Sécurisation des communications

1 Motivation

Avec le développement du commerce en ligne dans le Web il est vite apparu que les communications devaient être sécurisées. Divers protocoles ont été créés alors pour chiffrer les communications
Dans un premier temps nous allons nous intéresser au chiffrement à clef privée puis au chiffrement à clef publique enfin au protocole TLS (Transport Security Layer)
L'objectif de la cryptographie est de permettre la circulation sous forme cachée ou chiffrée C d'un message en clair M entre deux personnes, traditionnellement appelées Alice et Bob de telle sorte qu'une troisième personne, appelée Oscar, non autorisée ne puisse pas retrouver M à partir de C dans un temps "raisonnable"

La transformation de M en C appelée **chiffrement** se fait par l'intermédiaire d'une fonction (au sens mathématique et algorithmique aussi) de chiffrement E (encryption) telle que $C = E(M)$

Le **déchiffrement** de C en M se fait par l'intermédiaire d'une fonction D telle que $M = D(C)$

Par conséquent $D(E(M)) = M$

2 Chiffrement à clef privée

Les relations fondamentales ci-dessus s'écrivent avec un paramètre K appelé **clé**
$C = E_K(M)$ et $M = D_K(C)$
Cette clé est **partagée secrètement entre Alice et Bob**
Alice et Bob conviennent aussi d'un algorithme de chiffrement et de déchiffrement
La "sécurité de ce système" repose sur la clé qui est acheminée entre Alice et Bob par un "canal sécurisé"

C'est le principe de Kerckhoffs(1883) :

"La sécurité d'un système de chiffrement ne doit pas dépendre de la préservation du secret de l'algorithme. La sécurité ne repose que sur le secret de la clé."

Jusqu'au jour où quelques individus ont remis en cause le principe de Kerckhoffs. (on verra cela dans la révolution du chiffrement à clef publique (1977))

2.1 Chiffrement par décalage

Dans une première approche on suppose que le texte à chiffrer est composé uniquement que de lettres latines non accentuées minuscules

Par exemple : **les sanglots longs des violons de l'automne blessent mon coeur d'une langueur monotone**

La clef est un entier K tel que $1 \leqslant K \leqslant 25$

Supposons que $K = 5$

On associe à chaque lettre minuscule dans l'ordre alphabétique un entier entre 0 et 25 et à chaque entier entre 0 et 25 une lettre minuscule

Ex 1

Créer deux dictionnaires Python `code` et `car` tels que par xemple `code['c'] = 3` et `car[3] = 'c'`

La lettre a est associée à 0, b à 1, etc... et z à 25 donc décaler de 1 c'est chiffrer a par b , b par c et z par a, et décaler de 5 c'est chiffrer a par e, b par f et z par d

Puisqu'on veut numériser ce processus on introduit quelques notations :

1. On note $\mathbb{Z}_{26}$ les entiers de 0 à 25 muni de l'addition telle que
   $a + b = (a + b)\%26$ ainsi par exemple $4 + 23 = 2$
2. Ainsi l'opposé d'un entier $a \in \mathbb{Z}_{26}$ est $b \in \mathbb{Z}_{26}$ tel que $a + b = 26$ ainsi l'opposé de 5 , noté -5 est 21
3. Ainsi $E_K(c) = \text{code}[c] + K$ où c est une lettre minuscule latine non accentuée

Ex 2

1. Définir $D_K()$
2. Définir une fonction Python `chiffrer(lettre,K)`
3. Pourquoi n'est il pas nécessaire d'écrire une fonction `dechiffrer(lettre,K)` ?
4. Définir une fonction `chiffrer2(message,K)` où message est une chaîne de caractères composé uniquement de lettres latines minuscules non accentuées

Ex 3

(A faire en binôme)

1. Vous êtes Alice : Aller sur le Web chercher les paroles d'une chanson en Anglais, la copier et la chiffrer par décalage (il faudra peut-être transformer les lettres majuscules en minuscules ...)
   Puis donner le message chiffré à Bob (et la clé par un autre canal sécurisé)
2. Bob doit déchiffrer le message
3. Alice recommence avec une autre chanson et une autre clé mais cette fois ci le binôme est Oscar (il n'a pas la clé) et il doit essayer de casser le code. Proposer une méthode
4. Recommencer avec un texte plus volumineux (utiliser un fichier texte)

2.2 One-time-Pad

(Voir TP)

2.3 Chiffrement affine

Lorsqu'Alice veut chiffrer une lettre dont le code est $n \in \mathbb{Z}_{26}$ elle utilise une fonction affine f définie par $f(x) = ax + b$

Dont la lettre chiffrée est $f(n) = an + b = m \in \mathbb{Z}_{26}$

Pour que Bob puisse déchiffrer $an + b = m$ et retrouver n il est nécessaire que a soit **inversible** dans $\mathbb{Z}_{26}$ c'est à dire il existe $c \in \mathbb{Z}_{26}$

tel que $ac = 1$

Par exemple 15 est inversible dans $\mathbb{Z}_{26}$ car $15 \times 7 = 105 = 4 \times 26 + 1 = 1$ modulo 26

Pour déchiffrer Bob calcule $c(m-b)$ car $c(m-b) = c(an) = (ac)n = 1 \times n = n$
Alice et Bob partagent la clé privée constituée par le couple (a, b) le problème est :

Comment à partir de a Bob peut trouver c ?

Définition 1 *Deux entiers sont dits premiers entre eux s'ils n'ont pas de diviseurs communs*

Exemple : 15 et 26 sont premiers entre eux mais 4 et 26 ne le sont pas car ils ont un diviseur commun 2

Théorème 1 *a et b premiers entre eux $\iff$ il existe u et v tel que $au + bv = 1$*

Exemple :

$15 \times 7 + 26 \times (-4) = 1$

L' **algorithme d'Euclide étendu** permet d'avoir u et v étant donnés a et b (Voir TP)

Théorème 2 *a est inversible dans $\mathbb{Z}_{26}$ $\iff$ a est premier avec 26*

Preuve :

a est premier avec 26 $\iff$ il existe u et v tel que $au+26v = 1$ $\iff$ $au = 1-26v = 1$ dans $\mathbb{Z}_{26}$ ce qui signifie que a est inversible dans $\mathbb{Z}_{26}$

3 Système cryptographique à clé publique

"*Comme nous, Ralph était un peu fou. Il faut être fou pour mener jusqu'au bout une recherche originale, et seuls des fous la poursuivront envers et contre tout. Vous avez une première idée, vous vous enthousiasmez, et ça ne marche pas. Alors vous passez à l'idée numéro 2, vous vous enthousiasmez, et ça ne marche pas. Alors vous avez l'idée 99, vous vous enthousiasmez, et ça ne marche pas. Seul un fou sera encore enthousiaste à sa centième idée, et il faudra peut-être avoir essayé cent voies avant que l'une conduise quelque part. A moins d'être assez fou pour retrouver à chaque fois tout votre enthousiasme, vous vous découragerez, et n'aurez pas l'énergie de mener les choses jusqu'au bout. Dieu bénit les fous.*"(Martin Hellman, professeur d"informatique à l'université de Stanford - Californie (1975))

3.1 Le Protocole de Diffie-Hellman (1976)

Alice et Bob vont utiliser une méthode traditionnelle de chiffrement à clé secrète K mais **ils veulent que cette clé ne soit pas acheminée mais déduite par eux seuls par calcul.**

Définition 2 *p premier*

$(\mathbb{Z}_p^*, \times)$ est un groupe de $p-1$ éléments

On dit que r est une racine primitive modulo p si :

$\forall x \in \mathbb{Z}_p^* \; \exists \alpha \in \mathbb{N}$ *tel que* $x = r^\alpha$

Exemples

1. Pour $p = 5$

   2 est une racine primitive modulo 5 car les 4 éléments non nuls de $\mathbb{Z}_5$ c'est à dire 1,2,3 et 4 peuvent s'écrire comme des puissances de 2 :
   $2^1 = 2, 2^2 = 4, 2^3 = 3, 2^4 = 1$
2. Pour $p = 7$

   2 n'est pas une racine primitive modulo 7 car
   $2^1 = 2,\ 2^2 = 4,\ 2^3 = 1$ et 5 par exemple ne peut pas s'exprimer comme une puissance de 2 dans $\mathbb{Z}_7^*$
   3 est une racine primitive modulo 7 car :
   $3^1 = 3, 3^2 = 2, 3^3 = -1, 3^4 = 4, 3^5 = 5, 3^6 = 1$

Théorème 3 *Pour tout p premier il existe une racine primitive modulo p*

Protocole

1. Alice et Bob se mettent d'accord **publiquement** sur un "grand" nombre premier p (ici on prendra $p = 7$) et une racine primitive modulo p soit r (ici $r = 3$)
2. Alice choisit un nombre a **secret**, connu d'elle seule. Choisissez $a =$ .... (de 1 à 6). Et elle transmet à Bob et à qui veut l'entendre le nombre $\alpha = r^a \mod p$.
   Que vaut ici $\alpha = 3^a \mod 7 =$.....?
3. Bob choisit de même un nombre b **secret** et connu de lui seul,et transmet à Alice $\beta = r^b \mod p$. Choisir une valeur pour b. $b =$ .....
   $\beta =$ .....
4. Ici est le coeur du protocole : **la clé secrète** K **est** $K = \alpha^b \mod p = \beta^a \mod p$
   $\alpha^b \mod 7 =$ ..... et $\beta^a \mod 7 =$ .....
   Autrement dit Alice prend ce que lui a envoyé Bob , c'est à dire β et l'élève à la puissance "le nombre secret" d'Alice et obtient la clé secrète K qui servira à chiffrer.
   De même Bob prend ce que lui a envoyé Alice, c'est à dire α et l'élève à la puissance "le nombre secret" de Bob et obtient ô miracle mathématique le même nombre K
5. Oscar a peut-être intercepté les valeurs de p puis r puis r^a puis r^b mais il arrivera difficilement à obtenir a à partir de α ou b à partir de β, parce que l'exponentiation modulaire est une fonction **à sens unique** (voir plus loin)

Théorème 4 *p premier*
r une racine primitive modulo p
Si $a, b \in [\![1, p-1]\!]$ *et* $\alpha = r^a \ mod\ p$ *et* $\beta = r^b \ mod\ p$
Alors $\alpha^b = \beta^a \ mod\ p$

Preuve
$\alpha^b = (r^a)^b = r^{ab} = (r^b)^a = \beta^a$

Théorème 5 *(Théorème du logarithme discret)*
r une racine primitive modulo p
$\mathbb{Z}_p^* \ni n \to r^n \in \mathbb{Z}_p^*$ *est injective*

Preuve

Si $r^n = r^m$ or on a vu sur des exemples plus haut que la suite des puissances de r^k est périodique de période $p-1$ car l'exponentiation modulaire est surjective et à cause du Théorème de Fermat $r^{p-1} \equiv 1\ [p]$

or n et m appartiennent à $[\![1, p-1]\!]$ donc $n = m$

Etant injective et surjective, l'exponentiation modulaire est **bijective** donc inversible

Définition 3 *La fonction réciproque de l'exponentiation modulaire de base r modulo p premier est :*

le logarithme discret de base r, noté $\log_r$*, où r est une racine primitive modulo p premier et définie par :*

$\log_r(r^n) = n$ *avec* $n \in \mathbb{Z}_p^*$

Pratiquement il se trouve que calculer l'image d'un élément de $\mathbb{Z}_p^*$ par l'exponentiation modulaire est "facile" même si p est un "grand" nombre premier (voir exercice)

Par contre pour calculer l'image d'un élément de $\mathbb{Z}_p^*$ par la fonction réciproque, le logarithme discret de base r, le temps mis sera "très grand" (plusieurs années)

On dit alors que l'exponentiation modulaire est une **fonction à sens unique** dans le sens où étant donné une image il est difficile de trouver l'antécédent

3.2 Le cryptosystème RSA (Rivest, Shamir, Adleman) (1977). Chiffrement asymétrique et à clé publique

Chiffrement RSA

1. Alice et Bob ont chacun leurs clés publiques P_A et P_B (comme des numéros de téléphone) et leurs clés secrètes S_A et S_B. Ces clés sont des **paires** d'entiers.
2. Les clés d'Alice (et de même pour Bob) définissent des fonctions $P_A()$ et $S_A()$ à sens unique, **commutatives** c'est à dire :

$$S_A(P_A(M)) = P_A(S_A(M)) = M$$

3. Supposons que Bob veuille envoyer un message M chiffré à Alice. Il se procure la clé publique d'Alice P_A comme un numéro de téléphone dans un annuaire.
4. Bob calcule le texte chiffré $C = P_A(M)$ et envoie C à Alice.
5. Lorsque Alice reçoit C elle calcule $S_A(C) = S_A(P_A(M)) = M$ et elle a accès au texte en clair M.

Comment être sûr qu'un message vient bien de la personne censée l'avoir écrit ?

RSA permet de définir la : **Signature numérique d'un message**

1. Alice envoie un message M' à X . Elle associe à M' une signature définie par $s = S_A(M')$
2. Lorsque X reçoit (M',s) pour s'assurer que c'est bien Alice qui a envoyé M' ,il calcule $P_A(s)$ et regarde s'il obtient M'
3. Si $s = S_A(M')$ alors $P_A(s) = P_A(S_A(M')) = M'$, sinon si $s \neq S_A(M')$ il n'obtiendra pas M'

Un message peut être à la fois **chiffré** et **signé** (voir exercice)

Comment ça marche mathématiquement ?Pour bien comprendre comment fonctionne RSA il nous faut préciser quelques notions mathématiques :

Définition 4 *Attention n un entier non premier on note $\mathbb{Z}_n^*$ l'ensemble des entiers strictement positifs et **premiers avec** n C'est aussi les éléments inversibles modulo n*

Le nombre d'éléments de $\mathbb{Z}_n^*$ est noté $\phi(n)$ où la fonction ϕ est appelée indicatrice d'Euler

Théorème 6 *1. Si p premier alors $\phi(p) = p - 1$*

2. Si n et m sont premiers entre eux alors $\phi(nm) = \phi(n)\phi(m)$

(Voir le théorème des restes chinois (plus loin))

Théorème 7 *(Théorème d'Euler)*

$\forall a \in \mathbb{Z}_n^*$ *on a* $a^{\phi(n)} \equiv 1\ [n]$

Preuve Soit a un élément quelconque de $\mathbb{Z}_n^*$

Considérons la fonction $f_a : \mathbb{Z}_n^* \ni x \to ax \in \mathbb{Z}_n^*$

injective car si $ax = ax'$ en multipliant à gauche par l'inverse de a on obtient $x = x'$

surjective car pour tout $y \in \mathbb{Z}_n^*$ il existe $x \in \mathbb{Z}_n^*$ tel que $y = ax$

En effet $x = a^{-1}y$

Donc f_a est bijective et on peut voir f_a comme une **permutation** sur l'ensemble **fini** $\mathbb{Z}_n^*$

Donc $\prod_{x\in\mathbb{Z}_n^*} f_a(x) = \prod_{x\in\mathbb{Z}_n^*} x$

Or $\prod_{x\in\mathbb{Z}_n^*} f_a(x) = a^{\phi(n)} \prod_{x\in\mathbb{Z}_n^*} x$ (commutativité de $\times$)

Et donc $a^{\phi(n)} \prod_{x\in\mathbb{Z}_n^*} x = \prod_{x\in\mathbb{Z}_n^*} x$ donc $a^{\phi(n)} \equiv 1\ [n]$

Théorème 8 *(Théorème des restes chinois) Soit n_1 et n_2 deux entiers **premiers entre eux** et $n = n_1 n_2$ Soit f la fonction définie par :*

$Z_n \ni a \to (a_1, a_2) \in Z_{n_1} \times Z_{n_2}$ *où $a \equiv a_i\ [n_i]$ avec i = 1et 2*

Cette fonction est bijective et respecte l'addition et la multiplication au sens où :

$f(a + b) = f(a) + f(b)$ *et* $f(ab) = f(a) \times f(b)$

Preuve

Montrons l'existence directement de f^{-1}

Puisque n_1 et n_2 sont **premiers entre eux** donc n_1 est inversible dans Z_{n_2} et n_2 est inversible dans Z_{n_1}

Soit $c_1 = n_2 \times (n_2^{-1} \mathrm{mod} n_1)$ on a $c_1 \equiv 1\ [n_1]$ et $c_1 \equiv 0\ [n_2]$

De même $c_2 = n_1 \times (n_1^{-1} \mathrm{mod} n_2)$ on a $c_2 \equiv 1\ [n_2]$ et $c_2 \equiv 0\ [n_1]$

On définit $f^{-1}(a_1, a_2) = (c_1a_1 + c_2a_2) \mathrm{mod}\ \ n$

Et $(c_1a_1 + c_2a_2) \equiv a_1\ [n_1]$ et $(c_1a_1 + c_2a_2) \equiv a_2\ [n_2]$

Exemple

$a \equiv 2\ [5]$ et $a \equiv 3\ [13]$

L'inverse de 13 modulo 5, noté 13_5^{-1} est 2, L'inverse de 5 modulo 13, noté 5_{13}^{-1} est 8

$c_1 = 13 \times 13_5^{-1} = 26$ et $c_2 = 5 \times 5_{13}^{-1} = 40$

Donc $a = 2 \times 26 + 3 \times 40 = 172 \equiv 42\ [65]$

Corollaire 1

Le couple (a, a) de $Z_{n_1} \times Z_{n_2}$ est l'image de a dans Z_n

Corollaire 2

Les inversibles modulo n sont en même quantité que les couples inversibles dans $Z_{n_1} \times Z_{n_2}$ donc $\phi(n) = \phi(n_1) \times \phi(n_2)$

Algorithme RSA

Du côté d'Alice

1. Choisir aléatoirement deux grands nombres premiers p et q différents (de taille chacun de plus de 1024 bits)
2. Calculer $n = pq$
3. Choisir un petit entier e impair premier avec $\phi(n) = (p-1)(q-1)$
4. Calculer d l'inverse de e modulo $\phi(n)$ (Algorithme d'Euclide étendu)
5. $P_A = (e, n)$ est la clé publique RSA d'Alice
6. $S_A = (d, n)$ est la clé secrète RSA d'Alice

Du côté de Bob qui veut envoyer $M \in Z_n$ à Alice :

Le chiffrement est $E(M) = P_A(M) = M^e$ mod n

Déchiffrement du message chiffré $C \in Z_n$ venant de n'importe qui :

Le déchiffrement est $D(C) = S_A(C) = C^d$ mod n

La validité de cet algorithme repose sur le fait que

Pour tout $M \in Z_n$ on doit avoir $P_A(S_A(M) = S_A(P_A(M)) = M$

c'est à dire $M^{ed} \equiv M\ [n]$

Est ce bien le cas ?

Preuve

Puisque e et d sont inverses modulo $\phi(n) = (p-1)(q-1)$ alors il existe k entier tel que

$ed = 1 + k(p-1)(q-1)$

Donc si M n'est pas un multiple de p

$M^{ed} \equiv M(M^{p-1})^{k(q-1)} \equiv M \times (1)^{k(q-1)} \equiv M\ [p]$ (d'après le Théorème de Fermat)

Si $M \equiv 0[p]$ alors $M^{ed} \equiv 0[p]$ et $M^{ed} \equiv M[p]$

De même $M^{ed} \equiv M\ [q]$

Donc $M^{ed} \equiv M\ [p]$ et $M^{ed} \equiv M\ [q]$, d'après le corollaire du théorème des restes chinois

$M^{ed} \equiv M\ [n]$

Sécurité

la sécurité de RSA repose sur la "difficulté" de factoriser des grands entiers, mais cela suppose aussi d'être capable de trouver des grands nombres premiers

Rapidité

On combine RSA avec des systèmes rapides à clé secrète de la manière suivante si Alice veut envoyer un long message M à Bob

1. Elle choisit une clé courte K et chiffre M avec K pour obtenir C
2. Elle chiffre K avec la clé publique RSA de Bob
3. Elle transmet à Bob l'ensemble (C, $P_B(K)$)

3.3 Exercices

Ex 1

1. Pour chaque diviseur d de 10, vérifier que 2^d n'est pas congru à 1 modulo 11
2. En déduire que 2 est une racine primitive modulo 11
3. En tâtonnant trouver le logarithme discret de 3 en base 2 modulo 11
4. Supposons que p est un nombre premier de 200 chiffres et que r est une racine primitive modulo p, combien d'essais doit on faire pour trouver le logarithme discret de 3 en base r modulo p ?
5. Même question avec p un nombre premier de 1024 bits

Ex 2

Comment avec RSA peut on à la fois envoyer un message **chiffré** et **signé** ?

Ex 3

1. Calculer $\phi(15)$ et $\phi(27)$
2. Vérifier que si $n = pq$ avec p et q premiers alors $\phi(n) = n(1 - \frac{1}{p})(1 - \frac{1}{q})$
3. On admettra que par extension $\phi(n) = n \prod_{p \text{ premier}|n} (1 - \frac{1}{p})$

   Calculer $\phi(45)$ puis $\phi(319)$

Ex 4 (Exponentiation modulaire rapide)

S'inspirer de l'exemple suivant pour proposer un algorithme puis un programme Python pour calculer "plus vite" a^b modulo n (récursif ou itératif)

On veut calculer a^7 par la méthode appelée "élévation récursive au carré"

$a^7 = a \times a^6 = a \times (a^3)^2 = a \times (a \times a^2)^2$

Il y a eu : 2 élévations au carré et 2 multiplications donc en tout 4 multiplications contre 6 si on procède de manière naïve

Ex 5

Centres étrangers 11 juin 2018

Le but de cet exercice est d'envisager une méthode de cryptage à clé publique d'une information numérique, appelée système RSA, en l'honneur des mathématiciens Ronald Rivest, Adi Shamir et Leonard Adleman, qui ont inventé cette méthode de cryptage en 1977 et l'ont publiée en 1978.

Les questions 1 et 2 sont des questions préparatoires, la question 3 aborde le cryptage, la question 4 le décryptage.

1. Cette question envisage de calculer le reste dans la division euclidienne par 55 de certaines puissances de l'entier 8.
   (a) Vérifier que $8^7 \equiv 2 \mod 55$.
   En déduire le reste dans la division euclidienne par 55 du nombre 8^{21}.
   (b) Vérifier que $8^2 \equiv 9 \mod 55$, puis déduire de la question **a.** le reste dans la division euclidienne par 55 de 8^{23}.
2. Dans cette question, on considère l'équation (E) $23x - 40y = 1$, dont les solutions sont des couples $(x\ ;\ y)$ d'entiers relatifs.

(a) Justifier le fait que l'équation (E) admet au moins un couple solution.

(b) Donner un couple, solution particulière de l'équation (E).

(c) Déterminer tous les couples d'entiers relatifs solutions de l'équation (E).

(d) En déduire qu'il existe un unique entier d vérifiant les conditions $0 \leqslant d < 40$ et $23d \equiv 1 \mod 40$.

3. Cryptage dans le système RSA

Une personne A choisit deux nombres premiers p et q, puis calcule les produits $N = pq$ et $n = (p-1)(q-1)$. Elle choisit également un entier naturel c premier avec n.

La personne A publie le couple $(N\ ;\ c)$, qui est une clé publique permettant à quiconque de lui envoyer un nombre crypté.

Les messages sont numérisés et transformés en une suite d'entiers compris entre 0 et $N-1$.

Pour crypter un entier a de cette suite, on procède ainsi : on calcule le reste b dans la division euclidienne par N du nombre a^c, et le nombre crypté est l'entier b.

Dans la pratique, cette méthode est sûre si la personne A choisit des nombres premiers p et q très grands, s'écrivant avec plusieurs dizaines de chiffres.

On va l'envisager ici avec des nombres plus simples : $p = 5$ et $q = 11$.

La personne A choisit également $c = 23$.

(a) Calculer les nombres N et n, puis justifier que la valeur de c vérifie la condition voulue.

(b) Un émetteur souhaite envoyer à la personne A le nombre $a = 8$.

Déterminer la valeur du nombre crypté b.

4. Décryptage dans le système RSA

La personne A calcule dans un premier temps l'unique entier naturel d vérifiant les conditions $0 \leqslant d < n$ et $cd \equiv 1 \mod n$.

Elle garde secret ce nombre d qui lui permet, et à elle seule, de décrypter les nombres qui lui ont été envoyés cryptés avec sa clé publique.

Pour décrypter un nombre crypté b, la personne A calcule le reste a dans la division euclidienne par N du nombre b^d, et le nombre en clair – c'est-à-dire le nombre avant cryptage – est le nombre a.

On admet l'existence et l'unicité de l'entier d, et le fait que le décryptage fonctionne.

Les nombres choisis par A sont encore $p = 5$, $q = 11$ et $c = 23$.

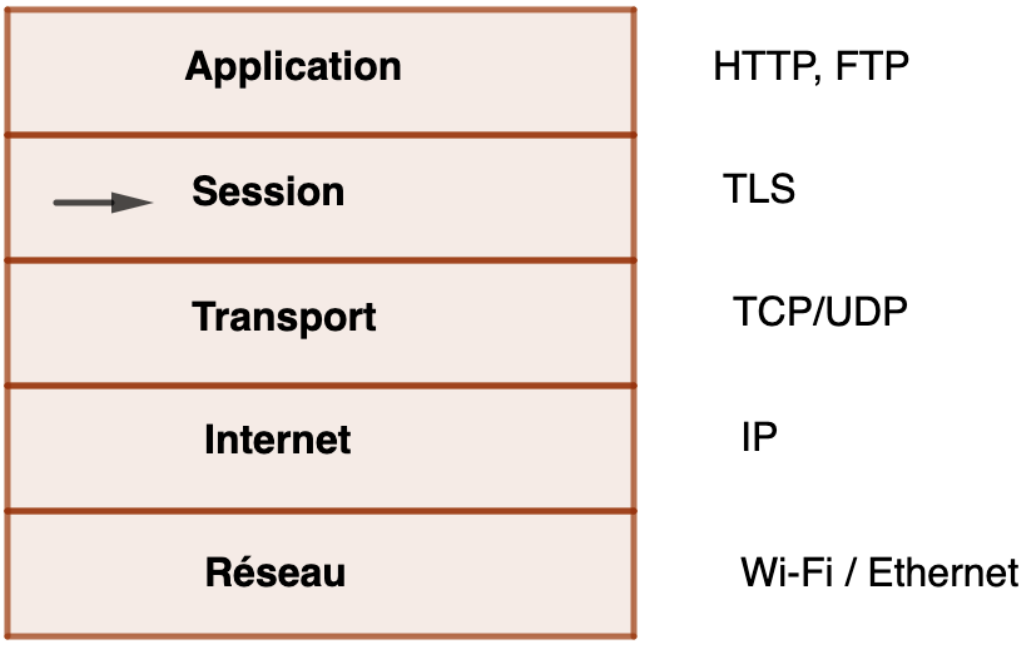
(a) Quelle est la valeur de d ?

(b) En appliquant la règle de décryptage, retrouver le nombre en clair lorsque le nombre crypté est $b = 17$.

4 Transport Security Layer

Nous avons vu le modèle en couches TCP/IP

On ajoute une couche supplémentaire entre la couche Application et la couche Transport. Cette couche s'appelle la couche **Session**. C'est à ce niveau qu'on ajoute un en-tête TLS (Transport Security Layer)



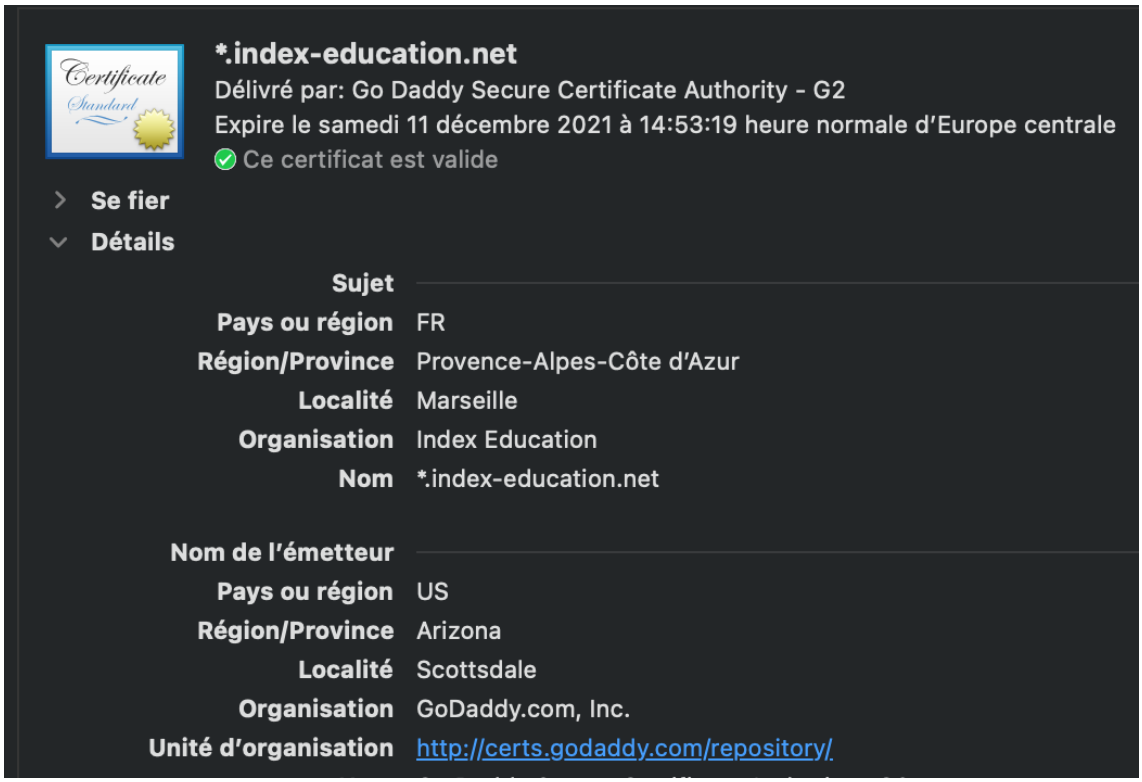
Comment TLS se met en place ?

On parle du TLS handshake, et on peut le voir grâce à un logiciel de capture de paquets comme Wireshark

Time	Source	Destination	Protocol	Length	Info
153.342734	192.168.0.40	87.248.100.168	TLSv1…	583	Client Hello
153.364134	87.248.100.168	192.168.0.40	TCP	66	443 → 52006 [ACK] Seq=1 Ack=518 Win=30336 Len=0 TSva
153.364918	87.248.100.168	192.168.0.40	TLSv1…	1514	Server Hello, Change Cipher Spec, Application Data
153.364983	192.168.0.40	87.248.100.168	TCP	66	52006 → 443 [ACK] Seq=518 Ack=1449 Win=130304 Len=0
153.365913	87.248.100.168	192.168.0.40	TCP	1514	443 → 52006 [ACK] Seq=1449 Ack=518 Win=30336 Len=144
153.365919	87.248.100.168	192.168.0.40	TLSv1…	814	Application Data, Application Data, Application Data
153.365996	192.168.0.40	87.248.100.168	TCP	66	52006 → 443 [ACK] Seq=518 Ack=3645 Win=128832 Len=0
153.371315	192.168.0.40	87.248.100.168	TLSv1…	130	Change Cipher Spec, Application Data
153.372024	192.168.0.40	87.248.100.168	TLSv1…	910	Application Data
153.392200	87.248.100.168	192.168.0.40	TCP	66	443 → 52006 [ACK] Seq=3645 Ack=1426 Win=32000 Len=0
153.393105	87.248.100.168	192.168.0.40	TLSv1…	369	Application Data
153.393108	87.248.100.168	192.168.0.40	TLSv1…	369	Application Data
153.393268	192.168.0.40	87.248.100.168	TCP	66	52006 → 443 [ACK] Seq=1426 Ack=4251 Win=130432 Len=0
153.393453	87.248.100.168	192.168.0.40	TLSv1…	718	Application Data
153.393457	87.248.100.168	192.168.0.40	TLSv1…	804	Application Data

1. Etape 1 : Client Hello : Le client envoie un certain nombre d'informations au serveur en vue de réaliser un échange d'informations cryptées
2. Etape 2 : Serveur Hello : Le serveur répond (en bleu foncé) à ce stade le serveur envoie au client son certificat d'authentification. Voici l'exemple du certificat du webmail de l'académie de Versailles (on peut voir le certificat en cliquant sur le cadenas de la barre URL) :

Voici celui de Pronote :



Il envoie aussi un certain nombre d'informations sur les techniques de chiffrement

Voici ci-dessous les informations envoyées par le serveur d'un hébergeur de site Web à un client avant une communication cryptée

Protocole :	TLS1.2	Chiffrement :	AES-256-GCM
Échange de clé :	ECDHE-SECP256R1-RSA-SHA384	Mac :	AEAD

3. Etape 3 : Il y a un autre aller-retour entre le client et le serveur pour finaliser le contrat de communication cryptée entre eux (**échange de clé (Diffie-Hellmann)** et autres informations)

   Voici une copie d'écran du RFC 2246 de l'I.E.T.F concernant la version 1 de TLS (1999)

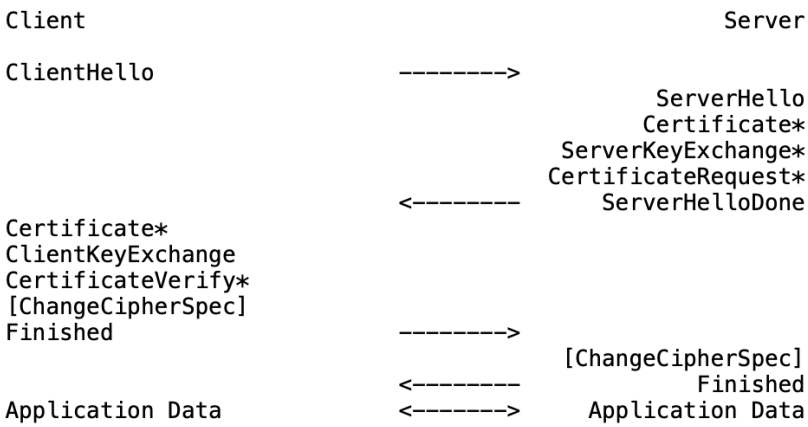


Fig. 1 – Message flow for a full handshake