KEX and Host Key Algorithms in SSH

565 Nesvin KN March 21, 2025 General 35440

What are KEX and Host Key Algorithms?

Overview: This article explains KEX and host key algorithms and guides Linux users on how to view and modify the algorithms used for SSH connections between a client and a server.

KEX: It is the short form of **Key Exchange.** The algorithm is chosen to compute the secret encryption key. Examples would be '**diffie-hellman-group-exchange-sha1**' and modern '**ecdh-sha2-nistp512**'.

Public_key or Server Host key: The asymmetric encryption algorithm used in the server's private-public host key pair. Examples would be '**ssh-rsa**' and elliptic curve '**ecdsa-sha2-nistp521**'.

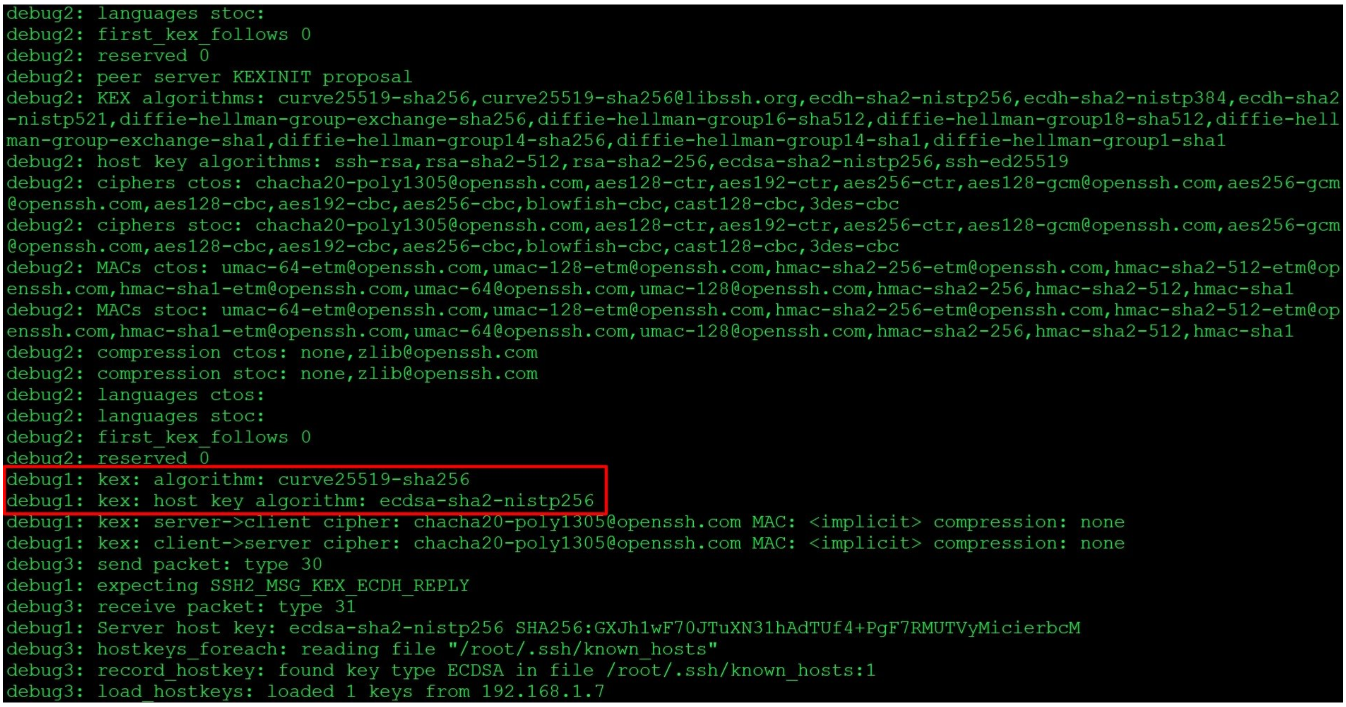
1. How to find the KEX (Key Exchange) and Host Key Algorithms in SSH?

Step 1(A): SSH from one linux machine to another in verbose mode to get the detailed process.

```
root@linux_server ~]# ssh username@linux_server_IP -vvv
```

Step 1(B): KEX and host key algorithms used to SSH can be found in **debug 1 level logs**. Refer below example of **KEX and host key algorithms.**

```
debug1: kex: algorithm: curve25519-sha256
debug1: kex: host key algorithm: ecdsa-sha2-nistp256
```



```
debug2: languages stoc:
debug2: first_kex_follows 0
debug2: reserved 0
debug2: peer server KEXINIT proposal
debug2: KEX algorithms: curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2
-nistp521,diffie-hellman-group-exchange-sha256,diffie-hellman-group16-sha512,diffie-hellman-group18-sha512,diffie-hell
man-group-exchange-sha1,diffie-hellman-group14-sha256,diffie-hellman-group14-sha1,diffie-hellman-group1-sha1
debug2: host key algorithms: ssh-rsa,rsa-sha2-512,rsa-sha2-256,ecdsa-sha2-nistp256,ssh-ed25519
debug2: ciphers ctos: chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm
@openssh.com,aes128-cbc,aes192-cbc,aes256-cbc,blowfish-cbc,cast128-cbc,3des-cbc
debug2: ciphers stoc: chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm
@openssh.com,aes128-cbc,aes192-cbc,aes256-cbc,blowfish-cbc,cast128-cbc,3des-cbc
debug2: MACs ctos: umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@op
enssh.com,hmac-sha1-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1
debug2: MACs stoc: umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@op
enssh.com,hmac-sha1-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1
debug2: compression ctos: none,zlib@openssh.com
debug2: compression stoc: none,zlib@openssh.com
debug2: languages ctos:
debug2: languages stoc:
debug2: first_kex_follows 0
debug2: reserved 0
debug1: kex: algorithm: curve25519-sha256
debug1: kex: host key algorithm: ecdsa-sha2-nistp256
debug1: kex: server->client cipher: chacha20-poly1305@openssh.com MAC: <implicit> compression: none
debug1: kex: client->server cipher: chacha20-poly1305@openssh.com MAC: <implicit> compression: none
debug3: send packet: type 30
debug1: expecting SSH2_MSG_KEX_ECDH_REPLY
debug3: receive packet: type 31
debug1: Server host key: ecdsa-sha2-nistp256 SHA256:GXJh1wF70JTuXN31hAdTUf4+PgF7RMUTVyMicierbcM
debug3: hostkeys_foreach: reading file "/root/.ssh/known_hosts"
debug3: record_hostkey: found key type ECDSA in file /root/.ssh/known_hosts:1
debug3: load_hostkeys: loaded 1 keys from 192.168.1.7
```

2. How to change the KEX and host key algorithm on the server machine(the machine you are connecting to from the client)?

Step 2(A): Edit the sshd_config file on the server machine (the machine you are connecting to from the client) and add the following lines to specify KexAlgorithms and HostKeyAlgorithms.

```
root@linux_server ~]# vim /etc/ssh/sshd_config

KexAlgorithms diffie-hellman-group16-sha512
HostKeyAlgorithms rsa-sha2-512
```

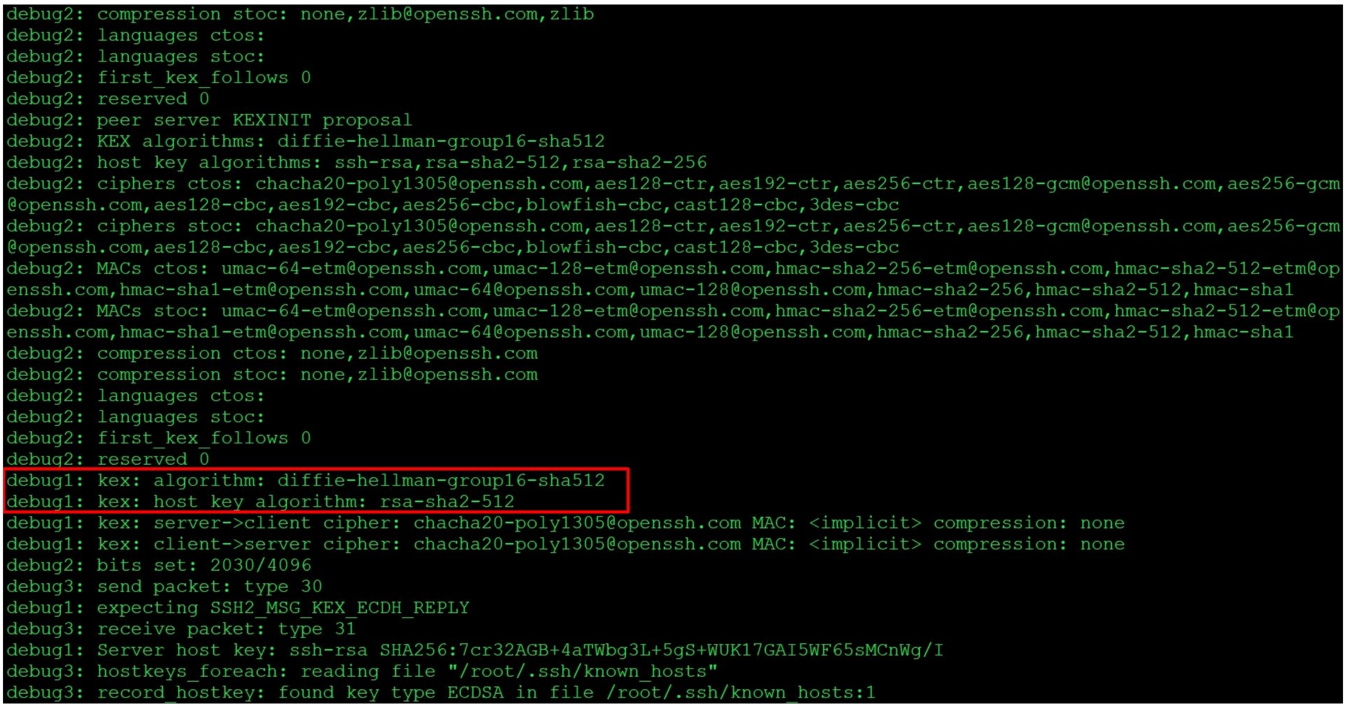
Step 2(B): Restart the SSHD service to apply the changes made in **sshd_config**.

```
root@linux_server ~]# systemctl restart sshd
```

Step 2(C): SSH from the client machine to the server machine to view the changed **KEX and host key algorithms**.

```
root@linux_server ~]# ssh username@linux_machine_IP -vvv


debug1: kex: algorithm: diffie-hellman-group16-sha512
debug1: kex: host key algorithm: rsa-sha2-512
```



```
debug2: compression stoc: none,zlib@openssh.com,zlib
debug2: languages ctos:
debug2: languages stoc:
debug2: first_kex_follows 0
debug2: reserved 0
debug2: peer server KEXINIT proposal
debug2: KEX algorithms: diffie-hellman-group16-sha512
debug2: host key algorithms: ssh-rsa,rsa-sha2-512,rsa-sha2-256
debug2: ciphers ctos: chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm
@openssh.com,aes128-cbc,aes192-cbc,aes256-cbc,blowfish-cbc,cast128-cbc,3des-cbc
debug2: ciphers stoc: chacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,aes256-gcm
@openssh.com,aes128-cbc,aes192-cbc,aes256-cbc,blowfish-cbc,cast128-cbc,3des-cbc
debug2: MACs ctos: umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@op
enssh.com,hmac-sha1-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1
debug2: MACs stoc: umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-etm@op
enssh.com,hmac-sha1-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sha1
debug2: compression ctos: none,zlib@openssh.com
debug2: compression stoc: none,zlib@openssh.com
debug2: languages ctos:
debug2: languages stoc:
debug2: first_kex_follows 0
debug2: reserved 0
debug1: kex: algorithm: diffie-hellman-group16-sha512
debug1: kex: host key algorithm: rsa-sha2-512
debug1: kex: server->client cipher: chacha20-poly1305@openssh.com MAC: <implicit> compression: none
debug1: kex: client->server cipher: chacha20-poly1305@openssh.com MAC: <implicit> compression: none
debug2: bits set: 2030/4096
debug3: send packet: type 30
debug1: expecting SSH2_MSG_KEX_ECDH_REPLY
debug3: receive packet: type 31
debug1: Server host key: ssh-rsa SHA256:7cr32AGB+4aTWbg3L+5gS+WUK17GAI5WF65sMCnWg/I
debug3: hostkeys_foreach: reading file "/root/.ssh/known_hosts"
debug3: record_hostkey: found key type ECDSA in file /root/.ssh/known_hosts:1
```

3. How to view the list of KEX and Keys in the Linux server?

Step 3(A): Run below command to list **keys** in the Linux server.

```
root@linux_server ~]# ssh -Q key

ssh-ed25519
ssh-ed25519-cert-v01@openssh.com
ssh-rsa
ssh-dss
ecdsa-sha2-nistp256
ecdsa-sha2-nistp384
ecdsa-sha2-nistp521
ssh-rsa-cert-v01@openssh.com
ssh-dss-cert-v01@openssh.com
ecdsa-sha2-nistp256-cert-v01@openssh.com
ecdsa-sha2-nistp384-cert-v01@openssh.com
ecdsa-sha2-nistp521-cert-v01@openssh.com
```

Step 3(B): Run below command to list **KEX** in the Linux server.

```
root@linux_server ~]# ssh -Q kex

diffie-hellman-group1-sha1
diffie-hellman-group14-sha1
diffie-hellman-group14-sha256
diffie-hellman-group16-sha512
diffie-hellman-group18-sha512
diffie-hellman-group-exchange-sha1
diffie-hellman-group-exchange-sha256
ecdh-sha2-nistp256
ecdh-sha2-nistp384
ecdh-sha2-nistp521
curve25519-sha256
curve25519-sha256@libssh.org
```

Related Articles:

DSA key based authentication is not working

signature algorithm ssh-rsa not in PubkeyAcceptedAlgorithms

signature algorithm ssh-dss not in PubkeyAcceptedAlgorithms

Online URL: https://www.ezeelogin.com/kb/article/kex-and-host-key-algorithms-in-ssh-565.html