

# Informazioni su Secure Shell Packet Exchange

## Sommario

---

[Introduzione](#)

[Prerequisiti](#)

[Requisiti](#)

[Componenti usati](#)

[Protocollo SSH](#)

[SSH Exchange](#)

[Informazioni correlate](#)

---

## Introduzione

In questo documento viene descritto lo scambio a livello di pacchetto durante la negoziazione SSH (Secure Shell).

## Prerequisiti

### Requisiti

Cisco raccomanda la conoscenza dei concetti base relativi alla sicurezza:

- Autenticazione
- Riservatezza
- Integrità
- Metodi di scambio chiave

### Componenti usati

Il documento può essere consultato per tutte le versioni hardware.

Le informazioni discusse in questo documento fanno riferimento a dispositivi usati in uno specifico ambiente di emulazione. Su tutti i dispositivi menzionati nel documento la configurazione è stata ripristinata ai valori predefiniti.

## Protocollo SSH

Il protocollo SSH è un metodo per effettuare in modo sicuro il login remoto da un computer all'altro. Le applicazioni SSH si basano su un'architettura client-server e connettono un'istanza del client SSH a un server SSH.

# SSH Exchange

1. La prima fase del SSH è denominata Identification String Exchange.

a. Il client crea un pacchetto e lo invia al server contenente:

- Versione protocollo SSH
- Versione del software

```
323 5.946818 10.65.54.8 10.106.51.72 SSHv2 82 Client: Protocol (SSH-2.0-PuTTY_Release_0.76)
> Frame 323: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface 0
> Ethernet II, Src: Cisco_3c:7a:00 (00:05:9a:3c:7a:00), Dst: Cimsys_33:44:55 (00:11:22:33:44:55)
> Internet Protocol Version 4, Src: 10.65.54.8, Dst: 10.106.51.72
> Transmission Control Protocol, Src Port: 56127, Dst Port: 22, Seq: 1, Ack: 1, Len: 28
v SSH Protocol
  Protocol: SSH-2.0-PuTTY_Release_0.76
```

La versione del protocollo client è SSH2.0, la versione del software è Putty\_0.76.

b. Il server risponde con la propria stringa di identificazione Exchange, incluse la versione del protocollo SSH e la versione del software.

```
326 6.016955 10.106.51.72 10.65.54.8 SSHv2 73 Server: Protocol (SSH-2.0-Cisco-1.25)
> Frame 326: 73 bytes on wire (584 bits), 73 bytes captured (584 bits) on interface 0
> Ethernet II, Src: Cimsys_33:44:55 (00:11:22:33:44:55), Dst: Cisco_3c:7a:00 (00:05:9a:3c:7a:00)
> Internet Protocol Version 4, Src: 10.106.51.72, Dst: 10.65.54.8
> Transmission Control Protocol, Src Port: 22, Dst Port: 56127, Seq: 1, Ack: 29, Len: 19
v SSH Protocol
  Protocol: SSH-2.0-Cisco-1.25
```

La versione del protocollo del server è SSH2.0, la versione del software è Cisco 1.25

2. Il passo successivo è **Algorithm Negotiation**. In questo passo, sia il client che il server negoziano i seguenti algoritmi:

- Scambio chiave
- Crittografia
- HMAC (codice di autenticazione del messaggio basato su hash)
- Compressione

1. Il client invia un messaggio di inizializzazione scambio chiave al server, specificando gli algoritmi supportati. Gli algoritmi sono elencati in ordine di preferenza.

```
329 6.021990 10.65.54.8 10.106.51.72 SSHv2 238 Client: Key Exchange Init
> Frame 329: 238 bytes on wire (1904 bits), 238 bytes captured (1904 bits) on interface 0
> Ethernet II, Src: Cisco_3c:7a:00 (00:05:9a:3c:7a:00), Dst: Cimsys_33:44:55 (00:11:22:33:44:55)
> Internet Protocol Version 4, Src: 10.65.54.8, Dst: 10.106.51.72
> Transmission Control Protocol, Src Port: 56127, Dst Port: 22, Seq: 1101, Ack: 20, Len: 184
> [3 Reassembled TCP Segments (1256 bytes): #327(536), #328(536), #329(184)]
v SSH Protocol
  SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
    Packet Length: 1252
    Padding Length: 11
  Key Exchange
    Message Code: Key Exchange Init (20)
  Algorithms
```

Inizializzazione scambio chiave

```

Algorithms
Cookie: 47a96215afc92003180b60342970a105
kex_algorithms length: 315
kex_algorithms string [truncated]: curve448-sha512,curve25519-sha256,curve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,dif
server_host_key_algorithms length: 123
server_host_key_algorithms string: rsa-sha2-512,rsa-sha2-256,ssh-rsa,ssh-ed448,ssh-ed25519,ecdsa-sha2-nistp256,ecdsa-sha2-nistp384,ecdsa-sha2-nistp521,ssh-dss
encryption_algorithms_client_to_server length: 189
encryption_algorithms_client_to_server string: aes256-ctr,aes256-cbc,rijndael-cbc@lysator.liu.se,aes192-ctr,aes192-cbc,aes128-ctr,aes128-cbc,chacha20-poly1305
encryption_algorithms_server_to_client length: 189
encryption_algorithms_server_to_client string: aes256-ctr,aes256-cbc,rijndael-cbc@lysator.liu.se,aes192-ctr,aes192-cbc,aes128-ctr,aes128-cbc,chacha20-poly1305
mac_algorithms_client_to_server length: 155
mac_algorithms_client_to_server string: hmac-sha2-256,hmac-sha1,hmac-sha1-96,hmac-md5,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha1-96-etm
mac_algorithms_server_to_client length: 155
mac_algorithms_server_to_client string: hmac-sha2-256,hmac-sha1,hmac-sha1-96,hmac-md5,hmac-sha2-256-etm@openssh.com,hmac-sha1-etm@openssh.com,hmac-sha1-96-etm
compression_algorithms_client_to_server length: 26
compression_algorithms_client_to_server string: none,zlib,zlib@openssh.com
compression_algorithms_server_to_client length: 26
compression_algorithms_server_to_client string: none,zlib,zlib@openssh.com

```

Algoritmi supportati dal client

b. Il server risponde con il proprio messaggio di inizializzazione scambio chiave, elencando gli algoritmi supportati.

c. Poiché questi messaggi vengono scambiati contemporaneamente, entrambe le parti confrontano i propri elenchi di algoritmi. Se esiste una corrispondenza negli algoritmi supportati da entrambi i lati, questi procedono al passaggio successivo. Se non esiste una corrispondenza esatta, il server seleziona il primo algoritmo dall'elenco del client supportato.

d. Se il client e il server non concordano su un algoritmo comune, lo scambio di chiave non riesce.

```

334 6.093250 10.106.51.72 10.65.54.8 SSHv2 366 Server: Key Exchange Init
> Frame 334: 366 bytes on wire (2928 bits), 366 bytes captured (2928 bits) on interface 0
> Ethernet II, Src: Cimsys_33:44:55 (00:11:22:33:44:55), Dst: Cisco_3c:7a:00 (00:05:9a:3c:7a:00)
> Internet Protocol Version 4, Src: 10.106.51.72, Dst: 10.65.54.8
> Transmission Control Protocol, Src Port: 22, Dst Port: 56127, Seq: 20, Ack: 1285, Len: 312
SSH Protocol
  SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
    Packet Length: 308
    Padding Length: 4
    Key Exchange
      Message Code: Key Exchange Init (20)
      Algorithms

```

Inizializzazione scambio chiave server

3. Dopo questa operazione, entrambi i dispositivi entrano nella Key Exchange fase di generazione del segreto condiviso utilizzando lo scambio di chiave DH e autenticano il server:

a. Il client genera una coppia di chiavi Public and Private e invia la chiave pubblica DH nel pacchetto Init di Exchange del gruppo DH. Questa coppia di chiavi viene utilizzata per il calcolo della chiave segreta.

```

337 6.201114 10.65.54.8 10.106.51.72 SSHv2 326 Client: Diffie-Hellman Group Exchange Init
> Frame 337: 326 bytes on wire (2608 bits), 326 bytes captured (2608 bits) on interface 0
> Ethernet II, Src: Cisco_3c:7a:00 (00:05:9a:3c:7a:00), Dst: Cimsys_33:44:55 (00:11:22:33:44:55)
> Internet Protocol Version 4, Src: 10.65.54.8, Dst: 10.106.51.72
> Transmission Control Protocol, Src Port: 56127, Dst Port: 22, Seq: 1309, Ack: 612, Len: 272
SSH Protocol
  SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
    Packet Length: 268
    Padding Length: 6
    Key Exchange
      Message Code: Diffie-Hellman Group Exchange Init (32)
      Multi Precision Integer Length: 256
      DH client e: 1405ab00ff368031363467ad6653967d5a64eac4734e5dc6
      Padding String: 5c81f2cffc95

```

Inizializzazione scambio chiave pubblica DH client e gruppo Diffie-Hellman

b. Il server genera una propria coppia `Public and Private` di chiavi. Utilizza la chiave pubblica del client e la sua coppia di chiavi per calcolare il segreto condiviso.

c. Il server calcola anche un hash di Exchange con questi input:

- Stringa di identificazione client
- Stringa di identificazione server
- Payload di KEXINIT client
- Payload di Server KEXINIT
- Chiave pubblica server da chiavi host ( coppia di chiavi RSA )
- Chiave pubblica DH client
- Chiave pubblica DH del server
- Chiave privata condivisa

d. Dopo aver calcolato l'hash, il server lo firma con la chiave privata RSA.

e. Il server crea un messaggio `DH_Exchange_Reply` che include:

- RSA-Chiave pubblica del server (per consentire al client di autenticare il server)
- DH-Chiave pubblica del server (per il calcolo del segreto condiviso)
- HASH (per autenticare il server e dimostrare che il server ha generato il segreto condiviso, in quanto la chiave privata fa parte del calcolo dell'hash)

```
343 6.330017 10.106.51.72 10.65.54.8 SSHv2 350 Server: Diffie-Hellman Group Exchange Reply
Internet Protocol Version 4, Src: 10.106.51.72, Dst: 10.65.54.8
Transmission Control Protocol, Src Port: 22, Dst Port: 56127, Seq: 1148, Ack: 1581, Len: 296
[2 Reassembled TCP Segments (832 bytes): #342(536), #343(296)]
SSH Protocol
  SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
    Packet Length: 828
    Padding Length: 8
    Key Exchange
      Message Code: Diffie-Hellman Group Exchange Reply (33)
      KEX host key (type: ssh-rsa)
        Host key length: 279
        Host key type length: 7
        Host key type: ssh-rsa
        Multi Precision Integer Length: 3
        RSA public exponent (e): 010001
        Multi Precision Integer Length: 257
        RSA modulus (N): 0098c7d23c9ababd730f07b5c2aee1e4e51bac67970aa5af...
        Multi Precision Integer Length: 256
        DH server f: 3a17a0995531f12d629a48ab6f25715bc181ea3deb6c6793...
        KEX H signature length: 271
        KEX H signature: 000000077373682d72736100000100691d2c896761bc7481...
        Padding String: 0000000000000000
```

Risposta di scambio chiave pubblica DH del server e gruppo Diffie-Hellman

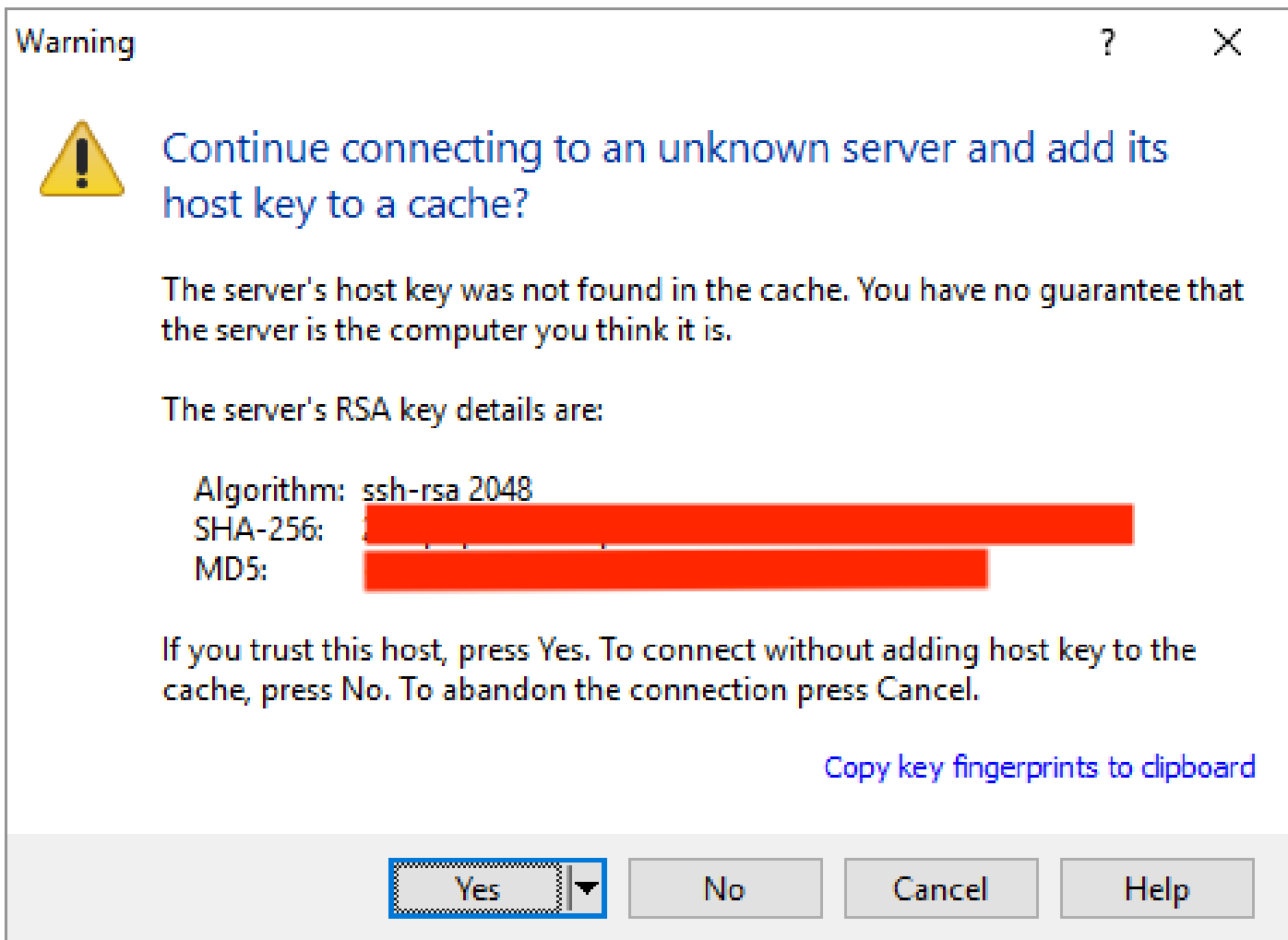
f. Dopo aver ricevuto `DH_Exchange_Reply`, il client calcola l'hash nello stesso modo e lo confronta con l'hash ricevuto, decrittografandolo con la chiave pubblica RSA del server.

g. Prima di decrittografare l'HASH ricevuto, il client deve verificare la chiave pubblica del server. Questa verifica viene eseguita tramite un certificato digitale firmato da un'Autorità di certificazione (CA). Se il certificato non esiste, spetta al client decidere se accettare la chiave pubblica del server.



Nota: quando si accede per la prima volta al protocollo SSH su un dispositivo che non usa un certificato digitale, è possibile che venga visualizzata una schermata di popup in cui si chiede di accettare manualmente la chiave pubblica del server. Per evitare di visualizzare questo popup ogni volta che ci si connette, è possibile scegliere di aggiungere la chiave host del server alla cache.

---



Chiave RSA del server

4. Poiché il segreto condiviso è ora generato, entrambi gli endpoint lo utilizzano per derivare queste chiavi:

- Chiavi di crittografia
- Chiavi IV: numeri casuali utilizzati come input per algoritmi simmetrici al fine di migliorare la sicurezza
- Chiavi di integrità

La fine dello scambio di chiavi è segnalata dallo scambio del `NEW KEYS'` messaggio, che informa ciascuna parte che tutti i messaggi futuri saranno crittografati e protetti utilizzando queste nuove chiavi.

Seq	Len	Src	Dst	Protocol	Info
346	6.330368	10.106.51.72	10.65.54.8	SSHv2	70 Server: New Keys
347	6.365552	10.65.54.8	10.106.51.72	SSHv2	70 Client: New Keys

```
> Frame 346: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0
> Ethernet II, Src: Cimsys_33:44:55 (00:11:22:33:44:55), Dst: Cisco_3c:7a:00 (00:05:9a:3c:7a:00)
> Internet Protocol Version 4, Src: 10.106.51.72, Dst: 10.65.54.8
> Transmission Control Protocol, Src Port: 22, Dst Port: 56127, Seq: 1444, Ack: 1581, Len: 16
✓ SSH Protocol
  ✓ SSH Version 2 (encryption:aes256-ctr mac:hmac-sha2-256 compression:none)
    Packet Length: 12
    Padding Length: 10
    ✓ Key Exchange
      Message Code: New Keys (21)
      Padding String: 000000000000000000000000
```

Nuove chiavi client e server

5. La fase finale è la richiesta di assistenza. Il client invia un pacchetto di richiesta del servizio SSH al server per avviare l'autenticazione dell'utente. Il server risponde con un messaggio di accettazione del servizio SSH, in cui viene richiesto al client di eseguire l'accesso. Questo scambio avviene tramite il canale sicuro stabilito.

## Informazioni correlate

- <https://www.cisco.com/c/en/us/support/docs/security-vpn/secure-shell-ssh/4145-ssh.html>
- <https://datatracker.ietf.org/doc/html/rfc4253>
- [Supporto tecnico Cisco e download](#)

## Informazioni su questa traduzione

Cisco ha tradotto questo documento utilizzando una combinazione di tecnologie automatiche e umane per offrire ai nostri utenti in tutto il mondo contenuti di supporto nella propria lingua. Si noti che anche la migliore traduzione automatica non sarà mai accurata come quella fornita da un traduttore professionista. Cisco Systems, Inc. non si assume alcuna responsabilità per l'accuratezza di queste traduzioni e consiglia di consultare sempre il documento originale in inglese (disponibile al link fornito).