Université de Guelma
Département Informatique

جامعة 8 ماي 1945 قالمة
UNIVERSITE 8 MAI 1945 GUELMA

Chapitre 2 : Les protocoles de sécurité et la cryptographie

Cours - Sécurité Informatique

3 année LMD SI & ISIL

Par : Dr. M. A. Ferrag

Plan du cours

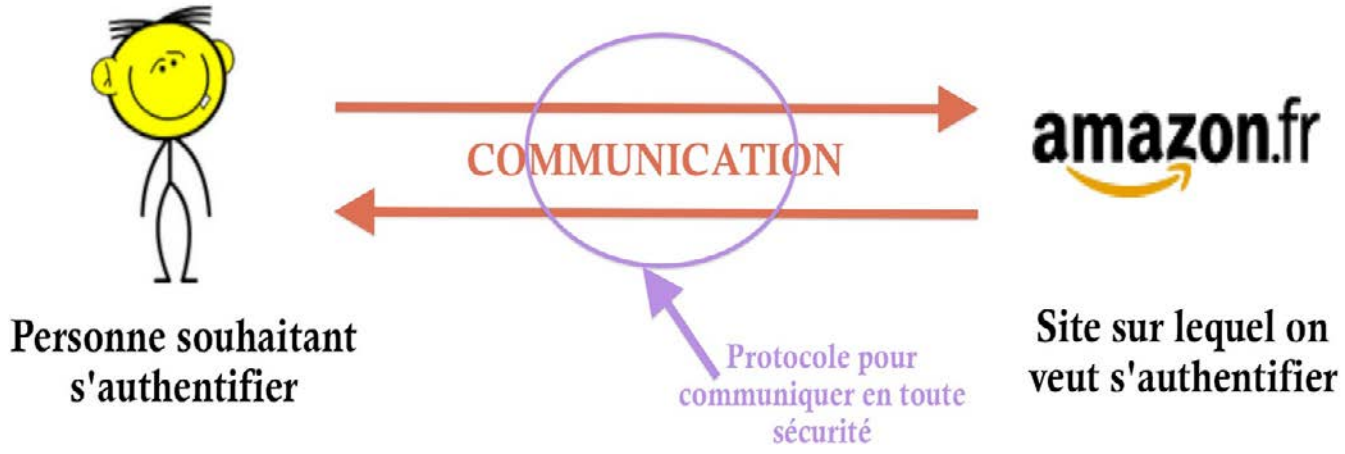
- **Initiation à la cryptographie**
- **RADIUS**
- **Diameter**
- **EAP**

Pourquoi les protocoles de sécurité ?

- **Sur internet (paiement en ligne, envoi de mots de passe, ...), ou encore quand on utilise une carte bancaire, on a besoin de :**
    - Établir une communication sécurisée entre 2 individus - **Sécurité**
    - Être sûr de communiquer avec la bonne personne, et pas un intrus voulant voler des informations (comme le mot de passe) - **Authentification**
    - Être sûr que les données ne sont pas modifiées en cours de route - **Intégrité**

Qu’est-ce qu’un protocole de sécurité ?

- Ensemble de règles régissant le comportement d'individus pour répondre aux besoins d’une application (paiement en ligne, vote électronique, authentification d’individus, etc)

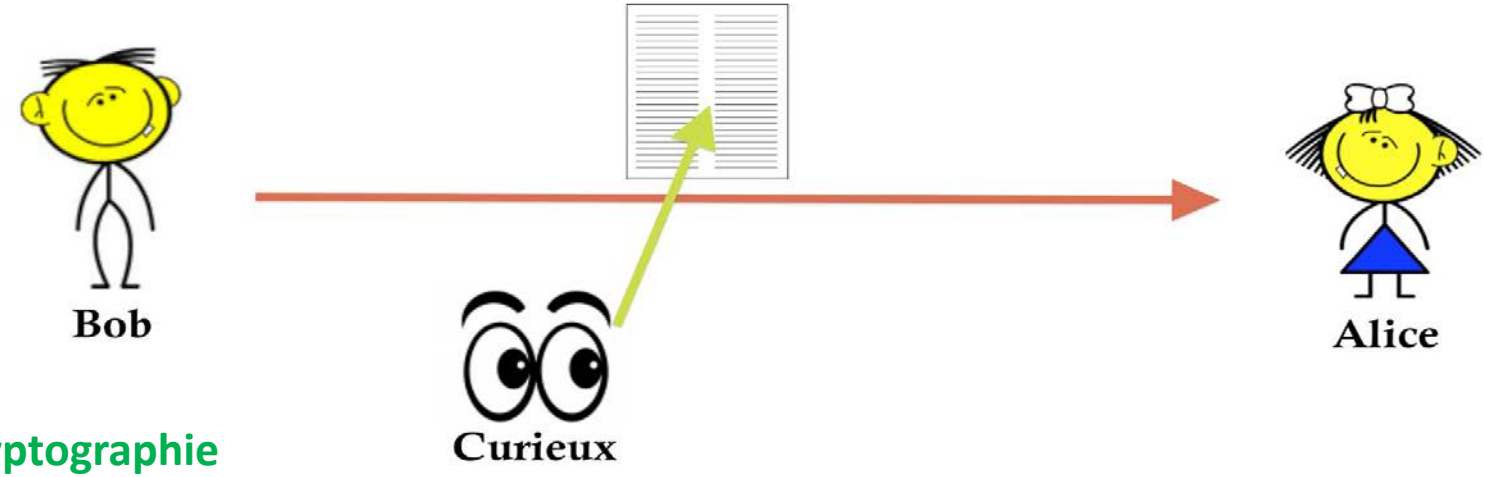


L'utilisation des protocoles est transparente pour l'utilisateur

Sécuriser les messages

- Communication entre 2 individus A et B --- **échange de messages ….**Besoin que ces messages soient chiffrés pour garantir leur confidentialité
- Exemple : Durant leurs cours, Alice et Bob, qui ne sont pas côte à côte dans la classe, communiquent en se faisant passer des petits mots.

Problème : n'importe qui peut lire le mot...



Utilisation de la cryptographie

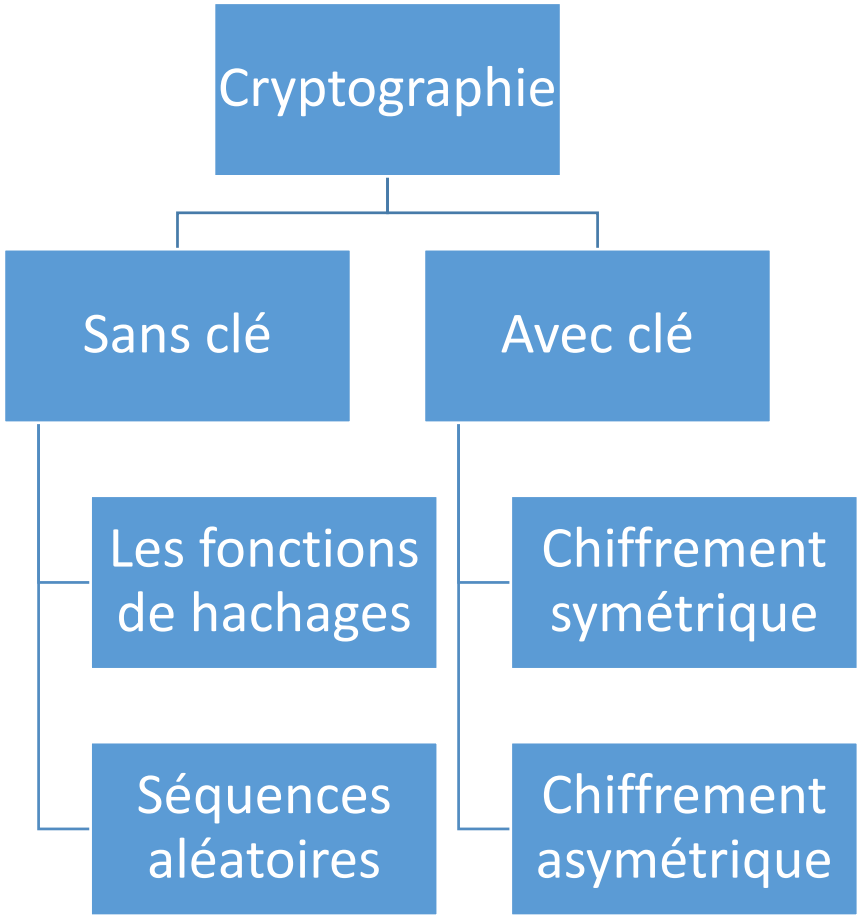
La cryptographie (1)

- Qu’est-ce que la cryptographie ? Un ensemble de méthodes permettant de chiffrer un message numérique, grâce à une clé.

Rend le message incompréhensible pour quiconque ne possédant pas la clé.

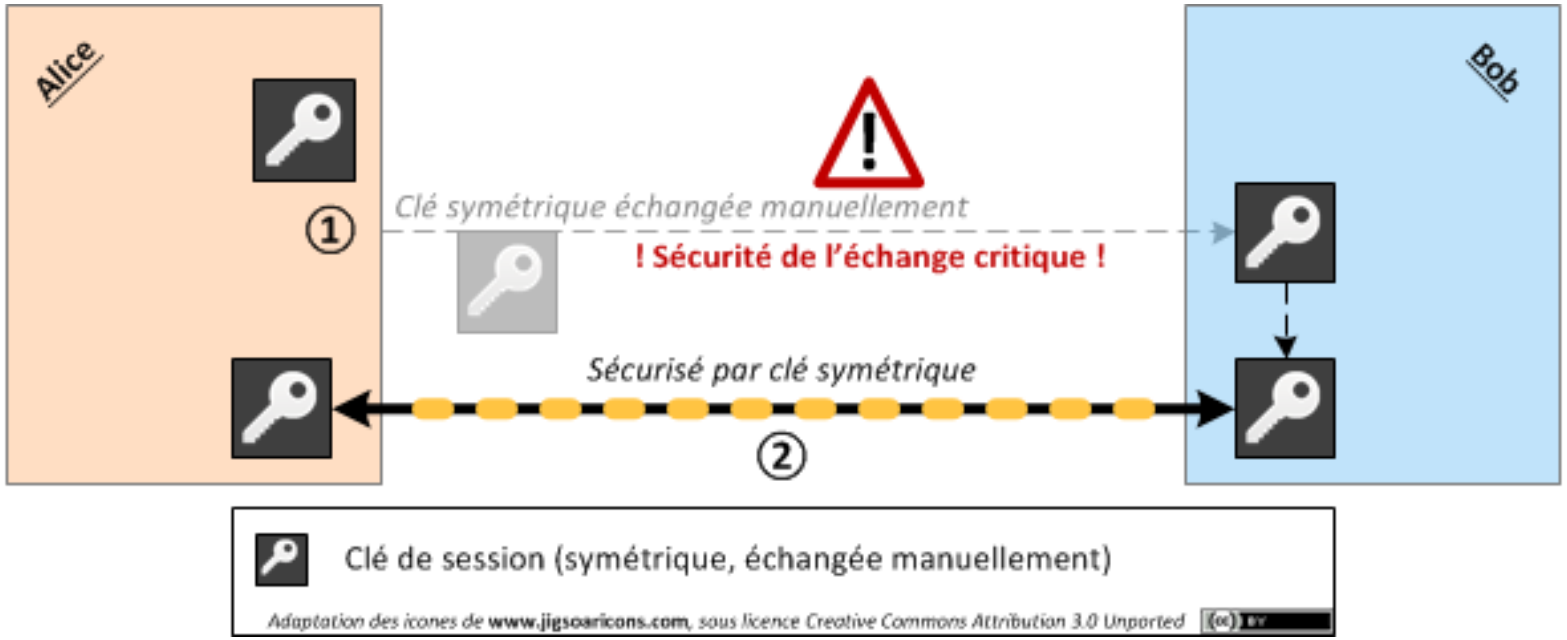
- Chiffrement symétrique : une seule clé partagée pour chiffrer et déchiffrer
- Chiffrement asymétrique : une clé pour chiffrer, une autre pour déchiffrer

La cryptographie (2)



- **Chiffrement symétrique :** chiffrement plus rapide, mais nécessite de se "rencontrer" pour pourvoir s'échanger la clé commune.
- **Chiffrement asymétrique :** algorithmes de cryptages plus complexes, donc plus lent, mais communication sans échange préalable de clé.
- Les fonctions de hachage sont généralement utilisées pour garantir l'intégrité.

Chiffrement symétrique



La clé doit être échangée à un moment donné. Et cet échange peut être intercepté, rendant le chiffrement inutile…
La clé est définie par un des participants, et n'est pas renouvelée automatiquement, la rendant plus vulnérable à des attaques par dictionnaire, par exemple.

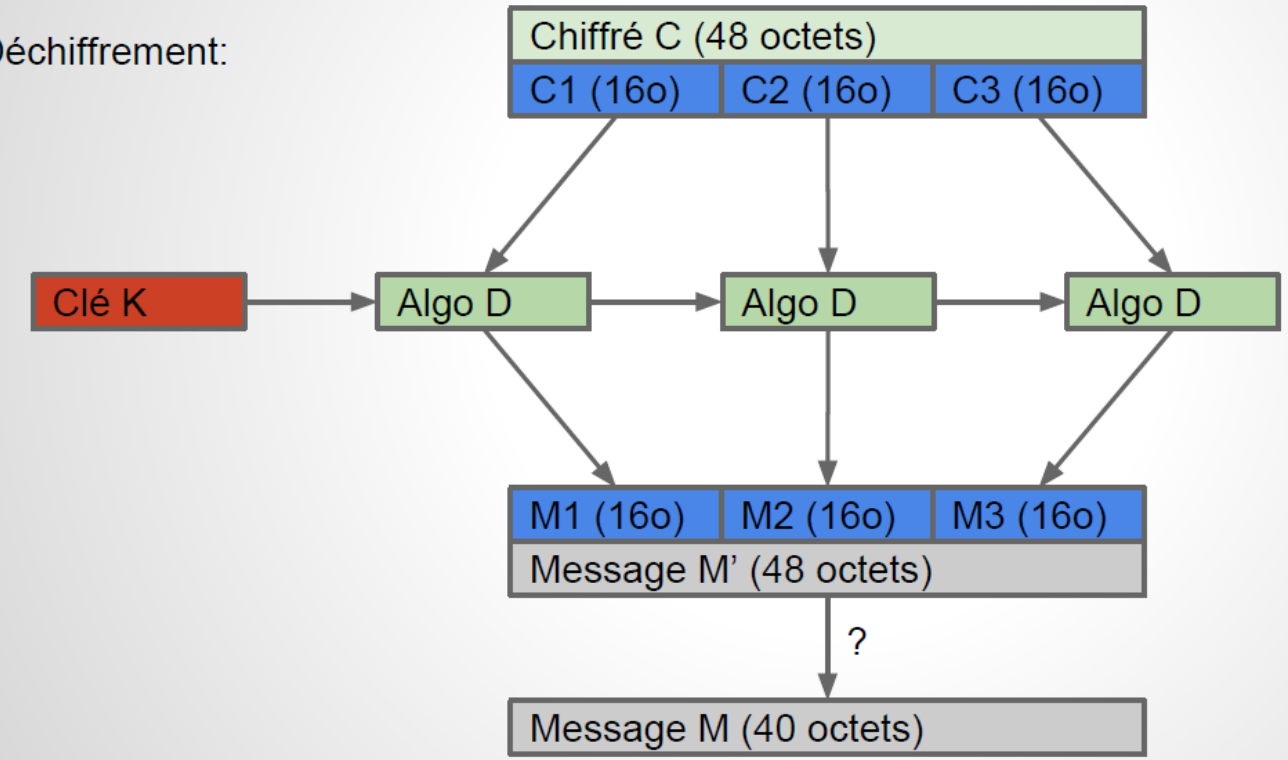
Source : https://www.nexcom.fr/

Par blocs / par flot

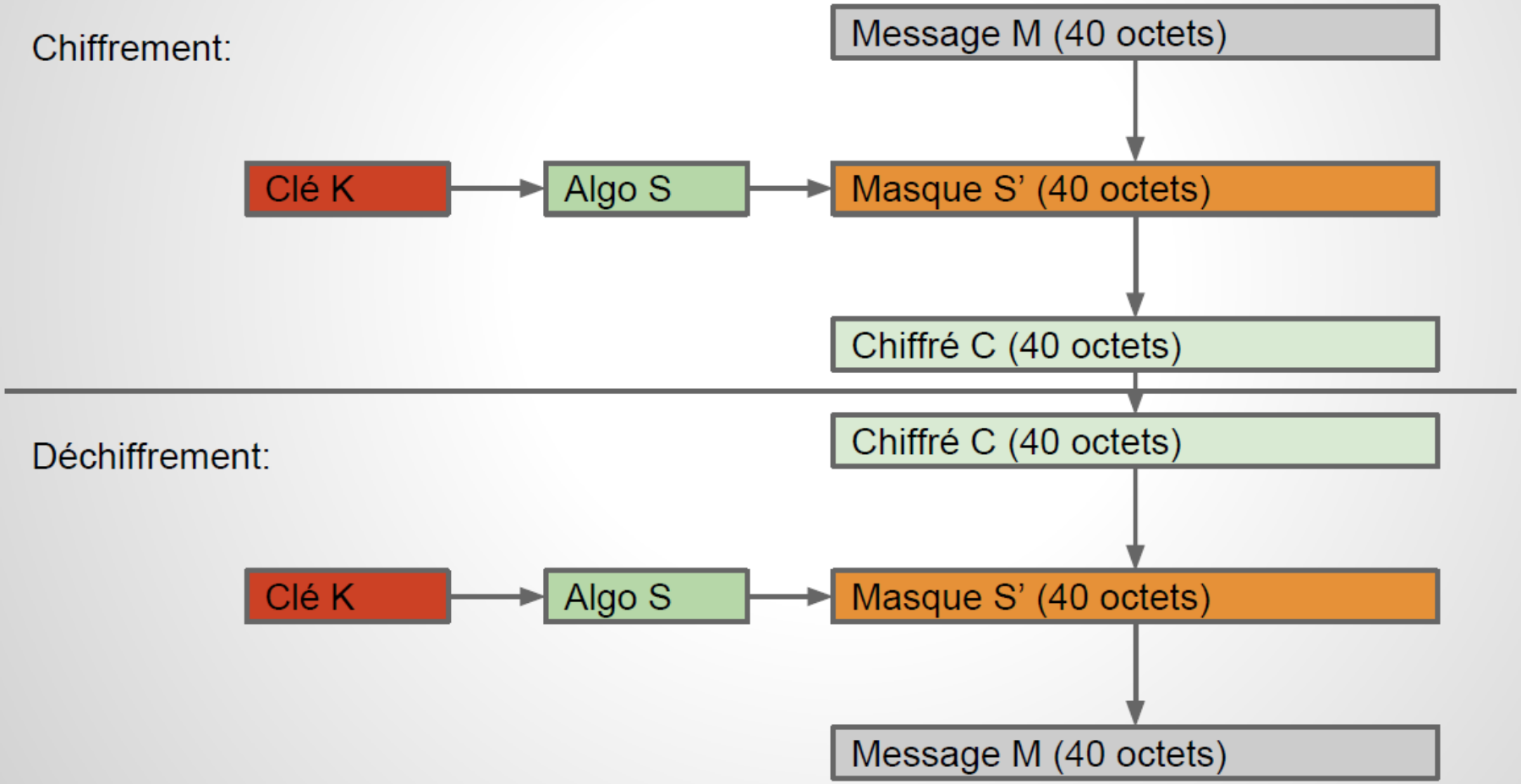
On distingue deux types d'algorithmes de chiffrement symétrique: par blocs ou par flot.

- Le chiffrement par bloc permet de travailler sur des blocs de taille fixée (le plus souvent 16 octets). On traite un message long comme une succession de blocs. Ces algorithmes nécessitent en général d'ajouter du bourrage lorsque le message initial n'est pas un multiple de la taille de bloc.
- Le chiffrement par flot permet de travailler sur un message de taille arbitraire, et le traite octet par octet (voir bit par bit). Ces algorithmes sont généralement plus rapide mais moins résistants que les chiffrements par blocs.

Par blocs / par flot



Par blocs / par flot



Quelques exemples

Chiffrement par bloc (**block cipher**)

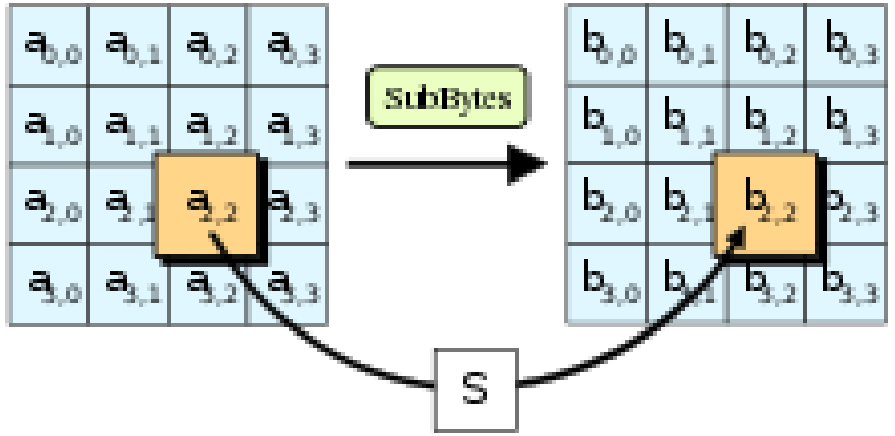
- DES (Data Encryption Standard): bloc=64bits, clé=56bits
- AES (Advanced Encryption Standard): bloc=128bits, clé=128/192/256bits
- Blowfish: bloc=64bits, clé=32..448bits

Chiffrement par flot (ou chiffrement en continu, **stream cipher**)

- A5/1 (chiffrement GSM): clé=64bits (seulement 54 pour le GSM)
- RC4 (WEP): clé variable
- E0 (Bluetooth): clé=128bits en général

Liste d'algorithme symétrique (AES)

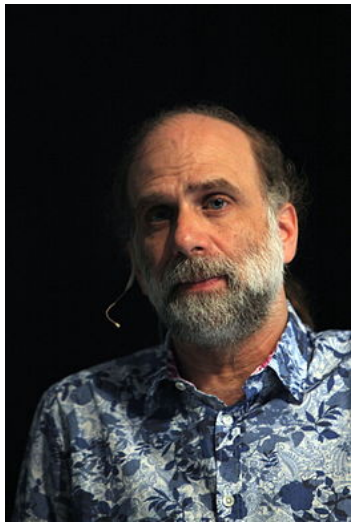
- Advanced Encryption Standard ou AES, aussi connu sous le nom de Rijndael, est un algorithme de chiffrement symétrique.
- Il remporta en octobre 2000 le concours AES, lancé en 1997 par le NIST et devint le nouveau standard de chiffrement pour les organisations du gouvernement des États-Unis.
- Il a été approuvé par la NSA (National Security Agency) dans sa liste des algorithmes cryptographiques. Il est actuellement le plus utilisé et le plus sûr.
- L'algorithme prend en entrée un bloc de 128 bits (16 octets), la clé fait 128, 192 ou 256 bits. Les 16 octets en entrée sont permutés selon une table définie au préalable. Ces octets sont ensuite placés dans une matrice de 4x4 éléments et ses lignes subissent une rotation vers la droite.



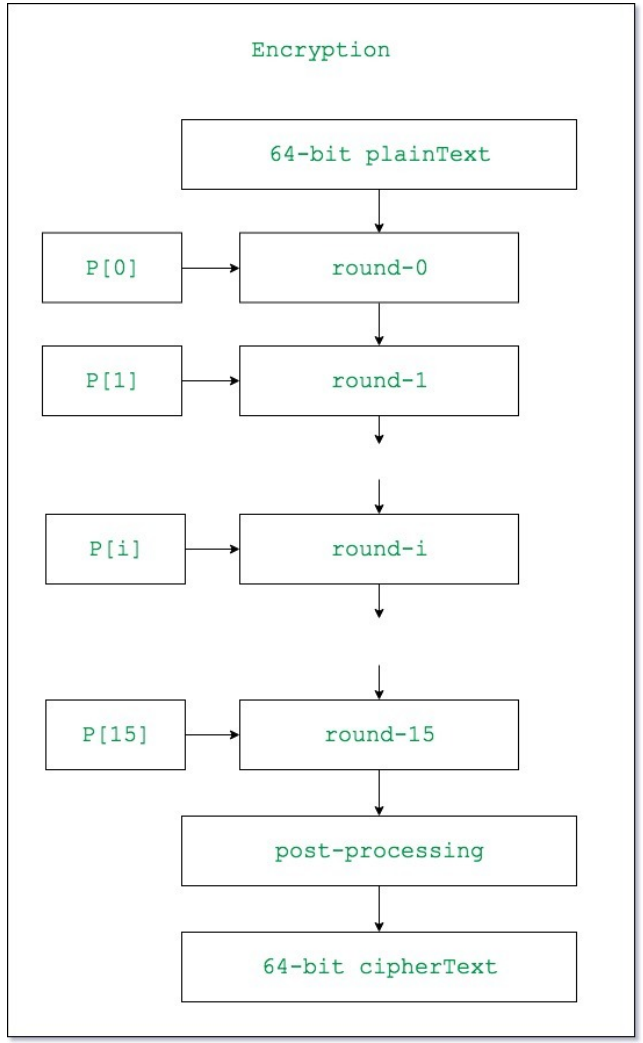
Vincent Rijmen (un cryptologue belge)

Liste d'algorithme symétrique (Blowfish)

- Blowfish utilise une taille de bloc de 64 bits et la clé, de longueur variable, peut aller de 32 à 448 bits avec 16 permutation.

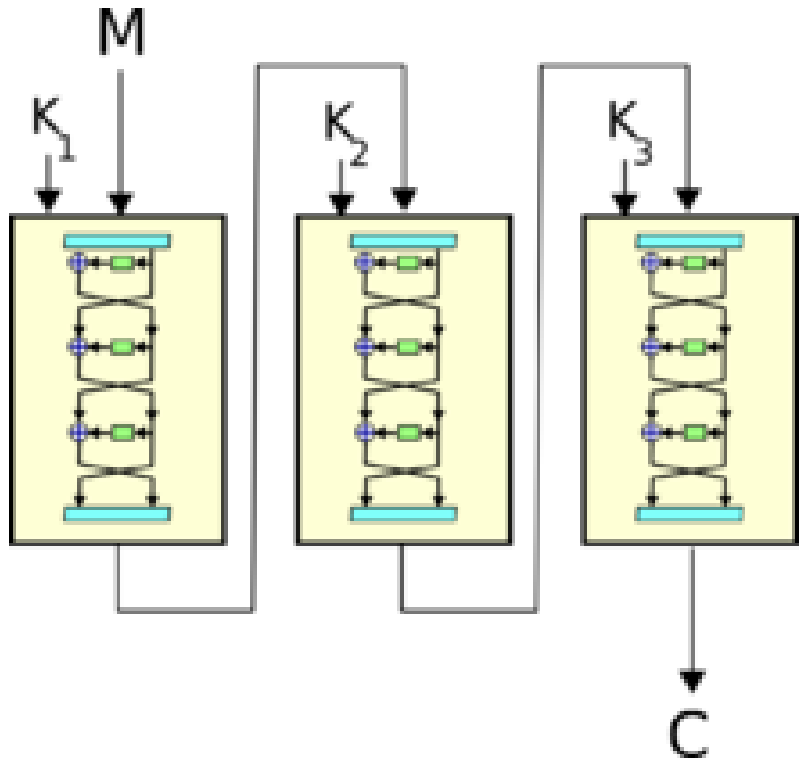


Bruce Schneier
Un cryptologue américain



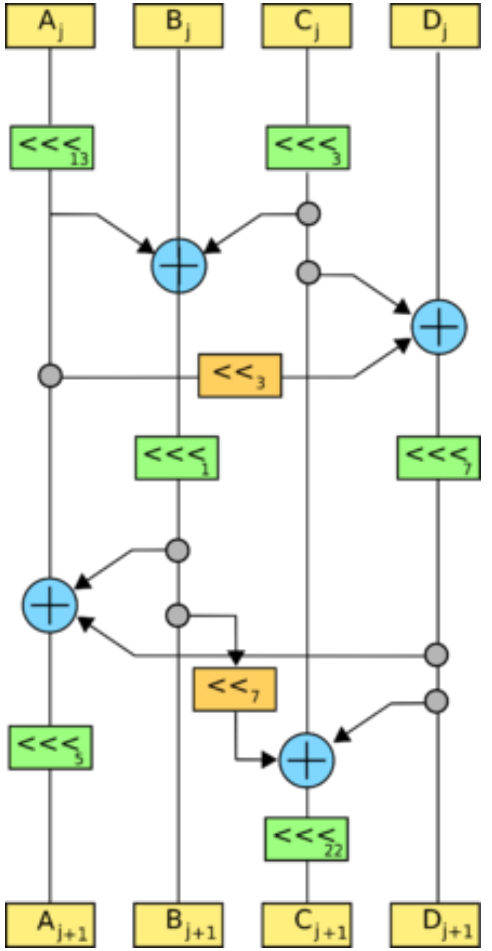
Liste d'algorithme symétrique (DES)

- Le Data Encryption Standard (DES) est un algorithme de chiffrement symétrique (chiffrement par bloc) utilisant des clés de 56 bits.
- Son emploi n'est plus recommandé aujourd'hui, du fait de sa lenteur à l'exécution et de son espace de clés trop petit.
- Quand il est encore utilisé c'est généralement en Triple DES, ce qui ne fait rien pour améliorer ses performances.
- DES a notamment été utilisé dans le système de mots de passe UNIX.
- L'algorithme initialement conçu par IBM utilisait une clé de 112 bits. L'intervention de la NSA a ramené la taille de clé à 56 bits.
- De nos jours, le Triple DES reste très répandu, et le DES « simple » ne subsiste que dans d'anciennes applications. Le standard DES a été remplacé en 2001 par l'AES (Advanced Encryption Standard).



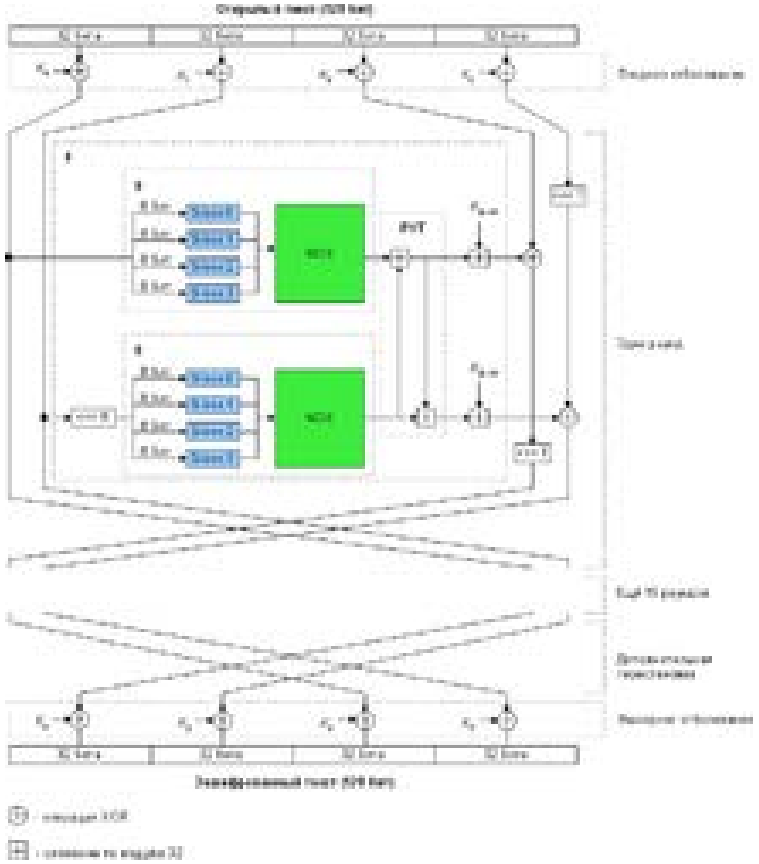
Liste d'algorithme symétrique (Serpent)

- Serpent est un algorithme de chiffrement par bloc finaliste pour le concours AES.
- Serpent a été conçu par Ross J. Anderson, Eli Biham et Lars Knudsen.
- Tout comme les autres candidats pour AES, Serpent a une taille de bloc de 128 bits et supporte des clés de 128, 192 ou 256 bits, mais également d'autres longueurs inférieures (multiple de 8 bits).
- L'algorithme comporte 32 tours d'un réseau de substitution-permutation opérant sur quatre mots de 32 bits.
- Chaque tour utilise 32 copies de la même S-Box de 16x16 éléments, il y a 8 S-Boxes en tout qui sont utilisées chacune tous les 8 tours.

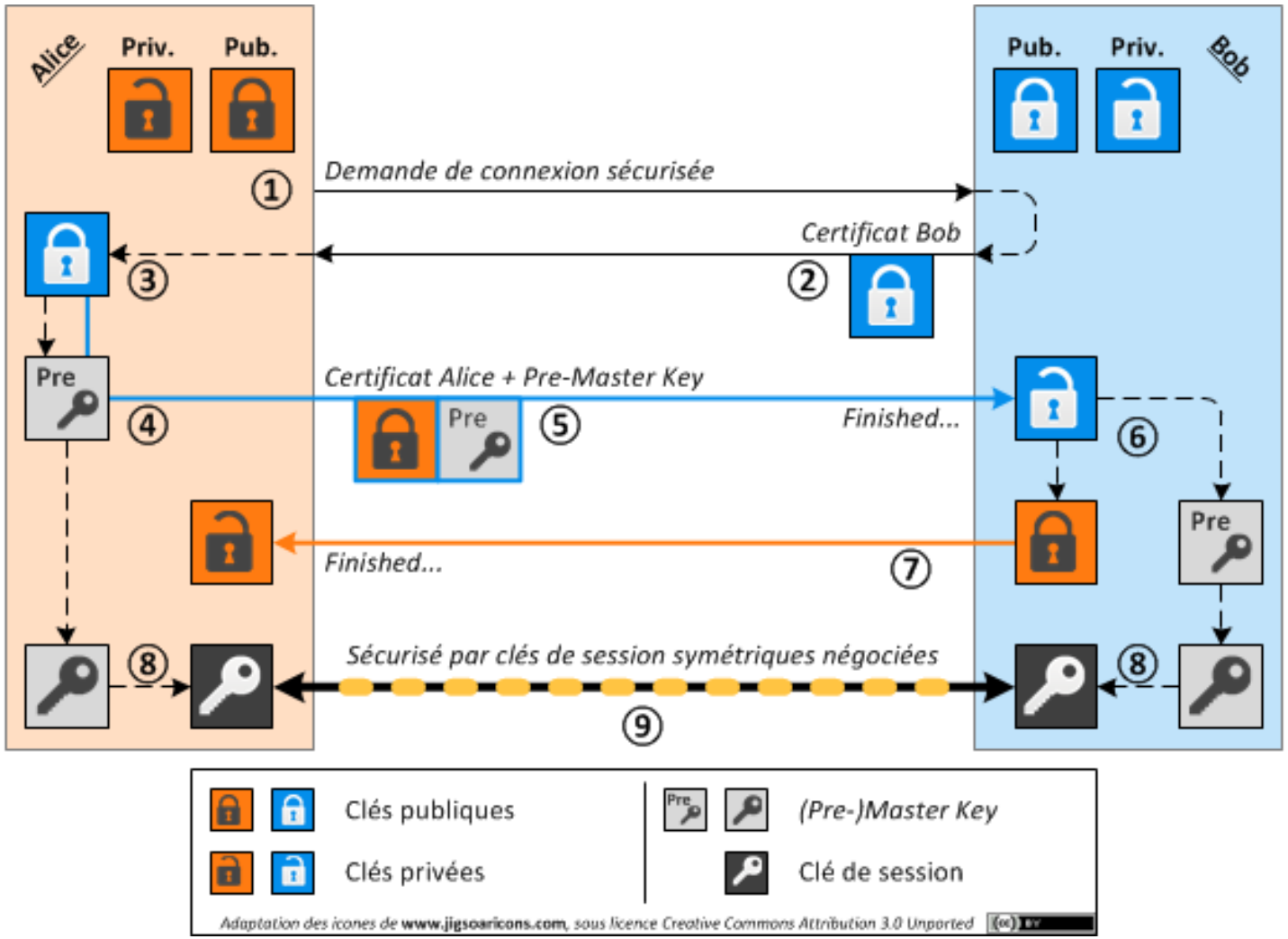


Liste d'algorithme symétrique (Twofish)

- Twofish est un algorithme de chiffrement symétrique par bloc inventé et analysé par Bruce Schneier, Niels Ferguson, John Kelsey, Doug Whiting, David Wagner et Chris Hall.
- Il chiffre des blocs de 128 bits avec une clé de 128, 192 ou 256 bits.
- Twofish était l'un des cinq finalistes du concours AES mais il n'a pas été sélectionné pour le standard. Il reprend en partie des concepts présents dans le populaire Blowfish, du même auteur.



Chiffrement asymétrique et hybride



1- Alice demande une connexion sécurisée avec Bob.
2- Bob transmet, de manière non sécurisée, son certificat, à savoir sa clé publique.
3- Alice récupère la clé publique de Bob et authentifie celui-ci. La connexion est refusée si cette identité ne peut être vérifiée.
4- Alice génère une Pre-Master Key. Cette clé ainsi que la clé publique d'Alice sont transmises à Bob. Le secret de la Pre-Master Key doit être préservé pour assurer l'efficacité de la méthode.
5- Alice possédant la clé publique de Bob, elle l'utilise pour chiffrer son message.
6- A la réception du message chiffré, Bob utilise sa clé privée pour déchiffrer le message, chiffré avec sa clé publique par Alice. Il en extrait deux éléments : la clé publique d'Alice et, encore plus important, la Pre-Master Key.
7- Bob authentifie Alice à l'aide de sa clé publique. Si l'identité est vérifiée correctement, la négociation est terminée. Ce message peut être transmis de manière sécurisée grâce à la clé publique d'Alice. Celle-ci utilise alors sa clé privée pour décoder le message.
8- Dans le même temps, les deux parties génèrent la même clé maître (Master Key) finale via des procédés cryptographiques. De cette clé, ils peuvent en déduire une clé de session identique (et donc symétrique). Cette clé est régénérée régulièrement afin d'éviter les problèmes inhérents aux clés symétriques.
9- Cette clé de session symétrique permet de chiffrer efficacement le trafic entre les deux entités, sans la surcharge imprimée par un chiffrement asymétrique.

Source : https://www.nexcom.fr/

Liste d'algorithme asymétrique

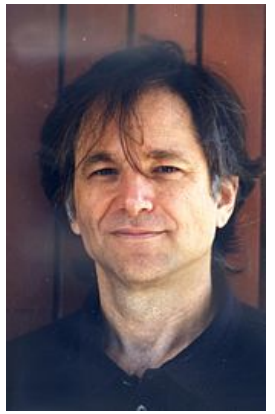
- RSA
- Elliptic Curve Cryptography
- ElGamal

Le chiffrement asymétrique est plus lent que le chiffrement symétrique, mais plus résistant. Il ne dépend pas d'une unique clé secrète, mais d'une paire de clés (privée/publique). Pour chiffrer il est nécessaire de connaître la clé publique, alors que la personne voulant déchiffrer doit connaître la clé privée.

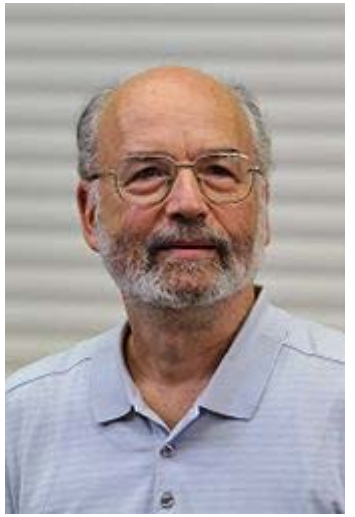
Liste d'algorithme asymétrique (RSA)

- Le chiffrement RSA (nommé par les initiales de ses trois inventeurs) est un algorithme de cryptographie asymétrique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet.
- il utilise une paire de clés (des nombres entiers) composée d'une clé publique pour chiffrer et d'une clé privée pour déchiffrer des données confidentielles.

Ronald Rivest



Leonard Adleman



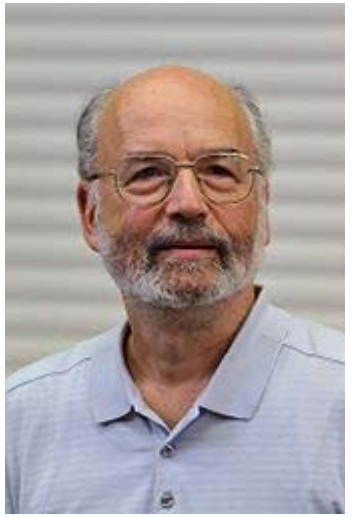
Adi Shamir

Liste d'algorithme asymétrique (RSA)
1. Création des clés

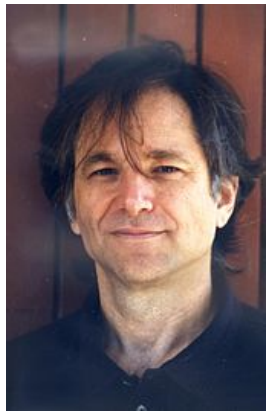
- L'étape de création des clés est à la charge d'Alice. Elle n'intervient pas à chaque chiffrement car les clés peuvent être réutilisées. La difficulté première, que ne règle pas le chiffrement, est que Bob soit bien certain que la clé publique qu'il détient est celle d'Alice. Le renouvellement des clés n'intervient que si la clé privée est compromise, ou par précaution au bout d'un certain temps (qui peut se compter en années).

- Choisir p et q, deux nombres premiers distincts ;
- calculer leur produit n = pq, appelé module de chiffrement ;
- calculer $\phi(n) = (p - 1)(q - 1)$ (c'est la valeur de l'indicatrice d'Euler en n) ;
- choisir un entier naturel e premier avec $\phi(n)$ et strictement inférieur à $\phi(n)$, appelé exposant de chiffrement ;
- calculer l'entier naturel d, inverse de e modulo $\phi(n)$, et strictement inférieur à $\phi(n)$, appelé exposant de déchiffrement ; d peut se calculer efficacement par l'algorithme d'Euclide étendu.

Ronald Rivest



Adi Shamir



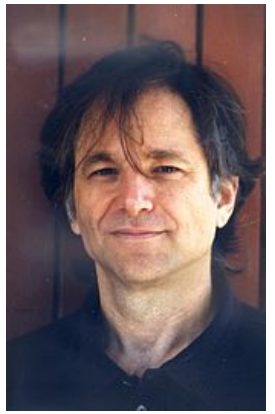
Leonard Adleman

Liste d'algorithme asymétrique (RSA)
2. Chiffrement du message

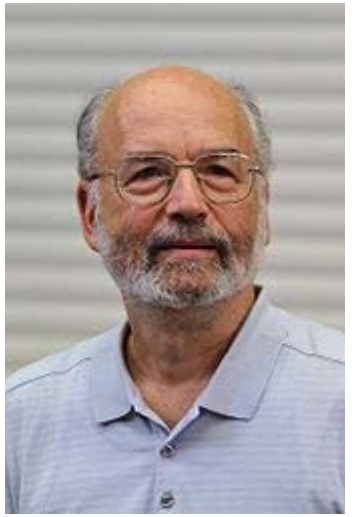
- Si M est un entier naturel strictement inférieur à n représentant un message, alors le message chiffré sera représenté par

$$C \equiv M^e \pmod{n}$$

Ronald Rivest



Leonard Adleman



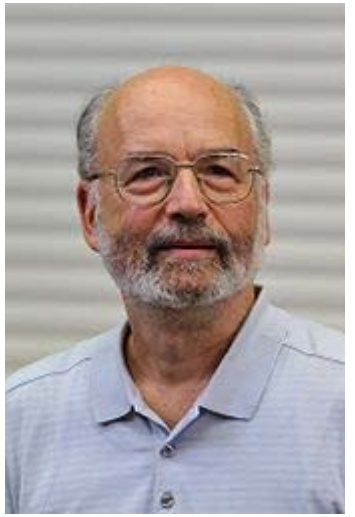
Adi Shamir

Liste d'algorithme asymétrique (RSA)
3. Déchiffrement du message

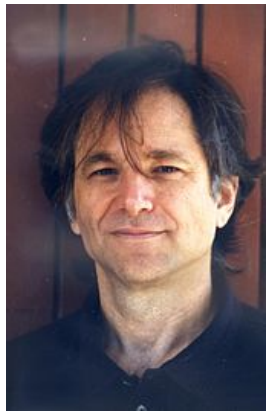
- Pour déchiffrer C, on utilise d, l'inverse de e modulo (p – 1)(q – 1), et l'on retrouve le message clair M par

$$M \equiv C^d \pmod{n}$$

Ronald Rivest



Adi Shamir



Leonard Adleman

Liste d'algorithme asymétrique (RSA)

- Comme e est premier avec $\phi(n)$, d'après le théorème de Bachet-Bézout il existe deux entiers d et k tels que $ed = 1 + k\phi(n)$, c'est-à-dire que $ed \equiv 1 \pmod{\phi(n)}$: e est bien inversible modulo $\phi(n)$.
- Le couple (n, e) — ou (e, n) — est la *clé publique* du chiffrement, alors que sa *clé privée* est le nombre d, sachant que l'opération de déchiffrement ne demande que la clef privée d et l'entier n, connu par la clé publique (la clé privée est parfois aussi définie comme le couple (d, n) ou le triplet (p, q, d)).

Liste d'algorithme asymétrique (RSA)

- **Théorème de Bachet-Bézout**

En mathématiques, et plus précisément en arithmétique élémentaire, le théorème de Bachet-Bézout ou identité de Bézout est un résultat d'arithmétique élémentaire, qui prouve l'existence de solutions à l'équation diophantienne linéaire :

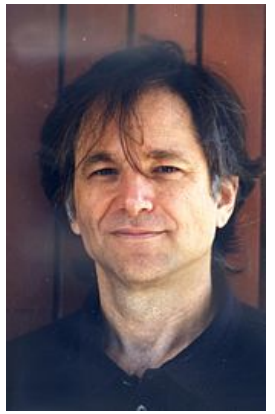
$$ax + by = pgcd(a, b)$$

d'inconnues x et y entiers relatifs, où a et b sont des coefficients entiers relatifs et où pgcd(a, b) est le plus grand commun diviseur de a et b. Le théorème de Bézout affirme que les entiers a et b sont premiers entre eux si et seulement si l'équation ax + by = 1 admet des solutions.

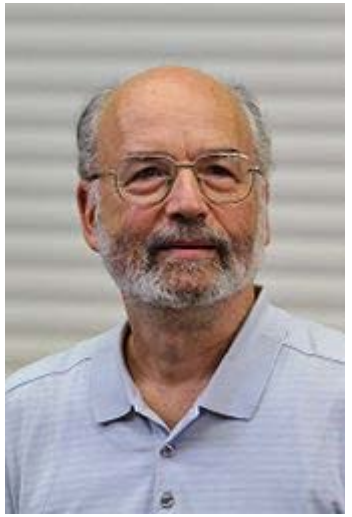
Liste d'algorithme asymétrique (RSA) Exemple

- Un exemple avec de petits nombres premiers (en pratique il faut de très grands nombres premiers) :
- on choisit deux nombres premiers $p = 3$, $q = 11$;
- leur produit $n = 3 \times 11 = 33$ est le module de chiffrement ;
- $\phi(n) = (3 - 1) \times (11 - 1) = 2 \times 10 = 20$;
- on choisit $e = 3$ (premier avec 20) comme exposant de chiffrement ;
- l'exposant de déchiffrement est $d = 7$, l'inverse de 3 modulo 20 (en effet $ed = 3 \times 7 \equiv 1 \bmod 20$).
- La clé publique d'Alice est $(n, e) = (33, 3)$, et sa clé privée est $(n, d) = (33, 7)$. Bob transmet un message à Alice.
- Chiffrement de $M = 4$ par Bob avec la *clé publique* d'Alice : $4^3 \equiv 31 \bmod 33$, le chiffré est $C = 31$ que Bob transmet à Alice ;
- Déchiffrement de $C = 31$ par Alice avec sa *clé privée* : $31^7 \equiv 4 \bmod 33$, Alice retrouve le message initial $M = 4$.

Ronald Rivest



Leonard Adleman



Adi Shamir

Liste d'algorithme asymétrique (RSA) Attaques

- **Attaque de Wiener**

L'attaque de Wiener (1989) est exploitable si l'exposant secret d est inférieur à 1/3 * N^1/4

- **Attaque de Håstad**

L'attaque de Håstad, l'une des premières attaques découvertes (en 1985), repose sur la possibilité que l'exposant public e soit suffisamment petit. En interceptant le même message envoyé à au moins e destinataires différents, il est possible de retrouver le message originel à l'aide du théorème des restes chinois.

- **Attaque par chronométrage (*timing attacks*)**

Paul Kocher a décrit en 1995 une nouvelle attaque contre RSA : en supposant que l'attaquante Ève en connaisse suffisamment sur les documents d'Alice et soit capable de mesurer les temps de déchiffrement de plusieurs documents chiffrés, elle serait en mesure d'en déduire rapidement la clef de déchiffrement. Il en irait de même pour la signature.

- **Attaque à chiffrés choisis (*Adaptive chosen ciphertext attacks*)**

En 1998, Daniel Bleichenbacher décrit la première attaque pratique de type « chiffré choisi adaptable » contre des messages RSA

Cryptographie à clé mixte

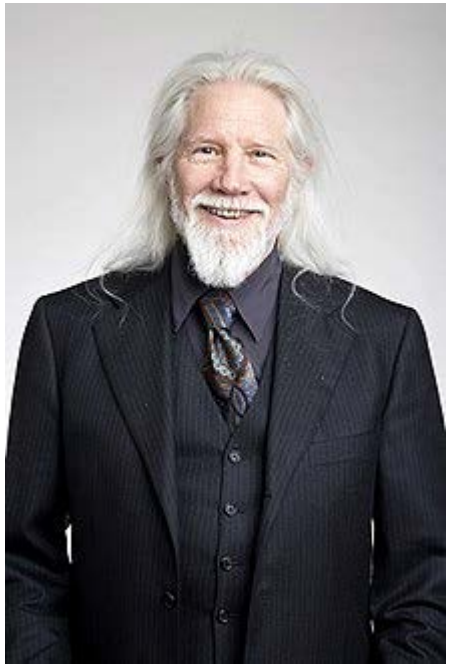
- Fait appel aux 2 techniques à clé symétrique et à clé publique et combine les avantages des deux tout en évitant leurs inconvénients.
- Le principe général de la cryptographie à clé mixte
  - chiffrement des données avec des clés symétriques
  - envoi de la clé symétrique par un algorithme à clé publique.

Cryptographie à clé mixte

- Principe des clés de session :
  - Génération aléatoire d'une clé de session de taille raisonnable.
  - Chiffrement de cette clé à l'aide de la clé publique du destinataire. (Le destinataire est en mesure de déchiffrer la clé de session à l'aide de sa clé privée.)
  - Ainsi, expéditeur et destinataire sont en possession d'une clé́ commune dont ils sont seuls connaisseurs.
  - Il leur est alors possible de s'envoyer des documents chiffrés à l'aide d'un algorithme de chiffrement symétrique.

Cryptographie à clé mixte

- Algorithme de Diffie-Hellman
  - Mis au point en 1976 afin de permettre l'échange de clés à travers un canal non sécurisé.
  - Cet algorithme est sensible à l'attaque « *Man in the middle* »
    - RFC 2631 - Diffie-Hellman Key Agreement Method
    - RFC 2875 - Diffie-Hellman Proof-of-Possession Algorithms



Whitfield 'Whit' Diffie

New Directions in Cryptography

Invited Paper

WHITFIELD DIFFIE AND MARTIN E. HELLMAN, MEMBER, IEEE

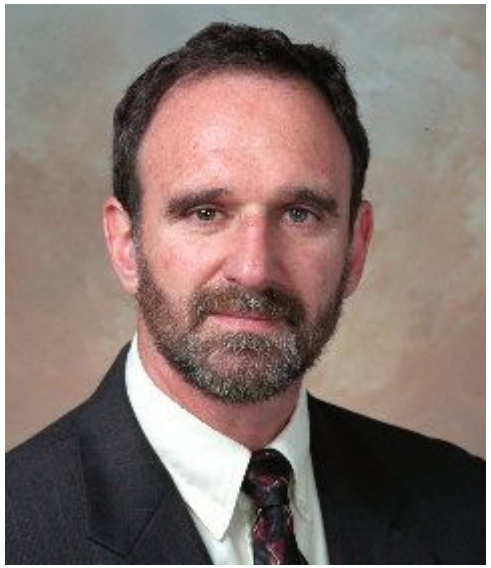
***Abstract*—Two kinds of contemporary developments in cryptography are examined. Widening applications of teleprocessing have given rise to a need for new types of cryptographic systems, which minimize the need for secure key distribution channels and supply the equivalent of a written signature. This paper suggests ways to solve these currently open problems. It also discusses how the theories of communication and computation are beginning to provide the tools to solve cryptographic problems of long standing.**

I. Introduction

WE STAND TODAY on the brink of a revolution in cryptography. The development of cheap digital hardware has freed it from the design limitations of mechanical computing and brought the cost of high grade

The best known cryptographic problem is that of privacy: preventing the unauthorized extraction of information from communications over an insecure channel. In order to use cryptography to insure privacy, however, it is currently necessary for the communicating parties to share a key which is known to no one else. This is done by sending the key in advance over some secure channel such as private courier or registered mail. A private conversation between two people with no prior acquaintance is a common occurrence in business, however, and it is unrealistic to expect initial business contacts to be postponed long enough for keys to be transmitted by some physical means. The cost and delay imposed by this key distribution problem is a major barrier to the transfer of business communications to large teleprocessing networks.

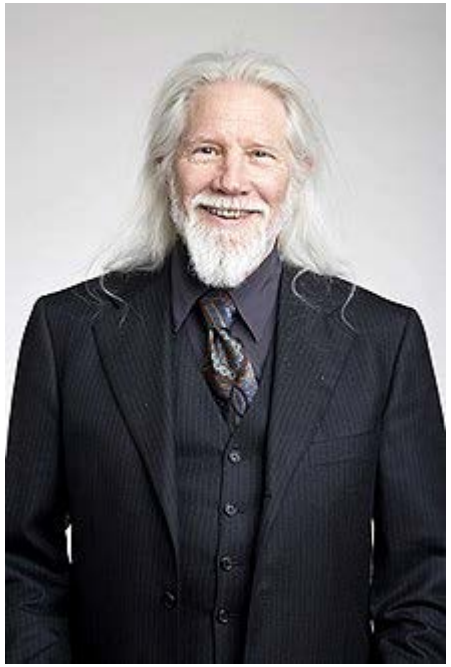
Section III proposes two approaches to transmitting



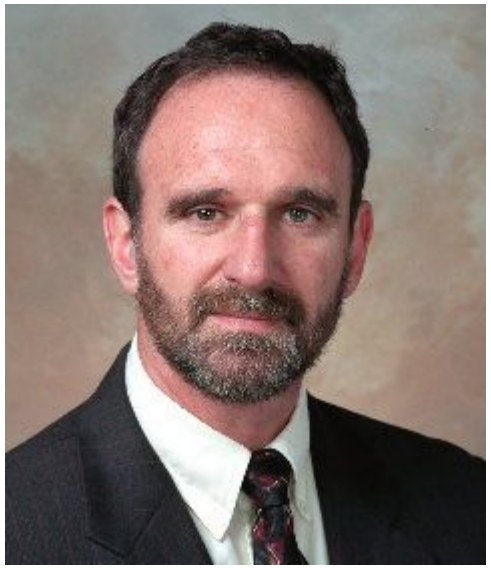
Martin Hellman

Cryptographie à clé mixte

1. Alice et Bob ont choisi un groupe fini (soit un corps fini, dont ils n'utilisent que la multiplication, soit une courbe elliptique) et un générateur g de ce groupe;
2. Alice choisit un nombre au hasard a, élève g à la puissance a, et dit à Bob g^a (calculé dans le groupe ; si par exemple ils travaillent dans le corps fini **Z**/p**Z**, ils échangeront les nombres modulo p, comme montré dans l'exemple ci-dessous), c'est-à-dire le nombre A;
3. Bob fait de même avec le nombre b: il transmet le nombre B = g à la puissance b modulo p;
4. Alice, en élevant le nombre B reçu de Bob à la puissance a, obtient g^{ba} (toujours calculé modulo p par exemple).
5. Bob fait le calcul analogue avec le A reçu d'Alice et obtient g^{ab}, qui est le même résultat.
6. Mais puisqu'il est difficile d'inverser l'exponentiation dans un corps fini (ou sur une courbe elliptique), c'est-à-dire de calculer le logarithme discret, Ève ne peut pas découvrir, donc ne peut pas calculer g^{ab} [mod p] ;
7. Finalement, Alice et Bob connaissent donc tous les deux le nombre g^{ab} [mod p] dont Ève n'a pas connaissance.

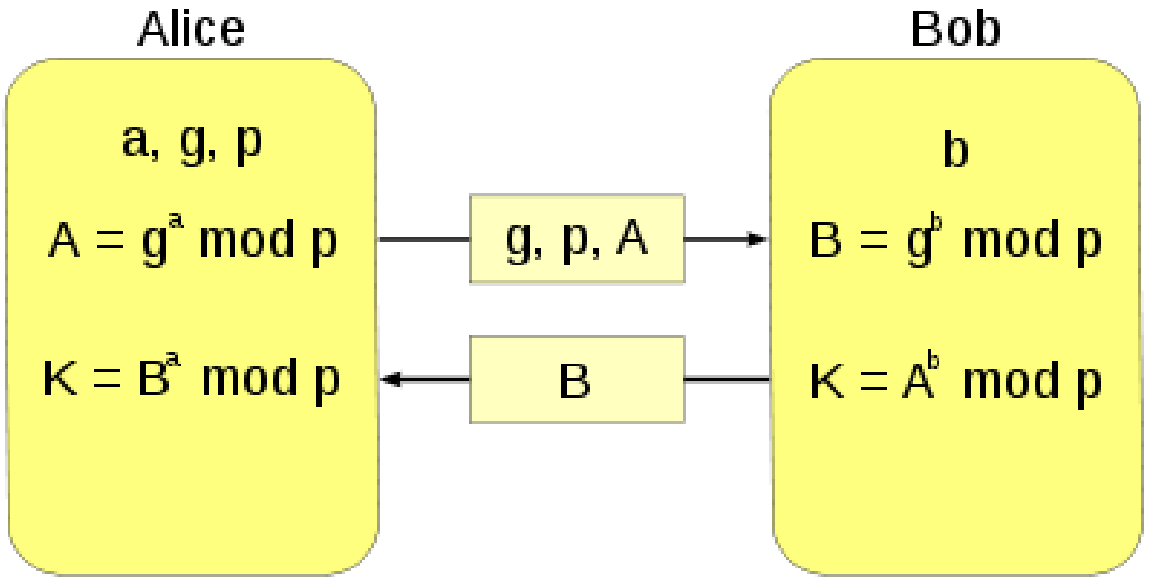


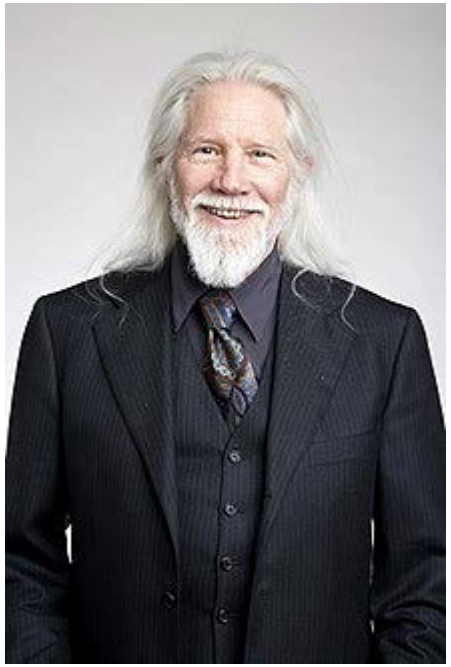
Whitfield 'Whit' Diffie



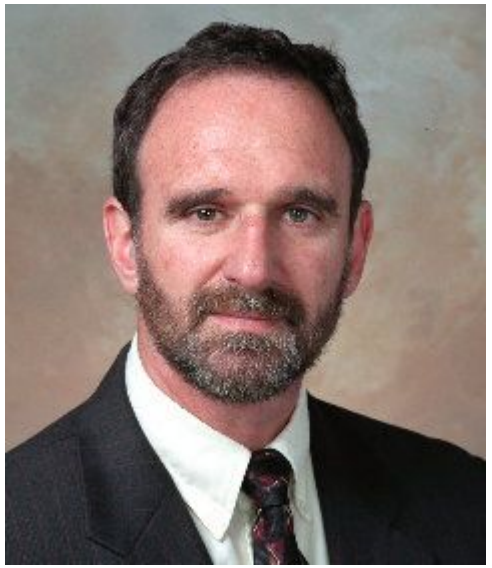
Martin Hellman

Cryptographie à clé mixte





Whitfield 'Whit' Diffie



Martin Hellman

Cryptographie à clé mixte

Alice et Bob ont choisi un nombre premier p et une base g. Dans notre exemple, p=23 et g=5

Alice choisit un nombre secret a=6

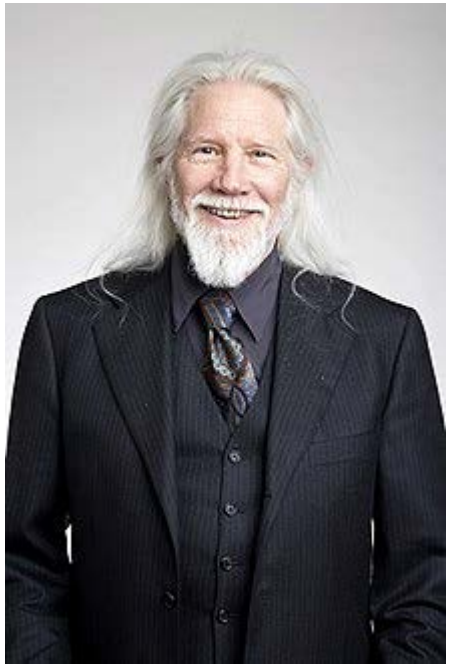
Elle envoie à Bob la valeur $A = g^a$ [mod p] = 5^6 [23] = 8

Bob choisit à son tour un nombre secret b=15

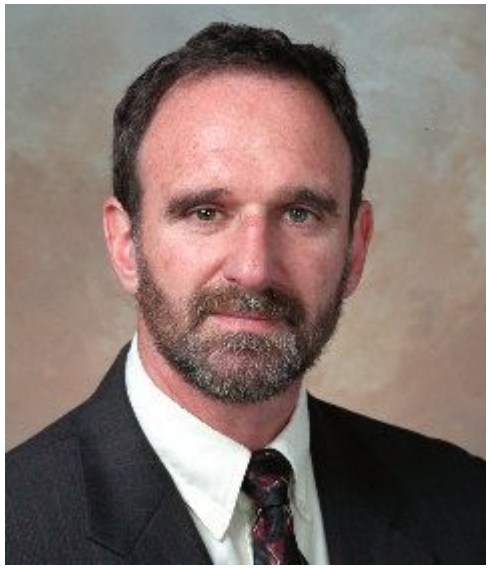
Bob envoie à Alice la valeur $B = g^b$ [mod p] = 5^{15} [23] = 19

Alice peut maintenant calculer la clé secrète : B^a [mod p] = 19^6 [23] = 2

Bob fait de même et obtient la même clé qu'Alice : A^b [mod p] = 8^{15} [23] = 2



Whitfield 'Whit' Diffie



Martin Hellman

Authenticité de l'expéditeur

- Comment garantir l'authenticité de l'expéditeur (fonction d'*authentification*) et de vérifier l'intégrité du message reçu ?

- La signature électronique permet d'identifier et d'authentifier l'expéditeur des données tout en vérifiant l'intégrité des données pour certaines méthodes.

Signature électronique

- Signature à clés publiques :
  - Principe entre un expéditeur A et un destinataire B avec 2 couples de clés clé publique/privée A (PA, SA) et B (PB, SB) :

Phase d'envoi

A code son message avec sa clé secrète : SA (m)

puis avec la clé publique de B : PB (SA(m)) et l'envoie à B.

Phase de réception

B décode avec sa clé privé :

SB (PB (SA (m))) = SA (m) Sécurité de l'envoi

Puis avec la clé publique de A, il décode m :

PA(SA(m)) = m Certification de A grâce à sa SA

Fonctionnement lent, utilisation de deux paires de clés et il n'y a pas de contrôle d'intégrité des données.

Signature avec hachage

- Le hachage consiste à calculer un résumé très petit du message.
  - Le résumé (appelé digest ou haché)
    - ne doit pas permettre de reconstituer le texte initial s'il est pris tout seul,
    - doit être sensible (toute modification du message provoque une modification du résumé).
  - Cette méthode
    - permet de s'assurer de l'intégrité du message.
    - couplée à la cryptographie à clé publique permet aussi l'authentification de l'expéditeur.

Signature avec hachage

phase d'envoi

- A calcule le résumé H(m) le code avec sa clé privé SA(H(m))
- A code avec la clé publique de B le message: PB(m)
- il les envoie à B : PB(m) et SA(H(m))

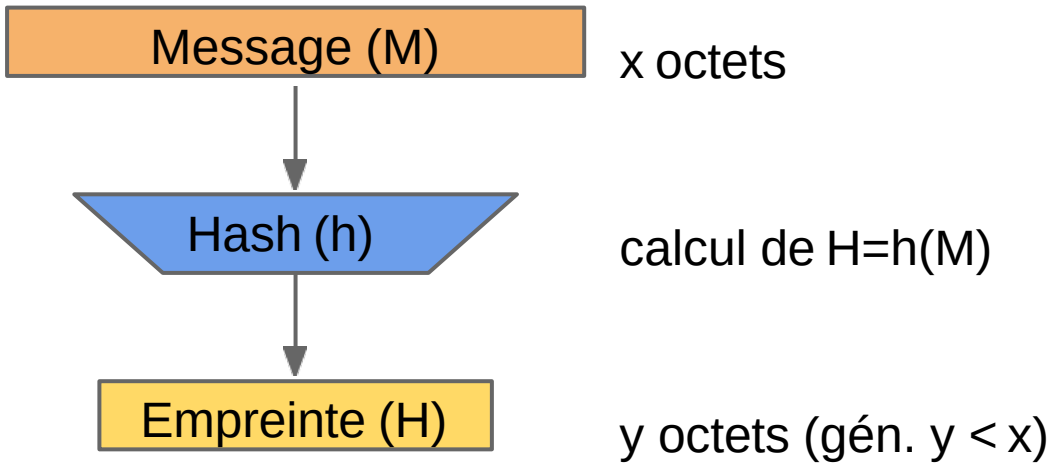
phase de réception

- B décode le message avec sa clé privée : SB (PB (m)) = m’
- B résume ce message H (m’) et décode le résumé reçu avec la clé publique de A : PA (SA (H (m)))
- si H (m')= H (m) alors A est bien authentifié et le message est correct.

Fonctions de hachage

Principe général

- M peut avoir “n’importe quelle taille”:x (ex. SHA256: ~2M)
- H est d’une taille “fixée” à l’ avance: y
- Il n’y a pas de lien entre x et y: la taille de l’empreinte est indépendante de celle du message
- Le calcul de H=h(M) doit être “simple”, l’inverse doit être complexe (“impossible”)



Définitions

“Action de hacher.” (Larousse)

“On nomme fonction de hachage une fonction particulière qui, à partir d'une donnée fournie en entrée, calcule une empreinte servant à identifier rapidement, bien qu'incomplètement, la donnée initiale.” (Wikipédia)

Le résultat d’une fonction de hachage est appelé une empreinte ou un haché. On peut également trouver d’autres termes (condensé, condensat, ou somme de contrôle).

En anglais:

- hash function
- hash: empreinte (parfois aussi utilisé en français…)
- checksum: somme de contrôle

Propriétés

Une fonction de hachage est une fonction:

- **Unidirectionnelle**
  On peut facilement calculer une empreinte, mais l'inverse doit être difficile
- **Déterministe**
  Une entrée produit toujours la même sortie
- **Uniforme**
  Les sorties possibles sont distribuées de façon uniforme pour chaque entrée

On peut aussi parler de la propriété suivante (en dehors d'un contexte cryptographique):

- **Continuité**
  Deux messages proches doivent avoir une empreinte proche

Propriétés - cryptographie

En cryptographie, on parle de fonction de hachage cryptographique, ou cryptographiquement sûre. Dans ce cas, l'uniformité des sorties devient importante.

De plus, on cherche à avoir l'inverse de la continuité: pour un changement minime du message initial, on souhaite une modification importante de l' empreinte obtenue.

Par exemple, avec SHA-256:

- "Message de test" > `1e895a85dd70a21b11fa1be3be3061c9e0db5da8c7ffd16bbb8886c8987020e4`
- "message de test" > `23dd6f609f3bde93b4d9874561647bb73579fc42759544df670368c503afd3c6`
- M=01001101b
- m=01101101b

Propriétés - résistances

Lorsque l'on parle de fonction de hachage cryptographiquement sûre, on s' intéresse particulièrement à trois propriétés:

- Résistance de la première préimage
- Résistance de la seconde préimage
- Résistance aux collisions

On peut noter que ces propriétés s'appuient sur deux choses:

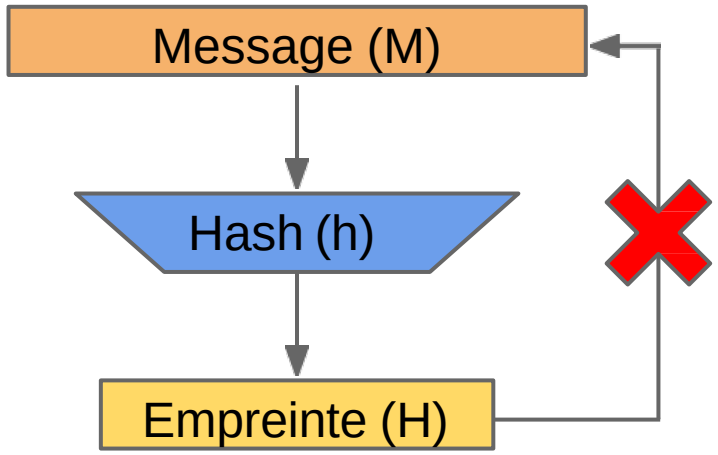
- La résistance de l'algorithme, qui ne doit pas exposer de défauts permettant de réduire ces résistances
- Les capacités de calculs, grâce auxquels une attaque "brutale" peut être réalisée

On peut donc parler d'estimation de durée de vie lorsque l'on parle d'algorithme cryptographiques...

Propriétés - résistances

Première préimage:

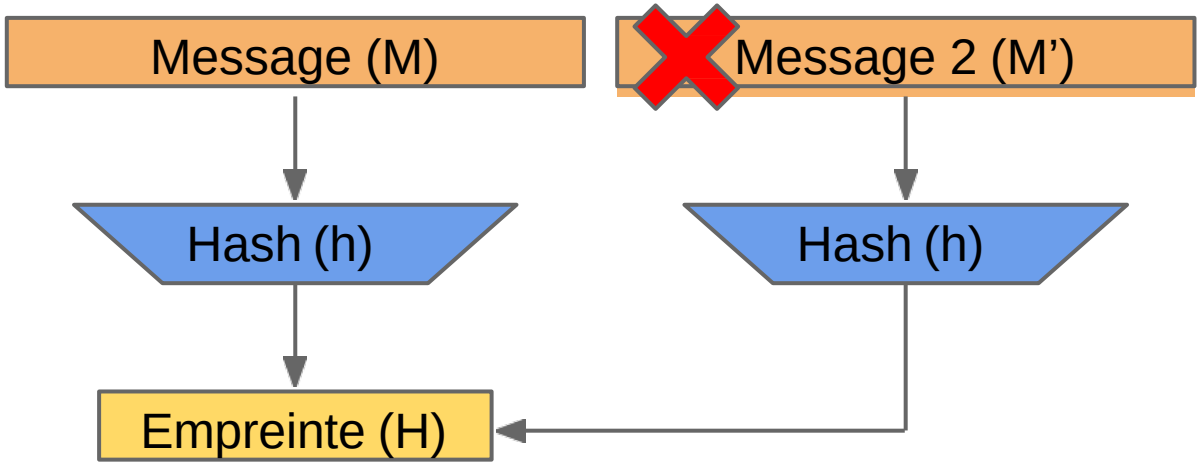
- Connaissant H, il est impossible de retrouver M tel que h(M)=H
- On parle également de fonction à sens unique



Propriétés - résistances

Seconde préimage:

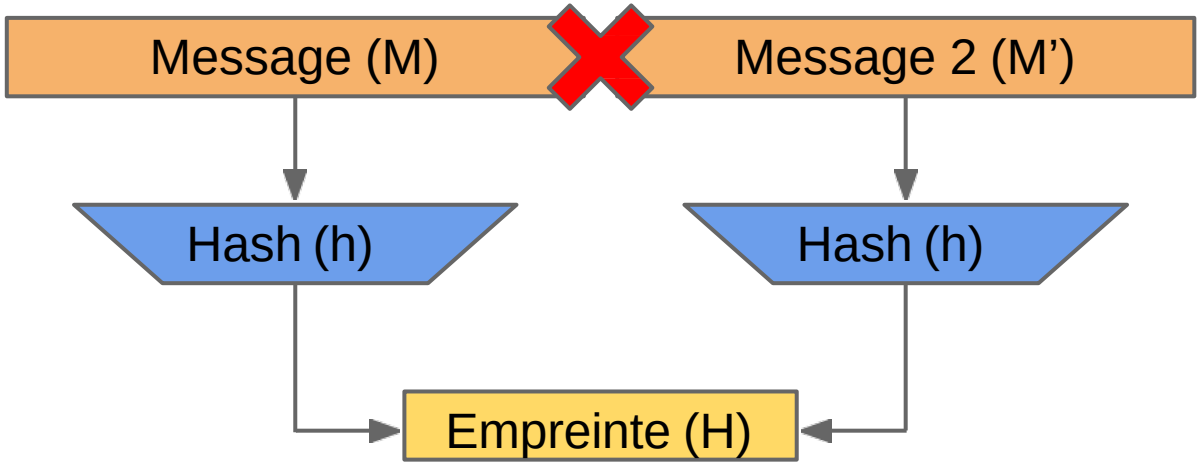
- Connaissant M (et H), il est impossible de trouver un message M' tel que h(M')=h(M) =H
- On parle aussi de résistance faible aux collisions



Propriétés - résistances

Résistance forte aux collisions:

- Il est très difficile de trouver deux messages M et M’ ayant la même empreinte



Fonctionnement

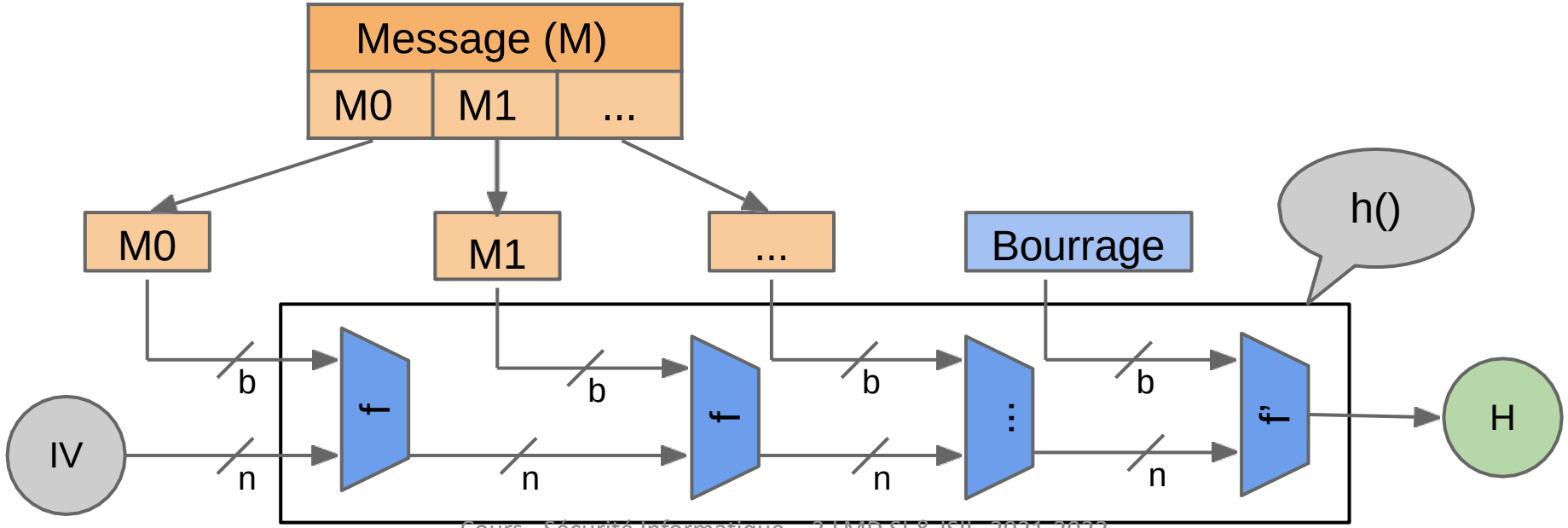
Les fonctions de hachage fonctionnent à peu près de la même façon que les chiffrements par blocs:

- On découpe le message initial en “blocs” sur lesquelles on opère séquentiellement.
- Au coeur d’une fonction de hachage se trouve une fonction de compression (f).
- Certaines fonctions de hachage peuvent se voir utilisées avec un IV (vecteur d’initialisation). Il peut être utilisé comme une clé pour des systèmes d’authentification.

Fonctionnement

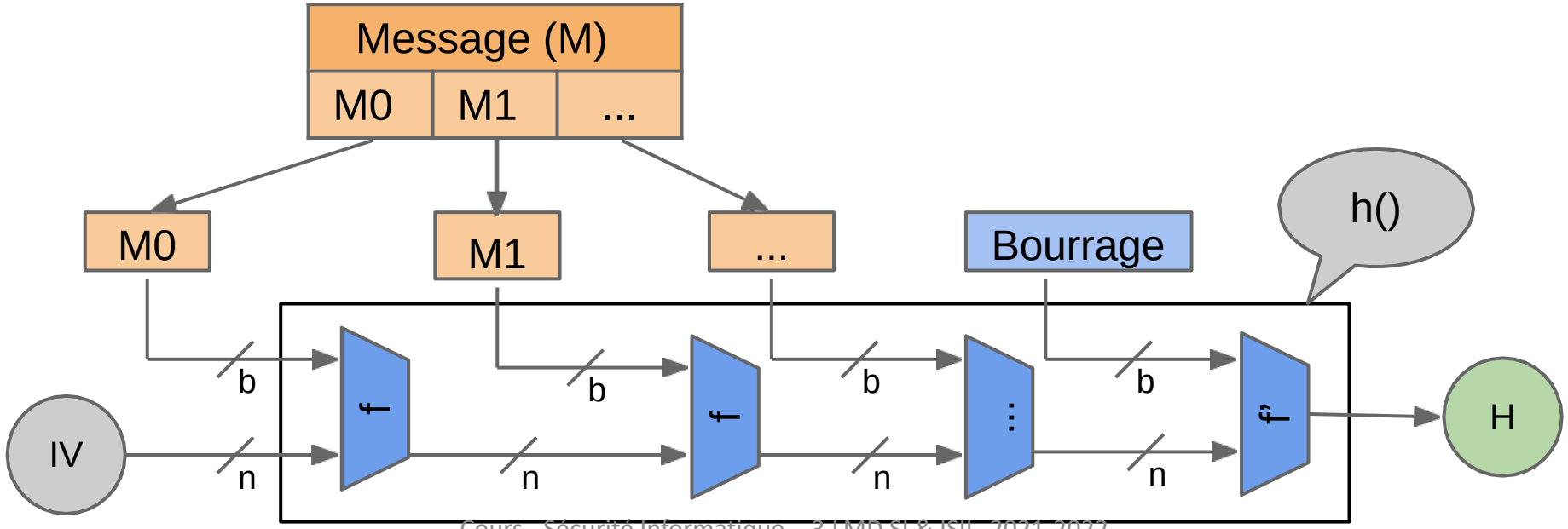
- f: fonction de compression
- b: taille de bloc pour l'entrée de la fonction de compression
- n: taille de l'état de la fonction de hachage
- Bourrage: donnée de fin ajoutée au message (optionnel)

Il s'agit de la construction de Merkle-Damgård.



Fonctionnement

- Pour certains algorithmes, n peut avoir la taille de H. Pour d'autre, n peut être différent (généralement plus grand)
- Le "bourrage" n'est pas toujours présent
- L'IV aussi est optionnel. On peut également utiliser un premier bloc, inséré avant le message, pour jouer un rôle similaire.
- f et f' peuvent être la même fonction, mais ce n'est pas toujours le cas



Algorithmes

Algorithmes de hachage:

- MD2, MD4, **MD5**, MD6
- RIPEMD
- SHA-0, SHA-1, **SHA-2**

Le “dernier” algorithme de hachage en cours de standardisation est SHA-3:

- Compétition du NIST, finalistes: BLAKE, Grøstl, JH, Keccak, Skein
- Le 2 octobre 2012: choix de **Keccak**
- C’est un choix “préventif”: SHA-2 est toujours considéré comme sûr. L’ objectif étant d’avoir un successeur “prèt”.
- Keccak est basé sur un fonctionnement interne différent de SHA-2

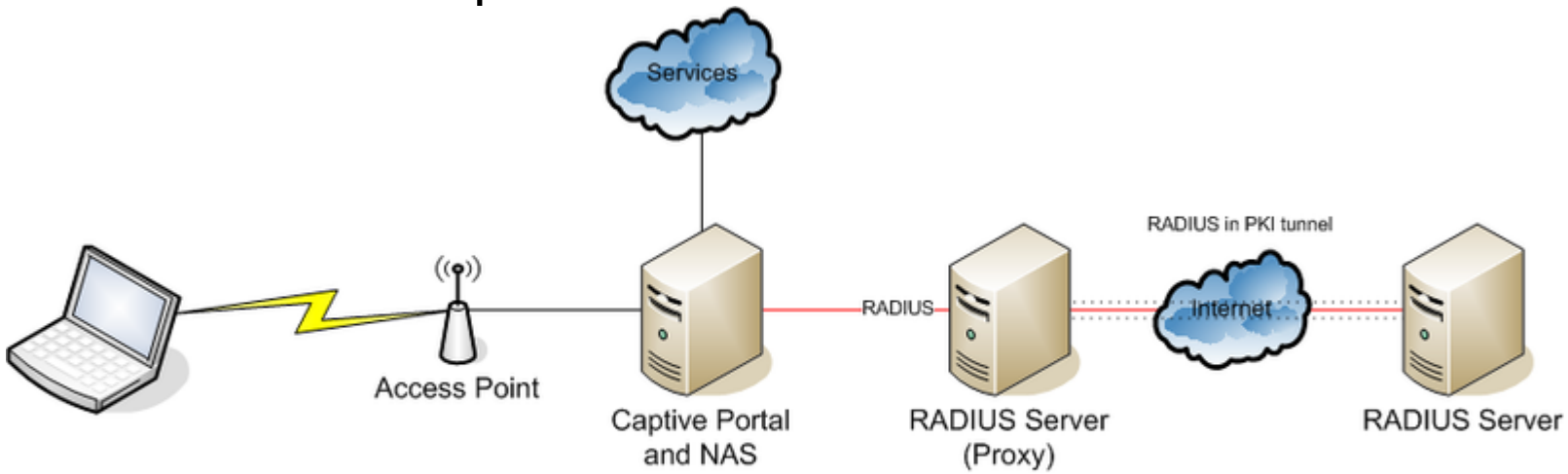
Les protocoles de sécurité

Protocole AAA

- En sécurité informatique, **AAA** correspond à un protocole qui réalise trois fonctions : l'authentification, l'autorisation, et la traçabilité.
- AAA est un modèle de sécurité implémenté dans certains routeurs Cisco mais que l'on peut également utiliser sur toute machine qui peut servir de NAS (Network Access Server), ou certains switches Alcatel.
- AAA est la base des protocoles de télécommunication Radius et Diameter qui sont notamment utilisés dans les réseaux mobiles UMTS et LTE pour authentifier et autoriser l'accès des terminaux mobiles au réseau.

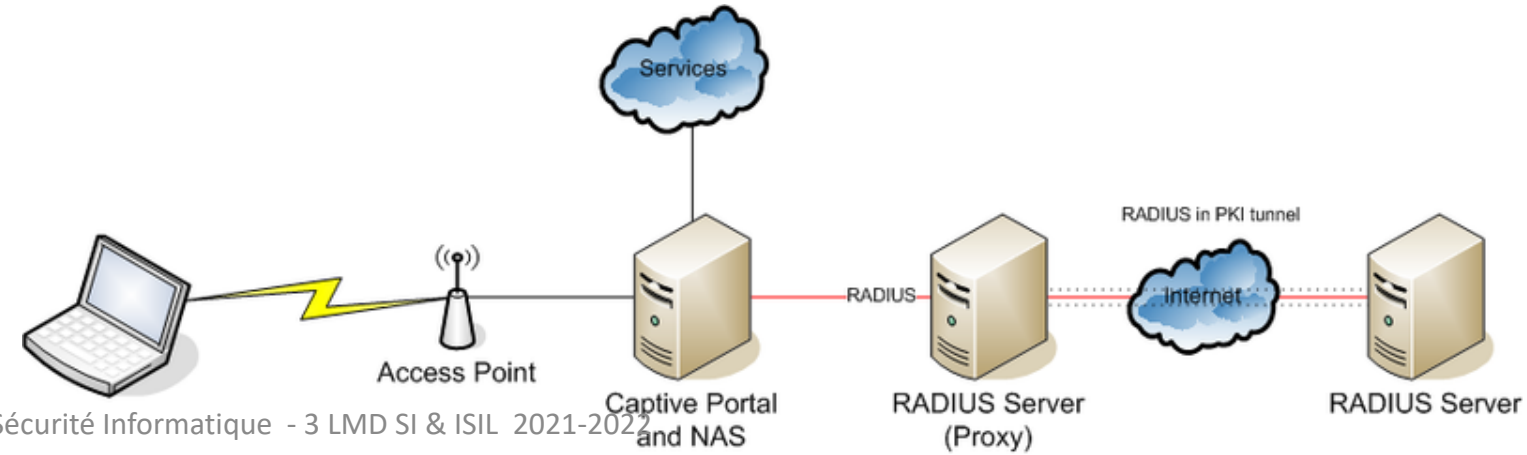
RADIUS - Principes généraux

- Protocole standard d'authentification, initialement mis au point par Livingston.
- Défini au sein des RFC 2865 et 2866.
- Fonctionnement basé sur un système client/serveur chargé de définir les accès d'utilisateurs distants à un réseau.
- Le protocole RADIUS permet de faire la liaison entre des besoins d'identification et une base d'utilisateurs en assurant le transport des données d'authentification de façon normalisée.



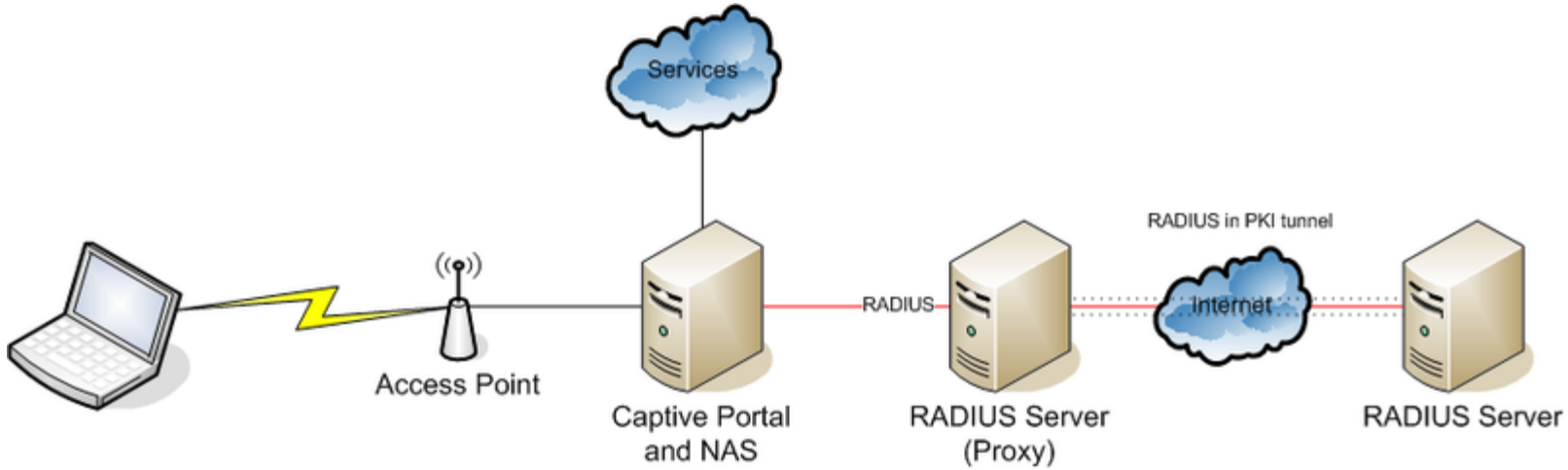
RADIUS - Scénario de fonctionnement (1/2)

- Un utilisateur envoie une requête au NAS afin d'autoriser une connexion à distance.
- Le NAS achemine la demande au serveur RADIUS.
- Le serveur RADIUS consulte sa base de données d'identification afin de connaître le type de scénario d'identification demandé pour l'utilisateur.
- Soit le scénario actuel convient, soit une autre méthode d'identification est demandée à l'utilisateur. Le serveur RADIUS retourne ainsi une des quatre réponses suivantes :
    - **ACCEPT** : l'identification a réussi.
    - **REJECT** : l'identification a échoué.
    - **CHALLENGE** : le serveur RADIUS souhaite des informationssupplémentaires de la part de l'utilisateur et propose un « défi ».



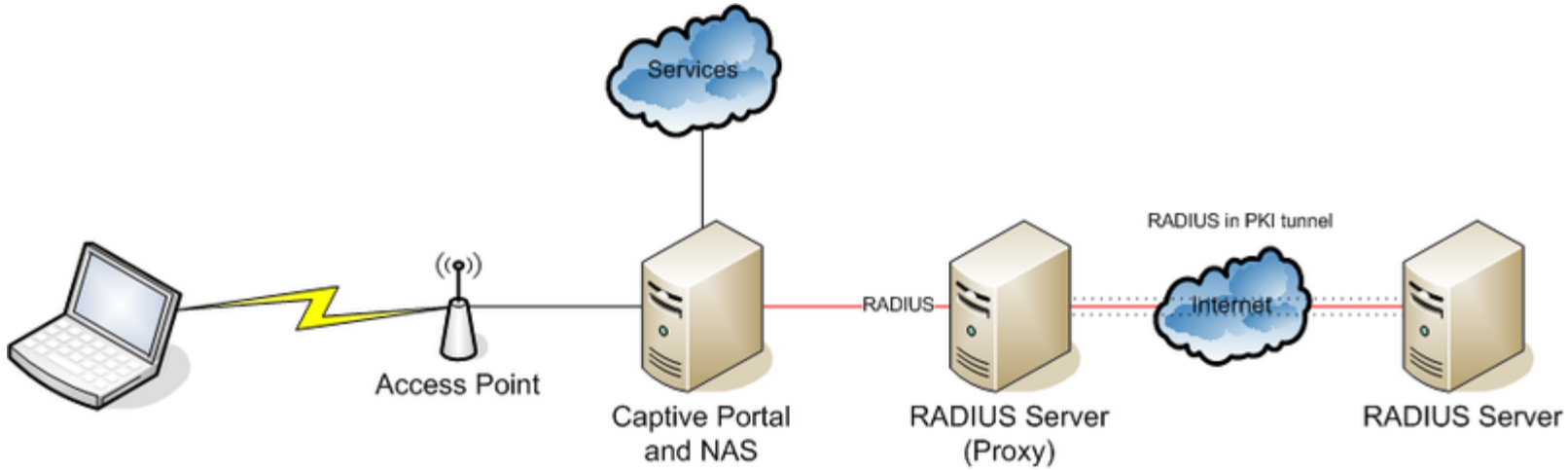
RADIUS - Scénario de fonctionnement (2/2)

- Une autre réponse est possible : **CHANGE PASSWORD** où le serveur RADIUS demande à l'utilisateur un nouveau mot de passe.
- Change-password est un attribut VSA (Vendor-Specific Attributes), c'est-à-dire qu'il est spécifique à un fournisseur.
- Suite à cette phase dit d'authentification, débute une phase d'autorisation où le serveur retourne les autorisations de l'utilisateur.

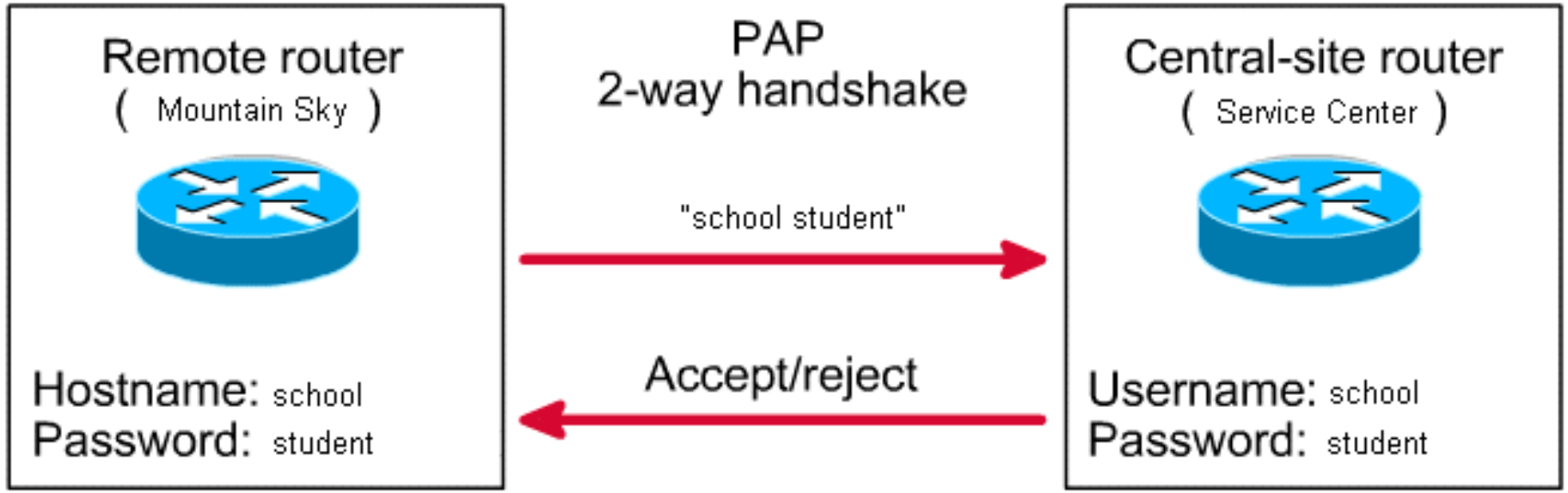


RADIUS - Protocoles de mots de passe

- RADIUS connaît nativement deux protocoles de mots de passe :
    - ***PAP (échange en clair du nom et du mot de passe),***
    - ***CHAP (échange basé sur un hachage de part et d'autre avec échange seulement du 'challenge').***
- Le protocole prévoit deux attributs séparés : User Password et CHAP-Password.

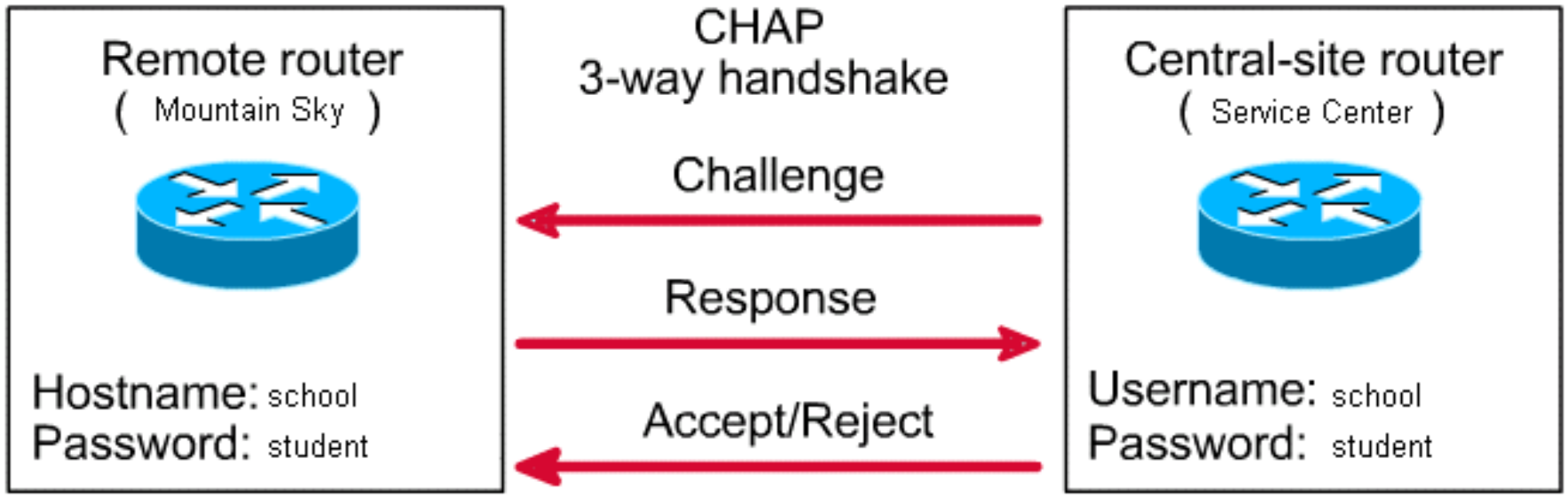


RADIUS - Protocoles de mots de passe Password Authentication Protocol (PAP)



Point-to-Point Protocol (PPP)

RADIUS - Protocoles de mots de passe Challenge-Handshake Authentication Protocol



(CHAP) est un protocole d'authentification pour PPP à base de challenge

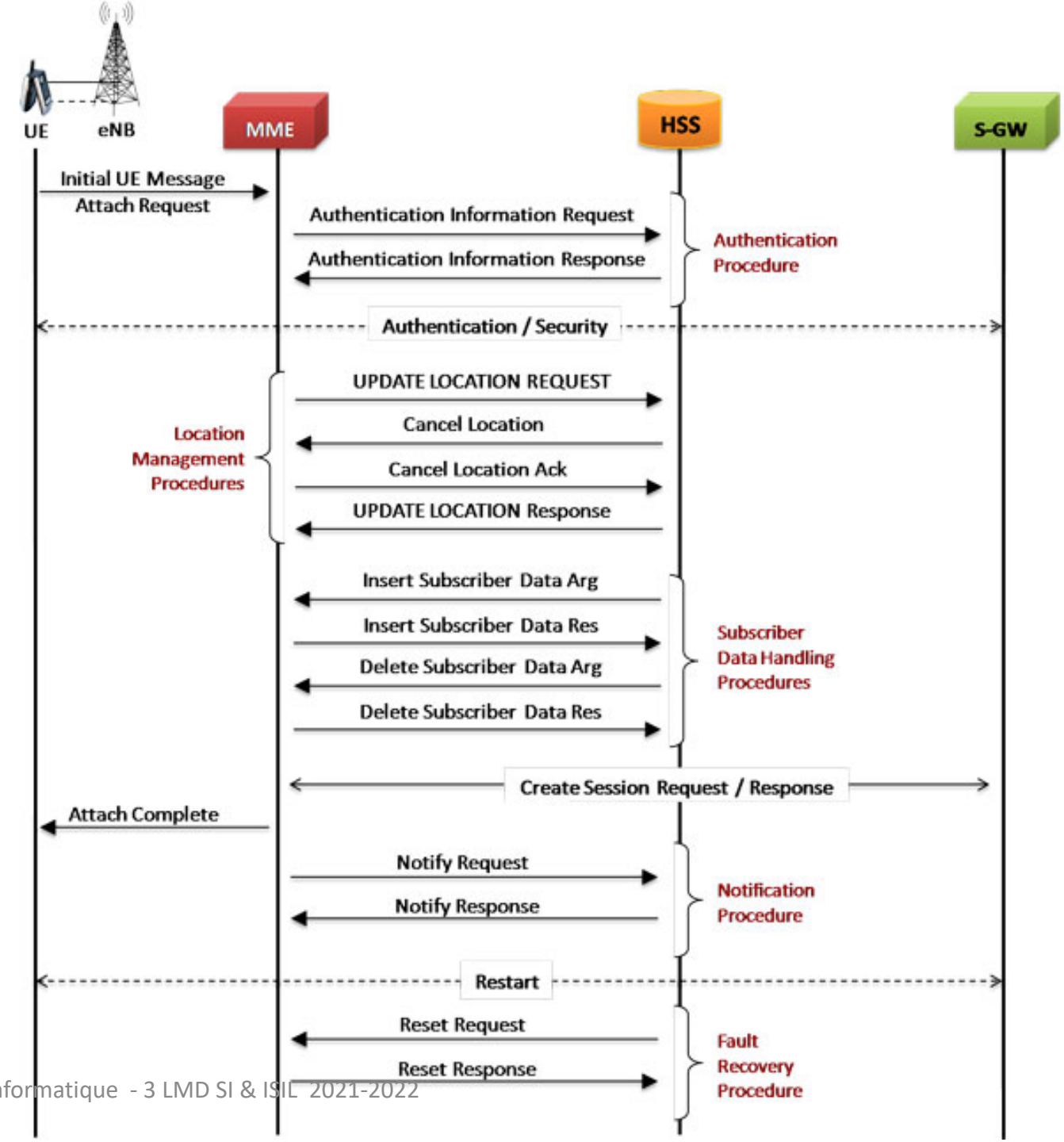
Diameter

Diameter (1)

- Diameter est un protocole d'authentification, successeur du protocole RADIUS.
- Ce protocole est défini par la RFC 35881, et définit les pré-requis minimums nécessaire pour un protocole AAA.
- Il est notamment utilisé dans le cœur des réseaux de téléphonie mobile pour accéder aux bases de données HSS permettant d'identifier, d'authentifier et de localiser les abonnés mobiles 3G et LTE /4G.

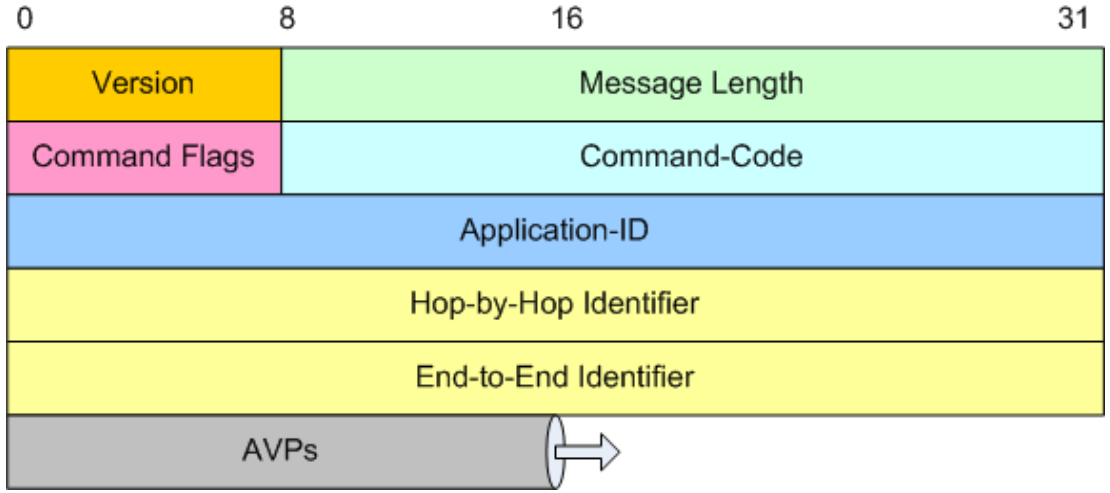
Diameter (2)

- **MME (Mobility management entity)**
- **HSS (Home Subscriber Server)**
- **S-GW (Gateway)**



Diameter (2) - Structure du message de Diameter

- Le Diameter est un protocole basé sur les messages (paquets). Il existe deux types de messages, à savoir, le message Request et le message Answer. La structure de ces deux messages est présentée dans la figure

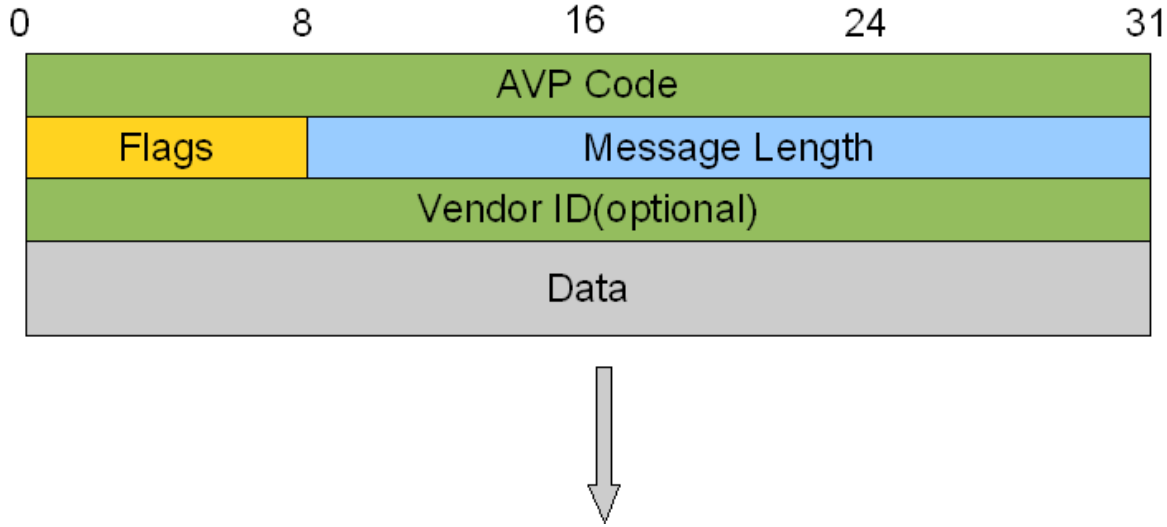


Structure du message de Diameter

- Version : Ce champ de version doit être réglé sur 1 pour indiquer la version 1.
- Longueur du message (Message Length) : Contenir la longueur de Message Header + (Data) Avp.
- Drapeaux de commande (Command Flags) : Le champ drapeaux de commandes est de huit bits.
- ID d'application (ID d'application) : Pour identifier de manière unique chaque application.
- Hop-by-Hop Identifier : L'identificateur Hop-by-Hop est un champ entier non signé de 32 bits (en ordre d'octet réseau) et aide à faire correspondre les demandes et les réponses.
- Identificateur de bout en bout (End-to-End Identifier): L'identificateur de bout en bout est un champ entier non signé de 32 bits (en ordre d'octet de réseau) et sert à détecter des messages en double.

Diameter (2) - Les AVP de Diameter

- Les AVP de Diameter sont l'unité de base dans le message Diameter qui contient les données (données d'authentification, données de sécurité, données relatives à l'application, etc.). Il doit y avoir au moins un AVP à l'intérieur du message Diameter. La structure de l'AVP Diameter est présentée dans la figure



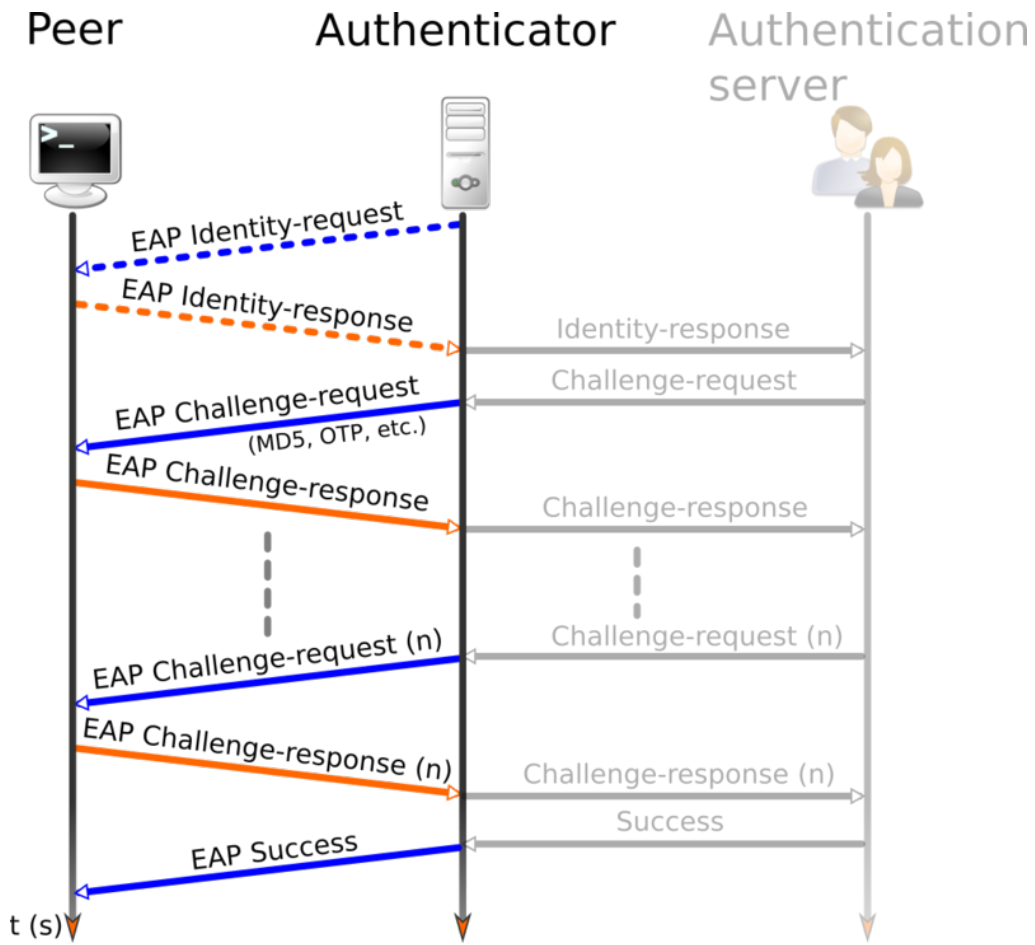
Structure de l'AVP Diameter

- Code AVP (4 octets) : Le code AVP, combiné avec le champ Vendor-Id, identifie l'attribut uniquement. Les numéros AVP 256 et supérieurs sont utilisés pour le Diamètre.
- Drapeaux (Flags) : Indicateurs de bits qui spécifient comment chaque attribut doit être traité. Une description complète est disponible dans la section 4.1 de RFC 3588.
- AVP Longueur (AVP Length): Indique le nombre d'octets dans l'AVP, y compris les informations suivantes: Code AVP, AVP Longueur, Drapeaux AVP, Champ d'identification du fournisseur (s'il y a lieu) et Données AVP.
- Fournisseur ID (Vendor-ID): Un octet optionnel qui identifie l'AVP dans l'espace d'application. Le code AVP et AVP Vendor-ID créent un identifiant unique pour AVP.

Le protocol EAP

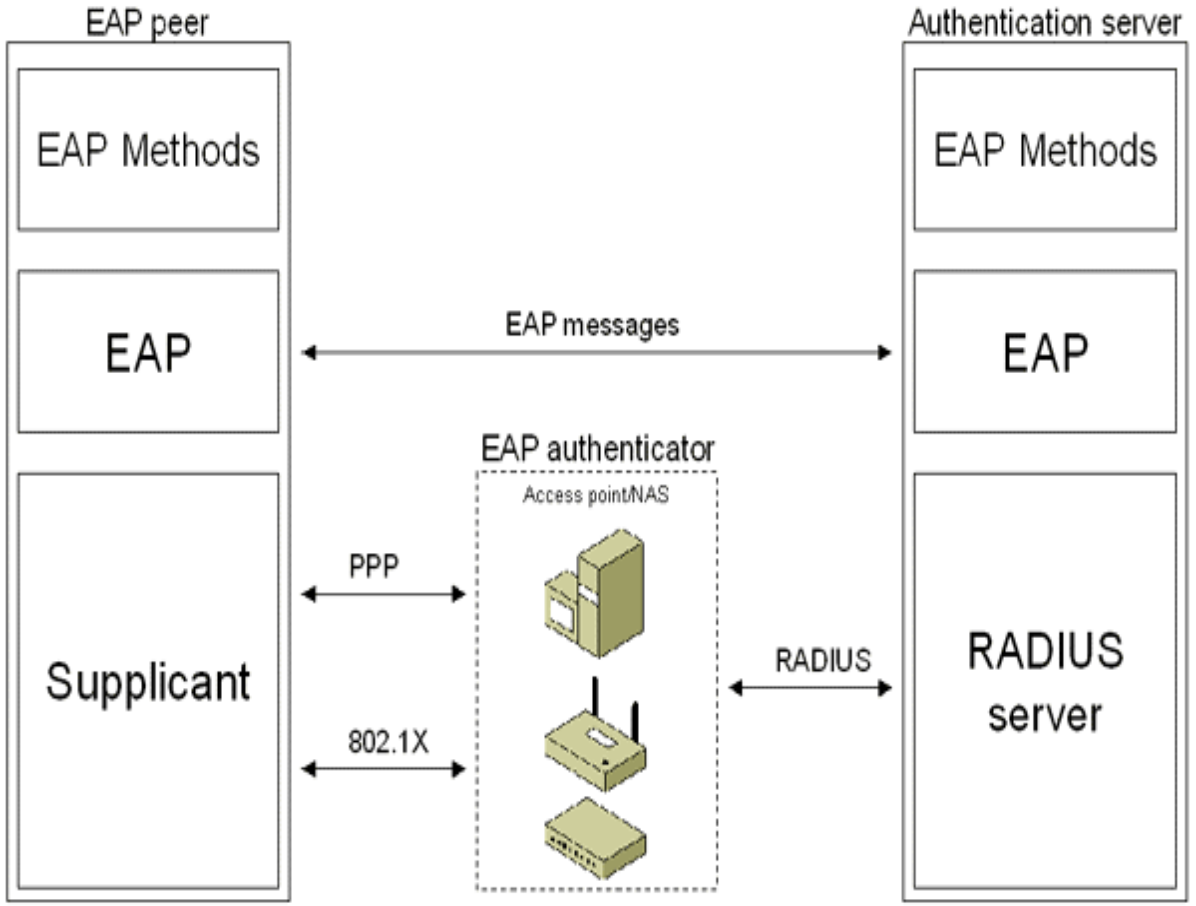
Le Protocol EAP

- Extensible Authentication Protocol ou EAP est un protocole de communication réseau embarquant de multiples méthodes d'authentification, pouvant être utilisé sur les liaisons point à point (RFC 22841), les réseaux filaires et les réseaux sans fil (RFC 37482, RFC 52473) tel que les réseaux Wi-Fi.
- Plusieurs méthodes d'authentification sont prédéfinies (MD5, OTP, Generic Token Card, etc.) mais il est possible d'en rajouter sans qu'il soit nécessaire de changer ou de créer un nouveau protocole réseau.



Le Protocol EAP

- D'un point de vue architectural, une infrastructure EAP est constituée des éléments suivants, comme présenté dans la figure :
- *Homologue EAP* : Ordinateur qui tente d'accéder à un réseau, également appelé client d'accès.
- *Authentificateur EAP*: Point d'accès ou serveur d'accès réseau qui nécessite une authentification EAP avant d'accorder l'accès à un réseau.
- *Serveur d'authentification* : Ordinateur serveur qui négocie l'utilisation d'une méthode EAP spécifique avec un homologue EAP, qui valide les informations d'identification de l'homologue EAP et qui autorise l'accès au réseau. En général, le serveur d'authentification est un serveur RADIUS (Remote Authentication Dial-In User Service).



L'infrastructure EAP et le flux d'informations

EAP dans les différentes versions de Windows

La prise en charge EAP dans Microsoft Windows a débuté avec Windows 2000, qui prenait en charge les méthodes EAP suivantes :

- EAP-Message Digest 5 Challenge Handshake Authentication Protocol (EAP-MD5 CHAP)
- EAP-Transport Layer Security (EAP-TLS)
- Security Dynamics' ACE/Agent

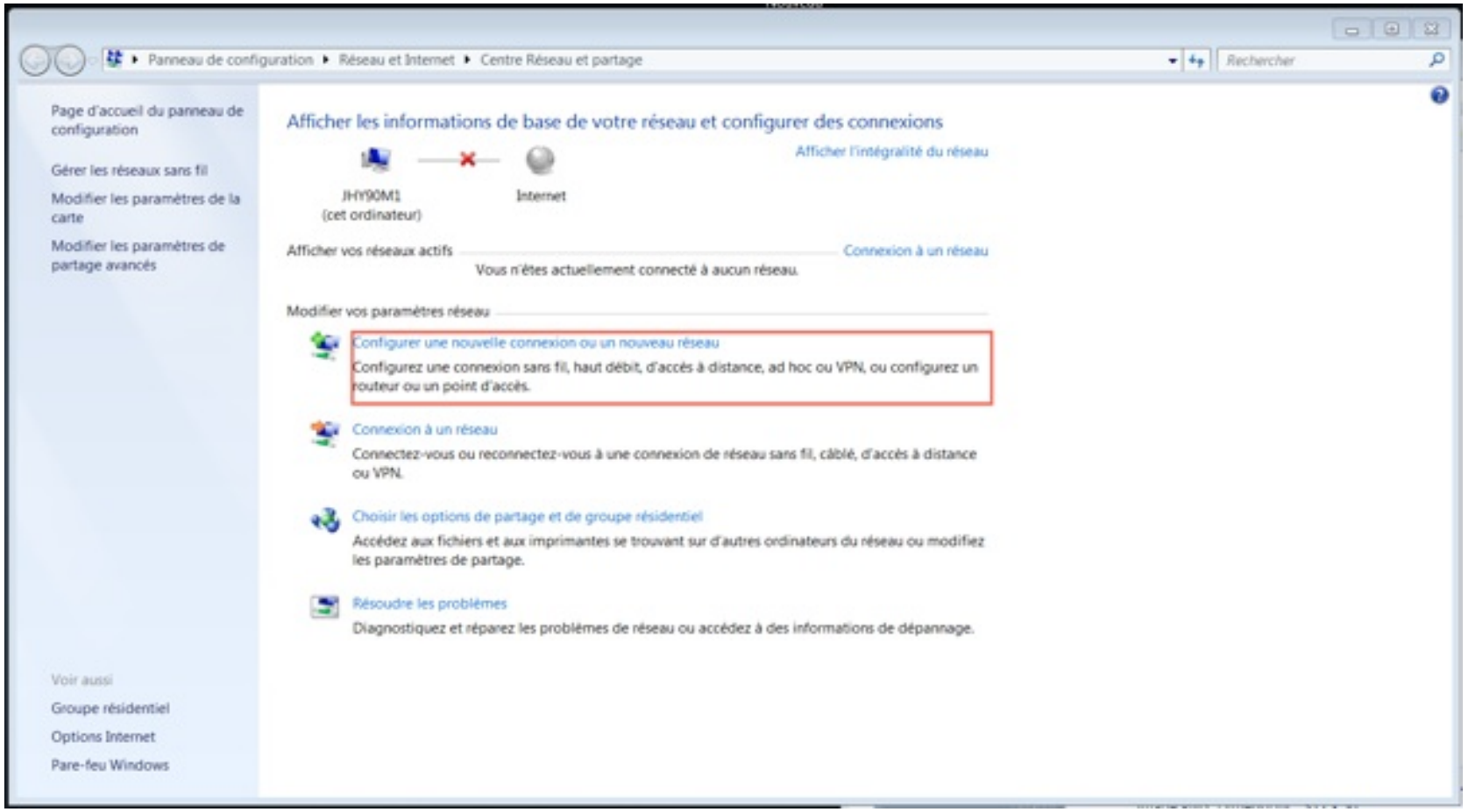
Windows XP Service Pack 1, Windows XP Service Pack 2, Windows Server 2003, et Windows 2000 Service Pack 4 supportent aussi the les méthodes EAP suivantes:

- Protected EAP (PEAP)
- PEAP-MS-CHAP v2
- PEAP-TLS

Configuration du protocole EAP (1)

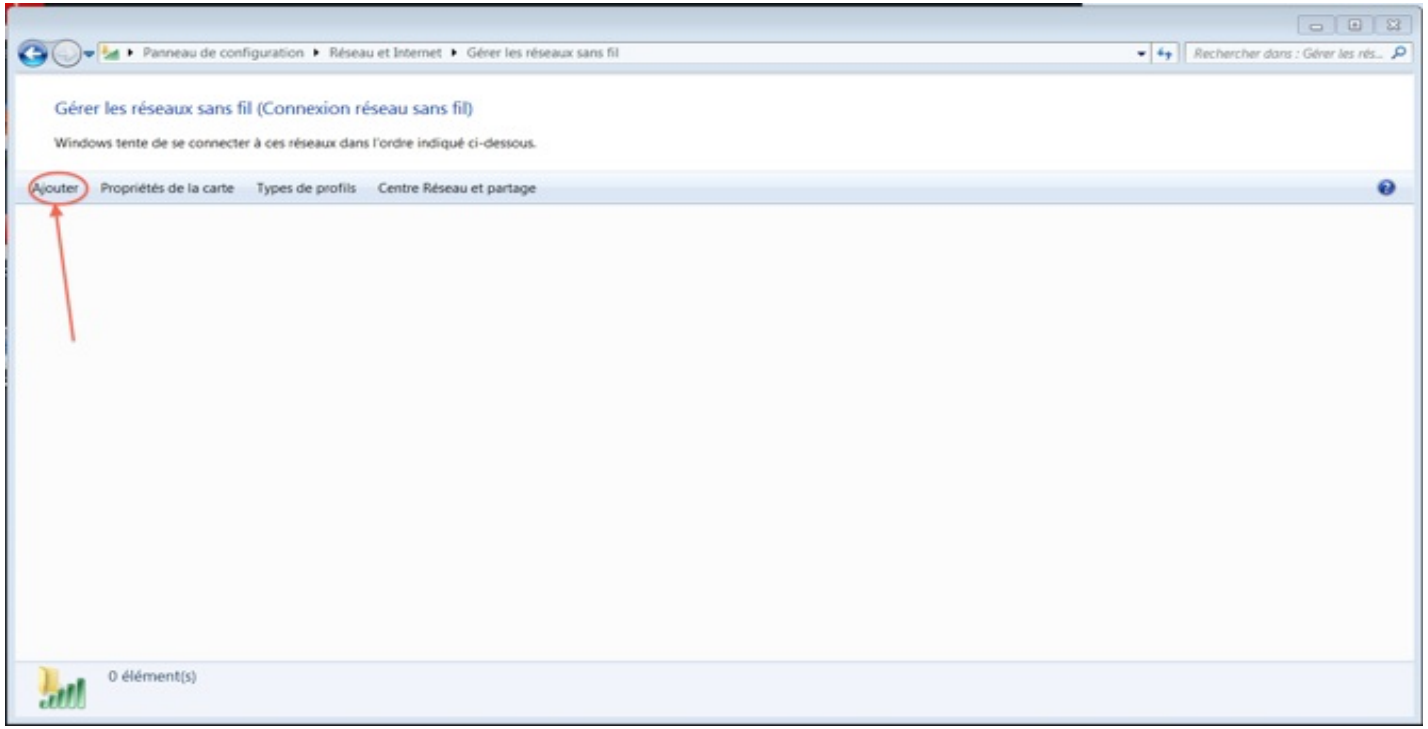
- Pour les ordinateurs exécutant Windows, le client EAP (client d'accès) est un ordinateur qui tente une connexion et le serveur d'authentification est un ordinateur exécutant Windows Server 2003 ou Windows 2000 Server et le service de routage d'authentification Internet (IAS) (pour tous les types de connexions). Dans le reste de cette sous-section, nous allons voir comment configurer du réseau wifi sous windows 7 utilisant le protocole EAP.
- Commencez par aller dans Panneau de configuration -> Réseaux et internet -> Centre réseau et Partage et selectionnez ‘Configurez une nouvelle connexion ou un nouveau réseau’

Configuration du protocole EAP (2)



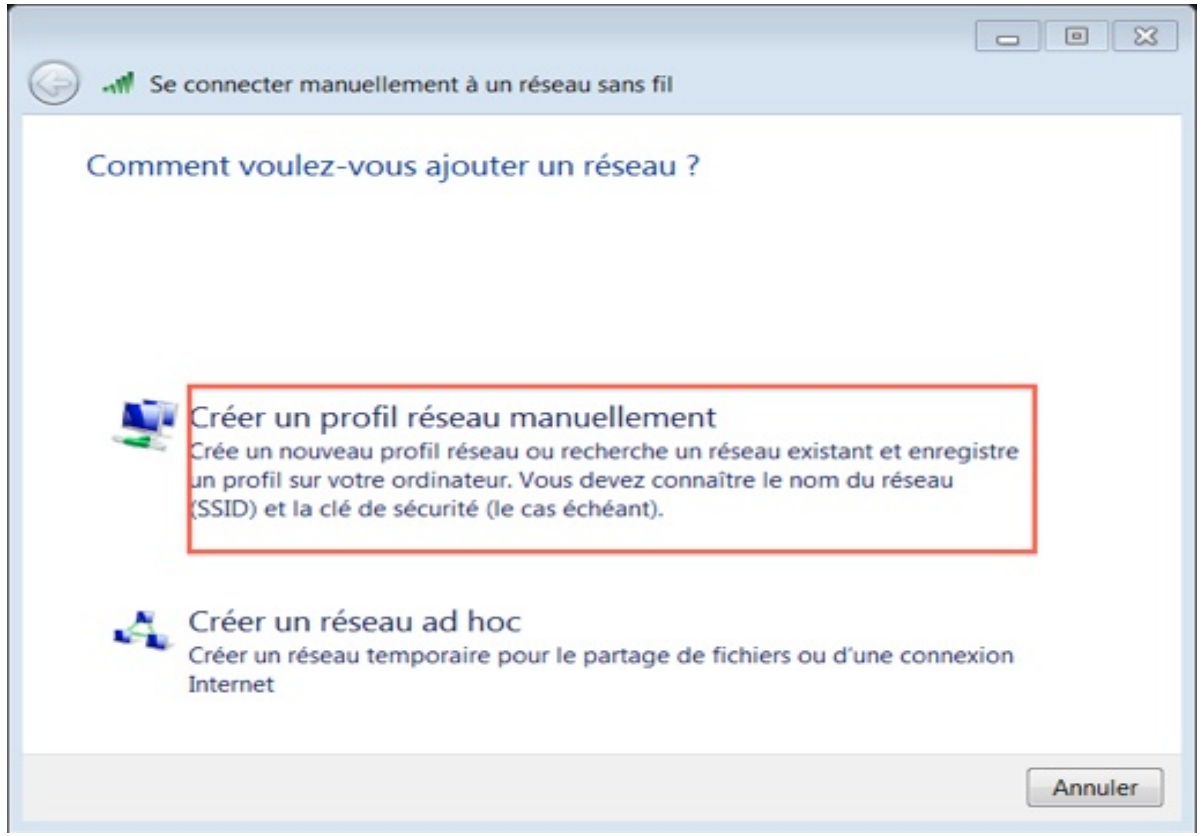
Configuration du protocole EAP (3)

- Cliquez sur le bouton ‘Ajouter’



Configuration du protocole EAP (4)

- Choisissez ‘Créer un profil réseau manuellement’



- Renseignez les champs comme ci-dessous, puis cliquez sur ‘Suivant’

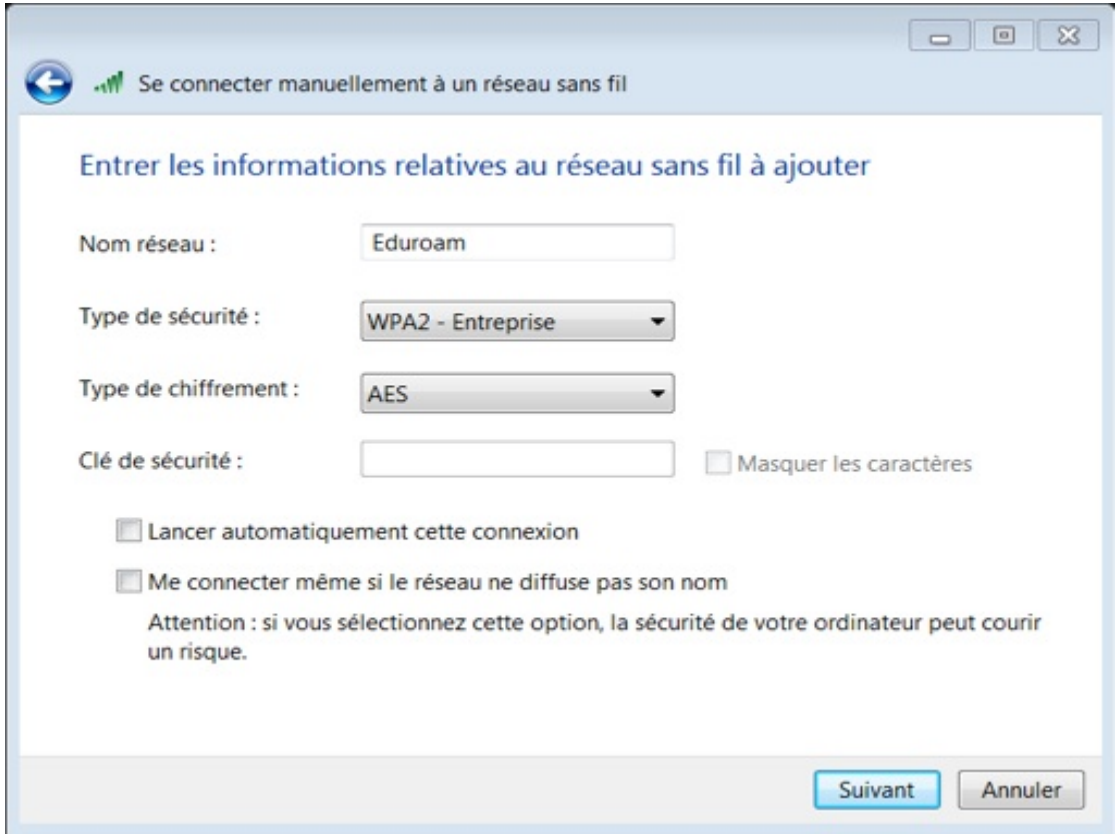
  Nom du réseau : Eduroam

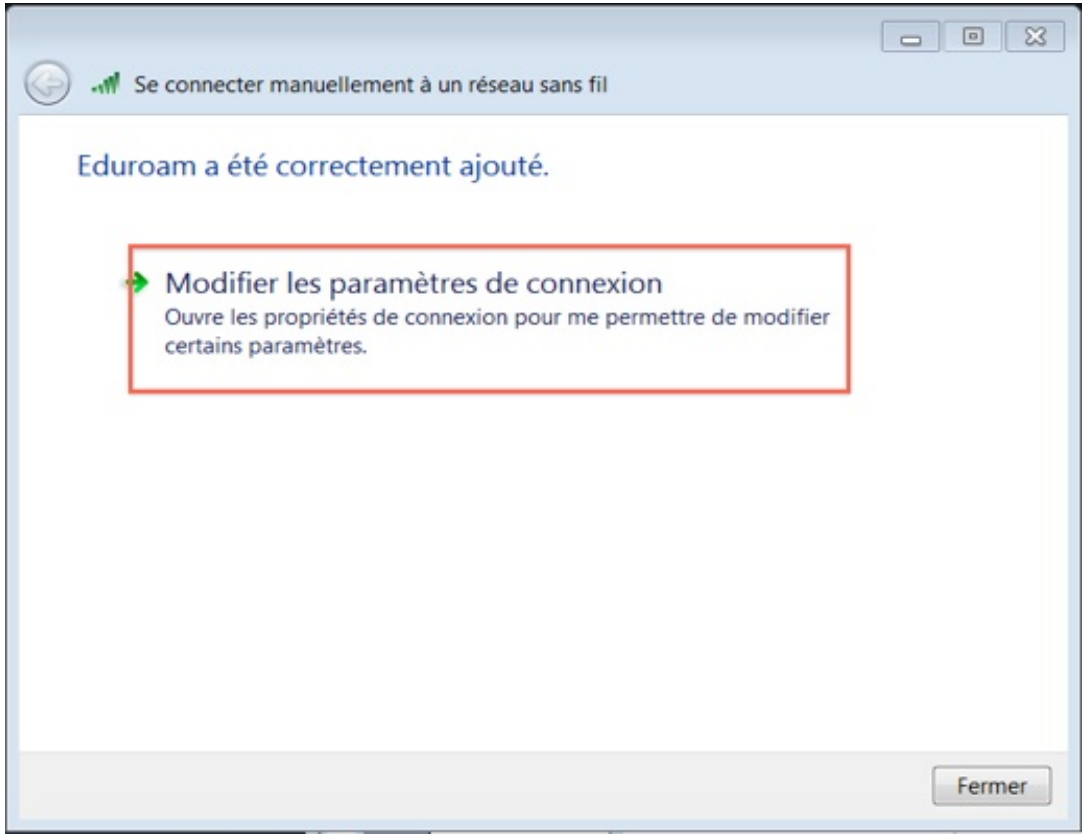
  Type de sécurité : WPA2 - Entreprise

  Type de chiffrement : AES

Configuration du protocole EAP (5)

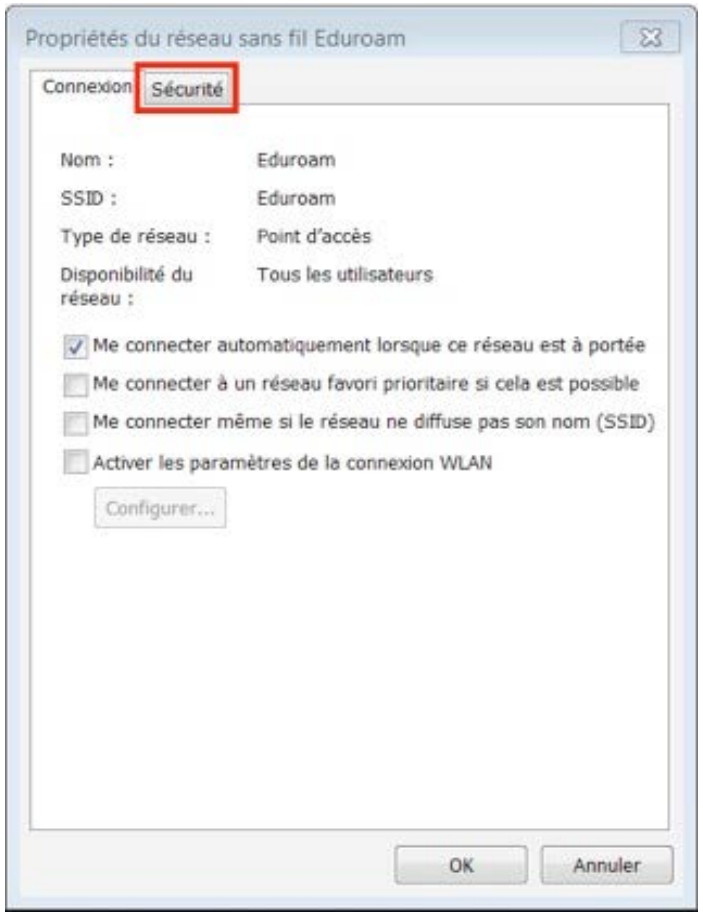
Le profil maintenant créé, il faut à présent configurer la connexion.
Donc cliquez sur ‘Modifier les paramètres de connexion’





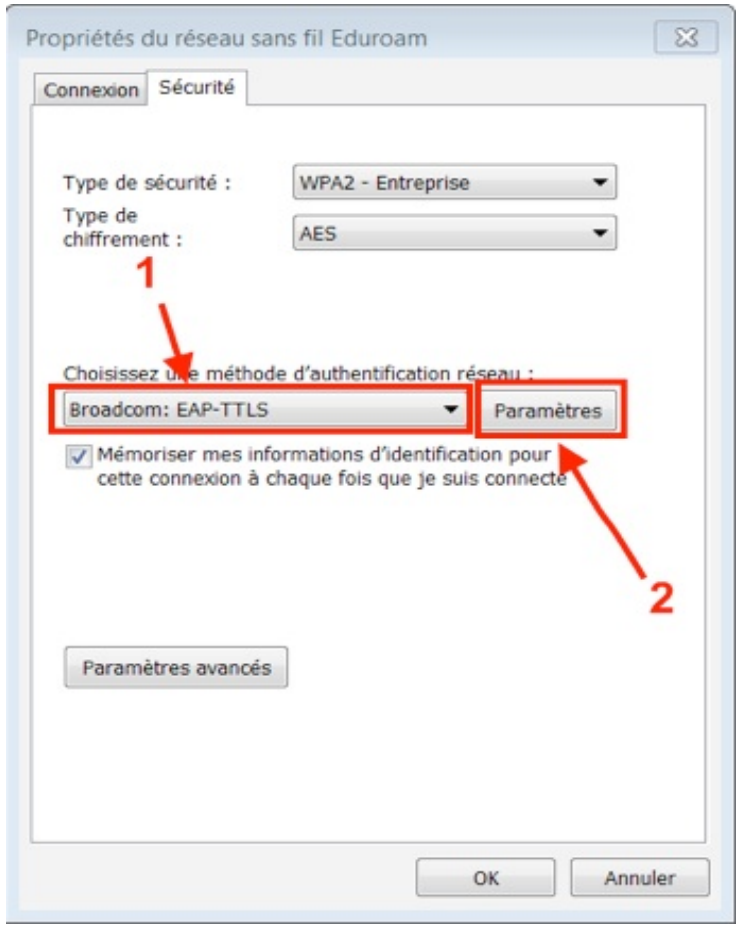
Configuration du protocole EAP (6)

- Allez dans ‘Sécurité’



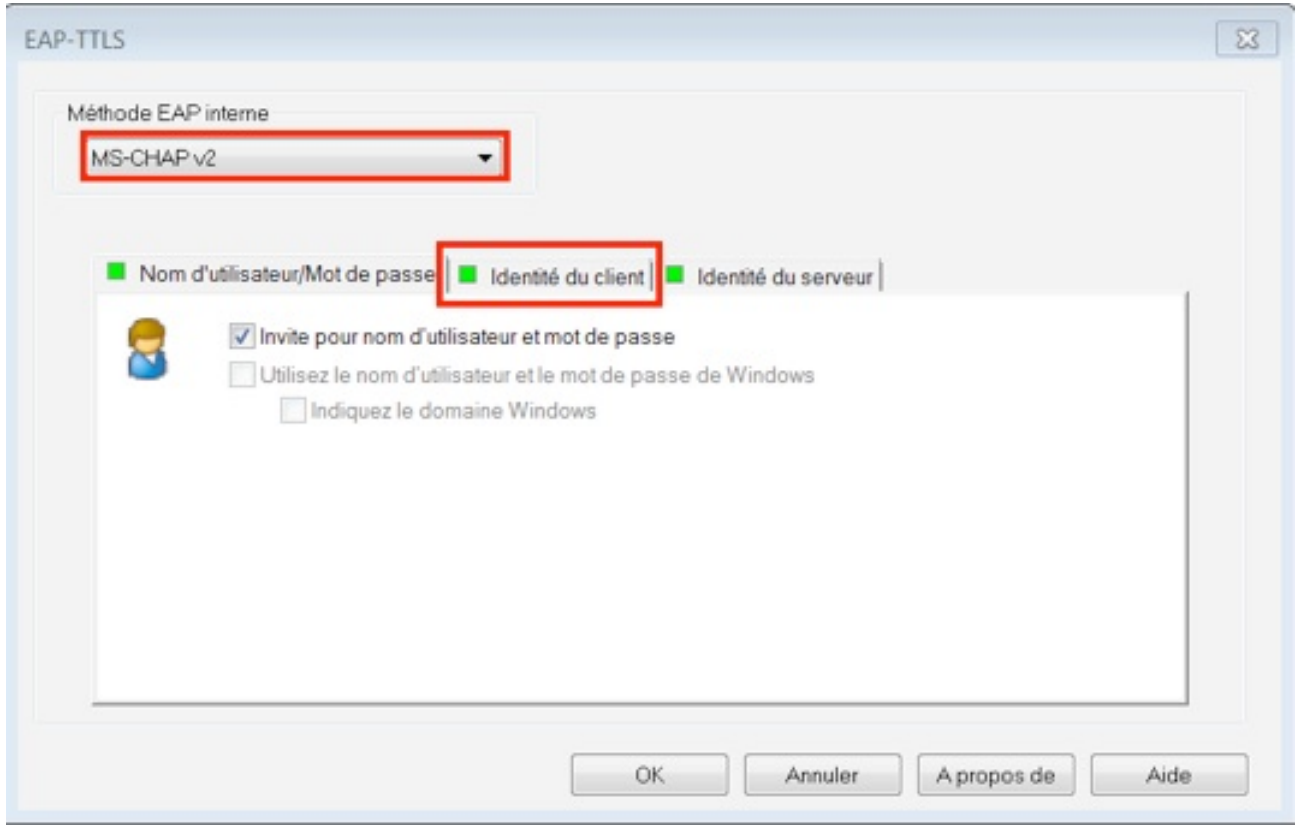
Configuration du protocole EAP (7)

Choisissez la méthode d'authentification réseaux 'Broadcom : EAP-TTLS' puis allez dans 'Paramètres'.



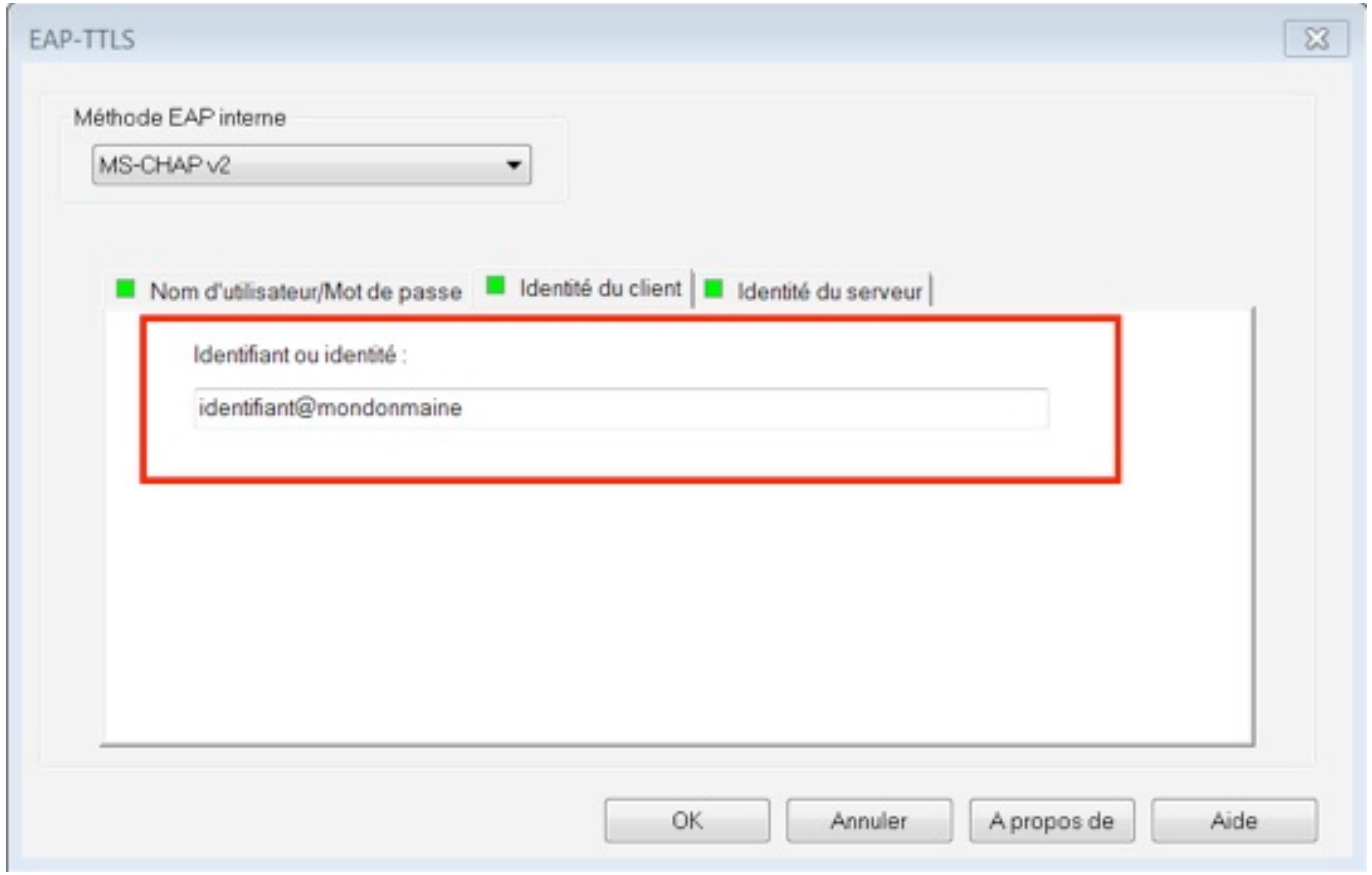
Configuration du protocole EAP (8)

Laissez les options par défaut et allez dans l'onglet 'Identité du client'.



Configuration du protocole EAP (9)

Renseignez le champs ‘Identifiant ou identité’ avec votre identifiant sous la forme ‘identifiant@mondomaine’.



Reference

- Dieter Gollmann "Computer Security" (3ème édition, mais 2ème est également bien)

Http://www.amazon.com/Computer-Security-Dieter-Gollmann/dp/0470741155

- Ross Anderson " Security Engineering "

Http://www.amazon.com/Security-Engineering-Building-Dependable-Distributed/dp/0470068523/

(Également disponible en ligne à: http://www.cl.cam.ac.uk/~rja14/book.html)

- Avoine, G., Junod, P., & Oechslin, P. (2015). Sécurité informatique-Cours et exercices corrigés. (3ème édition, mais 2ème est également bien)

Disponible à la bibliothèque de l'Université de Guelma