Protocoles de sécurité

Mathieu Cunche[1]

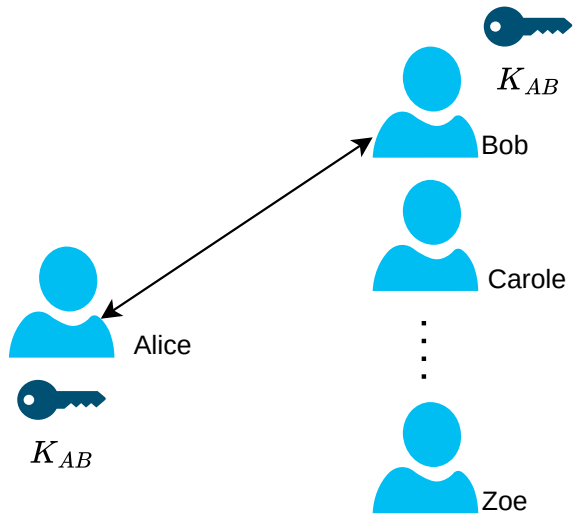
[1]INSA-Lyon

Problème du partage de clefs

Pour communiquer, Alice et Bob utilisent une clef commune (K_{AB}) qu'ils se sont échangés lors de leur dernière rencontre.

Comment faire si Alice veut aussi communiquer avec Carole, ..., Zoe ?

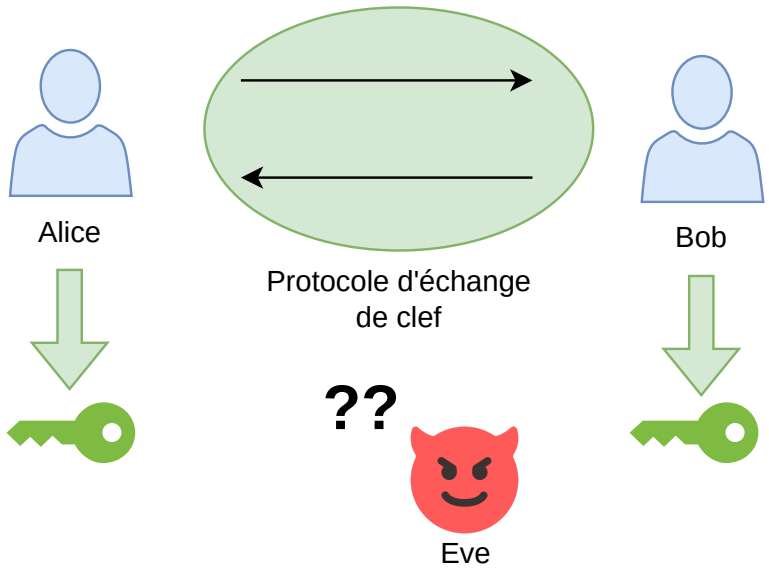
- Une clef secrète avec chacun d'eux
  - Autant de clefs que de correspondants
  - Rencontre préalable de tous les correspondants
  - Ne passe pas à l'échelle !
- Utiliser un protocole d'échange de clef !



Protocole d'échange de clefs

Protocole d'échange de clef :

- "mécanisme par lequel plusieurs participants se mettent d'accord sur une clé cryptographique"[1]
- implique des calculs (ex.: chiffrement, hachage) et des échanges de messages

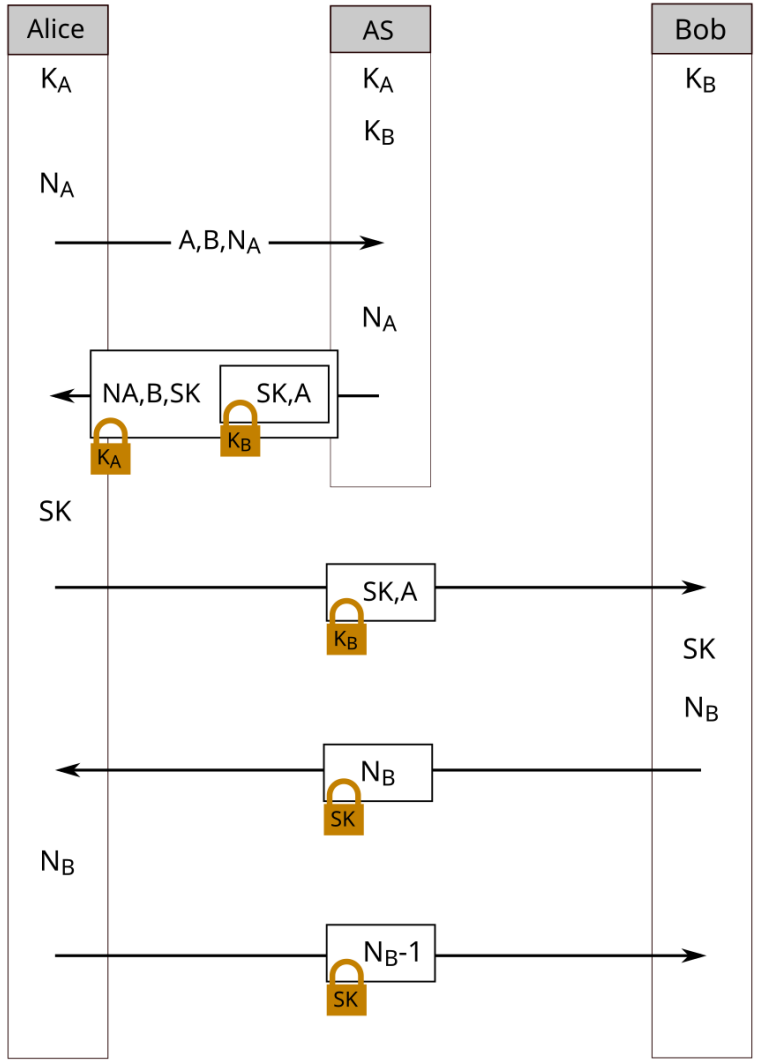


[1] source : wikipedia https://fr.wikipedia.org/wiki/%C3%89change_de_cl%C3%A9

Protocole de Needham Schroeder

- Acteurs :
  - Alice et Bob
  - AS : serveur d'authentification
- Hypothèse de départ :
  - Alice et Bob n'ont pas de secret partagé
  - Serveur AS partage une clef avec chaque acteur (K_A et K_B)
- Eléments importants :
  - Des *Nonce* (N_A, N_B) : nombres aléatoires
  - Une clef de session SK

Alice et Bob partagent maintenant une clef secrète SK !



Groupe

Ensemble d'éléments associés à une opération avec des propriétés : loi de composition interne, associativité, élément neutre, inverse.

Le Groupe $(\mathbb{Z}_p, \times)$: ensemble des entiers > 0 modulo un entier p premier.
Exemple avec $p = 5$: $(\mathbb{Z}_5, \times)$

- Eléments : $\{1, 2, 3, 4\}$
- Opération interne : $2 \times 4 \bmod 5 = 3$

Groupe cyclique :

- Il existe un g tels que les puissances successives g^i permettent de générer l'ensemble des éléments du groupe
- g est appelé un générateur du groupe
- 2 est générateur de $(\mathbb{Z}_5, \times)$ car $1 = 2^4$, $2 = 2^1$, $3 = 2^3$ et $4 = 2^2$

Problème du logarithme discret

Logarithme discret

Log. discret[a] : Etant donné un groupe $(\mathbb{Z}_p, \times)$, avec son générateur g, et un élément x, trouver y tel que $g^y = x$.

[a] "Discret" car il s'agit d'entiers et non de réels.

Le problème du logarithme discret est **difficile** !

Pour un groupe $(\mathbb{Z}_p, \times)$, chaque élément est représenté sur n bits, avec $p \simeq 2^n$.

- Approche exhaustive : $pour i \in [1..p]$ calculer g^i
  - Complexité : $\mathcal{O}(p) = \mathcal{O}(2^n)$ (exponentiel par rapport à n)
- Algo. Rho de Pollard
  - Complexité : $\mathcal{O}(\sqrt{p}) = \mathcal{O}(2^{n/2})$ (exponentiel par rapport à n)
- On ne connaît pas d'algo efficace (complexité polynomiale : $\mathcal{O}(n^c)$)

Si on choisit p suffisament grand, alors il devient **impossible en pratique** de calculer le log. discret.

Diffie-Hellman I

Le protocole de Diffie-Hellman

- Un protocole d'échange de clef publié en 1976 (prix Turing en 2015)
- Sécurité basée sur le problème du logarithme discret
- Première construction de cryptographie à clef publique

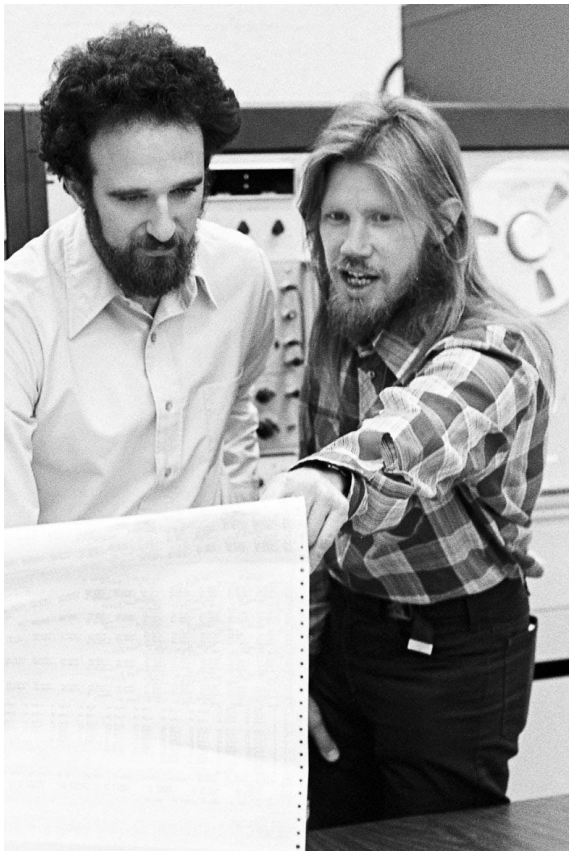
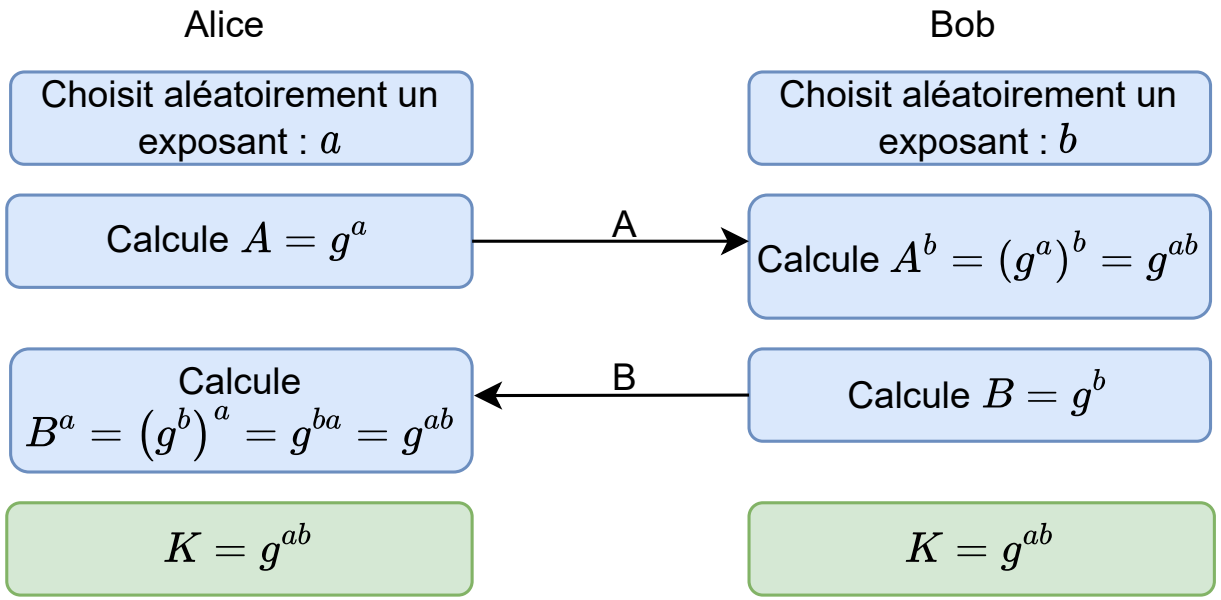


Figure: Whitfield Diffie and Martin Hellman

Diffie-Hellman II

- Hypothèse : Alice et Bob se sont mis préalablement d'accord sur un groupe et son générateur g



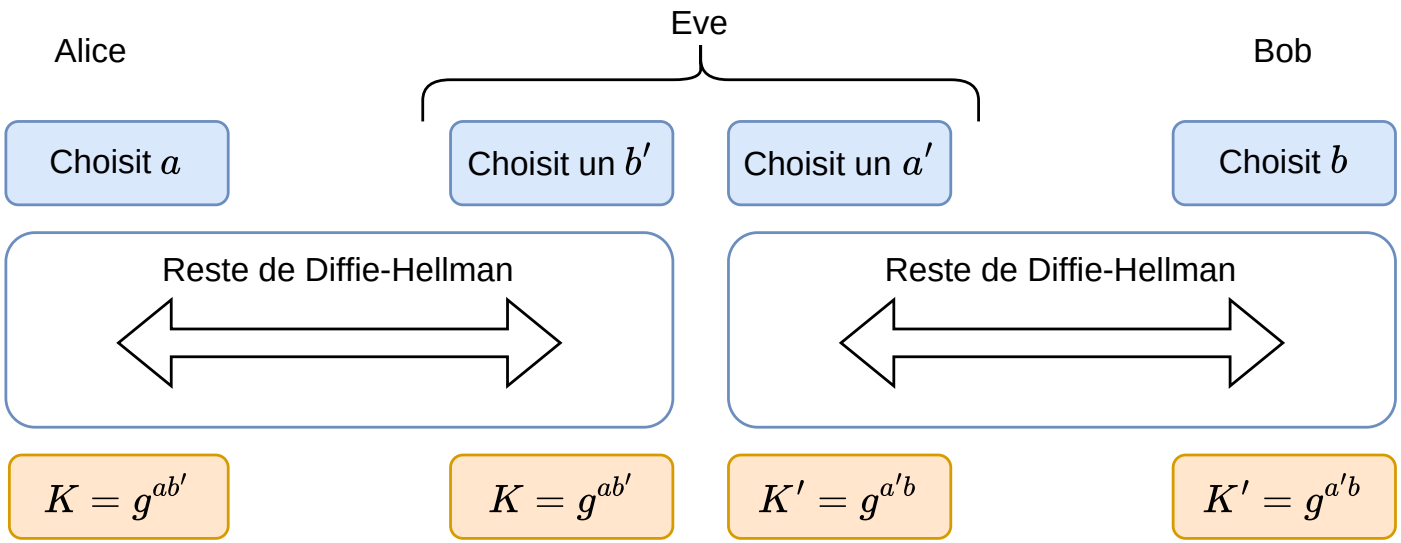
Alice et Bob ont maintenant une clef commune $K = g^{ab}$
[2]

[2] Exercice : pourquoi Eve ne peut pas retrouver cette clef ?

Diffie-Hellman : attaque MitM

Attaque *Man(ipulator) in the Middle (MitM)*

- Eve va se placer entre Alice et Bob
  - Eve se fait passer pour Bob auprès de Alice et inversement



Eve connaît la clef que Alice (resp. Bob) va utiliser. Elle déchiffre puis rechiffre tous les messages et accède ainsi à l'échange en clair tandis que Alice et Bob ne se doutent de rien !

PKI

Infrastructure à clef publique : PKI (*Public Key Infrastructure*)

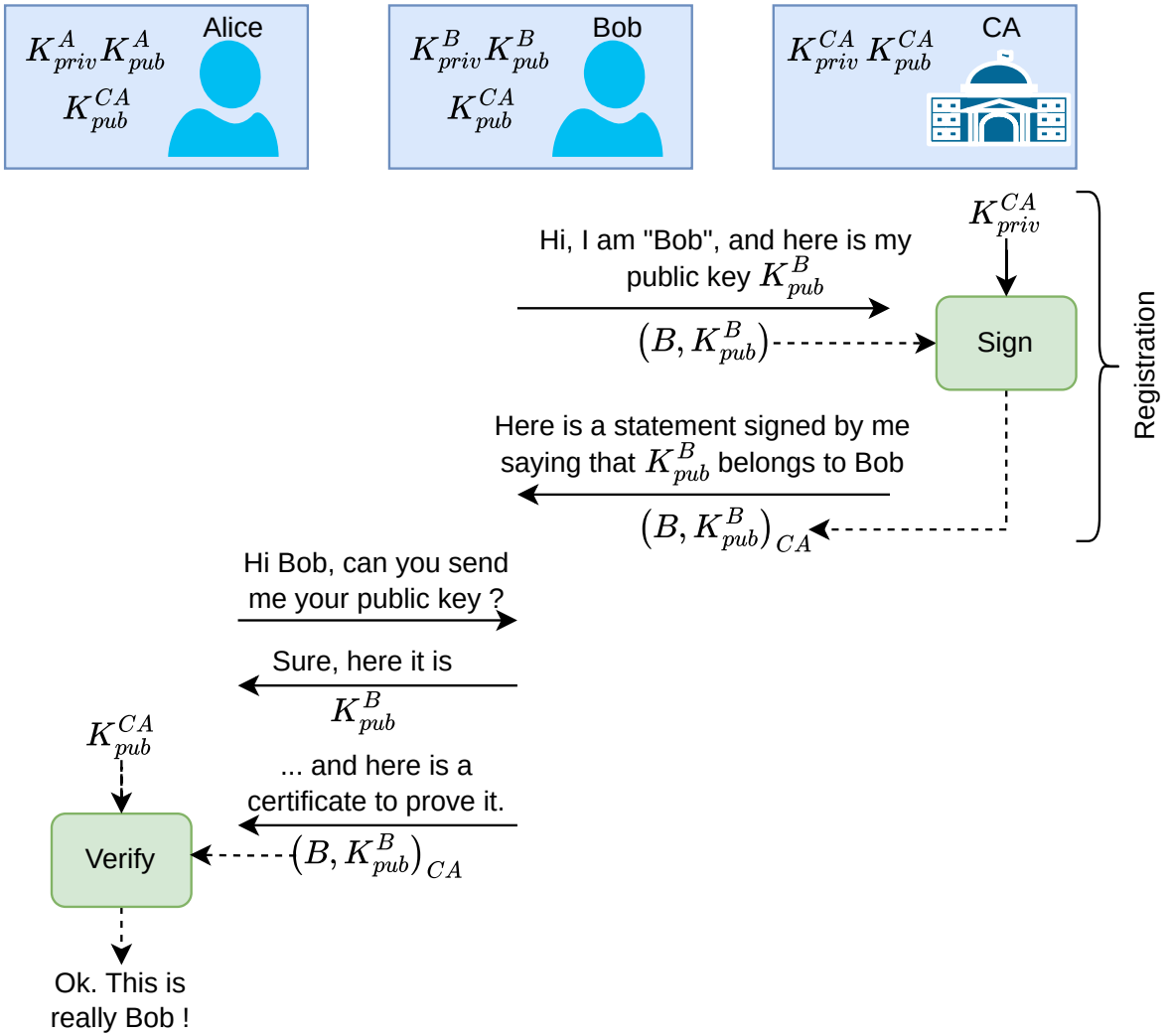
- Alternative efficace aux serveurs de clefs (ex.: Needham-Schroeder)
- Permet de savoir à qui appartient chaque clef
  - Association `Identité` $\leftrightarrow$ `Clef publique`
- Implique (au moins une) Autorité de Certification (CA)
  - Tiers de *confiance*
  - Dispose d'une paire de clef
    - La clef publique est connue de tous !
  - Émet des certificats
    - Association (`Id.` $\leftrightarrow$ `Clef pub.`) signée avec sa clef publique[3]

[3] En pratique, un certificat contient d'autres informations (Identité du CA, date de validité ...)

PKI : principe général

Hypothèses :

- Alice, Bob et l'autorité de certification (CA) ont tous une paire (K^X_{priv}, K^X_{pub})
- Alice et Bob connaissent K^{CA}_{pub}
- Alice et Bob font *confiance* à CA



Alice connaît la clef publique de Bob !

PKI

La suite en TD :

- Certificat
- Liste de révocation
- ...

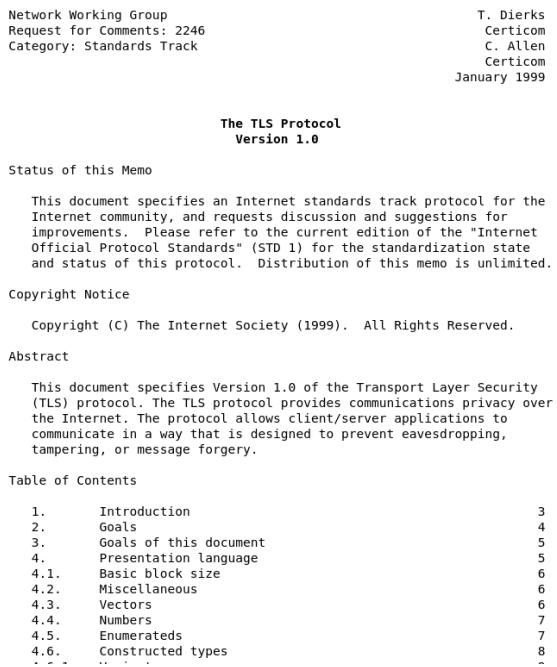
TLS

Le protocole TLS : *Transport Layer Security*

- Successeur de SSL (*Secure Socket Layer*)
- TLS 1.0 publié ne 1999 (RFC 2246)

  The TLS protocol provides communications **privacy** *over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent* **eavesdropping, tampering, or message forgery.**

- Objectifs : **Confidentialité**, **Intégrité** des échanges et **Authentification** des correspondants



Network Working Group — T. Dierks, Certicom
Request for Comments: 2246 — C. Allen, Certicom
Category: Standards Track — January 1999

The TLS Protocol
Version 1.0

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Abstract

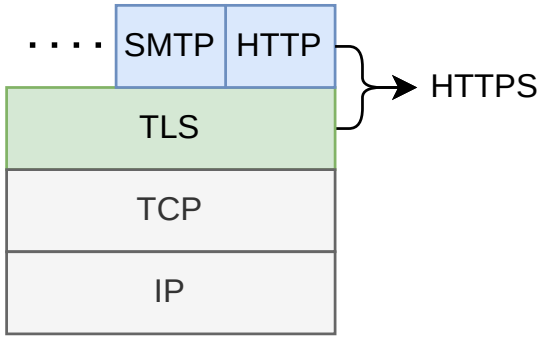
This document specifies Version 1.0 of the Transport Layer Security (TLS) protocol. The TLS protocol provides communications privacy over the Internet. The protocol allows client/server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery.

Table of Contents

1. Introduction 3
2. Goals 4
3. Goals of this document 5
4. Presentation language 5
4.1. Basic block size 6
4.2. Miscellaneous 6
4.3. Vectors 6
4.4. Numbers 7
4.5. Enumerateds 7
4.6. Constructed types 8

TLS dans la pile protocolaire

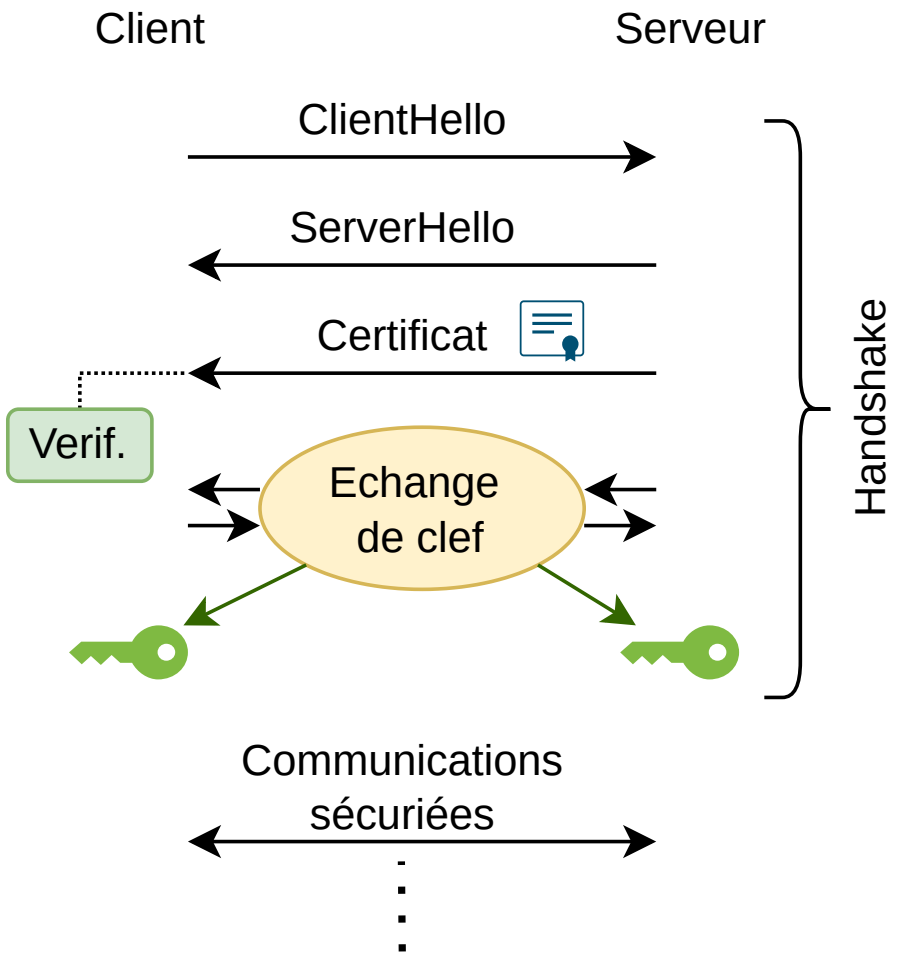
- TLS : un protocole de la couche session (entre la couche transport et la couche applicative)
- Utilisé pour sécuriser de nombreux protocoles (SMTP, HTTP, ..)
- HTTPS = HTTP + TLS



TLS : le protocole

Version simplifiée :

- *Handshake* TLS
  - Vérification du certificat du serveur
  - Génération d'une clef commune (ex. : Diffie-Hellman)
- Communications sécurisées
  - Chiffrement + verif. intégrité + authentification



TLS : les faiblesses

TLS a été affecté par des problèmes de sécurité :
BEAST, CRIME, Lucky 13, Heartbleed, FREAK, POODLE, Logjam, DROWN ...

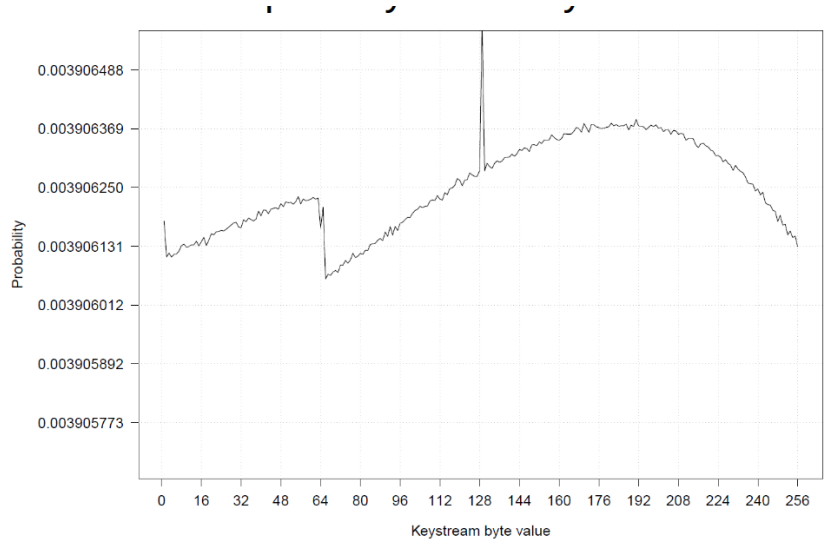
Problèmes qui ont été globalement corrigés.

TLS : les faiblesses

RC4 NoMore

- *Downgrade attack* : force l'utilisation du chiffrement RC4 (faible)
- Exploitation de biais statistiques : certaines valeurs sont plus probables que d'autres
- Récupération du message en clair

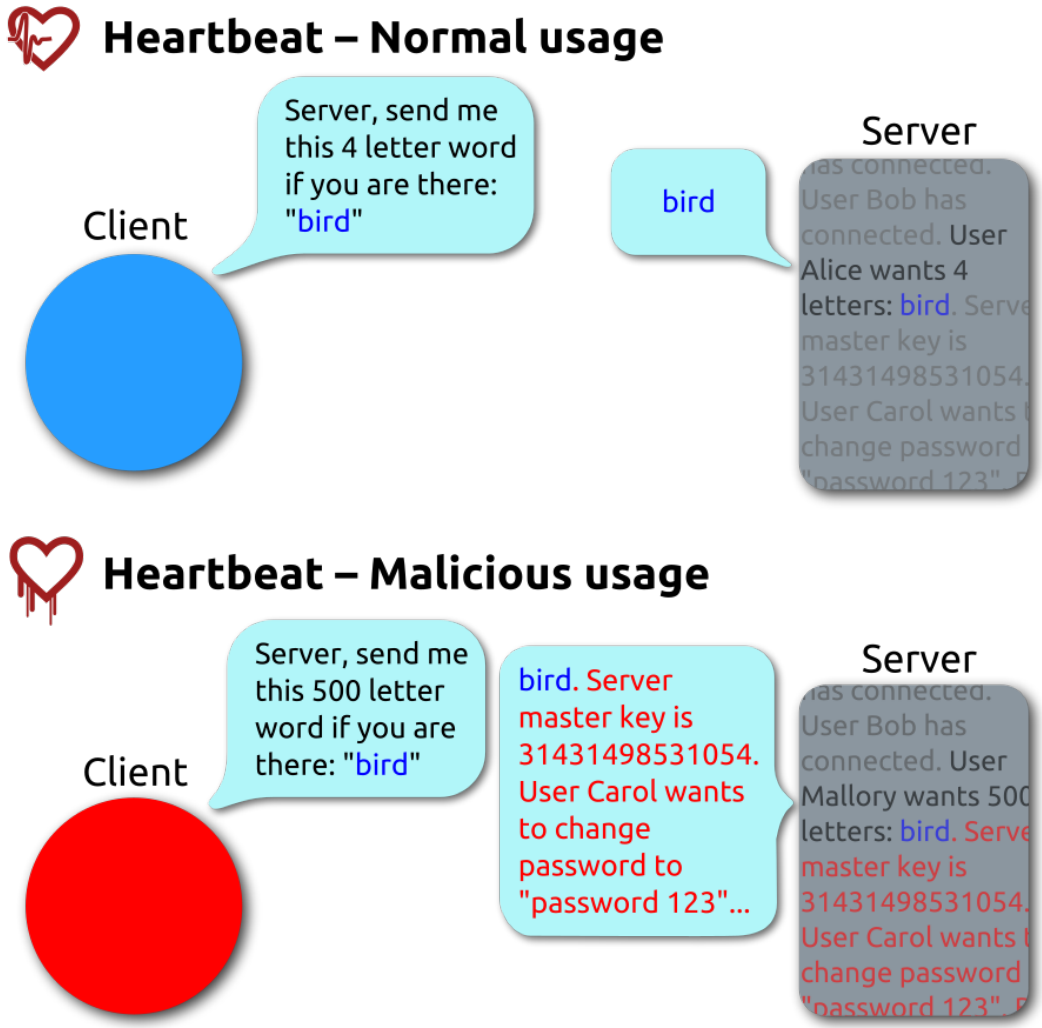
Figure: Mathy Vanhoef



TLS : les faiblesses

Heartbleed (2014)

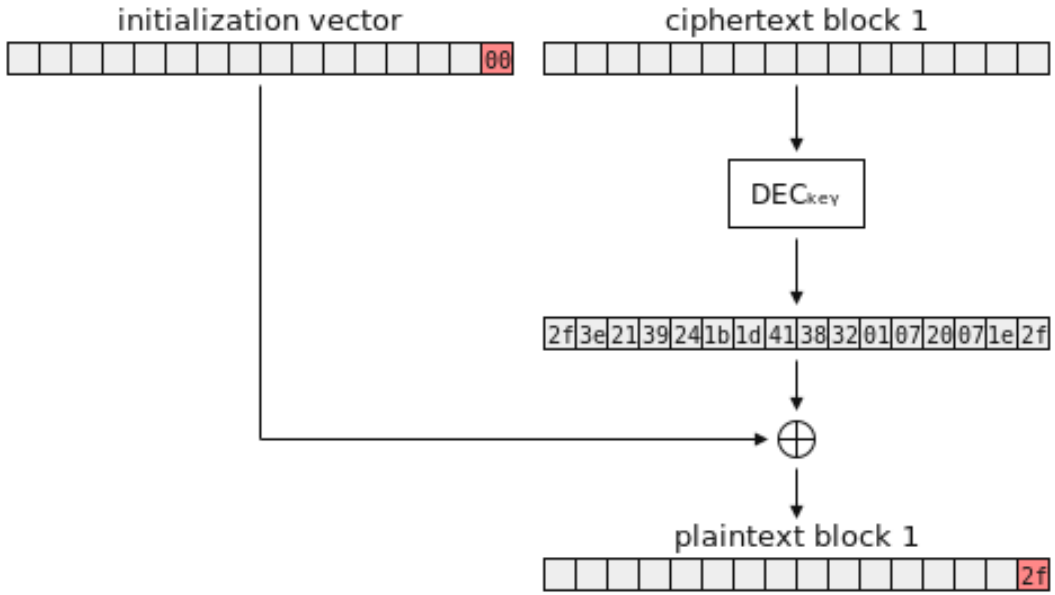
- *buffer over-read* : lire plus de données que prévues
- Accès en lecture à des infos internes telles que des clefs secrètes



TLS : les faiblesses

POODLE (2014) : "**Padding Oracle** On **Downgraded** Legacy Encryption"

- Downgrade attack : force l'utilisation d'une ancienne version (SSL 3.0)
- Exploitation d'un oracle : retrouve le message en testant la valeur des octets un par un (en moyenne 256 requête SSL 3.0 pour trouver un octet)



Source : https://www.nccgroup.com/us/research-blog/cryptopals-exploiting-

Sécurité : les maillons faibles I

La sécurité peut être menacée par :

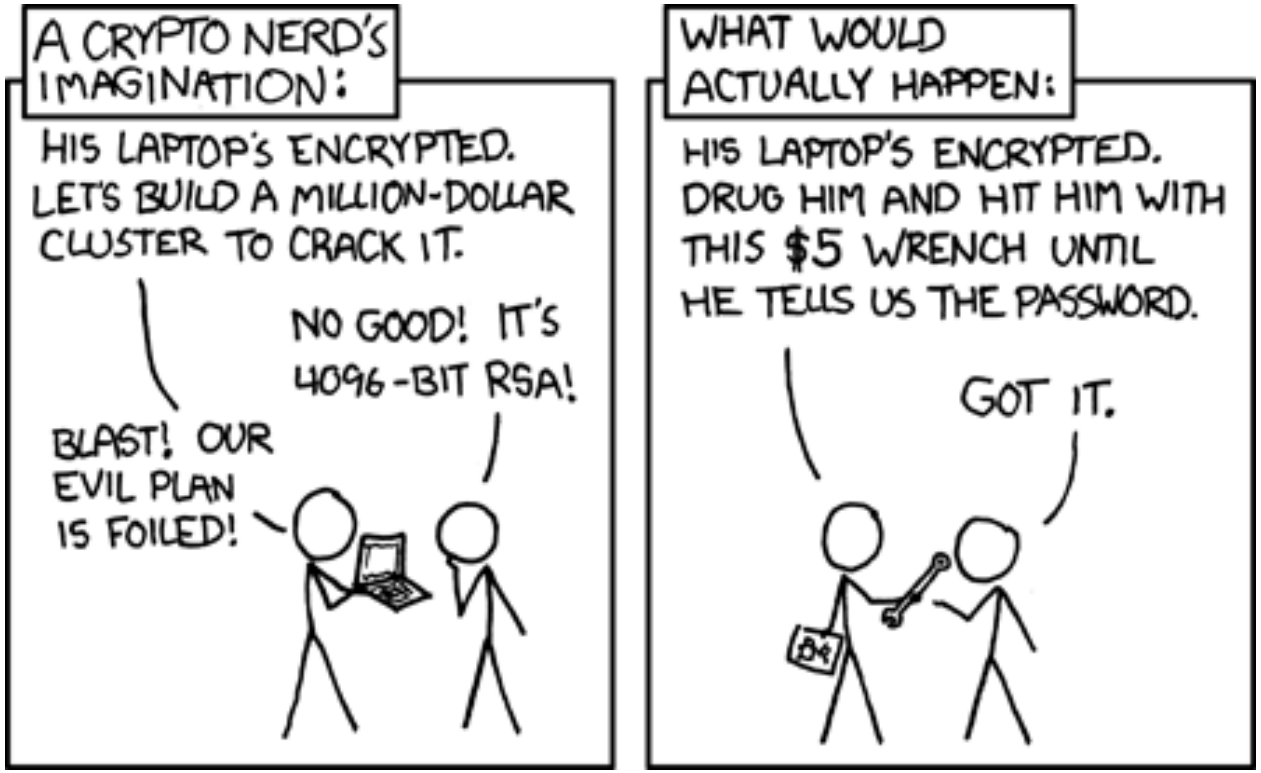
- Faiblesses des primitives cryptographiques (ex. : MD5, DES, RC4 ...)
- Erreurs de conception des protocoles/systèmes (ex. : POODLE)
- Erreurs d'implémentation (ex. : Heartbleed)

Mais aussi et surtout par ...

Sécurité : les maillons faibles II

L'humain !

- Faillible, corruptible, vulnérable ...



https://xkcd.com/538/

Sécurité : les métiers

Travailler dans la *Cybersécurité*[4]

- Diversité de métiers : développement, opérationnel, management ...
- Diversité de secteurs : informatique, télécom, service, banque, commerce, santé, industrie ...

Exemples de métiers :

- Analyste
- Architecte sécurité
- Responsable de la Sécurité des Systèmes d'Information (RSSI)
- Consultant.e en cybersécurité
- Auditeur-rice / Pentesteur-euse
- Chercheur-euse
- ...

[4] https://cyber.gouv.fr/publications/panorama-des-metiers-de-la-cybersecurite

Pour aller plus loin

Actualités :

- Schneier on Security `https://www.schneier.com/`
- Slashdot `https://slashdot.org/`
- ...

Challenges :

- la plateforme RootMe `https://www.root-me.org/`
- les CTF de l'ANSSI `https://cyber.gouv.fr/se-former-par-le-jeu-avec-les-ctf-de-lanssi`
- le le MOOC de l'ANSSI `https://secnumacademie.gouv.fr/`