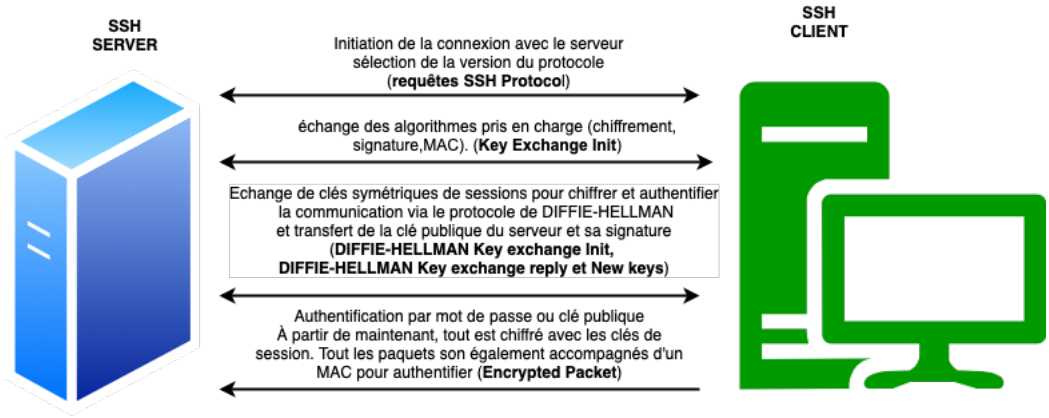
UN PEU PLUS LOIN SUR LE SSH

Processus Détaillé de Connexion SSH



	Time	Source	Destination	Protocol	Info
94	5…	192.168.1.39	37.187.123.212	SSHv2	Client: Protocol (SSH-2.0-OpenSSH_8.6)
98	5…	37.187.123.212	192.168.1.39	SSHv2	Server: Protocol (SSH-2.0-OpenSSH_7.2p2 Ubuntu-4ubuntu2.10)
100	5…	192.168.1.39	37.187.123.212	TCP	[TCP segment of a reassembled PDU]
101	5…	192.168.1.39	37.187.123.212	SSHv2	Client: Key Exchange Init
102	5…	37.187.123.212	192.168.1.39	SSHv2	Server: Key Exchange Init
105	6…	192.168.1.39	37.187.123.212	SSHv2	Client: Diffie-Hellman Key Exchange Init
106	6…	37.187.123.212	192.168.1.39	SSHv2	Diffie-Hellman Key Exchange Reply, New Keys
108	6…	192.168.1.39	37.187.123.212	SSHv2	Client: New Keys
119	6…	192.168.1.39	37.187.123.212	SSHv2	Client: Encrypted packet (len=44)
125	6…	192.168.1.39	37.187.123.212	SSHv2	Client: Encrypted packet (len=60)
129	6…	192.168.1.39	37.187.123.212	SSHv2	Client: Encrypted packet (len=500)
373	2…	192.168.1.39	37.187.123.212	SSHv2	Client: Encrypted packet (len=908)
376	2…	192.168.1.39	37.187.123.212	SSHv2	Client: Encrypted packet (len=112)
386	2…	192.168.1.39	37.187.123.212	SSHv2	Client: Encrypted packet (len=452)

Établissement de la Connexion

1. Initiation de la Connexion :
   - Le client SSH contacte le serveur SSH et demande une connexion.
   - Sélection de la version du protocole (Version 2)
2. Le client et le serveur envoient chacun un message "**Key Exchange Init**" pour indiquer les algorithmes pris en charge (algorithme de chiffrement, de signature et MC)

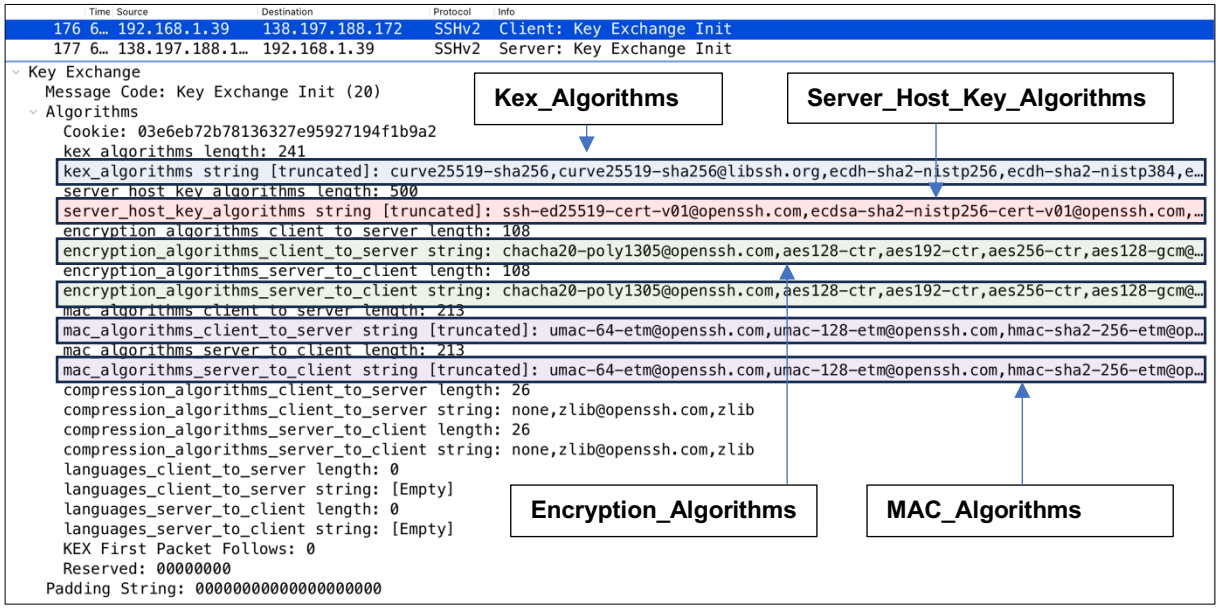
Kex_Algorithms (Algorithmes d'échange de clés) :
Signification : Les algorithmes d'échange de clés sont utilisés pour sécuriser l'établissement d'un secret partagé entre le client et le serveur sur un canal non sécurisé. Ce secret partagé est ensuite utilisé pour générer les clés de chiffrement symétrique nécessaires à la sécurisation de la session.
Exemples : Diffie-Hellman, Curve25519.

Server_Host_Key_Algorithms (Algorithmes de clé d'hôte du serveur)
Signification : Ces algorithmes sont utilisés pour l'authentification du serveur auprès du client. Le serveur prouve qu'il détient la clé privée correspondant à une clé publique connue du client, sans révéler sa clé privée. Cette étape est cruciale pour prévenir les attaques de type Man-in-the-Middle.

Exemples : RSA, ECDSA, ED25519.

Encryption_Algorithms (Algorithmes de chiffrement)
Signification : Une fois la session établie, ces algorithmes sont utilisés pour chiffrer les données échangées entre le client et le serveur, garantissant la confidentialité de la communication. Le client et le serveur utilisent les clés de chiffrement symétrique dérivées du secret partagé établi pendant l'échange de clés.
Exemples : AES (Advanced Encryption Standard) , CHACHA20-POLY1305.

MAC Algorithms (Algorithmes de Message Authentication Code)
Signification : Les algorithmes MAC sont utilisés pour garantir l'intégrité des données transmises et pour authentifier les messages. Un MAC est calculé pour chaque paquet de données en utilisant une clé secrète partagée, et le destinataire calcule et compare le MAC pour s'assurer que le message n'a pas été altéré.
Exemples : HMAC-SHA256, HMAC-SHA1, UMAC-64.



Il peut arriver que le serveur un peu ancien n'est disponible que des algorithmes un peu anciens. Dans ce cas-là, il prévient le client avec un message du type (cas d'un switch Cisco non mis à jour) :

Dans ce cas il est possible de spécifier au serveur des algorithmes non présentés dans le **key_exchange_init** pour satisfaire l'offre du serveur lors de la commande de connexion :

```
ssh -oKexAlgorithms=diffie-hellman-group1-sha1 -oCiphers=aes256-cbc -oHostKeyAlgorithms=ssh-rsa ciel@192.168.1.1
```

On spécifie ici l'algorithme d'échange de clé, l'algorithme de chiffrement et l'algorithme de signature de la clé publique.

- -oKexAlgorithms=diffie-hellman-group1-sha1 pour l'algorithme d'échange de clé
- -oCiphers=aes256-cbc pour l'algorithme de chiffrement
- -oHostKeyAlgorithms=ssh-rsa pour l'algorithme de signature de la clé publique

On peut également spécifier l'algorithme de MAC avec : -oMACs=xxxxxxxxx

Échange de Clés pour le Chiffrement de Session

1. **Échange de Clés Diffie-Hellman** :

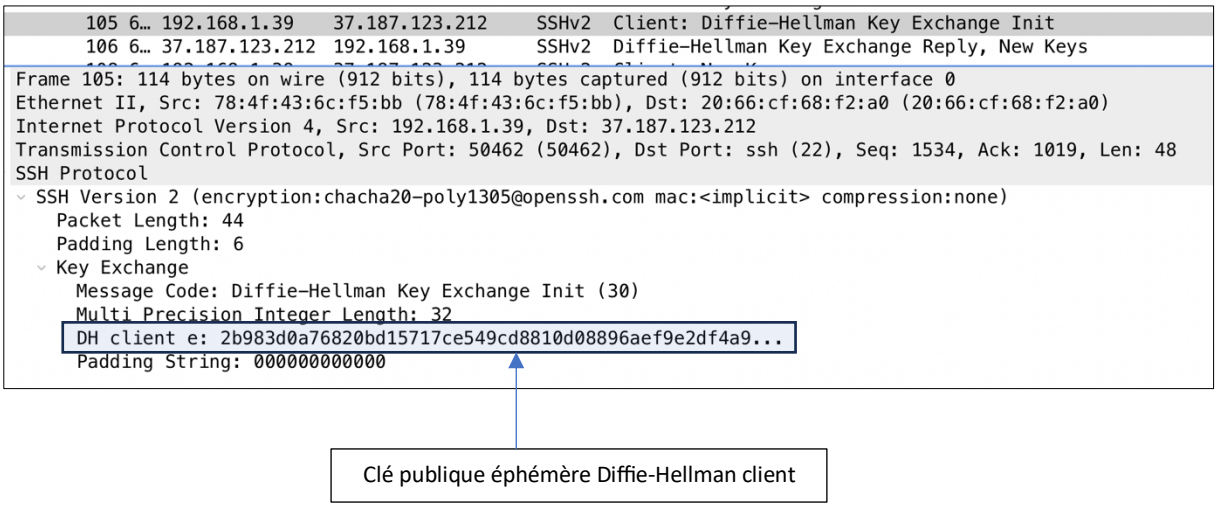
- Le client envoie sa clé publique éphémère Diffie-Hellman (**Diffie-Hellman Key Exchange Init**")
- Le serveur répond avec son propre message "**Diffie-Hellman Key Exchange Reply**", qui contient plusieurs éléments :
    - La clé publique éphémère Diffie-Hellman du serveur.
    - La clé publique d'authentification du serveur, qui est une clé publique plus permanente et est différente de la clé éphémère utilisée pour l'échange de clés Diffie-Hellman.
    - Une signature numérique qui permet au client de vérifier que la clé publique d'authentification du serveur est légitime et n'a pas été altérée. Cette signature est créée à partir d'un hachage de l'échange de clés Diffie-Hellman (y compris les clés publiques éphémères des deux parties) et est signée avec la clé privée d'authentification du serveur.

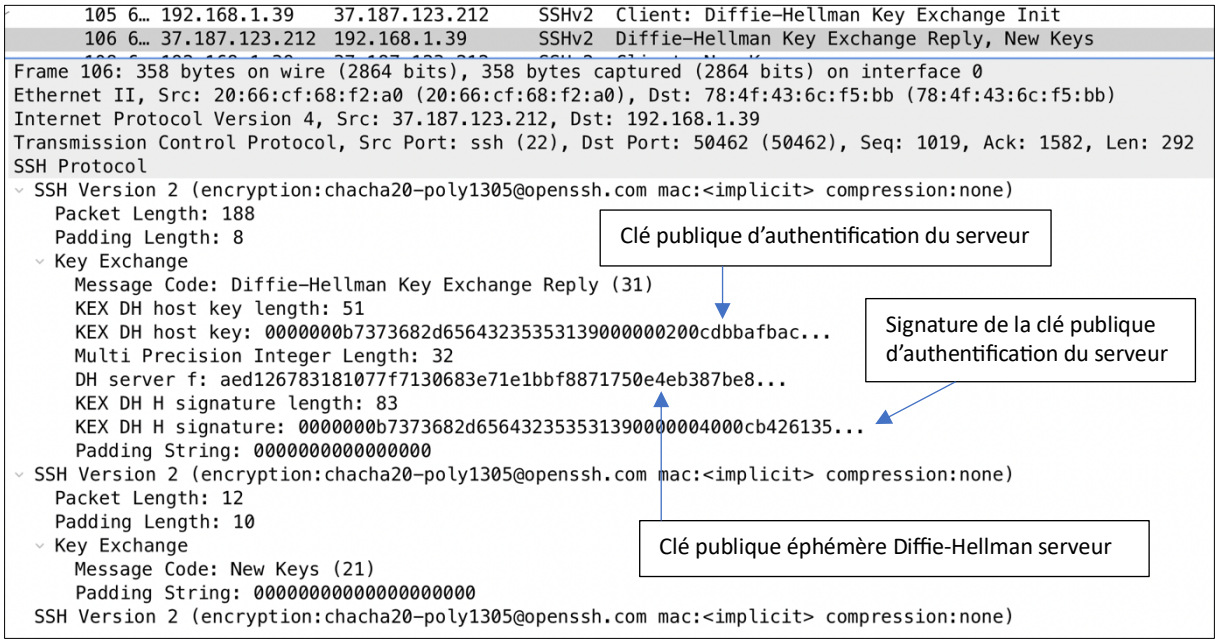
2. **Vérification par le Client** :

- Le client utilise la clé publique d'authentification du serveur, qui devrait être connue à l'avance ou stockée dans le fichier known_hosts du client, pour vérifier la signature numérique.
- Si la signature est valide, le client peut être sûr que la clé publique d'authentification appartient bien au serveur et que la connexion n'a pas été interceptée par un attaquant.

3. **Établissement du Secret Partagé** :

- Après avoir authentifié le serveur, le client et le serveur utilisent leurs clés privées éphémères Diffie-Hellman et les clés publiques éphémères de l'autre partie pour générer un secret partagé.
- Ce secret partagé est ensuite utilisé pour dériver les clés de session qui seront utilisées pour chiffrer et authentifier les données pour le reste de la session SSH.





Authentification :

1. Authentification :
   - Une fois que les clés de session sécurisée sont établies, le client s'identifie part mot de passe ou par clé publique.
   - Chaque paquet transmis est chiffré. Il est accompagné d'un MAC pour vérifier l'intégrité et l'authenticité du paquet.