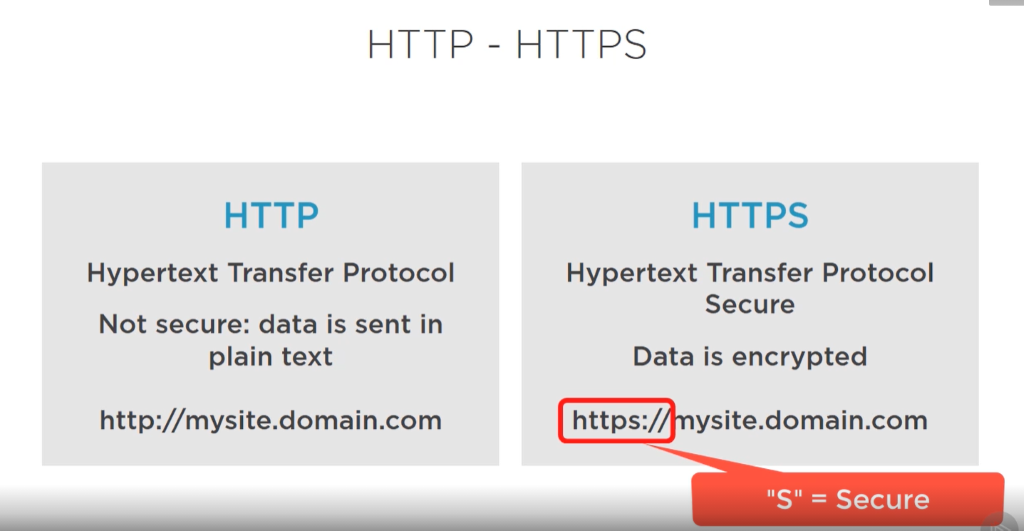


HTTP +SSL --> HTTPS





Une **suite cryptographique** (***cipher suite*** en anglais) est le nom de la combinaison des [algorithmes](https://fr.wikipedia.org/wiki/Algorithme) d'[échange de clés](https://fr.wikipedia.org/wiki/Cryptographie_asym%C3%A9trique#Fonctionnement_pratique), d'[authentification](https://fr.wikipedia.org/wiki/Authentification), de [chiffrement par bloc](https://fr.wikipedia.org/wiki/Chiffrement_par_bloc) et génération du [code d'authentification de message](https://fr.wikipedia.org/wiki/Code_d%27authentification_de_message) (MAC) utilisée afin d'établir le paramétrage de sécurité pour une connexion réseau utilisant le [protocole de communication](https://fr.wikipedia.org/wiki/Protocole_de_communication) [Secure Socket Layer](https://fr.wikipedia.org/wiki/Secure_Socket_Layer) (SSL) et [Transport Layer Security](https://fr.wikipedia.org/wiki/Transport_Layer_Security) (TLS).

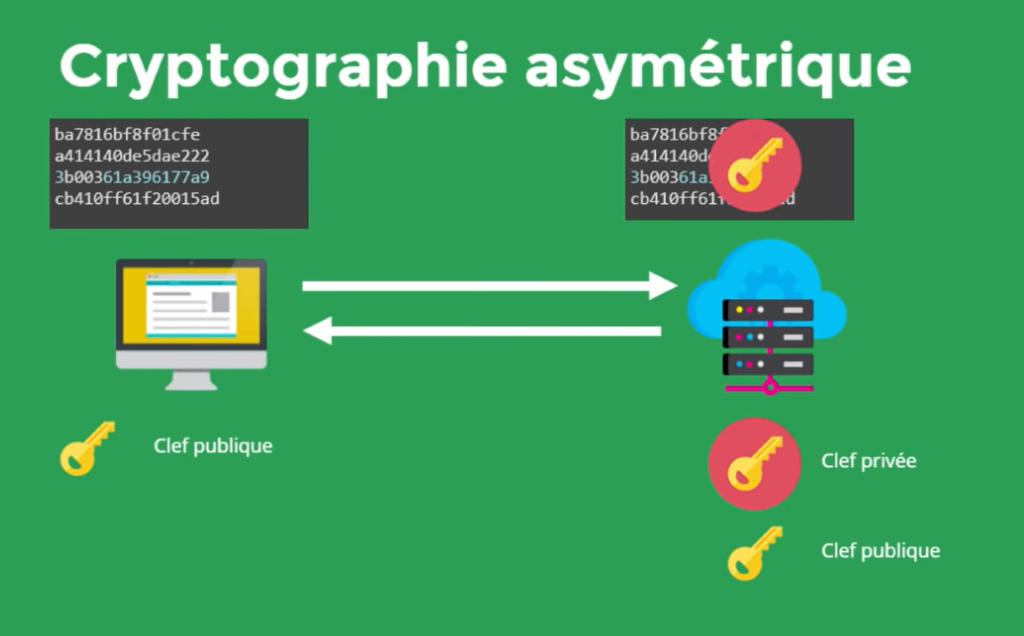
Lors de l'établissement d'une connexion TLS, un [*handshake*](https://fr.wikipedia.org/wiki/Handshaking)*s'effectue entre un client et un serveur. Au cours de ce processus, un message client hello* (ClientHello) ainsi qu'un message *server hello* (ServerHello) sont échangés[1](https://fr.wikipedia.org/wiki/Suite_cryptographique#cite_note-RFC-5246-r-t-d-p37-1). En premier lieu, le client émet au serveur la liste des suites cryptographiques qu'il supporte, par ordre de préférence (de la plus robuste à la plus faible). En retour, le serveur renvoie la suite cryptographique préférée pour l'établissement de la communication[2](https://fr.wikipedia.org/wiki/Suite_cryptographique#cite_note-RFC-5246-r-t-d-p40-2).

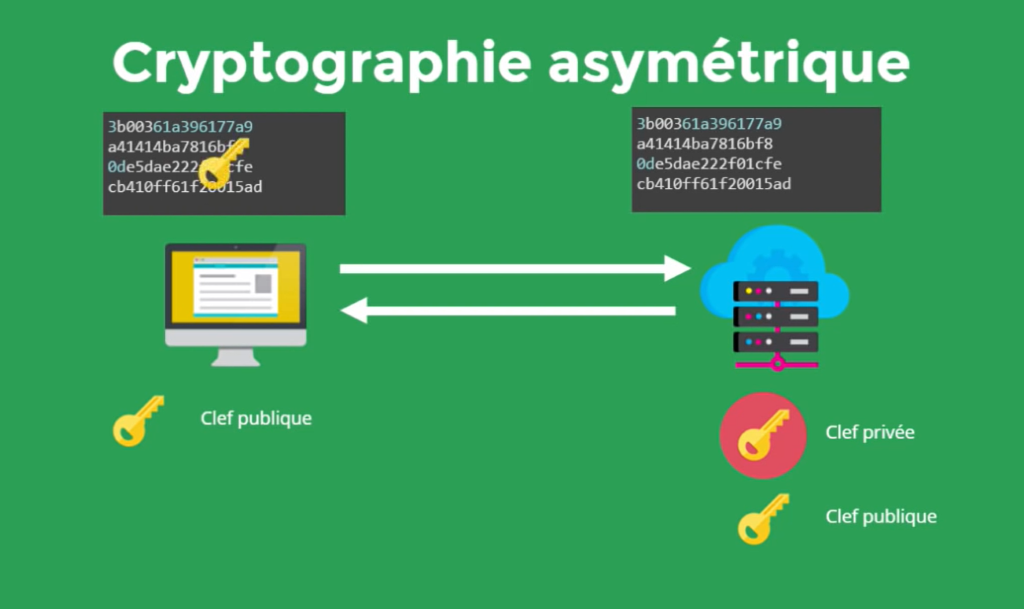
Il est possible d'obtenir la liste des suites cryptographiques supportées par un serveur à l'aide de logiciels scanner SSL/TLS (TLSSLed[3](https://fr.wikipedia.org/wiki/Suite_cryptographique#cite_note-3) ou TestSSLServer[4](https://fr.wikipedia.org/wiki/Suite_cryptographique#cite_note-4) par exemple).





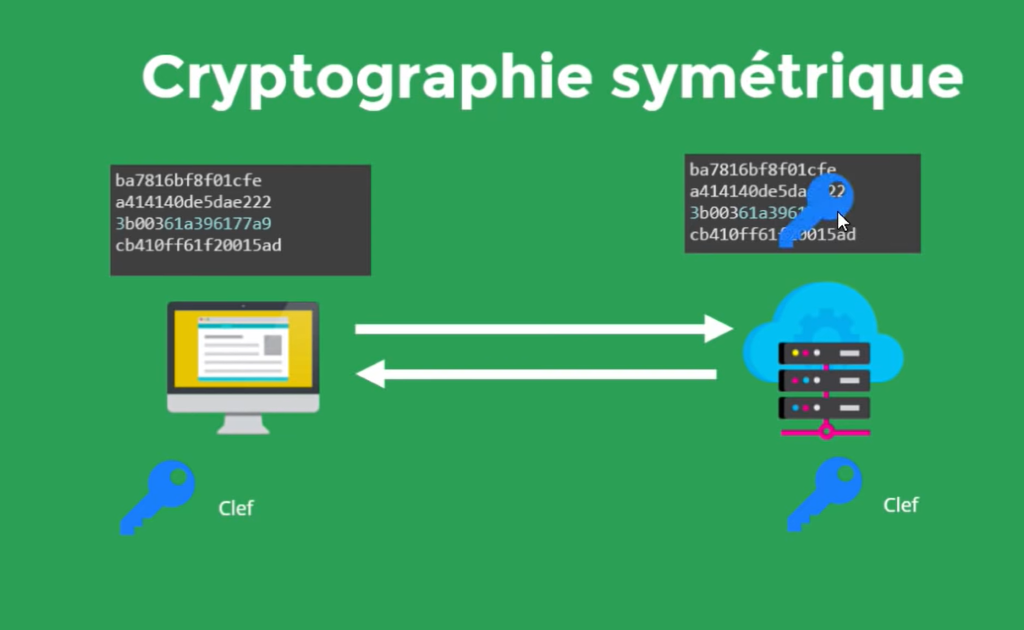
Le cryptage dans les deux sens:





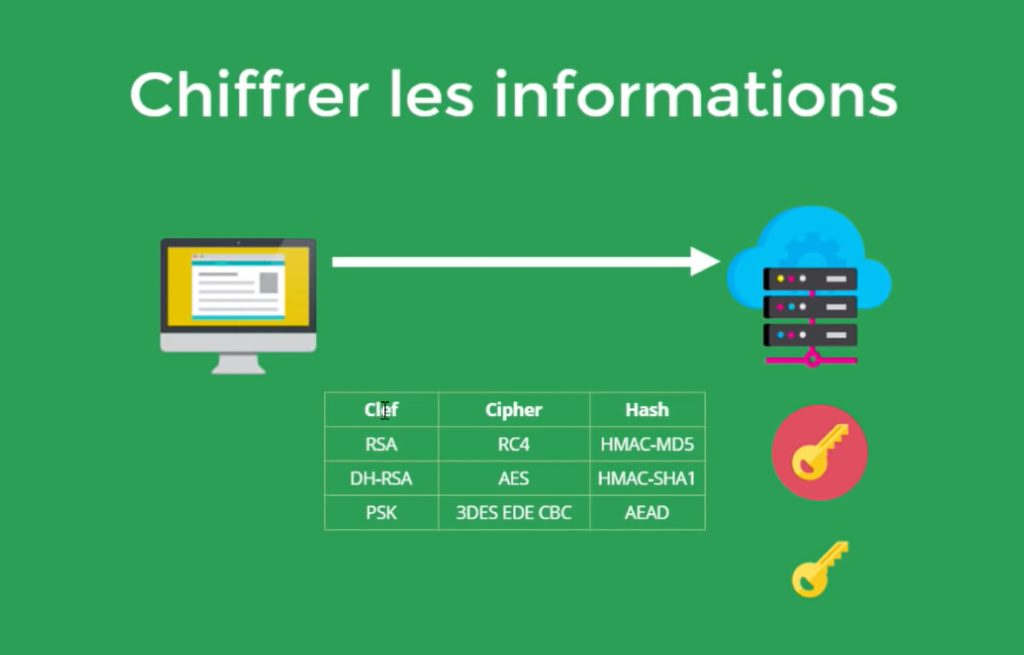
Cryptage symetrique: meme clé pour les deux extrémité --> plus sécurisé et plus performant

 --> problème: comment echanger cette clé?

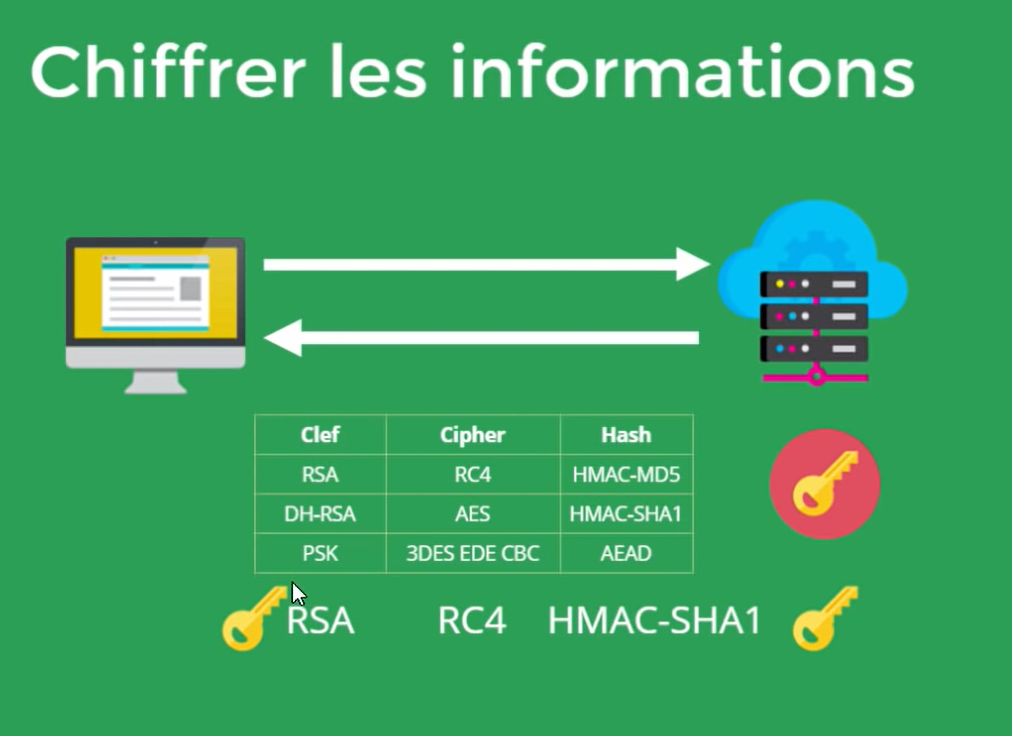


On premier lieu, le client entre en contacte avec le serveur et commence par envoyer les données suivantes:

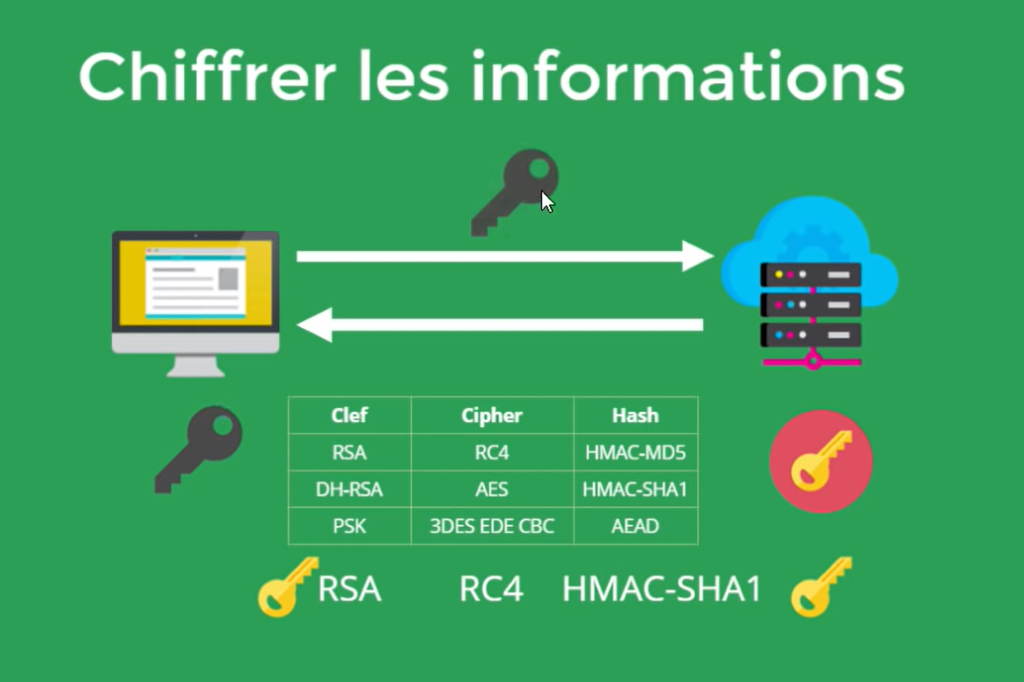
* Les types de clés qu'il supporte
* Les algorithmes de chiffrage supporté
* Les hash utilisés pour vérifier l'integrité des messages



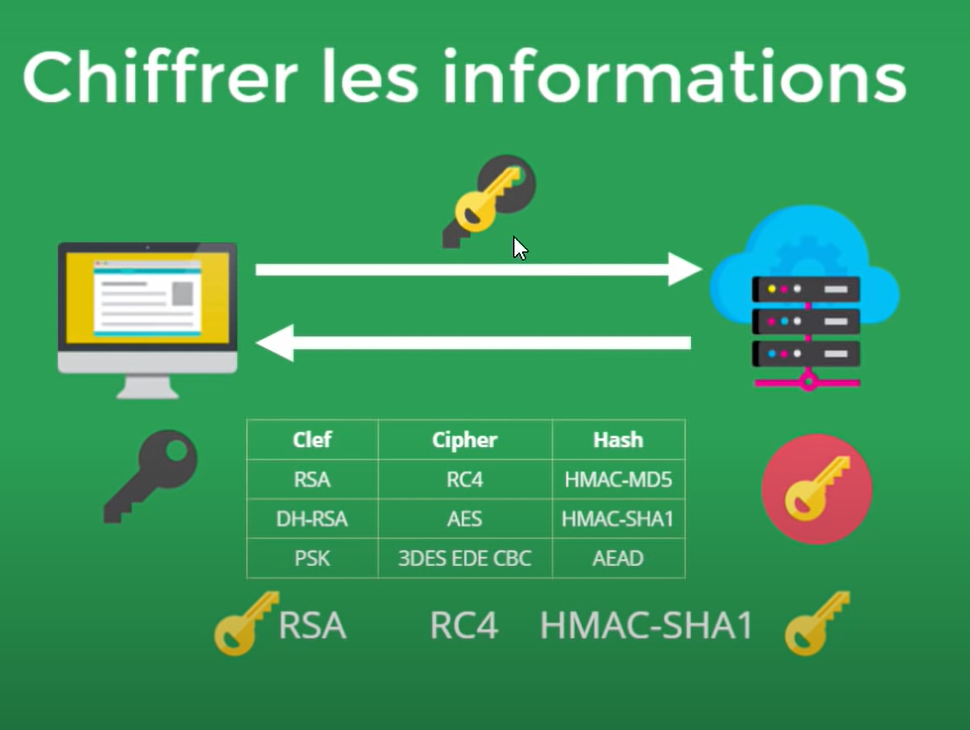
Le serveur choisi les élements qui lui convient selon un ordre de priorité et envoie son choix aux client ainsi que sa clé public



A la réception de ces informations, le client va générer un "premaster password" qui va l'utiliser pour générer la clé symétrique.  Après, il doit envoyer ce prémaster password au serveur pour que lui aussi génére la meme clé symetrique.

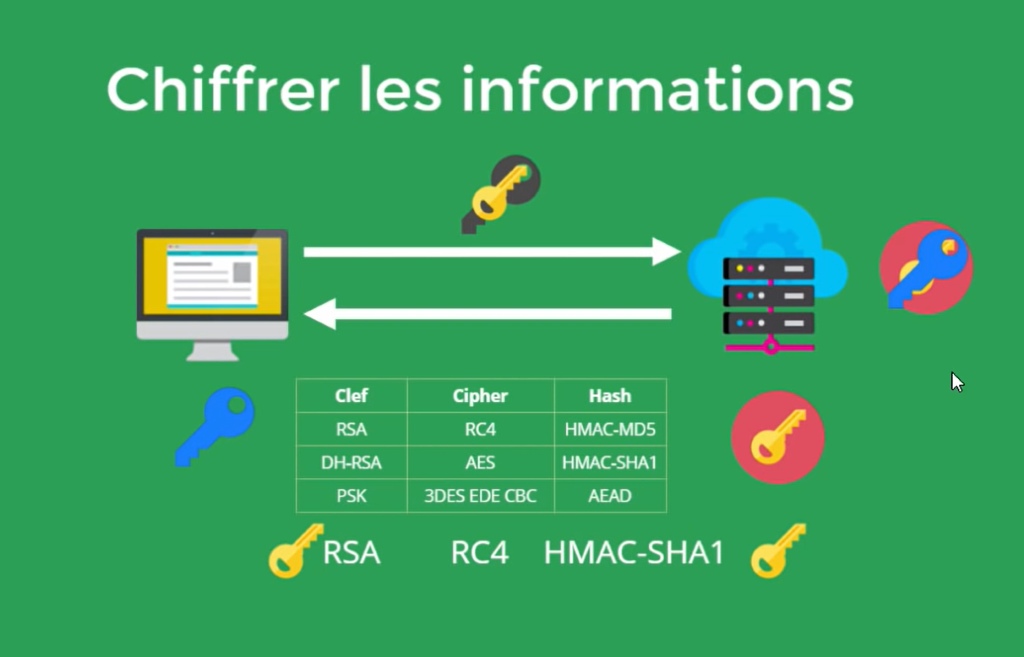


Le problème est qu'on peut pas envoyer ce premaster password en clair. On l'envoie alors chiffré par la clé public du serveur



A la reception, le serveur déchiffre le message avec sa clé privé et obtient le premaster password.

Finallement, on a la meme clé symétrique chez les deux cotés: serveur et client.



Comme ça on résoud le probléme de la clé de chiffrement. Cependant, on n'a pas encore résolut le problème d'idendité: n'importe qui peut fair semblant d'etre le serveur, genere une clé public et une clé privé et l'envoie au client.

 --> Solution : le certficat: elle fonctionne comme une carte d'identité des serveurs pour s'assurer qu'on parle avec la bonne personne.

Pour ce faire, on doit passer par une authorité de certificat



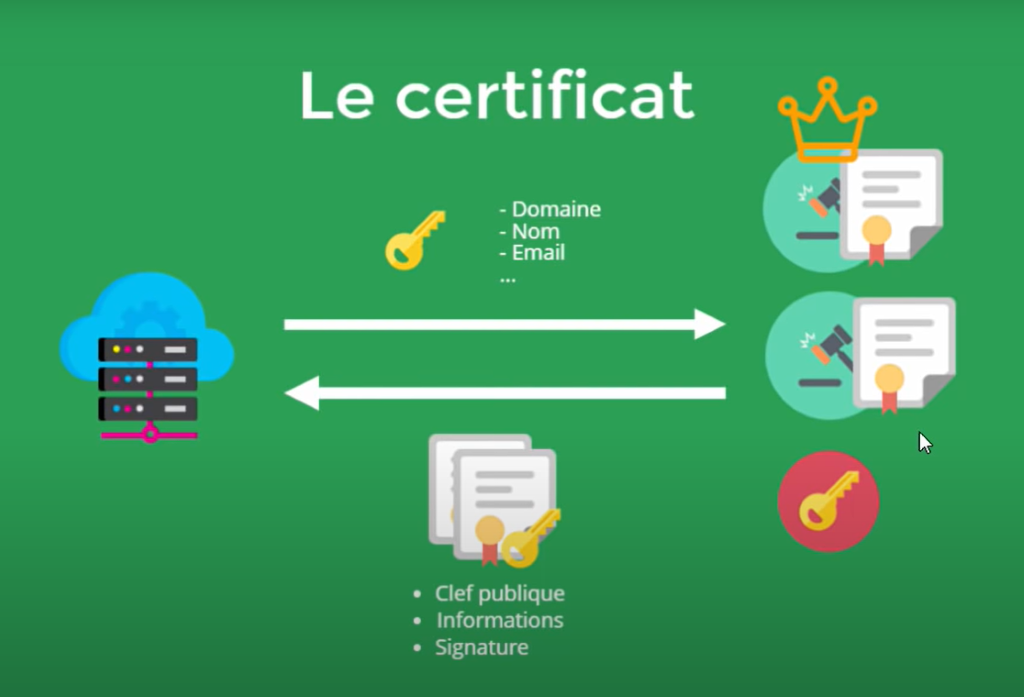
Après avoir reçu la demande de certificat envoyé par le serveur avec les informations necessaires qui l'identifie et sa clé publique, l'authorité de certificat va vérifier l'identité du serveur et si tout est bon elle lui envoie son certificat. Le certificat est composé des informations du serveur qu'il a envoyé en premier lieu, de la clé publique du serveur et de la signature.

La signature répresente la combinaison des données que le serveur a envoyé et de sa clé publique, le tout chiffrées par la clé privé de l'authorité de certificat.

Pour avoir le certificat,  on doit le déchiffré avec la clé publique de l'authorité de certificat.

NB: l'authorité de certificat peut aussi envoyer son pro pre certificat: un certificat intermediaire. En fait, on a plusieurs authorité de certificats actif. Pour cela, on distingue entre deux type d'authorité de certficat:

* Authorité de certificat racine: il s'occupe de donner les certificat pour les authorités de certificat intermédiaires.
* Authorité de certificat intermédiaire:  s'occupe de délivrer les certificats aux individues et aux entreprises.



Le système d'exploitation arrive généralement avec un ensemble de clé publiques des authorité de certificat racine

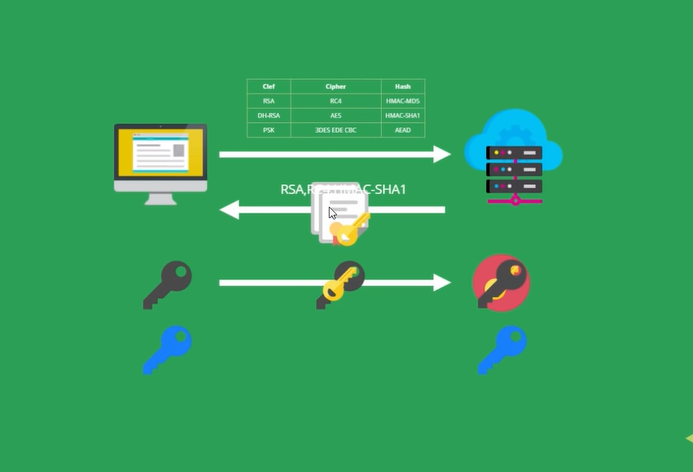
La chaine de confiance:

* Le client reçoi les deux certificationx
* Il utilise la clé publique de l'authorité de certificat racine pour déchiffrer la signature de la certification intermédiaire
* Une fois la signature est valide, on prend la clé publique de l'authorité de certificat intermédiare et on l'utilise pour déchiffrer la signature dans le certificat du serveur.
* Si on trouve la méme clé publique dans la signature alors on fait confiance au serveur et on
* Si quelqu'un intercepte le transfert du certificat, il peut modifier la clé publique en claire mais pas celle dans la signature vue que la signature est chiffré par la clé privé de l'authorité de certificat---> ainsi le client peut facilement détecter qu'il y a une menace et n'aura pas confiance au serveur.

---> Tout ça durant le handshake

En résumé:

On commence par le handshake comme ça:



Après on commence l'échange en utilisant la clé symétrique.