# SSH Client Configuration Mode Commands

The Secure Shell Client Configuration Mode manages SSH client key pairs that support secure access with external servers.

**Command Modes**

Exec > Global Configuration > Context Configuration > SSH Configuration

**configure > client ssh**

Entering the above command sequence results in the following prompt:

```
[local]host_name(config-ssh)#
```

**Important**

The commands or keywords/variables that are available are dependent on platform type, product version, and installed license(s).



# ciphers

Configures the cipher priority list in SSH client symmetric encryption that is used to generate an SSH client key pair.

**Product**

All

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > Global Configuration > SSH Client Configuration

**configure > client ssh**

Entering the above command results in the following prompt:

```
[context_name]host_name(config-ssh)#
```

## Syntax Description

**[ default ] ciphers** *algorithm*

### default

Resets the value of *algorithm* in a Normal build to:

`aes256-ctr,aes192-ctr,aes128-ctr,aes256-gcm@openssh.com,aes128-gcm@openssh.com,chacha20-poly1305@openssh.com,blowfish-cbc,3des-cbc,aes128-cbc`

Resets the value of *algorithm* in a Trusted build to:

`aes256-ctr,aes192-ctr,aes128-ctr`

### *algorithm*

Specifies the algorithm(s) to be used as a single string of comma-separated variables (no spaces) in priority order from those shown below:

- **blowfish-cbc** – symmetric-key block cipher, Cipher Block Chaining, CBC
- **3des-cbc** – Triple Data Encryption Standard, CBC
- **aes128-cbc** – Advanced Encryption Standard, 128-bit key size, CBC
- **aes128-ctr** –Advanced Encryption Standard, 128-bit key size, Counter-mode encryption, CTR
- **aes192-ctr** – Advanced Encryption Standard, 192-bit key size, CTR
- **aes256-ctr** – Advanced Encryption Standard, 256-bit key size, CTR
- **aes128-gcm@openssh.com** – Advanced Encryption Standard, 128-bit key size, Galois Counter Mode [GCM], OpenSSH
- **aes256-gcm@openssh.com** – Advanced Encryption Standard, 256-bit key size, GCM, OpenSSH
- **chacha20-poly1305@openssh.com** – ChaCha20 symmetric cipher, Poly1305 cryptographic Message Authentication Code [MAC], OpenSSH

*algorithm* is a string of 1 through 511 alphanumeric characters.

**Important** For release 20.0 and higher Trusted builds, only the AES128-CTR, AES-192-CTR and AES-256CTR ciphers are available.

## Usage Guidelines

Use this command to configure the cipher priority list for SSH client symmetric encryption that is used to generate an SSH client key pair.

### Example

The following command sets the supported SSH algorithms and their priority.

```
ciphers blowfish-cbc,aes128-cbc,aes128-ctr,aes192-ctr,aes256-ctr
```

# do show

Executes all **show** commands while in Configuration mode.

**Product** All

**Privilege** Security Administrator, Administrator

**Syntax Description**

```
do show
```

**Usage Guidelines** Use this command to run all Exec mode **show** commands while in Configuration mode. It is not necessary to exit the Config mode to run a **show** command.

The pipe character **|** is only available if the command is valid in the Exec mode.

**Caution** There are some Exec mode **show** commands which are too resource intensive to run from Config mode. These include: **do show support collection**, **do show support details**, **do show support record** and **do show support summary**. If there is a restriction on a specific **show** command, the following error message is displayed:

```
Failure: Cannot execute 'do show support' command from Config mode.
```

# end

Exits the current configuration mode and returns to the Exec mode.

**Product** All

**Privilege** Security Administrator, Administrator

**Syntax Description**

```
end
```

**Usage Guidelines** Use this command to return to the Exec mode.

# exit

Exits the current mode and returns to the parent configuration mode.

**Product** All

**Privilege** Security Administrator, Administrator

**Syntax Description**

```
exit
```

**Usage Guidelines** Use this command to return to the parent configuration mode.

# preferredauthentications

Specifies the order in which the client should try SSH client protocol authentication methods. This allows the client to prioritize one method over another method – public key, keyboard-interactive and password.

**Product**

All

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > Global Configuration > SSH Client Configuration

**configure > client ssh**

Entering the above command results in the following prompt:

`[context_name]host_name(config-ssh)#`

**Syntax Description**

**`[ default ] preferredauthentications`** *`methods`*

**default**

Resets the value of *methods* to:

`publickey,password`

***methods***

Specifies the preferred methods of authentication to be used as a single string of comma-separated variables (no spaces) in priority order (left to right) from those shown below:

- **publickey** – authentication via SSH v2-RSA protocol.
- **keyboard-interactive** – request for an arbitrary number of pieces of information. For each piece of information the server sends the label of the prompt.
- **password** – simple request for a single password

**Usage Guidelines**

Use this command to specify the order in which the client should try SSH client protocol authentication methods. This allows the client to prioritize one method over another method – public key, keyboard-interactive and password.

**Example**

The following command sets the supported SSH authentication protocols and their priority.

```
preferredauthentications publickey,keyboard-interactive,password
```

# ssh

Allows you to specify SSH client key parameters and generate an SSH client key pair.

**Product**

All

**Privilege**

Security Administrator, Administrator

**Command Modes**

Exec > Global Configuration > SSH Client Configuration

**configure > client ssh**

Entering the above command results in the following prompt:

```
[context_name]host_name(config-ssh)#
```

**Syntax Description**

**ssh { generate key | key** *private_key_string* **length** *length_value* **} [ type v2-rsa ] [ key-size { 2048 | 3072 | 4096 | 5120 | 6144 | 7168 | 9216 } ]**
**no ssh key type v2-rsa**

**no ssh key**

Removes the specified SSH client key configuration. The only supported SSH client key type is V2-RSA.

**generate key [ type v2-rsa ]**

Generates SSH client key pairs based on parameters specified via the **ssh key** command. The only supported SSH client key type is V2-RSA.

**key *private_key_string* length *length_value* [ type v2-rsa ]**

Sets parameters for the SSH client keys.

- **key** *private_key_string* specifies a private key value as an alphanumeric string of 1 through 4499 characters.
- **length** *key_length* specifies the length of the key in bytes as an integer from 0 through 65535.
- **type v2-rsa** specifies the SSH client key type. The only supported SSH client key type is V2-RSA.

**key-size { 2048 | 3072 | 4096 | 5120 | 6144 | 7168 | 9216 }**

Specifies the key size for SSH client.

**Usage Guidelines**

Use this command to specify SSH client private key values or generate an SSH client key pair. You can then push the public key to external servers via the Exec mode **push ssh-key** command. Pushing the key supports SSH access without a password between the StarOS gateway and external servers.

**Example**

The following command sequence specifies a private key and generates an SSH client key pair.

```
ssh key
AAAAB3NzaC1yc2EAAAADAQABAAABAQDn0X5xmZ1BrK2sEvzS+CRvD8mwOKHxb8Nwq64sunvjzcdc
length 512 type v2-rsa
```

```
ssh generate key type v2-rsa
```