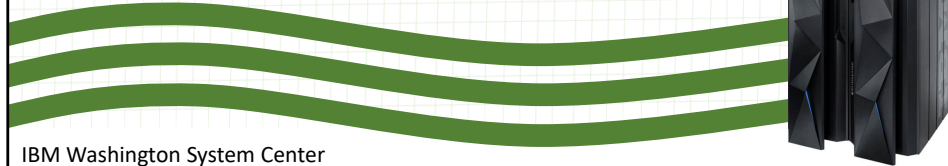


# Securing and Encrypting Network Traffic with z/OS Communications Server and Policy Agent

## Security Workshop

### Comparison of IPsec, AT-TLS, and SSH FTP Traffic



IBM Washington System Center  
IBM Technical Sales Support

# Trademarks

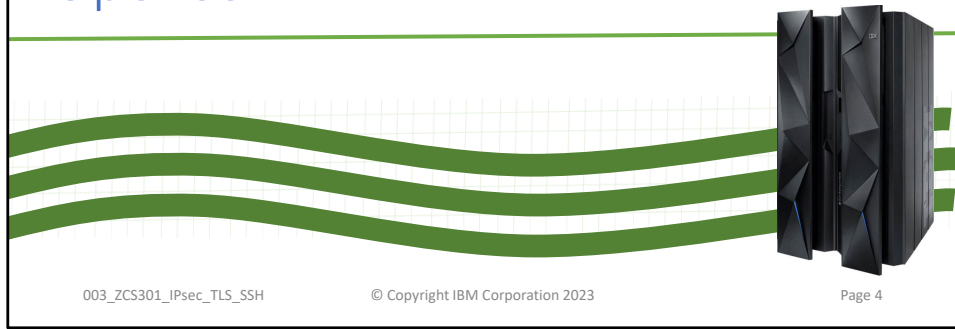
- **The following are Registered Trademarks of the International Business Machines Corporation in the United States and/or other countries.**
  - IBM
  - z/OS
- **The following are trademarks or registered trademarks of other companies.**
  - Microsoft is a registered trademark of Microsoft Corporation in the United States and other countries.
- All other products may be trademarks or registered trademarks of their respective companies.
- Refer to [www.ibm.com/legal](http://www.ibm.com/legal) for further legal information.
  
- OSA-Express Features
- There are many different types of OSA-Express features. In this document where OSA is mentioned it refers to an OSA port on any of the OSA-Express features unless a specific OSA-Express feature is mentioned.

## Agenda

---

- Standard protocols: IPsec, TLS, and OpenSSH
- What does it mean to "secure" or "protect" a file transfer?
- Ambiguous Acronyms: What Does Secured FTP Mean?
- Protocol Comparison: Transferring Files Securely with TCP/IP
- Comparison of Offload and Cryptographic Hardware Usage by SSL/TLS, IPsec, SSH

## Standard protocols: IPsec, TLS, and OpenSSH

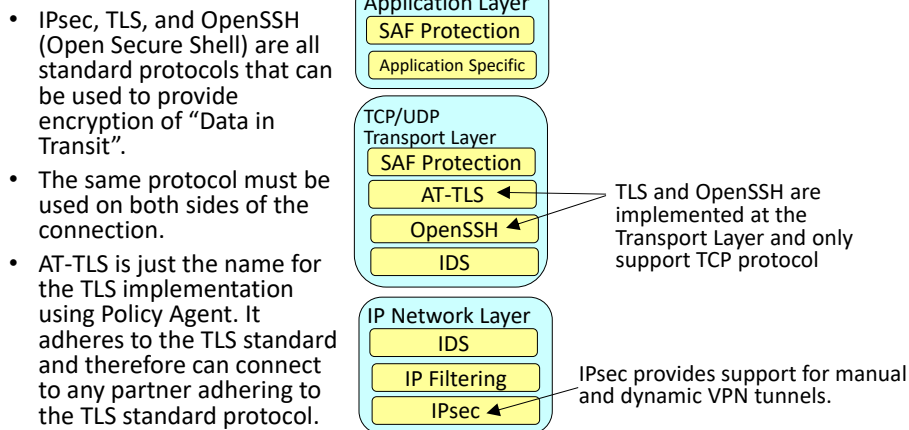


003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

Page 4

## Protocol Stack View of TCP/IP Security Features



003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

Page 5

z/OS CS TCP/IP applications use SAF to authenticate users and prevent unauthorized access to datasets, files, and SERVAUTH protected resources.

The SAF SERVAUTH class is used to prevent unauthorized user access to TCP/IP resources (stack, ports, networks).

Intrusion detection services (IDS) protects against attacks of various types on the system's legitimate (open) services. IDS protection is provided at both the IP and transport layers.

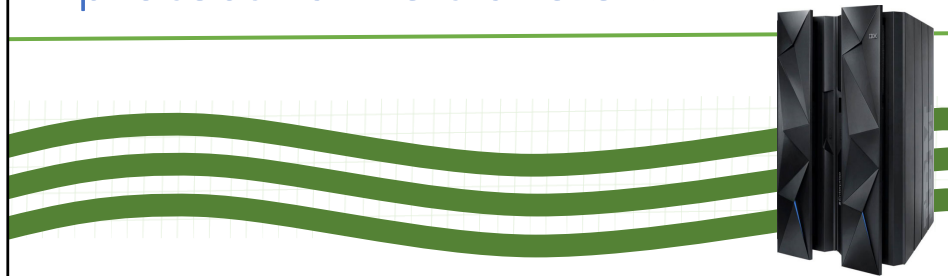
IP packet filtering blocks out all IP traffic that this systems doesn't specifically permit.

Examples of application protocols with built-in security extensions are SNMPv3, DNS, and OSPF.

Both Kerberos and SSL/TLS are located as extensions to the sockets APIs and applications have to be modified to make use of these security functions. Both SSL/TLS and Kerberos are connection-based and only applicable to TCP (stream sockets) applications, not UDP. AT-TLS is a TCP/IP stack service that provides SSL/TLS services at the TCP transport layer and is transparent to upper-layer protocols. It is available to TCP applications in all programming languages except PASCAL.

IPSec resides at the networking layer and is transparent to upper-layer protocols, including both transport layer protocol and application protocol.

What does it mean to “secure” or  
“protect” a file transfer?



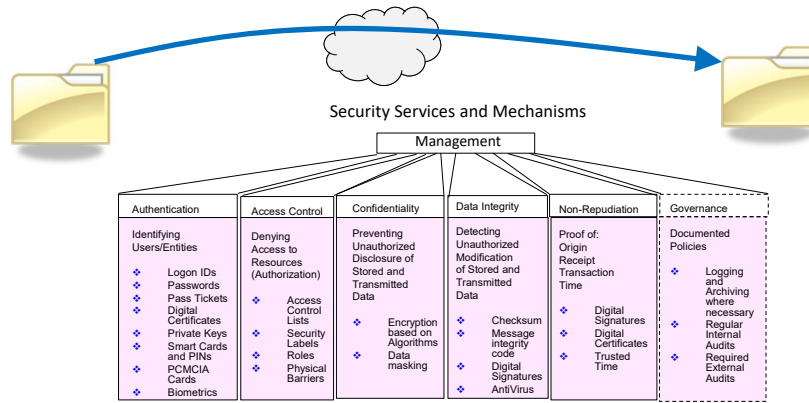
003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

Page 6

## Protection and Security Services for Transferring a File

1. Authenticate the sender and/or recipient.
2. Keep the data confidential through encryption or masking.
3. Verify the data has not been changed in flight.
4. Verify that the sender is the legitimate owner of the identity he has assumed.



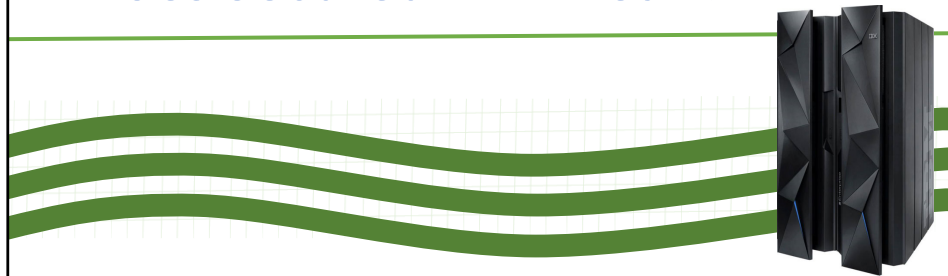
003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

Page 7

Remember the ISO Security Architecture and that we are focusing on protecting “Data in Flight”, data being sent from one host to another.

## Ambiguous Acronyms: What Does Secured FTP Mean?



003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

Page 8

## The Acronyms --- Confusion

Acronym	Meaning	RFC	Function
SFTP	Simple File Transfer Protocol	913	File Transfer - TCP Port 115
TFTP	Trivial File Transfer Protocol	1350	File Transfer - UDP Port 69
SCP	Secure Copy Program	BSD Remote Copy Protocol	Secure File Transfer over SSH - TCP Port 22
FTP	File Transfer Protocol	959 & 2428	File Transfer - TCP Ports 20 & 21
SFTP	SSL File Transfer Protocol	959, 2428 & 4217	Secure File Transfer – TCP Ports 20 & 21
FTPS	File Transfer Protocol Secure or File Transfer Protocol SSL	959, 2228 & 4217	Secure File Transfer – TCP Ports 20 & 21
ftpd & ftpdms	file transfer protocol daemon & file transfer protocol new server	959, 2228 & 4217	File Transfer Unix processes for FTP Server and forked task for client log in – TCP Ports 20 & 21
SFTP & FTP over SSH	SSH File Transfer Protocol or File Transfer Protocol over SSH	959, 2428 & 4251	Secure File Transfer - TCP Port 22
SSH	Secured Shell	4251	Secure Tunnel for communication
sftp & sftpd	secure file transfer protocol client & daemon	959, 2428 & 4251	Secure File Transfer server and listener - TCP Port 22
MFTP	Managed File Transfer Protocol	Proprietary	File Transfer with automated recovery

003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

Page 9

As you can see, some acronyms have several different meanings.

Simple File Transfer Protocol (SFTP) provides unsecure file transfer.

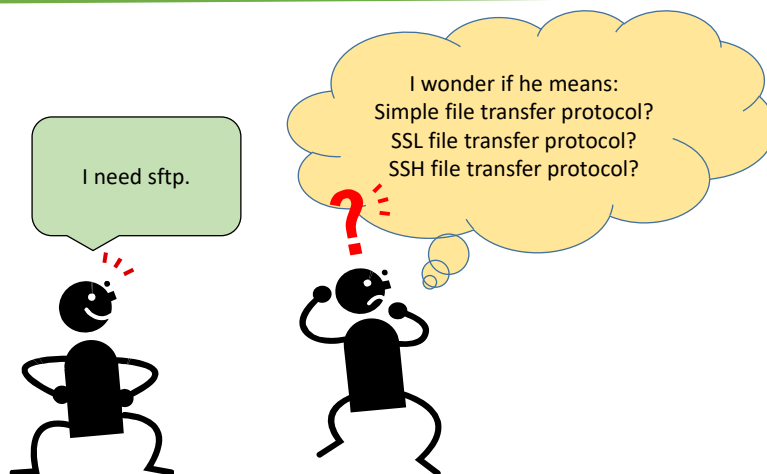
Trivial File Transfer Protocol (TFTP) provides a limited unsecure file transfer, usually boot files for routers.

Secured Shell (SSH) provides a secure tunnel for file transfer, terminal commands interaction, and other.

Secure file transfer protocol client (sftp) is the SSH client command for file transfer.

Secure file transfer protocol daemon (sftpd) is the SSH file transfer server listener.

## The Acronyms --- Confusion



003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

Page 10

sftp can mean several different things.

## Solution

- Don't use terms or acronyms:
  - Secured FTP
  - SFTP or sftp
- Ask what they mean when someone else uses them...
  - What technology is being used to secure the File Transfer?
    - SSH
    - SSL, TLS, AT-TLS
    - VPN (IPsec Virtual Private Network)
    - Proprietary coding
    - Other
  - What Security Service is required?
    - Authentication
    - Access Control
    - Confidentiality (Encryption, Data Masking)
    - Data Integrity Preservation
    - Non-repudiation
    - Recovery/Restart Capability
  - What platforms are involved?
  - What types of files need to be transferred?
    - File Organization: Record (MVS), Stream, VSAM, DB2, etc.
  - Is FIPS 140 required?
  - Other?

003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

Page 11

Always clarify what you and others are talking about.

## Protocol Comparison: Transferring Files Securely with TCP/IP

### Note:

Secure File Transfer is only one aspect for File Transfer Protocols. Certain file transfer protocols have application extensions for security that we do not cover here. Please consult the IP Configuration Reference or other presentations on Security in FTP.



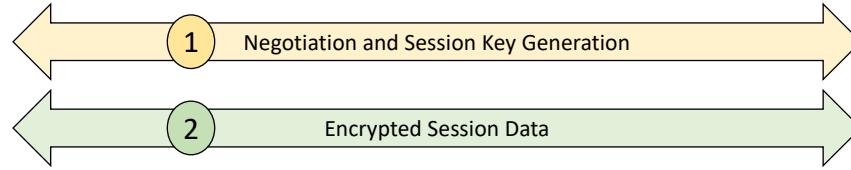
003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

Page 12

User Exits  
Secure Hostname check  
Verify user check  
etc.

## General Architecture of Encryption Flow



Encryption Flow	What Happens	SSL/TLS Terminology	IPsec Terminology	OpenSSH Terminology
Stage 1 Asymmetric Algorithms	Negotiation of Secure Connection: Authentication and Generation of and encrypted Transmit of Session Key	Handshake Layer	Phase I Phase II	No official terminology; just negotiation stage
Stage 2 Symmetric Algorithms	Encryption and Decryption of Data Payload (Session Data)	Record Layer	Phase II Tunnel	No official terminology; just data transfer stage

- Essentially all these security protocols use the same basic architecture:

- Authenticate the partner; generate a symmetric key
  - Encrypt symmetric key with asymmetric algorithm and send
- Encrypt session data with symmetric ("Session") key and transmit session data

003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

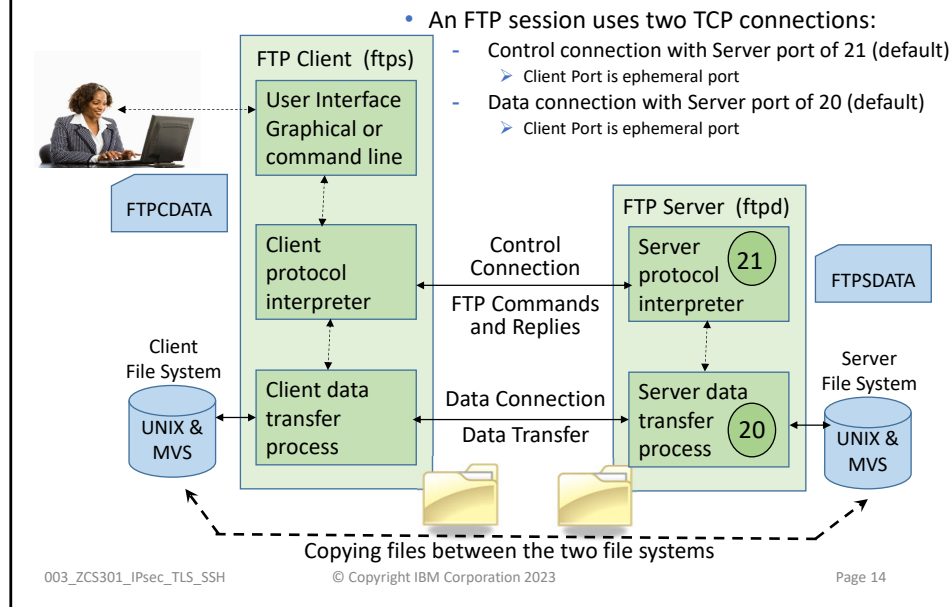
Page 13

From the z/OS Security Server RACF Security Administrator's Guide, Chapter 21 "RACF and Digital Certificates":

"Each party, both client and server, has its own certificate, a matching private key, and a list of trusted certificate-authority (CA) certificates. When the client needs to authenticate itself to the server to be able to perform a transaction, both the server and client need to verify one another. The protocol for a secure handshake for mutual verification begins with the parties exchanging certificates. Each party then separately validates the other's certificate to make sure that its signature is valid, that the subject name in the certificate is correct, and that the certificate originated from a trusted certificate authority. If successful, each party must prove to the other that it owns the private key that matches its public key certificate. This step establishes proof of possession and can be accomplished by having each party sign a known unique value, such as a hash of the message traffic between the two parties. If each signature can be validated using the associated public key, the proofs are successful. The final step in this handshake is for one of the parties to generate a random symmetric key, encrypt it using the other party's public key, and send it to the other party. This random symmetric key may then be used to encrypt the data for the remainder of the session. Once the secure handshake is complete, secure transactions can be safely handled in the z/OS environment between this client and server."

Note that IPsec with IKEv2 uses two negotiation phases. The second negotiation phase is the one that transmits the symmetric data or session key that will be used for the subsequent data encryption over what is called the "Phase 2 Tunnel."

# File Transfer Protocol (FTP) (RFC 959)



FTP is one of the most widely used applications in a TCP/IP network.

The basic functions of FTP are defined in an old RFC - RFC959. Some additions and clarifications have been documented in later RFCs, but RFC959 is still the current FTP RFC. There are other RFCs besides RFC 959 that are part of most FTP implementations. These RFCs include FTP subcommand enhancements and security enhancements.

An FTP session uses at least two TCP connections:

- One for the control connection. Server port is 21. This connection stays up during the whole session.
- One for each data transfer. Generally port 20 is used for data connections. The data connection with ACTIVE FTP mode is one less than the Server port number (Here that would be:  $21-1=20$ ). The data connection stays up for the duration of a single file transfer. Many files may be transferred in the same session.

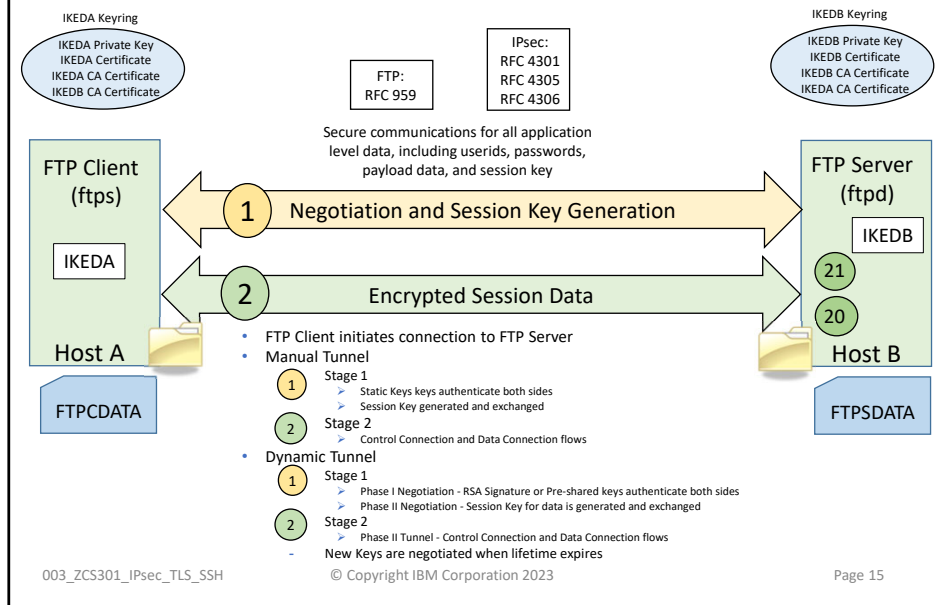
This diagram shows an ACTIVE FTP Connection with the default FTP Server Ports. These ports can be specified differently in the FTP Server Startup procedure or the daemon startup.

There are also PASSIVE FTP connections that are often used for security purposes.

- An FTP client can override the default data port by directing the server to run in passive mode. In passive mode, the server uses an ephemeral port for the data port. Passive mode is requested by firewall friendly clients and by clients initiating three-way data transfers.

FTPD on System z has exploited many security enhancements described in RFCs; it has also implemented RFC 4217 which integrates SSL/TLS protocols into FTP to provide secure connections between server and client.

## Secure FTP with IPsec



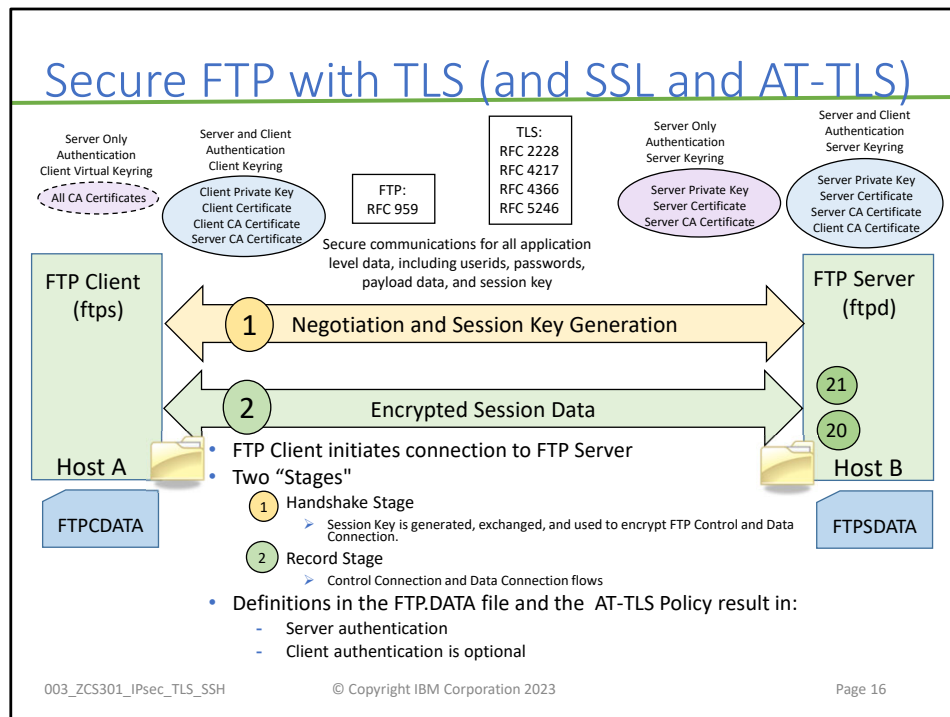
FTP Client initiates connection to FTP Server.

Matching IPsec Policy results in:

- Manual Tunnel static keys authenticate both sides.
- Session Key is generated, exchanged, and used to encrypt data
- or Dynamic Tunnel RSA Signature or Pre-shared keys authenticate both sides.
- Session Key is negotiated, exchanged, and used to encrypt data

When implementing Dynamic IPsec, new session Keys are negotiated when lifetime expires

FTP Control Connection and Data Connection flow across VPN



This diagram is a general representation of connections using SSL, TLS, or AT-TLS to secure the traffic between an FTP client and server. The FTP Client initiates a connection to the FTP Server.

Two "Layers" or phases occur for such a secured transfer.

- Handshake Layer (negotiation of keys, other security parameters)
  - Session Key is generated, exchanged, and used to encrypt FTP Control and Data Connection.
- Record Layer (actual transfer of data payload)

If using AT-TLS, the AT-TLS Policy together with the FTP.DATA file results in:

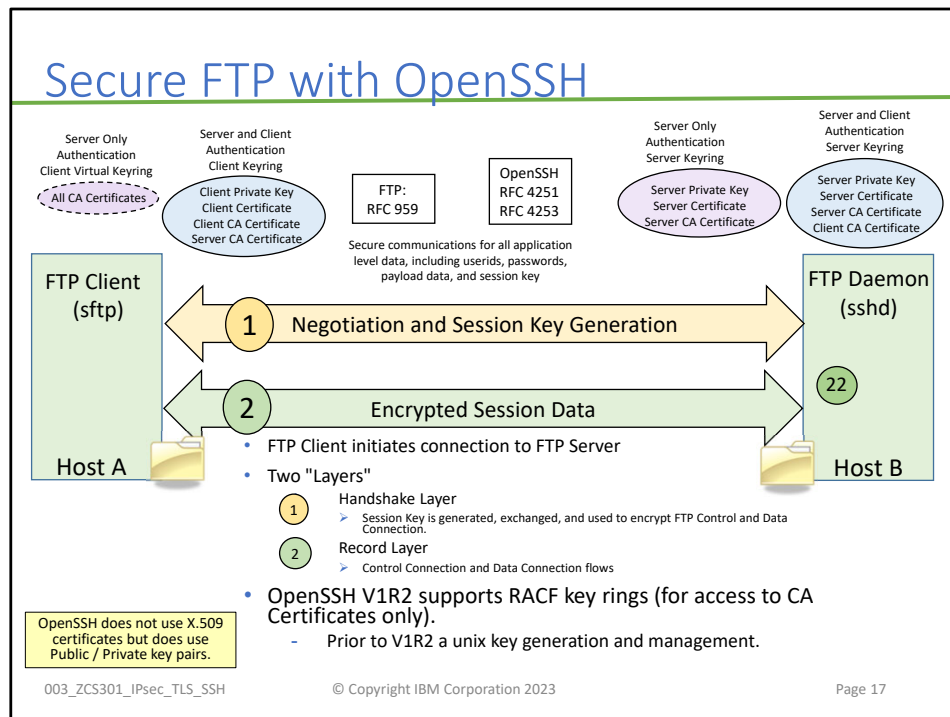
- Server authentication
- Client authentication is optional

If Client authentication is not required, the client end of the connection can identify a Virtual RACF keyring.

If using a virtual keyring, the Server CA is found as a trusted CA certificate in the RACF DIGTCERT class of the Client end-entity.

If not using Client authentication, the Client CA certificate is not necessary on the Server Ring.

If Client authentication is required by the Server, the Client Keyring contains the server CA certificate, the Client CA certificate, the Client Certificate, and the Client Private Key.



Just as with FTP using RFC 959 and either TLS or IPsec to protect the transmissions of control data and data payload, OpenSSH FTP sftp command in UNIX also provides: secure communications for the userid, password, data, and session key. It stores asymmetric keys in key files at the SSH client and server sides of a connection and then negotiates the session key securely. Natively OpenSSH only supports sending/receiving UNIX file systems and is therefore popular on pure UNIX platforms. When it comes to transferring data from an MVS dataset or dataset member, the data must be moved into the UNIX file system before "sftp" can work with it.

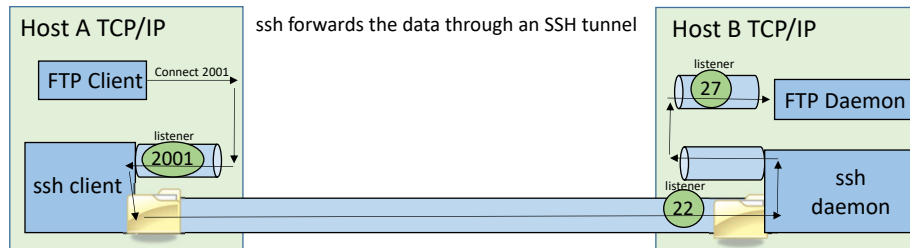
This can be accomplished manually or handled with REXX or SHELL scripts in UNIX or in MVS.

Some OEM vendor platforms have implementations that can automatically interface with UNIX or MVS file systems.

Prior to V1R2 OpenSSH keys and key rings were only supported in unix files:

- ssh-keygen
  - creates public/private key pairs
- ssh-agent
  - holds private keys in memory, saving you from retyping your passphrase
- ssh-add
  - loads private keys into the agent
- ssh-keyscan
  - gathers SSH public host keys

## OpenSSH with TCP/IP Port Forwarding



- Encrypts Userid and Password and Data in communication flows
  - Through Port Forwarding or "Tunneling"
- Scenario:
  - Application Client at Port 2001 on Host A
  - Application Server at Port 27 on Host B
  - SSH is configured to support Port Forwarding
  - The SSH client forwards the data through an SSH tunnel to the SSH daemon
  - The SSH daemon delivers the data to the server at Port 27

003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

Page 18

OpenSSH may be configured to implement TCP/IP Port Forwarding. This would permit any TCP application, including FTP (RFC 959) to flow over a secured SSH channel as depicted in the visual.

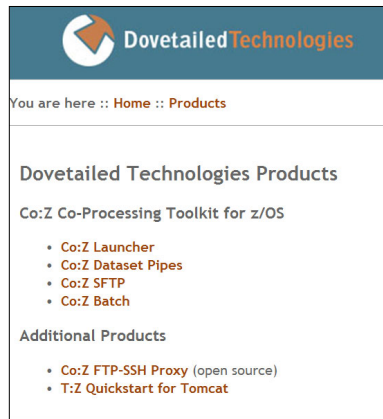
This implementation of FTP is different from "sftp".

### SCENARIO DESCRIPTION:

- Encrypts Userid and Password and Data in communication flows
- An Application Client (like FTP or other application) at Port 2001 on Host A wants to communicate with the Application Server at Port 27 on Host B.
- SSH is configured to support Port Forwarding.
- The Application Client request connects through an SSH client to an SSH daemon.
- The SSH client forwards the data through an SSH tunnel to the "sshd" daemon.
- The SSH daemon delivers the data to the server at Port 27.

## Dovetail Technologies and Co:Z FTP

<http://dovetail.com>



- The Co:Z Co-Processing Toolkit for z/OS includes Co:Z SFTP - a port of the OpenSSH (v5.0p1) sftp-server subsystem and sftp command (renamed as cozsftp). Extensive enhancements have been made to support z/OS facilities such as MVS datasets and spool files.

003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

Page 19

Dovetail Co:Z is an OpenSSH implementation that supports MVS files as well as Unix files.

## z/OS OpenSSH

- z/OS OpenSSH is part of z/OS base in Unix System Services
  - Starting in z/OS V2.2
  - z/OS OpenSSH User's Guide, SC27-6806
- OpenSSH was part of IBM Ported Tools for z/OS
  - Prior to z/OS V2.2
  - IBM Ported Tools for z/OS User's Guide (SA22-7985-06)
  - OpenSSH User's Guide (SA23-2246-02)
- The Internet Engineering Task Force (<http://www.ietf.org/>)
- Four main SECSH internet drafts are:
  - SSH Transport Layer Protocol
    - draft-ietf-secsh-transport-17.txt
  - SSH Authentication Protocol
    - draft-ietf-secsh-userauth-20.txt
  - SSH Protocol Architecture
    - draft-ietf-secsh-architecture-15.5.txt
  - SSH File Transfer Protocol
    - draft-ietf-secsh-filexfer-05.txt

IBM Ported Tools for z/OS are not part of z/OS Communications Server.

## Managed FTP Applications

- Many alternatives for transferring files over the internet:
  - FTP without Encryption
  - FTP with TLS implemented in the application
  - FTP with AT-TLS
  - FTP over IPSec Tunnel
  - FTP over OpenSSH
  - FTP over OpenSSH with Port Forwarding
- Some of these options have manually triggered recovery/restart capabilities
  - Others (OpenSSH) do not
- Managed File Transfer Applications
  - Provide automatic recovery/restart
  - Examples:
    - Tivoli Storage Manager (formerly known as ADSM)
    - Sterling DB2 Connect and Connect Direct (formerly Network Data Mover - NDM)
      - Connect Direct can call System SSL directly and can offload System SSL and some compression operations to zIIP
    - MQ Series File Transfer Enhanced (MQ FTE)
    - and many more

003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

Page 21

There are many ways to move files across a network. You have seen several of them in this presentation. However, not all of these ways serve all needs.

- Although FTP (RFC959) provides recovery and restart capabilities, these must be triggered through operator intervention (or a batch job).
  - This cannot operate against VSAM files, although it can issue FTP queries, whose results are returned in flat-file format.
- Although OpenSSH provides security to userids, passwords, and data flows, it does not provide any type of recovery and restart.
  - It also operates only against UNIX file systems. (Although there are OEM vendor implementations that use methods to bypass this restriction.)

Many installations use both FTP (RFC 959) and OpenSSH sftp.

But other installations find that they need the automated ability to manage File transfers and this is where a "Managed FTP Application" fits in.

- There are many examples of these, of which we list only a few.

MQ FTE is gaining in popularity due to the prevalence of MQ Series applications at many customer sites.

- It operates against all file types
- It has built-in and automated recovery and restart.
- It also can implement SSL/TLS.

## MQ File Transfer Edition (FTE)

- FMID 5655-U80
- Popular Managed FTP product
- Provides Reliable Transfer
  - Binary
  - DB2
  - Microsoft Products
  - MVS files
  - Open Database Connectivity (ODBC)
  - Oracle Informix
  - SQL
  - Sybase
  - Text
  - Unix files
  - VSAM
- and Optional Security with TLS

003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

Page 22

IBM United States Software Announcement 209-119, dated April 14, 2009

- IBM WebSphere MQ File Transfer Edition for z/OS V7.0.1

Asynchronous

- Not all resources must be available at the same time

Strong Security

- SSL Channels provide Authentication, Encryption and optionally digital signature of messages
- All transfers can be audited providing for non-repudiation
- Auditable and Manageable Transfers

Centralized Monitoring and Management

- Logging of data movements, visibility and reporting capabilities

Increased Data Integrity

- Assured once and only once delivery of files (no partial files, no duplicate files)
- Comprehensive platform coverage, assured code page conversion
- No loss of data, no corruption of data
- Files are moved in a transactional manner

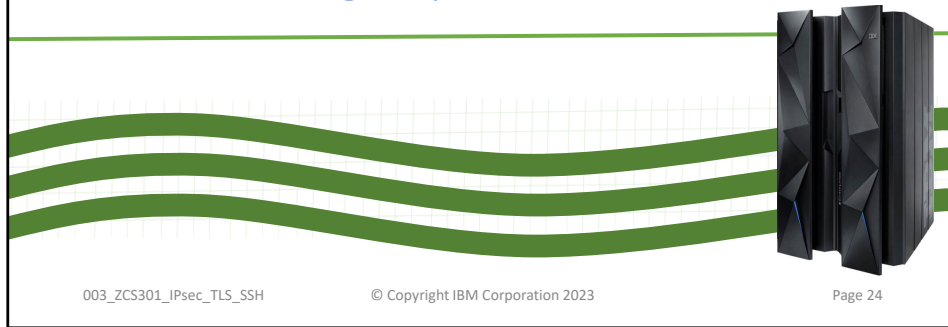
High Performance Transfers

- High-performance, bi-directional concurrent transfer capabilities
- Impervious to Network Failures
- High volume data transfer infrastructure (no file or database size limitations)

## Summary

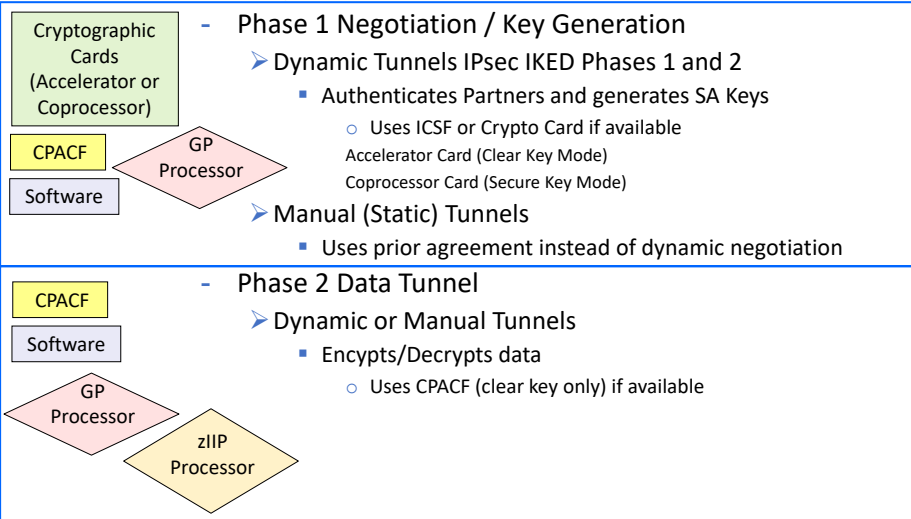
Support	IPsec	TLS	OpenSSH
IETF Standard	Yes	Yes	Yes
MVS Datasets	Yes	Yes	No
Unix Files	Yes	Yes	Yes
Server Authentication	Required	Required	Required
Client Authentication	Required	Optional	Optional
System SSL	Yes	Yes	OpenSSL instead
CPACF	Yes	Yes	Yes
Crypto Cards	Authentication Only	Authentication Only	Random Number Generation (RNG) Only
TCP Protocol	Yes	Yes	Yes
UDP Protocol, etc.	Yes	No	No
FIPS 140	Yes	Yes	Yes

## Comparison of Offload and Cryptographic Hardware Usage by TLS, IPsec, SSH



## IPsec

- Two Stages



003\_ZCS301\_IPsec\_TLS\_SSH

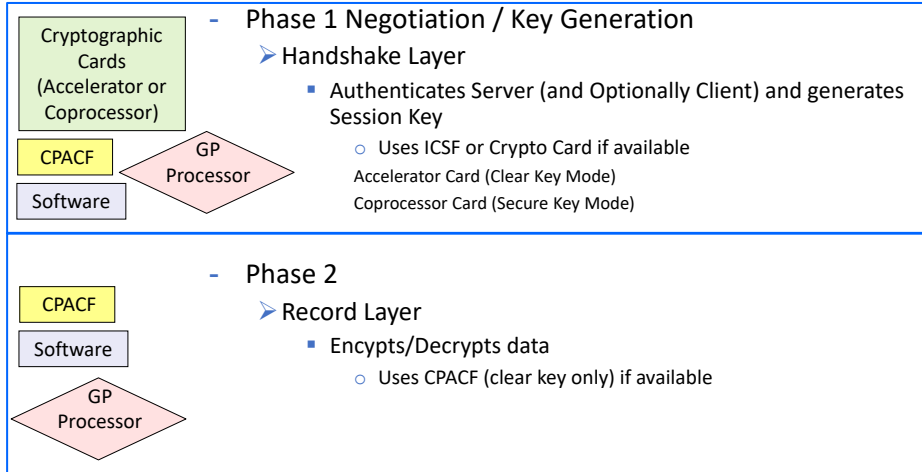
© Copyright IBM Corporation 2023

Page 25

Link to document on zIIP and IPsec is at <http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100988>

## AT-TLS

- Two Stages



## OpenSSH Use of Cryptographic Hardware

Cipher	Available Hardware	
	CPACF only	CPACF & Coprocessor
RNG (Random Number Generation)	In software	In Coprocessor
RSA	In software	In software
DSA	In software	In software

- RSA, DSA
  - for Authentication of peer
  - for Generation of Session Key and Digital Signature
- RNG
  - OpenSSH on z accesses the hardware Coprocessor only during the Random Number Generation (RNG) that is used in the process of generating the symmetric key which will be used during the data transfer stage of SSH.

003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

Page 27

RSA is the default. Choose DSA only when you have a specific need for DSA.

Diffie-Hellman key exchange (D-H) is a cryptographic protocol that allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher.

Prior to OpenSSH V1.2 Perform setup for server authentication using unix commands:

- Generate host keys for server
  - allows a client to verify the identity of the server
- Use ssh-keygen to create host keys:
  - `ssh-keygen -t dsa -f /etc/ssh/ssh_host_dsa_key -N ""`
  - `ssh-keygen -t rsa -f /etc/ssh/ssh_host_rsa_key -N ""`
- Create local and remote ssh\_known\_hosts files
  - Contains host public keys for all hosts you know about
- Copy local host's public keys to the remote hosts
- Gather public keys of remote hosts

OpenSSH for IBM z/OS: Program Product: IBM Ported Tools for z/OS

- unpriced, runs on z/OS 1.4 and higher
- order from ShopzSeries, under "MVS: System Mgmt. and Security" – now part of z/OS
- GA Version info: OpenSSH 3.5p1, OpenSSL 0.9.7b, zlib 1.1.4
- OA10315 version is: OpenSSH 3.8.1p1, OpenSSL 0.9.7d, zlib 1.1.4
- Base SSH: Uses Public Key Infrastructure for authentication and encryption

Authentication (both client and server) through:

- Public key cryptography
- Existing login passwords
- Trusted hosts authentication

Data Privacy - through encryption

- Data Integrity - guarantees data traveling over the network is unaltered
- Authorization – regulates access control to accounts
- Forwarding (a.k.a. tunneling) – encryption of other TCP/IP-based sessions

Prior to OpenSSH V1.2 key management and distribution for large numbers of users are difficult because there is no concept of a Certificate Authority. As a result, the keys themselves or the trusted hosts file itself for each server needs to be distributed to the participants.

Only for UNIX files with SFTP; only for UNIX shell with SSH (Tectia and coZ extensions allow usage on MVS files); only uses crypto card for generation of the keys.

# End of Topic



003\_ZCS301\_IPsec\_TLS\_SSH

© Copyright IBM Corporation 2023

Page 28