L3 Informatique, FST, Université de Lorraine
Semestre 6, printemps 2024

Introduction à la sécurité et à la cryptographie

Marine Minier
marine.minier@loria.fr

Les enseignant·e·s

- Cours : Marine Minier (Marine.Minier@loria.fr)

Les enseignant·e·s

- Cours : Marine Minier (Marine.Minier@loria.fr)
- Travaux dirigés :
  - TD1 : MM
  - TD2 : Sébastien Duval (sebastien.duval@loria.fr)

Les enseignant·e·s

- Cours : Marine Minier (Marine.Minier@loria.fr)
- Travaux dirigés :
  - TD1 : MM
  - TD2 : Sébastien Duval (sebastien.duval@loria.fr)
- Travaux pratiques :
  - TP1 : MM
  - TP2 : SD
  - TP3 : Marie Bolzer

Les enseignant·e·s

- Cours : Marine Minier (Marine.Minier@loria.fr)
- Travaux dirigés :
  - TD1 : MM
  - TD2 : Sébastien Duval (sebastien.duval@loria.fr)
- Travaux pratiques :
  - TP1 : MM
  - TP2 : SD
  - TP3 : Marie Bolzer
- Il est toujours possible de demander un rendez-vous par e-mail à vos enseignant·e·s si vous souhaitez les rencontrer

Objectifs du cours

- Initiation à la cryptographie :
  - Comprendre la cryptographie à travers son histoire
  - Comprendre et utiliser quelques algorithmes les plus courants

Objectifs du cours

- Initiation à la cryptographie :
  - Comprendre la cryptographie à travers son histoire
  - Comprendre et utiliser quelques algorithmes les plus courants
- Initiation à la sécurité :
  - Comprendre les risques et enjeux de la sécurité des réseaux et des systèmes
  - Vulnérabilités humaines, systèmes, réseaux
  - Quelques exemples d'utilisation de la cryptographie dans la pratique de la sécurité : SSL/TLS et GnuPG

Objectifs du cours

- Initiation à la cryptographie :
  - Comprendre la cryptographie à travers son histoire
  - Comprendre et utiliser quelques algorithmes les plus courants
- Initiation à la sécurité :
  - Comprendre les risques et enjeux de la sécurité des réseaux et des systèmes
  - Vulnérabilités humaines, systèmes, réseaux
  - Quelques exemples d'utilisation de la cryptographie dans la pratique de la sécurité : SSL/TLS et GnuPG
- Modalités de contrôle des connaissances :
  - Devoir à la maison (40% de la note du module)
  - Examen final (60% de la note du module)

Première partie

Introduction à la sécurité

(Slides de Marion Videau)

Sécurité, qu'es aquò ?

Sécurité, qu'es aquò ?

- D'après le TLFi :
  Situation objective, reposant sur des conditions matérielles, économiques, politiques, qui entraîne l'absence de dangers pour les personnes ou de menaces pour les biens et qui détermine la confiance.

Sécurité, qu'es aquò ?

- D'après le TLFi :
  Situation objective, reposant sur des conditions matérielles, économiques, politiques, qui entraîne l'absence de dangers pour les personnes ou de menaces pour les biens et qui détermine la confiance.

Sécurité, qu'es aquò ?

- D'après le TLFi :
  Situation objective, reposant sur des conditions matérielles, économiques, politiques, qui entraîne l'absence de dangers pour les personnes ou de menaces pour les biens et qui détermine la confiance.
- De qui, de quoi parle-t-on ici ?

Sécurité, qu'es aquò ?

- D'après le TLFi :
  Situation objective, reposant sur des conditions matérielles, économiques, politiques, qui entraîne l'absence de dangers pour les personnes ou de menaces pour les biens et qui détermine la confiance.
- De qui, de quoi parle-t-on ici ?
  - sécurité des systèmes d'information (SSI)
  - sécurité des systèmes et des réseaux (SSR)

Sécurité, qu'es aquò ?

- D'après le TLFi :
  Situation objective, reposant sur des conditions matérielles, économiques, politiques, qui entraîne l'absence de dangers pour les personnes ou de menaces pour les biens et qui détermine la confiance.
- De qui, de quoi parle-t-on ici ?
  - sécurité des systèmes d'information (SSI)
  - sécurité des systèmes et des réseaux (SSR)
- Enjeu principal : sécurité de l'**information**

Objectifs de la SSI

Objectifs de la SSI

- Confidentialité : garantit que l'accès aux données n'est possible que pour les personnes dûment autorisées à les connaître

Objectifs de la SSI

- Confidentialité : garantit que l'accès aux données n'est possible que pour les personnes dûment autorisées à les connaître
- Intégrité : garantit que les fonctions et données sensibles ne sont pas altérées, et conservent toute leur pertinence

Objectifs de la SSI

- Confidentialité : garantit que l'accès aux données n'est possible que pour les personnes dûment autorisées à les connaître
- Intégrité : garantit que les fonctions et données sensibles ne sont pas altérées, et conservent toute leur pertinence
- Disponibilité : garantit qu'une ressource sera accessible au moment précis où quelqu'un souhaitera s'en servir

Objectifs de la SSI

- Confidentialité : garantit que l'accès aux données n'est possible que pour les personnes dûment autorisées à les connaître
- Intégrité : garantit que les fonctions et données sensibles ne sont pas altérées, et conservent toute leur pertinence
- Disponibilité : garantit qu'une ressource sera accessible au moment précis où quelqu'un souhaitera s'en servir
- Authentification : vérifie qu'une entité est bien celle qu'elle prétend être

Objectifs de la SSI

- Confidentialité : garantit que l'accès aux données n'est possible que pour les personnes dûment autorisées à les connaître
- Intégrité : garantit que les fonctions et données sensibles ne sont pas altérées, et conservent toute leur pertinence
- Disponibilité : garantit qu'une ressource sera accessible au moment précis où quelqu'un souhaitera s'en servir
- Authentification : vérifie qu'une entité est bien celle qu'elle prétend être
- Non-répudiation : interdit à une entité de pouvoir nier avoir pris part à une action (fortement lié à la notion d'imputabilité)

Objectifs de la SSI

- Confidentialité : garantit que l'accès aux données n'est possible que pour les personnes dûment autorisées à les connaître
- Intégrité : garantit que les fonctions et données sensibles ne sont pas altérées, et conservent toute leur pertinence
- Disponibilité : garantit qu'une ressource sera accessible au moment précis où quelqu'un souhaitera s'en servir
- Authentification : vérifie qu'une entité est bien celle qu'elle prétend être
- Non-répudiation : interdit à une entité de pouvoir nier avoir pris part à une action (fortement lié à la notion d'imputabilité)

Source : Pierre Lasbordes, *La sécurité des systèmes d'information : un enjeu majeur pour la France*, 2006

Menaces, vulnérabilités, attaques et risque

Menaces, vulnérabilités, attaques et risque

- Menace : événement potentiel, malveillant ou pas, qui pourrait nuire à une ressource du système d'information

Menaces, vulnérabilités, attaques et risque

- Menace : événement potentiel, malveillant ou pas, qui pourrait nuire à une ressource du système d'information
- Vulnérabilité : faiblesse dans le système d'information qui rend possible une menace

Menaces, vulnérabilités, attaques et risque

- Menace : événement potentiel, malveillant ou pas, qui pourrait nuire à une ressource du système d'information
- Vulnérabilité : faiblesse dans le système d'information qui rend possible une menace
- Attaque : action malveillante qui exploite une vulnérabilité afin d'exécuter une menace

Menaces, vulnérabilités, attaques et risque

- Menace : événement potentiel, malveillant ou pas, qui pourrait nuire à une ressource du système d'information
- Vulnérabilité : faiblesse dans le système d'information qui rend possible une menace
- Attaque : action malveillante qui exploite une vulnérabilité afin d'exécuter une menace
- Contre-mesure : système de défense mis en place pour s'opposer à ou prévenir une menace

Menaces, vulnérabilités, attaques et risque

- Menace : événement potentiel, malveillant ou pas, qui pourrait nuire à une ressource du système d'information
- Vulnérabilité : faiblesse dans le système d'information qui rend possible une menace
- Attaque : action malveillante qui exploite une vulnérabilité afin d'exécuter une menace
- Contre-mesure : système de défense mis en place pour s'opposer à ou prévenir une menace
- Évaluation du risque :

$$\text{Risque} = \frac{\text{Menaces} \times \text{Vulnérabilités}}{\text{Contre-mesures}}$$

Il n'y a pas de sécurité absolue

Il n'y a pas de sécurité absolue

- Asymétrie attaque / défense
  - Les moyens de l'« attaquant » sont potentiellement sans limite : celui-ci n'est pas contraint par la légalité
  - Les moyens du « défenseur » sont très contraints : il doit tout imaginer sans pouvoir riposter (pas de légitime défense)

Il n'y a pas de sécurité absolue

- Asymétrie attaque / défense
  - Les moyens de l'« attaquant » sont potentiellement sans limite : celui-ci n'est pas contraint par la légalité
  - Les moyens du « défenseur » sont très contraints : il doit tout imaginer sans pouvoir riposter (pas de légitime défense)
- Menaces à tous les niveaux
  - Utilisateur : insouciance et négligence
  - Personne malveillante : intrusion au sein du système
  - Programme malveillant : porte dérobée, perte de données
  - Sinistre : vol, incendie

Il n'y a pas de sécurité absolue

- Asymétrie attaque / défense
  - Les moyens de l'« attaquant » sont potentiellement sans limite : celui-ci n'est pas contraint par la légalité
  - Les moyens du « défenseur » sont très contraints : il doit tout imaginer sans pouvoir riposter (pas de légitime défense)
- Menaces à tous les niveaux
  - Utilisateur : insouciance et négligence
  - Personne malveillante : intrusion au sein du système
  - Programme malveillant : porte dérobée, perte de données
  - Sinistre : vol, incendie
    - Incendie des serveurs OVH de Strasbourg en 2021

Il n'y a pas de sécurité absolue

- Asymétrie attaque / défense
  - Les moyens de l'« attaquant » sont potentiellement sans limite : celui-ci n'est pas contraint par la légalité
  - Les moyens du « défenseur » sont très contraints : il doit tout imaginer sans pouvoir riposter (pas de légitime défense)
- Menaces à tous les niveaux
  - Utilisateur : insouciance et négligence
  - Personne malveillante : intrusion au sein du système
  - Programme malveillant : porte dérobée, perte de données
  - Sinistre : vol, incendie
    - Incendie des serveurs OVH de Strasbourg en 2021
- La sécurité résulte toujours d'un compromis
  - Besoin de protection
  - Besoin opérationnel (utilisabilité, coopérations)
  - Resources financières et techniques limitées

Il n'y a pas de sécurité absolue

- Asymétrie attaque / défense
  - Les moyens de l'« attaquant » sont potentiellement sans limite : celui-ci n'est pas contraint par la légalité
  - Les moyens du « défenseur » sont très contraints : il doit tout imaginer sans pouvoir riposter (pas de légitime défense)
- Menaces à tous les niveaux
  - Utilisateur : insouciance et négligence
  - Personne malveillante : intrusion au sein du système
  - Programme malveillant : porte dérobée, perte de données
  - Sinistre : vol, incendie
    - Incendie des serveurs OVH de Strasbourg en 2021
- La sécurité résulte toujours d'un compromis
  - Besoin de protection
  - Besoin opérationnel (utilisabilité, coopérations)
  - Resources financières et techniques limitées
- Accepter l'existence d'un risque résiduel et évaluer son importance

Motivations et cibles (I)

Motivations et cibles (I)

- Des attaques aux motivations très variées

Motivations et cibles (I)

- Des attaques aux motivations très variées
  - Ludique : amusement, curiosité, défi, réputation

Motivations et cibles (I)

- Des attaques aux motivations très variées
  - Ludique : amusement, curiosité, défi, réputation
    - hackers lambda

Motivations et cibles (I)

- Des attaques aux motivations très variées
  - Ludique : amusement, curiosité, défi, réputation
    - hackers lambda
  - Idéologie (voire terrorisme) : vandalisme, déni de service

Motivations et cibles (I)

- Des attaques aux motivations très variées
  - Ludique : amusement, curiosité, défi, réputation
    - hackers lambda
  - Idéologie (voire terrorisme) : vandalisme, déni de service
    - 2007, 1er exemple de cyberterrorisme/cyberguerre en Estonie, déplacement de statue de soldat soviétique à Tallinn (Estonie) $\rightarrow$ vague de cyberattaques sans précédent orchestrées par la Russie via des botnets, sites gouvernementaux et banques ciblées, attaques de type déni de service

Motivations et cibles (I)

- Des attaques aux motivations très variées
  - Ludique : amusement, curiosité, défi, réputation
    - hackers lambda
  - Idéologie (voire terrorisme) : vandalisme, déni de service
    - 2007, 1er exemple de cyberterrorisme/cyberguerre en Estonie, déplacement de statue de soldat soviétique à Tallinn (Estonie) $\rightarrow$ vague de cyberattaques sans précédent orchestrées par la Russie via des botnets, sites gouvernementaux et banques ciblées, attaques de type déni de service
    - 2015, cyberattaque contre TV5Monde, site web défiguré par des messages pour l'état islamique

Motivations et cibles (I)

- Des attaques aux motivations très variées
  - Ludique : amusement, curiosité, défi, réputation
    - hackers lambda
  - Idéologie (voire terrorisme) : vandalisme, déni de service
    - 2007, 1er exemple de cyberterrorisme/cyberguerre en Estonie, déplacement de statue de soldat soviétique à Tallinn (Estonie) $\rightarrow$ vague de cyberattaques sans précédent orchestrées par la Russie via des botnets, sites gouvernementaux et banques ciblées, attaques de type déni de service
    - 2015, cyberattaque contre TV5Monde, site web défiguré par des messages pour l'état islamique
  - Cupidité : vol de données bancaires, extorsion (*ransomware*)

Motivations et cibles (I)

- Des attaques aux motivations très variées
  - Ludique : amusement, curiosité, défi, réputation
    - hackers lambda
  - Idéologie (voire terrorisme) : vandalisme, déni de service
    - 2007, 1er exemple de cyberterrorisme/cyberguerre en Estonie, déplacement de statue de soldat soviétique à Tallinn (Estonie) $\rightarrow$ vague de cyberattaques sans précédent orchestrées par la Russie via des botnets, sites gouvernementaux et banques ciblées, attaques de type déni de service
    - 2015, cyberattaque contre TV5Monde, site web défiguré par des messages pour l'état islamique
  - Cupidité : vol de données bancaires, extorsion (*ransomware*)
    - Depuis 2012, explosion des rançongiciels en parallèle des échanges de cryptomonnaies type bitcoin Petya WannaCry, sur kaspersky.fr

Motivations et cibles (I)

- Des attaques aux motivations très variées
  - Ludique : amusement, curiosité, défi, réputation
    - hackers lambda
  - Idéologie (voire terrorisme) : vandalisme, déni de service
    - 2007, 1er exemple de cyberterrorisme/cyberguerre en Estonie, déplacement de statue de soldat soviétique à Tallinn (Estonie) $\rightarrow$ vague de cyberattaques sans précédent orchestrées par la Russie via des botnets, sites gouvernementaux et banques ciblées, attaques de type déni de service
    - 2015, cyberattaque contre TV5Monde, site web défiguré par des messages pour l'état islamique
  - Cupidité : vol de données bancaires, extorsion (*ransomware*)
    - Depuis 2012, explosion des rançongiciels en parallèle des échanges de cryptomonnaies type bitcoin Petya WannaCry, sur kaspersky.fr
    - Une vidéo sur les rançongiciels : `https://www.lemonde.fr/pixels/video/2021/02/17/comment-fonctionnent-les-cyberattaques-aux-rancongiciels-qui-ont-cible-des-hopitaux_6070246_4408996.html`

Motivations et cibles (I)

- Des attaques aux motivations très variées
  - Ludique : amusement, curiosité, défi, réputation
    - hackers lambda
  - Idéologie (voire terrorisme) : vandalisme, déni de service
    - 2007, 1er exemple de cyberterrorisme/cyberguerre en Estonie, déplacement de statue de soldat soviétique à Tallinn (Estonie) → vague de cyberattaques sans précédent orchestrées par la Russie via des botnets, sites gouvernementaux et banques ciblées, attaques de type déni de service
    - 2015, cyberattaque contre TV5Monde, site web défiguré par des messages pour l'état islamique
  - Cupidité : vol de données bancaires, extorsion (*ransomware*)
    - Depuis 2012, explosion des rançongiciels en parallèle des échanges de cryptomonnaies type bitcoin Petya WannaCry, sur kaspersky.fr
    - Une vidéo sur les rançongiciels : `https://www.lemonde.fr/pixels/video/2021/02/17/comment-fonctionnent-les-cyberattaques-aux-rancongiciels-qui-ont-cible-des-hopitaux_6070246_4408996.html`
    - **Vous avez une sauvegarde ?**

Motivations et cibles (I)

- Des attaques aux motivations très variées
  - Ludique : amusement, curiosité, défi, réputation
    - hackers lambda
  - Idéologie (voire terrorisme) : vandalisme, déni de service
    - 2007, 1er exemple de cyberterrorisme/cyberguerre en Estonie, déplacement de statue de soldat soviétique à Tallinn (Estonie) → vague de cyberattaques sans précédent orchestrées par la Russie via des botnets, sites gouvernementaux et banques ciblées, attaques de type déni de service
    - 2015, cyberattaque contre TV5Monde, site web défiguré par des messages pour l'état islamique
  - Cupidité : vol de données bancaires, extorsion (*ransomware*)
    - Depuis 2012, explosion des rançongiciels en parallèle des échanges de cryptomonnaies type bitcoin Petya WannaCry, sur kaspersky.fr
    - Une vidéo sur les rançongiciels : `https://www.lemonde.fr/pixels/video/2021/02/17/comment-fonctionnent-les-cyberattaques-aux-rancongiciels-qui-ont-cible-des-hopitaux_6070246_4408996.html`
    - **Vous avez une sauvegarde ?**
  - Espionnage : industriel (concurrence) ou étatique (surveillance)

Motivations et cibles (I)

- Des attaques aux motivations très variées
  - Ludique : amusement, curiosité, défi, réputation
    - hackers lambda
  - Idéologie (voire terrorisme) : vandalisme, déni de service
    - 2007, 1er exemple de cyberterrorisme/cyberguerre en Estonie, déplacement de statue de soldat soviétique à Tallinn (Estonie) → vague de cyberattaques sans précédent orchestrées par la Russie via des botnets, sites gouvernementaux et banques ciblées, attaques de type déni de service
    - 2015, cyberattaque contre TV5Monde, site web défiguré par des messages pour l'état islamique
  - Cupidité : vol de données bancaires, extorsion (*ransomware*)
    - Depuis 2012, explosion des rançongiciels en parallèle des échanges de cryptomonnaies type bitcoin Petya WannaCry, sur kaspersky.fr
    - Une vidéo sur les rançongiciels : `https://www.lemonde.fr/pixels/video/2021/02/17/comment-fonctionnent-les-cyberattaques-aux-rancongiciels-qui-ont-cible-des-hopitaux_6070246_4408996.html`
    - **Vous avez une sauvegarde ?**
  - Espionnage : industriel (concurrence) ou étatique (surveillance)
    - 2010 Stuxnet, ver ciblant les infrastructures nucléaires iraniennes (via clés USB infectées ?) centrifugeuses Siemens endomagées, développement USA-Israël soupçonné.

Motivations et cibles (II)

Motivations et cibles (II)

- Tout système d'information est une cible potentielle
  (Source : Pierre Lasbordes, *op. cit.*)

Motivations et cibles (II)

- Tout système d'information est une cible potentielle

  (Source : Pierre Lasbordes, *op. cit.*)

  - Infrastructures « vitales » : réseaux électrique, de communications, de transports, centrales nucléaires, hôpitaux

Motivations et cibles (II)

- Tout système d'information est une cible potentielle

  (Source : Pierre Lasbordes, *op. cit.*)

  - Infrastructures « vitales » : réseaux électrique, de communications, de transports, centrales nucléaires, hôpitaux
    - Loi de Programmation Militaire 2014–2019, décrets 2016– décrivant les obligations de sécurité informatique par secteur, page OIV ANSSI
      Tous les "flux" d'une ville : alimentation, eau, énergie...

Motivations et cibles (II)

- Tout système d'information est une cible potentielle

  (Source : Pierre Lasbordes, *op. cit.*)

  - Infrastructures « vitales » : réseaux électrique, de communications, de transports, centrales nucléaires, hôpitaux
    - Loi de Programmation Militaire 2014–2019, décrets 2016– décrivant les obligations de sécurité informatique par secteur, page OIV ANSSI
      Tous les "flux" d'une ville : alimentation, eau, énergie...
  - États : sites gouvernementaux et militaires

Motivations et cibles (II)

- Tout système d'information est une cible potentielle

  (Source : Pierre Lasbordes, *op. cit.*)

  - Infrastructures « vitales » : réseaux électrique, de communications, de transports, centrales nucléaires, hôpitaux
    - Loi de Programmation Militaire 2014–2019, décrets 2016– décrivant les obligations de sécurité informatique par secteur, page OIV ANSSI
      Tous les "flux" d'une ville : alimentation, eau, énergie...
  - États : sites gouvernementaux et militaires
    - 2004, Paris, toit de l'ambassade US bâché avec fenêtres en trompe-l'œil, station d'écoutes (antennes en tout genre)
    - 2013, affaire d'espionnage du ministère des affaires étrangères par les USA
    - Crypto AG, sur Swiss Info, entreprise suisse de Boris Hagelin de systèmes de chiffrement pas si neutre, espionnage de pays tiers par la NSA et le GCHQ de 1956 à 1991

Motivations et cibles (II)

- Tout système d'information est une cible potentielle

  (Source : Pierre Lasbordes, *op. cit.*)

  - Infrastructures « vitales » : réseaux électrique, de communications, de transports, centrales nucléaires, hôpitaux
    - Loi de Programmation Militaire 2014–2019, décrets 2016– décrivant les obligations de sécurité informatique par secteur, page OIV ANSSI
      Tous les "flux" d'une ville : alimentation, eau, énergie...
  - États : sites gouvernementaux et militaires
    - 2004, Paris, toit de l'ambassade US bâché avec fenêtres en trompe-l'œil, station d'écoutes (antennes en tout genre)
    - 2013, affaire d'espionnage du ministère des affaires étrangères par les USA
    - Crypto AG, sur Swiss Info, entreprise suisse de Boris Hagelin de systèmes de chiffrement pas si neutre, espionnage de pays tiers par la NSA et le GCHQ de 1956 à 1991
  - Entreprises : cyber-espionnage, vengeance

Motivations et cibles (II)

- Tout système d'information est une cible potentielle

  (Source : Pierre Lasbordes, *op. cit.*)

  - Infrastructures « vitales » : réseaux électrique, de communications, de transports, centrales nucléaires, hôpitaux
    - Loi de Programmation Militaire 2014–2019, décrets 2016– décrivant les obligations de sécurité informatique par secteur, page OIV ANSSI
      Tous les "flux" d'une ville : alimentation, eau, énergie...
  - États : sites gouvernementaux et militaires
    - 2004, Paris, toit de l'ambassade US bâché avec fenêtres en trompe-l'œil, station d'écoutes (antennes en tout genre)
    - 2013, affaire d'espionnage du ministère des affaires étrangères par les USA
    - Crypto AG, sur Swiss Info, entreprise suisse de Boris Hagelin de systèmes de chiffrement pas si neutre, espionnage de pays tiers par la NSA et le GCHQ de 1956 à 1991
  - Entreprises : cyber-espionnage, vengeance
    - On ne prépare pas ses slides de réunion confidentielle dans le TGV !

Motivations et cibles (II)

- Tout système d'information est une cible potentielle

  (Source : Pierre Lasbordes, *op. cit.*)

  - Infrastructures « vitales » : réseaux électrique, de communications, de transports, centrales nucléaires, hôpitaux
    - Loi de Programmation Militaire 2014–2019, décrets 2016– décrivant les obligations de sécurité informatique par secteur, page OIV ANSSI
      Tous les "flux" d'une ville : alimentation, eau, énergie...
  - États : sites gouvernementaux et militaires
    - 2004, Paris, toit de l'ambassade US bâché avec fenêtres en trompe-l'œil, station d'écoutes (antennes en tout genre)
    - 2013, affaire d'espionnage du ministère des affaires étrangères par les USA
    - Crypto AG, sur Swiss Info, entreprise suisse de Boris Hagelin de systèmes de chiffrement pas si neutre, espionnage de pays tiers par la NSA et le GCHQ de 1956 à 1991
  - Entreprises : cyber-espionnage, vengeance
    - On ne prépare pas ses slides de réunion confidentielle dans le TGV !
  - Entités académiques : universités, laboratoires de recherche

Motivations et cibles (II)

- Tout système d'information est une cible potentielle

  (Source : Pierre Lasbordes, *op. cit.*)

  - Infrastructures « vitales » : réseaux électrique, de communications, de transports, centrales nucléaires, hôpitaux
    - Loi de Programmation Militaire 2014–2019, décrets 2016– décrivant les obligations de sécurité informatique par secteur, page OIV ANSSI
      Tous les "flux" d'une ville : alimentation, eau, énergie...
  - États : sites gouvernementaux et militaires
    - 2004, Paris, toit de l'ambassade US bâché avec fenêtres en trompe-l'œil, station d'écoutes (antennes en tout genre)
    - 2013, affaire d'espionnage du ministère des affaires étrangères par les USA
    - Crypto AG, sur Swiss Info, entreprise suisse de Boris Hagelin de systèmes de chiffrement pas si neutre, espionnage de pays tiers par la NSA et le GCHQ de 1956 à 1991
  - Entreprises : cyber-espionnage, vengeance
    - On ne prépare pas ses slides de réunion confidentielle dans le TGV !
  - Entités académiques : universités, laboratoires de recherche
    - 2015, attaque de l'Université de Calgary la semaine de la conférence Canadian Number Theory Association, plus aucun mail, données perdues

Motivations et cibles (III)

Motivations et cibles (III)

- Tout système d'information est une cible potentielle
  (Source : Pierre Lasbordes, *op. cit.*)

Motivations et cibles (III)

- Tout système d'information est une cible potentielle

  (Source : Pierre Lasbordes, *op. cit.*)

  - Individus : cibles vulnérables, peu sensibilisées, ne maîtrisent pas toutes les données qu'elles produisent ; leurs machines peuvent aussi servir de relais (*botnet*)

Motivations et cibles (III)

- Tout système d'information est une cible potentielle

  (Source : Pierre Lasbordes, *op. cit.*)

  - Individus : cibles vulnérables, peu sensibilisées, ne maîtrisent pas toutes les données qu'elles produisent ; leurs machines peuvent aussi servir de relais (*botnet*)
    - botnet : réseau d'ordinateurs zombies, pour les envois de mails infectés, les attaques par déni de service

Motivations et cibles (III)

- Tout système d'information est une cible potentielle

  (Source : Pierre Lasbordes, *op. cit.*)

  - Individus : cibles vulnérables, peu sensibilisées, ne maîtrisent pas toutes les données qu'elles produisent ; leurs machines peuvent aussi servir de relais (*botnet*)
    - botnet : réseau d'ordinateurs zombies, pour les envois de mails infectés, les attaques par déni de service
    - Attaque Mirai (*avenir* en japonais), détails sur le blog Pixels, cible les objets connectés, PlayStation Sony, caméras de vidéo-surveillance...

Motivations et cibles (III)

- Tout système d'information est une cible potentielle

  (Source : Pierre Lasbordes, *op. cit.*)

  - Individus : cibles vulnérables, peu sensibilisées, ne maîtrisent pas toutes les données qu'elles produisent ; leurs machines peuvent aussi servir de relais (*botnet*)
    - botnet : réseau d'ordinateurs zombies, pour les envois de mails infectés, les attaques par déni de service
    - Attaque Mirai (*avenir* en japonais), détails sur le blog Pixels, cible les objets connectés, PlayStation Sony, caméras de vidéo-surveillance...
    - mais aussi les pacemakers ! (Dick Cheney a fait modifier le sien pour éviter le scénario de la série Homeland)

Motivations et cibles (III)

- Tout système d'information est une cible potentielle

  (Source : Pierre Lasbordes, *op. cit.*)

  - Individus : cibles vulnérables, peu sensibilisées, ne maîtrisent pas toutes les données qu'elles produisent ; leurs machines peuvent aussi servir de relais (*botnet*)
    - botnet : réseau d'ordinateurs zombies, pour les envois de mails infectés, les attaques par déni de service
    - Attaque Mirai (*avenir* en japonais), détails sur le blog Pixels, cible les objets connectés, PlayStation Sony, caméras de vidéo-surveillance...
    - mais aussi les pacemakers ! (Dick Cheney a fait modifier le sien pour éviter le scénario de la série Homeland)

La série Mr. Robot donne plein d'exemples réels

Modèle simplifié de système d'information

- Superposition de plusieurs couches

Modèle simplifié de système d'information

- Superposition de plusieurs couches

  - Matériel (machines, routeurs, câbles, etc.)

Modèle simplifié de système d'information

- Superposition de plusieurs couches

  - Système d'exploitation
  - Matériel (machines, routeurs, câbles, etc.)

Modèle simplifié de système d'information

- Superposition de plusieurs couches

  - Programmes et bibliothèques logicielles
  - Système d'exploitation
  - Matériel (machines, routeurs, câbles, etc.)

Modèle simplifié de système d'information

- Superposition de plusieurs couches

  - Protocoles (IP, TCP, HTTP, etc.)

  - Programmes et bibliothèques logicielles
  - Système d'exploitation
  - Matériel (machines, routeurs, câbles, etc.)

Modèle simplifié de système d'information

- Superposition de plusieurs couches
  - Utilisateurs
  - Protocoles (IP, TCP, HTTP, etc.)

  - Programmes et bibliothèques logicielles
  - Système d'exploitation
  - Matériel (machines, routeurs, câbles, etc.)

Modèle simplifié de système d'information

- Superposition de plusieurs couches
  - Utilisateurs
  - Protocoles (IP, TCP, HTTP, etc.)
  - Primitives cryptographiques (AES, RSA, etc.)
  - Programmes et bibliothèques logicielles
  - Système d'exploitation
  - Matériel (machines, routeurs, câbles, etc.)

Modèle simplifié de système d'information

- Superposition de plusieurs couches
  - Utilisateurs
  - Protocoles (IP, TCP, HTTP, SSL/TLS, etc.)
  - Primitives cryptographiques (AES, RSA, etc.)
  - Programmes et bibliothèques logicielles
  - Système d'exploitation
  - Matériel (machines, routeurs, câbles, etc.)

Modèle simplifié de système d'information

- Superposition de plusieurs couches
  - Utilisateurs
  - Protocoles (IP, TCP, HTTP, SSL/TLS, etc.)
  - Primitives cryptographiques (AES, RSA, etc.)
  - Programmes et bibliothèques logicielles
  - Système d'exploitation
  - Matériel (machines, routeurs, câbles, etc.)
- Chaque couche présente des vulnérabilités et donc autant de risques d'attaque

Modèle simplifié de système d'information

- Superposition de plusieurs couches
  - Utilisateurs
  - Protocoles (IP, TCP, HTTP, SSL/TLS, etc.)
  - Primitives cryptographiques (AES, RSA, etc.)
  - Programmes et bibliothèques logicielles
  - Système d'exploitation
  - Matériel (machines, routeurs, câbles, etc.)
- Chaque couche présente des vulnérabilités et donc autant de risques d'attaque
- Attaque sophistiquée : plusieurs vecteurs d'attaque exploités conjointement

Modèle simplifié de système d'information

- Superposition de plusieurs couches
  - Utilisateurs
  - Protocoles (IP, TCP, HTTP, SSL/TLS, etc.)
  - Primitives cryptographiques (AES, RSA, etc.)
  - Programmes et bibliothèques logicielles
  - Système d'exploitation
  - Matériel (machines, routeurs, câbles, etc.)
- Chaque couche présente des vulnérabilités et donc autant de risques d'attaque
- Attaque sophistiquée : plusieurs vecteurs d'attaque exploités conjointement
- Besoin d'un plan de sécurité globale :
  analogie de la porte blindée et des fenêtres ouvertes

Risques d'attaque et contre-mesures

Risques d'attaque et contre-mesures

- Risques humains : maladresse, inconscience, malveillance, ingénierie sociale, espionnage, sabotage, etc.

Risques d'attaque et contre-mesures

- Risques humains : maladresse, inconscience, malveillance, ingénierie sociale, espionnage, sabotage, etc.
- Contre-mesures :
  - Sensibilisation, formation, habilitation des utilisateurs
  - Politique de contrôle d'accès (physique et logique)
  - Cloisonnement de l'information (droits d'accès)
  - Traçabilité (authentification et non-répudiation)

Risques d'attaque et contre-mesures

- Attaques par messagerie :
  - *Spam* (pourriel)
  - *Phishing* (hameçonnage)
  - *Hoax* (canular)
  - Envoi de programmes malveillants

Risques d'attaque et contre-mesures

- Attaques par messagerie :
  - *Spam* (pourriel)
  - *Phishing* (hameçonnage)
  - *Hoax* (canular)
  - Envoi de programmes malveillants
- Contre-mesures :
  - Filtres anti-spam
  - Anti-virus
  - Sensibilisation des utilisateurs
  - Authentification de l'expéditeur

Risques d'attaque et contre-mesures

- Attaques d'applications Web :
  - Vulnérabilités PHP
  - Accès aux bases de données (injection SQL)
  - XSS (*Cross-Site Scripting*)
  - CSRF (*Cross-Site Request Forgery*)

Risques d'attaque et contre-mesures

- Attaques d'applications Web :
  - Vulnérabilités PHP
  - Accès aux bases de données (injection SQL)
  - XSS (*Cross-Site Scripting*)
  - CSRF (*Cross-Site Request Forgery*)
- Contre-mesures :
  - Programmation défensive
  - Isoler les applications Web du reste du système d'information (DMZ)
  - Ne pas stocker les données sensibles en clair (mots de passe, cartes bancaires, données personnelles)
  - Ne jamais faire confiance aux données provenant de l'utilisateur
  - Protection des cookies (HttpOnly) et jetons CSRF

Risques d'attaque et contre-mesures

- Attaques d'applications Web :
  - Vulnérabilités PHP
  - Accès aux bases de données (injection SQL)
  - XSS (*Cross-Site Scripting*)
  - CSRF (*Cross-Site Request Forgery*)
- Contre-mesures :
  - Programmation défensive
  - Isoler les applications Web du reste du système d'information (DMZ)
  - Ne pas stocker les données sensibles en clair (mots de passe, cartes bancaires, données personnelles)
  - Ne jamais faire confiance aux données provenant de l'utilisateur
  - Protection des cookies (HttpOnly) et jetons CSRF
- Recommandations :
  - OWASP : *Open Web Application Security Project*

Risques d'attaque et contre-mesures

- Attaques réseau :
  - Interception de paquets (*sniffing*)
  - Usurpation d'identité (*spoofing*)
  - Attaque *malicious intruder in the middle* (« personne au milieu »)
  - Cartographie réseau (*port scan*)
  - Déni de service par saturation (*DDoS*)

Risques d'attaque et contre-mesures

- Attaques réseau :
  - Interception de paquets (*sniffing*)
  - Usurpation d'identité (*spoofing*)
  - Attaque *malicious intruder in the middle* (« personne au milieu »)
  - Cartographie réseau (*port scan*)
  - Déni de service par saturation (*DDoS*)
- Contre-mesures :
  - Cryptographie (chiffrement et authentification)
  - Pare-feu (*firewall*)
  - Infrastructure redondante

Risques d'attaque et contre-mesures

- Attaques cryptographiques :
  - Cryptanalyse (protocoles et primitives)
  - Découverte et exploitation de failles

Risques d'attaque et contre-mesures

- Attaques cryptographiques :
  - Cryptanalyse (protocoles et primitives)
  - Découverte et exploitation de failles
- Contre-mesures :
  - Utilisation de cryptographie prouvée (ou tout du moins bien connue, étudiée, et présumée sûre)
  - Dimensionnement des clés
  - Veille de sécurité sur la découverte de failles (dans les primitives, les protocoles ou les implémentations)
  - Mises à jour de sécurité

Risques d'attaque et contre-mesures

- Attaques cryptographiques :
  - Cryptanalyse (protocoles et primitives)
  - Découverte et exploitation de failles
- Contre-mesures :
  - Utilisation de cryptographie prouvée (ou tout du moins bien connue, étudiée, et présumée sûre)
  - Dimensionnement des clés
  - Veille de sécurité sur la découverte de failles (dans les primitives, les protocoles ou les implémentations)
  - Mises à jour de sécurité
- Recommandations :
  - Organismes gouvernementaux : ANSSI (Fr.), NIST (É.U.), BSI (All.)
  - Organismes internationaux de standardisation : ISO/IEC, IETF, etc.
  - Travaux académiques

Risques d'attaque et contre-mesures

- Attaques cryptographiques :
  - Cryptanalyse (protocoles et primitives)
  - Découverte et exploitation de failles
- Contre-mesures :
  - Utilisation de cryptographie prouvée (ou tout du moins bien connue, étudiée, et présumée sûre)
  - Dimensionnement des clés
  - Veille de sécurité sur la découverte de failles (dans les primitives, les protocoles ou les implémentations)
  - Mises à jour de sécurité
- Recommandations :
  - Organismes gouvernementaux : ANSSI (Fr.), NIST (É.U.), BSI (All.)
  - Organismes internationaux de standardisation : ISO/IEC, IETF, etc.
  - Travaux académiques
  - À qui faire confiance ? (ex : Dual_EC_DRGB standardisé par le NIST)

Risques d'attaque et contre-mesures

- Attaques des mots de passe :
  - Force brute
  - Dictionnaires : RockYou (2009, 32M), Cupid Media (42M), Adobe (152M) BREACH (2017, 1.4G), COMB (3.2G) RockYou2021 (8.4G)
  - Tables arc-en-ciel (*rainbow tables*)

- TD1

Risques d'attaque et contre-mesures

- Attaques des mots de passe :
  - Force brute
  - Dictionnaires : RockYou (2009, 32M), Cupid Media (42M), Adobe (152M) BREACH (2017, 1.4G), COMB (3.2G) RockYou2021 (8.4G)
  - Tables arc-en-ciel (*rainbow tables*)
- Contre-mesures :
  - Sensibilisation des utilisateurs
  - Choix et renouvellement régulier des mots de passe
  - Stockage des mots de passe : hachage cryptographique + sel
  - Authentification cryptographique par certificat
- TD1

Risques d'attaque et contre-mesures

- Programmes malveillants :
  - Virus, ver
    - Morris, 1988, 1er ver informatique, création du 1er CERT
  - Cheval de Troie (*trojan*)
  - Porte dérobée (*backdoor*)
  - Enregistreur de frappe (*keylogger*)
  - C2 (*Command & Control*)

Risques d'attaque et contre-mesures

- Programmes malveillants :
  - Virus, ver
    - Morris, 1988, 1er ver informatique, création du 1er CERT
  - Cheval de Troie (*trojan*)
  - Porte dérobée (*backdoor*)
  - Enregistreur de frappe (*keylogger*)
  - C2 (*Command & Control*)
- Contre-mesures :
  - Anti-virus
  - IDS (*Intrusion Detection System*)
  - Authentification des programmes exécutés

Risques d'attaque et contre-mesures

- Programmes malveillants :
  - Virus, ver
    - Morris, 1988, 1er ver informatique, création du 1er CERT
  - Cheval de Troie (*trojan*)
  - Porte dérobée (*backdoor*)
  - Enregistreur de frappe (*keylogger*)
  - C2 (*Command & Control*)
- Contre-mesures :
  - Anti-virus
  - IDS (*Intrusion Detection System*)
  - Authentification des programmes exécutés
- Mises en œuvre :
  - appel à expertise
  - par Bug-Bounty

Risques d'attaque et contre-mesures

- Programmes malveillants :
  - Virus, ver
    - Morris, 1988, 1er ver informatique, création du 1er CERT
  - Cheval de Troie (*trojan*)
  - Porte dérobée (*backdoor*)
  - Enregistreur de frappe (*keylogger*)
  - C2 (*Command & Control*)
- Contre-mesures :
  - Anti-virus
  - IDS (*Intrusion Detection System*)
  - Authentification des programmes exécutés
- Mises en œuvre :
  - appel à expertise
  - par Bug-Bounty
    - Un bug trouvé à Nancy au Loria dans le système de vote de Moscou
    - 40kE BugBounty SwissPost pour le Loria, système de vote électronique

Risques d'attaque et contre-mesures

- Vulnérabilités système et logicielles :
  - Dépassement de tampon (*buffer overflow*)
  - *Shellcode*
  - Élévation des privilèges
  - *Exploits* et *0-days*

Risques d'attaque et contre-mesures

- Vulnérabilités système et logicielles :
  - Dépassement de tampon (*buffer overflow*)
  - *Shellcode*
  - Élévation des privilèges
  - *Exploits* et *0-days*
- Contre-mesures :
  - Isolation (*sandboxing*) des programmes potentiellement vulnérables
  - Contre-mesures système : restrictions d'accès à la mémoire (bit NX *No eXecute*), protection de la pile, randomisation des adresses (Address Space Layout Randomization, ASLR)
  - Veille de sécurité sur la découverte de failles
  - Mises à jour de sécurité
  - Développement logiciel : attention à la gestion mémoire

Risques d'attaque et contre-mesures

- Vulnérabilités système et logicielles :
  - Dépassement de tampon (*buffer overflow*)
  - *Shellcode*
  - Élévation des privilèges
  - *Exploits* et *0-days*
- Contre-mesures :
  - Isolation (*sandboxing*) des programmes potentiellement vulnérables
  - Contre-mesures système : restrictions d'accès à la mémoire (bit NX *No eXecute*), protection de la pile, randomisation des adresses (Address Space Layout Randomization, ASLR)
  - Veille de sécurité sur la découverte de failles
  - Mises à jour de sécurité
  - Développement logiciel : attention à la gestion mémoire
- Alertes de sécurité :
  - CERT-FR (ANSSI) : `https://www.cert.ssi.gouv.fr/`
  - CVE (MITRE Corp., É.U.) : `https://www.cve.org/`

Risques d'attaque et contre-mesures

- Attaques par canaux cachés :
  - Émissions électromagnétiques ou sons (TEMPEST)
  - Consommation électrique
  - Temps de réponse
  - Brouillage

Risques d'attaque et contre-mesures

- Attaques par canaux cachés :
  - Émissions électromagnétiques ou sons (TEMPEST)
  - Consommation électrique
  - Temps de réponse
  - Brouillage
- Contre-mesures :
  - Isolation physique (cage de Faraday)
  - Parasites, bruitage des mesures
  - Algorithmes et circuits spécifiques pour minimiser les fuites d'information

Risques d'attaque et contre-mesures

- Attaques par canaux cachés :
  - Émissions électromagnétiques ou sons (TEMPEST)
  - Consommation électrique
  - Temps de réponse
  - Brouillage
- Contre-mesures :
  - Isolation physique (cage de Faraday)
  - Parasites, bruitage des mesures
  - Algorithmes et circuits spécifiques pour minimiser les fuites d'information
- TD 5

Risques d'attaque et contre-mesures

- Attaques physiques :
  - Vol, perte
  - Destruction

Risques d'attaque et contre-mesures

- Attaques physiques :
  - Vol, perte
  - Destruction
- Contre-mesures :
  - Politique stricte du stockage de données sensibles
  - Chiffrement
  - Contrôle d'intégrité
  - Mécanisme de sauvegarde (*backup*)

Top 5 des vulnérabilités

- D'après Cryptome (https://cryptome.org/) :

Top 5 des vulnérabilités

- D'après Cryptome (https://cryptome.org/) :
  - *No. 1 vulnerability of crypto is the user,*

Top 5 des vulnérabilités

- D'après Cryptome (`https://cryptome.org/`) :
  - *No. 1 vulnerability of crypto is the user,*
  - *2nd passphrases,*

Top 5 des vulnérabilités

- D'après Cryptome (`https://cryptome.org/`) :
  - *No. 1 vulnerability of crypto is the user,*
  - *2nd passphrases,*
  - *3rd overconfidence,*

Top 5 des vulnérabilités

- D'après Cryptome (`https://cryptome.org/`) :
  - *No. 1 vulnerability of crypto is the user,*
  - *2nd passphrases,*
  - *3rd overconfidence,*
  - *4th trust in the producer,*

Top 5 des vulnérabilités

- D'après Cryptome (https://cryptome.org/) :
  - *No. 1 vulnerability of crypto is the user,*
  - *2nd passphrases,*
  - *3rd overconfidence,*
  - *4th trust in the producer,*
  - *5th believing backdoors are No. 1.*

Établir une politique de SSI

Établir une politique de SSI

1. Évaluation
   - Identifier les actifs à protéger, les vulnérabilités et les menaces
   - Analyse des risques (méthodes EBIOS, MEHARI, ISO 27005, etc.)

Établir une politique de SSI

1. Évaluation
   - Identifier les actifs à protéger, les vulnérabilités et les menaces
   - Analyse des risques (méthodes EBIOS, MEHARI, ISO 27005, etc.)
2. Planification
   - Traiter chaque risque (accepter, éviter, transférer ou réduire)
   - Sélection des mesures à mettre en place

Établir une politique de SSI

1. Évaluation
   - Identifier les actifs à protéger, les vulnérabilités et les menaces
   - Analyse des risques (méthodes EBIOS, MEHARI, ISO 27005, etc.)
2. Planification
   - Traiter chaque risque (accepter, éviter, transférer ou réduire)
   - Sélection des mesures à mettre en place
3. Mise en œuvre
   - Mise en place des mesures de sécurité
   - Sensibilisation et formation des utilisateurs

Établir une politique de SSI

1. Évaluation
   - Identifier les actifs à protéger, les vulnérabilités et les menaces
   - Analyse des risques (méthodes EBIOS, MEHARI, ISO 27005, etc.)
2. Planification
   - Traiter chaque risque (accepter, éviter, transférer ou réduire)
   - Sélection des mesures à mettre en place
3. Mise en œuvre
   - Mise en place des mesures de sécurité
   - Sensibilisation et formation des utilisateurs
4. Mesures de contrôle : contrôle interne, audits internes, tests d'intrusion (*pentests*)

Établir une politique de SSI

1. Évaluation
   - Identifier les actifs à protéger, les vulnérabilités et les menaces
   - Analyse des risques (méthodes EBIOS, MEHARI, ISO 27005, etc.)
2. Planification
   - Traiter chaque risque (accepter, éviter, transférer ou réduire)
   - Sélection des mesures à mettre en place
3. Mise en œuvre
   - Mise en place des mesures de sécurité
   - Sensibilisation et formation des utilisateurs
4. Mesures de contrôle : contrôle interne, audits internes, tests d'intrusion (*pentests*)
5. Rétroaction : correction, amélioration des processus

Gestion des incidents de sécurité

Gestion des incidents de sécurité

- Rôles d'un CERT (*Computer Emergency Response Team*)
  ex. : CERT-FR (ANSSI, administration française), CERT-RENATER

Gestion des incidents de sécurité

- Rôles d'un CERT (*Computer Emergency Response Team*)
  ex. : CERT-FR (ANSSI, administration française), CERT-RENATER
- A priori : prévention
  - Veille de sécurité sur l'évolution des menaces (recommandations)
  - Informer lorsque des vulnérabilités sont découvertes
  - Alerter lorsque des attaques sont détectées

Gestion des incidents de sécurité

- Rôles d'un CERT (*Computer Emergency Response Team*)
  ex. : CERT-FR (ANSSI, administration française), CERT-RENATER
- A priori : prévention
  - Veille de sécurité sur l'évolution des menaces (recommandations)
  - Informer lorsque des vulnérabilités sont découvertes
  - Alerter lorsque des attaques sont détectées
- Pendant : intervention
  - Délai d'intervention le plus court possible
  - Diagnostiquer l'origine et l'ampleur de l'attaque
  - Prendre des mesures d'urgence pour circonscrire et déjouer l'attaque

Gestion des incidents de sécurité

- Rôles d'un CERT (*Computer Emergency Response Team*)
  ex. : CERT-FR (ANSSI, administration française), CERT-RENATER
- A priori : prévention
  - Veille de sécurité sur l'évolution des menaces (recommandations)
  - Informer lorsque des vulnérabilités sont découvertes
  - Alerter lorsque des attaques sont détectées
- Pendant : intervention
  - Délai d'intervention le plus court possible
  - Diagnostiquer l'origine et l'ampleur de l'attaque
  - Prendre des mesures d'urgence pour circonscrire et déjouer l'attaque
- A posteriori : analyse post-mortem
  - Analyser les traces pour retrouver les vecteurs d'attaque exploités
  - Déterminer les systèmes et les données compromises
  - S'assurer que l'attaque est achevée
  - Corriger les vulnérabilités exploitées

Gestion des incidents de sécurité

- Rôles d'un CERT (*Computer Emergency Response Team*)
  ex. : CERT-FR (ANSSI, administration française), CERT-RENATER
- A priori : prévention
  - Veille de sécurité sur l'évolution des menaces (recommandations)
  - Informer lorsque des vulnérabilités sont découvertes
  - Alerter lorsque des attaques sont détectées
- Pendant : intervention
  - Délai d'intervention le plus court possible
  - Diagnostiquer l'origine et l'ampleur de l'attaque
  - Prendre des mesures d'urgence pour circonscrire et déjouer l'attaque
- A posteriori : analyse post-mortem
  - Analyser les traces pour retrouver les vecteurs d'attaque exploités
  - Déterminer les systèmes et les données compromises
  - S'assurer que l'attaque est achevée
  - Corriger les vulnérabilités exploitées
- Un vrai exemple en vidéo (SSTIC 2017, TV5Monde)
  https://www.sstic.org/2017/presentation/2017_cloture/

Sources

- ANSSI, *Référentiel général de sécurité, version 2.0*, 2014.
- Pierre Lasbordes, *La sécurité des systèmes d'information : un enjeu majeur pour la France*, 2006.
- Verizon, *2023 Data Breach Investigations Report*, 2023.
- OWASP, *OWASP Top 10 for 2021*, 2021.
- Wikipédia, *Sécurité des systèmes d'information*, 2023.
- Wikipédia, *Risque informatique*, ou l'insécurité du système d'information, 2023.

Sources

- ANSSI, *Référentiel général de sécurité, version 2.0*, 2014.
- Pierre Lasbordes, *La sécurité des systèmes d'information : un enjeu majeur pour la France*, 2006.
- Verizon, *2023 Data Breach Investigations Report*, 2023.
- OWASP, *OWASP Top 10 for 2021*, 2021.
- Wikipédia, *Sécurité des systèmes d'information*, 2023.
- Wikipédia, *Risque informatique*, ou l'insécurité du système d'information, 2023.
- Pour s'entraîner, challenges de sécurité :
  - Root-Me : `https://www.root-me.org/`
  - Hack This Site : `https://www.hackthissite.org/`
  - Matasano crypto challenges : `https://cryptopals.com/`

Deuxième partie

Introduction à la cryptographie

(Slides de Marion Videau)

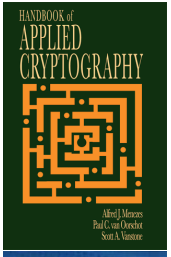
Bibliographie : aspects historiques

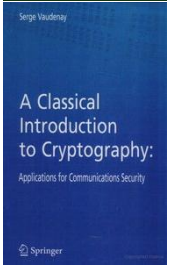
Simon Singh, *Histoire des Codes Secrets*.
Livre de Poche, 2001.

Jacques Stern, *La Science du Secret*.
Odile Jacob, 1998.

David Kahn, *The Codebreakers, revised edition*.
Schribner, 1996.

Bibliographie : ouvrages de référence

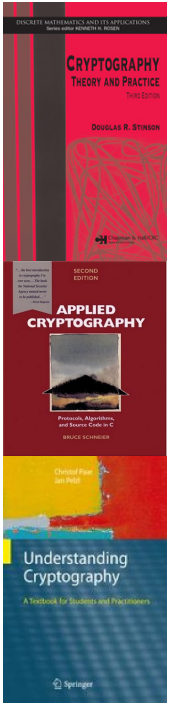




Alfred J. Menezes, Paul C. van Oorschot et
Scott A. Vanstone, *Handbook of Applied Cryptography*.
Chapman & Hall / CRC, 1996.
http://www.cacr.math.uwaterloo.ca/hac/

Serge Vaudenay, *A Classical Introduction to Cryptography*.
Springer, 2005.

Bibliographie : ouvrages de référence



Douglas R. Stinson,
Cryptography: Theory and Practice, 3rd edition.
Chapman & Hall / CRC, 2005.

Bruce Schneier, *Applied Cryptography, 2nd edition*.
Wiley, 1996.

Christof Paar et Jan Pelzl,
Understanding Cryptography, a Textbook for Students and Practitioners.
Springer, 2010.

Référentiel Général de Sécurité de l'ANSSI

(Agence Nationale de la Sécurité des Systèmes d'Information)

Note : le site de l'ANSSI migre de https://www.ssi.gouv.fr/ vers https://cyber.gouv.fr/ jusqu'en 2024.

https://cyber.gouv.fr/publications/mecanismes-cryptographiques

- Règles et recommandations concernant le choix et le dimensionnement des mécanismes cryptographiques – v2.04 du 01/01/2020 https://cyber.gouv.fr/sites/default/files/2021/03/anssi-guide-mecanismes_crypto-2.04.pdf
- Guide de sélection d'algorithmes cryptographiques – v1.0 du 08/03/2021 https://cyber.gouv.fr/sites/default/files/2021/03/anssi-guide-selection_crypto-1.0.pdf
- Règles et recommandations concernant la gestion des clés utilisées dans des mécanismes cryptographiques http://www.ssi.gouv.fr/uploads/2014/11/RGS_v-2-0_B2.pdf
- Règles et recommandations concernant les mécanismes d'authentification http://www.ssi.gouv.fr/uploads/2014/11/RGS_v-2-0_B3.pdf

Introduction à la cryptographie

- Contexte
- Repères historiques
- Généralités
- Exemple : améliorer un canal authentifié
- Cryptographie symétrique / asymétrique
- Sécurisation de canaux de communication

Introduction à la cryptographie

- Contexte
- Repères historiques
- Généralités
- Exemple : améliorer un canal authentifié
- Cryptographie symétrique / asymétrique
- Sécurisation de canaux de communication

Position du problème

Caractéristiques des systèmes d'information

- Information numérique
- Communications sur un canal public
- Machines reliées par réseau
- Multi-utilisateurs

Que veut-on protéger ?

Attention à ne pas confondre information et donnée !

Définition

Une donnée est la représentation d'une information sous une forme conventionnelle destinée à faciliter son traitement. [Glossaire ANSSI]

- On sait faire subir un traitement automatique à une donnée
- Une donnée peut être porteuse de plusieurs informations, même involontaires (dépend du contexte que peut connaître l'attaquant)
- Protéger une donnée ne suffit pas toujours à protéger l'information

Les besoins génériques de sécurité

- Confidentialité
  - Maintien au secret des informations vis-à-vis de tiers
- Intégrité
  - État des informations qui n'ont pas été modifiées
- Authenticité
  - Garantie de l'identité d'une entité ou de l'origine d'une communication
- Disponibilité
  - Garantit l'accessibilité de l'information

Moyens de protection

- Il existe plusieurs techniques :
  - Cryptographie
  - Sécurité matérielle
  - Sécurité système
  - Sécurité réseau
  - Tempest
- Chaque technique propose des solutions contre certaines menaces
- Pour se protéger efficacement, il faut (au moins) combiner les techniques

La science du secret

Définition (Cryptographie)

Étymologie : crypto-, du grec *κρυπτός*, caché, et
-graphie, du grec *γράφειν*, écrire.
Art d'écrire en langage codé, secret, chiffré.
[Dictionnaire de l'Académie française, 9ème édition]

De nombreuses applications dans la vie courante :

- SSL/TLS (`https`), SSH, GnuPG, *etc.*
- carte bleue, téléphone cellulaire, WiFi, Bluetooth
- *etc.*

En quoi consiste cette science ?

Cryptologie

Définition (Cryptologie)

Étude de la protection (algorithmique) de l'information sous forme numérique contre des accès ou manipulations non-autorisés.

$$\text{cryptologie} = \text{cryptographie} + \text{cryptanalyse}$$

- Cryptographie : conception des algorithmes cryptographiques
- Cryptanalyse : évaluation de la sécurité des algorithmes cryptographiques

Introduction à la cryptographie

- Contexte
- Repères historiques
- Généralités
- Exemple : améliorer un canal authentifié
- Cryptographie symétrique / asymétrique
- Sécurisation de canaux de communication

Cryptographie artisanale

Antiquité – 19e s. César (1er s. av. J.C.), Vigenère (1586), *etc.*
Transpositions et substitutions alphabétiques

Cryptographie mécanique

1883 *La Cryptographie Militaire* [Kerckhoffs]
Formalisation des systèmes de chiffrement

1926 *Cipher Printing Telegraph Systems for Secret Wire and Radio Telegraphic Communications* [Vernam]
Chiffrement de Vernam (masque jetable)

1939-44 Enigma et les « bombes » de Bletchley Park

1950-60 Machines Hagelin de Boris Hagelin, Crypto AG

Cryptographie industrielle

1949 *Communication Theory of Secrecy Systems* [Shannon]
Notion de sécurité inconditionnelle

1973-77 Standardisation de DES (*Data Encryption Standard*)

1976 *New Directions in Cryptography* [Diffie-Hellman]
Invention de la cryptographie à clé publique

1978 *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems* [Rivest-Shamir-Adleman]
Invention de RSA

1997-00 Standardisation d'AES (*Advanced Encryption Standard*)

2007-12 Standardisation de SHA-3 (*Secure Hash Algorithm*)

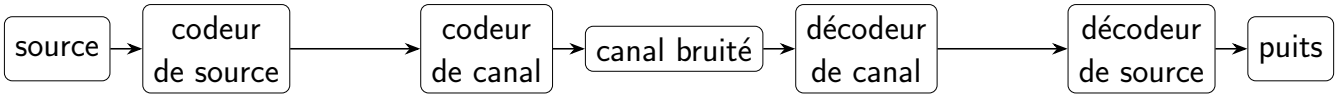
2016- Appel à standardisation d'algorithmes cryptographiques à clé publique post-quantiques

Introduction à la cryptographie

- Contexte
- Repères historiques
- Généralités
- Exemple : améliorer un canal authentifié
- Cryptographie symétrique / asymétrique
- Sécurisation de canaux de communication

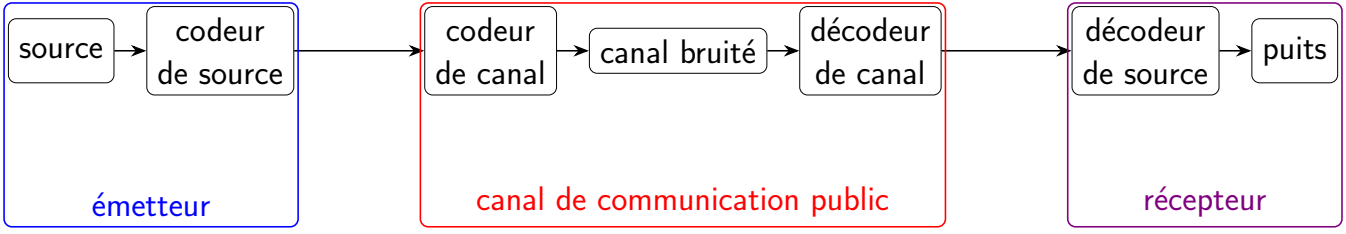
Modèle de communication

Modèle d'un système de communication général [Shannon, 1948]



Modèle de communication

Modèle d'un système de communication sans bruit



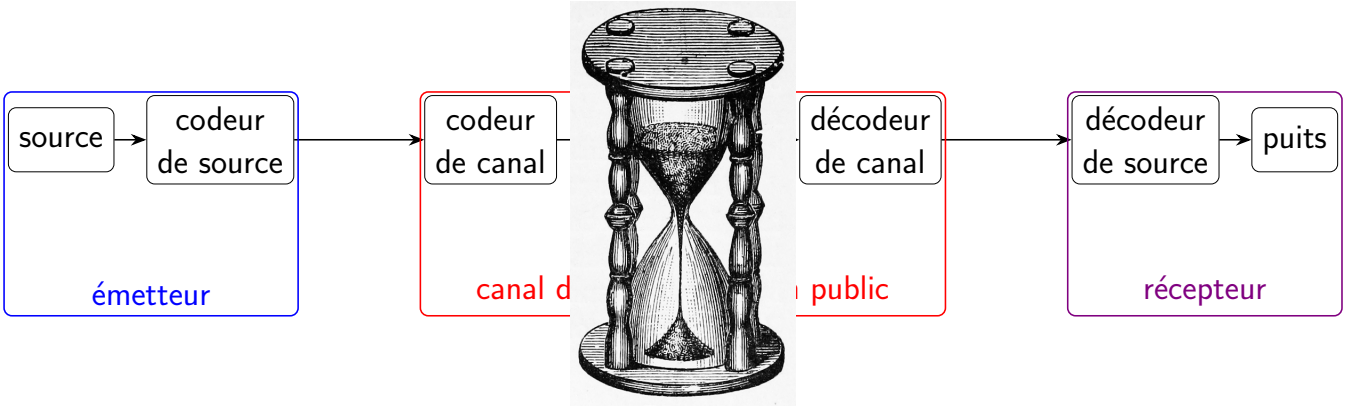
Modèle de communication

Modèle d'un système de communication sans bruit

https://commons.wikimedia.org/w/index.php?curid=805736

Modèle de communication

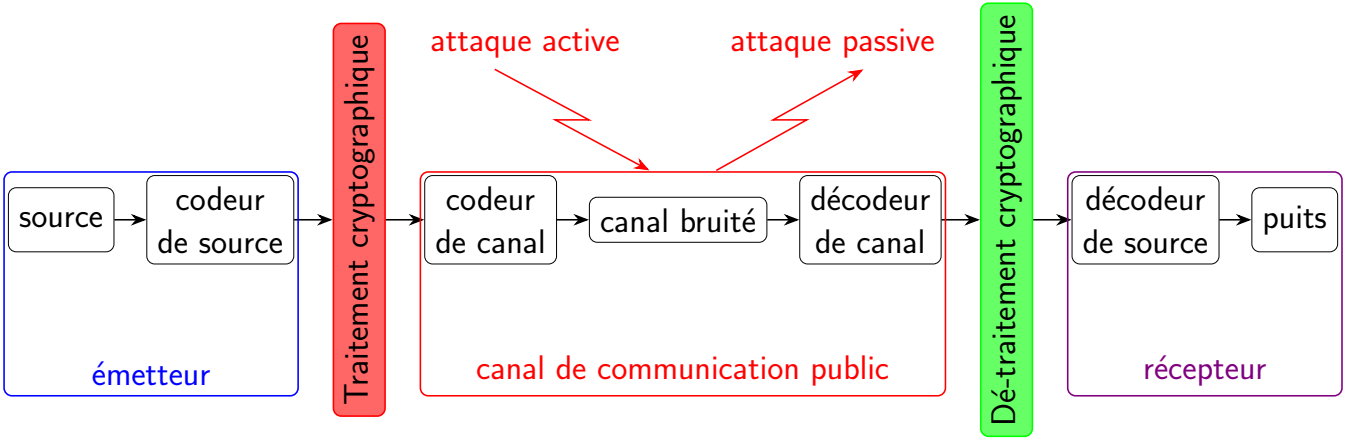
Modèle d'un système de communication sans bruit



https://commons.wikimedia.org/w/index.php?curid=1015977

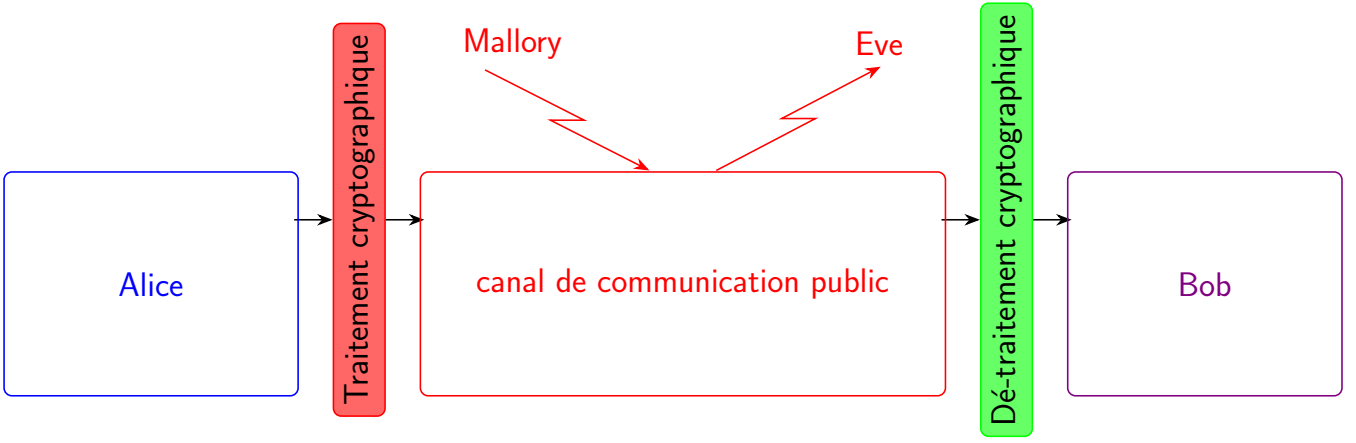
Modèle de communication

Modèle d'un système de communication cryptographique



Modèle de communication

Modèle d'un système de communication cryptographique



Différentes menaces

Une attaque peut être

- passive : espionnage
- active :
  - usurpation d'identité (de l'émetteur ou du récepteur)
  - altération des données = modification du contenu du message
  - répudiation du message = l'émetteur nie avoir envoyé le message
  - répétition du message = rejeu ultérieur

  - retardement de la transmission
  - destruction du message

Réponses de la cryptographie

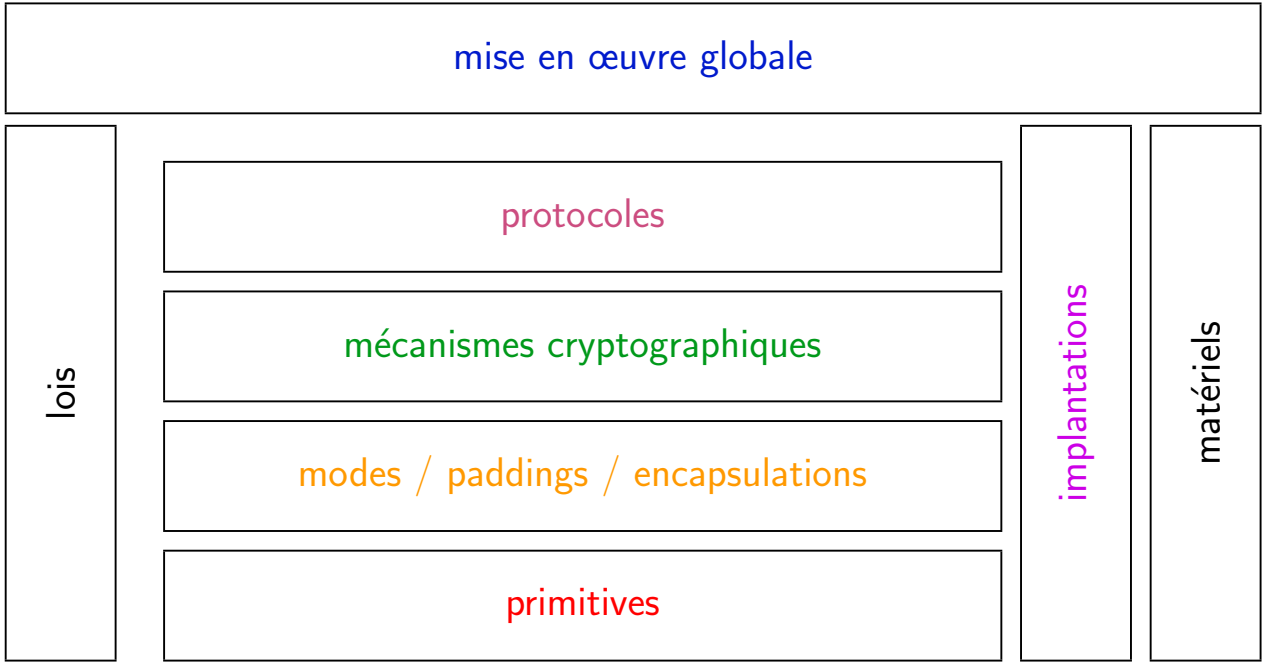
Trio fondamental :

- la confidentialité (Qui recevra le message ?) : l'information ne doit pas parvenir à des personnes qui n'ont pas à la connaître
- l'intégrité : l'information ne doit pas subir d'altérations frauduleuses
- l'authenticité (Qui a émis le message ?) : l'information doit provenir de son véritable auteur

Des fonctionnalités plus élaborées :

- la non répudiation,
- la divulgation nulle de connaissance,
- le vote électronique,
- le partage de secrets,
- la diffusion chiffrée (*broadcast encryption*),
- *etc.*

La cryptographie : un empilement de couches



Mécanismes cryptographiques fondamentaux

Algorithmes fournissant une fonctionnalité cryptographique fondamentale

- Contrôle d'intégrité cryptographique → fonction de hachage
- Génération de clés, d'IV (Init Vector), d'aléa → générateur d'aléa cryptographique
- Authentification de l'origine des messages → code d'authentification de message (MAC), algorithme de signature
- Confidentialité → chiffrement

Classification possible des mécanismes cryptographiques fondamentaux

Mécanismes	sans clé		à clé secrète		à clé publique	
Types	fonction de hachage	générateur d'aléa	MAC	chiffrement	signature	chiffrement
Exemples	SHA-384	`/dev/random`	AES-OMAC	AES-CTR	RSA-PSS	RSA-OAEP
Exemples de modes, de *paddings* ou d'encapsulations	chop-MD, éponge, MD-strengthening	Barak-Halevi	HMAC, OMAC, Wegman-Carter	flot, bloc, auto-synchronisant, synchrone, authentifiant, dédié, CBC, CFB, OFB, CTR, CCM, GCM, OCB, XEX	PSS	OAEP
Exemples de primitives	fonction de compression	fonction de hachage, chiffrement par flot, chiffrement par bloc, MAC	fonction de hachage, chiffrement par flot, chiffrement par bloc, fonction de hachage universel	fonction de rétroaction / filtrage / initialisation, permutation aléatoire paramétrée (chiffrement par bloc)	RSA, ElGamal, ECDSA EdDSA, NTRU, McEliece	RSA, ElGamal, NTRU, McEliece
Mécanismes complémentaires (éventuels)	générateur d'aléa (sel)	générateur physique d'aléa, source d'entropie	générateur d'aléa (clé)	générateur d'aléa (clé, IV)	générateur d'aléa (clé, sel), fonction de hachage	générateur d'aléa (clé, sel), fonction de hachage
Exemples de standards	FIPS 180-3 (SHS)	NIST SP 800-90	NIST SP 800-38B	FIPS 197 (AES), NIST SP 800-38A	FIPS 186-3 (DSS)	PKCS#1 v2.1

Introduction à la cryptographie

- Contexte
- Repères historiques
- Généralités
- Exemple : améliorer un canal authentifié
- Cryptographie symétrique / asymétrique
- Sécurisation de canaux de communication

Améliorer un canal authentifié

Contrôle d'intégrité avec fonction de hachage

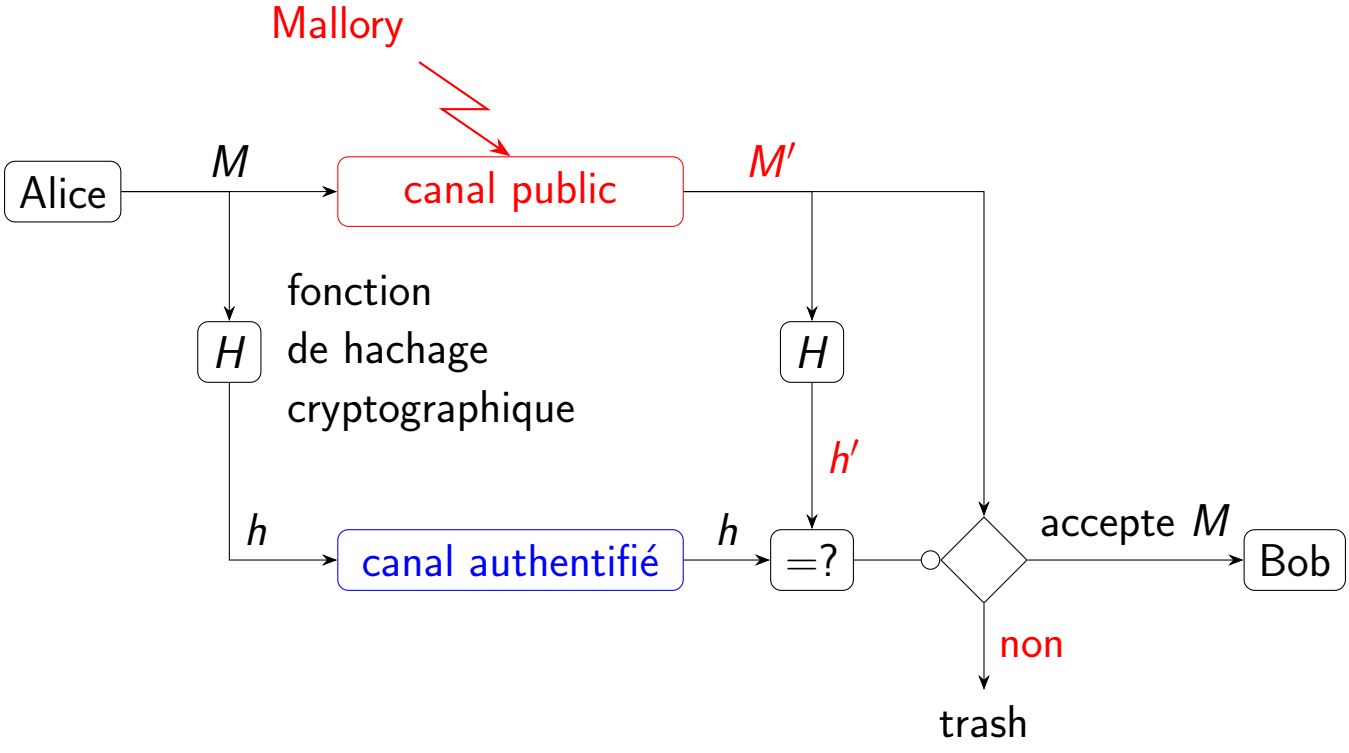
- Un canal public pour transmettre des messages de grande taille
- Un canal authentifié pour transmettre un contrôle d'intégrité de petite taille

Exemple : télécharger depuis un site mirroir, calculer le haché du fichier, vérifier qu'il correspond au haché donné sur le site officiel, canal "authentifié" via la confiance dans les DNS et leur sécurité

Définition (partielle)

Une fonction de hachage est un algorithme (efficace) qui calcule une valeur de taille fixe, appelée empreinte ou haché, à partir de messages de taille quelconque.

Améliorer un canal authentifié



- On utilise le canal authentifié après la création du message

Modèle d'attaques

- Pour que H soit qualifié de cryptographique, il faut que H résiste aux attaques par calcul de
  - premier antécédent : étant donné y,
    il est difficile de trouver x tel que $H(x) = y$
  - deuxième antécédent : étant donnés x et $H(x)$,
    il est difficile de trouver $x' \neq x$ tel que $H(x') = H(x)$
  - collision :
    il est difficile de trouver x et $x' \neq x$ tels que $H(x') = H(x)$
- La pertinence des modèles d'attaques dépend des applications.

Le contrôle d'intégrité est assuré sur le canal authentifié par la vérification de l'égalité des hachés.

Sécurité calculatoire

Les systèmes utilisés dans la pratique sont théoriquement cassables

Définition (Sécurité pratique, ou calculatoire)

Un attaquant possédant les spécifications de l'algorithme, autant de données que possible à sa disposition et une grande puissance de calcul ne peut pas casser le système cryptographique en un temps humainement raisonnable.

Complexité d'une attaque : ordres de grandeur

$$\text{Complexité} = O(\text{Temps} + \text{Mémoire} + \text{Données})$$

- On estime que 2^{128} opérations représente aujourd'hui un niveau raisonnable de sécurité

n	$2^n = 10^x$	**Exemples**
32	$2^{32} = 10^{9.6}$	nombre d'êtres humains sur Terre
46	$2^{46} = 10^{13.8}$	distance Terre - Soleil en millimètres nombre d'opérations effectuées par jour par un ordinateur à 1 GHz
55	$2^{55} = 10^{16.6}$	nombre d'opérations effectuées par an par un ordinateur à 1 GHz
82	$2^{82} = 10^{24.7}$	masse de la Terre en kilogrammes
90	$2^{90} = 10^{27.1}$	nombre d'opérations effectuées en 15 milliards d'années (âge de l'univers) par un ordinateur à 1 GHz
155	$2^{155} = 10^{46.7}$	nombre de molécules d'eau sur Terre
256	$2^{256} = 10^{77.1}$	nombre d'électrons dans l'univers

Sécurité des fonctions de hachage

Les collisions sont inévitables, on veut qu'elles soient difficiles à calculer

Exemples :

- MD5 (128 bits),
- SHA-1 (160 bits),
- SHA-2 (224, 256, 384 et 512 bits)
- SHA-3 (224, 256, 384 et 512 bits)

Définition (Attaque générique par paradoxe des anniversaires)

S'il y a 2^ℓ valeurs de hachés possibles, on trouve une collision avec probabilité supérieure à $1/2$ dès que l'on teste plus de $\sqrt{2^\ell \pi/2} = O(2^{\ell/2})$ entrées aléatoires.

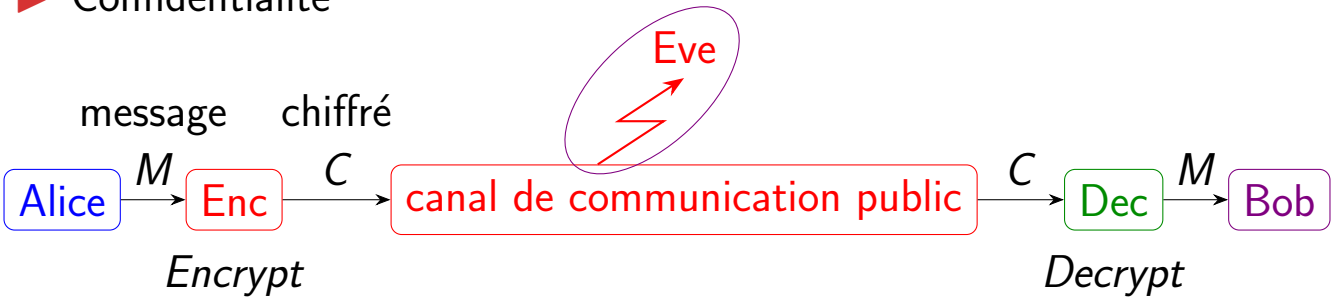
- Sécurité : 2^{64} pour MD5, 2^{80} pour SHA-1 $\rightarrow$ trop faible
- Compétition SHA-3 organisée par le NIST :
  - 24 janvier 2007 : appel à propositions pour les fonctions de hachage
  - 2 octobre 2012 : sélection de Keccak pour le nouveau standard
  - 5 août 2015 : publication du standard SHA-3 FIPS 202

Introduction à la cryptographie

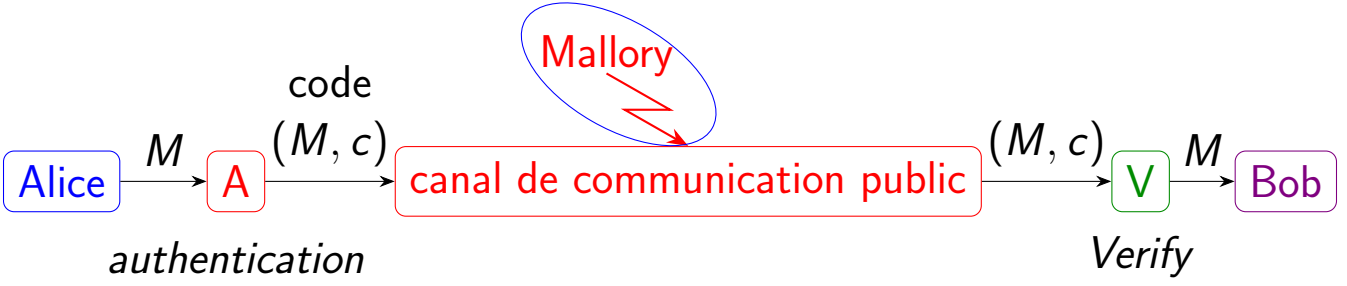
- Contexte
- Repères historiques
- Généralités
- Exemple : améliorer un canal authentifié
- Cryptographie symétrique / asymétrique
- Sécurisation de canaux de communication

Cryptographie symétrique / asymétrique

- Confidentialité



- Authentification, contrôle d'intégrité



Dec (resp. *A*) se déduit facilement de *Enc* (resp. *V*) $\Rightarrow$ symétrique

Dec (resp. *A*) **ne** se déduit **pas** facilement de *Enc* (resp. *V*) $\Rightarrow$ asymétrique

Les algorithmes cryptographiques

- Un algorithme :
  - est long à concevoir
  - doit être implanté sur du matériel
  - doit être transmis aux utilisateurs
  - doit être maintenu
- Les algorithmes doivent-ils être secrets ? Si non, comment assurer la sécurité ? Sur quel secret ? Secret pour qui ?
- Confidentialité $\Rightarrow$ déchiffrement possible **seulement** par le récepteur
- Authentification $\Rightarrow$ authentifiant calculable **seulement** par l'émetteur

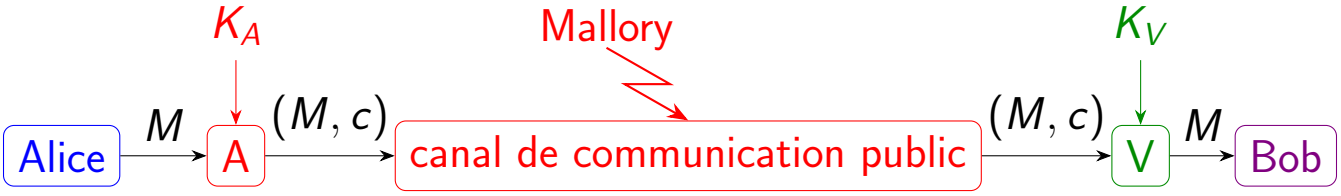
Les desiderata de Kerckhoffs (1883)

1. Le système doit être matériellement, sinon mathématiquement, indéchiffrable ;
2. Il faut qu'il **n'**exige **pas** le secret [...] ;
3. La clef doit pouvoir en être [...] retenue sans le secours de notes écrites, et être changée [...] ;
4. Il faut qu'il soit applicable à la correspondance télégraphique ;
5. Il faut qu'il soit portatif [...] ;
6. Enfin, il est nécessaire [...] que le système soit d'un usage facile, [...].

Cryptographie avec clé(s)

- Confidentialité

- Authentification, contrôle d'intégrité



Algorithme Dec (resp. A) symétrique $\Rightarrow$ clé secrète

Algorithme Dec (resp. A) asymétrique $\Rightarrow$ clé publique

Cryptographie sans clé

On rappelle malgré tout qu'il existe des mécanismes cryptographiques publics ET sans clé :

- fonctions de hachage
- générateurs d'aléa

Introduction à la cryptographie

- Contexte
- Repères historiques
- Généralités
- Exemple : améliorer un canal authentifié
- Cryptographie symétrique / asymétrique
- Sécurisation de canaux de communication

Retour sur le modèle de communication

Différents types de canaux de communication parfaits :

- public (ni authentifié ni confidentiel) — p.ex. Internet [universel]
- authentifié — p.ex. (partiellement) le réseau téléphonique [voix]
- confidentiel — p.ex. le réseau postal [loi]
- authentifié et confidentiel — p.ex. le téléphone rouge [dédié]

Sécurité, disponibilité, débit, coût variables

Intervention de la cryptographie

Construire des canaux authentifiés et/ou confidentiels à partir

- d'un canal public et
- d'un canal authentifié et/ou confidentiel

Utilisation différente et/ou asynchrone des canaux $\Rightarrow$ souplesse d'utilisation, nouvelles fonctionnalités, *etc.*

Attention :

- Chiffrer $\neq$ authentifier (ce n'est pas le même modèle de sécurité)
- Un défaut d'authentification de l'origine des messages peut conduire à un défaut de confidentialité (p.ex. chiffrement malléable) !
- Il faut rajouter des mécanismes supplémentaires pour éviter les rejeux (randomisation, n° de transaction, horodatage, *etc.*)

Créer un canal confidentiel (I)

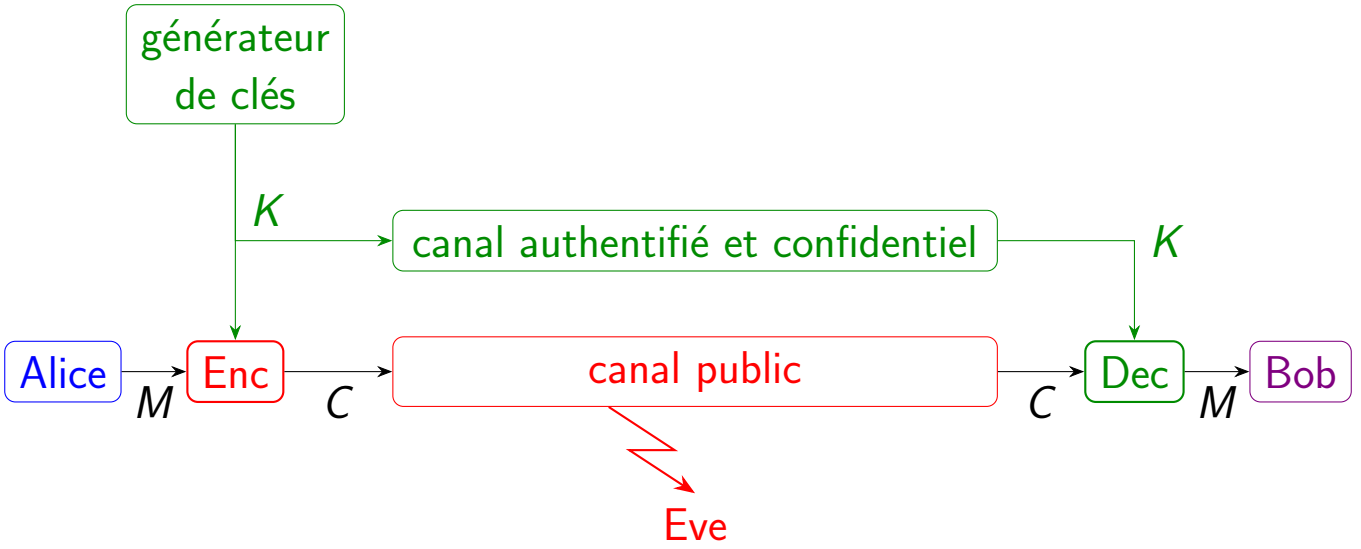
Utiliser un chiffrement à clé secrète (symétrique)

- Un canal public pour transmettre des messages chiffrés
- Un canal authentifié et confidentiel pour transmettre la clé secrète

Définition (partielle)

Un chiffrement à clé secrète, E, est un algorithme paramétré par une chaîne binaire secrète, K, partagée entre deux entités qui transforme un message clair M en un message chiffré C. Le déchiffrement associé, D, utilise le même paramètre K.

Canal confidentiel contre attaques passives (I)



- On a besoin du canal authentifié et confidentiel une fois au préalable de l'envoi d'un nombre élevé de messages

Modèle d'attaques

- Connaissant certains couples (M, C) et un chiffré C_0, un attaquant ne doit pas pouvoir
  - retrouver la clé secrète K
  - retrouver le clair M_0 correspondant à C_0 sans connaître la clé secrète K
  - distinguer l'algorithme de chiffrement d'une permutation aléatoire

Créer un canal confidentiel (II)

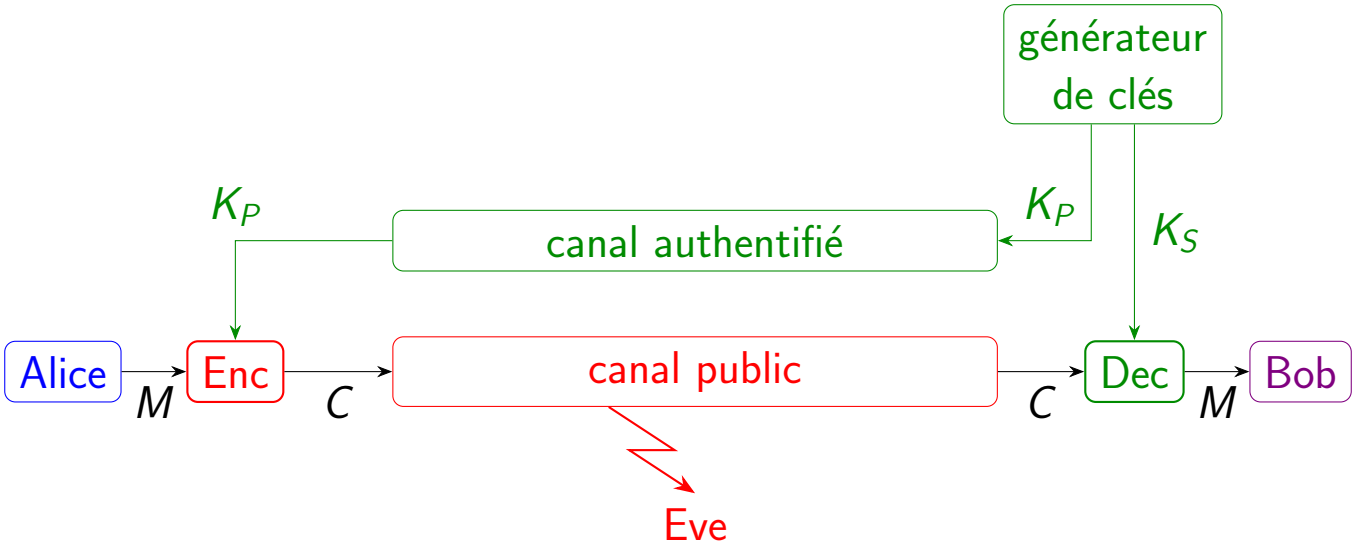
Utiliser un chiffrement à clé publique (asymétrique)

- Un canal public pour transmettre des messages chiffrés
- Un canal authentifié pour transmettre la clé publique

Définition (partielle)

Un chiffrement à clé publique, E, est un algorithme paramétré par une chaîne binaire publique, K_P, connue de tous et qui transforme un message clair M en un message chiffré C. Le déchiffrement associé D utilise un paramètre privé K_S.

Canal confidentiel contre attaques passives (II)



- On a besoin du canal authentifié une fois au préalable de l'envoi d'un nombre élevé de messages

Modèle d'attaques

- Connaissant certains couples (M, C), un chiffré C_0, et la clé publique K_P, un attaquant ne doit pas pouvoir
  - retrouver la clé privée K_S
  - retrouver le clair M_0 correspondant à C_0 sans connaître la clé privée K_S
  - distinguer l'algorithme de chiffrement d'une permutation aléatoire

Chiffrement à clé secrète : le coffre-fort

- Alice et Bob ont la clé du coffre
- Alice envoie un message à Bob
  1. Alice utilise la clé pour déposer un courrier dans le coffre
  2. Bob utilise la clé pour lire le courrier déposé par Alice
- Propriétés du coffre-fort
  - seuls Alice et Bob peuvent déposer du courrier dans le coffre
  - seuls Alice et Bob peuvent retirer le courrier déposé dans le coffre
- Attention à la métaphore : un chiffrement peut être malléable

Chiffrement à clé publique : la boîte aux lettres

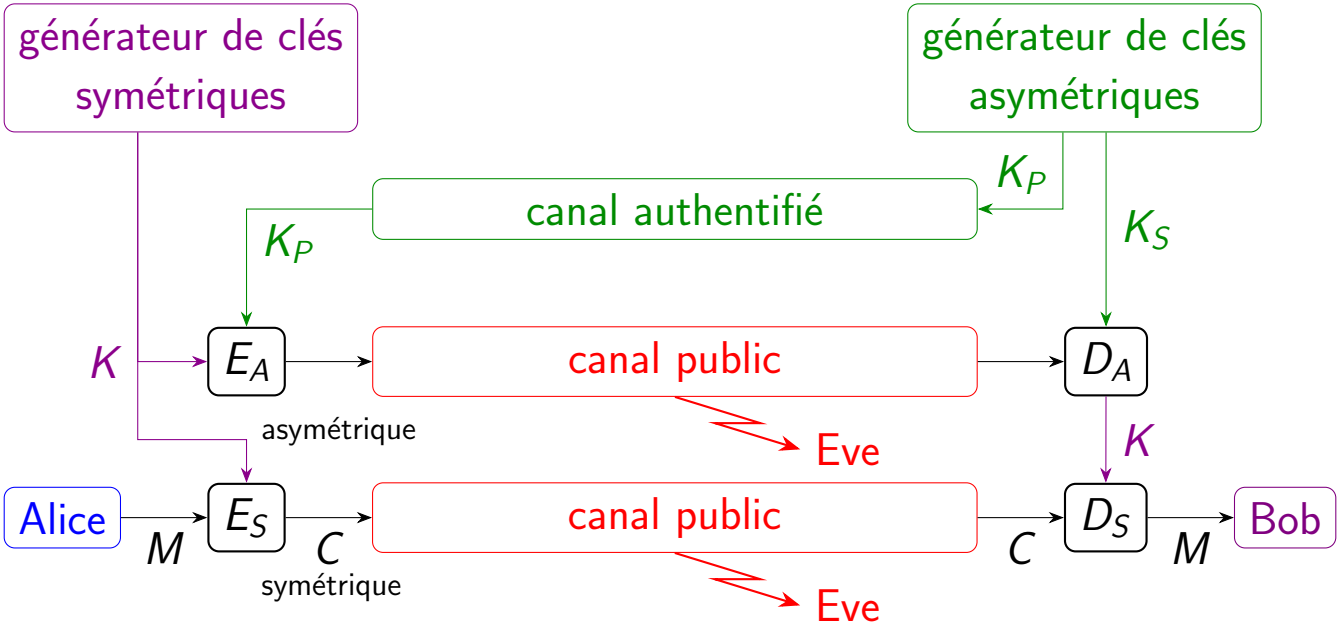
- Seul Bob a la clé de sa boîte
- Alice obtient l'adresse de Bob dans un annuaire
- Alice envoie un message à Bob
  1. Alice dépose un courrier dans la boîte de Bob
  2. Bob utilise sa clé pour retirer le courrier déposé dans sa boîte
- Propriétés de la boîte aux lettres
  - tout le monde peut envoyer du courrier à Bob
  - seul Bob peut lire le courrier déposé dans sa boîte aux lettres
- Attention à la métaphore : un chiffrement peut être malléable

Algorithmes à clé secrète / à clé publique

	Clé secrète	Clé publique
Gestion	la clé est secrète aux deux extrémités	seule la clé privée est secrète
	nombre de clés : $O(n^2)$	nombre de clés : $O(n)$
	canal auxiliaire authentifié et confidentiel	canal auxiliaire authentifié
Sécurité	pas de preuve formelle de sécurité	repose sur la difficulté supposée de problèmes mathématiques
Perf.	très rapides $\sim$ 10-100 Mo/s	très lents $\sim$ 10-100 Ko/s

Canal confidentiel : transport de clé

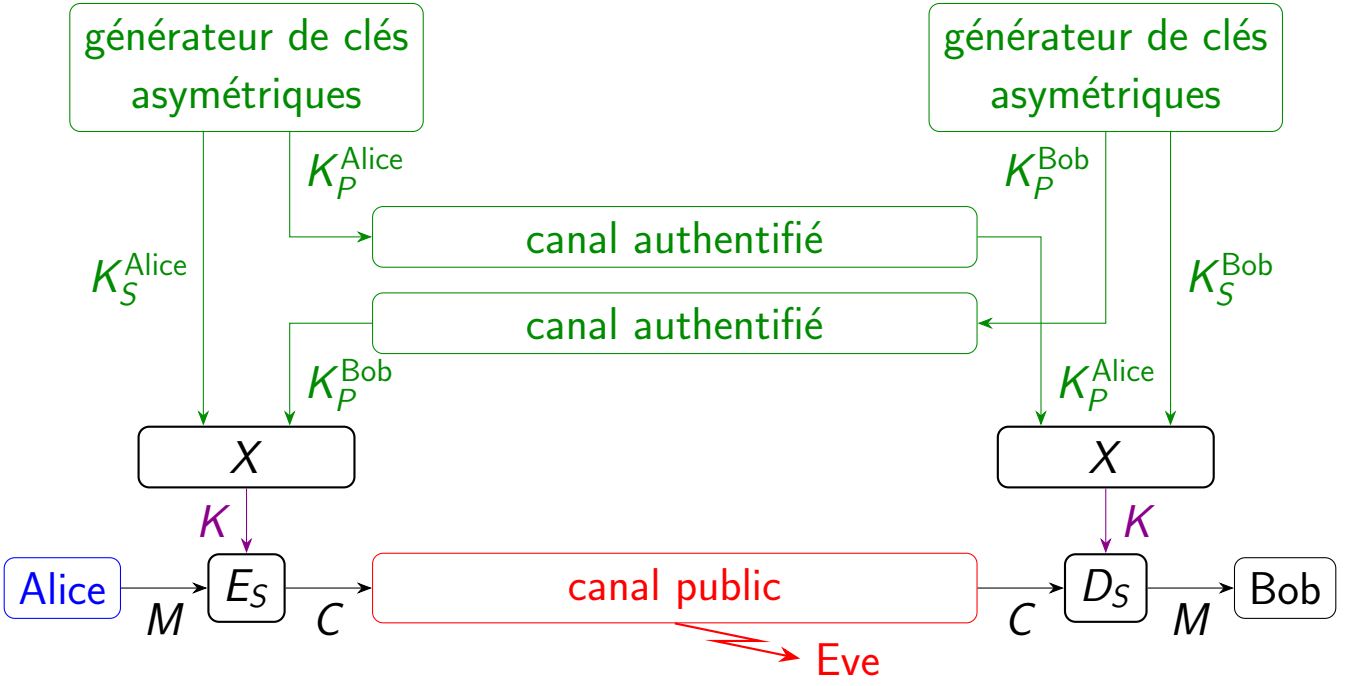
Utiliser un système hybride



- Le chiffrement asymétrique garantit la confidentialité de la clé secrète K

Canal confidentiel : échange de clés

Comment ne pas avoir à communiquer la clé K ?



Échange de clés : $K = X(K_S^{\text{Alice}}, K_P^{\text{Bob}}) = X(K_S^{\text{Bob}}, K_P^{\text{Alice}})$

Étant donnés K_P^{Alice}, K_P^{Bob}, Eve ne doit pas pouvoir calculer K, K_S^{Alice}, K_S^{Bob}

Créer un canal authentifié (I)

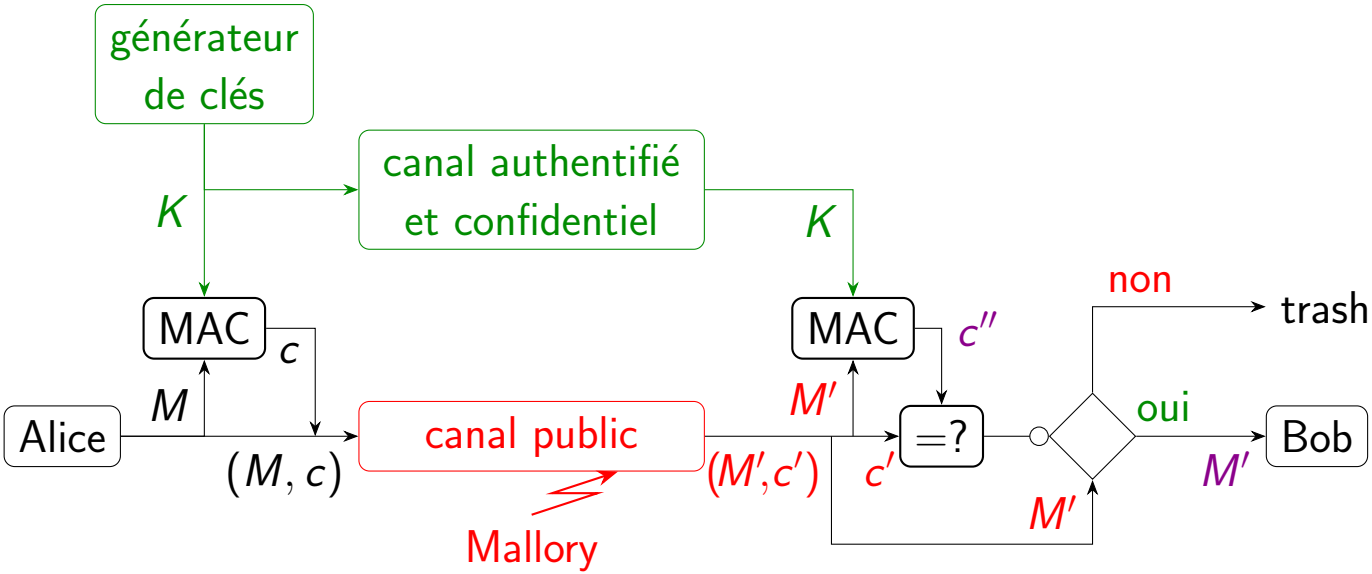
Utiliser un code d'authentification de messages (MAC)

- Un canal public pour transmettre des messages et leur code d'authentification
- Un canal authentifié et confidentiel pour transmettre la clé secrète

Définition (partielle)

Un code d'authentification de message (MAC) est un algorithme qui calcule une valeur de taille fixe, appelée (aussi) MAC, à partir de messages de taille quelconque et d'une clé secrète K partagée entre émetteur et récepteur.

Créer un canal authentifié (I)



- On a besoin du canal authentifié et confidentiel une fois au préalable de l'envoi d'un nombre élevé de messages

Modèle d'attaques

- Connaissant certains couples (M, c), un attaquant ne doit pas pouvoir
  - retrouver la clé secrète K
  - créer un nouveau couple (M', c') valide sans connaître la clé secrète K
  - distinguer l'algorithme de MAC d'une fonction aléatoire
- Le contrôle d'intégrité est assuré sur le canal public par l'authentification de l'origine des messages.

Créer un canal authentifié (II)

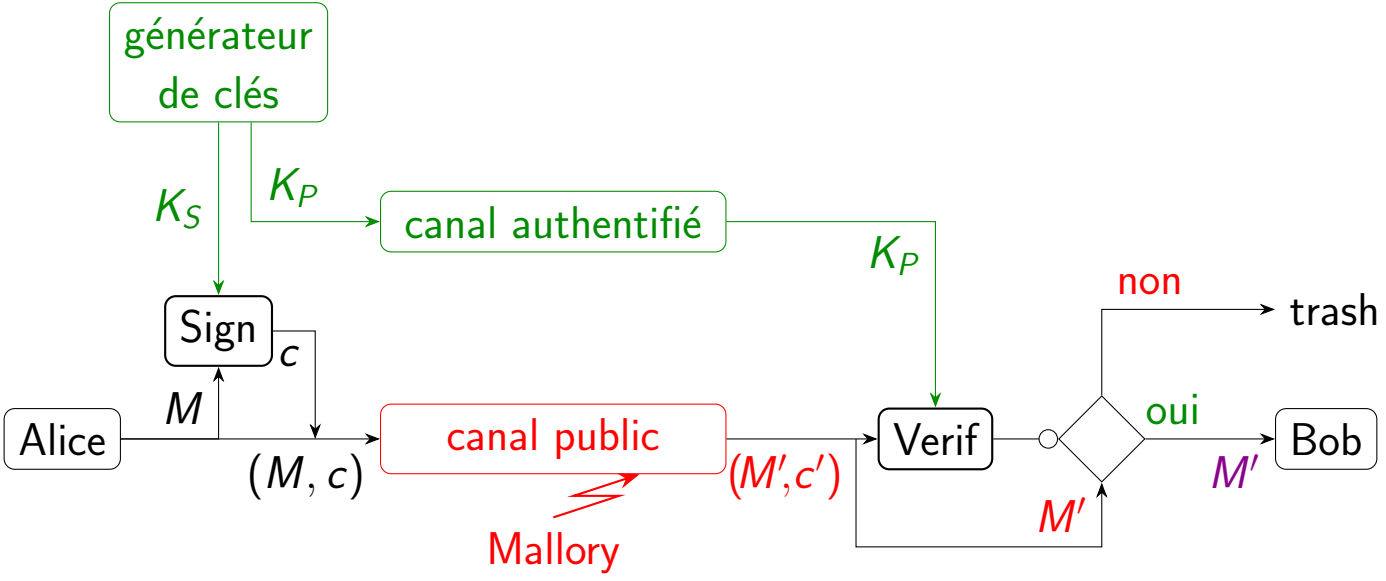
Utiliser un algorithme de signature

- Un canal public pour transmettre des messages et leur signature
- Un canal authentifié pour transmettre la clé publique

Définition (partielle)

Un algorithme de signature calcule une valeur appelée signature, de taille fixe, à partir de messages de taille quelconque et de la clé privée K_S de l'émetteur. La vérification par le récepteur se fait grâce à la clé publique K_P de l'émetteur.

Créer un canal authentifié (II)



- On a besoin du canal authentifié une fois au préalable de l'envoi d'un nombre élevé de messages

Modèle d'attaques

- Connaissant certains couples (M, c) et la clé publique K_P, un attaquant ne doit pas pouvoir
  - retrouver la clé privée K_S
  - créer un nouveau couple (M', c') valide sans connaître la clé privée K_S
  - distinguer l'algorithme de signature d'une fonction aléatoire
- Le contrôle d'intégrité est assuré sur le canal public par l'authentification de l'origine des messages.

Authentification $\Leftarrow\!\!\not\Rightarrow$ Signature

- L'authentification permet de répondre à la question :

  Qui a émis le message ?

  Pour savoir si on peut parler de signature, il faut savoir qui pose la question

- MAC : l'autre possesseur de la clé secrète $\Rightarrow$ une personne
  Deux personnes peuvent calculer l'authentifiant

- Signature : un possesseur de la clé publique $\Rightarrow$ tout le monde
  Une seule personne peut calculer l'authentifiant $\Rightarrow$ non-répudiation

Troisième partie

Mécanismes cryptographiques fondamentaux

(Slides de Marion Videau)

Mécanismes cryptographiques fondamentaux

- Mécanismes de chiffrement
- Échange de clés
- Authentification de l'origine des messages

Mécanismes cryptographiques fondamentaux

- Mécanismes de chiffrement
- Échange de clés
- Authentification de l'origine des messages

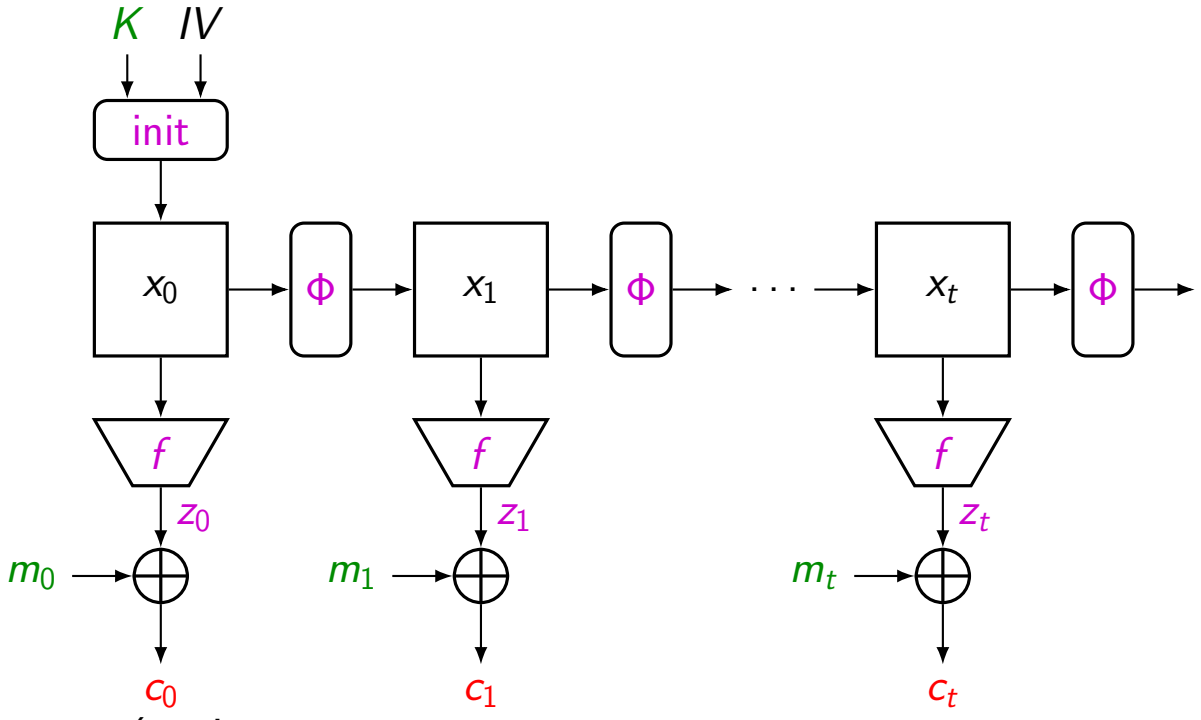
Mécanismes de chiffrement

- Chiffrement symétrique
  - chiffrement par flot : A5/1, RC4, Snow3G, etc.
  - chiffrement par bloc : DES, Blowfish, AES, etc.
  - modes de chiffrement : ECB, CBC, CFB, OFB, CTR, etc.
- Chiffrement asymétrique
  - fondé sur la difficulté de la factorisation : RSA
  - fondé sur la difficulté du logarithme discret : ElGamal
  - schéma d'encapsulation : PKCS #1

Chiffrement par flot

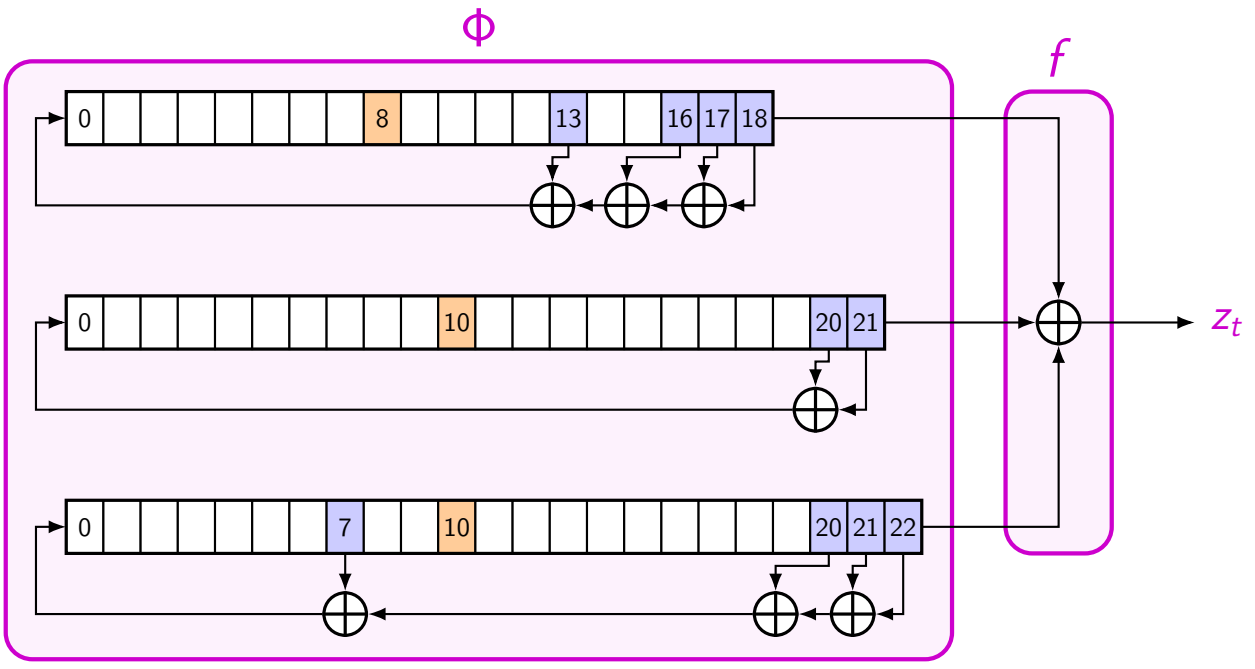
- Principe général :
  - inspiré du masque jetable (chiffrement de Vernam)
    $\rightarrow$ simuler une clé infinie (qui sera de fait finie)
  - message : suite de bits $M \in \{0,1\}^*$
  - suite chiffrante : masque pseudo-aléatoire $Z \in \{0,1\}^*$
  - chiffrement : OU-exclusif bit à bit (XOR, noté $\oplus$) entre le message et la suite chiffrante : $C = M \oplus Z \in \{0,1\}^*$
  - déchiffrement : $M = C \oplus Z$
- Suite chiffrante :
  - obtenue grâce à un générateur pseudo-aléatoire déterministe
  - longue période, et ne peut être distinguée d'une suite aléatoire
  - état initial du générateur dérivé à partir de la clé secrète K et d'un IV (*initialization vector*) aléatoire, transmis avec le chiffré
  - l'IV n'a pas à être secret mais imprédictible par l'attaquant
- Particularités :
  - Ne propage pas les erreurs de chiffrement
  - Très utilisé en Télécoms, 4G, 5G

Modélisation d'un chiffrement par flot synchrone



- x_t : état interne
- f : fonction de filtrage
- Φ : fonction de transition

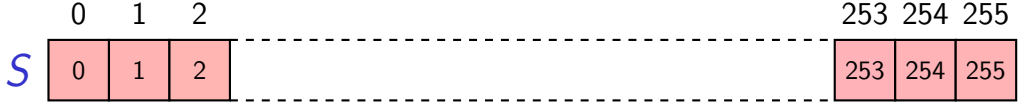
Exemple : A5/1 [1987]



- Clé secrète de 64 bits, IV de 22 bits
- Cryptanalyse en quelques minutes [Nohl & Paget, 2009]
- Snowden : *NSA « can process encrypted A5/1 »*
- Toujours utilisé dans des milliards de téléphones portables (GSM)

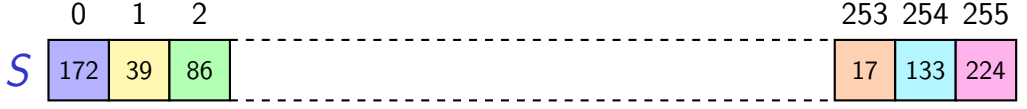
Exemple : RC4 [Rivest, 1987]

- Utilisé dans SSL/TLS, SSH, WEP, WPA, etc.
- Très efficace en logiciel, mais nombreux biais statistiques sur Z
- Clé secrète de 40 à 2048 bits, pas d'IV en tant que tel
- État interne : inspiré du shuffle (permutation aléatoire) de Knuth
  - tableau S de 256 octets : permutation de $\{0, 1, \ldots, 255\}$
  - indices i et $j \in \{0, 1, \ldots, 255\}$



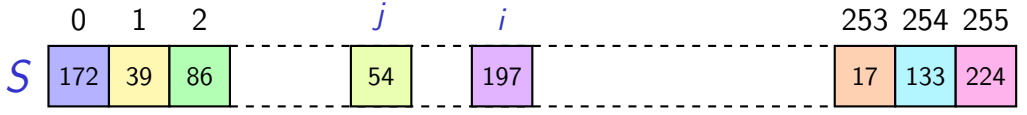
Exemple : RC4 [Rivest, 1987]

- Utilisé dans SSL/TLS, SSH, WEP, WPA, etc.
- Très efficace en logiciel, mais nombreux biais statistiques sur Z
- Clé secrète de 40 à 2048 bits, pas d'IV en tant que tel
- État interne : inspiré du shuffle (permutation aléatoire) de Knuth
  - tableau S de 256 octets : permutation de $\{0, 1, \ldots, 255\}$
  - indices i et $j \in \{0, 1, \ldots, 255\}$



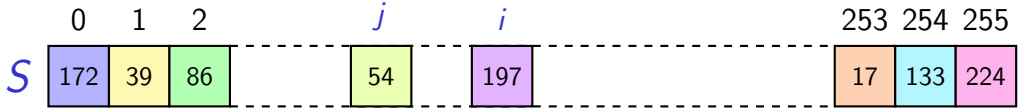
Exemple : RC4 [Rivest, 1987]

- Utilisé dans SSL/TLS, SSH, WEP, WPA, etc.
- Très efficace en logiciel, mais nombreux biais statistiques sur Z
- Clé secrète de 40 à 2048 bits, pas d'IV en tant que tel
- État interne : inspiré du shuffle (permutation aléatoire) de Knuth
  - tableau S de 256 octets : permutation de $\{0, 1, \ldots, 255\}$
  - indices i et $j \in \{0, 1, \ldots, 255\}$



Exemple : RC4 [Rivest, 1987]

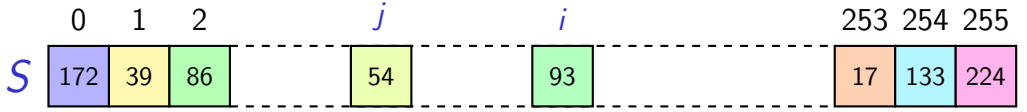
- Utilisé dans SSL/TLS, SSH, WEP, WPA, etc.
- Très efficace en logiciel, mais nombreux biais statistiques sur Z
- Clé secrète de 40 à 2048 bits, pas d'IV en tant que tel
- État interne : inspiré du shuffle (permutation aléatoire) de Knuth
  - tableau S de 256 octets : permutation de $\{0, 1, \ldots, 255\}$
  - indices i et $j \in \{0, 1, \ldots, 255\}$



- Fonction de transition Φ :
  - $i \leftarrow (i + 1) \bmod 256$
  - $j \leftarrow (j + S[i]) \bmod 256$
  - échanger les valeurs de $S[i]$ et $S[j]$

Exemple : RC4 [Rivest, 1987]

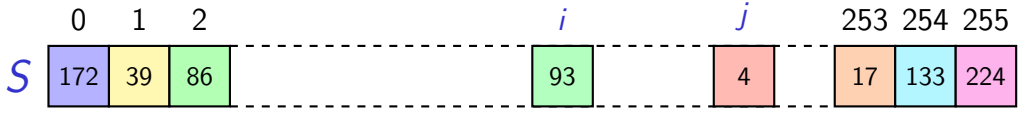
- Utilisé dans SSL/TLS, SSH, WEP, WPA, etc.
- Très efficace en logiciel, mais nombreux biais statistiques sur Z
- Clé secrète de 40 à 2048 bits, pas d'IV en tant que tel
- État interne : inspiré du shuffle (permutation aléatoire) de Knuth
  - tableau S de 256 octets : permutation de $\{0, 1, \ldots, 255\}$
  - indices i et $j \in \{0, 1, \ldots, 255\}$



- Fonction de transition Φ :
  - $i \leftarrow (i + 1) \bmod 256$
  - $j \leftarrow (j + S[i]) \bmod 256$
  - échanger les valeurs de $S[i]$ et $S[j]$

Exemple : RC4 [Rivest, 1987]

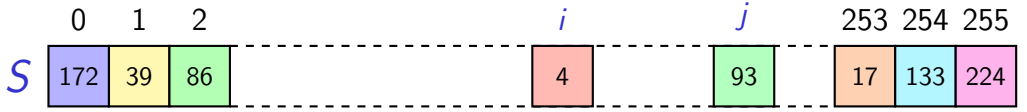
- Utilisé dans SSL/TLS, SSH, WEP, WPA, etc.
- Très efficace en logiciel, mais nombreux biais statistiques sur Z
- Clé secrète de 40 à 2048 bits, pas d'IV en tant que tel
- État interne : inspiré du shuffle (permutation aléatoire) de Knuth
  - tableau S de 256 octets : permutation de $\{0, 1, \ldots, 255\}$
  - indices i et $j \in \{0, 1, \ldots, 255\}$



- Fonction de transition Φ :
  - $i \leftarrow (i + 1) \bmod 256$
  - $j \leftarrow (j + S[i]) \bmod 256$
  - échanger les valeurs de $S[i]$ et $S[j]$

Exemple : RC4 [Rivest, 1987]

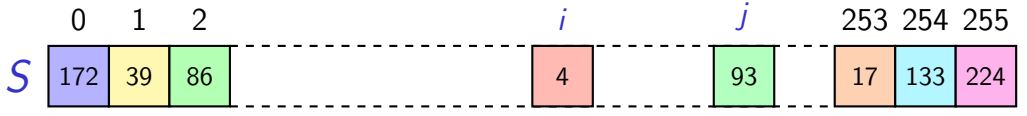
- Utilisé dans SSL/TLS, SSH, WEP, WPA, etc.
- Très efficace en logiciel, mais nombreux biais statistiques sur Z
- Clé secrète de 40 à 2048 bits, pas d'IV en tant que tel
- État interne : inspiré du shuffle (permutation aléatoire) de Knuth
  - tableau S de 256 octets : permutation de $\{0, 1, \ldots, 255\}$
  - indices i et $j \in \{0, 1, \ldots, 255\}$



- Fonction de transition Φ :
  - $i \leftarrow (i + 1) \bmod 256$
  - $j \leftarrow (j + S[i]) \bmod 256$
  - échanger les valeurs de $S[i]$ et $S[j]$

Exemple : RC4 [Rivest, 1987]

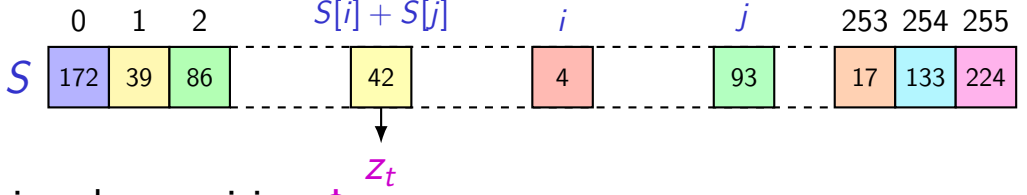
- Utilisé dans SSL/TLS, SSH, WEP, WPA, etc.
- Très efficace en logiciel, mais nombreux biais statistiques sur Z
- Clé secrète de 40 à 2048 bits, pas d'IV en tant que tel
- État interne : inspiré du shuffle (permutation aléatoire) de Knuth
  - tableau S de 256 octets : permutation de $\{0, 1, \ldots, 255\}$
  - indices i et $j \in \{0, 1, \ldots, 255\}$



- Fonction de transition Φ :
  - $i \leftarrow (i + 1) \bmod 256$
  - $j \leftarrow (j + S[i]) \bmod 256$
  - échanger les valeurs de $S[i]$ et $S[j]$
- Fonction de filtrage f : $z_t \leftarrow S[(S[i] + S[j]) \bmod 256]$

Exemple : RC4 [Rivest, 1987]

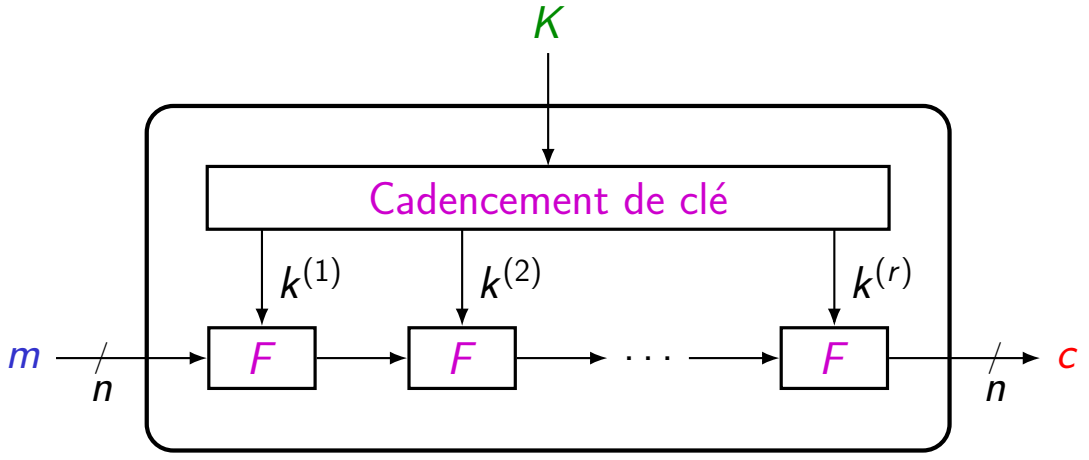
- Utilisé dans SSL/TLS, SSH, WEP, WPA, etc.
- Très efficace en logiciel, mais nombreux biais statistiques sur Z
- Clé secrète de 40 à 2048 bits, pas d'IV en tant que tel
- État interne : inspiré du shuffle (permutation aléatoire) de Knuth
  - tableau S de 256 octets : permutation de $\{0, 1, \ldots, 255\}$
  - indices i et $j \in \{0, 1, \ldots, 255\}$



- Fonction de transition Φ :
  - $i \leftarrow (i + 1) \bmod 256$
  - $j \leftarrow (j + S[i]) \bmod 256$
  - échanger les valeurs de $S[i]$ et $S[j]$
- Fonction de filtrage f : $z_t \leftarrow S[(S[i] + S[j]) \bmod 256]$

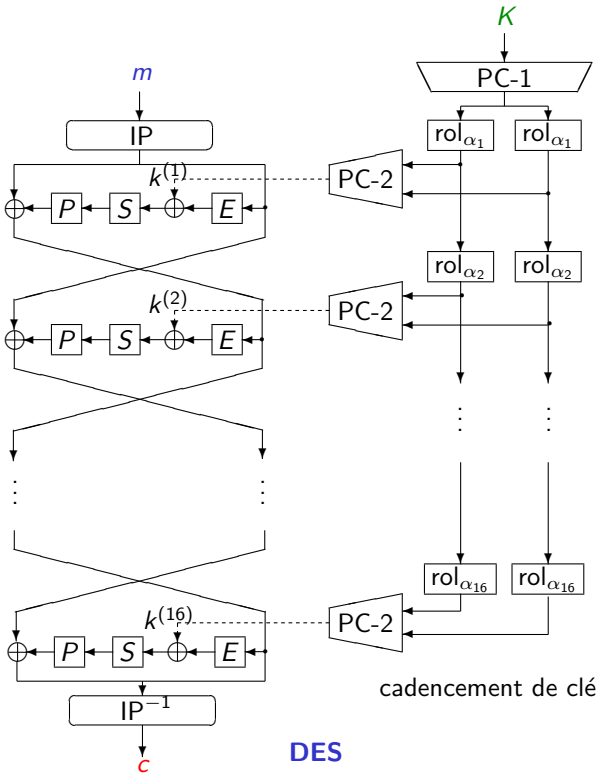
Modélisation d'un chiffrement itératif par bloc

- Famille de permutations paramétrées constituée de r itérations (ou tours) sur des blocs de n bits



- Nécessite l'utilisation de modes (ECB, CBC, CTR, etc.)

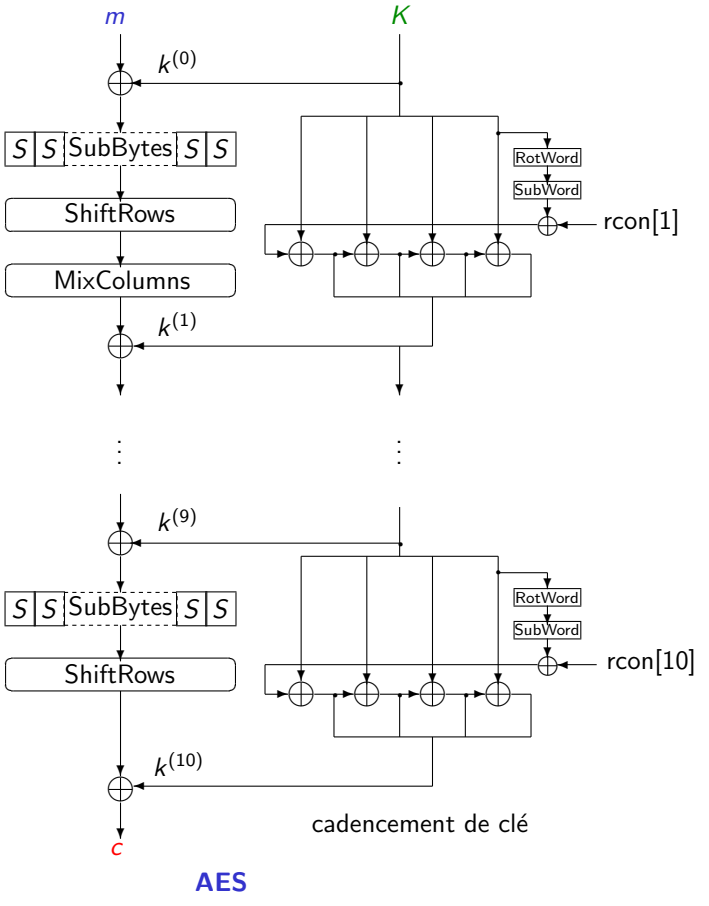
Exemple historique : DES [1977]



- Taille de clé : 56 bits
- Taille des blocs : 64 bits
- Chiffrement de type schéma de Feistel
- 16 tours
- Attaque par force brute en 56 h [EFF, 1998]
- Utilisation du triple-DES :

$$c = E_{K_3}(D_{K_2}(E_{K_1}(m)))$$

Standard actuel : AES [2001]



AES

- Taille de clé : 128, 192 ou 256 bits
- Taille des blocs : 128 bits
- Chiffrement de type réseau S-P
- 10, 12, 14 tours selon taille de clé
- Nettement plus rapide que DES
- Instructions dédiées sur processeurs récents (AES-NI)

Standard actuel : AES [2001]

http://www.formaestudio.com/rijndaelinspector/

Animation :

https://formaestudio.com/rijndaelinspector/archivos/Rijndael_Animation_v4_eng-html5.html

Modes de chiffrement

- Pour chiffrer un long message M :
  - on découpe M en $\ell = \lceil |M|/n \rceil$ blocs de n bits chacun

$$M = m_1 \| m_2 \| \dots \| m_\ell$$

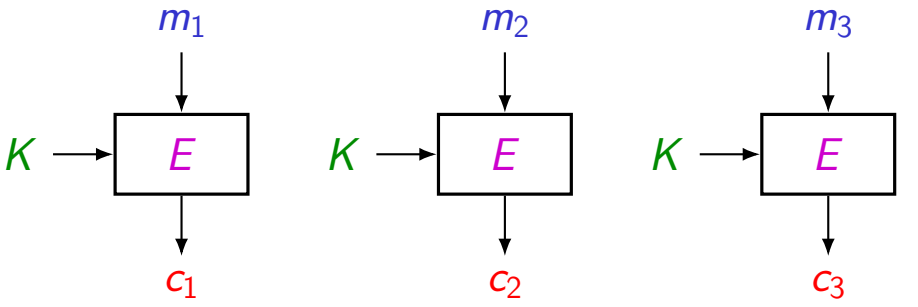
  - comment remplir le dernier bloc s'il n'est pas complet ?
    $\rightarrow$ question du *padding* (bourrage)
  - et ensuite ?...
- Première idée :
  - on calcule $c_i = E_K(m_i)$ pour i de 1 à ℓ
  - on obtient le chiffré

$$C = c_1 \| c_2 \| \dots \| c_\ell$$

  - pour déchiffrer, on fait la même chose dans l'autre sens
  - c'est le mode ECB (pour *Electronic Codebook*)

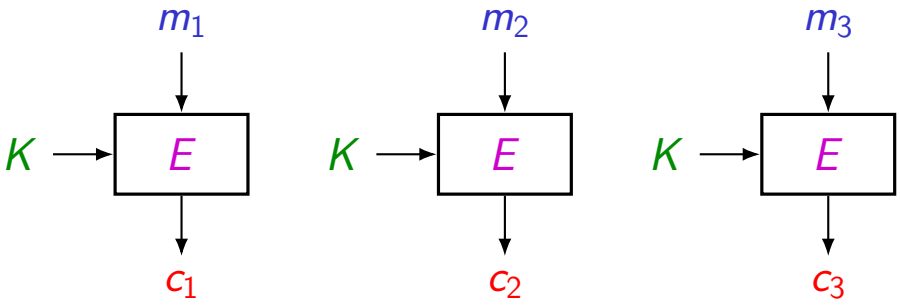
Mode de chiffrement ECB

- Chiffrement :

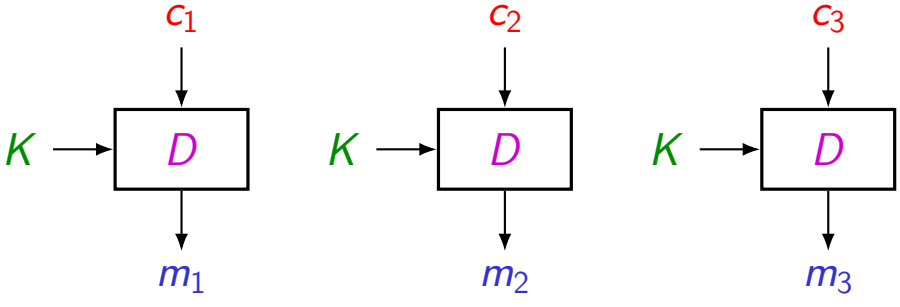


Mode de chiffrement ECB

- Chiffrement :



- Déchiffrement :

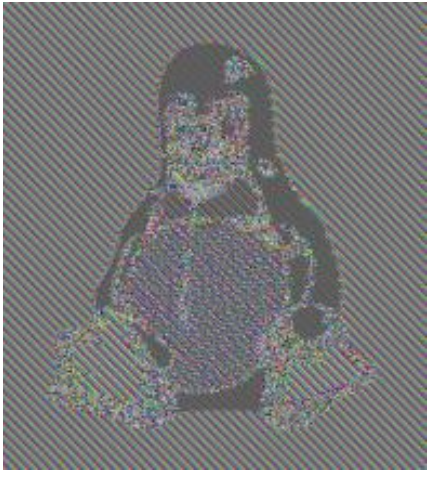


Mode de chiffrement ECB

- Problème :
  - si deux blocs m_i et m_j du message sont identiques...
  - alors les blocs chiffrés le seront aussi : $c_i = c_j$

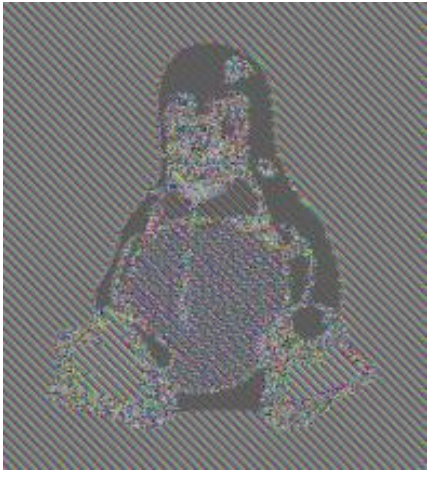
Mode de chiffrement ECB

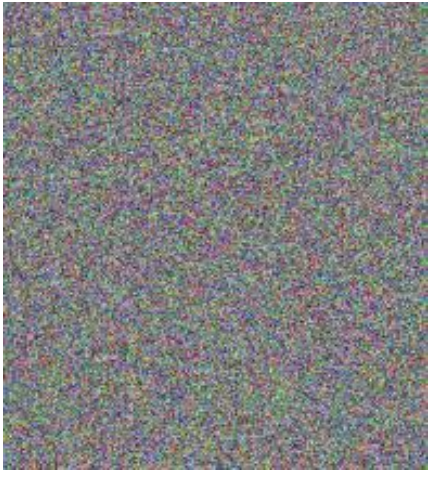
- Problème :
  - si deux blocs m_i et m_j du message sont identiques...
  - alors les blocs chiffrés le seront aussi : $c_i = c_j$



Mode de chiffrement ECB

- Problème :
  - si deux blocs m_i et m_j du message sont identiques...
  - alors les blocs chiffrés le seront aussi : $c_i = c_j$

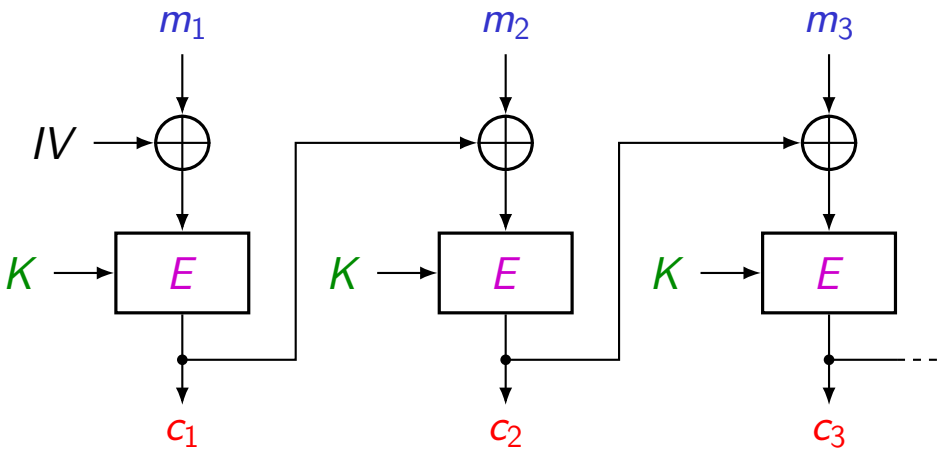




(Source : Wikipédia)

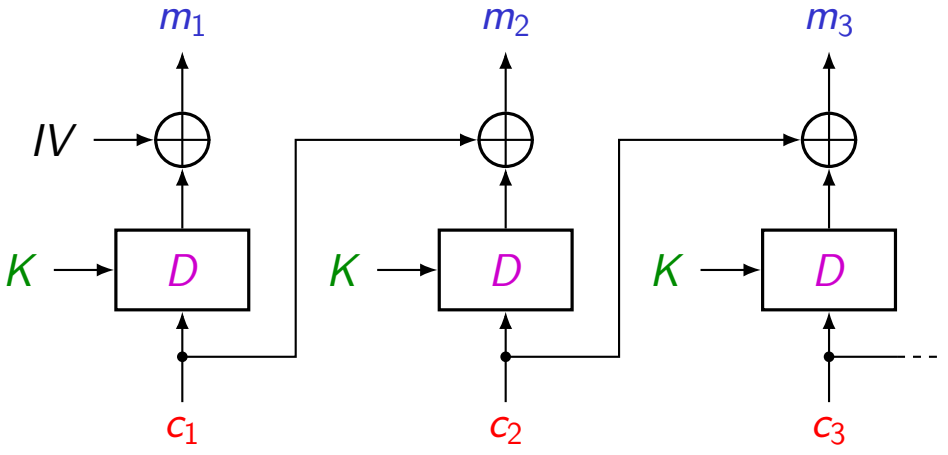
Exemples de modes de chiffrement

- Mode CBC (pour *Cipher Block Chaining*)
  - chiffrement séquentiel



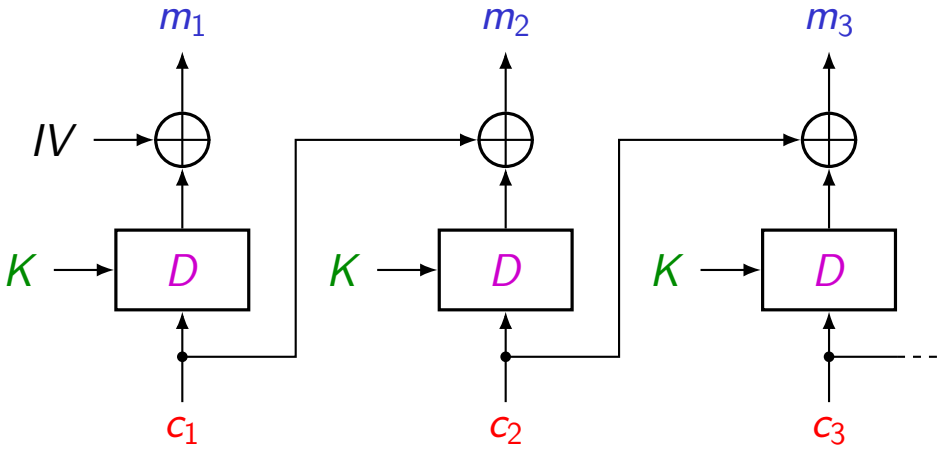
Exemples de modes de chiffrement

- Mode CBC (pour *Cipher Block Chaining*)
  - chiffrement séquentiel
  - déchiffrement parallèle



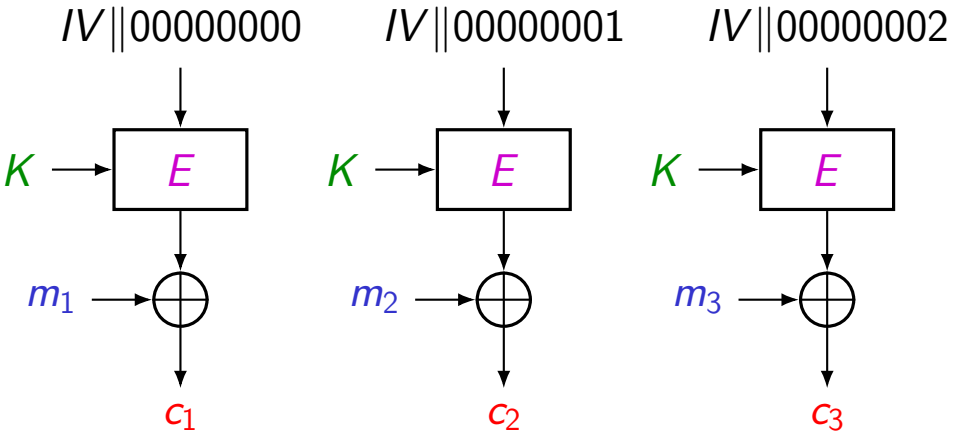
Exemples de modes de chiffrement

- Mode CBC (pour *Cipher Block Chaining*)
  - chiffrement séquentiel
  - déchiffrement parallèle
  - vulnérable à des attaques sur le *padding*



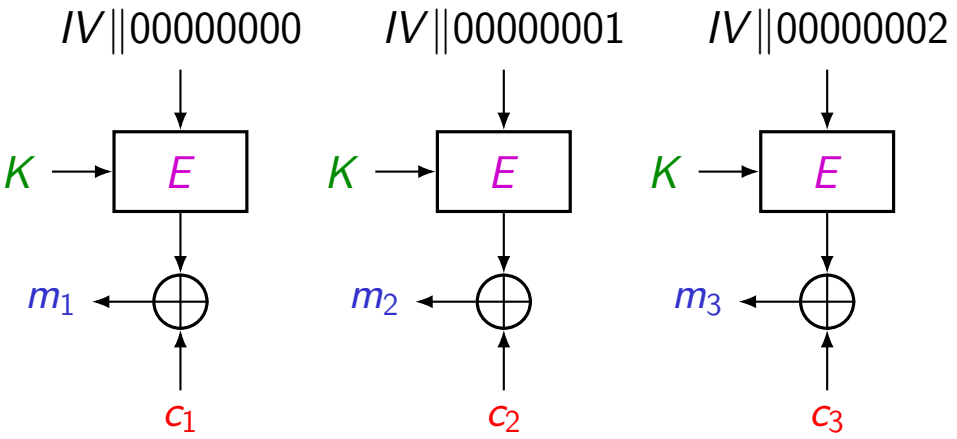
Exemples de modes de chiffrement

- Mode CTR (pour *Counter*)
  - transforme un chiffrement par bloc en chiffrement par flot
    Seul mode qui ne propage pas les erreurs d'un bloc à l'autre
  - chiffrement parallèle



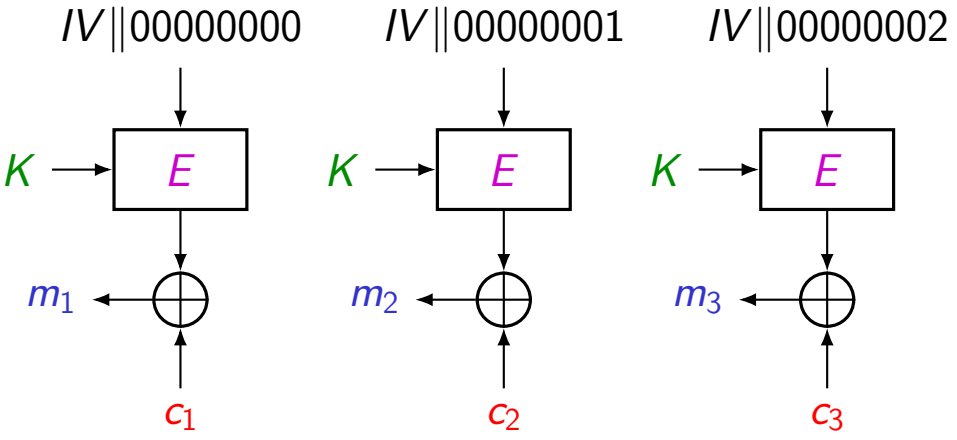
Exemples de modes de chiffrement

- Mode CTR (pour *Counter*)
  - transforme un chiffrement par bloc en chiffrement par flot
    Seul mode qui ne propage pas les erreurs d'un bloc à l'autre
  - chiffrement parallèle
  - déchiffrement parallèle



Exemples de modes de chiffrement

- Mode CTR (pour *Counter*)
  - transforme un chiffrement par bloc en chiffrement par flot
    Seul mode qui ne propage pas les erreurs d'un bloc à l'autre
  - chiffrement parallèle
  - déchiffrement parallèle
  - 3G



Chiffrement asymétrique : RSA

- [Rivest, Shamir & Adleman, 1978]
- Idée :
  - $N = p \times q$, avec p et q de grands nombres premiers, alors il est « difficile » de retrouver p et q à partir de N
  - espace des messages : $m \in \mathbb{Z}/N\mathbb{Z}$
  - pour (presque) tout entier e, il existe un unique entier d tel que

$$m^{ed} \bmod N = m, \text{ pour tout message } m \in \mathbb{Z}/N\mathbb{Z}$$

Chiffrement asymétrique : RSA

- Génération de clés (paramètre de sécurité : n, taille du module) :
  - choisir au hasard deux nombres premiers p et q de $n/2$ bits chacun, et calculer le module $N = p \times q$ de n bits
  - choisir un exposant public e co-premier avec $p-1$ et $q-1$
  - calculer l'exposant privé d tel que $ed \equiv 1 \pmod{(p-1)(q-1)}$
  - clé publique : (e, N) ; et clé privée : (d, N)

Chiffrement asymétrique : RSA

- Génération de clés (paramètre de sécurité : n, taille du module) :
  - choisir au hasard deux nombres premiers p et q de $n/2$ bits chacun, et calculer le module $N = p \times q$ de n bits
  - choisir un exposant public e co-premier avec $p-1$ et $q-1$
  - calculer l'exposant privé d tel que $ed \equiv 1 \pmod{(p-1)(q-1)}$
  - clé publique : (e, N) ; et clé privée : (d, N)
- Chiffrement d'un message $m \in \mathbb{Z}/N\mathbb{Z}$: $c = m^e \bmod N$

Chiffrement asymétrique : RSA

- Génération de clés (paramètre de sécurité : n, taille du module) :
  - choisir au hasard deux nombres premiers p et q de $n/2$ bits chacun, et calculer le module $N = p \times q$ de n bits
  - choisir un exposant public e co-premier avec $p-1$ et $q-1$
  - calculer l'exposant privé d tel que $ed \equiv 1 \pmod{(p-1)(q-1)}$
  - clé publique : (e, N) ; et clé privée : (d, N)
- Chiffrement d'un message $m \in \mathbb{Z}/N\mathbb{Z}$: $c = m^e \bmod N$
- Déchiffrement d'un chiffré c : $c^d \bmod N = (m^e)^d \bmod N = m$

Chiffrement asymétrique : RSA

- Génération de clés (paramètre de sécurité : n, taille du module) :
  - choisir au hasard deux nombres premiers p et q de $n/2$ bits chacun, et calculer le module $N = p \times q$ de n bits
  - choisir un exposant public e co-premier avec $p-1$ et $q-1$
  - calculer l'exposant privé d tel que $ed \equiv 1 \pmod{(p-1)(q-1)}$
  - clé publique : (e, N) ; et clé privée : (d, N)
- Chiffrement d'un message $m \in \mathbb{Z}/N\mathbb{Z}$: $c = m^e \bmod N$
- Déchiffrement d'un chiffré c : $c^d \bmod N = (m^e)^d \bmod N = m$
- Sécurité :
  - retrouver d est équivalent à factoriser N
  - on ne sait pas si retrouver m à partir de c, e et N est équivalent à la factorisation
  - chiffrement malléable : si $c_1 = E^{\mathrm{RSA}}_{e,N}(m_1)$ et $c_2 = E^{\mathrm{RSA}}_{e,N}(m_2)$, alors $c_1 c_2 \bmod N = E^{\mathrm{RSA}}_{e,N}(m_1 m_2 \bmod N)$

RSA : taille du module N

- La factorisation de N est la seule méthode connue pour attaquer RSA
  - meilleur algorithme connu : NFS (*Number Field Sieve*)
  - complexité sous-exponentielle :
    elle croît exponentiellement en $\sqrt[3]{\text{taille de } N} = \sqrt[3]{n}$
  - on ne sait pas prouver qu'il n'existe pas d'algorithme polynomial
    (d'ailleurs, algo. polynomial sur l'ordinateur quantique [Schor, 1994])

RSA : taille du module N

- La factorisation de N est la seule méthode connue pour attaquer RSA
  - meilleur algorithme connu : NFS (*Number Field Sieve*)
  - complexité sous-exponentielle :
    elle croît exponentiellement en $\sqrt[3]{\text{taille de } N} = \sqrt[3]{n}$
  - on ne sait pas prouver qu'il n'existe pas d'algorithme polynomial
    (d'ailleurs, algo. polynomial sur l'ordinateur quantique [Schor, 1994])
- En pratique, où en est-on ? https://members.loria.fr/PZimmermann/records/factor.html
  - 512 bits : factorisation en 7,5 h pour $\sim$ 100$ sur Amazon EC2
    taille des clés RSA EXPORT dans SSL 3.0 $\rightarrow$ attaque FREAK
  - 768 bits (232 dd) : [2009]

RSA : taille du module N

- La factorisation de N est la seule méthode connue pour attaquer RSA
  - meilleur algorithme connu : NFS (*Number Field Sieve*)
  - complexité sous-exponentielle :
    elle croît exponentiellement en $\sqrt[3]{\text{taille de } N} = \sqrt[3]{n}$
  - on ne sait pas prouver qu'il n'existe pas d'algorithme polynomial
    (d'ailleurs, algo. polynomial sur l'ordinateur quantique [Schor, 1994])
- En pratique, où en est-on ? https://members.loria.fr/PZimmermann/records/factor.html
  - 512 bits : factorisation en 7,5 h pour $\sim$ 100$ sur Amazon EC2
    taille des clés RSA EXPORT dans SSL 3.0 $\rightarrow$ attaque FREAK
  - 768 bits (232 dd) : [2009]
  - 795 bits (240 dd) : [2019] équipe internationale dont Nancy, suivi de
  - 829 bits (250 dd) : record [2020] de factorisation d'un module RSA
    $\sim$ 2700 années de calcul sur un cœur Xeon Gold 6130 à 2.1 GHz, de l'ordre de $2^{67.3}$ op.

RSA : taille du module N

- La factorisation de N est la seule méthode connue pour attaquer RSA
  - meilleur algorithme connu : NFS (*Number Field Sieve*)
  - complexité sous-exponentielle :
    elle croît exponentiellement en $\sqrt[3]{\text{taille de } N} = \sqrt[3]{n}$
  - on ne sait pas prouver qu'il n'existe pas d'algorithme polynomial
    (d'ailleurs, algo. polynomial sur l'ordinateur quantique [Schor, 1994])
- En pratique, où en est-on ? https://members.loria.fr/PZimmermann/records/factor.html
  - 512 bits : factorisation en 7,5 h pour $\sim$ 100$ sur Amazon EC2
    taille des clés RSA EXPORT dans SSL 3.0 $\rightarrow$ attaque FREAK
  - 768 bits (232 dd) : [2009]
  - 795 bits (240 dd) : [2019] équipe internationale dont Nancy, suivi de
  - 829 bits (250 dd) : record [2020] de factorisation d'un module RSA
    $\sim$ 2700 années de calcul sur un cœur Xeon Gold 6130 à 2.1 GHz, de l'ordre de $2^{67.3}$ op.
  - 1024 bits : correspond à 2^{75} opérations $\rightarrow$ à éviter !
  - 2048 bits : $\sim 2^{105}$ opérations, suffisant jusqu'en 2020 [ANSSI]
  - 3072 bits : $\sim 2^{127}$ opérations
  - 4096 bits : $\sim 2^{145}$ opérations

Chiffrement asymétrique : ElGamal [1985]

- Génération des paramètres et des clés (paramètre de sécurité : n) :
  - un nombre premier p de n bits
  - un générateur g du groupe $(\mathbb{Z}/p\mathbb{Z})^*$
  - un exposant secret x tiré au hasard dans $(\mathbb{Z}/(p-1)\mathbb{Z})^*$
  - l'élément public $y = g^x \bmod p$
  - paramètres : p et g ; clé publique : y ; clé privée : x

Chiffrement asymétrique : ElGamal [1985]

- Génération des paramètres et des clés (paramètre de sécurité : n) :
  - un nombre premier p de n bits
  - un générateur g du groupe $(\mathbb{Z}/p\mathbb{Z})^*$
  - un exposant secret x tiré au hasard dans $(\mathbb{Z}/(p-1)\mathbb{Z})^*$
  - l'élément public $y = g^x \bmod p$
  - paramètres : p et g ; clé publique : y ; clé privée : x
- Chiffrement d'un message $m \in (\mathbb{Z}/p\mathbb{Z})^*$ en $c = (u, v)$:
  - tirer r au hasard dans $(\mathbb{Z}/(p-1)\mathbb{Z})^*$
  - calculer $u = g^r \bmod p$ et $v = my^r \bmod p$

Chiffrement asymétrique : ElGamal [1985]

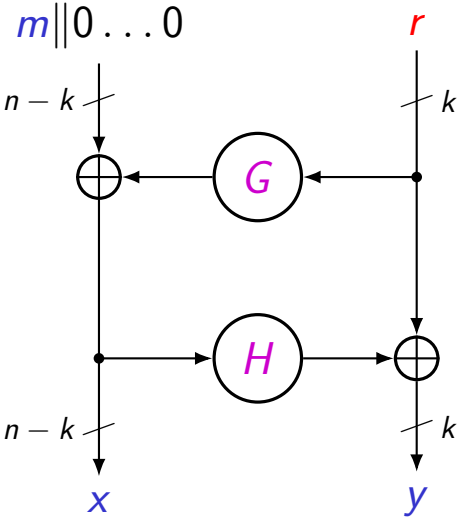
- Génération des paramètres et des clés (paramètre de sécurité : n) :
  - un nombre premier p de n bits
  - un générateur g du groupe $(\mathbb{Z}/p\mathbb{Z})^*$
  - un exposant secret x tiré au hasard dans $(\mathbb{Z}/(p-1)\mathbb{Z})^*$
  - l'élément public $y = g^x \bmod p$
  - paramètres : p et g ; clé publique : y ; clé privée : x
- Chiffrement d'un message $m \in (\mathbb{Z}/p\mathbb{Z})^*$ en $c = (u, v)$:
  - tirer r au hasard dans $(\mathbb{Z}/(p-1)\mathbb{Z})^*$
  - calculer $u = g^r \bmod p$ et $v = my^r \bmod p$
- Déchiffrement d'un chiffré : $vu^{-x} \bmod p = my^r g^{-rx} \bmod p = m$

Chiffrement asymétrique : ElGamal [1985]

- Génération des paramètres et des clés (paramètre de sécurité : n) :
  - un nombre premier p de n bits
  - un générateur g du groupe $(\mathbb{Z}/p\mathbb{Z})^*$
  - un exposant secret x tiré au hasard dans $(\mathbb{Z}/(p-1)\mathbb{Z})^*$
  - l'élément public $y = g^x \bmod p$
  - paramètres : p et g ; clé publique : y ; clé privée : x
- Chiffrement d'un message $m \in (\mathbb{Z}/p\mathbb{Z})^*$ en $c = (u, v)$:
  - tirer r au hasard dans $(\mathbb{Z}/(p-1)\mathbb{Z})^*$
  - calculer $u = g^r \bmod p$ et $v = my^r \bmod p$
- Déchiffrement d'un chiffré : $vu^{-x} \bmod p = my^r g^{-rx} \bmod p = m$
- Sécurité :
  - retrouver x revient à calculer le logarithme discret de y modulo p
    $\rightarrow$ meilleur algorithme connu : NFS-DL, sous-exponentiel
  - retrouver m à partir de u, v, y et des paramètres p et g est équivalent au problème de Diffie-Hellman
  - chiffrement malléable : $c_1 c_2 = E_y^{\mathsf{ElGamal}_{p,g}}(m_1 m_2)$

Exemple d'encapsulation : OAEP

- *Optimal Asymmetric Encryption Padding* [Bellare & Rogaway, 1994]
  - r : chaîne aléatoire de k bits
  - G et H : fonctions de hachage cryptographiques
  - message final (avant chiffrement) : $m' = x \| y$



Mécanismes cryptographiques fondamentaux

- Mécanismes de chiffrement
- Échange de clés
- Authentification de l'origine des messages

Échange de clés Diffie-Hellman

- Génération des paramètres communs (paramètre de sécurité : n) :
  - choisir un nombre premier p de n bits
  - choisir un générateur g du groupe $(\mathbb{Z}/p\mathbb{Z})^*$

Échange de clés Diffie-Hellman

- Génération des paramètres communs (paramètre de sécurité : n) :
  - choisir un nombre premier p de n bits
  - choisir un générateur g du groupe $(\mathbb{Z}/p\mathbb{Z})^*$
- Génération des clés :
  - Alice choisit au hasard $a \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$ et calcule $k_A = g^a \bmod p$
  - Bob choisit au hasard $b \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$ et calcule $k_B = g^b \bmod p$

Échange de clés Diffie-Hellman

- Génération des paramètres communs (paramètre de sécurité : n) :
  - choisir un nombre premier p de n bits
  - choisir un générateur g du groupe $(\mathbb{Z}/p\mathbb{Z})^*$
- Génération des clés :
  - Alice choisit au hasard $a \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$ et calcule $k_A = g^a \bmod p$
  - Bob choisit au hasard $b \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$ et calcule $k_B = g^b \bmod p$
- Échange des clés : Alice et Bob s'envoient leurs clés publiques k_A et k_B
- Calcul de la clé secrète partagée :
  - Alice calcule $k = {k_B}^a \bmod p = g^{ab} \bmod p$
  - Bob calcule $k = {k_A}^b \bmod p = g^{ab} \bmod p$

Échange de clés Diffie-Hellman

- Génération des paramètres communs (paramètre de sécurité : n) :
  - choisir un nombre premier p de n bits
  - choisir un générateur g du groupe $(\mathbb{Z}/p\mathbb{Z})^*$
- Génération des clés :
  - Alice choisit au hasard $a \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$ et calcule $k_A = g^a \bmod p$
  - Bob choisit au hasard $b \in (\mathbb{Z}/(p-1)\mathbb{Z})^*$ et calcule $k_B = g^b \bmod p$
- Échange des clés : Alice et Bob s'envoient leurs clés publiques k_A et k_B
- Calcul de la clé secrète partagée :
  - Alice calcule $k = {k_B}^a \bmod p = g^{ab} \bmod p$
  - Bob calcule $k = {k_A}^b \bmod p = g^{ab} \bmod p$
- Sécurité :
  - retrouver a à partir de k_A revient à calculer le logarithme discret de k_A dans $(\mathbb{Z}/p\mathbb{Z})^*$
  - retrouver k à partir de k_A, k_B et des paramètres p et g est appelé problème de Diffie-Hellman

Mécanismes cryptographiques fondamentaux

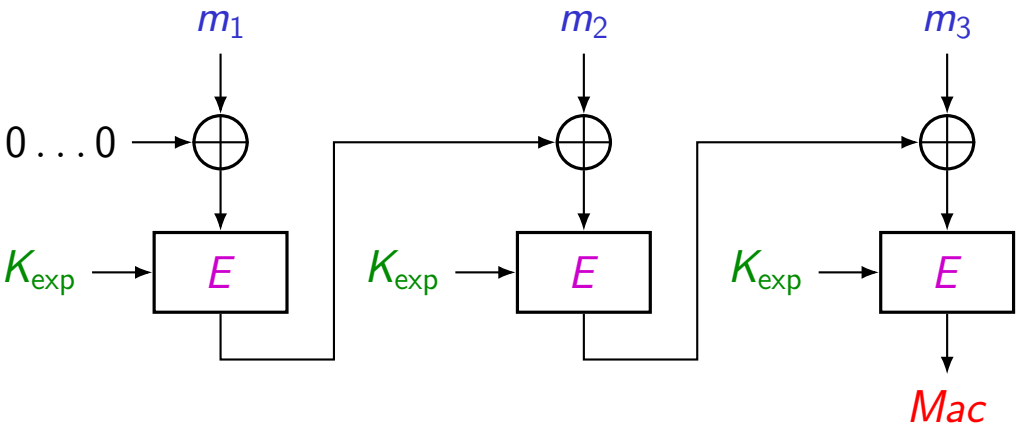
- Mécanismes de chiffrement
- Échange de clés
- Authentification de l'origine des messages

Authentification de l'origine des messages

- Fonctionnement général :
  - l'expéditeur envoie le couple $(M, A_{K_{\text{exp}}}(M))$
  - le destinataire vérifie que $A_{K_{\text{exp}}}(M)$ correspond bien à M et K_{exp}
- Symétrique : code d'authentification de message (MAC)
  - mode authentifiant de chiffrement par bloc : CBC-MAC, CMAC, GCM
  - fonction de hachage cryptographique avec clé : HMAC
  - hachage universel et chiffrement par flot : UMAC
- Asymétrique : signature
  - fondé sur la difficulté de la factorisation : RSA + PKCS #1
  - fondé sur la difficulté du logarithme discret : DSA
  - fondé sur la difficulté du logarithme discret et les courbes elliptiques : ECDSA, EdDSA

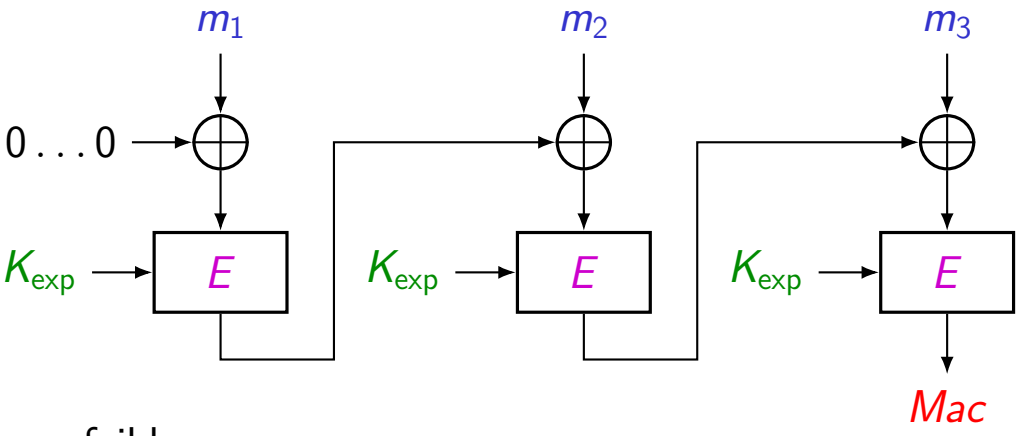
Exemple de mode authentifiant : CBC-MAC

- Chiffrement par bloc E
  - message découpé en blocs : $M = m_1 \| m_2 \| \dots \| m_\ell$
  - attention au *padding*



Exemple de mode authentifiant : CBC-MAC

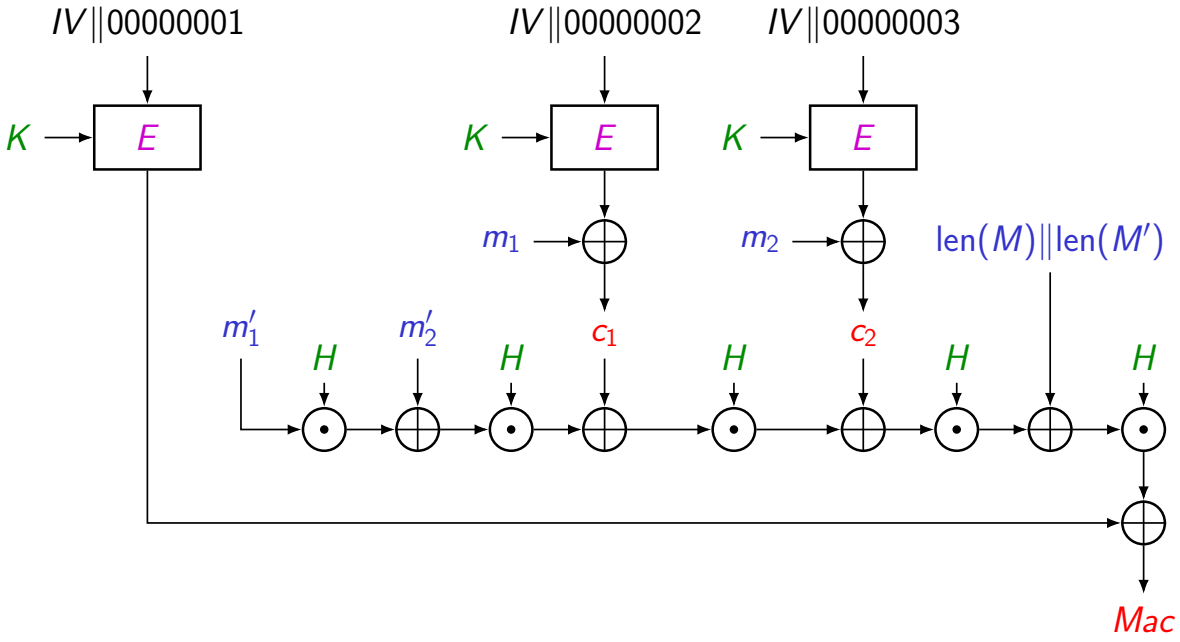
- Chiffrement par bloc E
  - message découpé en blocs : $M = m_1 \| m_2 \| \dots \| m_\ell$
  - attention au *padding*



- Quelques faiblesses
  - importance de fixer l'IV à 0
  - ne pas utiliser la même clé pour chiffrer et pour authentifier !
  - messages de taille variable : $\text{CBC-MAC}^{E}_{K_{\text{exp}}}(\mathit{len}(M) \| M)$ ou $E_{K'_{\text{exp}}}(\text{CBC-MAC}^{E}_{K_{\text{exp}}}(M))$

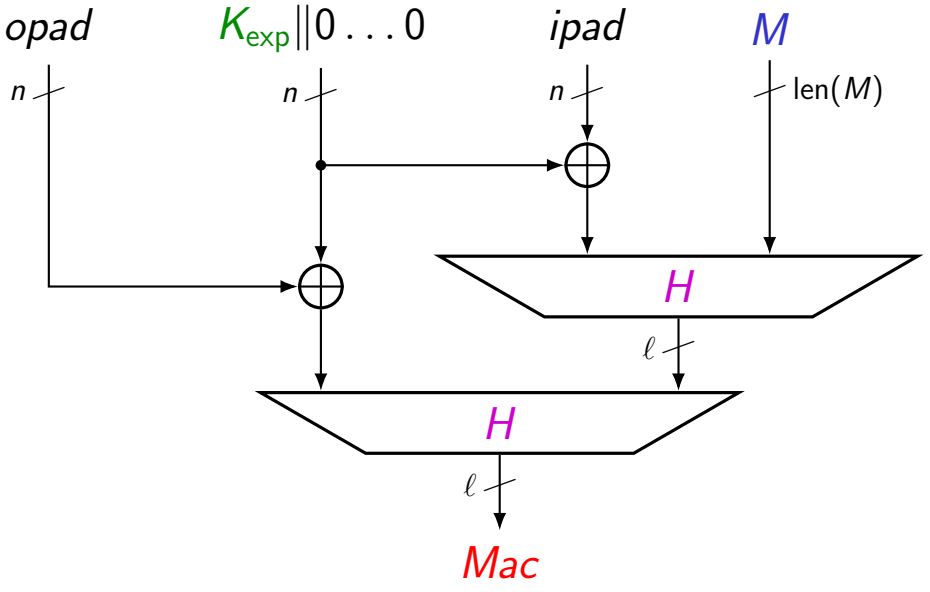
Ex. de mode chiffrant ET authentifiant : GCM

- *Galois/Counter Mode*
  - mode AEAD : *Authenticated Encryption with Associated Data*
  - chiffrement et authentification d'un message $M = m_1 \| \ldots \| m_\ell$
  - authentification de données supplémentaires $M' = m'_1 \| \ldots \| m'_{\ell'}$
  - constante $H = E_K(0)$; opération $\odot$: multiplication dans $\mathrm{GF}(2^n)$



Mode de fonction de hachage avec clé : HMAC

- Fonction de hachage cryptographique H
  - travaillant sur des blocs de n bits ($n = 512$ pour MD5 et SHA-1)
  - produisant des hachés de ℓ bits
  - blocs $ipad = 0x36 \ldots 36$ et $opad = 0x5c \ldots 5c$
  - $\mathrm{HMAC}^{H}_{K_{exp}}(M) = H(\ (K_{exp} \oplus opad) \ \|\ H((K_{exp} \oplus ipad) \| M)\ )$



Signature cryptographique

- Une signature doit (entre autres) :
  - dépendre du document qu'elle approuve
  - émaner uniquement de la personne qui approuve
  - pouvoir être vérifiée par tous
- La cryptographie à clé publique rend possible la signature
  - clé privée connue uniquement par une personne
  - clé publique connue de tous

Signature RSA naïve

- Génération de clés :
  - module $N = p \times q$, avec p et q nombres premiers de $n/2$ bits chacun
  - exposants e et d tels que $m^{de} \bmod N = m$, pour tout $m \in \mathbb{Z}/N\mathbb{Z}$
  - clé publique : (e, N) ; clé privée : (d, N)

Signature RSA naïve

- Génération de clés :
  - module $N = p \times q$, avec p et q nombres premiers de $n/2$ bits chacun
  - exposants e et d tels que $m^{de} \bmod N = m$, pour tout $m \in \mathbb{Z}/N\mathbb{Z}$
  - clé publique : (e, N) ; clé privée : (d, N)
- Signature d'un message $m \in \mathbb{Z}/N\mathbb{Z}$: $s = m^d \bmod N$

Signature RSA naïve

- Génération de clés :
  - module $N = p \times q$, avec p et q nombres premiers de $n/2$ bits chacun
  - exposants e et d tels que $m^{de} \bmod N = m$, pour tout $m \in \mathbb{Z}/N\mathbb{Z}$
  - clé publique : (e, N) ; clé privée : (d, N)
- Signature d'un message $m \in \mathbb{Z}/N\mathbb{Z}$: $s = m^d \bmod N$
- Vérification : $s^e \bmod N \stackrel{?}{=} m$

Signature RSA naïve

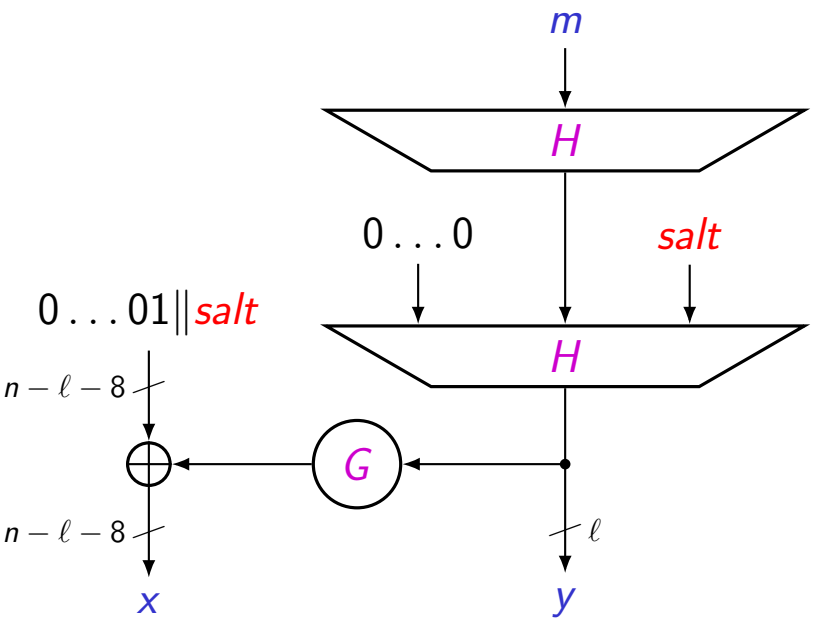
- Génération de clés :
  - module $N = p \times q$, avec p et q nombres premiers de $n/2$ bits chacun
  - exposants e et d tels que $m^{de} \bmod N = m$, pour tout $m \in \mathbb{Z}/N\mathbb{Z}$
  - clé publique : (e, N) ; clé privée : (d, N)
- Signature d'un message $m \in \mathbb{Z}/N\mathbb{Z}$: $s = m^d \bmod N$
- Vérification : $s^e \bmod N \stackrel{?}{=} m$
- Nombreux problèmes : par exemple, propriété multiplicative
  - si s_1 et s_2 signatures valides pour les messages m_1 et m_2, resp.

Signature RSA naïve

- Génération de clés :
  - module $N = p \times q$, avec p et q nombres premiers de $n/2$ bits chacun
  - exposants e et d tels que $m^{de} \bmod N = m$, pour tout $m \in \mathbb{Z}/N\mathbb{Z}$
  - clé publique : (e, N) ; clé privée : (d, N)
- Signature d'un message $m \in \mathbb{Z}/N\mathbb{Z}$: $s = m^d \bmod N$
- Vérification : $s^e \bmod N \stackrel{?}{=} m$
- Nombreux problèmes : par exemple, propriété multiplicative
  - si s_1 et s_2 signatures valides pour les messages m_1 et m_2, resp.
  - ... alors $s_1 s_2 \bmod N$ signature valide pour le message $m_1 m_2 \bmod N$
- Besoin d'encapsulation

Exemple d'encapsulation : RSA-PSS

- Définie dans PKCS #1
  - *salt* : chaîne aléatoire de k bits
  - G et H : fonctions de hachage cryptographiques
  - message final (avant signature) : $m' = x \| y \| \texttt{0xbc}$



Digital Signature Standard (DSS)

- Génération des paramètres et des clés :
  - une fonction de hachage cryptographique H
  - deux premiers p (de n bits) et q (de ℓ bits) tels que q divise $p-1$
  - un élément $g \in (\mathbb{Z}/p\mathbb{Z})^*$ d'ordre q (c'est-à-dire, $g^q \bmod p = 1$)
  - un exposant privé x tiré au hasard dans $(\mathbb{Z}/q\mathbb{Z})^*$
  - l'élément public $y = g^x \bmod p$
  - paramètres : H, p, q et g ; clé publique : y ; clé privée : x

Digital Signature Standard (DSS)

- Génération des paramètres et des clés :
  - une fonction de hachage cryptographique H
  - deux premiers p (de n bits) et q (de ℓ bits) tels que q divise $p-1$
  - un élément $g \in (\mathbb{Z}/p\mathbb{Z})^*$ d'ordre q (c'est-à-dire, $g^q \bmod p = 1$)
  - un exposant privé x tiré au hasard dans $(\mathbb{Z}/q\mathbb{Z})^*$
  - l'élément public $y = g^x \bmod p$
  - paramètres : H, p, q et g ; clé publique : y ; clé privée : x
- Signature (r, s) d'un message m :
  - tirer k au hasard dans $(\mathbb{Z}/q\mathbb{Z})^*$
  - calculer $r = (g^k \bmod p) \bmod q$ et $s = \dfrac{H(m) + xr}{k} \bmod q$

Digital Signature Standard (DSS)

- Génération des paramètres et des clés :
  - une fonction de hachage cryptographique H
  - deux premiers p (de n bits) et q (de ℓ bits) tels que q divise $p-1$
  - un élément $g \in (\mathbb{Z}/p\mathbb{Z})^*$ d'ordre q (c'est-à-dire, $g^q \bmod p = 1$)
  - un exposant privé x tiré au hasard dans $(\mathbb{Z}/q\mathbb{Z})^*$
  - l'élément public $y = g^x \bmod p$
  - paramètres : H, p, q et g ; clé publique : y ; clé privée : x
- Signature (r, s) d'un message m :
  - tirer k au hasard dans $(\mathbb{Z}/q\mathbb{Z})^*$
  - calculer $r = (g^k \bmod p) \bmod q$ et $s = \dfrac{H(m) + xr}{k} \bmod q$
- Vérification :

$$\left(\left(g^{(H(m)/s) \bmod q} \times y^{(r/s) \bmod q}\right) \bmod p\right) \bmod q \stackrel{?}{=} r$$

Digital Signature Standard (DSS)

- Génération des paramètres et des clés :
  - une fonction de hachage cryptographique H
  - deux premiers p (de n bits) et q (de ℓ bits) tels que q divise $p-1$
  - un élément $g \in (\mathbb{Z}/p\mathbb{Z})^*$ d'ordre q (c'est-à-dire, $g^q \bmod p = 1$)
  - un exposant privé x tiré au hasard dans $(\mathbb{Z}/q\mathbb{Z})^*$
  - l'élément public $y = g^x \bmod p$
  - paramètres : H, p, q et g ; clé publique : y ; clé privée : x
- Signature (r, s) d'un message m :
  - tirer k au hasard dans $(\mathbb{Z}/q\mathbb{Z})^*$
  - calculer $r = (g^k \bmod p) \bmod q$ et $s = \dfrac{H(m) + xr}{k} \bmod q$
- Vérification :

$$\left(\left(g^{(H(m)/s) \bmod q} \times y^{(r/s) \bmod q}\right) \bmod p\right) \bmod q \stackrel{?}{=} r$$

- Sécurité en 2^{128} opérations : p de $n = 3072$ bits et q de $\ell = 256$ bits

Quatrième partie

Exemple de protocole cryptographique : SSL/TLS

(Slides de Jérémie Detrey)

Le protocole SSL/TLS

- Protocole de sécurisation des échanges sur Internet
- Mode client-serveur, au dessus de TCP (Transmission Control Protocol)
- Permet de garantir
  - l'authentification du serveur
  - la confidentialité des données échangées
  - l'intégrité et l'authentification de l'origine des données échangées
  - l'authentification du client (optionnel)
- Permet d'encapsuler de manière transparente des protocoles de la couche application
  - HTTP (80) $\rightarrow$ HTTPS (443)
  - IMAP (143) $\rightarrow$ IMAPS (993)
  - POP3 (110) $\rightarrow$ POP3S (995)
  - STARTTLS (pour IMAP, POP3, SMTP, FTP, etc.) sur le même port

Un peu d'histoire

- Au début, SSL (*Secure Sockets Layer*), développé par Netscape
  - SSL 1.0 (1994) : protocole théorique, jamais utilisé
  - SSL 2.0 (1995-2011) : première version utilisée
  - SSL 3.0 (1996-. . ., RFC 6101) : dernière version, sur laquelle TLS sera basé

Un peu d'histoire

- Au début, SSL (*Secure Sockets Layer*), développé par Netscape
  - SSL 1.0 (1994) : protocole théorique, jamais utilisé
  - SSL 2.0 (1995-2011) : première version utilisée
  - SSL 3.0 (1996-..., RFC 6101) : dernière version, sur laquelle TLS sera basé
- Puis TLS (*Transport Layer Security*), développé par l'IETF (*Internet Engineering Task Force*)
  - TLS 1.0 (1999-..., RFC 2246) : successeur de SSL, diverses amélioration jusqu'en 2002
  - TLS 1.1 (2006-..., RFC 4346)
  - TLS 1.2 (2008-..., RFC 5246)
  - TLS 1.3 (2018-..., RFC 8446) Quelques différences de 1.2 à 1.3

Un peu d'histoire

- Au début, SSL (*Secure Sockets Layer*), développé par Netscape
  - SSL 1.0 (1994) : protocole théorique, jamais utilisé
  - SSL 2.0 (1995-2011) : première version utilisée
  - SSL 3.0 (1996-..., RFC 6101) : dernière version, sur laquelle TLS sera basé
- Puis TLS (*Transport Layer Security*), développé par l'IETF (*Internet Engineering Task Force*)
  - TLS 1.0 (1999-..., RFC 2246) : successeur de SSL, diverses amélioration jusqu'en 2002
  - TLS 1.1 (2006-..., RFC 4346)
  - TLS 1.2 (2008-..., RFC 5246)
  - TLS 1.3 (2018-..., RFC 8446) Quelques différences de 1.2 à 1.3
- Compatibilité des serveurs HTTPS (source : SSL Pulse, 12/2023)

SSL 2.0	SSL 3.0	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3
0,2 %	1,6 %	29,1 %	31,4 %	99,9 %	66,9 %

Mécanismes cryptographiques dans TLS

- TLS offre le choix entre plusieurs mécanismes cryptographiques pour être capable de s'adapter aux contraintes de chaque application et de chaque système

Mécanismes cryptographiques dans TLS

- TLS offre le choix entre plusieurs mécanismes cryptographiques pour être capable de s'adapter aux contraintes de chaque application et de chaque système
- Authentification du serveur (et du client, optionnellement) :
  - clé publique : RSA, DSS, ECDSA
  - clé secrète ou mot de passe partagé : PSK (*Pre-Shared Key*), SRP (*Secure Remote Password*)
  - pas d'authentification : ANON

Mécanismes cryptographiques dans TLS

- TLS offre le choix entre plusieurs mécanismes cryptographiques pour être capable de s'adapter aux contraintes de chaque application et de chaque système
- Authentification du serveur (et du client, optionnellement) :
  - clé publique : RSA, DSS, ECDSA
  - clé secrète ou mot de passe partagé : PSK (*Pre-Shared Key*), SRP (*Secure Remote Password*)
  - pas d'authentification : ANON
- Échange de clés :
  - clé publique (toujours la même clé privée côté serveur) : RSA
  - Diffie-Hellman statique (même problème) : DH, ECDH
  - clé secrète ou mot de passe partagé : PSK, SRP
  - Diffie-Hellman éphémère (clés privées propres à chaque connexion ; garantit la *forward secrecy*) : DHE, ECDHE

Mécanismes cryptographiques dans TLS

- Chiffrement :
  - chiffrement par bloc : AES-CBC, 3DES-CBC, DES-CBC, etc.
  - chiffrement par bloc avec mode authentifiant : AES-CCM, AES-GCM, etc.
  - chiffrement par flot : RC4
  - pas de chiffrement : NULL

Mécanismes cryptographiques dans TLS

- Chiffrement :
  - chiffrement par bloc : AES-CBC, 3DES-CBC, DES-CBC, etc.
  - chiffrement par bloc avec mode authentifiant : AES-CCM, AES-GCM, etc.
  - chiffrement par flot : RC4
  - pas de chiffrement : NULL
- Intégrité et authentification de l'origine des messages :
  - HMAC : HMAC-MD5, HMAC-SHA1, HMAC-SHA256, etc.
  - mode authentifiant du chiffrement par bloc : AEAD

Mécanismes cryptographiques dans TLS

- Chiffrement :
  - chiffrement par bloc : AES-CBC, 3DES-CBC, DES-CBC, etc.
  - chiffrement par bloc avec mode authentifiant : AES-CCM, AES-GCM, etc.
  - chiffrement par flot : RC4
  - pas de chiffrement : NULL
- Intégrité et authentification de l'origine des messages :
  - HMAC : HMAC-MD5, HMAC-SHA1, HMAC-SHA256, etc.
  - mode authentifiant du chiffrement par bloc : AEAD
- Une combinaison de ces mécanismes est appelée *cipher suite*
  - par exemple : TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

Mécanismes cryptographiques dans TLS

- Chiffrement :
  - chiffrement par bloc : AES-CBC, 3DES-CBC, DES-CBC, etc.
  - chiffrement par bloc avec mode authentifiant : AES-CCM, AES-GCM, etc.
  - chiffrement par flot : RC4
  - pas de chiffrement : NULL
- Intégrité et authentification de l'origine des messages :
  - HMAC : HMAC-MD5, HMAC-SHA1, HMAC-SHA256, etc.
  - mode authentifiant du chiffrement par bloc : AEAD
- Une combinaison de ces mécanismes est appelée *cipher suite*
  - par exemple : TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- En bonus, TLS peut aussi supporter un mode de compression des données : NULL ou DEFLATE

Établissement d'une connexion SSL/TLS

- Contexte : un client souhaite établir une connexion SSL/TLS avec un serveur
- Objectifs :
  - se mettre d'accord sur une *cipher suite* commune
  - authentifier le serveur
  - échanger des clés secrètes pour le chiffrement et l'authentification des messages
- Protocole de *handshake* en 4 étapes

Handshake SSL/TLS simple

1. Client $\rightarrow$ Serveur
   - `ClientHello` : plus haute version du protocole supportée, liste des *cipher suites* et modes de compression supportés

Handshake SSL/TLS simple

1. Client → Serveur
   - `ClientHello` : plus haute version du protocole supportée, liste des *cipher suites* et modes de compression supportés
2. Serveur → Client
   - `ServerHello` : version du protocole, *cipher suite* et mode de compression choisis
   - `Certificate` (opt.) : la clé publique du serveur dans un certificat X.509 permettant d'en vérifier l'authenticité
   - `ServerKeyExchange` (opt.) : une clé publique Diffie-Hellman pour l'échange de clés
   - `ServerHelloDone`

Handshake SSL/TLS simple

3. Client → Serveur
   - `ClientKeyExchange` : soit un secret (*PreMasterKey*) chiffré avec la clé publique du serveur, soit une clé publique Diffie-Hellman pour l'échange de clés
   - `ChangeCipherSpec` (marque la fin du *handshake* côté client)
   - `Finished` : message chiffré et authentifié contenant un MAC des messages précédents du *handshake*

Handshake SSL/TLS simple

3. Client → Serveur
   - `ClientKeyExchange` : soit un secret (*PreMasterKey*) chiffré avec la clé publique du serveur, soit une clé publique Diffie-Hellman pour l'échange de clés
   - `ChangeCipherSpec` (marque la fin du *handshake* côté client)
   - `Finished` : message chiffré et authentifié contenant un MAC des messages précédents du *handshake*

4. Serveur → Client (si le `Finished` du client est valide)
   - `ChangeCipherSpec` (marque la fin du *handshake* côté serveur)
   - `Finished` : message chiffré et authentifié contenant un MAC des messages précédents du *handshake*

Handshake SSL/TLS simple

3. Client → Serveur
   - `ClientKeyExchange` : soit un secret (*PreMasterKey*) chiffré avec la clé publique du serveur, soit une clé publique Diffie-Hellman pour l'échange de clés
   - `ChangeCipherSpec` (marque la fin du *handshake* côté client)
   - `Finished` : message chiffré et authentifié contenant un MAC des messages précédents du *handshake*
4. Serveur → Client (si le `Finished` du client est valide)
   - `ChangeCipherSpec` (marque la fin du *handshake* côté serveur)
   - `Finished` : message chiffré et authentifié contenant un MAC des messages précédents du *handshake*
5. Si le `Finished` du serveur est aussi valide, la connexion est établie

Authentification du serveur

- Dans le *handshake* précédent, le serveur envoie lui-même sa clé publique afin d'authentifier ses messages
  $\rightarrow$ N'y a-t-il pas comme un problème ?
- Comment vérifier l'authenticité de la clé publique du serveur ?

Authentification du serveur

- Dans le *handshake* précédent, le serveur envoie lui-même sa clé publique afin d'authentifier ses messages
  $\rightarrow$ N'y a-t-il pas comme un problème ?
- Comment vérifier l'authenticité de la clé publique du serveur ?
  - Clé publique signée par le serveur ?

Authentification du serveur

- Dans le *handshake* précédent, le serveur envoie lui-même sa clé publique afin d'authentifier ses messages
  $\rightarrow$ N'y a-t-il pas comme un problème ?
- Comment vérifier l'authenticité de la clé publique du serveur ?
  - Clé publique signée par le serveur ?
    $\rightarrow$ La vérification nécessite de faire confiance à la clé publique...

Authentification du serveur

- Dans le *handshake* précédent, le serveur envoie lui-même sa clé publique afin d'authentifier ses messages
  $\rightarrow$ N'y a-t-il pas comme un problème ?
- Comment vérifier l'authenticité de la clé publique du serveur ?
  - Clé publique signée par le serveur ?
    $\rightarrow$ La vérification nécessite de faire confiance à la clé publique...
  - Clé publique signée par une tierce partie (appelée autorité de certification, ou CA) ?

Authentification du serveur

- Dans le *handshake* précédent, le serveur envoie lui-même sa clé publique afin d'authentifier ses messages
  $\rightarrow$ N'y a-t-il pas comme un problème ?
- Comment vérifier l'authenticité de la clé publique du serveur ?
  - Clé publique signée par le serveur ?
    $\rightarrow$ La vérification nécessite de faire confiance à la clé publique...
  - Clé publique signée par une tierce partie (appelée autorité de certification, ou CA) ?
    $\rightarrow$ OK, mais comment vérifier cette signature ?

Authentification du serveur

- Dans le *handshake* précédent, le serveur envoie lui-même sa clé publique afin d'authentifier ses messages
  $\rightarrow$ N'y a-t-il pas comme un problème ?
- Comment vérifier l'authenticité de la clé publique du serveur ?
  - Clé publique signée par le serveur ?
    $\rightarrow$ La vérification nécessite de faire confiance à la clé publique...
  - Clé publique signée par une tierce partie (appelée autorité de certification, ou CA) ?
    $\rightarrow$ OK, mais comment vérifier cette signature ?
  - Clé publique signée par un CA, et clé publique du CA ?

Authentification du serveur

- Dans le *handshake* précédent, le serveur envoie lui-même sa clé publique afin d'authentifier ses messages
  $\rightarrow$ N'y a-t-il pas comme un problème ?
- Comment vérifier l'authenticité de la clé publique du serveur ?
  - Clé publique signée par le serveur ?
    $\rightarrow$ La vérification nécessite de faire confiance à la clé publique...
  - Clé publique signée par une tierce partie (appelée autorité de certification, ou CA) ?
    $\rightarrow$ OK, mais comment vérifier cette signature ?
  - Clé publique signée par un CA, et clé publique du CA ?
    $\rightarrow$ Comment vérifier l'authenticité de la clé publique du CA ? On a juste déplacé le problème...

Authentification du serveur

- Dans le *handshake* précédent, le serveur envoie lui-même sa clé publique afin d'authentifier ses messages
  → N'y a-t-il pas comme un problème ?
- Comment vérifier l'authenticité de la clé publique du serveur ?
  - Clé publique signée par le serveur ?
    → La vérification nécessite de faire confiance à la clé publique...
  - Clé publique signée par une tierce partie (appelée autorité de certification, ou CA) ?
    → OK, mais comment vérifier cette signature ?
  - Clé publique signée par un CA, et clé publique du CA ?
    → Comment vérifier l'authenticité de la clé publique du CA ? On a juste déplacé le problème...
- Il s'agit d'un problème difficile, qui nécessite la mise en place d'une infrastructure à clés publiques (ou PKI)

Infrastructure à clés publiques

- Permet de répondre à la question :

  Comment faire confiance à une clé publique ?

- Certificat : signature de la clé publique par un tiers de confiance

Infrastructure à clés publiques

- Permet de répondre à la question :

  Comment faire confiance à une clé publique ?

- Certificat : signature de la clé publique par un tiers de confiance
- Infrastructures possibles :
  - hiérarchique : autorités de certification (SSL/TLS et X.509, EMV)
  - décentralisée : réseau de confiance (PGP, GnuPG)

Autorités de certification

- Différents niveaux :
  - CA racines : peuvent certifier les clés publiques d'autres CA
  - CA intermédiaires : en général, ne peuvent pas certifier d'autres CA
    $\rightarrow$ certifient les clés publiques des serveurs

Autorités de certification

- Différents niveaux :
  - CA racines : peuvent certifier les clés publiques d'autres CA
  - CA intermédiaires : en général, ne peuvent pas certifier d'autres CA
    $\rightarrow$ certifient les clés publiques des serveurs
- Chaîne de certification :

$$\text{CA racine} \xrightarrow{\text{signe}} \text{CA 1} \xrightarrow{\text{signe}} \text{CA 2} \xrightarrow{\text{signe}} \text{Serveur}$$

Autorités de certification

- Différents niveaux :
  - CA racines : peuvent certifier les clés publiques d'autres CA
  - CA intermédiaires : en général, ne peuvent pas certifier d'autres CA
    $\rightarrow$ certifient les clés publiques des serveurs
- Chaîne de certification :

$$\text{CA racine} \xrightarrow{\text{signe}} \text{CA 1} \xrightarrow{\text{signe}} \text{CA 2} \xrightarrow{\text{signe}} \text{Serveur}$$

- Authentification : le serveur envoie trois certificats
  - sa propre clé publique, signée par CA 2
  - la clé publique de CA 2, signée par CA 1
  - la clé publique de CA 1, signée par CA racine

Autorités de certification

- Différents niveaux :
  - CA racines : peuvent certifier les clés publiques d'autres CA
  - CA intermédiaires : en général, ne peuvent pas certifier d'autres CA
    $\rightarrow$ certifient les clés publiques des serveurs
- Chaîne de certification :

$$\text{CA racine} \xrightarrow{\text{signe}} \text{CA 1} \xrightarrow{\text{signe}} \text{CA 2} \xrightarrow{\text{signe}} \text{Serveur}$$

- Authentification : le serveur envoie trois certificats
  - sa propre clé publique, signée par CA 2
  - la clé publique de CA 2, signée par CA 1
  - la clé publique de CA 1, signée par CA racine
- Vérification : si le client connaît (et fait confiance à) la clé publique de CA racine, il peut vérifier la validité de tous les certificats

Autorités de certification

- Toujours les mêmes problèmes :
  - comment obtenir la clé publique des CA racines ?
  - quelle confiance leur accorder ?

Autorités de certification

- Toujours les mêmes problèmes :
  - comment obtenir la clé publique des CA racines ?
  - quelle confiance leur accorder ?
- Liste de CA racines de confiance maintenue par les navigateurs
  - actuellement, 80+ CA racines intégrés dans Firefox
  - mais comment avoir confiance en le navigateur que l'on télécharge ?!
    $\rightarrow$ contrôle d'intégrité et d'authenticité de l'archive téléchargée ?
    $\rightarrow$ problème de la poule et de l'œuf...

Autorités de certification

- Toujours les mêmes problèmes :
  - comment obtenir la clé publique des CA racines ?
  - quelle confiance leur accorder ?
- Liste de CA racines de confiance maintenue par les navigateurs
  - actuellement, 80+ CA racines intégrés dans Firefox
  - mais comment avoir confiance en le navigateur que l'on télécharge ?!
    $\rightarrow$ contrôle d'intégrité et d'authenticité de l'archive téléchargée ?
    $\rightarrow$ problème de la poule et de l'œuf...
- Attention à la sécurité des CA (racines et intermédiaires) !
  - si la clé privée d'un CA est compromise : émission de faux certificats
    p.ex. DigiNotar en 2011 : 500 faux certificats, dont `*.google.com`
  - audits de sécurité réguliers
  - mécanisme de révocation de certificats compromis : CRL (*Certificate Revocation List*) ou OCSP (*Online Certificate Status Protocol*)

Failles dans SSL/TLS

- Des vulnérabilités sont toujours découvertes dans SSL/TLS
  - exemple : DROWN, rendue publique le 1er mars 2016

Failles dans SSL/TLS

- Des vulnérabilités sont toujours découvertes dans SSL/TLS
  - exemple : DROWN, rendue publique le 1er mars 2016
- Des correctifs existent, mais les serveurs doivent être mis à jour

Failles dans SSL/TLS

- Des vulnérabilités sont toujours découvertes dans SSL/TLS
  - exemple : DROWN, rendue publique le 1er mars 2016
- Des correctifs existent, mais les serveurs doivent être mis à jour
  - renégociation non sécurisée (2009, MITM) : 1,8 + 0,6 % vulnérables

Failles dans SSL/TLS

- Des vulnérabilités sont toujours découvertes dans SSL/TLS
  - exemple : DROWN, rendue publique le 1er mars 2016
- Des correctifs existent, mais les serveurs doivent être mis à jour
  - renégociation non sécurisée (2009, MITM) : 1,8 + 0,6 % vulnérables
  - BEAST (*Browser Exploit Against SSL/TLS*, 2011, violation de la contrainte d'origine des cookies) : 91,5 %

Failles dans SSL/TLS

- Des vulnérabilités sont toujours découvertes dans SSL/TLS
  - exemple : DROWN, rendue publique le 1er mars 2016
- Des correctifs existent, mais les serveurs doivent être mis à jour
  - renégociation non sécurisée (2009, MITM) : 1,8 + 0,6 % vulnérables
  - BEAST (*Browser Exploit Against SSL/TLS*, 2011, violation de la contrainte d'origine des cookies) : 91,5 %
  - CRIME (*Compression Ratio Info-leak Made Easy*, 2012) : 3,2 %

Failles dans SSL/TLS

- Des vulnérabilités sont toujours découvertes dans SSL/TLS
  - exemple : DROWN, rendue publique le 1er mars 2016
- Des correctifs existent, mais les serveurs doivent être mis à jour
  - renégociation non sécurisée (2009, MITM) : 1,8 + 0,6 % vulnérables
  - BEAST (*Browser Exploit Against SSL/TLS*, 2011, violation de la contrainte d'origine des cookies) : 91,5 %
  - CRIME (*Compression Ratio Info-leak Made Easy*, 2012) : 3,2 %
  - dégradation du protocole (forcer l'utilisation de SSL 3.0) : 28,0 %

Failles dans SSL/TLS

- Des vulnérabilités sont toujours découvertes dans SSL/TLS
  - exemple : DROWN, rendue publique le 1er mars 2016
- Des correctifs existent, mais les serveurs doivent être mis à jour
  - renégociation non sécurisée (2009, MITM) : 1,8 + 0,6 % vulnérables
  - BEAST (*Browser Exploit Against SSL/TLS*, 2011, violation de la contrainte d'origine des cookies) : 91,5 %
  - CRIME (*Compression Ratio Info-leak Made Easy*, 2012) : 3,2 %
  - dégradation du protocole (forcer l'utilisation de SSL 3.0) : 28,0 %
  - attaques sur RC4 (2013) : 8,5 + 34,8 %

Failles dans SSL/TLS

- Des vulnérabilités sont toujours découvertes dans SSL/TLS
  - exemple : DROWN, rendue publique le 1er mars 2016
- Des correctifs existent, mais les serveurs doivent être mis à jour
  - renégociation non sécurisée (2009, MITM) : 1,8 + 0,6 % vulnérables
  - BEAST (*Browser Exploit Against SSL/TLS*, 2011, violation de la contrainte d'origine des cookies) : 91,5 %
  - CRIME (*Compression Ratio Info-leak Made Easy*, 2012) : 3,2 %
  - dégradation du protocole (forcer l'utilisation de SSL 3.0) : 28,0 %
  - attaques sur RC4 (2013) : 8,5 + 34,8 %
  - POODLE sur TLS (2014) : 3,0 %

Failles dans SSL/TLS

- Des vulnérabilités sont toujours découvertes dans SSL/TLS
  - exemple : DROWN, rendue publique le 1er mars 2016
- Des correctifs existent, mais les serveurs doivent être mis à jour
  - renégociation non sécurisée (2009, MITM) : 1,8 + 0,6 % vulnérables
  - BEAST (*Browser Exploit Against SSL/TLS*, 2011, violation de la contrainte d'origine des cookies) : 91,5 %
  - CRIME (*Compression Ratio Info-leak Made Easy*, 2012) : 3,2 %
  - dégradation du protocole (forcer l'utilisation de SSL 3.0) : 28,0 %
  - attaques sur RC4 (2013) : 8,5 + 34,8 %
  - POODLE sur TLS (2014) : 3,0 %
  - Heartbleed (2014, accès mémoire arbitraire dans OpenSSL) : 0,3 %

Failles dans SSL/TLS

- Des vulnérabilités sont toujours découvertes dans SSL/TLS
  - exemple : DROWN, rendue publique le 1er mars 2016
- Des correctifs existent, mais les serveurs doivent être mis à jour
  - renégociation non sécurisée (2009, MITM) : 1,8 + 0,6 % vulnérables
  - BEAST (*Browser Exploit Against SSL/TLS*, 2011, violation de la contrainte d'origine des cookies) : 91,5 %
  - CRIME (*Compression Ratio Info-leak Made Easy*, 2012) : 3,2 %
  - dégradation du protocole (forcer l'utilisation de SSL 3.0) : 28,0 %
  - attaques sur RC4 (2013) : 8,5 + 34,8 %
  - POODLE sur TLS (2014) : 3,0 %
  - Heartbleed (2014, accès mémoire arbitraire dans OpenSSL) : 0,3 %
  - pas de *forward secrecy* : 21,2 + 29,4 %

Failles dans SSL/TLS

- Des vulnérabilités sont toujours découvertes dans SSL/TLS
  - exemple : DROWN, rendue publique le 1er mars 2016
- Des correctifs existent, mais les serveurs doivent être mis à jour
  - renégociation non sécurisée (2009, MITM) : 1,8 + 0,6 % vulnérables
  - BEAST (*Browser Exploit Against SSL/TLS*, 2011, violation de la contrainte d'origine des cookies) : 91,5 %
  - CRIME (*Compression Ratio Info-leak Made Easy*, 2012) : 3,2 %
  - dégradation du protocole (forcer l'utilisation de SSL 3.0) : 28,0 %
  - attaques sur RC4 (2013) : 8,5 + 34,8 %
  - POODLE sur TLS (2014) : 3,0 %
  - Heartbleed (2014, accès mémoire arbitraire dans OpenSSL) : 0,3 %
  - pas de *forward secrecy* : 21,2 + 29,4 %
  - clés trop petites : 0,1 % (clé publique), 6,9 + 25,2 % (éch. de clés)

Failles dans SSL/TLS

- Des vulnérabilités sont toujours découvertes dans SSL/TLS
  - exemple : DROWN, rendue publique le 1er mars 2016
- Des correctifs existent, mais les serveurs doivent être mis à jour
  - renégociation non sécurisée (2009, MITM) : 1,8 + 0,6 % vulnérables
  - BEAST (*Browser Exploit Against SSL/TLS*, 2011, violation de la contrainte d'origine des cookies) : 91,5 %
  - CRIME (*Compression Ratio Info-leak Made Easy*, 2012) : 3,2 %
  - dégradation du protocole (forcer l'utilisation de SSL 3.0) : 28,0 %
  - attaques sur RC4 (2013) : 8,5 + 34,8 %
  - POODLE sur TLS (2014) : 3,0 %
  - Heartbleed (2014, accès mémoire arbitraire dans OpenSSL) : 0,3 %
  - pas de *forward secrecy* : 21,2 + 29,4 %
  - clés trop petites : 0,1 % (clé publique), 6,9 + 25,2 % (éch. de clés)
  - etc.