

NAME

ssh-add - adds private key identities to the OpenSSH authentication agent

SYNOPSIS

```
ssh-add [-cDdKkLlqvXx] [-E fingerprint_hash] [-H hostkey_file] [-h destination_constraint]
        [-S provider] [-t life] [file ...]
ssh-add -s pkcs11
ssh-add -e pkcs11
ssh-add -T pubkey ...
```

DESCRIPTION

ssh-add adds private key identities to the authentication agent, `ssh-agent(1)`. When run without arguments, it adds the files `~/.ssh/id_rsa`, `~/.ssh/id_ecdsa`, `~/.ssh/id_ecdsa_sk`, `~/.ssh/id_ed25519`, `~/.ssh/id_ed25519_sk`, and `~/.ssh/id_dsa`. After loading a private key, **ssh-add** will try to load corresponding certificate information from the filename obtained by appending `-cert.pub` to the name of the private key file. Alternative file names can be given on the command line.

If any file requires a passphrase, **ssh-add** asks for the passphrase from the user. The passphrase is read from the user's tty. **ssh-add** retries the last passphrase if multiple identity files are given.

The authentication agent must be running and the `SSH_AUTH_SOCK` environment variable must contain the name of its socket for **ssh-add** to work.

The options are as follows:

- c** Indicates that added identities should be subject to confirmation before being used for authentication. Confirmation is performed by `ssh-askpass(1)`. Successful confirmation is signaled by a zero exit status from `ssh-askpass(1)`, rather than text entered into the requester.
- D** Deletes all identities from the agent.
- d** Instead of adding identities, removes identities from the agent. If **ssh-add** has been run without arguments, the keys for the default identities and their corresponding certificates will be removed. Otherwise, the argument list will be interpreted as a list of paths to public key files to specify keys and certificates to be removed from the agent. If no public key is found at a given path, **ssh-add** will append `.pub` and retry. If the argument list consists of "-" then **ssh-add** will read public keys to be removed from standard input.
- E *fingerprint_hash***
Specifies the hash algorithm used when displaying key fingerprints. Valid options are: "md5"

and "sha256". The default is "sha256".

-e *pkcs11*

Remove keys provided by the PKCS#11 shared library *pkcs11*.

-H *hostkey_file*

Specifies a known hosts file to look up hostkeys when using destination-constrained keys via the **-h** flag. This option may be specified multiple times to allow multiple files to be searched.

If no files are specified, **ssh-add** will use the default `ssh_config(5)` known hosts files:

~/.ssh/known_hosts, *~/.ssh/known_hosts2*, */etc/ssh/ssh_known_hosts*, and */etc/ssh/ssh_known_hosts2*.

-h *destination_constraint*

When adding keys, constrain them to be usable only through specific hosts or to specific destinations.

Destination constraints of the form ‘[user@]dest-hostname’ permit use of the key only from the origin host (the one running `ssh-agent(1)`) to the listed destination host, with optional user name.

Constraints of the form ‘src-hostname>[user@]dst-hostname’ allow a key available on a forwarded `ssh-agent(1)` to be used through a particular host (as specified by ‘src-hostname’) to authenticate to a further host, specified by ‘dst-hostname’.

Multiple destination constraints may be added when loading keys. When attempting authentication with a key that has destination constraints, the whole connection path, including `ssh-agent(1)` forwarding, is tested against those constraints and each hop must be permitted for the attempt to succeed. For example, if key is forwarded to a remote host, ‘host-b’, and is attempting authentication to another host, ‘host-c’, then the operation will be successful only if ‘host-b’ was permitted from the origin host and the subsequent ‘host-b>host-c’ hop is also permitted by destination constraints.

Hosts are identified by their host keys, and are looked up from known hosts files by **ssh-add**. Wildcards patterns may be used for hostnames and certificate host keys are supported. By default, keys added by **ssh-add** are not destination constrained.

Destination constraints were added in OpenSSH release 8.9. Support in both the remote SSH client and server is required when using destination-constrained keys over a forwarded `ssh-agent(1)` channel.

It is also important to note that destination constraints can only be enforced by `ssh-agent(1)` when a key is used, or when it is forwarded by a **cooperating** `ssh(1)`. Specifically, it does not prevent an attacker with access to a remote `SSH_AUTH_SOCK` from forwarding it again and using it on a different host (but only to a permitted destination).

- K** Load resident keys from a FIDO authenticator.
- k** When loading keys into or deleting keys from the agent, process plain private keys only and skip certificates.
- L** Lists public key parameters of all identities currently represented by the agent.
- l** Lists fingerprints of all identities currently represented by the agent.
- q** Be quiet after a successful operation.
- S *provider***
Specifies a path to a library that will be used when adding FIDO authenticator-hosted keys, overriding the default of using the internal USB HID support.
- s *pkcs11***
Add keys provided by the PKCS#11 shared library *pkcs11*.
- T *pubkey ...***
Tests whether the private keys that correspond to the specified *pubkey* files are usable by performing sign and verify operations on each.
- t *life*** Set a maximum lifetime when adding identities to an agent. The lifetime may be specified in seconds or in a time format specified in `sshd_config(5)`.
- v** Verbose mode. Causes **ssh-add** to print debugging messages about its progress. This is helpful in debugging problems. Multiple **-v** options increase the verbosity. The maximum is 3.
- X** Unlock the agent.
- x** Lock the agent with a password.

ENVIRONMENT

`DISPLAY`, `SSH_ASKPASS` and `SSH_ASKPASS_REQUIRE`

If **ssh-add** needs a passphrase, it will read the passphrase from the current terminal if it was run

from a terminal. If **ssh-add** does not have a terminal associated with it but `DISPLAY` and `SSH_ASKPASS` are set, it will execute the program specified by `SSH_ASKPASS` (by default "ssh-askpass") and open an X11 window to read the passphrase. This is particularly useful when calling **ssh-add** from a *.xsession* or related script.

`SSH_ASKPASS_REQUIRE` allows further control over the use of an askpass program. If this variable is set to "never" then **ssh-add** will never attempt to use one. If it is set to "prefer", then **ssh-add** will prefer to use the askpass program instead of the TTY when requesting passwords. Finally, if the variable is set to "force", then the askpass program will be used for all passphrase input regardless of whether `DISPLAY` is set.

SSH_AUTH_SOCK

Identifies the path of a UNIX-domain socket used to communicate with the agent.

SSH_SK_PROVIDER

Specifies a path to a library that will be used when loading any FIDO authenticator-hosted keys, overriding the default of using the built-in USB HID support.

FILES

~/.ssh/id_dsa

~/.ssh/id_ecdsa

~/.ssh/id_ecdsa_sk

~/.ssh/id_ed25519

~/.ssh/id_ed25519_sk

~/.ssh/id_rsa

Contains the DSA, ECDSA, authenticator-hosted ECDSA, Ed25519, authenticator-hosted Ed25519 or RSA authentication identity of the user.

Identity files should not be readable by anyone but the user. Note that **ssh-add** ignores identity files if they are accessible by others.

EXIT STATUS

Exit status is 0 on success, 1 if the specified command fails, and 2 if **ssh-add** is unable to contact the authentication agent.

SEE ALSO

ssh(1), ssh-agent(1), ssh-askpass(1), ssh-keygen(1), sshd(8)

AUTHORS

OpenSSH is a derivative of the original and free ssh 1.2.12 release by Tatu Ylonen. Aaron Campbell,

Bob Beck, Markus Friedl, Niels Provos, Theo de Raadt and Dug Song removed many bugs, re-added newer features and created OpenSSH. Markus Friedl contributed the support for SSH protocol versions 1.5 and 2.0.