milieu
- **Authentification** des entités
  Exemple : déguisement
- **Non-prépudiation** d'origine et de destination
  Exemple : preuves, matériel juridique

Définition fondamentales et notations

- **Clair**, $m \in M$: message non chiffré, l'information est accessible
- **Chiffré**, $c \in C$: message chiffré ou cryptogramme, l'information n'est pas accessible
- **Clé**, $k \in K$: secret indispensable pour transformer un clair en chiffré ou un chiffré en clair. On parle respectivement de clé de chiffrement et de clé de déchiffrement
- **Générateur de clé** : génération des clés
- **Chiffrement** $\{\}$ ou $E()$: transformation d'un clair en chiffré pour une clé de chiffrement donnée
- **Déchiffrement** $[]$ ou $D()$: transformation d'un chiffré en clair pour une clé de déchiffrement donnée

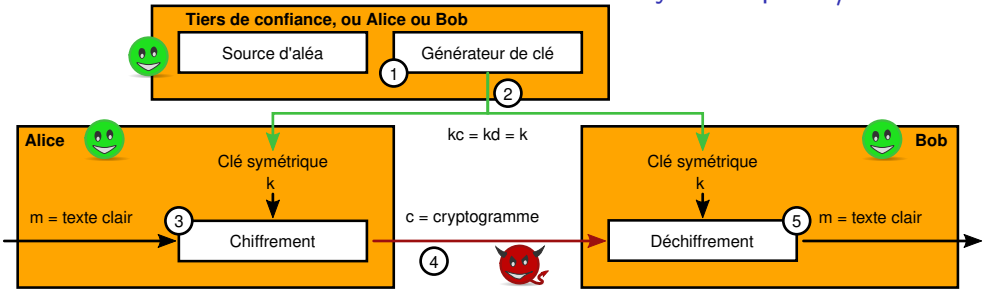
Constructions fondamentales : chiffrement



Notation

- Chiffrement : $C = \{M\}_{k_c}$ ou $C = E_{k_c}(M)$
- Déchiffrement : $M = [C]_{k_d}$ ou $M = D_{k_d}(C)$

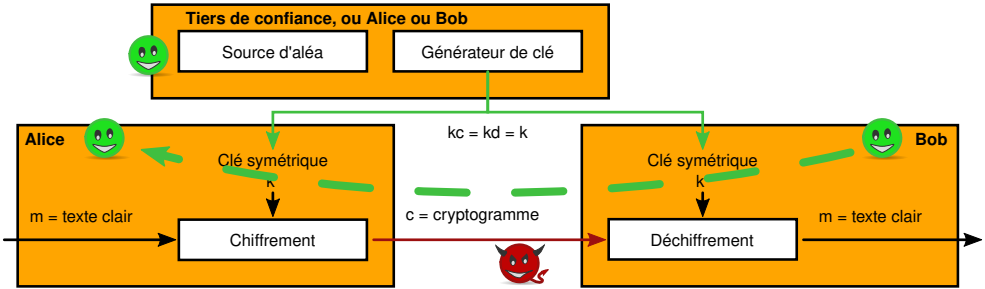
Constructions fondamentales : chiffrement symétrique 1/2



Procédure

1. Alice ou Bob génère une clé secrète unique : K
2. Distribution de la clé à l'aide d'un canal sécurisé
3. Alice chiffre le message avec la clé secrète K
4. Le message est transmis au travers d'un canal non sécurisé
5. Bob déchiffre le message avec la clé secrète K

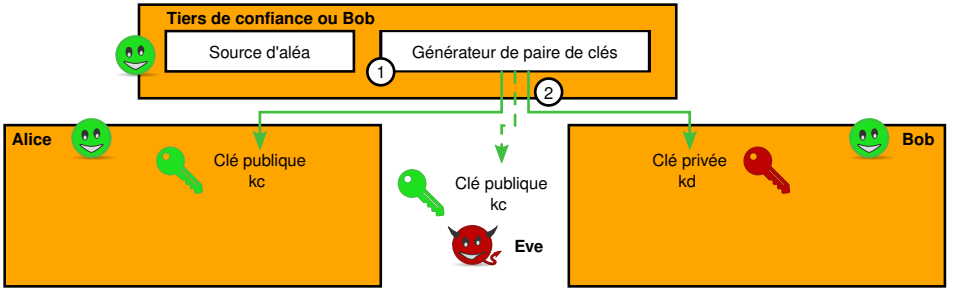
Constructions fondamentales : chiffrement symétrique 2/2



Propriétés

- $k_c = k_d = K$
- Authentification de l'origine
- M confidentiel

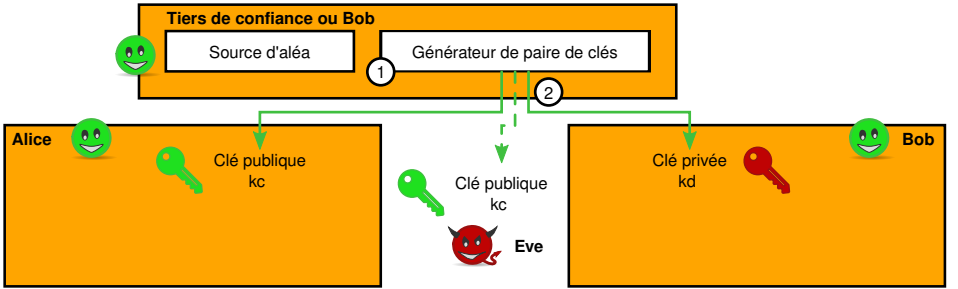
Constructions fondamentales : chiffrement asymétrique 1/6



Procédure : distribution des clés

1. Bob génère une paire de clés unique : (k_c, k_d)
2. Distribution de k_d à Bob l'aide d'un canal sécurisé
3. Distribution de k_c à Alice et au monde

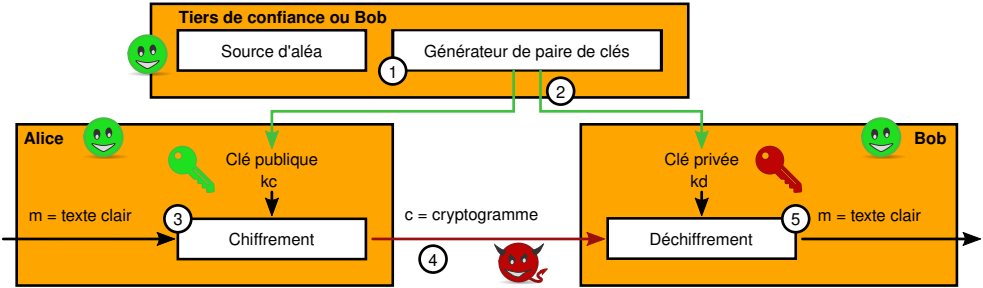
Constructions fondamentales : chiffrement asymétrique 2/6



Propriétés chiffrement asymétrique

- $k_c \neq k_d$
- $\exists$ une unique paire $(k_c, k_d) | M = D_{k_d}(E_{k_c}(M))$
- k_c est connue à la fois d'Alice et Bob, mais aussi de l'attaquant Eve

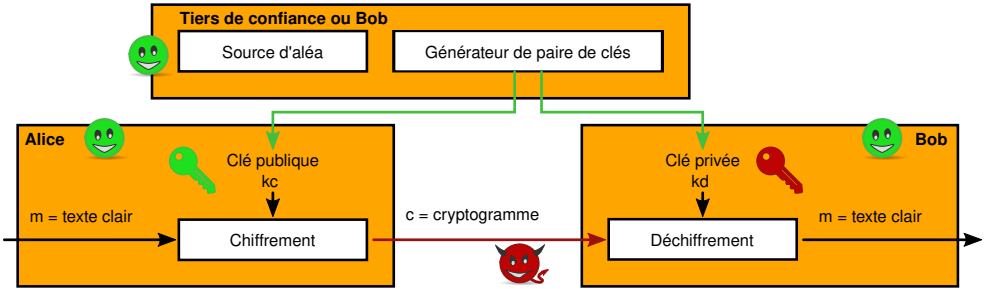
Constructions fondamentales : chiffrement asymétrique 3/6



Procédure : chiffrement $k_c \rightarrow k_d$

3. Alice chiffre le message avec la clé publique k_c
4. Le message est transmis au travers d'un canal non sécurisé
5. Bob déchiffre le message avec la clé secrète k_d

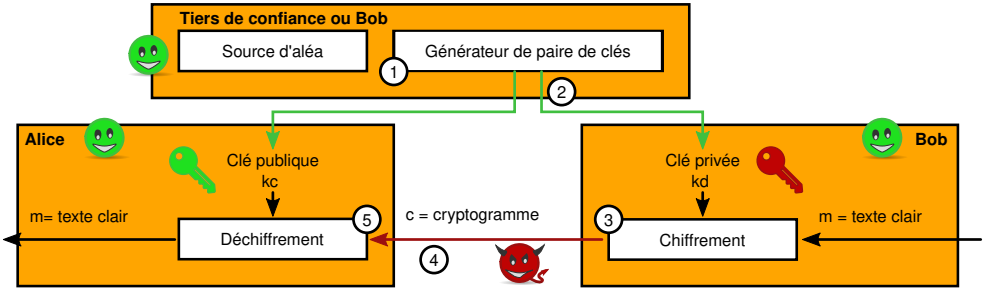
Constructions fondamentales : chiffrement asymétrique 4/6



Propriétés chiffrement $k_c \rightarrow k_d$

- M confidentiel

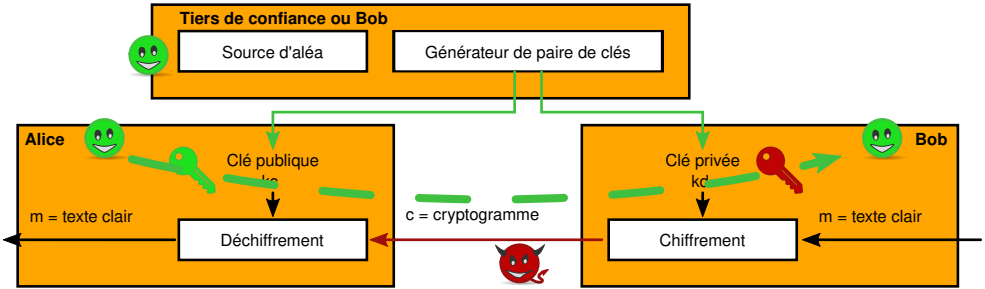
Constructions fondamentales : chiffrement asymétrique 5/6



Procédure : chiffrement $k_c \rightarrow k_d$

3. Bob chiffre le message avec sa clé privée k_d
4. Le message est transmis à Alice et au monde au travers d'un canal non sécurisé
5. Alice et le monde déchiffre le message avec la clé secrète k_d

Constructions fondamentales : chiffrement asymétrique 6/6



$k_c \rightarrow k_d$: propriétés

- Authentification de l'entité Bob : seul Bob peut calculer $E_{k_d}(M)$
- M non confidentiel

Notions de déterminisme et d'aléa

Chiffrement et déchiffrement

- Algorithmes déterministes
- Fonctions : $\exists$ une seule image y $\forall$ antécédent x de l'algorithme
- Propriété de cohérence : $M = D_{k_d}(E_{k_c}(M))$

Algorithmes aléatoires

- $\exists$ plus d'une image y $\forall$ antécédent x de l'algorithme
- Soit A l'ensemble des sorties possibles pour un algorithme
- Distribution, $\mathcal{D}$: on associe une probabilité d'occurrence à chaque élément de A (Ω)
- Distribution uniforme : $\forall y \in A, \mathcal{D}(y) = 1/|A|$

Conception d'un bon algorithme de chiffrement symétrique

Fonctions attendues

- Primitive de chiffrement
- Primitive de déchiffrement
- Propriété de cohérence

Sécurité d'un algorithme de chiffrement symétrique

- Sans connaître k_d, il doit être "impossible" de retrouver M

→ Le chiffré ne doit révéler aucune information sur le clair ni le chiffré

- Il doit être "impossible" de trouver k_d, même connaissant C et M
- Il doit être "impossible" de trouver k_d, même choisissant M

Étude d'un exemple

One Time Pad (OTP)

Propriétés du ou exclusif $\oplus$

Ou exclusif ou addition binaire (sans la retenue)

Table de vérité

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Propriétés

1. $x \oplus 0$
2. $x \oplus x$
3. $y \oplus x \oplus x$

Propriétés du ou exclusif $\oplus$

Ou exclusif ou addition binaire (sans la retenue)

Table de vérité

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Propriétés

1. $x \oplus 0 = x$
2. $x \oplus x$
3. $y \oplus x \oplus x$

Propriétés du ou exclusif $\oplus$

Ou exclusif ou addition binaire (sans la retenue)

Table de vérité

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Propriétés

1. $x \oplus 0 = x$
2. $x \oplus x = 0$
3. $y \oplus x \oplus x$

Propriétés du ou exclusif $\oplus$

Ou exclusif ou addition binaire (sans la retenue)

Table de vérité

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

Propriétés

1. $x \oplus 0 = x$
2. $x \oplus x = 0$
3. $y \oplus x \oplus x = y$

Un candidat : le *One Time Pad* (OTP, Vernam, 1917)

Masque jetable en français

Définition

- $M = C = K = \{0,1\}^n$
- Chiffrement : $c = E_k(m) = k \oplus m$
- Déchiffrement : $m = D_k(c) = k \oplus c$
- $E_k \Leftrightarrow D_k$

Propriétés

- Cohérent
- Performant, mise en œuvre simple

Contrainte

$|\mathrm{K}| = |\mathrm{M}|$ ...

Sécurité du *One Time Pad*

Propriété de sécurité 1

$M = K = C = \{0,1\}^n$
$c_i \in c$ à la position i
$P[c_i = 1]$?

```
mi : 0 1 0 1
ki : 0 0 1 1
-----------
ci : 0 1 1 0
```

Sécurité du *One Time Pad*

Propriété de sécurité 1

$M = K = C = \{0,1\}^n$
$c_i \in c$ à la position i
$P[c_i = 1]$?

```
mi : 0 1 0 1
ki : 0 0 1 1
-----------
ci : 0 1 1 0
```

1. $P[c_i = 1] = P[k_i = 0 \cap m_i = 1] + P[k_i = 1 \cap m_i = 0]$

Sécurité du *One Time Pad*

Propriété de sécurité 1

$M = K = C = \{0,1\}^n$
$c_i \in c$ à la position i
$P[c_i = 1]$?

```
mi : 0 1 0 1
ki : 0 0 1 1
-----------
ci : 0 1 1 0
```

1. $P[c_i = 1] = P[k_i = 0 \cap m_i = 1] + P[k_i = 1 \cap m_i = 0]$
2. $P[c_i = 1] = P[k_i = 0] \times P[m_i = 1] + P[k_i = 1] \times P[m_i = 0]$

Sécurité du *One Time Pad*

Propriété de sécurité 1

$M = K = C = \{0,1\}^n$
$c_i \in c$ à la position i
$P[c_i = 1]$?

```
mi : 0 1 0 1
ki : 0 0 1 1
-----------
ci : 0 1 1 0
```

1. $P[c_i = 1] = P[k_i = 0 \cap m_i = 1] + P[k_i = 1 \cap m_i = 0]$
2. $P[c_i = 1] = P[k_i = 0] \times P[m_i = 1] + P[k_i = 1] \times P[m_i = 0]$
3. $P[c_i = 1] = 1/2 \times 1/2 + 1/2 \times 1/2 = 1/2$

Sécurité du *One Time Pad*

Propriété de sécurité 1

$M = K = C = \{0,1\}^n$
$c_i \in c$ à la position i
$P[c_i = 1]$?

```
mi : 0 1 0 1
ki : 0 0 1 1
-----------
ci : 0 1 1 0
```

1. $P[c_i = 1] = P[k_i = 0 \cap m_i = 1] + P[k_i = 1 \cap m_i = 0]$
2. $P[c_i = 1] = P[k_i = 0] \times P[m_i = 1] + P[k_i = 1] \times P[m_i = 0]$
3. $P[c_i = 1] = 1/2 \times 1/2 + 1/2 \times 1/2 = 1/2$

$\Rightarrow P[c_i = x]$ est uniforme

Sécurité du *One Time Pad*

Propriété de sécurité 2

$M = K = C = \{0,1\}^2$

```
m : 0 1 | 0 0 | 1 1 | 1 0
k : 0 0 | 0 1 | 1 0 | 1 1
-------------------------
c : 0 1 | 0 1 | 0 1 | 0 1
```

Sécurité du *One Time Pad*

Propriété de sécurité 2

$M = K = C = \{0,1\}^2$

```
m : 0 1 | 0 0 | 1 1 | 1 0
k : 0 0 | 0 1 | 1 0 | 1 1
------------------------
c : 0 1 | 0 1 | 0 1 | 0 1
```

Pour un chiffré donné, tout clair peut être un antécédent

Sécurité parfaite 1/2

Pour $n \in \mathbb{N} \Rightarrow |K| = |M| = |C|$ de taille quelconque

Définition

$\forall m, \in M, \forall c \in C$ on a $P[M = m|C = c] = Pr[M = m]$

- L'attaquant n'apprend rien du chiffré
- Une attaque utilisant seulement le chiffré est impossible

Stragégie de preuve, intuition

Montrer que $P[M = m]$ et $P[C = c]$ sont indépendants
$\rightarrow P[M = m|C = c] \Leftrightarrow P[M = m \cap C = c]/P[C = c] = P[M = m] \times P[C = c]/P[C = c] = P[M = m]$

Indiscernabilité parfaite

$\forall m_0, m_1 \in M, \forall c \in C$ on a $P[E_k(m_0) = c] = P[E_k(m_1) = c]$
avec $k \in K$ variable aléatoire

Sécurité parfaite 2/2

Intuition

(1) Pour $c \in C$ et $m \in M$ fixés, combien de clé peuvent chiffrer m en c ?

(2) Quel est le cardinal de K, $|K|$?

Sécurité parfaite 2/2

Intuition

(1) Pour $c \in C$ et $m \in M$ fixés, combien de clé peuvent chiffrer m en c ?
Réponse : 1

(2) Quel est le cardinal de K, $|K|$?
Réponse : 2^n

Sécurité parfaite 2/2

Intuition

(1) Pour $c \in C$ et $m \in M$ fixés, combien de clé peuvent chiffrer m en c ?
Réponse : 1

(2) Quel est le cardinal de K, $|K|$?
Réponse : 2^n

Indiscernabilité parfaite

$P[E_k(m) = c] = (1)/(2)$
$P[E_k(m) = c] = 1/2^n = P[E_k(m_0) = c] = P[E_k(m_1) = c]$

Limites 1/2

Maléabilité

Soit $m \in M, c \in C, k \in K$
$m = E_k(m)$
Et après une attaque : $c_2|c_2 = c \oplus x$
$D_k(c_2) = c_2 \oplus k = c \oplus k \oplus x \oplus k = c \oplus x$
L'attaquant $\oplus$ directement le clair ! !

Réutilisation de la clé impossible

Soit $m_1, m_2 \in M, c_1, c_2 \in C, k \in K$
$c_1 = E_k(m_1)$ et $c_2 = E_k(m_2)$
$c_1 \oplus c_2 = m_1 \oplus k \oplus m_2 \oplus k = m_1 \oplus m_2$
Ou exclusif des clairs ! !

Limites 2/2

Réutilisation de la clé impossible 2

Soit $m_1, m_2 \in M, c_1, c_2 \in C, k \in K$
$c_1 = E_k(m_1)$ et $c_2 = E_k(m_2)$
Si m_1 est connu par l'attaquant
$m_1 \oplus c_1 = m_1 \oplus m_1 \oplus k = k$
L'attaquant peut déchiffrer m_2

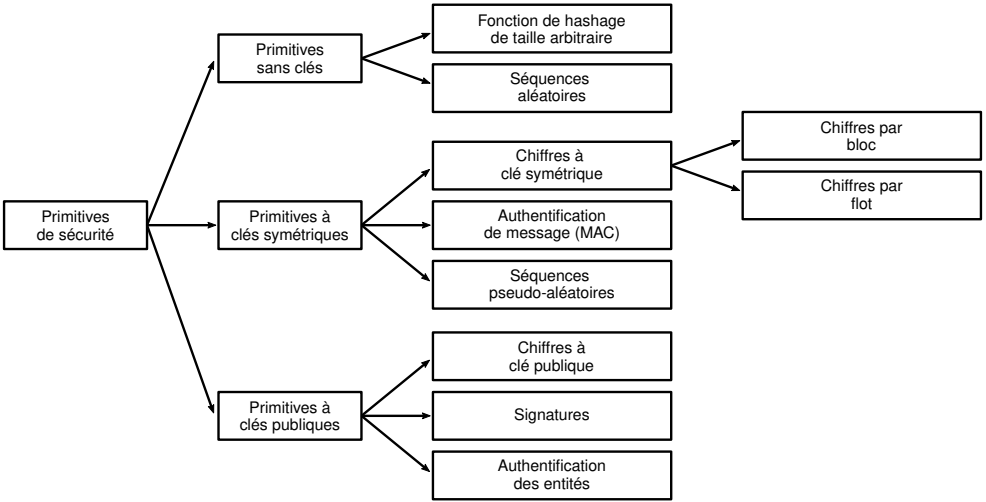
$|K| >= |M|$

On montre que $P[M = m] = 1/2^n$

Constructions fondamentales : modèles de confiance

- Root of trust / Pinned
- TOFU
- Web of trust
- PKI

Terminologie : *mindmap*



Nomenclature simplifiée des primitives de sécurité cryptographiques [9]

Chiffrement symétriques : $k_c = k_d$

- Tous les chiffres connus jusqu'en 1976 !
- Chiffre par bloc
  - Texte découpé en blocs de taille fixe pour traitement
  - Souvent associé à un mode d'opération
  - Certains modes transforment en primitive par flot (CTR, CFB)
- Chiffre par flot
  - Génération indépendante de la clé
  - Puis application d'une fonction réversible sur le clair ($\oplus$)
  - Texte clair de taille arbitraire
- Exemples :

Bloc :

- DES (1976)
  Clés de 56 bits (plus 8 bits de parité)
  Blocs de 64 bits
- AES (2000)
  Clés de 128, 192, 256 bits
  Blocs de 128 bits

Flot :

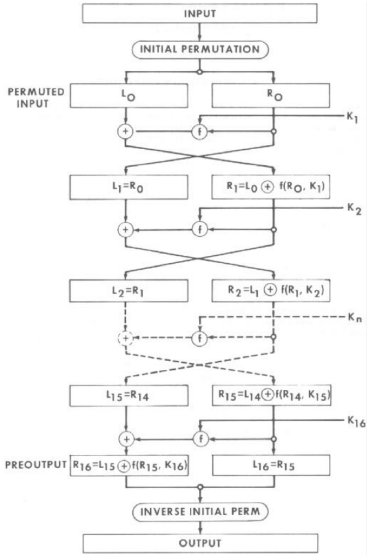
- RC4 (1987)
  Ronald Rivest
- Salsa20 (2005)
  Daniel J. Bernstein
- AES-CTR
  Pré-calcul de la clé

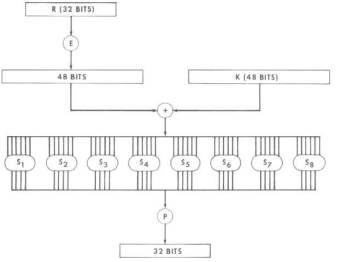
DES, Data Encryption Standard

csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

1. Diversification de la clé → 16 sous clés $K_{1..16}$ de 48 bits
   Chaque K_i est composé de 48 bits de K pris dans un certain précis
2. Fractionnement du texte en blocs $B_{1..n}$ de 64 bits
3. Pour chaque bloc B_j
   1. Permutation initiale du bloc B_j
   2. Découpage du bloc B_j en parties gauche G_0 et droite D_0
   3. Pour chaque sous clé, K_i
      1. $G_i = D_{i-1}$
      2. $D_i = G_{i-1} \oplus f(D_{i-1}, K_i)$
   4. Reconstitution du bloc B'_j à partir de G_{16} et D_{16}
   5. Permutation initiale inverse du bloc B'_j

DES, Data Encryption Standard





Fonction $f(R, K)$

IP

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

Permutation initiale

Cryptanalyse : niveaux d'attaques

Niveau de puissance de l'attaquant

1. Attaque à texte chiffré : → récupérer le clair, voire la clé
   - Possède des messages chiffrés
2. Attaque à clair connu :
   - Possède des couples de message clair / chiffré
3. Attaque à clair choisi :
   - Construit des couples de message clair / chiffré
   - Choisi le clair à chiffrer, Chiffre en mode boite noire
3. Attaque à chiffré choisi :
   - Construit des couples de message clair / chiffré
   - Choisi le chiffré à chiffrer, Chiffre en mode boite noire

Cryptanalyse : types d'attaques

Précédent la cryptographie moderne

- Analyse fréquentielle (texte chiffré)
- Indice de coïncidence (texte chiffré)
- Mot probable (clair connu)
- Force Brute

Cryptographie moderne

- Cryptanalyse linéaire (clair connu)
- Cryptanalyse différentielle (clair choisi)
- Canal auxiliaire (temps, consommation, e.m.)

Cryptanalyse : modèles de sécurité

Modèle pour caractériser le niveau de sécurité d'un chiffre

- Sécurité inconditionnelle (*perfect secrecy*)
  - Théorie de l'information Shannon
  - $H(M) = H(M|C)$, $H(X)$ entropie de X, incertitude
  - Taille de la clé nécessairement aussi grande que le message
    One-time pad
- Sécurité prouvable
  - *Équivalence du chiffre avec un problème difficile connu*

  → Réduction à un problème NP (ex : réseaux euclidiens)
- Sécurité par complexité de calcul (*computational security*)
  - Hypothèse sur la puissance de calcul de l'attaquant ($O(2^{80})$)
  - *Quantité de calcul à exécuter avec la meilleure méthode connue*
  - Concerne la plupart des chiffres modernes
  - ex : bits de clés (chiffre symétrique), RSA, logarithme discret

Cryptanalyse : mise en œuvre des chiffres

De la théorie vers la pratique

- La mise en œuvre des chiffres est non triviale
- Protection des secrets en mémoire (TEE, HSM)
- Gestion de l'aléa (matériel quantique / chaotique, post traitement)
- Protection contre les attaques intrusives
- Protection contre le canaux auxiliaires
- Et bien d'autres...

À retenir

- Mettre en œuvre de la cryptographie est très difficile

→ Préférer les projets ouverts, de spécialistes et à l'état de l'art

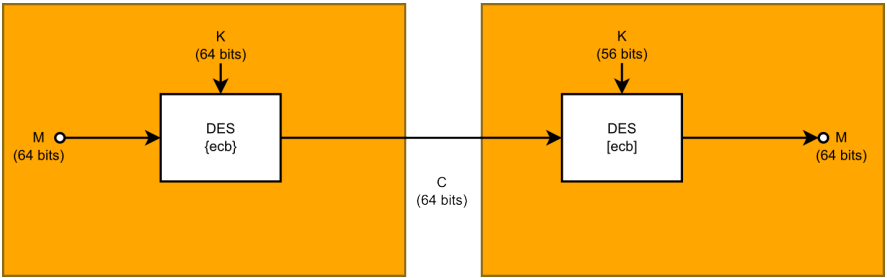
- NaCL, libsodium, etc (Daniel J. Bernstein)
- openssl : largement éprouvé par une communauté

Cryptanalyse : niveau de sécurité

Cible de sécurité : combinaison subtile des paramètres suivants :

- La configuration du chiffre (taille des clés, etc.)
- Niveau d'attaque / attaques à considérer
- Le modèle de sécurité considéré (qui évolue : bits de sécurité)
- La qualité de la mise en œuvre
- Niveau et durée d'évaluation du chiffre
- Niveau et durée évaluation de la mise en œuvre
- D'autres paramètres d'environnement : qualité de l'aléa, etc.
- Et bien d'autres...

DES, mode *ECB* (*Electronic-Code-Book*)



Mode d'opération par bloc

Avantages

- Parallélisme
- Accès aléatoire

DES, mode *ECB* (*Electronic-Code-Book*)

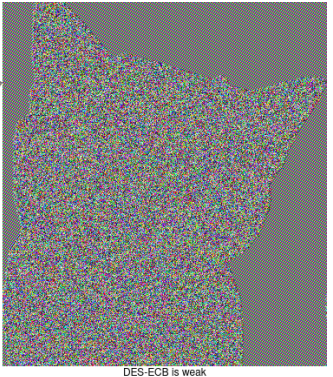
Faiblesses

- Tout bloc clair dans un texte, donnera systématiquement le même bloc chiffré

DES, mode *ECB* (*Electronic-Code-Book*)

Faiblesses

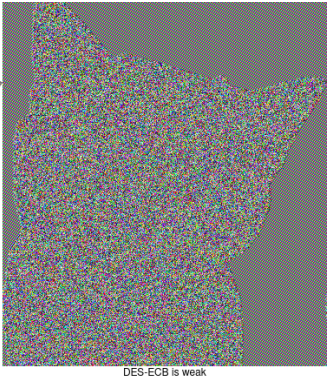
- Tout bloc clair dans un texte, donnera systématiquement le même bloc chiffré



DES, mode *ECB* (*Electronic-Code-Book*)

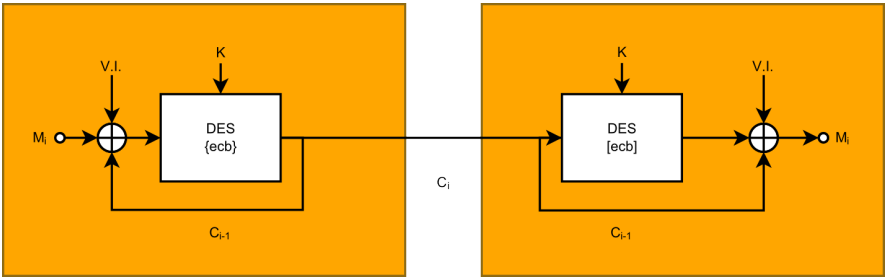
Faiblesses

- Tout bloc clair dans un texte, donnera systématiquement le même bloc chiffré



Déconseillé

DES, mode *CBC* (*Cipher-Block-Chaining*)



Mode d'opération par bloc

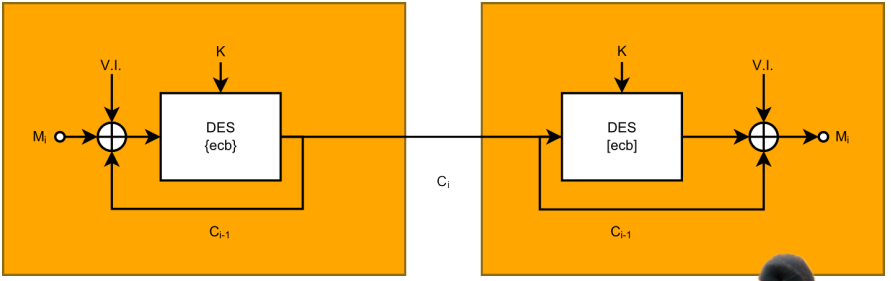
Avantages

- Accès aléatoire
- Déchiffrement en parallèle

Désavantages

- Chiffrement séquentiel

DES, mode *CBC* (*Cipher-Block-Chaining*)



Mode d'opération par bloc

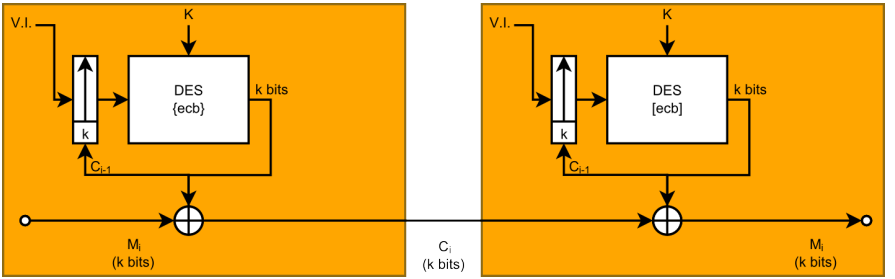
Forces

- Deux même blocs de texte clair seront deux blocs chiffrés différents (*idem* autres modes)

Faiblesses

- Bourrage nécessaire (attaque POODLE SSLv3)

DES, mode *OFB* (*Output-Feedback-Block*)



Mode d'opération par flot synchrone

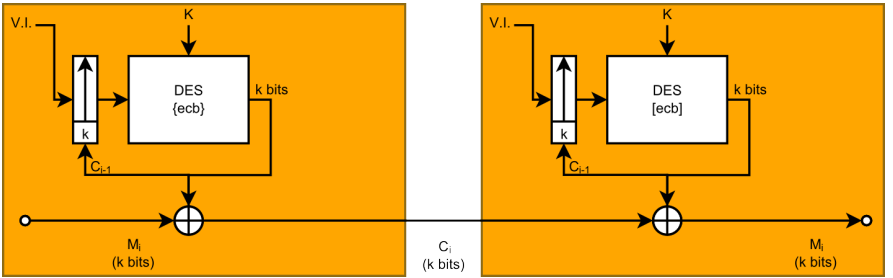
Avantages

- Même circuit de chiffrement et déchiffrement
- $\exists$ codes correcteurs d'erreurs sont applicables sur le chiffré

Désavantages

- Chiffrement Séquentiel
- Déchiffrement Séquentiel
- Besoin de synchronisation parfaite (client / serveur)

DES, mode *OFB* (*Output-Feedback-Block*)



Mode d'opération par flot synchrone

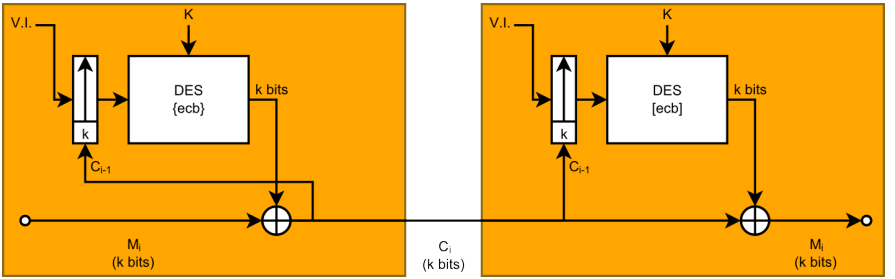
Forces

- En plus ?

Faiblesses

- Attaque active facilitée : 1 *bit flip* clair = 1 *bit flip* chiffré
- Attaque à clair connu : 1 IV + 1 K = 1 keystream

DES, mode *CFB* (*Cipher-Feedback-Block*)



Mode d'opération par flot auto synchrone (si k bits perdus ou ajoutés)

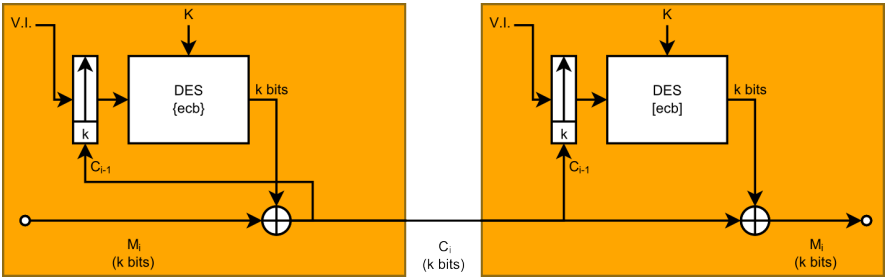
Avantages

- Accès aléatoire
- Déchiffrement en parallèle
- Synchronisation modulo k (client / serveur)

Désavantages

- Chiffrement séquentiel

DES, mode *CFB* (*Cipher-Feedback-Block*)



Mode d'opération par flot auto synchrone (si k bits perdus ou ajoutés)

Forces

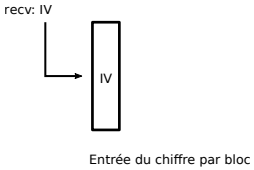
- En plus ?

Faiblesses

- ?

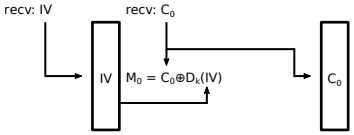
DES, mode *CFB* (*Cipher-Feedback-Block*)

Fonctionnement normal sans *shift register*



DES, mode *CFB* (*Cipher-Feedback-Block*)

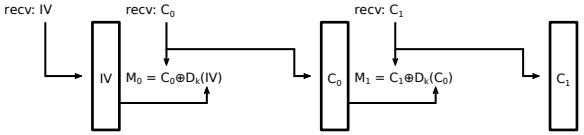
Fonctionnement normal sans *shift register*



Entrée du chiffre par bloc

DES, mode *CFB* (*Cipher-Feedback-Block*)

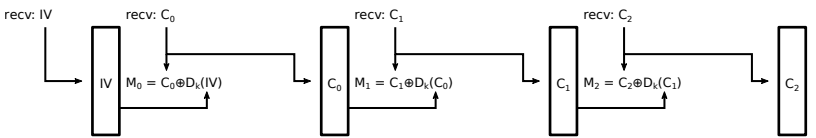
Fonctionnement normal sans *shift register*



Entrée du chiffre par bloc

DES, mode *CFB* (*Cipher-Feedback-Block*)

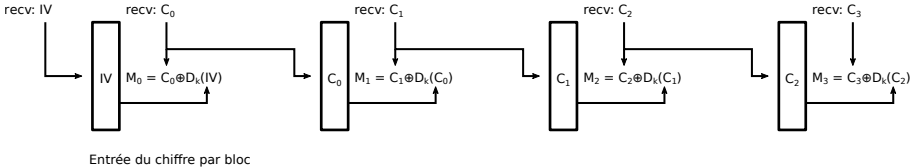
Fonctionnement normal sans *shift register*



Entrée du chiffre par bloc

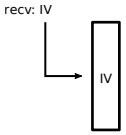
DES, mode *CFB* (*Cipher-Feedback-Block*)

Fonctionnement normal sans *shift register*



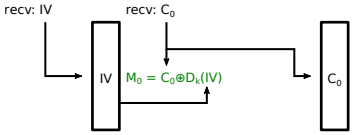
DES, mode *CFB* (*Cipher-Feedback-Block*)

Perte de n bits et resynchronisation après n bits reçus



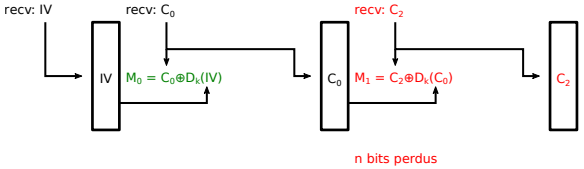
DES, mode *CFB* (*Cipher-Feedback-Block*)

Perte de n bits et resynchronisation après n bits reçus



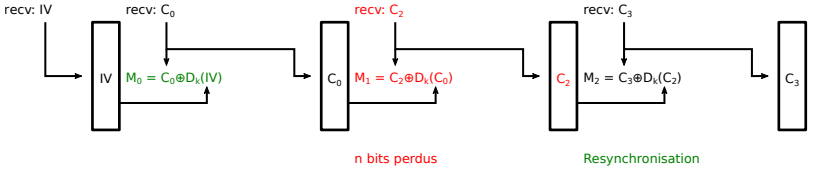
DES, mode *CFB* (*Cipher-Feedback-Block*)

Perte de n bits et resynchronisation après n bits reçus



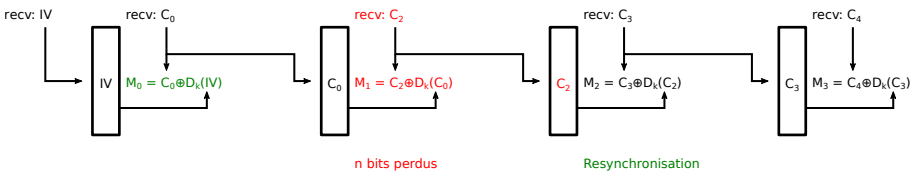
DES, mode *CFB* (*Cipher-Feedback-Block*)

Perte de n bits et resynchronisation après n bits reçus



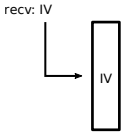
DES, mode *CFB* (*Cipher-Feedback-Block*)

Perte de n bits et resynchronisation après n bits reçus



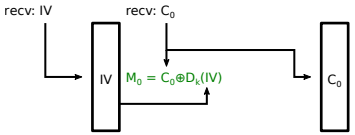
DES, mode *CFB* (*Cipher-Feedback-Block*)

Désynchronisation totale après perte de $n/2$ bits



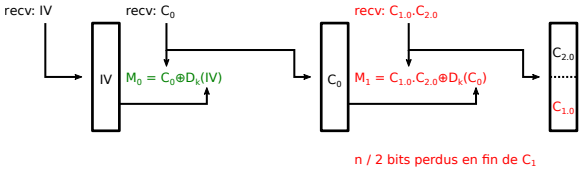
DES, mode *CFB* (*Cipher-Feedback-Block*)

Désynchronisation totale après perte de $n/2$ bits



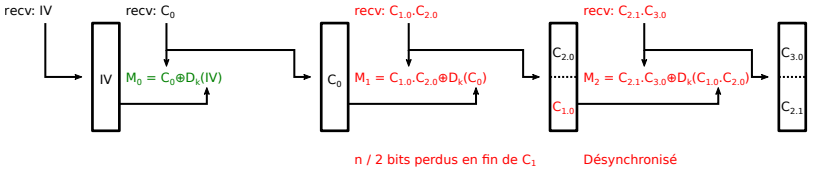
DES, mode *CFB* (*Cipher-Feedback-Block*)

Désynchronisation totale après perte de $n/2$ bits



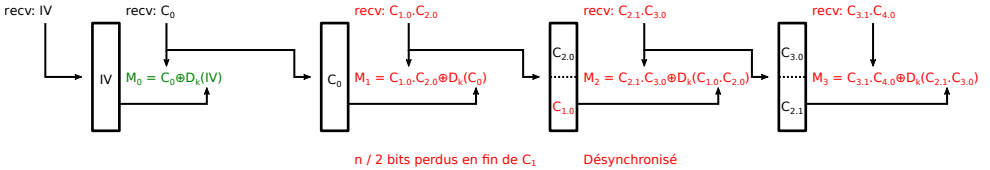
DES, mode *CFB* (*Cipher-Feedback-Block*)

Désynchronisation totale après perte de $n/2$ bits



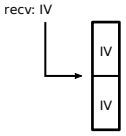
DES, mode *CFB* (*Cipher-Feedback-Block*)

Désynchronisation totale après perte de $n/2$ bits



DES, mode *CFB* (*Cipher-Feedback-Block*)

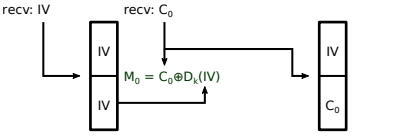
Fonctionnement normal avec un *shift register* décalant de $k = n/2$ bits



Shift register de n bits, k = n / 2; C_i, M_i n / 2 bits

DES, mode *CFB* (*Cipher-Feedback-Block*)

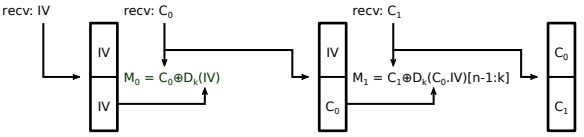
Fonctionnement normal avec un *shift register* décalant de $k = n/2$ bits



Shift register de n bits, k = n / 2; C_i, M_i n / 2 bits

DES, mode *CFB* (*Cipher-Feedback-Block*)

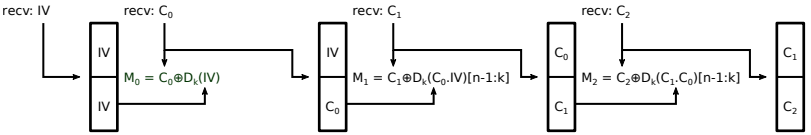
Fonctionnement normal avec un *shift register* décalant de $k = n/2$ bits



Shift register de n bits, k = n / 2; C_i, M_i n / 2 bits

DES, mode *CFB* (*Cipher-Feedback-Block*)

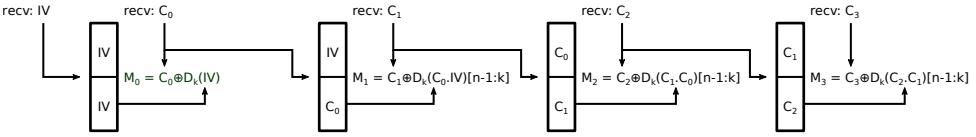
Fonctionnement normal avec un *shift register* décalant de $k = n/2$ bits



Shift register de n bits, k = n / 2; C_i, M_i n / 2 bits

DES, mode *CFB* (*Cipher-Feedback-Block*)

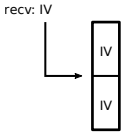
Fonctionnement normal avec un *shift register* décalant de $k = n/2$ bits



Shift register de n bits, k = n / 2; C_i, M_i n / 2 bits

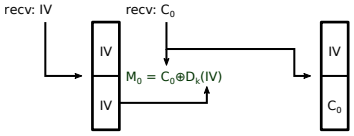
DES, mode *CFB* (*Cipher-Feedback-Block*)

Perte de $n/2$ bits et resynchronisation après n bits reçus



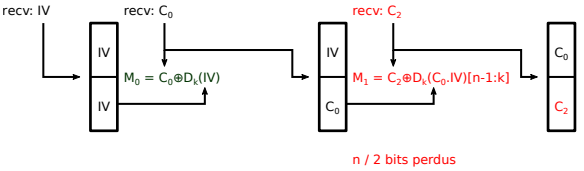
DES, mode *CFB* (*Cipher-Feedback-Block*)

Perte de $n/2$ bits et resynchronisation après n bits reçus



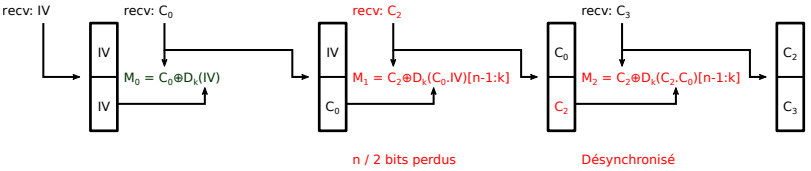
DES, mode *CFB* (*Cipher-Feedback-Block*)

Perte de $n/2$ bits et resynchronisation après n bits reçus



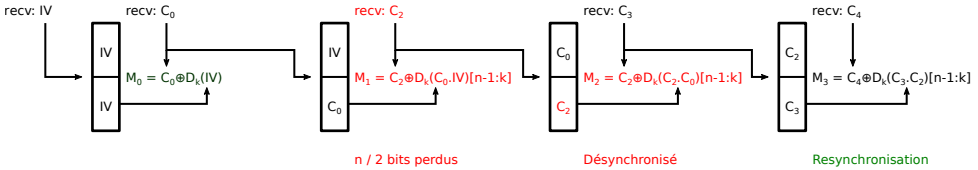
DES, mode *CFB* (*Cipher-Feedback-Block*)

Perte de $n/2$ bits et resynchronisation après n bits reçus



DES, mode *CFB* (*Cipher-Feedback-Block*)

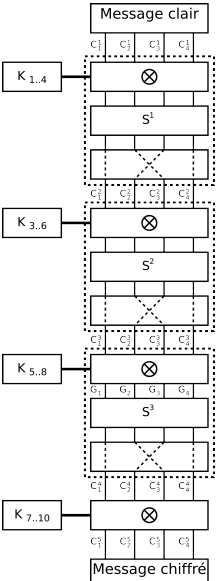
Perte de $n/2$ bits et resynchronisation après n bits reçus



Cryptanalyse linéaire

- Attaque à clairs connus
  - $\mathcal{T} = \{(\text{message}_i, \text{ciphertext}_i)\}$
- Sur les réseaux de substitution-permutation (SPN)
  - AES, DES,. . .
- L'attaquant dispose de l'algorithme et recherche la clé
- Exploitation d'un manque de non linéarité pour établir des approximations

  ⇒ Réduction de l'espace de recherche
  Être plus rapide que la force brute



Cryptanalyse linéaire – non linéaire ?

- Si substitution S^i est non linéaire $\Rightarrow$ robuste à cette attaque. . .
- $\forall y = S^i(x), \mathcal{P}(\oplus_{i=1}^{n} x_i = \oplus_{i=1}^{n} y_i) = 1/2$
- Parfois non vérifié pour des sous parties des vecteurs y et x
- ex : si (1) $x_4 = y_4 \oplus y_3 \oplus y_2$ vérifié 14 fois sur 16
- Biais $\epsilon = |\mathcal{P}[(1)] - 1/2| = 3/8$

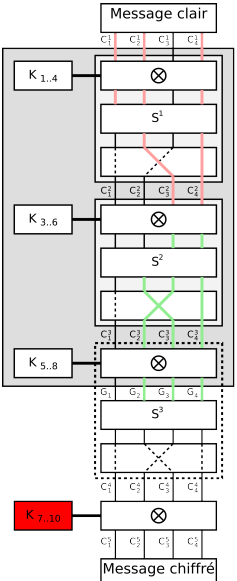
$\Rightarrow$ Utiliser ϵ comme distingueur

- C'est à dire, pour un grand nombre de vecteurs x et y, si l'équation est vérifiée avec un biais de 3/8, y a été bien généré par $S^i(x)$, pour les bits concernés
- **Intuition :** vérifier le biais sur tous les couples (clair, chiffré) est plus rapide qu'appliquer la subsitution
- Sur une approximation globale de l'algorithme $\rightarrow$ exctraction plus rapide de bits de clés qu'une attaque par force brute sans approximation

Cryptanalyse linéaire – démarche

- Approximation d'une partie de l'algorithme
  - Indépendante de la clé, pour réduire l'espace de recherche
  - Dépendante des messages clairs et de l'entrée du dernier bloc (tous les *rounds* sauf le dernier)
  - **Rapide à exécuter**
- Attaque par force brute sur la partie non approximée de l'algorithme
  - Première étape du déchiffrement avec une clé candidate

⇒ Trouver une partie de la clé K, correspondant au dernier *round*



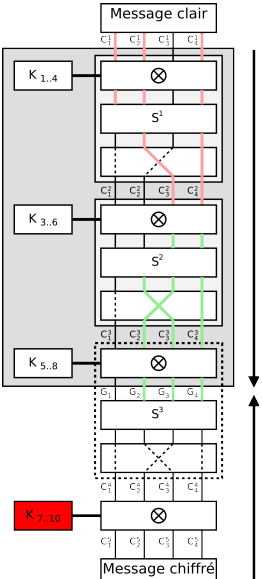
Cryptanalyse linéaire – algorithme

Version simplifiée

for candidat = 0 .. $2^4 - 1$ **do**
  $\text{score}_{\text{candidat}} \leftarrow 0$
  for all (message, ciphertext) $\in \mathcal{T}$ **do**
    valeur $\leftarrow$ ciphertext $\oplus$ candidat
    valeur $\leftarrow (\text{Permute}^3)^{-1}$(valeur)
    valeur $\leftarrow (S^3)^{-1}$(valeur)
    Test si l'approximation est vérifiée par le couple valeur, message
    if Approximation_Avérée(valeur, message) **then**
      $\text{score}_{\text{candidat}} \leftarrow \text{score}_{\text{candidat}} + 1$
    end if
  end for
end for
résultat $\leftarrow \underset{x}{\text{argmax}}\ \text{abs}(\text{score}_x - |\mathcal{T}|/2)$

On sélectionne le candidat qui a le biais le plus proche :

$$\epsilon = \left| \frac{\text{score}_{\text{candidat}} - |\mathcal{T}|/2}{|\mathcal{T}|} \right|$$



Cryptanalyse linéaire

- Approximation linéaire des premiers *rounds*
- **Objectif :** obtenir une équation de la forme :

$$(\bigoplus_{i=1}^{4} a_i \wedge C_i^1) \oplus (\bigoplus_{i=1}^{4} b_i \wedge G_i^4) \oplus \text{Constante} = 0$$

$$\mathcal{P}[(\bigoplus_{i=1}^{4} a_i \wedge C_i^1) \oplus (\bigoplus_{i=1}^{4} b_i \wedge G_i^4) \oplus \text{Constante} = 0] = 1/2 + \epsilon$$

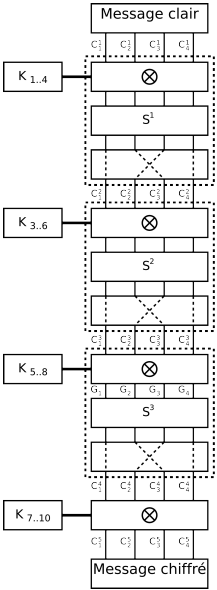
a_i et b_i sont des masques

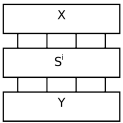
- **Objectif :** associer le biais attendu :
- Le biais attendu est calculé par combinaison des ϵ de chaque round
- Utilisation du lemme *Piling-Up* produit par Mitsuru Matsui
- Constante : clé de chiffrement → biais positif ou négatif ($\oplus = 1$ ou 0)

Cryptanalyse linéaire

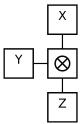
- Approximation linéaire des premiers *rounds*
  - Dernier *round* non approximé : toute ou partie de la clé correspondante va être recherchée
  - Pour chaque clé candidate au dernier *round*, il est possible de calculer C_i^4 correspondant, puis G_i^4
  - La clé vraisemblablement utilisée pour chiffrer les message$_i$ en ciphertext$_i$ est celle pour laquelle l'approximation est la meilleure ...

Cryptanalyse linéaire – exemple





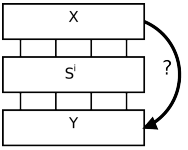
XXXX	YYYY
0000	0011
0001	1000
0010	1001
0011	0101
0100	1011
0101	0001
0110	0010
0111	0100
1000	0000
1001	1111
1010	0111
1011	0110
1100	1110
1101	1010
1110	1100
1111	1101



XY	Z
00	0
01	1
10	1
11	0

- Plusieurs *rounds*
- Fonction non linéaire S^i, ne peut pas être exprimée sous la forme de xor

Cryptanalyse linéaire – exemple, table des approximations



$N(\{(a_1, a_2, a_3, a_4), (b_1, b_2, b_3, b_4)\}) = |\{(X_1, X_2, X_3, X_4), (Y_1, Y_2, Y_3, Y_4) / \bigoplus_{i=1}^{4} a_i X_i \bigoplus_{i=1}^{4} b_i Y_i = 0\}|$

(a_1, a_2, a_3, a_4) \ (b_1, b_2, b_3, b_4)	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0001	0	0	-2	2	2	-2	0	0	0	0	-2	2	2	6	0	0
0010	0	0	-2	2	4	0	2	2	-2	-2	4	0	2	-2	0	0
0011	0	0	0	0	-2	-2	-2	6	2	2	2	2	0	0	0	0
0100	0	-2	0	2	0	2	4	2	2	0	-2	0	-2	0	-2	4
0101	0	-2	2	0	2	0	0	-2	2	0	0	6	0	-2	2	0
0110	0	2	2	4	0	2	-2	0	0	2	-2	0	4	-2	-2	0
0111	0	2	0	-2	2	0	-2	0	4	-2	0	-2	2	0	2	4
1000	0	-2	2	0	4	-2	-2	0	2	0	0	-2	-2	0	-4	-2
1001	0	-2	-4	-2	2	0	-2	0	-2	4	-2	0	0	-2	0	2
1010	0	-2	0	2	0	-2	0	2	0	-2	-4	-2	0	-2	4	-2
1011	0	-2	-2	4	-2	0	-4	-2	0	-2	2	0	-2	0	0	2
1100	0	0	2	-2	0	0	-2	2	-4	-4	-2	2	0	0	-2	2
1101	0	0	0	0	-2	-6	2	-2	0	0	0	0	2	-2	-2	2
1110	0	4	-4	0	0	0	0	0	2	-2	-2	2	-2	-2	-2	-2
1111	0	-4	-2	-2	-2	2	0	0	2	-2	0	0	4	0	-2	-2

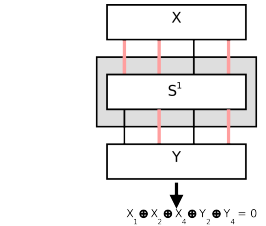
XXXX	YYYY	XXXYY	⊕
0000	0011	00001	1
0001	1000	00100	1
0010	1001	00001	1
0011	0101	00111	1
0100	1011	01001	0
0101	0001	01101	1
0110	0010	01000	1
0111	0100	01110	1
1000	0000	10000	1
1001	1111	10111	0
1010	0111	10011	1
1011	0110	10110	1
1100	1110	11010	1
1101	1010	11100	1
1110	1100	11010	1
1111	1101	11111	1

XXXX	YYYY	XXYYY	⊕
0000	0011	00011	0
0001	1000	01000	1
0010	1001	10001	0
0011	0101	11101	0
0100	1011	00011	0
0101	0001	01001	0
0110	0010	10010	0
0111	0100	11100	1
1000	0000	00000	0
1001	1111	01111	0
1010	0111	10111	0
1011	0110	11110	0
1100	1110	00110	0
1101	1010	01010	0
1110	1100	10100	0
1111	1101	11101	0

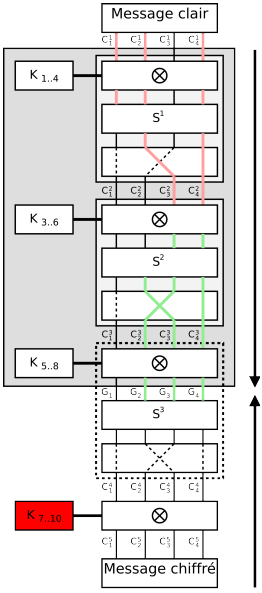
XXXX	YYYY	XXYYY	⊕
0000	0011	00011	0
0001	1000	01100	0
0010	1001	00101	0
0011	0101	01001	0
0100	1011	10111	0
0101	0001	11001	1
0110	0010	10010	0
0111	0100	11000	0
1000	0000	00000	0
1001	1111	01111	0
1010	0111	00011	0
1011	0110	01010	0
1100	1110	10110	1
1101	1010	11110	0
1110	1100	10100	0
1111	1101	11101	0

XXXX	YYYY	XXYY	⊕
0000	0011	0001	0
0001	1000	1100	0
0010	1001	0101	0
0011	0101	1011	0
0100	1011	0101	0
0101	0001	1001	1
0110	0010	0000	0
0111	0100	1010	0
1000	0000	0000	0
1001	1111	1111	0
1010	0111	0011	0
1011	0110	1010	0
1100	1110	0110	1
1101	1010	1100	0
1110	1100	0110	0
1111	1101	1111	0

Cryptanalyse linéaire – exemple, approximation



Cryptanalyse linéaire – exemple, algorithme



$$C_1^1 \oplus C_2^1 \oplus C_4^1 \oplus C_2^3 \oplus C_4^3 \oplus K_1 \oplus K_3 \oplus K_4 = 0$$

$$C_3^1 \oplus C_4^1 \oplus C_2^3 \oplus C_3^3 \oplus C_4^3 \oplus K_3 \oplus K_4 = 0$$

$$C_1^1 \oplus C_3^1 \oplus C_4^1 \oplus C_2^2 \oplus C_4^2 \oplus K_1 \oplus K_2 \oplus K_4$$

$$\oplus C_3^2 \oplus C_4^2 \oplus C_2^3 \oplus C_4^3 \oplus K_3 \oplus K_4 \oplus$$

$$C_2^2 \oplus C_3^2 \oplus C_4^2 \oplus G_2 \oplus G_3 \oplus G_4 \oplus K_6 \oplus K_7 \oplus K_8 = 0$$

$$C_1^1 \oplus C_3^1 \oplus C_4^1 \oplus G_2 \oplus G_3 \oplus G_4$$

$$\oplus K_3 \oplus K_4 \oplus K_3 \oplus K_5 \oplus K_6 \oplus K_7 \oplus K_8 = 0$$

Constante

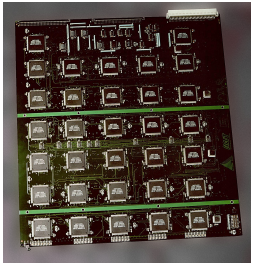
$$C_1^1 \oplus C_2^1 \oplus C_4^1 \oplus G_4 \oplus G_3 \oplus G_4 = 0$$

for $k = 0 \,..\, 2^4 - 1$ **do**
  $s_k \leftarrow 0$
  for all $(m, c) \in \mathcal{T}$ **do**
    $g \leftarrow c \oplus k$
    $g \leftarrow (\text{Permute}^3)^{-1}(g)$
    $g \leftarrow (S^3)^{-1}(g)$
    if $g_2 \oplus g_3 \oplus g_4 \oplus m_1 \oplus m_2 \oplus m_4 = 0$ **then**
      $s_k \leftarrow s_k + 1$
    end if
  end for
end for
$r \leftarrow \underset{x}{\text{argmax}}\ \text{abs}(s_x - |\mathcal{T}|/2)$

DES, cryptanalyse

- Clé sur 56 bits $\Rightarrow 2^{56}$ clés possibles
- Possibilité d'attaques par brute force
  - *Deep Crack* – Cryptography Research, Advanced Wireless Technologie, EFF
  - 29 cartes de 64 puces (1 856 puces spécialisées pour le DES)
  - 90 m^{ds} de clés testées par seconde

  $\Rightarrow$ Environ 5 jours pour tester toutes les possibilités



Avantages des chiffres symétriques

- Rapides
  - Exemple avec AES
    - Jusqu'à 100 Gb/s sur du matériel spécifique
    - Jusqu'à 250 Mb/s avec du logiciel (MavBook Pro)
- Clés *courtes* : typiquement 80 bits pour résister aux attaques *brutales* (aujourd'hui)

  `www.rsa.com/rsalabs/node.asp?id=2103`
  - DES (ECB) cassé en octobre 1997 (22h avec un matériel spécifique)
  - RC5-56 cassé en octobre 1997 (250j sur Internet)
  - RC5-64 cassé en juillet 2002 (1757j sur Internet)
- Pratiques pour chiffrer des fichiers personnels
  → pas de clé à partager

Problèmes des chiffres symétriques

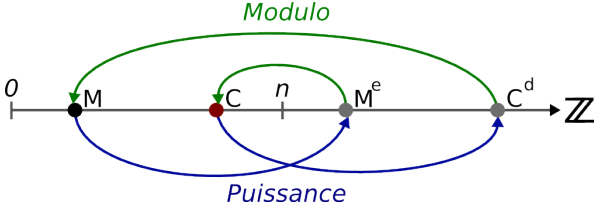
- Communication : clé secrète partagée
  Il faut que l'émetteur et le récepteur se fassent confiance, et gardent soigneusement la clé secrète
- Comment distribuer/renouveler la clé ?
  - Chiffrer la nouvelle clé de session avec l'ancienne
  - Chiffrer la clé de session avec une clé spécifique de chaque matériel ⇒ site de confiance (répertoire)
  - Cryptographie quantique
  - Utiliser un système à clé publique (Diffie-Hellman)

Chiffres à clé publique : $k_c \neq k_d$

- Connaissant k_c, il est "impossible" de trouver k_d
  - k_d est "privé" : seul celui qui connaît k_d peut déchiffrer
  - k_c est public : tout le monde peut chiffrer → répertoire de clés publiques
- Exemples
  - RSA (1978) → difficulté de factoriser de grands nombres[13]
  - El Gamal (1985) → difficulté de calcul des logarithmes discrets[4]

RSA – Rivest, Shamir, Adleman

- Création des clés
  - Choisir p et q deux nombres premiers distincts
  - $\Rightarrow$ Calculer le *module de chiffrement* n, $n = p \cdot q$
  - $\Rightarrow$ Calculer l'*indicatrice d'Euleur* de n, $\phi(n) = (p-1) \cdot (q-1)$
  - Choisir l'*exposant de chiffrement* e, un entier premier avec $\phi(n)$,
  - $\Rightarrow$ Calculer l'*exposant de déchiffrement* d, $e \cdot d \equiv 1 \mod \phi(n)$
    Algorithme d'Euclide étendu
  - $\Rightarrow$ $k_c = \{e, n\}$ $k_d = \{d, n\}$
- Chiffrement : $C = M^e \mod n$, avec $M < n$
- Déchiffrement : $M = C^d \mod n$
- Décomposition de n en produit de facteurs premiers $\rightarrow$ p et q $\mathcal{O}(e^n)$



RSA – Rivest, Shamir, Adleman

- Exemple
  - Création des clés
    - $p = 5,\ q = 11$

    $\Rightarrow n = p \cdot q = 5 \cdot 11 = 55$

    $\Rightarrow \phi(n) = (p-1) \cdot (q-1) = (5-1) \cdot (11-1) = 4 \cdot 10 = 40$
    - $e = 3$

    $\Rightarrow d = 27$ Vérification : $d \cdot e \stackrel{?}{\equiv} 1 \mod \phi(n)$

    $d \cdot e = 3 \cdot 27 = 81 = 2 \cdot 40 + 1 \equiv 1 \mod 40$

    $\Rightarrow k_c = \{n, e\} = \{55, 3\}$ $k_d = \{n, d\} = \{55, 27\}$
  - Chiffrement de $M = 19$
    - $C = M^c \mod n = 19^3 \mod 55 = 6859 \mod 55 = 39$
  - Déchiffrement de $C = 39$
    - $M = C^d \mod n = 39^{27} \mod 55 = 39$
    - $39^{27} = 909377887614652551975371341130628025063947 9$

Avantages des chiffres à clé publique

- Pas de confiance mutuelle entre émetteur et récepteur
- Gestion de clé "facile"
  - Répertoire public de clés publiques ou distribution entre pairs
  - La clé privée ne doit "jamais" être transmise
- Possibilité d'utilisations nouvelles : distribution de clés symétriques, signatures, certificats, etc.

Distribution de clés symétriques

- Exemple : Alice génère aléatoirement une clé de session K (symétrique) et la chiffre avec la clé publique de Bob
- Exemple : Diffie-Hellmann
  - Alice (A) et Bob (B) souhaitent communiquer (ex : groupe fini $\mathbb{Z}/p\mathbb{Z}$)

A	$\leftrightarrow$	B	Alice et Bob se mettent d'accord sur un nombre premier p
A	$\leftrightarrow$	B	Alice et Bob conviennent d'une racine primitive g
A	⏲	B	Alice choisi un nombre secret $0 \leq a \leq p-1$
A	$\rightarrow$	B	Alice envoie la valeur $g^a \mod p$ à Bob
A	⏲	B	Bob choisi un nombre secret $0 \leq b \leq p-1$
A	$\leftarrow$	B	Bob envoie la valeur $g^b \mod p$ à Alice
A	⏲	B	Alice calcule la clé secrète $K = (g^b \mod p)^a \mod p$
A	⏲	B	Bob calcule la clé secrète $K = (g^a \mod p)^b \mod p$

  - Eve écoute les transmissions
    - Eve connait p, g, $g^a \mod p$, $g^b \mod p$
    - Peut-il calculer a et b ?
    - $a = \log_g(g^a)$ et $b = \log_g(g^b)$ mod p
    - Problème du logarithme discret
- Est-ce qu'Alice est sûre d'échanger une clé avec Bob ?
  $\Rightarrow$ signature et authentification

Problèmes des chiffres à clé publique

- Calculs complexes : lents ($\backsim$ 1 Mbits/s), clé longue (1024 ou 2048 bits), sauf avec des courbes elliptiques ($\backsim$ 160 bits)
  Records actuels
  - RSA 200, 200 chiffres (2005) : 663 bits (BSI, U.Bonn, CWI)
  - RSA 640/173 (2005) : 4,5 mois à 80 opteron 2,2 GHz (BSI, U.Bonn)
  - Logarithme discret 613 bits (2005) : 17 jours à 64 Itanium2 (Bull, U. Versailles)
  - Certicom ECC2-109 (2004) : 15 mois à 2900 calculateurs
- Problèmes spécifiques
  - Intégrité des répertoires de clés publiques
  - Durée de vie des clés
  - Révocation
  - Nécessité de partager des clés privées ?
  - Limitation des algorithmes, par exemple : chiffrer un petit M par RSA

Factorisation – Défis

www.rsa.com/rsalabs/node.asp?id=2092

www.crypto-world.com/FactorWorld.html

Nombre	Nombre de décimales	Date	Vitesse	Algorithme
C116	116	1990	275 MIPS années	mpqs
RSA-120	120	06/1993	830 MIPS années	mpqs
RSA-129	129	04/1994	5000 MIPS années	mpqs
RSA-130	130	04/1996	1000 MIPS années	gnfs
RSA-140	140	02/1999	2000 MIPS années	gnfs
RSA-155	155	08/1999	8000 MIPS années	gnfs
C158	158	01/2002	3,4 Pentium 1GHz années	gnfs
RSA-160	160	03/2003	2,7 Pentium 1GHz années	gnfs
RSA-576	174	12/2003	13,2 Pentium 1GHz années	gnfs
C176	176	05/2005	48,6 Pentium 1GHz années	gnfs
RSA-200	200	05/2005	121 Pentium 1GHz années 55 Opteron 2,2GHz années	gnfs
RSA-640	193	11/2005	30 Opteron 2,2GHz années	gnfs

Factorisation – Défis *RSA-640*

- Durée : 4,5 mois
- Matérielle : 80 Opterons 2,2GHz
- Vitesse : 30 Opteron 2,2GHz années
- Le nombre : 193 chiffres – 640 bits
- La factorisation

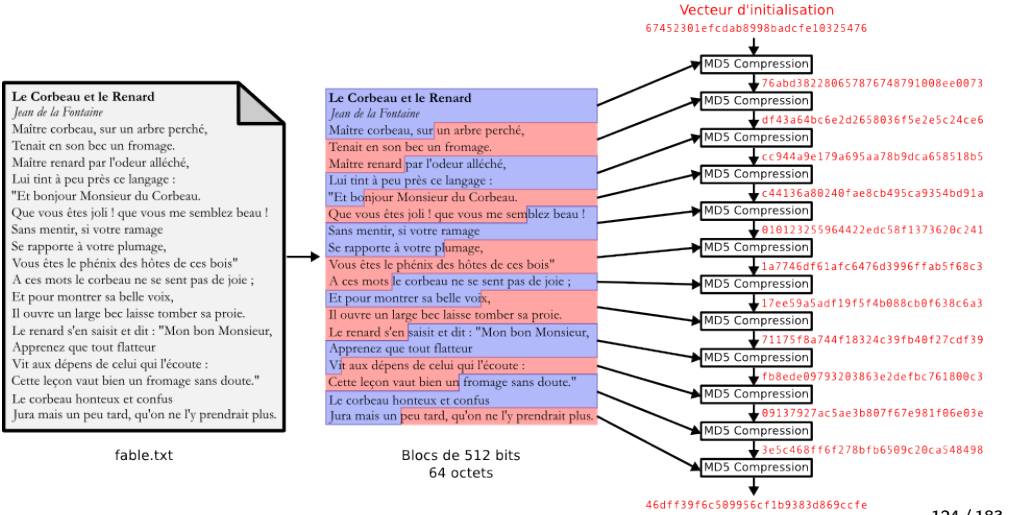
```
310741824049004372135075003588856793003734602284
272754572016194882320644051808150455634682967172 32
867824379162728380334154710731085019195485290073 3
772482278352574238645401469173660247765234660 9
=
163473364580925384844313388386509085984178367003 3
092312181110852389331001045081512121181675115 79
x
190087128166482211312685157393541397547189678996 8
515493666638539088027103802104498957191261465571
```

Fonctions de hachage → empreinte, condensat

- *One-way hash function* $\mathcal{H}$
  - L'empreinte $\mathcal{H}(M)$ est de taille fixe n, quelque soit la longueur de M
  - Si 1 bit de M est changé, environ $n/2$ bits de $\mathcal{H}(M)$ changent
  - Connaissant M, il est facile de calculer $\mathcal{H}(M)$
  - Collisions : $M \neq M', \mathcal{H}(M) = \mathcal{H}(M')$
    $|M|$ non borné et $|\mathcal{H}(M)|$ fixe $\Rightarrow \exists$ un nombre infini de collisions
  - Sécurité : sauf attaque par force brute ($\backsim 2^n$ essais)
    - **Préimage** : connaissant $x < 2^n$, il est "impossible" de trouver M tel que $\mathcal{H}(M) = x$
    - **Seconde préimage** : connaissant M, il est "impossible" de trouver M' tel que $M \neq M'$ et $\mathcal{H}(M) = \mathcal{H}(M')$
    - **Collision** : il est très difficile ($\backsim 2^{n/2}$ essais) de trouver M et M' tel que $M \neq M'$ et $\mathcal{H}(M) = \mathcal{H}(M')$
- Exemples : *DES – CBC* (64 bits), *MD5* (128 bits), *SHA-1* (160 bits)

MD5

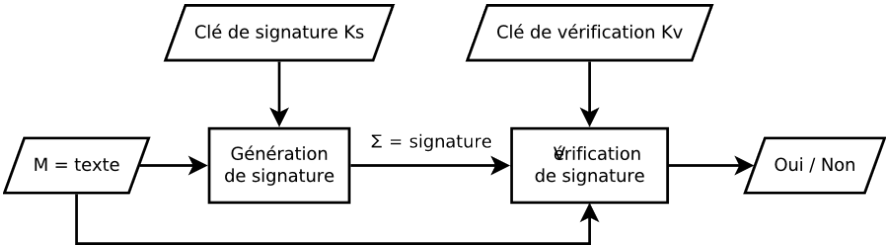
- M est découpé en z blocs de 512 bits, m_1, m_2, …, m_z
  $h_1 = \mathcal{F}(constante, m_1)$; $h_2 = \mathcal{F}(h_1, m_2)$; … ; $h_z = \mathcal{F}(h_{z-1}, m_z) = \mathcal{H}(M)$



Fonctions de hachage – Application : Intégrité

- Communications : contre interception et modification
  Transmettre le message et l'empreinte par des canaux indépendants
- Fichiers : détection de modification (*Tripwire*[10])
  - Sur une machine correcte, calculer les empreintes des fichiers stables (OS, programmes, configuration, etc.) et les stocker de manière sûre (par exemple, chiffrées)
  - Périodiquement, ou en cas de doute, ou au démarrage, recalculer les empreintes et les comparer (sur une machine saine)
- Faiblesse découverte récemment (MD5, etc.)
  - `www.stachliu.com/research_collisions.html`
  - 2006 : collision MD5 en 3/4 d'heure sur Pentium 4 1,6GHz ($\backsim$ 50 bits)
  - 2009 : collision SHA-1 en $\backsim 2^{52}$ essais (théoriquement)
- RIPEMD (128 bits, 160 bits, 256 bits), SHA-256, SHA-512, Whirlpool (512 bits)

Signature (intégrité)



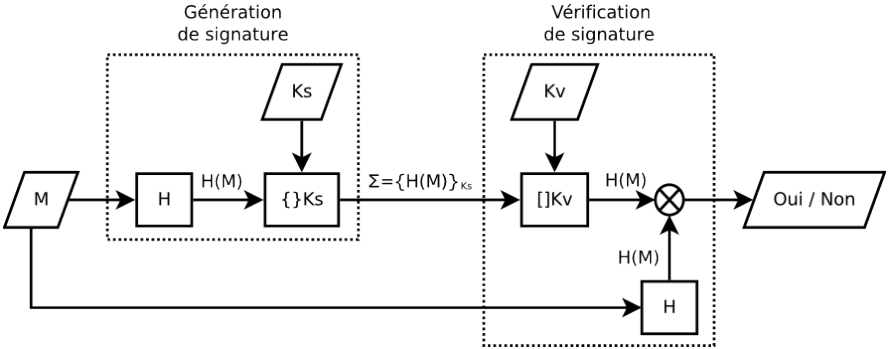
- k_s = clé de signature k_v = clé de vérification
- Intégrité
  - Sans connaître k_s, "impossible" de générer une signature valide
  - Il est "impossible" de trouver k_s, connaissant M et Σ (clair connu)
  - Il est "impossible" de trouver k_s, choisissant M (clair choisi)
- Pratique : Σ est de taille fixe et relativement petit, quelque soit la taille de M

Signature symétriques – $k_s = k_v$: secrètes !

- *MAC* : *Message Authentication Code*
- Exemples
  - *CBC-MAC* : Dernier bloc du *DES* en mode *CBC*
  - $\Sigma = \{\mathcal{H}(M)\}_{k_s} \stackrel{?}{=} \Sigma' = \{\mathcal{H}(M)\}_{k_v}$
  - *H-based MAC* : $\Sigma = \mathcal{H}(k_s \cdot M) \stackrel{?}{\rightarrow} \Sigma' = \mathcal{H}(k_v \cdot M)$
    - Vulnérabilités au attaques de *length extension* : *SHA-1, MD5*
    - $\mathcal{H}(K \cdot M \cdot \text{bad}) = \mathcal{H}(\mathcal{H}(K \cdot M) \cdot \text{bad})$
    - Faiblesse
  - Variante : $\Sigma = \mathcal{H}(k_s \cdot \mathcal{H}(k_s \cdot M) \stackrel{?}{\rightarrow} \Sigma' = \mathcal{H}(k_s \cdot \mathcal{H}(k_s \cdot M)$
- Inconvénients
  - Signataire et vérificateur doivent se faire confiance
  - Répudiation possible $\Rightarrow$ la signature n'est pas valable devant un juge

Signatures à clé publique – $k_s \neq k_v$

- Exemple : *RSA*
  - k_s = clé de signature = clé de chiffrement k_c privée
  - k_v = clé de vérification = clé de déchiffrement k_d publique



Propriétés des signatures à clé publique

- Vérifiables par des tiers : preuve de responsabilité du signataire
  la clé de signature ne doit jamais être transmise
- Peuvent servir à sécuriser les répertoires de clés publiques
  Infrastructure de gestion de clés (*IGC* ou *PKI*)
  - Chaque entrée de répertoire est signée par une *autorité de certification* (*AC* ou *CA*)
  - Les clés publiques des autorités de certification sont dans une répertoire, chacune signée par une *AC* de plus haut niveau, etc.
- Attention : être sûr de ce qu'on signe !
  What you sign is not necessarily what you see

Propriétés des signatures à clé publique

Julius. Caesar
Via Appia 1
Rome, The Roman Empire

May, 22, 2005

To Whom it May Concern:

Alice Falbala fulfilled all the requirements of the Roman Empire intern position. She was excellent at translating roman into her gaul native language, learned very rapidly, and worked with considerable independence and confidence.

Her basic work habits such as punctuality, interpersonal deportment, communication skills, and completing assigned and self-determined goals were all excellent.

I recommend Alice for challenging positions in which creativity, reliability, and language skills are required.

I highly recommend hiring her. If you'd like to discuss her attributes in more detail, please don't hesitate to contact me.

Sincerely,

Julius Caesar

Julius. Caesar
Via Appia 1
Rome, The Roman Empire

May, 22, 2005

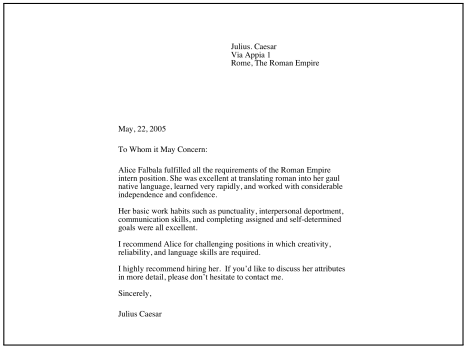
Order:

Alice Falbala is given full access to all confidential and secret information about GAUL.

Sincerely,

Julius Caesar

Propriétés des signatures à clé publique



Julius. Caesar
Via Appia 1
Rome, The Roman Empire

May, 22, 2005

To Whom it May Concern:

Alice Falbala fulfilled all the requirements of the Roman Empire intern position. She was excellent at translating roman into her gaul native language, learned very rapidly, and worked with considerable independence and confidence.

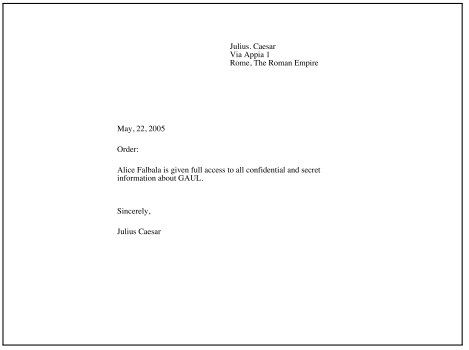
Her basic work habits such as punctuality, interpersonal deportment, communication skills, and completing assigned and self-determined goals were all excellent.

I recommend Alice for challenging positions in which creativity, reliability, and language skills are required.

I highly recommend hiring her. If you'd like to discuss her attributes in more detail, please don't hesitate to contact me.

Sincerely,

Julius Caesar



Julius. Caesar
Via Appia 1
Rome, The Roman Empire

May, 22, 2005

Order:

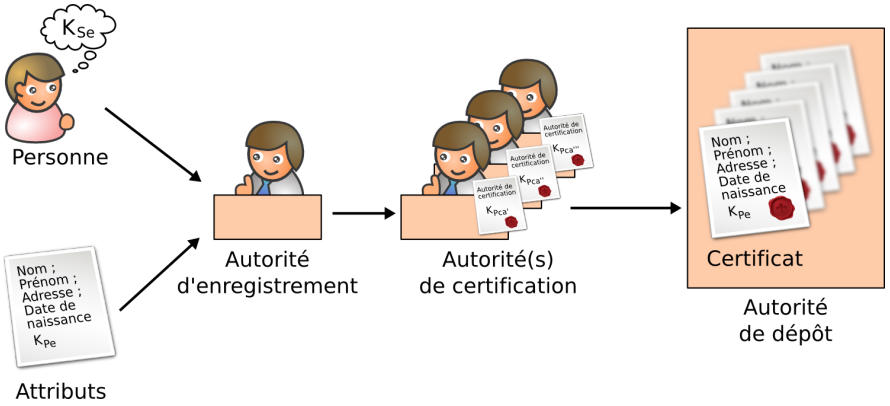
Alice Falbala is given full access to all confidential and secret information about GAUL.

Sincerely,

Julius Caesar

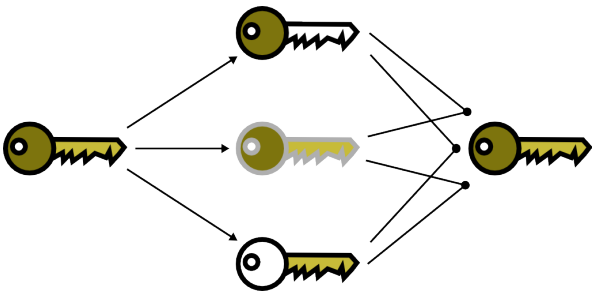
```
$ cat letter_of_rec.ps | openssl md5
a25f7f0b29ee0b3968c860738533a4b9
$ cat order.ps | openssl md5
a25f7f0b29ee0b3968c860738533a4b9
$ diff order.ps letter_of_rec.ps
Binary files order.ps and letter_of_rec.ps differ
```

Certificats et PKI – exemple *X509*

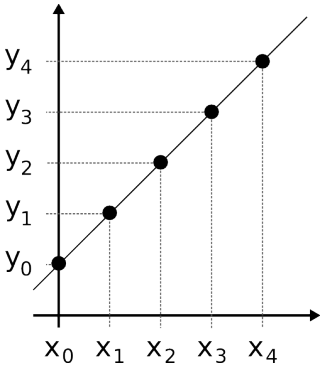


Schémas à seuil

- Stocker K sous la forme d'un ensemble de valeurs K_i (images), telles que :
  - S images permettent de reconstruire le secret (S = seuil)
  - $S - 1$ images n'apportent aucune information
- Si on sait générer N images (avec $N > S$), alors on tolère de perdre jusqu'à $N - S$ images



Schémas à seuil – exemple avec $S = 2$



$$
\begin{aligned}
P(x) = &\quad \{a \cdot a + b\} \\
y_0 = &\quad a \cdot 0 + b \\
y_1 = &\quad a \cdot 1 + b \\
y_2 = &\quad a \cdot 2 + b \\
&\quad \dots \\
y_m = &\quad a \cdot m + b
\end{aligned}
$$

- A partir de deux points quelconque on sait calculer a et b
  - $\Rightarrow$ avec 2 images quelconques, on reconstruit le secret
  - $\Rightarrow$ avec une seule image, on n'a rien

Schémas à seuil – généralisation à polynôme de degré n

- $P(x) = a_0 + a_1 \cdot x + a_2 \cdot x^2 + \ldots + a_n \cdot x^n$
- Si l'on connaît $S = n + 1$ points, on sait recalculer les coefficients du polynôme (S équations à S inconnues) : interpolation polynomiale de Lagrange
- Les calcults se font modulo q, dans un corps de Galois (avec q premier) : $GF(q) = \{0, 1, 2, \ldots, q-1\}$

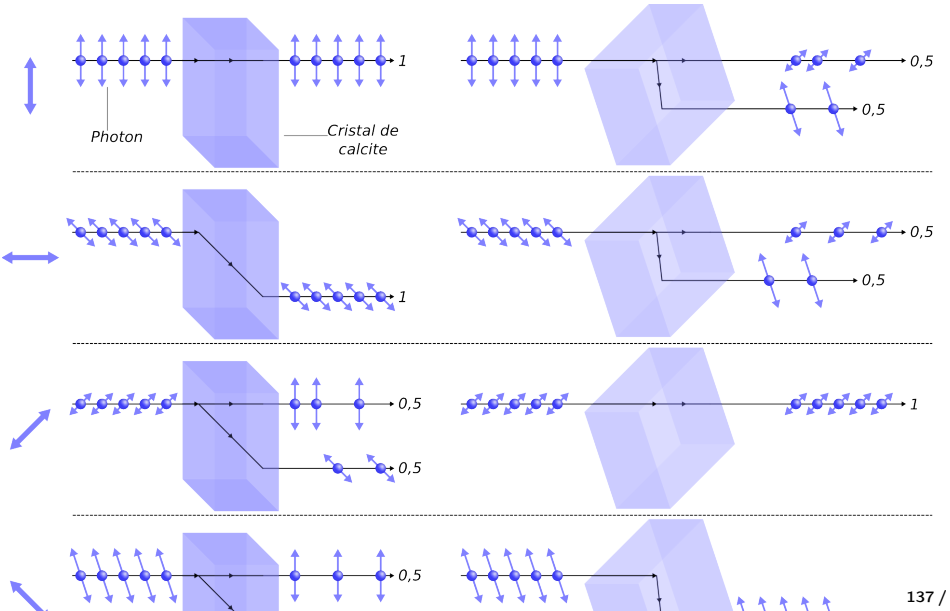
$$P(x) = \sum_{l \leq i \leq k} y_i \prod_{l \leq j \neq i \leq k} \frac{(x - x_j)}{(x_i - x_j)} \mod q$$

Quantum Key Distribution[?]

- Sécurité calculatoire des systèmes cryptographiques
  - *Logarithme discret*, *factorisation en nombre premiers*, etc.
  - Les performances des machines augmentent (ordinateur quantique)

  $\Rightarrow$ Sécurité à long terme menacée
- Le code de Vernam
  - Le seul permettant d'établir un canal inconditionnellement sûr
  - Utilisation du *ou-exclusif*
  - La clé ne doit être utilisée qu'une seule fois
  - Problème majeure : $|K| \geq |M|$
    - Par exemple, si Alice veut communiquer 1 Go de données à Bob, elle doit échanger avec Bob et de manière sûre, une clé de taille 1 Go
- Du côté de la mécanique quantique
  - Théorème de *non-clonage*[16]
    - Impossibilité de cloner des états quantiques inconnus
  - Application aux photons $\rightarrow$ cryptographie quantique

Quantum Key Distribution[?]



Quantum Key Distribution[?]

Correspondances

0 $\Leftrightarrow$ $\nearrow$, $\leftrightarrow$

1 $\Leftrightarrow$ $\searrow$, $\updownarrow$

Alice ($\mathcal{A}$) se prépare à l'envoi								
$\mathcal{A}$ choisit n bits aléatoirement	1	0	1	1	0	0	1	1
$\mathcal{A}$ choisit n polarisateurs aléatoirement	+	×	×	+	×	+	+	×
Bob ($\mathcal{B}$) se prépare à la réception								
$\mathcal{B}$ choisit n polarisateurs aléatoirement	+	×	+	+	×	×	+	×
Alice communique son secret à Bob (canal quantique)								
$\mathcal{A}$ envoie les photons	$\updownarrow$	$\nearrow$	$\searrow$	$\updownarrow$	$\nearrow$	$\leftrightarrow$	$\updownarrow$	$\searrow$
$\mathcal{B}$ détecte les photons	$\updownarrow$	$\nearrow$		$\updownarrow$	$\nearrow$	$\nearrow$	$\updownarrow$	$\searrow$
Alice et Bob comparent leurs connaissances (canal public)								
$\mathcal{B}$ envoie les polarisateurs des photons détectés	+	×		+	×	×	+	×
$\mathcal{A}$ indique les polarisateurs corrects	✓	✓		✓	✓	✗	✓	✓
$\mathcal{A}$ et $\mathcal{B}$ partage la chaîne secrète	1	0		1	0		1	1
Alice et Bob évaluent la probabilité d'avoir été espionné (canal public)								
$\mathcal{B}$ sacrifie 1/3 des bits correctement reçus	1				0			1
$\mathcal{A}$ confirme la validité des bits sacrifiés	✓				✓			✓
Résultat								
$\mathcal{A}$ et $\mathcal{B}$ partagent un secret		0		1			1	

Quantum Key Distribution[?] – avec écoute

Correspondances

0 $\Leftrightarrow$ $\nearrow$, $\leftrightarrow$

1 $\Leftrightarrow$ $\searrow$, $\updownarrow$

Alice ($\mathcal{A}$) se prépare à l'envoi								
$\mathcal{A}$ choisit n bits aléatoirement	1	0	1	1	0	0	1	1
$\mathcal{A}$ choisit n polarisateurs aléatoirement	+	×	×	+	×	+	+	×
Bob ($\mathcal{B}$) se prépare à la réception								
$\mathcal{B}$ choisit n polarisateurs aléatoirement	+	×	+	+	×	×	+	×
Alice communique son secret à Bob (canal quantique)								
$\mathcal{A}$ envoie les photons	$\updownarrow$	$\nearrow$	$\searrow$	$\updownarrow$	$\nearrow$	$\leftrightarrow$	$\updownarrow$	$\searrow$
$\mathcal{B}$ détecte les photons	$\updownarrow$	$\nearrow$		$\updownarrow$	$\nearrow$	$\nearrow$	$\updownarrow$	$\nearrow$
Alice et Bob comparent leurs connaissances (canal public)								
$\mathcal{B}$ envoie les polarisateurs des photons détectés	+	×		+	×	×	+	×
$\mathcal{A}$ indique les polarisateurs corrects	✓	✓		✓	✓	✗	✓	✓
$\mathcal{A}$ et $\mathcal{B}$ partage la chaîne secrète	1	0		1	0		1	0
Alice et Bob évaluent la probabilité d'avoir été espionné (canal public)								
$\mathcal{B}$ sacrifie 1/3 des bits correctement reçus	1				0			0
$\mathcal{A}$ confirme la validité des bits sacrifiés	✓				✓			✗
Résultat								
$\mathcal{A}$ et $\mathcal{B}$ partagent un secret		0		1			1	

Autres fonctions cryptographiques – sujets

- Cryptographie à clés publiques basées sur l'identité
- Stéganographie
- *Watermarking*
- Génération de nombres aléatoires et pseudo-aléatoires
- Générateur de nombres premiers
- Ecrous (*key escrow*)
- Vote
- Horodatage
- Cryptanalyse
- Protocoles

Sommaire

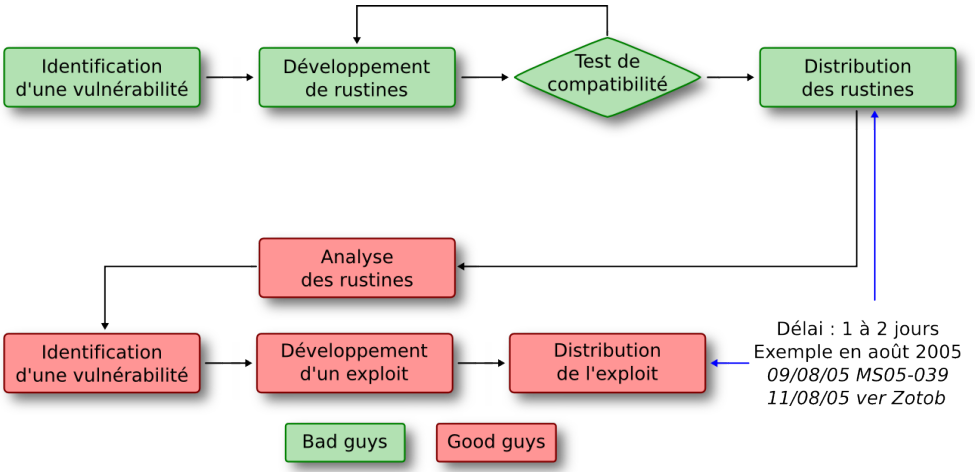
Prévention des vulnérabilités

- Vulnérabilités = fautes de conception ou de configuration
- Les systèmes commerciaux actuels sont trop complexes pour être sans fautes
- Objectifs divergents
  - Disponibilité / sécurité (*TCP/IP*)
  - Rentabilité-efficacité / sécurité
- Il existe des outils pour éviter d'introduire des vulnérabilités classiques (par exemple des débordements de tampons)

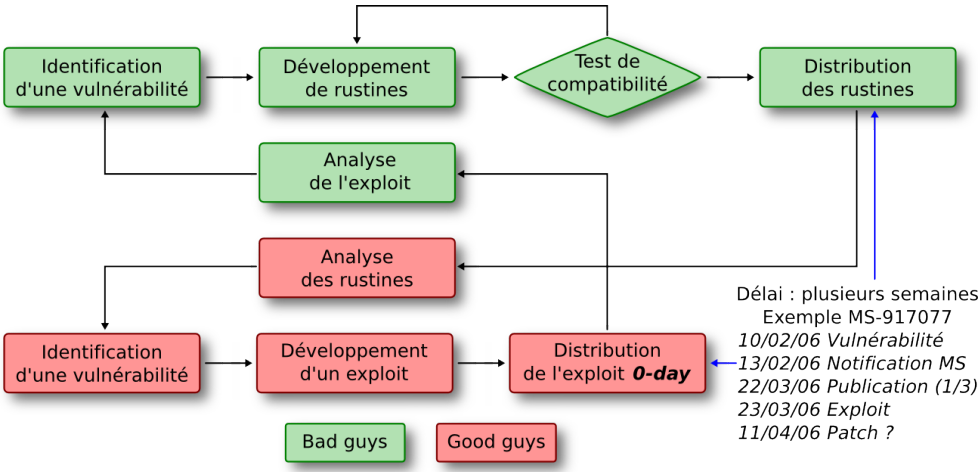
Elimination des vulnérabilités

- Cycle habituel
  - Identification d'une nouvelle vulnérabilité
  - *Exploit*
  - *Patches* (rustines)
  - Nouvelle version
- Mais
  - Nombreuses alertes → quelles sont celles qui sont pertinentes ?
  - Certains *patches* sont imparfaits → élimination d'une fonctionnalité indispensable
  - Certaines applications indispensables ne sont plus compatibles

Cycle de vie des patches



Cycle de vie des exploits



Sommaire

Cloisonnement

- Empêcher tout communication/interaction qui n'est pas nécessaire
  - Isoler les systèmes de développement des systèmes opérationnels, les systèmes de surveillance des systèmes surveillés
  - Fragmenter et disséminer l'information, séparer les pouvoirs
- Pare-feux
  - Filtrer les adresses sources/destination (*IP* + n° port), entrée/sortie
  - Traduction d'adresse (*NAT*)
  - Mandataire d'application (*proxy*) pour vérifier les protocoles d'application
    - Liaison avec *IDS stateful*
    - Option : outil *anti-reconnaissance*, *Intrusion Prevention System* (*IPS*)

Sommaire

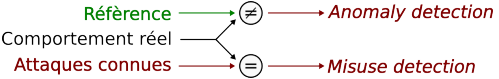
Audit – journalisation

- Enregistrer toutes les opérations liées à la sécurité (réussies ou non)
  - Connexion/déconnexion d'utilisateurs
  - Création/modification/destruction d'informations de sécurité
    - Droits d'accès
    - Mots de passe
    - Enregistrements d'audit
    - . . .
  - Changement de privilèges
- Informations enregistrées
  - Date, heure
  - Identité de l'utilisateur
  - Type d'opération, référence des objets
  - . . .

Sommaire

Détection d'intrusion – *IDS*

- Principe : détection d'erreurs dues à des intrusions
- Deux familles de techniques : analogie avec les détecteurs de virus



  - *Anomaly detection* : par discrimination entre les comportements normaux (utilisateurs non-malveillants) et les comportements anormaux (intrus) : profils statistiques, systèmes experts, systèmes immunitaires, etc.
  - *Misuse detection* : par reconnaissance de signatures correspondant à des attaques connues (*stateless*, *stateful*)
- Implémenté dans chaque calculateur (*host-based IDS*) ou sur des machines observant le réseau (*network-based IDS*)
- Problème
  - Taux de fausses alarmes (*false positives*)
  - Taux de non détection (*false negatives*)
- Les autres mécanismes de détection d'erreurs peuvent aussi être efficaces vis-à-vis des intrusions

Sommaire

Sommaire

Sommaire

Politiques de sécurité

Politique de sécurité

Une politique de sécurité est l'ensemble des lois, règles et pratiques qui régissent la façon dont l'information sensible et les autres ressources sont gérées, protégées et distribuées à l'intérieur d'un système spécifique.[?]

- Modèle de sécurité → formalisme mathématique

Politiques de sécurité

- Objectifs à satisfaire, par exemple :
  - Confidentialité : le dossier médical ne peut être consulté que par le patient et son ou ses médecins traitants
  - Intégrité : un chèque de plus de 1000 € doit être signé par le Président et le Trésorier
  - Disponibilité : si la carte et le *PIN* sont valides, le distributeur de billets doit fournir l'argent dans les 30 secondes
- Règles, par exemple :
  - Un fichier ne peut être lu que par les utilisateurs autorisés par le propriétaire du fichier
  - Un message de type *chèque de plus de 1000 €* n'est valide que s'il est signé par P_1 et T_2 et que les signatures sont valides
  - L'insertion d'une carte lance automatiquement l'action, etc.

Cohérence d'une politique

- La politique est cohérente si, partant d'un état quelconque où les objectifs sont satisfaits, il n'est pas possible d'atteindre, en respectant les règles, un état où ils ne sont plus satisfaits
- Intérêts d'un modèle formel
  - Décrire de manière précise les objectifs et les règles
  - Prouver des propriétés sur la politique (cohérence, complétude, etc.) et sur son implémentation par le système informatique

Politique, protection et contrôle d'accès

- Les règles doivent être mises en œuvre par des mécanismes (matériels, logiciels)
- Facile à imaginer pour les règles du type "*il est permis de . . .*" ou "*il est interdit de . . .*" → mécanismes de protection
  instructions privilégiées, contrôle d'accès à la mémoire, contrôle à l'ouverture de fichiers, etc.
  → autorisation : confidentialité, intégrité
- Difficile pour les règles du type "*il est obligatoire de . . .*" ou "*il est recommandé de . . .*"
  → actions automatiques, gestion des ressouces, etc : intégrité, disponibilité

Politique d'autorisation

- Un sujet a un droit d'accès sur un objet
  ⇔ le sujet est autorisé à exécuter la méthode d'accès sur cet objet
  - Sujet : processus qui s'exécute pour le compte d'un utilisateur
  - Utilisateur : personne physique ou service identifié dans le système
  - Objet : conteneur d'information, défini par un nom, un état et des méthodes, par exemple : fichier, périphérique, processus, etc.

Modèle *HRU*

- L'état de sécurité du système est défini par :
  - D : l'ensemble de tous les droits
  - S : l'ensemble des sujets courants
  - O : l'ensemble des objets courants, $S \subseteq O$
  - A : l'ensemble des droits courants de chaque sujet sur chaque objet
    A est représenté par une matrice avec une ligne par sujet s_i et une colonne pour chaque objet o_j
    $A_{ij} = d_{ij}$ avec $d_{ij} \subseteq D$

$$(s_i, o_j, d_k) \text{ est vrai} \Leftrightarrow s_i \text{ a le droit } d_k \text{ sur } o_j$$
$$d_{ij} = \{d_k \in D | (s_i, o_j, d_k)\}$$

Politique d'autorisation discrétionnaires

DAC : *Discretionary Access Control*

- Les droits d'accès à chaque information sont manipulés par le responsable de l'information (généralement le propriétaire), à sa discrétion
- Les droits peuvent être accordées à chaque utilisateur individuellement ou à des groupes d'utilisateurs ou les deux

Politique d'autorisation discrétionnaires

- Exemple : protection des fichiers UNIX
  - Règles
    1. Un utilisateur peut créer librement des fichiers dont il devient propriétaire
       $(A, F, \text{créer}) \xrightarrow{A} (A, F, \text{propriétaire}) \wedge (A, F, \text{écrire}) \wedge (A, F, \text{lire})$
    2. Les droits d'accès à un fichier sont définis librement par le propriétaire : par exemple, il peut décider quels utilisateurs sont autorisés à lire le fichier
       $(A, F, \text{propriétaire}) \xrightarrow{A} (B, F, \text{lire})$
  - Objectif
    - *Un utilisateur non-autorisé à lire un fichier ne peut obtenir aucune information contenue dans le fichier (même avec la complicité d'un utilisateur autorisé)* → impossible à garantir
  - Exemple
    3. $S = \{s_1, s_2, s_3\}$
    4. $O = \{f_1, f_2\}$
    5. $D = \{\text{propriétaire}, \text{lire}, \text{écrire}\}$
    6. $A = \{(s_1, f_1, \text{propriétaire})\}$

(2 et 6) 7. $(s_1, f_1, \text{propriétaire}) \xrightarrow{s_1} (s_2, f_1, \text{lire})$

(1 et 2) 8. $(s_2, f_2, \text{créer}) \xrightarrow{s_2} (s_2, f_2, \text{écrire}) \wedge (s_3, f_2, \text{lire})$

(7 et 8) 9. $(s_2, f_1, \text{lire}) \wedge (s_2, f_2, \text{écrire}) \wedge (s_3, f_2, \text{lire}) \xrightarrow{s_2} (s_3, k(f_1), \text{lire})$

Inconvénient des politiques *DAC*

- Possibilité d'abus de pouvoir
  (par malveillance ou par maladresse)
  - S'il est possible pour un utilisateur légitime d'accéder à certains objets ou d'en modifier les droits d'accès, il est possible qu'un *Cheval de Troie* en fasse de même
  - Si un utilisateur a le droit de lire une information, il a (en général) automatiquement le droit de la divulger à n'importe qui
  - Il est difficile de corriger les effets d'une divulgation

Politique d'autorisation obligatoires

MAC : *Mandatory Access Control*

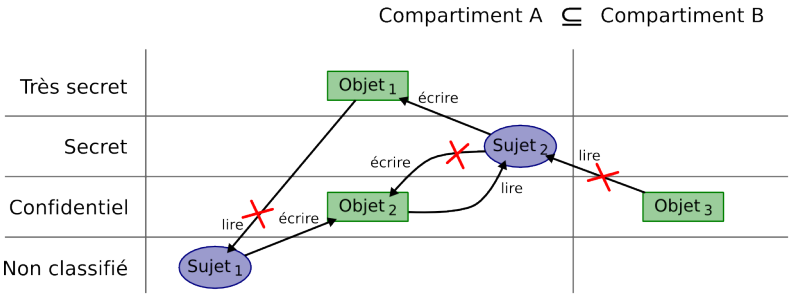
- Des règles incontournables sont imposées, en plus des règles discrétionnaires
- Exemple : politique de confidentialité multi-niveau *militaire*
  Des classes sont assignées aux utilisateurs (habilitation) et aux objets (classification)
  Une classe est définie par :

Un niveau (ordonné)	Un compartiment = {catégories}
Non-classifié	Cryptographie
Confidentiel	Nucléaire
Secret	OTAN
Très secret	Irak
	. . .

Politique de *Bell-LaPadula* (condifentialité)

- A chaque sujet s_i correspond une habilitation $h(s_i)$ avec un niveau n_i et un compartiment Σ_i
- A chaque objet o_j correspond une classification $c(o_j)$ avec un niveau n_j et un compartiment Σ_j
- Règle simple :
  $(s_i, o_j, \text{lire}) \Rightarrow n_j \leq n_i \wedge \Sigma_j \subseteq \Sigma_i \qquad (h(s_i)\ \textit{domine}\ c(o_j))$
- Règle étoile :
  $(s_i, o_j, \text{lire}) \wedge (s_i, o_k, \text{écrire}) \Rightarrow n_j \leq n_k \wedge \Sigma_j \subseteq \Sigma_k \qquad (c(o_k)\ \textit{domine}\ c(o_j))$
- Propriété :
  Si $h(s_n)$ *ne domine pas* $c(o_i)$, il n'existe pas de suite telle que
  $(s_l, o_i, \text{lire}) \wedge (s_l, o_j, \text{écrire}) \wedge (s_m, o_j, \text{lire}) \wedge \ldots \wedge (s_x, o_k, \text{écrire}) \wedge (s_n, o_k, \text{lire})$
  Interdire à tout sujet d'obtenir des informations d'un objet de niveau supérieur à son habilitation
  $\Rightarrow$ Pas de fuite d'information possible

Inconvénients de Bell-LaPadula

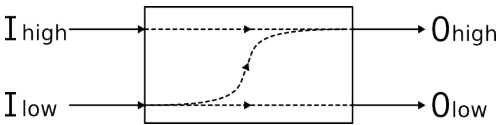


- Surclassification : au fur et à mesure que l'information est traitée, sa classification augmente

⇒ *Trusted process* pour déclassifier

Autres politiques de confidentialité

- *Muraille de Chine*
  Chez les agents de change (conflits d'intérêt)
- Modèle de non interférence : O_{low} ne dépend pas de I_{high}

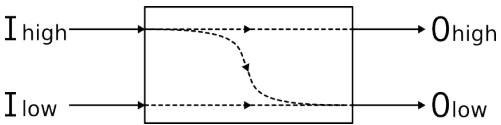


Politique de *Biba* (intégrité)

- Multiple niveaux d'intégrité (crédibilité, vérification, etc.)
  - A chaque sujet s_i correspond un niveau $is(s_i)$
  - A chaque objet o_j correspond un niveau $io(o_j)$
- Règles
  - $(s_i, o_j, \text{observer}) \Rightarrow is(s_i) \leq io(o_j)$
  - $(s_i, o_j, \text{modifier}) \Rightarrow io(o_j) \leq is(s_i)$
  - $(s_i, s_j, \text{invoquer}) \Rightarrow is(s_j) \leq is(s_i)$
- Propriété : empêcher la *contamination* des niveaux élevés : diffusion de fausses informations, propagation d'erreur, etc.
- Inconvénient : dégradation progressive des niveaux d'intégrité

Autres politiques d'intégrité

- Politique de *Clark-Wilson* (transactions financières)
  - Deux niveaux d'intégrité
    - *UDI* (données non contraintes)
    - *CDI* (données contraintes), vérifiables par des *IVP* (*Integrity Verification Procedures*), ne pouvant être manipulées que par des *TP* (*Transformation Procedures*) certifiées
  - Règles : listes de relations autorisées
    - *CDI* $\leftrightarrow$ *TP*, Utilisateurs $\leftrightarrow$ *TP* (avec éventuellement *separation of duty*), *UDI* $\leftrightarrow$ *CDI* par (*TP* + *IVP*)
- Modèle de non-interférence : O_{high} ne dépend pas de I_{low}



Politiques basées sur les rôles

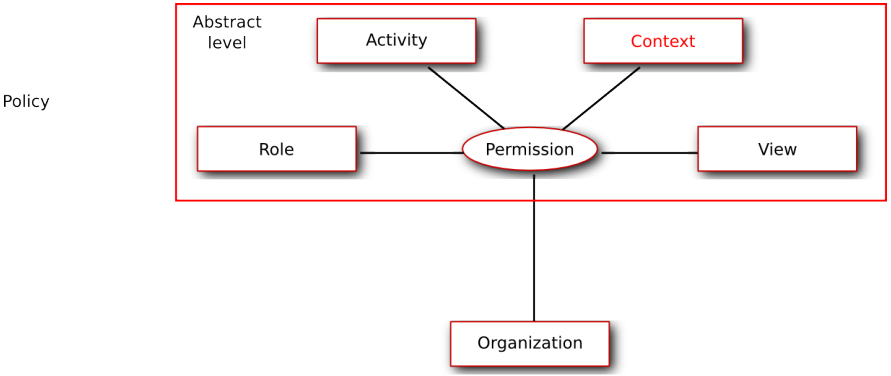
RBAC : *Role-Based Access Control*

- On définit des rôles, pour représenter des fonctions dans l'organisation
- On associe à chaque rôle les privilèges (ensemble de droits) nécessaires pour remplir la fonction
- On associe à chaque utilisateur le(s) rôle(s) qu'il peut jouer dans l'organisation
- Administration facilitée
  - Les rôles et leurs privilèges changenent rarement
  - Il suffit d'identifier les rôles que peut jouer un utilisateur

OrBAC – Organization-Based Access Control

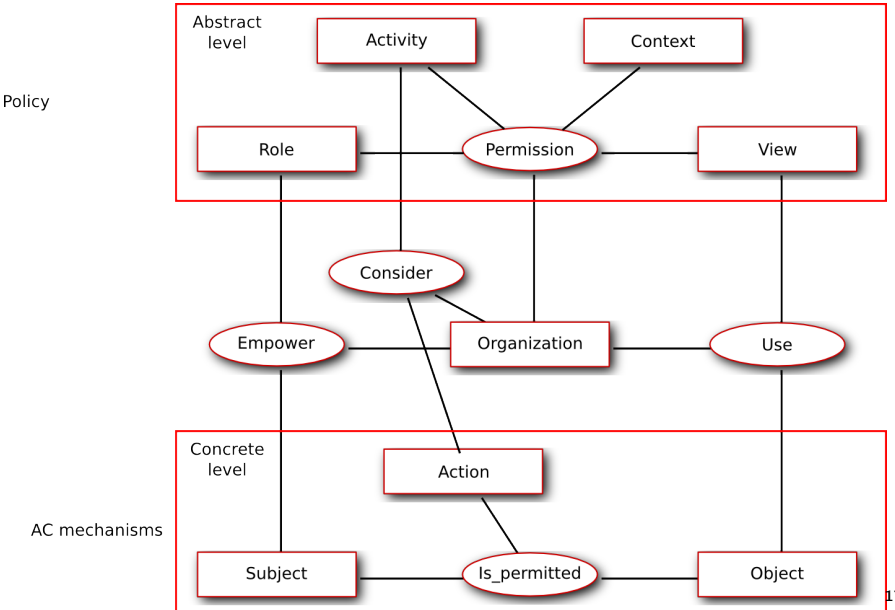
- Modèle conçu dans *MP6*
  Modèles et politiques de sécurité des systèmes d'information et de communication en santé et social
  Ernst & Young, *IRIT*, *LAAS-CNRS*, *ONERA*, *ENST-Bretagne*, etc.
- Abstractions
  - User → rôle
  - Objet → vue
  - Action → activité
- Liaisons entre niveaux abstrait (politique) et concret (mécanismes de contrôle d'accès) : définies par l'organisation
- Règles
  - Définies au niveau abstrait
  - Permissions/interdictions + obligations
  - Validées par le contexte (concret)

OrBAC – Organization-Based Access Control



- La politique est définie au niveau abstrait → définition de règles
  - Permission (Organization, Role, Activity, View, B(context))
  - Interdiction (Organization, Role, Activity, View, B(context))
  - Obligation (Organization, Role, Activity, View, B(context))

OrBAC – Organization-Based Access Control



Sommaire

Authentification des utilisateurs

Nécessaire pour l'autorisation et l'audit

- Authentification = identification + vérification de l'identité
- Identité = information (non confidentielle) spécifique à une personne, connue au moins par elle et par le vérificateur : nom, numéro, etc.
- Vérification de la correspondance entre l'identité et la personne, en utilisant :
  - Quelque chose qu'elle connaît (mot de passe, informations personnelles, etc.) ou qu'elle sait faire (reconnaissance de forme, association d'idées, etc.)
  - Quelque chose qu'elle possède : badge, carte à puce, etc.
  - Quelque chose qui lui est propre (biométrie) : empreinte digitale, signature, voix, fond de l'œil, iris, forme de la main, etc.
  - Ou plusieurs de ces moyens : carte à puce + PIN, etc.

Méthodes de vérification

- Secret partagé entre la personne et le vérificateur (mot de passe, informations personnelles, etc.)
- Secret correspondant à une caractéristique biométrique (stockée par le vérificateur ou non), non falsifiable, non rejouable
- Secret connu par la personne, vérifié par des informations ou protocoles publics (sans apport de connaissance, *zero-knowledge*

Qualité de l'authentification

- La qualité des systèmes d'authentification dépend du taux d'acceptation à tort (*false acceptance rate*, *FAR*) et tu taux de rejet à tort (*false rejection rate*, *FRR*)

  Empreinte digitale $FAR \approx 10^{-6}$ $FRR \approx 10^{-3}$ (SAGEM, Compaq, NEC)
  Iris $FAR \approx 10^{-12}$ $FRR \approx 10^{-4}$ (Sensar, IriScan)

- Il faut distinguer si la victime dont on prend l'identité est consentante (par exemple, transmet volontairement son mot de passe) ou non : supériorité des systèmes biométriques
- Mais, les systèmes biométtriques ont des limitations : falsification (prothèses), handicapés, acceptation sociale, difficulté de révocation, etc.

Références I

National institute of standards and technology.
Site Internet.
http://www.itl.nist.gov/lab/bulletns/bltnjun06.htm.

Rajat Chakraborty, Seetharam Narasimhan, and Swarup Bhunia.
Hardware trojan : Threats and emerging solutions.
pages 166 – 171, 12 2009.

Peng Cheng, Ibrahim Bagci, Utz Roedig, and Jeff Yan.
Sonarsnoop : active acoustic side-channel attacks.
International Journal of Information Security, 07 2019.

Références II


Taher El Gamal.
A public key cryptosystem and a signature scheme based on discrete logarithms.
In Proceedings of CRYPTO 84 on Advances in cryptology, pages 10–18, New York, NY, USA, 1985. Springer-Verlag New York, Inc.


Daniel Genkin, Itamar Pipman, and Eran Tromer.
Get your hands off my laptop : Physical side-channel key-extraction attacks on pcs.
In Lejla Batina and Matthew Robshaw, editors, Cryptographic Hardware and Embedded Systems – CHES 2014, pages 242–260, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.


Daniel Gruss, Clémentine Maurice, and Stefan Mangard.
Rowhammer.js : A remote software-induced fault attack in javascript, 2015.

Références III


J. P. Blanquart A. Costes Y. Crouzet Y. Deswarte J. C. Fabre H. Guillermain M. Kaâniche K. Kanoun C. Mazet D. Powell C. Rabéjac P. Thévenod J.-C. Laprie, J. Arlat.
Guide de la sûreté de fonctionnement.
Cépadues Editions, 1996.


Mohamed Kaaniche.
Evaluation de la sûreté de fonctionnement informatique. fautes physiques, fautes de conception, malveillances.
02 1999.


Jonathan Katz, Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone.
Handbook of applied cryptography.
CRC press, 1996.

Références IV

Gene H. Kim and Eugene H. Spafford.
The design and implementation of tripwire : a file system integrity checker.
In CCS '94 : Proceedings of the 2nd ACM Conference on Computer and communications security, pages 18–29, New York, NY, USA, 1994. ACM.

Ben Nassi, Yaron Pirutin, Adi Shamir, Yuval Elovici, and Boris Zadov.
Lamphone : Real-time passive sound recovery from light bulb vibrations.
Cryptology ePrint Archive, Report 2020/708, 2020.
https://eprint.iacr.org/2020/708.

Colin Percival.
Cache missing for fun and profit.
In Proc. of BSDCan 2005, 2005.

Références V


R. L. Rivest, A. Shamir, and L. Adleman.
A method for obtaining digital signatures and public-key cryptosystems.
Commun. ACM, 21(2) :120–126, 1978.


Huahong Tu, Adam Doupé, Ziming Zhao, and Gail-Joon Ahn.
Users really do answer telephone scams.
In 28th USENIX Security Symposium (USENIX Security 19), pages 1327–1340, Santa Clara, CA, August 2019. USENIX Association.


Xiaoyun Wang, Dengguo Feng, Xuejia Lai, and Hongbo Yu.
Collisions for hash functions MD4, MD5, HAVAL–128 and RIPEMD, 2004.
URL : `http://eprint.iacr.org/2004/199/`.


W. K. Wootters and W. H. Zurek.
A single quantum cannot be cloned.
Nature, 299 :802–803, oct 1982.

Références VI

Zhi Xu.
Abusing notification services on smartphones for phishing and spamming.
In 6th USENIX Workshop on Offensive Technologies (WOOT 12), Bellevue, WA, August 2012. USENIX Association.