Réseaux 8
Sécurité des réseaux

Juliusz Chroboczek

16 novembre 2020

Sécurité et fascisme

Sécurité :

- modèle d'attaque : ce que l'attaquant doit faire ;
- politique de sécurité : ce qu'on veut permettre ou interdire.

On en déduit un protocole de sécurité dont on fait une implémentation, et on montre leur adéquation au modèle d'attaque et à la politique de sécurité.

Sans modèle d'attaque et politique de sécurité, ce n'est pas de la sécurité, mais

- de la bidouille ; ou, pire,
- du fascisme.

Propriétés de sécurité

Essentiellement trois :

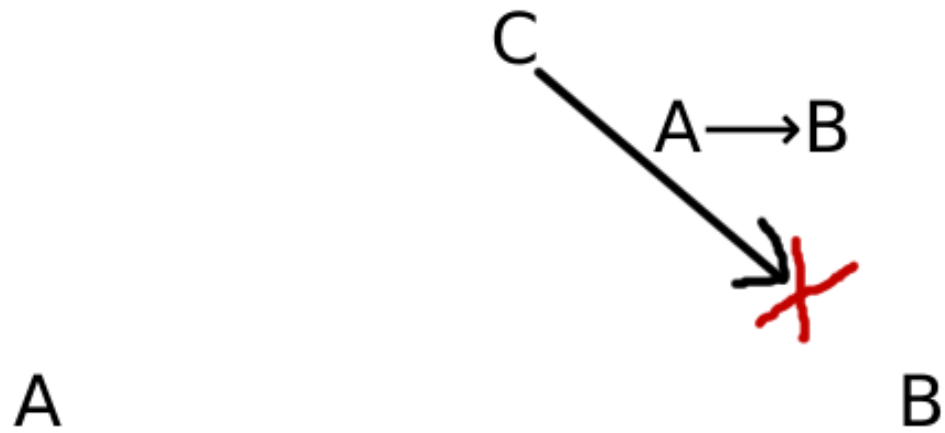
- authenticité : ce message a-t-il été émis par qui je crois ?
- intégrité : ce message est-il identique au message émis ?
- confidentialité : ce message a-t-il pu être vu par quelqu'un de non authorisé ?

Moins classique, et impossible en général :

- résistance au déni de service (DoS) : peut-on garantir que le service continue à être fourni ?

Authenticité

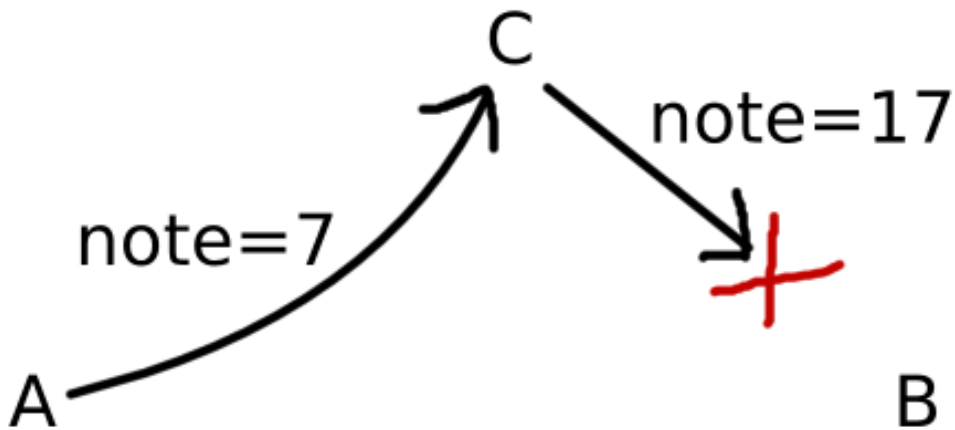
Authenticité : un message reçu par *B* disant provenir de *A* provient effectivement de *A*.



Exemple d'attaque : *C* injecte des paquets UDP non-authentifiés.

Intégrité

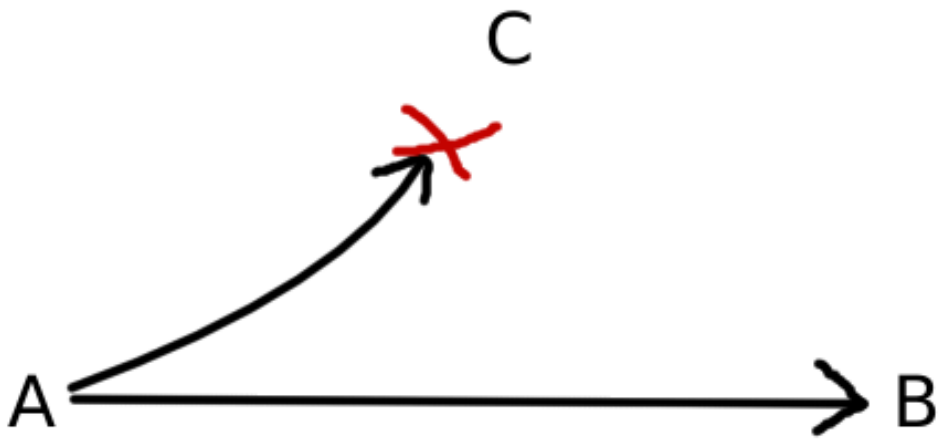
Intégrité : un message reçu par *B* n'a pas été modifé en transit.



Exemple d'attaque : *C* injecte des paquets dans une connection TCP authentifiée.

Confidentialité

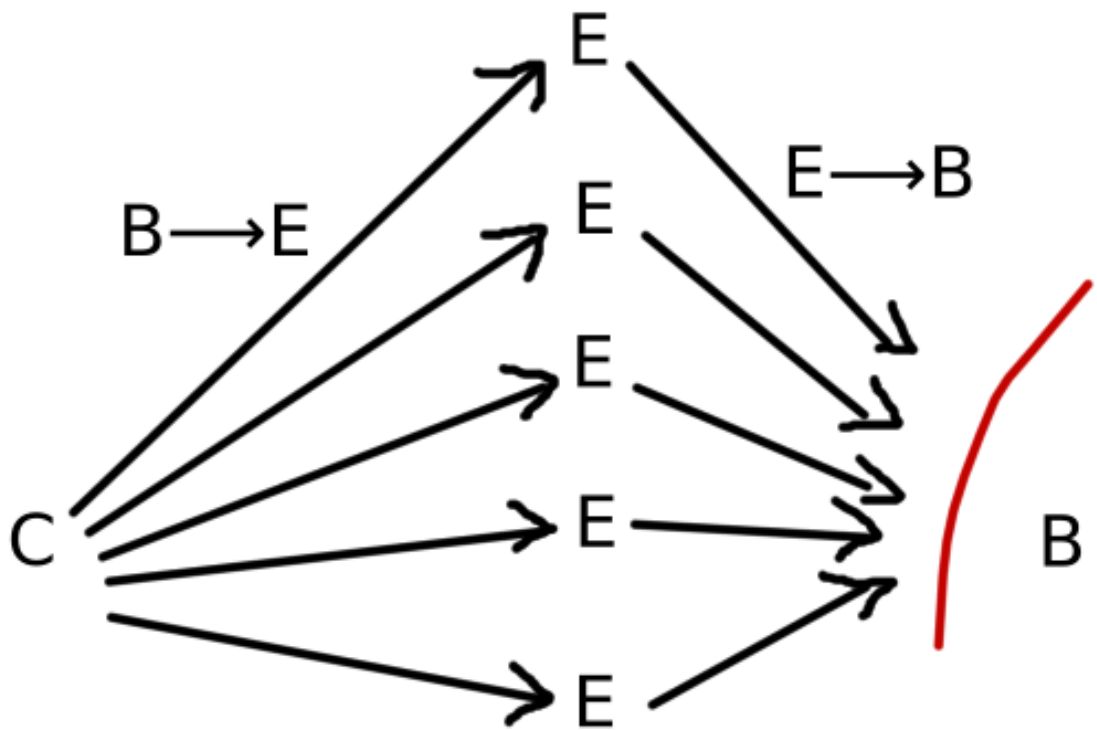
Confidentialité : un message envoyé par *A* à *B* n'a pas pu être vu par *C*.



Exemple d'attaque : *C* utilise `tcpdump`.

Résistance au déni de service

Un déni de service est une attaque qui vise à arrêter un service.



À quelle couche ?

À quelle couche doit se faire la sécurité ?
Pas de réponse claire :

- couche physique : prises Ethernet sous clé ;
- couche lien : WPA2, WPA3 ;
- couche réseau : VPN ;
- couche transport : SSL/TLS, DTLS ;
- couche application.

Faites votre choix :

- pas encore au point, on ne sait pas à quelle couche implémenter la sécurité ;
- la sécurité est forcément en profondeur, on ne peut pas la limiter à une seule couche.

(Je ne sais honnêtement pas quelle est la réponse.)

Choix de la couche : conséquences

Les propriétés de sécurité dépendent de la couche :

- couche lien ou physique : protection locale au lien ;
- couche réseau : protection entre deux hôtes (mais pas entre utilisateurs sur le même hôte) ;
- couche transport : protection le long de la connexion de couche transport (mais pas dans l'application) ;
- couche application : protection de bout en bout possible (mais pas automatique).

On combine des mécanismes aux différentes couches :

- TLS (4) + WPA2 ;
- SSH (7) + VPN + WPA2.

Primitives et constructions

Primitives :

- générateur aléatoire cryptographique ;
- fonction de hachage cryptographique ;
- chiffrage symétrique ;
- chiffrage asymétrique ;
- problèmes infaisables.

Constructions :

- MAC (HMAC, CMAC) ;
- échange de clés ;
- chiffrage asymétrique efficace ; etc.

On ne les choisit pas nous-mêmes :

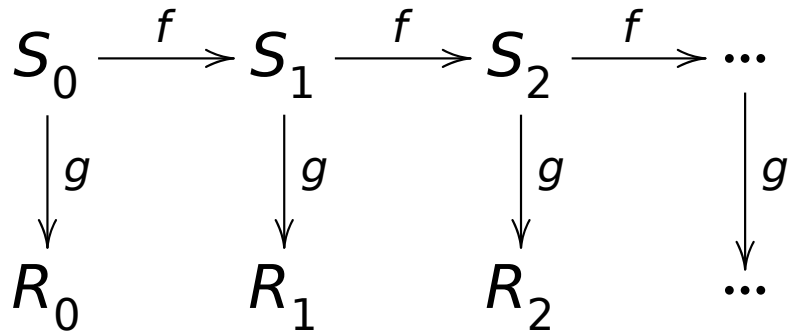
- publiés (pas de NDA !) ;
- vérifiés par des experts ;
- recommandés par NIST, ANSSI, etc.
  (méfiance cependant : double-EC PRNG).

Primitive : RNG cryptographique

Un générateur de nombre pseudo-aléatoires (PRNG) est un algorithme qui produit une suite de nombres ayant une bonne distribution statistique.

Exemple (construit par moi, jamais testé) :

$$S_0 \quad \text{« seed » aléatoire}$$
$$S_{n+1} = (3S_n + 13) \bmod 2^{16}$$
$$R_n = (S_n \bmod 6) + 1$$



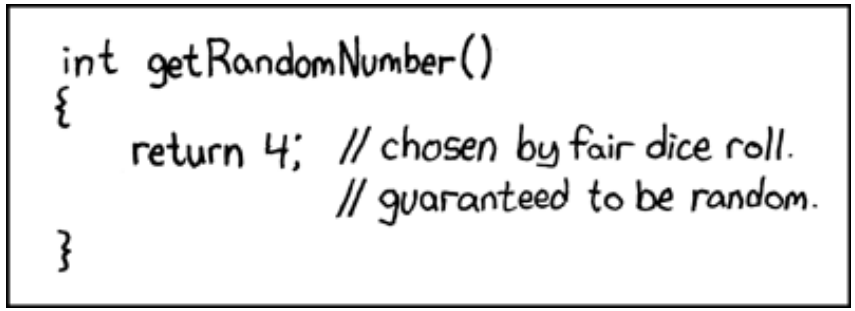
Un PRNG est cryptographique s'il est infaisable de prédire la prochaine valeur même si on connaît :

- l'algorithme (et les constantes) ;
- n valeurs R_i précédentes.

PRNG ci-dessus clairement pas cryptographique.

Application : génération de clés

Les mots de passe sont faciles à deviner.



On utilise des clés :

- générées par un générateur de nombres aléatoires vrais (TRNG) ;
- générées par un PRNG cryptographique initialisé par un TRNG ;
- générées en combinant un mot de passe avec du sel et en itérant une fonction de hachage cryptographique.

Primitive : hachage cryptographique

Une fonction de hachage est une fonction statistiquement injective.

$$\text{si } x \neq y \text{ alors probablement } h(x) \neq y$$

On appelle collision une paire (x, y) t.q. $h(x) = h(y)$.

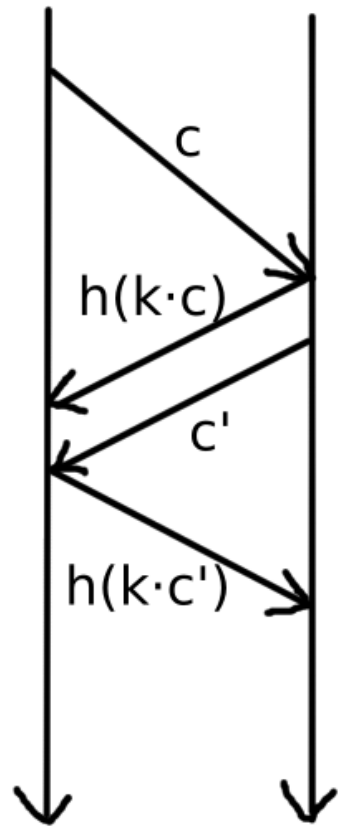
Une fonction de hachage cryptographique est une fonction de hachage pour laquelle il est infaisable de trouver une collision.

En ce moment : SHA-256 (SHA-1 est déprécié).

Application : authentification mutuelle

A et B ont une clé partagée. Comment vérifier que A parle à B et vice versa sans que C puisse capturer la clé ?

A : challenge c aléatoire.
B : $h(k \cdot c)$.
B : challenge c'.
A : $h(k \cdot c')$.



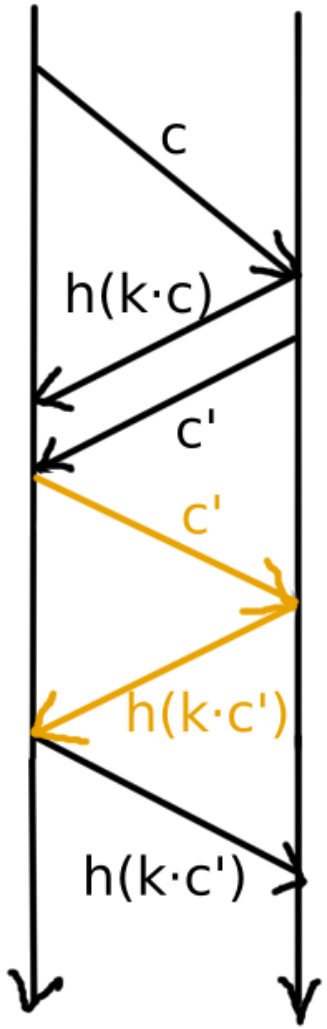
Authentification mutuelle : attaque

Après avoir reçu c' de B, A ouvre une nouvelle session.

A envoie c', B répond par le *hash* dont A a besoin.

Vérifier les protocoles :

- spécialistes de la sécurité ;
- preuves.



Parades

Quelles parades à cette attaque ?

- *channel binding* : on calcule
  - $h(A \cdot B \cdot k \cdot c)$ dans un sens ;
  - $h(B \cdot A \cdot k \cdot c)$ dans l'autre.
- clés différentes dans les deux sens.

Construction : HMAC

On aurait envie de signer un message en calculant

$$\text{sign}_k(m) = h(k \cdot m).$$

La plupart des fonctions de hachage sont vulnérables aux attaques par extension :

- si on connaît $h(m)$, alors on peut calculer $h(m \cdot m')$.

Attaque :

- l'attaquent capture $m, \text{sign}_k(m)$
- émet $m \cdot m', \text{sign}_k(m \cdot m')$.

HMAC :

$$\text{HMAC}_k(m) = h(k \oplus \text{opad}, h(k \oplus \text{ipad}, m))$$
$$\text{ipad} = 36_{16} \cdot 36_{16} \cdot 36_{16} \cdots$$
$$\text{opad} = 5C_{16} \cdot 5C_{16} \cdot 5C_{16} \cdots$$

Primitive : chiffrage symétrique

Un chiffrage, paramétré par une clé k, est une paire de fonctions

$$c_k, d_k$$

telle que

$$d_k \circ c_k = \mathrm{id}$$

et qu'il est infaisable de deviner k ou x en connaissant une collection de $c_k(x)$.

- chiffrage de flot : opère sur des flots ;
- chiffrage de bloc : opère sur des blocs de taille fixée.

En ce moment : AES-128.

Algorithmes symétriques

Les algorithmes vus ci-dessus sont symétriques :
la même clé est utilisée par l'émetteur et le récepteur.

Problème : le nombre total de clés est quadratique.

Les algorithmes asymétriques utilisent deux clés :

- une clé secrète pour l'authentification et une clé publique pour la vérification ;
- une clé publique pour le chiffrage et une clé privée pour le déchiffrage.

Exemple : dans un groupe modulaire $\mathbf{Z}/p\mathbf{Z}$,

- la clé privée est un élément inversible $a \in \mathbf{Z}/p\mathbf{Z}$;
- la clé publique est $A = a^{-1}$;
- le chiffrage est la multiplication par A,
  le déchiffrage est la multiplication par a.

Algorithmes lents, combinés à du symétrique.

Construction : Diffie-Helman

Diffie-Hellman est un algorithme d'échange de clés : se mettre d'accord sur une clé à travers un canal :

- non-confidentiel ;
- authentifié.

On se donne un groupe modulaire $\mathbf{Z}/p\mathbf{Z}$ et un générateur g. p grand, donc le logarithme discret est infaisable.

A choisit un secret a et calcule $\mathbf{A} = g^a$.
B choisit un secret b et calcule $\mathbf{B} = g^b$.
A envoie $\mathbf{A}$. B envoie $\mathbf{B}$.
A calcule $K_a = B^a$. B calcule $K_b = A^b$.

$$K_a = 2^{ba} = 2^{ab} = K_b$$

Si le canal n'est pas authentifié, vulnérable au MITM.

Protocoles de sécurité

Protocole de sécurité : procédures qui visent à :

- implémenter une politique de sécurité ;
- sous un certain modèle de sécurité.

On peut :

- utiliser un protocole standard ;
- implémenter un protocole ad-hoc.

Protocoles standard : WPA2 et WPA3

En filaire, on utilise de la sécurité de couche physique.

Sans fil :

- sécurité physique impossible ;
- sécurité de couche lien.

Longue histoire de protocoles cassés (WEP, A5/1).

Pour le WiFi, on utilise WPA2 et on transitionne à WPA3.

(Pourquoi a-t-on besoin de sécurité de couche lien ?)

IPSec

IPSec devait être un protocole de sécurité de couche réseau qui résout tous les problèmes.

Échec : uniquement adapté pour faire des VPN.

Et faire un VPN, c'est facile, on n'a pas besoin d'IPSec pour cela.

TLS et DTLS

TLS (anciennement SSL) est le protocole de sécurité de couche transport le plus généralement déployé.

C'est le « s » dans « https ».

Une histoire de failles de sécurité (heartbeat, etc.), mais on a aujourd'hui plusieurs bibliothèques utilisables.

De toute façon, on n'a pas le choix :
HTTP, SMTP, NNTP utilisent tous TLS.
Seul résistant : ssh (et sftp, et rsync).

DTLS est la variante datagramme de TLS.
Protocole connecté, inadapté à la communication n – n.

Protocoles de couche application

Protocole ad hoc à la couche application :

- les meilleures propriétés (bout-en-bout, adapté à l'application) ;
- difficile à implémenter.

C'est généralement le cas dans les protocoles de messagerie. Par contre, les logiciels de vidéoconférence sont sécurisés à la couche transport.

L’Internet des objets

L’Internet des Objets *(Internet of things)* :
mode consistant à tout connecter à l’Internet :

- machines à soda du MIT et CMU (1977–78) ;
- « assistant personnel » (Siri, Alexa, pas Robert) ;
- télévision connectée ;
- frigo connecté (!) ;
- ampoules connectées (!!) ;
- jouets connectés (!!!).

La sécurité de ces « objets » est une catastrophe.

(Mauvaise idée : pirater le nounours du fils du voisin.)