SSH.COM

Tectia® Client/Server 6.6

Product Description

20 December 2023

Tectia® Client/Server 6.6: Product Description

20 December 2023

THERE IS NO WARRANTY OF ANY KIND FOR THE ACCURACY, RELIABILITY OR USEFULNESS OF THIS INFORMATION EXCEPT AS REQUIRED BY APPLICABLE LAW OR EXPRESSLY AGREED IN WRITING.

For Open Source Software acknowledgements, see appendix *Open Source Software License Acknowledgements* in the *User Manual*.

SSH Communications Security Corporation
Karvaamokuja 2b, Suite 600, FI-00380 Helsinki, Finland

Table of Contents

Chapter 1 Introduction

Tectia offers software tools to secure end-to-end communications within corporate networks. Tectia client/server solution allows secure network services over an unsecured network, such as the Internet.

Tectia products can be deployed cost-effectively to large corporate networks, because their installation and maintenance can be managed centrally.

The award-winning Secure Shell technology provides secure encrypted and authenticated communications between two non-trusted hosts. Users can establish secure connections to remote hosts, execute commands on the remote hosts securely, copy remote files securely, and forward X11 sessions (on Unix). Arbitrary TCP/IP ports can also be forwarded (tunneled) over a secure channel, enabling secure application connections, for example, to a database server.

Tectia products are based on Secure Shell technology originally developed by the founders of SSH Communications Security. The Internet Engineering Task Force (IETF) has standardized the Secure Shell (SSH) protocol, see the original RFC 4251 at https://www.ietf.org/rfc/rfc4251.txt.

The future-proof design of the SSH protocol allows extensions, for example, the SHA-2 @ssh.com algorithms introduced in 2011 in Tectia Client/Server 6.2 version. Tectia Quantum Safe Edition introduced multiple Post Quantum Cryptography (PQC) Hybrid Key Exchange algorithms in 2022 in Tectia Client/Server 6.6 version to address the quantum threath.

1.1 Tectia Solution Components

The Tectia client/server solution utilizes client-server architecture. By default, the server listens to TCP port 22, which has been officially assigned for Secure Shell, and clients initiate connections to this port. File transfer and application users typically connect via load balancer to the server cluster and system administrators to the cluster node.

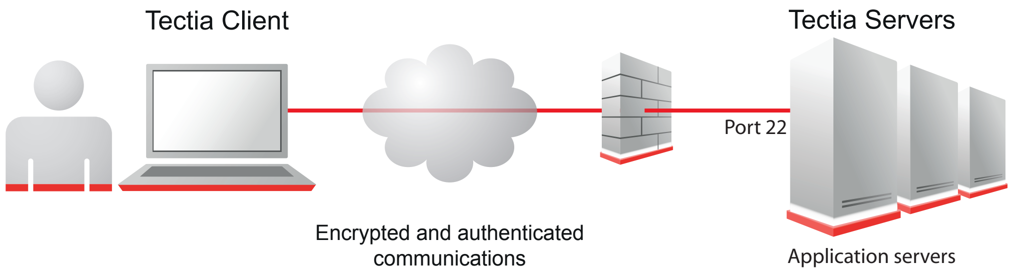


Figure 1.1. The basic idea of Tectia Client and Server

The Tectia products work ideally together, but they can also be used with other Secure Shell-based clients or servers.

1.1.1 Tectia Client

Tectia Client is a workstation product, available in English, and with additional license in Japanese, providing the Secure Shell client features and tools. Tectia Client takes care of securing remote connections and transfer of files. Users and system administrators need Tectia Client in order to access remote hosts running Tectia Server or another standard Secure Shell server. Tectia Client provides interactive file transfer and terminal client functionalities.

Tectia Client also includes advanced command-line tools for system administrators to set up secure automated file transfers, and tools for outgoing and incoming application tunneling.

Connection Broker

The Connection Broker is an integrated component of Tectia Client and Tectia ConnectSecure. The Connection Broker handles all cryptographic operations and authentication-related tasks on the client side. Connection Broker, by default waits for a few seconds after the last client quits before disconnecting from the server. If a client, like scpg3, reconnects, a new channel is opened inside already authenticated connection.

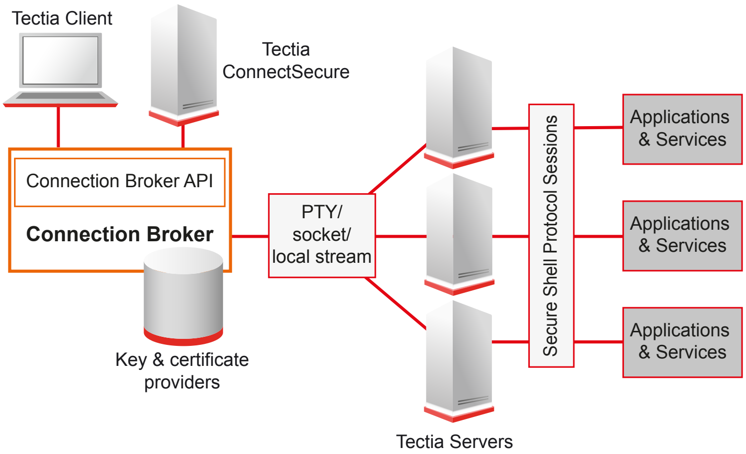


Figure 1.2. Connection Broker architecture

The Connection Broker as an authentication agent shown in Figure 1.3.

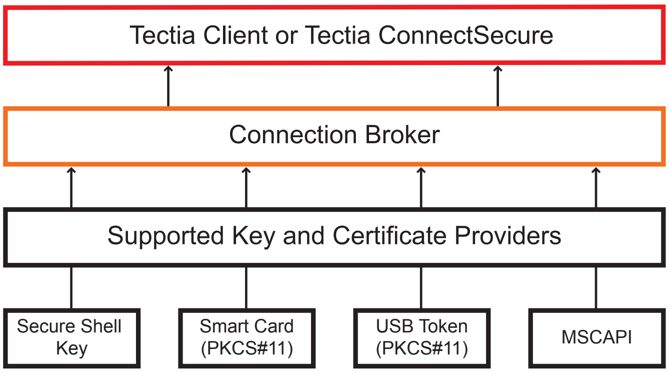


Figure 1.3. Connection Broker functions as agent

1.1.2 Tectia Server

Tectia Server provides the Secure Shell server features and tools. It enables secure file transfers, secure application connectivity, and secure remote administration services over unsecured networks.

Tectia Server is a robust, flexible, and field-tested server implementation of the Secure Shell protocol with easy-to-use graphical user interface for configuring Tectia Server Its technology has been the choice of numerous large corporations, banks, financial organizations, and governments around the world.

Tectia Server provides strong user authentication, traffic encryption/decryption, and both traffic and file integrity checking. It also provides out-of-the-box interfaces to integrate to existing third-party authentication or authorization systems (such as Pluggable Authentication Modules, RSA SecurID, GSSAPI, PKI with CAC and PIV cards) or seamlessly to Tectia Zero Trust Module with session recording and role-based access with just-in-time X.509 and OpenSSH certificates.

Tectia Server

Tectia Server is available for Unix and Windows platforms, such as Oracle Solaris, IBM AIX, HP-UX, SUSE and Red Hat Linux, and Microsoft Windows. The server license subscription includes also Tectia Client that can be installed on the same host.

Tectia Server offers all Secure Shell server functionality, including secure terminal, SFTP, and tunneling. When Tectia Client or Tectia ConnectSecure is acting as the counterpart, Tectia Server supports also the enhanced file transfer (EFT) features, such as streaming for fast transfer of large files.

Tectia Server for IBM z/OS

Tectia Server for IBM z/OS has been designed for z/OS platforms running on IBM mainframes. It provides the same services as Tectia Server on Unix and out-of-the-box support for direct MVS data set access and interactive data set listing, interfacing with JES, I/O streaming, configurable ASCII/

EBCDIC code set conversions, ISPF application, and FTP compatibility commands, such as the SITE command. z/OS client side tools support FTP-SFTP conversion, transparent FTP tunneling, and enhanced file transfer (EFT) features.

1.1.3 Tectia Quantum Safe Edition

Tectia Quantum Safe Edition makes Tectia quantum-safe for the future. It is a separate product enhancementad for Tectia Client/Server that can be enabled with an additional license without the need to reinstall Tectia products.

Tectia Quantum Safe supports multiple Post Quantum Cryptography (PQC) algorithms, including CRYSTALS-Kyber, FrodoKEM, SABER and Streamlined NTRU Prime that are used in a Hybrid Key Exchange in SSH together with a classical ECDH algorithm. Both the PQC and ECDH algorithm contribute to the key material resulting in a session key that is at least as hard to break as the strongest composite. The hybrid approach mitigates the risk of future attacks on recorded secure shell sessions if weaknesses are discovered in either algorithm.

Tectia Quantum Safe and FIPS

The FIPS 140-2 validation, and FIPS 140-3 validation does not cover PQC algorithms themselves. However, the FIPS 140 series cryptographic of validation allows additional inputs for the validated key derivation functions. The PQC algorithms are used for generating such inputs, and therefore the use of PQC is allowed on FIPS-140 validated cryptosystems. For example, the CRYSTALS-Kyber PQC algorithm selected by NIST and used with ECDH NIST curve P521 in Hybrid Key Exchange in Tectia, can be used in FIPS mode.

1.1.4 Tectia Zero Trust Edition

Tectia Zero Trust Edition allows an efficient role-based access control (RBAC) to bring scalability to managing access to large server estates. By operating without permanent credentials like SSH keys or passwords, Tectia Zero Trust eliminates the costly process of managing or rotating credentials while also greatly enhancing system security by removing a significant potential attack vector. Additionally, it increases transparency by centralizing system access audit logs and optionally provides advanced auditing including session recording. It is a separate product enhancementad module for Tectia Client/Server that interoperates also with recent 3rd party Secure Shell implementations. Tectia version 6.5 or above is officially supported, however any Tectia version that supports SHA-2 signature algorithms can be configured to support Tectia Zero Trust if needed.

1.2 Multi-Platform Support

Tectia offers extensive platform support for its products. Tectia client/server solution can be deployed into a heterogeneous environment where most commonly Microsoft Windows and different flavors of Unix and Linux operating systems are used. Tectia client/server solution can also connect to Tectia Server for IBM z/OS for transferring data to and from mainframe computers.

1.3 Customer Support Services

SSH Communications Security offers Professional Services, the most important of which are consulting, product knowledge and training, and software upgrades. The support engineers at our global support centers are backed by the developers of the SSH information security solutions. Read more about our Professional Services at https://www.ssh.com/products/services/.

Technical Support is available according to your selected support plan - whether your business critical systems require support for 24 hours a day, 7 days a week, or less urgent response times. All offered support levels have defined services level agreements (SLAs) and include software maintenance. Read more about SSH Support Services at https://www.ssh.com/products/support/.

Chapter 2 Key Applications

The key applications of Tectia client/server solution are secure file transfer, secure system administration, and secure application connectivity.

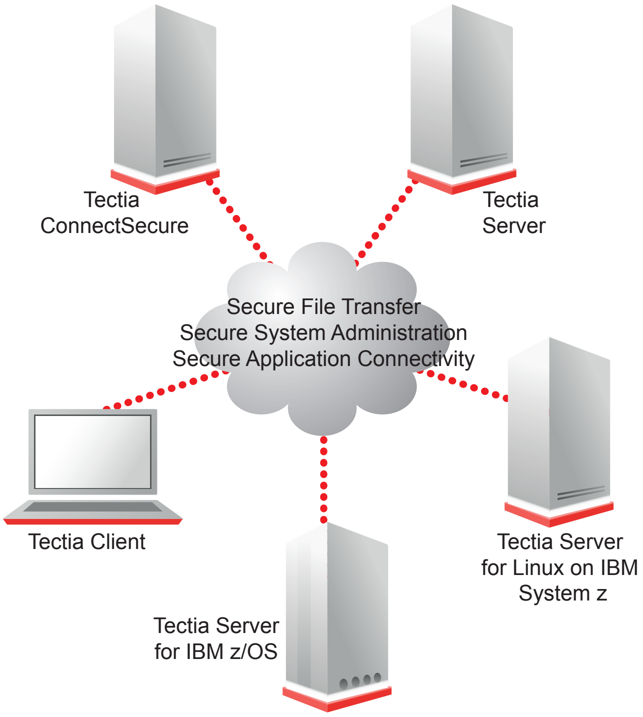


Figure 2.1. The key applications of Tectia products

2.1 Secure File Transfer - FTP Replacement

The Tectia client/server solution allows organizations to replace plaintext file transfer protocol (FTP) connections with secure file transfers in cross-platform environments. File transfers can be secured by applying the Secure File Transfer Protocol (SFTP) instead of FTP, or by using tunnels that encrypt the connection from the FTP client to the FTP server.

Tectia ConnectSecure and Tectia Server for IBM z/OS offer also transparent FTP tunneling to encrypt the FTP connection, and FTP-SFTP conversion to convert unsecured FTP traffic to use the secure SFTP protocol, instead.

2.1.1 Secure File Transfer Protocol (SFTP)

The Secure File Transfer Protocol (SFTP) is a de facto industry standard for secure file transfers, and it is natively supported by the Tectia client/server solution.

SFTP allows secure copying, moving, editing, and removing of files over TCP/IP networks. Scripted file transfers between enterprise servers can be secured by using the Tectia command-line SFTP and SCP (Secure Copy) tools with automated and ad hoc file transfers. For secure interactive file transfers, Windows users have the Tectia Secure File Transfer GUI.

Tectia supports the legacy OpenSSH SCP implementation used by default in OpenSSH version 8 and below for easy migration of OpenSSH environments to Tectia, creating a smoother transition to ensure seamless connectivity during the migration period.

SFTP features include:

File transfer resume

The file transfer resume feature allows resuming interrupted file transfers instead of restarting the whole operation. The file transfer resume uses file hashing to determine the point of resume. For increased performance, you can apply the checkpoint/restart mechanism for optimum handling of interruptions in large file transfers.

File integrity check

Tectia supports in SFTP digest command that can be used to check the hash of the remote or local file on Unix and Windows. This is particularly useful if users are restricted to SFTP services only in Tectia Server configuration.

Easy SFTP restrictions

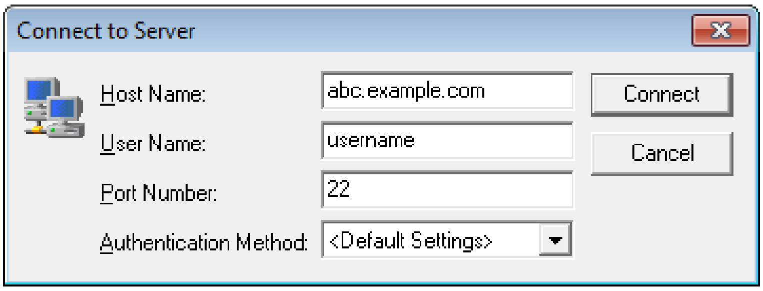
Tectia Server can be easily configured with subsystem chrooting on Unix platforms to confine users to a specific directory tree (e.g., home directory or user-specific directly on a network share) for added security and ease of use. Tectia supports versatile file system permissions so that it is possible to chroot for example read-only download users to a user-specific directory and upload user(s) to the parent directory so that root-privileges are not required for maintaining downloadable files.

On Windows Server, SFTP access to the file system is through virtual folders that provide additional layer to limit user access compared to what the operating system itself provides. Each SFTP user can be limited to multiple local folders or folders on network shares that can be named descriptively independent from the file system folder name.

2.2 Secure System Administration

The Tectia client/server solution is used by system administrators as a replacement for unsecured login protocols, such as Rlogin, Telnet, and FTP. The Tectia Client software is installed in the system administrator's workstation and the Tectia Server software in the managed server. Typically, the number of servers is much higher than the number of client installations. In numerous active Tectia implementations there are thousands of Tectia Server instances installed within a corporate network.

With the Tectia client/server solution, administrators can log in to remote hosts securely, as their user ID and authentication information are transmitted over the network in encrypted format.



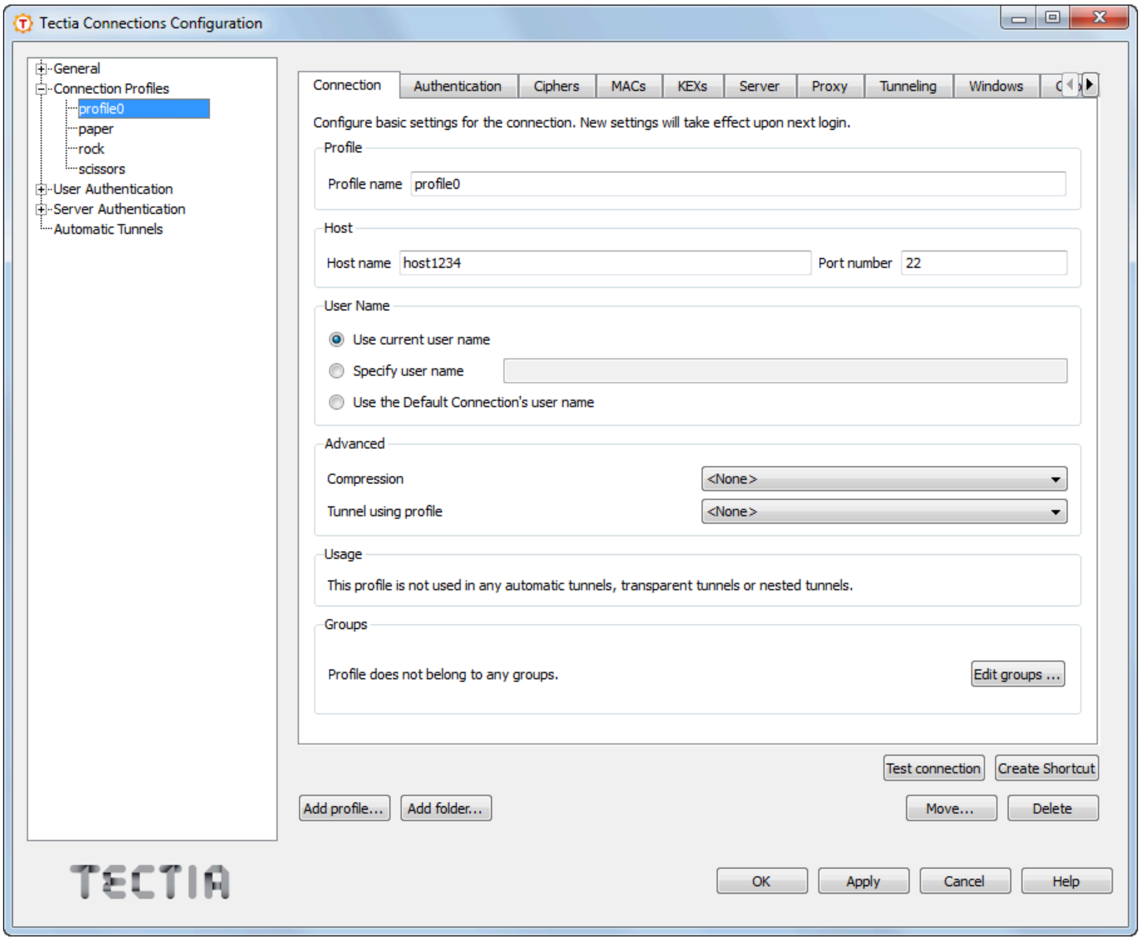
Host Name: abc.example.com
User Name: username
Port Number: 22
Authentication Method: <Default Settings>

Figure 2.2. Connecting to a remote server with Tectia SSH Terminal

Users can connect to remote servers also with a command-line tool. In this example, a user named Susan connects to a server host for the first time, and receives the host key for validation:

```
$ sshg3 susanstrict@examplehost
Host key not found from database.
Key fingerprint:
xecic-fifub-kivyh-kohag-zedyn-logum-pycuz-besug-galoh-gupah-xaxby
You can get a public key's fingerprint by running
% ssh-keygen-g3 -F publickey.pub
on the keyfile.
Are you sure you want to continue connecting (yes/no)?
```

Since users typically connect repeatedly to the same servers, they can create their own set of Connection Profiles which include all information for establishing the connection.



Configure basic settings for the connection. New settings will take effect upon next login.

Profile

Profile name: profile0

Host

Host name: host1234 Port number: 22

User Name

● Use current user name

○ Specify user name

○ Use the Default Connection's user name

Advanced

Compression: <None>

Tunnel using profile: <None>

Usage

This profile is not used in any automatic tunnels, transparent tunnels or nested tunnels.

Groups

Profile does not belong to any groups.

Figure 2.3. Configuring connection profiles

With the Tectia client/server solution, login can be easily done also in heterogeneous network environments including Windows, Unix, Linux, and IBM mainframe systems. This eliminates the need to deploy and maintain Secure Shell implementations from multiple vendors.

2.3 Secure Application Connectivity

The Tectia client/server solution can be used to replace unsecured TCP-based terminal connections (for example, Telnet) to business-critical enterprise applications. Through strong encryption and data integrity, the Tectia client/server solution protects sensitive data and passwords against unauthorized access, facilitating compliance with regulations and best practices. Strong authentication of users is supported through broad integration with third-party authentication systems or Tectia Zero Trust Module.

The Tectia client/server solution offers the following ways of securing data communications between standard TCP-based applications:

- **Tunneling**, or port forwarding, is a way to forward otherwise insecure application traffic through Secure Shell. Tunneling can provide secure application connectivity for weakly encrypted TCP connections or unencrypted connections, for example HTTP-based applications.

Tunneling provides encryption and strong two-factor authentication to third-party network client applications. Tectia allows different forms of tunnels depending on the environment and type of usage of the workstations or user terminals.

The Secure Shell v2 connection protocol provides channels that can be used for a wide range of purposes. All of these channels are multiplexed into a single encrypted tunnel and can be used for tunneling (forwarding) arbitrary TCP/IP ports and X11 connections.

- The `sshg3` command-line tool can be used interactively or in scripts. Alternatively, Tectia Client can be easily configured to open a secure tunnel when the tunneled application connects to a local listener or local SOCKS proxy service of Connection Broker.

Non-transparent tunnels

Tectia Client supports non-transparent application tunneling, which means that the tunneled applications need to be defined on the basis of the TCP ports they use. Applications with dynamic ports are not supported.

Transparent tunnels

With the transparent TCP tunneling feature activated on the client-side (on Tectia ConnectSecure), TCP-based applications can be tunneled transparently without changing the end-user experience or requiring any modifications on the applications, thus reducing the total cost of ownership.

Static tunnels

Tectia Client can also be used for application protection using the static tunneling feature. As opposed to transparent TCP tunneling, static tunnels are configured so that an application connects to a local port running Tectia Client, and the Client tunnels the application to a specified remote host.

Chapter 3 Features and Benefits

The Tectia client/server solution is an ideal lightweight solution for secure file transfer, secure remote logins for system administration purposes, and secure use of weakly encrypted business applications over a network. The Tectia client/server solution is available for most Unix and Linux platforms, for Microsoft Windows, and for IBM mainframes.

The Tectia client/server solution works with any type of Internet (TCP/IP) connection - ADSL, ISDN, modem, Ethernet, WLAN, PPPoE - thus making it widely applicable, totally independent of the network topology, and independent of network address translations.

There are some differences in the availability of features between versions of the Tectia products. For information on the features supported on each product version, see Table 6.3.

3.1 Tectia Client/Server Solution Features

The following general features are available with all products of the Tectia client/server solution.

Compliance with the IETF Secure Shell standards

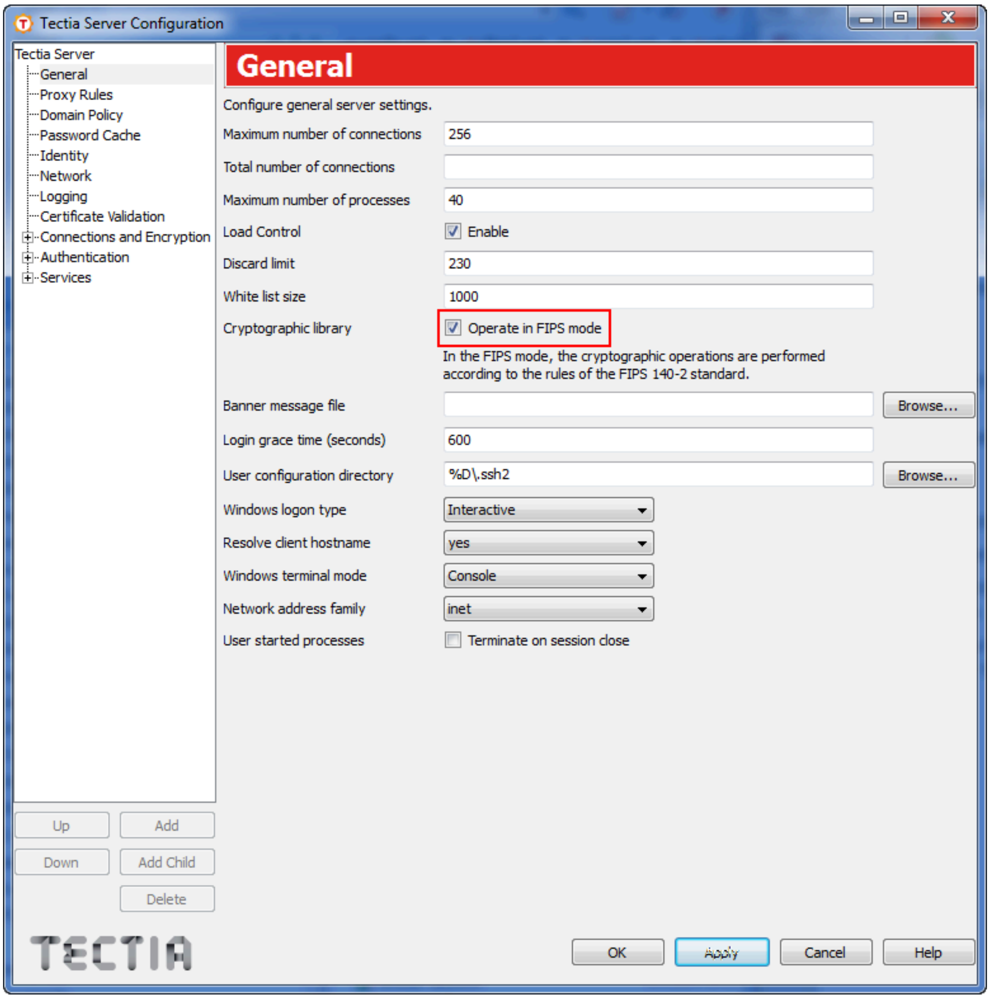
The Tectia client/server solution implements the Secure Shell (version 2) protocol as defined by the IETF Proposed Standard RFC specifications. SSH Communications Security is the original developer of Secure Shell and has been an active driver of the Secure Shell standardization in the IETF.

Comprehensive cryptographic support

The Tectia client/server solution offers state-of-the-art encryption with broad support for multiple Post Quantum Cryptography (PQC) algorithms to choose from in Tectia Quantum Safe Edition including CRYSTALS-Kyber, FrodoKEM, SABER and Streamlined NTRU Prime that are used in a Hybrid Key Exchange together with a classical ECDH algorithm. Supported symmetric ciphers in Tectia Client and Server include 3DES, AES (CTR, CBC and GCM), Arcfour, Blowfish, SEED, and Twofish. Supported message authentication and public-key algorithms include MD5, SHA-1, SHA-2, Diffie-Hellman, DSA, RSA, ECDSA and Ed25519.

FIPS-certified cryptographic library

The Tectia client/server solution incorporates a FIPS 140-2 certified cryptographic module to help ensure acceptance in government audits. The FIPS 140-2 Cryptographic Library has been validated for Windows, Solaris, and major Unix platforms. The mode of the cryptographic library can be changed easily in the Tectia Connections Configuration GUI or by editing the configuration file.



Tectia Server Configuration

Tectia Server
- General
- Proxy Rules
- Domain Policy
- Password Cache
- Identity
- Network
- Logging
- Certificate Validation
- Connections and Encryption
- Authentication
- Services

General

Configure general server settings.

Maximum number of connections: 256

Total number of connections:

Maximum number of processes: 40

Load Control: ☑ Enable

Discard limit: 230

White list size: 1000

Cryptographic library: ☑ Operate in FIPS mode

In the FIPS mode, the cryptographic operations are performed according to the rules of the FIPS 140-2 standard.

Banner message file:

Login grace time (seconds): 600

User configuration directory: %D\.ssh2

Windows logon type: Interactive

Resolve client hostname: yes

Windows terminal mode: Console

Network address family: inet

User started processes: □ Terminate on session close

Figure 3.1. Activating FIPS mode on Tectia Server

For a list of platforms on which the FIPS library has been validated or tested, see Section 6.5.

Versatile command-line tools

The Tectia products include versatile command-line tools that can be used for remote login, remote command execution, and file transfer operations. These tools allow easy scripting of automated jobs such as secure file transfers or starting and stopping of services in remote locations.

Tunneling (port forwarding)

One of the key features of Secure Shell, in addition to secure terminal access and secure file transfers, is its ability to tunnel TCP-based application connections. The Tectia products support static application tunneling where application client connections are routed through the local TCP port, and then securely tunneled to a remote Secure Shell server. If the tunneled application supports SOCKS (4 or 5), Connection Broker can be configured to act as a SOCKS server for the tunneled applications, creating forwards as requested by the SOCKS transaction.

Automatic tunneling

Before an application can be tunneled, a Secure Shell connection needs to be established. When using the automatic tunneling feature, the Tectia client-side component listens to a specific port and establishes the encrypted connection automatically when the specific application is connecting to the local host port.

Firewall traversal

The Tectia products themselves support SOCKS (4 and 5) and HTTP proxy for accessing Secure Shell servers located behind firewalls.

Multi-channel support

Multi-channel support allows users to have multiple terminal sessions, file transfers, and application tunnels that are multiplexed to a single Secure Shell connection without the need to authenticate every session separately.

Configurable re-keying policies

Administrators can configure the renewal period for session encryption keys according to the security requirements.

3.2 High Performance

Tectia applies third-generation Secure Shell protocol implementation, SSH G3, which has been optimized for higher performance in demanding file transfer and application tunneling environments. The SSH G3 architecture provides unparalleled Secure Shell encryption throughput and scalability for large organizations.

Connection scalability

SSH G3 implements an `n x m` server process architecture for optimized server-side memory consumption and performance. While each server process (total amount n) can handle multiple (m) connections, the memory consumption per connection is considerably lower than in the second-generation Secure Shell implementations, making Tectia an ideal solution especially for large-scale application tunneling.

Client-side Connection Broker

The Connection Broker is a key component in the SSH G3 architecture, handling all protocol and cryptographic operations. Client-side memory consumption is reduced since there needs to be only a single Connection Broker instance running per user. Security is also further improved by isolating all security-critical operations including authentication data handling in a single component.

Higher throughput

The SSH G3 architecture has been designed to minimize internal data handling such as data copy operations to minimize the throughput time in large file transfers.

Multi-threading

SSH G3 utilizes multi-threaded programming to fully leverage multi-processor servers for improved performance.

3.3 Ease of Use

Tectia client/server solution is easy to install, and it can be configured using a GUI or by editing an XML configuration file that can be standardized across the organization.

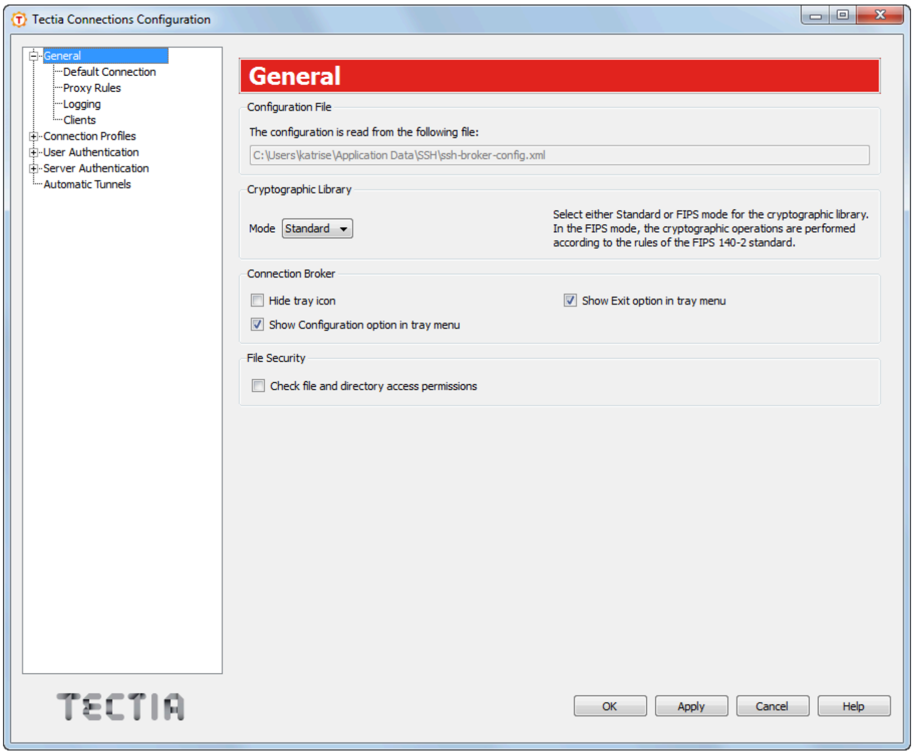
Easy installation

The installation process of Tectia products is effortless and requires no changes to the operating system. Tectia Client can also be easily installed by the end users themselves if the security policy allows.

Ease of Configuration

Tectia products allow complex security rules for connection setup, authentication, file transfers, and application tunneling definitions. The configuration files are defined in XML format by administrators.

On Windows, both Tectia Client and Server include also an intuitive GUI for locally configuring all relevant Secure Shell settings. The Connection Broker configuration can be edited in a GUI on Linux and Windows, and this tool is common to Tectia Client and ConnectSecure.



General

Configuration File

The configuration is read from the following file:

C:\Users\katrise\Application Data\SSH\ssh-broker-config.xml

Cryptographic Library

Mode: Standard

Select either Standard or FIPS mode for the cryptographic library. In the FIPS mode, the cryptographic operations are performed according to the rules of the FIPS 140-2 standard.

Connection Broker

☐ Hide tray icon

☑ Show Exit option in tray menu

☑ Show Configuration option in tray menu

File Security

☐ Check file and directory access permissions

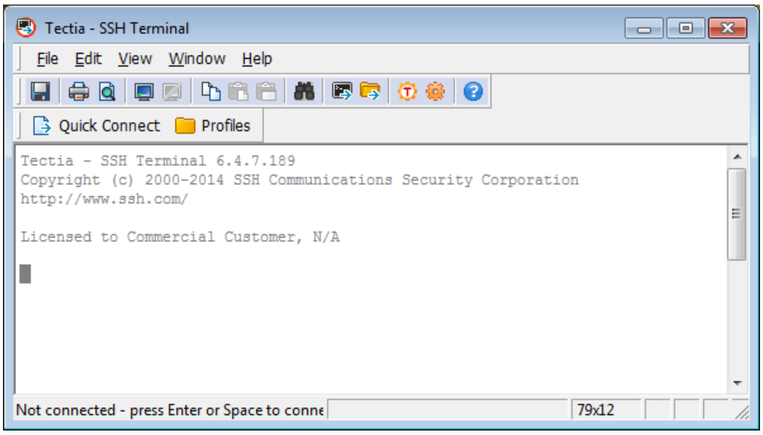
Figure 3.2. Tectia Connections Configuration GUI - General settings view

Windows domain authentication

To ease the end-user authentication, the Tectia client/server solution can be integrated with Windows domain authentication by using Kerberos/GSSAPI for fully transparent user authentication. Once the users are logged on to the domain, there is no need for additional interaction for Secure Shell user authentication.

Drop-in replacement for Telnet and FTP

The Tectia client/server solution provides easy and cost-effective means of securely replacing plaintext Telnet connections and file transfers in heterogeneous enterprise networks. Instead of plaintext connections to remote hosts, end users can use Tectia SSH Terminal that secures connections by encrypting all data. Alternatively, administrators can define that connections are automatically converted to secure SFTP or tunneled (encrypted) transparently to the users and their existing applications.


Tectia - SSH Terminal
File Edit View Window Help
Quick Connect
Profiles
Not connected - press Enter or Space to conne
79x12

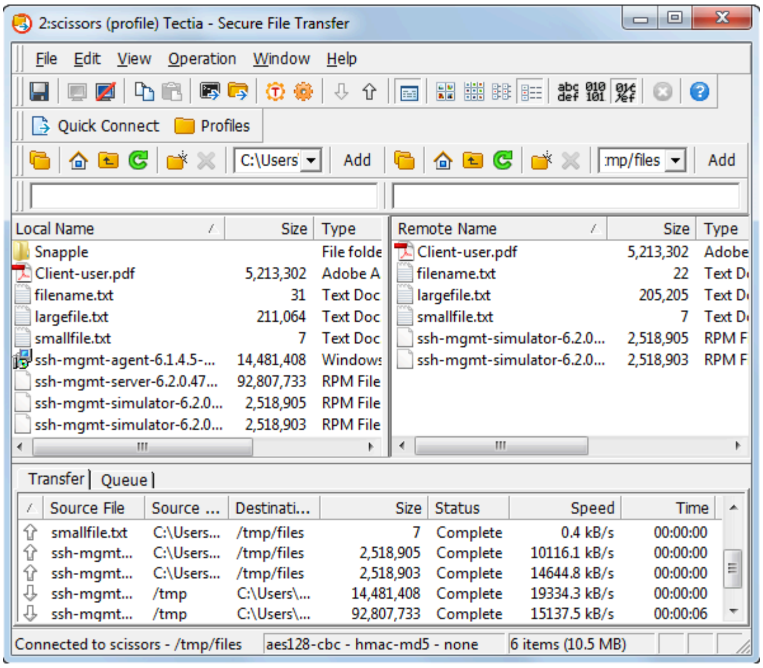
```
Tectia - SSH Terminal 6.4.7.189
Copyright (c) 2000-2014 SSH Communications Security Corporation
http://www.ssh.com/

Licensed to Commercial Customer, N/A
```

Figure 3.3. Tectia SSH Terminal for Secure Shell operations

Drag-and-drop file transfers

Windows users are provided with the Tectia Secure File Transfer user interface that allows users to securely drag-and-drop files between local Windows and remote Unix, Linux, Windows, and mainframe systems.


2:scissors (profile) Tectia - Secure File Transfer
File Edit View Operation Window Help
Quick Connect
Profiles
C:\Users
Add
mp/files
Add
Transfer
Queue

Local Name	Size	Type
Snapple		File folde
Client-user.pdf	5,213,302	Adobe A
filename.txt	31	Text Doc
largefile.txt	211,064	Text Doc
smallfile.txt	7	Text Doc
ssh-mgmt-agent-6.1.4.5-...	14,481,408	Windows
ssh-mgmt-server-6.2.0.47...	92,807,733	RPM File
ssh-mgmt-simulator-6.2.0...	2,518,905	RPM File
ssh-mgmt-simulator-6.2.0...	2,518,903	RPM File

Remote Name	Size	Type
Client-user.pdf	5,213,302	Adobe
filename.txt	22	Text D
largefile.txt	205,205	Text D
smallfile.txt	7	Text D
ssh-mgmt-simulator-6.2.0...	2,518,905	RPM F
ssh-mgmt-simulator-6.2.0...	2,518,903	RPM F

Source File	Source ...	Destinati...	Size	Status	Speed	Time
smallfile.txt	C:\Users...	/tmp/files	7	Complete	0.4 kB/s	00:00:00
ssh-mgmt...	C:\Users...	/tmp/files	2,518,905	Complete	10116.1 kB/s	00:00:00
ssh-mgmt...	C:\Users...	/tmp/files	2,518,903	Complete	14644.8 kB/s	00:00:00
ssh-mgmt...	/tmp	C:\Users\...	14,481,408	Complete	19334.3 kB/s	00:00:00
ssh-mgmt...	/tmp	C:\Users\...	92,807,733	Complete	15137.5 kB/s	00:00:06

Connected to scissors - /tmp/files | aes128-cbc - hmac-md5 - none | 6 items (10.5 MB)

Figure 3.4. Tectia Secure File Transfer GUI

3.4 Compatibility with IBM Mainframes

The Tectia product family includes a separate solution for IBM mainframes. Tectia Server for IBM z/OS is installed on mainframes, where it provides the same Secure Shell services as Tectia Server on other platforms, including secure file transfers, secure application connectivity, and secure remote access.

Both Tectia Client and ConnectSecure are capable of connecting to Tectia Server for IBM z/OS, and data can be transferred to and from the mainframes.

Tectia Client and ConnectSecure provide the following mainframe-specific features:

MVS data set handling

When used in conjunction with Tectia Server for IBM z/OS, users of Tectia Client and ConnectSecure can list IBM MVS (Multiple Virtual Storage) data sets as files and folders, facilitating seamless cross-platform file transfer between mainframe and non-mainframe systems.

ASCII/EBCDIC code set translation

Full and configurable ASCII/EBCDIC conversion is supported as well as configurable CONVXLAT conversion tables for seamless cross-platform compatibility between IBM z/OS and Unix/Linux/Windows hosts.

Tectia ConnectSecure offers additional SFTP functions and provides the following mainframe-specific features:

FTP-SFTP conversion

Tectia ConnectSecure provides FTP-SFTP conversion, which allows easy and cost-effective replacement of plaintext file transfers in large enterprise environments. Existing FTP scripts and client applications need no modifications. Only the FTP server will be eliminated and an SFTP server is used instead; the Tectia Server for IBM z/OS provides the SFTP server function.

Transparent FTP tunneling

Tectia ConnectSecure can capture and encrypt the file transfer connections to and from IBM mainframes. Transparent FTP tunneling provides a quick and easy way to secure FTP file transfers without the need to change the existing FTP scripts or applications.

Extended MVS streaming

When used in conjunction with Tectia Server for IBM z/OS, users of Tectia ConnectSecure can use extended streaming for direct, one-step processing of MVS data set transfers. Extended streaming provides optimized throughput, and on-the-fly conversions for file format and character sets between distributed hosts and IBM z/OS systems.

Tectia Server for IBM z/OS is described in the *Tectia Server for IBM z/OS Product Description*.

Tectia ConnectSecure is described in the *Tectia ConnectSecure Product Description*.

Chapter 4 Authentication

The Tectia client/server solution provides mutual authentication for the server and the client user. The Tectia client components authenticate the server and the Tectia Server authenticates the user's identity.

Secure Shell provides confidentiality also for user authentication. All data for identification and authentication purposes is exchanged in encrypted format between the client and server. User identity is not revealed to eavesdropping parties since the user is authenticated only after the connection has been secured.

Secure Shell supports multiple standardized methods for user authentication.

4.1 Server Authentication

Tectia uses cryptographic authentication for Secure Shell server hosts. Each server has a cryptographic key pair (a public key and a private key) that identifies the server. Whenever a Secure Shell client connects to a Secure Shell server, the server authenticates itself to the client cryptographically. This ensures that encryption and integrity protection are provided end-to-end between the client and the intended server. Server authentication also eliminates the danger of certain cryptographic attacks, especially man-in-the-middle attacks.

For the cryptographic authentication to work, the client must know the server's public key so that it can securely authenticate the server. The public key of the server must be distributed to each client. The private key of the server is never sent anywhere outside the server computer, but it is used by the server to create a digital signature that can then be verified by the client using the public key.

Secure Shell authenticates the Secure Shell server service of the server host. A host may run several Secure Shell server listeners on different ports. Each server can have a unique identity, if desired. However, typically there is only one Secure Shell server listener. If separate policies are needed or different services need to be offered for multiple use cases on a particular host, they can be defined dynamically in the Tectia Server configuration.

The server is authenticated with a digital signature based on a DSA, RSA, ECDSA or Ed25519 public-key algorithm. Each server must have a public-private key pair. In implementations without support for certificates, clients refer to a local database of trusted server public keys.

Secure Shell also supports certificate authentication. Servers can authenticate themselves to the client with X.509 v3 certificates or with OpenSSH certificates. When certificates are used, the client does not need to have a local database of trusted server public keys or server certificates. Instead, just the few trusted CA (certification authority) certificates or OpenSSH CA issuer keys are stored on the client, and the client trusts the servers whose certificates are signed by a trusted CA and certificate contents match the server hostname. Certificates provide scalability for authentication.

The Secure Shell server may have multiple identities (one RSA key, one DSA key, one ECDSA key, one Ed25519 key, one RSA certificate, one DSA certificate, one ECDSA certificate and one OpenSSH certificate). During the key exchange in the Secure Shell connection, the Secure Shell client and server agree on which identity will be used in server authentication. Typically, the client chooses based on what type of key it already has stored for the particular server or based on client's preference of signature algoritms.

4.2 User Authentication

Different methods can be used to authenticate users in Tectia. These authentication methods can be used separately or combined, depending on the level of functionality and security you want.

The Secure Shell server defines what methods are allowed in user authentication, and the Secure Shell client defines the order in which they will be tried.

In Tectia Server authentication chains can be used to define which methods are required and in which order they must be completed. For example, the server could require certificate or public-key authentication before requiring additional PAM authentication.

By default, the Tectia client/server solution uses these user authentication methods:

- public-key authentication
- password authentication
- keyboard-interactive
- GSSAPI

Public-key and certificate authentication are combined into the public-key authentication method.

The most commonly used user authentication methods are password, public-key, and X.509v3 certificate authentication. In public-key authentication, the users upload their public key files to the server and edit a configuration file. The server thus has a database of user public keys, similar to the client having its database of server public keys.

When certificates are used in user authentication, the user does not need to upload any public key files to the server prior to the connection, and the server does not need to have a database of user public keys or certificates. The server validates the user certificate by using the CA certificates that the server has been configured to trust and authorizes the login based on the user certificate contents.

Keyboard-interactive is not an authentication method in itself, but more like a common interface to various other authentication methods that are based on keyboard input. Password authentication, RSA SecurID, PAM (Pluggable Authentication Module), and RADIUS are examples of authentication methods that can be used over keyboard-interactive.

The highest security is achieved by using token-based certificate authentication where the certificate and the private key are stored on a cryptographic token, such as a smart card or HSM, or accessed via Tectia Zero Trust Module so that the user is never in posession of the private key. Secure Shell supports also several other strong authentication methods, including the proprietary RSA SecurID.

4.3 Strong Authentication

The Tectia client/server solution offers several methods for user and server authentication, and true strong authentication using either public keys or public-key certificates.

Server authentication with public keys or certificates

The Tectia client-side components authenticate the Secure Shell server in order to verify that they are connecting to the correct server. Likewise, the Secure Shell server authenticates the client user. The server can be authenticated by either (plain) public-key authentication or certificate authentication.

In (plain) public-key authentication, the server sends its public key to the client at the beginning of the first connection, and after the user has once verified and accepted the key, it is used in all future connections to that server.

In certificate authentication, the Tectia client-side components rely on a trusted third party, a certification authority (CA), to verify the server's identity. The signature of the certification authority in the server certificate guarantees the authenticity of the server certificate. When certificate authentication is used, the public key is included in the certificate that the server sends to the client.

User authentication with certificates and public keys

Client-side users can use certificates as proof of their identity. Certificates work like passports; the user proves his or her identity to a certificate authority once using public keys or other methods, receives a certificate, and from then on can authenticate using the certificate. In Tectia Zero Trust Edition the certificate is short-lived and access is verified for each new secure shell connection and periodically for ongoing connections.

Public keys

Public-key authentication (without certificates) provides an easy-to-deploy and secure means of authenticating the users without the need to deploy and maintain a public-key infrastructure (PKI). Users will create key pairs for themselves and upload the public keys to the server for verification.

Authentication agent

Tectia Client and ConnectSecure incorporate authentication agent functionality that allows the caching of passphrases, eliminating the need to retype the passphrase each time when a connection is made. Passphrases are used in public-key authentication, which is more secure than password authentication. In addition, authentication can be "forwarded" to another host, allowing administrators to hop from one server to another without the need to store private keys in multiple servers.

Passwords

Tectia supports secure password-based authentication. Unlike in plaintext protocols such as Telnet and FTP, passwords are never sent in plaintext format over the network, thus eliminating the risk of exposing the password to unauthorized parties.

X.509 v3 certificates

Tectia supports X.509 v3 certificates for further security and scalability in large and dynamic network environments. Comprehensive support for IETF PKIX and PKCS standards ensures seamless interoperability with third-party PKI products.

Flexible certificate revocation

Tectia supports both Certificate Revocation Lists (CRLs) and Online Certificate Status Protocol (OCSP) for centralized revocation of user credentials. CRLs are automatically fetched from a local file or by using HTTP or LDAP, depending on the local settings and the CRL Distribution Point extension in the certificate. CRLs can also be imported offline in legacy environments.

Certificate lifecycle management

Tectia supports IETF PKIX standards (CMPv2) and Cisco Systems' Simple Certificate Enrollment Protocol (SCEP) for online certificate enrollment. Certificates can also be imported by using the PKCS #12 envelope format supported by most Certification Authorities (CAs).

Smart cards and PKI tokens

Tectia Client and ConnectSecure support smart cards, USB tokens, and other PKI authentication devices by supporting PKCS #11 and MSCAPI for interfacing with authentication keys. Strong, two-factor authentication overcomes the inherent security issues of password authentication.

Host-based authentication on Unix

Host-based authentication is a form of delegated trust authentication, where the Secure Shell server trusts the Secure Shell client host to authenticate the user. The user is verified by a `suid` binary (ssh-signer) on the client host which then confirms the user identity to the server in a communication signed with a root-owned host key. The client host is authenticated strongly with public key cryptography, thus the authentication does not rely solely on a host IP address or domain name. The Secure Shell host-based authentication utilizes strong cryptography for host identity verification.

Keyboard-interactive

Keyboard-interactive is a standards-based method of integrating Secure Shell with third-party authentication mechanisms that are based on keyboard input, without the need to modify the client-side application (Tectia Client). Keyboard-interactive is commonly used in conjunction with PAM and RADIUS on the server side.

PAM support

Tectia Server supports Pluggable Authentication Module (PAM) for integrating with third-party authentication systems that have standards-based PAM libraries.

LDAP integration

Tectia Server can utilize standards-based third-party LDAP directories as centralized user repositories. The keyboard-interactive method and third-party PAM modules for LDAP can be used for integrating Tectia Server on Unix with LDAP directories.

RSA SecurID

Tectia Client, ConnectSecure, and Server support RSA SecurID for strong, two-factor authentication. The keyboard-interactive method is used for providing the password from Tectia Client or ConnectSecure to Server, which is integrated with the RSA Authentication Agent libraries for seamless interoperability.

RADIUS support

The RADIUS (Remote Authentication Dial-In User Service) protocol can be used with the Tectia client/server solution for checking users' authentication and authorization information from a remote server. Keyboard-interactive is used for sending the password to Tectia Server, which interfaces with the third-party RADIUS server such as Microsoft IAS or FreeRADIUS.

GSSAPI authentication (Kerberos)

Kerberos/GSSAPI authentication enables transparent, single sign-on authentication of Tectia Client users. Once the user has logged on to the network and received the logon credentials, there is no need to type in the authentication credentials again through Tectia Client user interface when accessing Secure Shell servers. Specifically, Kerberos/GSSAPI authentication enables the use of Windows domain authentication and Active Directory accounts with Tectia (SSPI API in Windows).

OpenSSH key support

Tectia Client, ConnectSecure, and Server support the legacy OpenSSH public-key format, eliminating the need for manual key conversions in multi-vendor Secure Shell environments. The key-compatibility feature also allows easy migration of OpenSSH environments to Tectia.

OpenSSH certificate support

Tectia Client, ConnectSecure, and Server support the OpenSSH certificates in addition to X.509v3 certificates.

Centrify DirectControl support

Integration of Tectia with Centrify DirectControl enables secure host access while leveraging Active Directory-based identity management throughout multi-platform enterprise networks.

Chapter 5 Use Cases

This chapter introduces typical use cases of the Tectia client/server solution. The use cases show examples of security services that you can quickly and effectively implement in typical enterprise environment.

For information on securing application connections, see Section 5.5 and Section 5.1.

For information on securing remote administrator connections, see Section 5.2 and Section 5.3.

5.1 Remote Access through Nested Tunnels

Tectia Client and ConnectSecure can be used to access Secure Shell servers on the corporate intranet from a remote location through nested tunnels. The transparent TCP tunneling feature (on Tectia ConnectSecure) is used to encrypt the application connections.

The first tunnel ends at the edge of the corporate intranet, and nested tunnels within the first tunnel continue to Tectia Servers located inside the intranet (as seen in Figure 5.1). Only one port needs to be open in the firewall. It is also possible to chain the nested tunnels within the intranet if necessary.

This helps to achieve end-to-end security without using NAT and a private IP inside the corporate network.

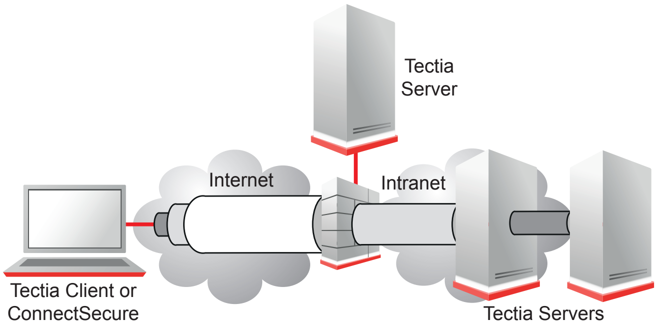


Figure 5.1. Remote access to Tectia Server with nested tunnels

5.2 Secure System Administration

One of the most widely used applications of Secure Shell-based products is to replace the unsecured login protocols, Rlogin, Telnet, and FTP, with secure alternatives. The Tectia client/server solution is based on SSH Secure Shell, used worldwide for secure system administration.

Figure 5.2 shows a typical Tectia environment for secure system administration. In this example, the managed servers reside in the perimeter network (DMZ), and the system administrator connects to them over the Internet using Tectia Client. In this scenario, Tectia Client software is installed in the system administrator's workstation and the Tectia Server software in the managed server.

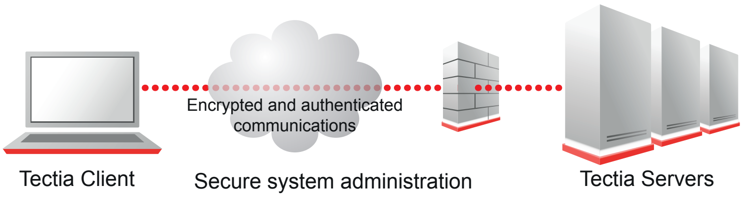

Encrypted and authenticated communications
Tectia Client
Secure system administration
Tectia Servers

Figure 5.2. Secure system administration with Tectia client/server solution

5.3 Secure System Administration with RSA SecurID

One of the benefits of the Tectia client/server solution is its wide support for various user authentication mechanisms, such as RSA SecurID. RSA SecurID is an authentication system that is based on a credit-card-sized authenticator token that generates a new passcode every 60 seconds. The user combines this code with a secret PIN code that enables secure login to protected resources.

The support for RSA SecurID has been implemented using a method called keyboard-interactive. It enables implementation of new authentication schemes based on keyboard interaction without the need to modify the client side. The keyboard-interactive authentication method is defined in *RFC 4256*.

In this use scenario (see Figure 5.3 for the overall architecture), Tectia Client is used to perform secure terminal-based administration and file transfer while using RSA SecurID for two-factor authentication. Tectia Server includes support for RSA Authentication Agent API. The RSA Authentication Agent must be installed on the same server. The Agent connects to the RSA Authentication Manager that performs the user identity validation against the one-time password generated by the token.

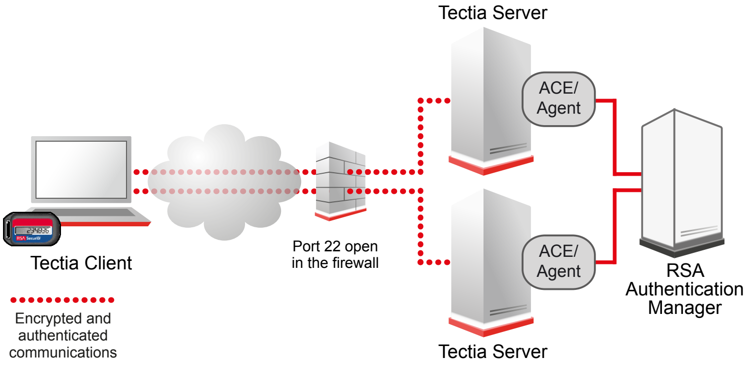

Tectia Server
ACE/ Agent
Tectia Client
Port 22 open in the firewall
ACE/ Agent
RSA Authentication Manager
Encrypted and authenticated communications
Tectia Server

Figure 5.3. Secure system administration using RSA SecurID for authentication

5.4 Integrating an Extranet Application with SFTP

Tectia Client can be integrated to an existing application through the SFTP API, for example, to an application responsible for data warehouse feeds, extranet applications, Windows desktop applications, or third-party file transfer management applications. The counterpart can be Tectia Server or another Secure Shell server.

In this scenario (see Figure 5.4), an end user needs to be able to download files from servers that are located on the Internet. However, direct access from the user workstations to those servers is not possible because of security reasons. Thus, the end users use a web access control application located on the perimeter network (DMZ). This application manages access rights to individual servers and files.

When the end user has selected the files to download, the web application triggers file transfer through the API. Tectia Client then transfers the files to the extranet and the user is provided HTML links to the files.

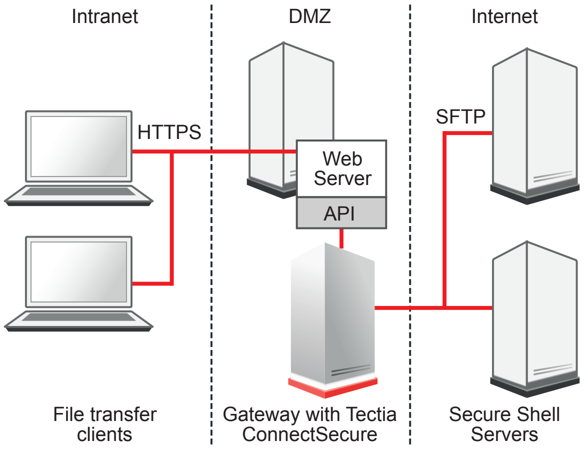


Figure 5.4. Integration through SFTP API

With this solution, end users are not accessing external services directly and it is easy to manage the access rights from a centralized web server. External parties do not have to modify their systems but can rely on the trusted Secure Shell standard.

5.5 Secure Application Login with Kerberos/GSSAPI

When both the client and server are located in the same Windows (NT or Active Directory) domain, it is possible to integrate Windows Domain logon to the Tectia client/server solution. This means that when a user logs on to a Windows Domain, the user gets a "ticket" that can be used for authenticating the user to Tectia Server. The authentication procedure is then non-interactive; the user is not prompted to enter a password when Tectia Client or ConnectSecure are connecting to Tectia Server.

GSSAPI authentication can also be used in a mixed Windows/Unix environment. On Unix, GSSAPI interoperates with standard Kerberos implementations that provide a GSSAPI mechanism.

Active Directory/Kerberos is used in the Windows 2003 Domains.

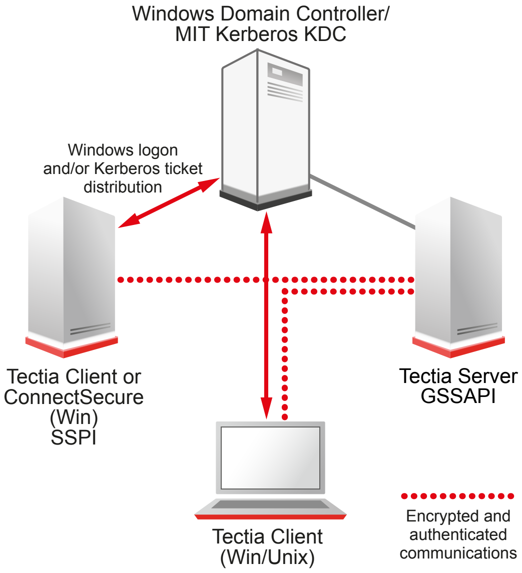


Figure 5.5. Secure application connectivity through GSSAPI

Chapter 6 Product Specification

In this section you can find information on the supported operating systems, hardware requirements and supported authentication methods, cryptographic algorithms and protocols, and third-party products.

6.1 Supported Operating Systems

Tectia client/server solution products can be installed on Linux, Unix, and Windows platforms and they can run on any standard hardware capable of running the supported operating systems. A separate Tectia Server for IBM z/OS is available for mainframes; for its description, refer to *Tectia Server for IBM z/OS Product Description*.

Tectia Server includes also the client-side tools that can be optionally installed on the same host.

Note

Keep the operating system always fully patched, according to recommendations by the operating system vendor. The minimum patch levels required by Tectia products are mentioned in the Tectia product-specific installation instructions in the User Manuals and Administrator Manuals.

Table 6.1. Supported operating systems for Tectia Client and Server

Operating System	Client	Server
HP-UX (PA-RISC) [a]	11i v3	11i v3
HP-UX (IA-64) [a]	11i v3	11i v3
IBM AIX (POWER)	7.1, 7.2, 7.3	7.1, 7.2, 7.3
Oracle Solaris (SPARC)	10, 11	10, 11
Oracle Solaris (x86-64)	10, 11	10, 11
Red Hat Enterprise Linux (x86-64)	7, 8, 9	7, 8, 9
Rocky Linux (x86-64)	8, 9	8, 9
Ubuntu (x86-64)	18.04, 20.04, 22.04	18.04, 20.04, 22.04
Debian GNU/Linux (x86-64)	11, 12	11, 12

Operating System	Client	Server
SUSE LINUX Enterprise Desktop (x86-64)	15	15
SUSE LINUX Enterprise Server (x86-64)	12, 15	12, 15
Microsoft Windows (x86-64)	10, 11, Server 2016, Server 2019, Server 2022	10, 11, Server 2016, Server 2019, Server 2022

[a] for this platform PQC is not supported.

6.2 Hardware and Space Requirements

The Tectia products can be run on any standard hardware capable of running the supported operating system versions. The machine should have a TCP/IP connection.

Table 6.2 summarizes the memory and disk space requirements.

Table 6.2. Memory and disk space requirements

	Client	ConnectSecure	Server
RAM	any	any	1 GB[a]
Disk space	100 MB	100 MB	100 MB

[a] For hundreds of simultaneous tunnels

6.3 Tectia Features per Product

The following table lists the features provided by Tectia Client, ConnectSecure, and Server.

Table 6.3. Tectia products and their features

Feature	Client	ConnectSecure	Server
FTP-SFTP conversion		x	
Transparent FTP tunneling		x	
Transparent TCP tunneling		x	
SFTP API for C (on all platforms)		x	
SFTP API for Java 1.4.2 (on 32-bit Linux, Solaris SPARC, and 32-bit and 64-bit Windows)		x	
Checkpoint/restart mechanism	x	x	x

Feature	Client	ConnectSecure	Server
Streaming for high-speed transfers	x[a]	x[a]	x
Prefix for files in transfer	x	x	x
Versatile command-line tools (sshg3, scpg3, sftpg3)	x	x	x
Terminal GUI on Windows	x	x	
File transfer GUI on Windows	x	x	
Configuration GUI on Windows	x	x	x
CryptiCore encryption (on Linux and Windows)	x	x	x
Support for connections to standard Secure Shell v2 servers	x	x	x
Secure terminal server			x
FTP-SFTP conversion (server side)			x
SFTP server			x
Transparent tunneling server			x
Support for SFTP API (from other platforms)			x

[a] Requires Tectia Server as the counterpart.

6.4 Supported Authentication Methods

6.4.1 Supported User Authentication Methods

The following user authentication methods are supported in the Tectia client/server solution.

Table 6.4. User authentication methods supported by the Tectia client/server solution

Authentication method	Tectia Server		Tectia Client and ConnectSecure	
	Unix	Windows	Unix	Windows
Password[a]	x	x	x	x

Authentication method	**Tectia Server**		**Tectia Client and ConnectSecure**	
	Unix	**Windows**	**Unix**	**Windows**
Public-key	x	x	x	x
Certificate	x	x	x	x
Host-based	x	x	x	
Keyboard-interactive	x	x	x	x
PAM[b]	x		x	x
RSA SecurID[b]	x	x	x	x
RADIUS[b]	x	x	x	x
GSSAPI/Kerberos	x	x	x	x

[a] On SELinux enabled systems, password method uses PAM internally on the server side.

[b] Through keyboard-interactive.

6.4.2 Compatibility with OpenSSH Keys

By default, the Tectia client/server solution uses private and public keys stored in the IETF standard Secure Shell v2 format. However, Tectia Client and Server can also use keys and related files in the legacy OpenSSH format.

The following OpenSSH-format keys are supported:

- server host key pair
- trusted server host public keys, which clients use to authenticate servers
- user private keys (used by clients to authenticate to a server)
- authorized user public keys (used by a server to authenticate users), including public-key options

6.5 Supported Protocols and Standards

6.5.1 FIPS-Certified Cryptographic Library

Tectia Client, ConnectSecure, and Server can be operated in FIPS mode, using a version of the cryptographic library that has been certified according to the Federal Information Processing Standard (FIPS) 140-2.

FIPS mode is supported on Linux, Solaris, and Windows.

6.6 Supported Third-Party Products

This section lists the third-party products that have been tested to work with the Tectia client/server solution.

6.6.1 Smart Cards/Hardware Tokens (Windows)

- RSA Authentication Manager 5.2 (for SecurID)
- RSA Authentication Manager 6.0 (for SecurID)
- RSA Authentication Manager 6.1 (for SecurID)
- RSA Authentication Agents (for SecurID)
- Secure Computing SafeWord PremierAccess 3.2
- ActivCard ActivClient 5.3 (Supports smart cards from multiple vendors)
- ActivCard Gold 2.2 for CAC (Supports smart cards from multiple vendors)
- Aladdin eToken with eToken RTE v3.60 on supported Windows platforms
- Safenet iKey

6.6.2 Certificate Authorities

- PrivX for just-in-time user certificate authentication
- OpenSSH certificate issuer
- Microsoft Windows 2000/2003 and later Certificate Authority
- RSA Keon

6.6.3 Other Supported Third-Party Products

- RADIUS support with Microsoft IAS and FreeRADIUS
- Centrify Direct Control 3.0

Appendix A Default and Supported SSH Algorithms

This section describes the SSH algorithms supported by Tectia products.

A.1 Ciphers

Table A.1. Default ciphers (in order of client-side preference)

Name in XML	Name in GUI	FIPS
crypticore128@ssh.com	CryptiCore (Tectia)	
aes128-gcm@openssh.com	AES-128-GCM (OpenSSH)	•
aes256-gcm@openssh.com	AES-256-GCM (OpenSSH)	•
AEAD_AES_128_GCM	AEAD_AES_128_GCM	•
AEAD_AES_256_GCM	AEAD_AES_256_GCM	•
aes128-ctr	AES-128-CTR	•
aes192-ctr	AES-192-CTR	•
aes256-ctr	AES-256-CTR	•

Table A.2. All supported ciphers

Name in XML	Name in GUI	FIPS
3des-cbc	3DES	•
AEAD_AES_128_GCM	AEAD_AES_128_GCM	•
AEAD_AES_256_GCM	AEAD_AES_256_GCM	•
aes128-cbc	AES-128-CBC	•

Name in XML	Name in GUI	FIPS
aes128-ctr	AES-128-CTR	•
aes128-gcm@openssh.com	AES-128-GCM (OpenSSH)	•
aes192-cbc	AES-192-CBC	•
aes192-ctr	AES-192-CTR	•
aes256-cbc	AES-256-CBC	•
aes256-ctr	AES-256-CTR	•
aes256-gcm@openssh.com	AES-256-GCM (OpenSSH)	•
arcfour	Arcfour	
blowfish-cbc	Blowfish	
cast128-cbc		
crypticore128@ssh.com	CryptiCore (Tectia)	
idea-cbc		
seed-cbc@ssh.com	SEED (Tectia)	
twofish128-cbc	Twofish-128	
twofish192-cbc	Twofish-192	
twofish256-cbc	Twofish-256	
twofish-cbc	Twofish	

A.2 Key-Exchange Algorithms

Table A.3. Default KEXs (in order of client-side preference)

Name in XML	Name in GUI	FIPS
ecdh-nistp521-kyber1024-sha512@ssh.com	PQC: ecdh-nistp521-kyber1024-sha512 (Tectia)	•
curve25519-frodokem1344-sha512@ssh.com	PQC: curve25519-frodokem1344-sha512 (Tectia)	•
sntrup761x25519-sha512@openssh.com	PQC: sntrup761x25519-sha512 (OpenSSH)	•
diffie-hellman-group-exchange-sha256	DH-GEX-SHA256	
diffie-hellman-group16-sha512	DH-Group16-SHA512	•
diffie-hellman-group18-sha512	DH-Group18-SHA512	•
diffie-hellman-group14-sha256	DH-Group14-SHA256	•
diffie-hellman-group14-sha256@ssh.com	DH-Group14-SHA256 (Tectia)	•
curve25519-sha256	Curve25519-sha256	•
curve25519-sha256@libssh.org	Curve25519-sha256 (libssh)	•

Table A.4. All supported KEXs

Name in XML	Name in GUI	FIPS
curve25519-frodokem1344-sha512@ssh.com	PQC: curve25519-frodokem1344-sha512 (Tectia)	•

Name in XML	Name in GUI	FIPS
curve25519-sha256	Curve25519-sha256	•
curve25519-sha256@libssh.org	Curve25519-sha256 (libssh)	•
curve448-kyber1024-sha512@ssh.com	PQC: curve448-kyber1024-sha512 (Tectia)	•
diffie-hellman-group14-sha1	DH-Group14-SHA1	•
diffie-hellman-group14-sha224@ssh.com	DH-Group14-SHA224 (Tectia)	•
diffie-hellman-group14-sha256	DH-Group14-SHA256	•
diffie-hellman-group14-sha256@ssh.com	DH-Group14-SHA256 (Tectia)	•
diffie-hellman-group15-sha256@ssh.com	DH-Group15-SHA256 (Tectia)	•
diffie-hellman-group15-sha384@ssh.com	DH-Group15-SHA384 (Tectia)	•
diffie-hellman-group16-sha384@ssh.com	DH-Group16-SHA384 (Tectia)	•
diffie-hellman-group16-sha512	DH-Group16-SHA512	•
diffie-hellman-group16-sha512@ssh.com	DH-Group16-SHA512 (Tectia)	•
diffie-hellman-group18-sha512	DH-Group18-SHA512	•
diffie-hellman-group18-sha512@ssh.com	DH-Group18-SHA512 (Tectia)	•
diffie-hellman-group1-sha1	DH-Group1-SHA1	
diffie-hellman-group-exchange-sha1	DH-GEX-SHA1	
diffie-hellman-group-exchange-sha224@ssh.com	DH-GEX-SHA224 (Tectia)	
diffie-hellman-group-exchange-sha256	DH-GEX-SHA256	
diffie-hellman-group-exchange-sha384@ssh.com	DH-GEX-SHA384 (Tectia)	
diffie-hellman-group-exchange-sha512@ssh.com	DH-GEX-SHA512 (Tectia)	
ecdh-nistp521-firesaber-sha512@ssh.com	PQC: ecdh-nistp521-firesaber-sha512 (Tectia)	•
ecdh-nistp521-kyber1024-sha512@ssh.com	PQC: ecdh-nistp521-kyber1024-sha512 (Tectia)	•
ecdh-sha2-nistp256	ECDH-NISTP256	•
ecdh-sha2-nistp384	ECDH-NISTP384	•
ecdh-sha2-nistp521	ECDH-NISTP521	•
sntrup761x25519-sha512@openssh.com	PQC: sntrup761x25519-sha512 (OpenSSH)	•

A.3 Message-Authentication Codes

Table A.5. Default MACs (in order of client-side preference)

Name in XML	Name in GUI	FIPS
crypticore-mac@ssh.com	CryptiCore (Tectia)	
hmac-sha2-256	HMAC-SHA2-256	•
hmac-sha256-2@ssh.com	HMAC-SHA256-2 (Tectia)	•
hmac-sha2-512	HMAC-SHA2-512	•

Name in XML	Name in GUI	FIPS
hmac-sha512@ssh.com	HMAC-SHA512 (Tectia)	•
hmac-sha1	HMAC-SHA1	•

Table A.6. All supported MACs

Name in XML	Name in GUI	FIPS
crypticore-mac@ssh.com	CryptiCore (Tectia)	
hmac-md5	HMAC-MD5	
hmac-md5-96	HMAC-MD5-96	
hmac-md5-96-etm@openssh.com	HMAC-MD5-96-ETM (OpenSSH)	
hmac-md5-etm@openssh.com	HMAC-MD5-ETM (OpenSSH)	
hmac-sha1	HMAC-SHA1	•
hmac-sha1-96	HMAC-SHA1-96	
hmac-sha1-96-etm@openssh.com	HMAC-SHA1-96-ETM (OpenSSH)	
hmac-sha1-etm@openssh.com	HMAC-SHA1-ETM (OpenSSH)	•
hmac-sha224@ssh.com	HMAC-SHA224 (Tectia)	
hmac-sha2-256	HMAC-SHA2-256	•
hmac-sha2-256-etm@openssh.com	HMAC-SHA2-256-ETM (OpenSSH)	•
hmac-sha2-512	HMAC-SHA2-512	•
hmac-sha2-512-etm@openssh.com	HMAC-SHA2-512-ETM (OpenSSH)	•
hmac-sha256@ssh.com	HMAC-SHA256 (Tectia/Old)	
hmac-sha256-2@ssh.com	HMAC-SHA256-2 (Tectia)	
hmac-sha384@ssh.com	HMAC-SHA384 (Tectia)	
hmac-sha512@ssh.com	HMAC-SHA512 (Tectia)	

A.4 Host-Key and Public Key Signature Algorithms

Table A.7. Default host-key algorithms (in order of client-side preference)

Name in XML	Name in GUI	FIPS
rsa-sha2-512	rsa-sha2-512	•
rsa-sha2-256	rsa-sha2-256	•
ssh-rsa-sha256@ssh.com	ssh-rsa-sha256 (Tectia)	•
ecdsa-sha2-nistp521	ecdsa-sha2-nistp521	•
ecdsa-sha2-nistp384	ecdsa-sha2-nistp384	•
ecdsa-sha2-nistp256	ecdsa-sha2-nistp256	•
x509v3-sign-rsa-sha256@ssh.com	x509v3-sign-rsa-sha256 (Tectia)	•
x509v3-ecdsa-sha2-nistp256	x509v3-ecdsa-sha2-nistp256	•
x509v3-ecdsa-sha2-nistp384	x509v3-ecdsa-sha2-nistp384	•
x509v3-ecdsa-sha2-nistp521	x509v3-ecdsa-sha2-nistp521	•

Name in XML	Name in GUI	FIPS
x509v3-rsa2048-sha256	x509v3-rsa2048-sha256	•
ssh-ed25519	ssh-ed25519	•
ecdsa-sha2-nistp256-cert-v01@openssh.com	ecdsa-sha2-nistp256-cert-v01@openssh.com	•
ecdsa-sha2-nistp384-cert-v01@openssh.com	ecdsa-sha2-nistp384-cert-v01@openssh.com	•
ecdsa-sha2-nistp521-cert-v01@openssh.com	ecdsa-sha2-nistp521-cert-v01@openssh.com	•
ssh-ed25519-cert-v01@openssh.com	ssh-ed25519-cert-v01@openssh.com	•
rsa-sha2-256-cert-v01@openssh.com	rsa-sha2-256-cert-v01@openssh.com	•
rsa-sha2-512-cert-v01@openssh.com	rsa-sha2-512-cert-v01@openssh.com	•

Table A.8. All supported host-key and public key signature algorithms

Name in XML	Name in GUI	FIPS
ecdsa-sha2-nistp256	ecdsa-sha2-nistp256	•
ecdsa-sha2-nistp256-cert-v01@openssh.com	ecdsa-sha2-nistp256-cert-v01@openssh.com	•
ecdsa-sha2-nistp384	ecdsa-sha2-nistp384	•
ecdsa-sha2-nistp384-cert-v01@openssh.com	ecdsa-sha2-nistp384-cert-v01@openssh.com	•
ecdsa-sha2-nistp521	ecdsa-sha2-nistp521	•
ecdsa-sha2-nistp521-cert-v01@openssh.com	ecdsa-sha2-nistp521-cert-v01@openssh.com	•
rsa-sha2-256	rsa-sha2-256	•
rsa-sha2-256-cert-v01@openssh.com	rsa-sha2-256-cert-v01@openssh.com	•
rsa-sha2-512	rsa-sha2-512	•
rsa-sha2-512-cert-v01@openssh.com	rsa-sha2-512-cert-v01@openssh.com	•
ssh-dss	ssh-dss	
ssh-dss-cert-v01@openssh.com	ssh-dss-cert-v01@openssh.com	
ssh-dss-sha224@ssh.com	ssh-dss-sha224 (Tectia)	•
ssh-dss-sha256@ssh.com	ssh-dss-sha256 (Tectia)	•
ssh-dss-sha384@ssh.com	ssh-dss-sha384 (Tectia)	•
ssh-dss-sha512@ssh.com	ssh-dss-sha512 (Tectia)	•
ssh-ed25519	ssh-ed25519	•
ssh-ed25519-cert-v01@openssh.com	ssh-ed25519-cert-v01@openssh.com	•
ssh-rsa	ssh-rsa	
ssh-rsa-cert-v01@openssh.com	ssh-rsa-cert-v01@openssh.com	
ssh-rsa-sha224@ssh.com	ssh-rsa-sha224 (Tectia)	•
ssh-rsa-sha256@ssh.com	ssh-rsa-sha256 (Tectia)	•
ssh-rsa-sha384@ssh.com	ssh-rsa-sha384 (Tectia)	•
ssh-rsa-sha512@ssh.com	ssh-rsa-sha512 (Tectia)	•
x509v3-ecdsa-sha2-nistp256	x509v3-ecdsa-sha2-nistp256	•
x509v3-ecdsa-sha2-nistp384	x509v3-ecdsa-sha2-nistp384	•
x509v3-ecdsa-sha2-nistp521	x509v3-ecdsa-sha2-nistp521	•
x509v3-rsa2048-sha256	x509v3-rsa2048-sha256	•

Name in XML	Name in GUI	FIPS
x509v3-sign-dss	x509v3-sign-dss	
x509v3-sign-dss-sha224@ssh.com	x509v3-sign-dss-sha224 (Tectia)	•
x509v3-sign-dss-sha256@ssh.com	x509v3-sign-dss-sha256 (Tectia)	•
x509v3-sign-dss-sha384@ssh.com	x509v3-sign-dss-sha384 (Tectia)	•
x509v3-sign-dss-sha512@ssh.com	x509v3-sign-dss-sha512 (Tectia)	•
x509v3-sign-rsa	x509v3-sign-rsa	
x509v3-sign-rsa-sha224@ssh.com	x509v3-sign-rsa-sha224 (Tectia)	•
x509v3-sign-rsa-sha256@ssh.com	x509v3-sign-rsa-sha256 (Tectia)	•
x509v3-sign-rsa-sha384@ssh.com	x509v3-sign-rsa-sha384 (Tectia)	•
x509v3-sign-rsa-sha512@ssh.com	x509v3-sign-rsa-sha512 (Tectia)	•
x509v3-ssh-dss	x509v3-ssh-dss	
x509v3-ssh-rsa	x509v3-ssh-rsa	