

SHARE
Technology • Connections • Results

Safe and Secure Transfers with z/OS FTP

Alfred B Christensen – alfredch@us.ibm.com
IBM Raleigh, NC, USA

Session: 8239
Thursday, March 3, 2011: 9:30 AM-10:30 AM



Safe and Secure Transfers with z/OS FTP

Session number:	8239
Date and time:	Thursday, March 3, 2011: 9:30 AM-10:30 AM
Location:	Room 212B (Anaheim Convention Center)
Program:	Communications Infrastructure
Project:	Communications Server
Track:	Network Support and Management, Security Administration and Security and Privacy Best Practices
Classification:	Technical
Speaker:	Alfred B Christensen, IBM
Abstract:	<p>FTP is a readily available, convenient, and inexpensive technology to transfers files and data sets between z/OS and a virtually unlimited number of other operating system platforms. FTP is not a bad technology, as some recent press might lead you to believe. FTP can be misused and cause problems if the FTP service isn't properly set up to prevent potential security exposures. This session will explore a wide range of aspects related to how FTP works on z/OS. The session will reveal 'hidden gems' of FTP on z/OS and will look at a set of usage scenarios, providing suggestions on how to best exploit selected features of the z/OS FTP technology. The session will especially focus on how you can secure both the FTP environment itself and the individual data transfers that z/OS FTP participates in both as a client and as a server.</p>

Trademarks, notices, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries or both:

- | | | | | |
|--|---|---|--|--|
| <ul style="list-style-type: none"> • Advanced Peer-to-Peer Networking® • AIX® • alphaWorks® • AnyNet® • AS/400® • BladeCenter® • Candle® • CICS® • DataPower® • DB2 Connect • DB2® • DRDA® • e-business on demand® • e-business (logo) • e business (logo)® • ESCON® • FICON® | <ul style="list-style-type: none"> • GDDM® • GDPS® • Geographically Dispersed Parallel Sysplex • HiperSockets • HPR Channel Connectivity • HyperSwap • i5/OS (logo) • i5/OS® • IBM eServer • IBM (logo)® • IBM® • IBM zEnterprise™ System • IMS • InfiniBand® • IP PrintWay • IPDS • iSeries • LANDP® | <ul style="list-style-type: none"> • Language Environment® • MQSeries® • MVS • NetView® • OMEGAMON® • Open Power • OpenPower • Operating System/2® • Operating System/400® • OS/2® • OS/390® • OS/400® • Parallel Sysplex® • POWER® • POWER7® • PowerVM • PR/SM • pSeries® • RACF® | <ul style="list-style-type: none"> • Rational Suite® • Rational® • Redbooks • Redbooks (logo) • Sysplex Timer® • System i5 • System p5 • System x® • System z® • System z9® • System z10 • Tivoli (logo)® • Tivoli® • VTAM® • WebSphere® • xSeries® • z9® • z10 BC • z10 EC | <ul style="list-style-type: none"> • zEnterprise • zSeries® • z/Architecture • z/OS® • z/VM® • z/VSE |
|--|---|---|--|--|
- * All other products may be trademarks or registered trademarks of their respective companies.

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries or both:

- Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.
- Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license there from.
- Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
- InfiniBand is a trademark and service mark of the InfiniBand Trade Association.
- Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.
- UNIX is a registered trademark of The Open Group in the United States and other countries.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.
- IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

Notes:

- Performance is in Internal Throughput Rate (ITR) ratio based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput improvements equivalent to the performance ratios stated here.
- IBM hardware products are manufactured from new parts, or new and serviceable used parts. Regardless, our warranty terms apply.
- All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.
- This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.
- All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.
- Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
- Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

Refer to www.ibm.com/legal/us for further legal information.

Agenda



- ❑ **FTP and Security – an oxymoron?**
- ❑ **SSL/TLS FTP: Keys and certificates overview**
- ❑ **z/OS FTP Server with server authentication only – client WS_FTP Pro on Windows**
- ❑ **z/OS FTP Server with server authentication and client authentication – client WS_FTP Pro on Windows**
- ❑ **z/OS FTP client connecting to a FileZilla Windows server with server authentication only**
- ❑ **Appendix A: z/OS FTP – local security**
- ❑ **Appendix B: Secure FTP - network traversal challenges and solutions**



Disclaimer: All statements regarding IBM future direction or intent, including current product plans, are subject to change or withdrawal without notice and represent goals and objectives only. All information is provided for informational purposes only, on an “as is” basis, without warranty of any kind.

Safe and Secure Transfers with z/OS FTP

FTP and Security – an oxymoron?



Let's try and clear a little common confusion from the start



RFC959 FTP

▪ FTP:

- Also referred to as RFC959 FTP or “normal” FTP
- The FTP protocol we all know and have used for years.
- The FTP protocol has been extended numerous times since the original RFC 959 was issued in 1985
 - Specific support for both Kerberos-based and SSL/TLS-based security has been added to the FTP protocol
 - RFC4217 "Securing FTP with TLS"
- What the z/OS CS FTP client and server have supported through many years
 - An RFC959 FTP client talks to an RFC959 FTP server, and not to an sftp server

Secure Shell FTP

▪ sftp:

- Secure Shell file transfer protocol
 - A sub-protocol of SSH (Secure Shell)
 - Supported on z/OS by "IBM Ported tools for z/OS" and at least two ISV products
 - Has nothing to do with RFC959 FTP - incompatible protocols
 - An sftp client talks to an sftp server and not an RFC959 FTP server

RFC4217 FTP

▪ FTPS:

- Also referred to as RFC4217 FTP, FTP AUTH-TLS, or FTP AUTH-SSL
- Secure RFC959 FTP using a standard security mechanism, such as Kerberos or SSL/TLS
 - RFC4217 "Securing FTP with TLS"
- The normal FTP protocol but extended with full network security (authentication, data integrity, and data privacy)
- Both control connection and data connection can be secured
 - No user IDs or password flowing in the clear

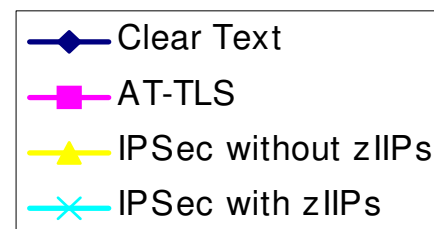
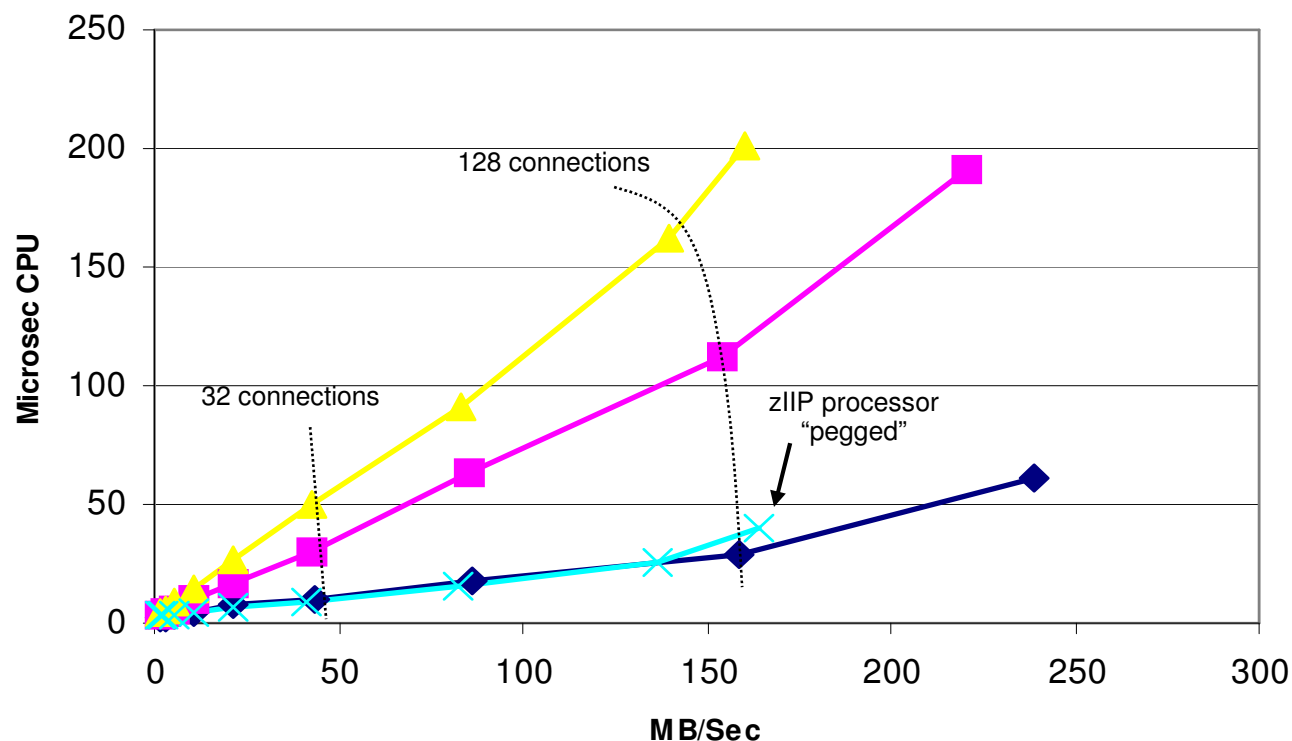
A quick comparison of selected z/OS file transfer technologies from a security perspective

	FTP With no security RFC959	FTPS FTP w. SSL/TLS RFC959 + RFC4217	FTP FTP w. IPSec Any RFC level	SFTP As implemented by IBM Ported Tools
User ID and password protection	No	Yes	Yes	Yes
Data protection (the file being transferred)	No	Yes	Yes	Yes
z/OS UNIX file support	Yes	Yes	Yes	Yes
z/OS MVS data set support	Yes	Yes	Yes	No (but add-on products do exist)
Use of System z hardware encryption technologies	n/a	Yes	Yes	Yes (for random number generation)
Partner authentication via locally stored copies of public keys	n/a	No	Yes (pre-shared key)	Yes
Partner authentication via X509 certificates	n/a	Yes	Yes	No
Use of SAF key rings and/or ICSF	n/a	Yes	Yes	Yes
FIPS 140-2 mode	n/a	Yes (z/OS V1R11)	No	No
Mutual authentication supported	n/a	Yes	Yes (at an IP address level)	Yes

MVS data set support example: Dovetailed Technologies' Co:Z SFTP

FTP Server CPU usage with and without security

FTP CPU Usage



All measurements done with z/OS V1R11
 Outbound Data (Gets) to an MVS client
 3DES encryption with SHA authentication
 From 1 to 128 parallel connections
 Highest throughput numbers obtained with 0 think-time

Client: 1 z10 LPAR (3 dedicated CPs)
 Server: 1 z10 LPAR (4 dedicated CPs)
 Connectivity: OSA-E3 10 GbE
 Encryption/Authentication: 3DES/SHA
 Transaction: 1 byte / 2 MB
 Target data sets: MVS data sets on 3390 DASD
 Think time: 1500 ms
 Number of connections: 1 to 128
 Driver tool: AWM

All performance data contained in this publication was obtained in the specific operating environment and under the conditions described and is presented as an illustration. Performance obtained in other operating environments may vary and customers should conduct their own testing.

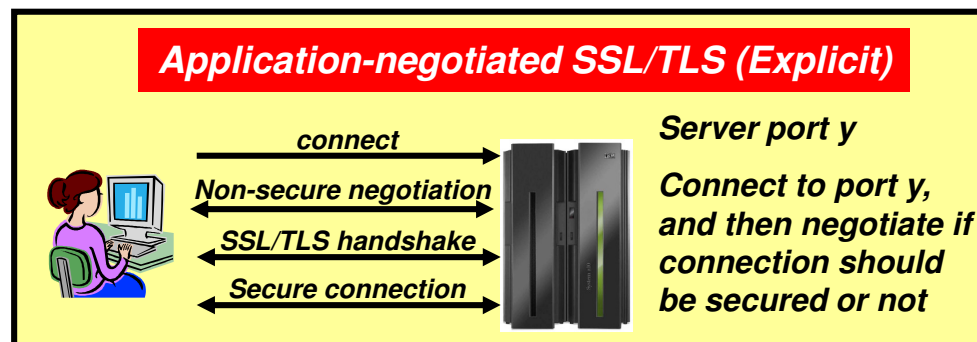
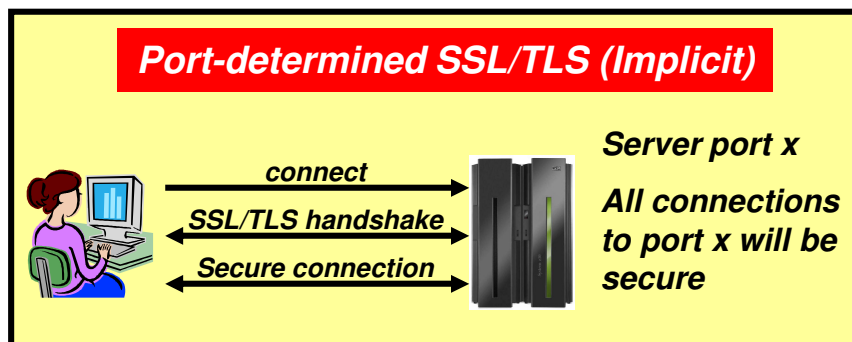
Safe and Secure Transfers with z/OS FTP

SSL/TLS FTP: Keys and certificates overview

Note: This will hurt your brain, but this is where you all run into problems when trying to set up SSL/TLS for the first time!



SSL/TLS application types

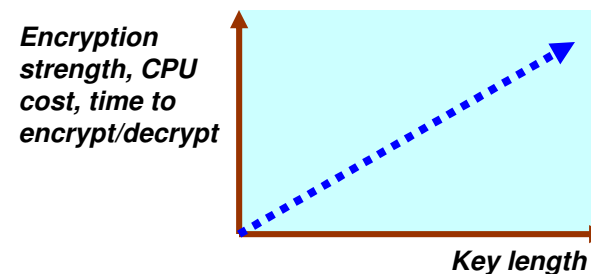


- As soon as a connection has been established with the server, the SSL/TLS handshake starts
- Examples are the HTTPS port (443), and FTP's secure port (990)
- AT-TLS considerations:
 - Can be done totally transparent to application code
 - This is referred to as an AT-TLS "Basic" application
 - Optionally the application may query SSL/TLS attributes, such as client user ID (if client authentication is used, cipher suite in use, etc)
 - This is referred to as an AT-TLS "Aware" application

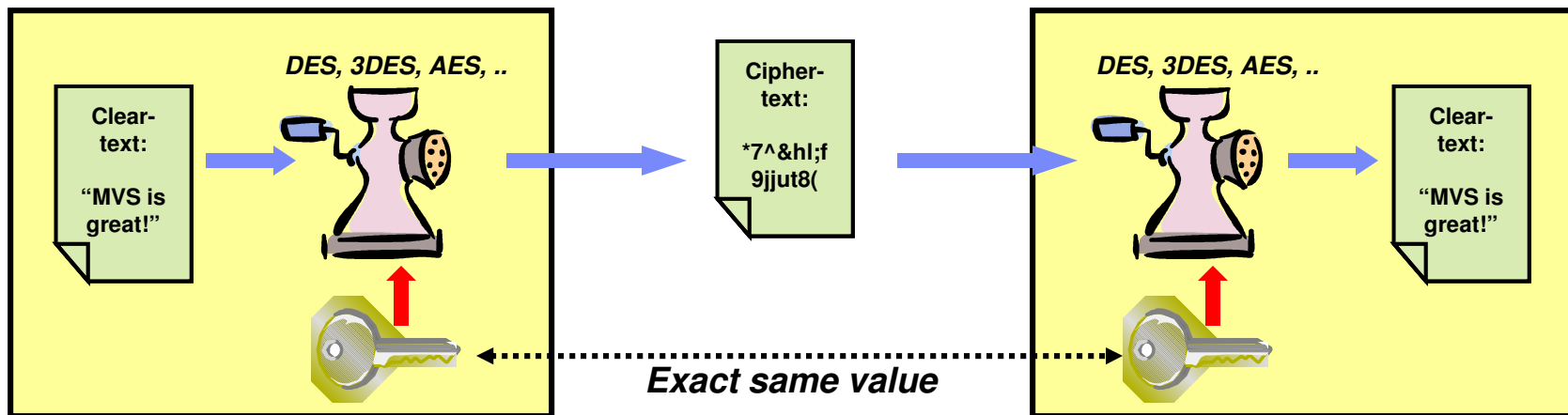
- Application protocol includes verbs to negotiate security protocol and options
- Examples are FTP that uses the AUTH FTP command to negotiate use of SSL/TLS or Kerberos, and in some cases a TN3270 server port (Conntype NegtSecure)
- AT-TLS considerations:
 - Application needs to "tell" AT-TLS when to start the SSL/TLS handshake
 - This is referred to as an AT-TLS "Controlling" application
 - Otherwise, use of AT-TLS is transparent to application
 - Optionally the application may query SSL/TLS attributes, such as client user ID (if client authentication is used, cipher suite in use, etc)

Cryptographic Basics

- Cryptography is the use of mathematical algorithms to transform data for the purposes of ensuring:
 - **Partner authentication** – proving the other end point of the secure communication is who it claims to be (certificates and asymmetric encryption)
 - **Data privacy** – hiding the data (encryption/decryption)
 - **Data integrity** – proving the data hasn't been modified since it was sent (message digests and secure message authentication codes)
 - **Data origin authentication** – proving the data's origin (message digests and secure message authentication codes)
- Cryptographic operations are compute intensive, hence the need for hardware assist technologies
- General rule: For a given algorithm: the longer keys, the stronger security, the more compute intensive
 - For example, AES-128 vs. AES-256
 - Increases the amount of work an attacker needs to do to crack the code

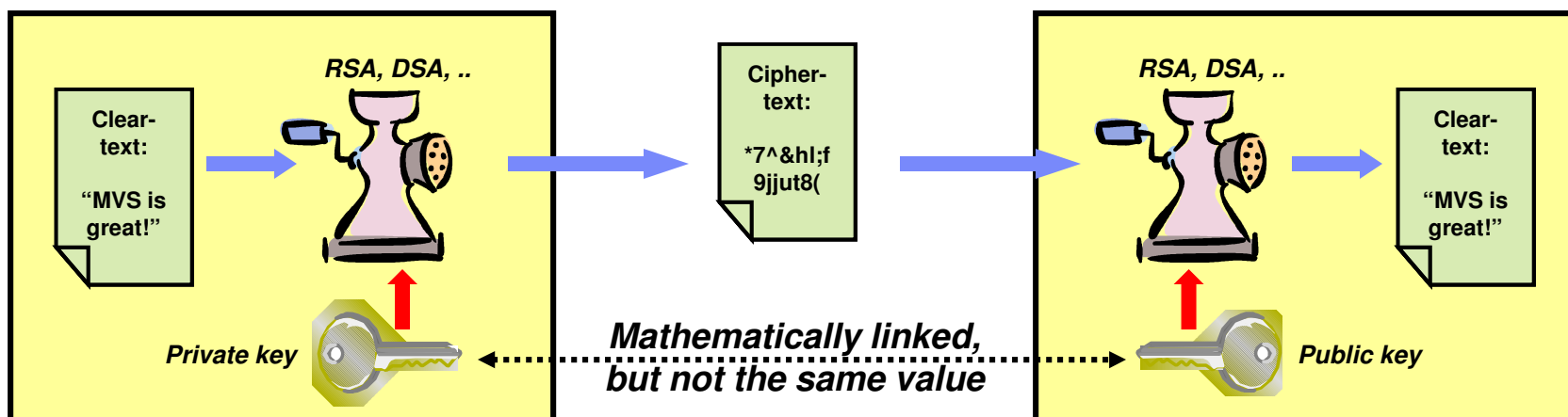


Symmetric encryption

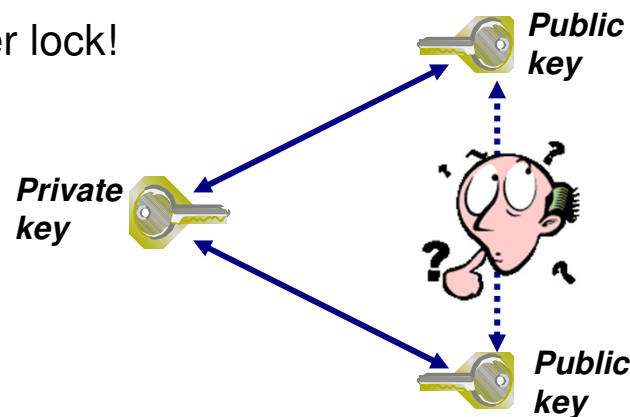


- Only one key value - “shared secret” between both parties
 - Used for both encryption and decryption
 - Hence, the symmetry; each side has the same key and use the same algorithm
- Much faster than asymmetric cryptography
 - You typically use symmetric encryption for bulk encryption/decryption
- Also known as...
 - “secret key encryption”
- Securely sharing and exchanging the key between both parties is a major issue

Asymmetric encryption

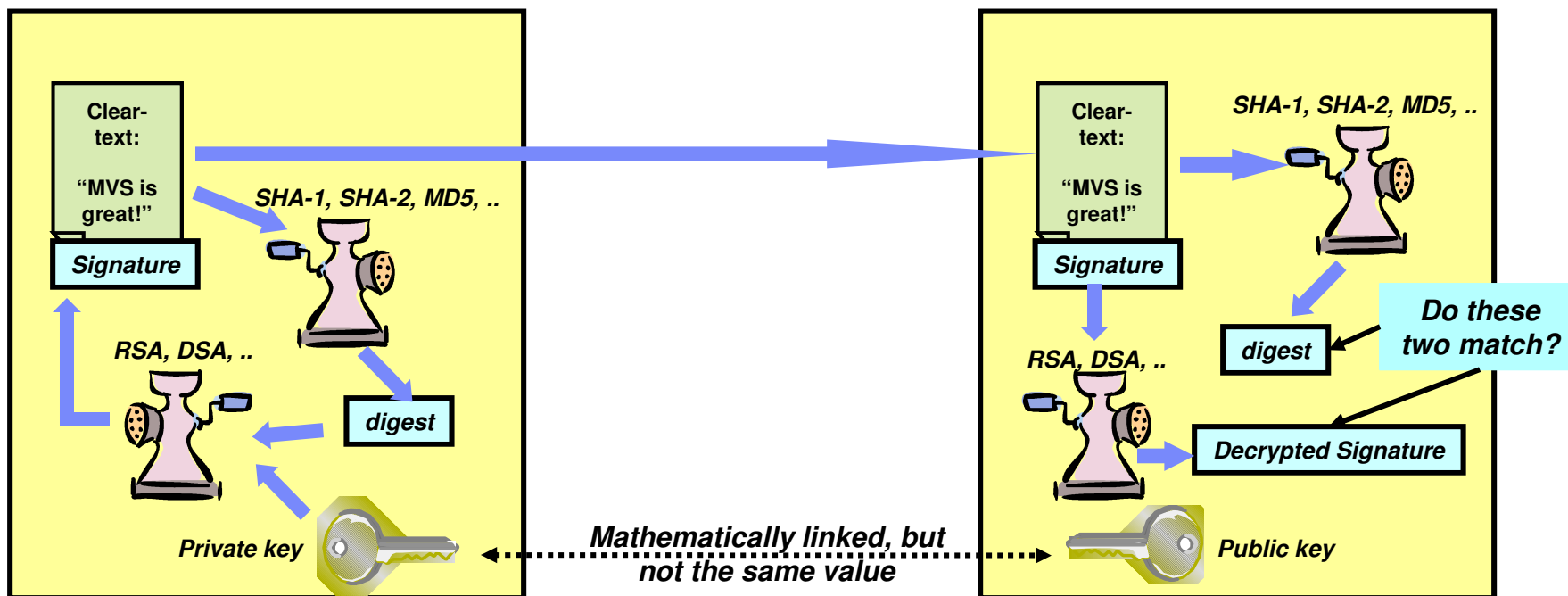


- Two different key values – no shared secrets!
 - Private key is known only to owner and is kept under lock!
 - Public key is freely distributed to others
 - Data encrypted with private key can only be decrypted with public key and vice versa
 - No way to derive one key value from the other
- Great for authentication and non-repudiation
 - “digital signatures” - signing with private key
- Very expensive computationally
 - Not so great for bulk encryption - usually used to encrypt small data objects like message digests or symmetric keys
- Also known as “public key cryptography”



RSA: Rivest, Shamir and Adleman
DSA: Digital Signature Algorithm

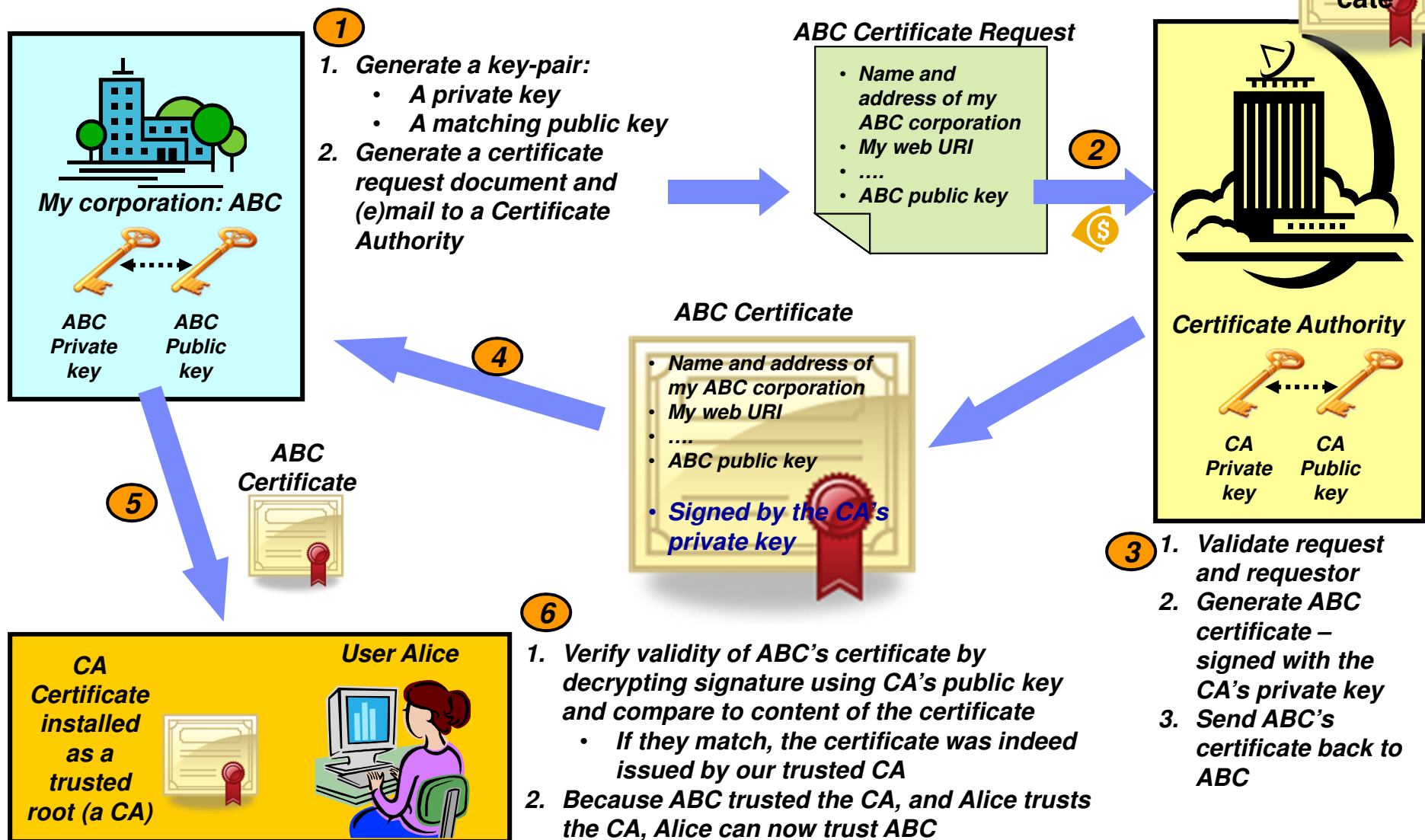
Digital signature



- A digital signature is a message digest that has been encrypted with the sender's private key.
- If the receiver recalculates the message digest, decrypts the signature with the sender's public key, and compares the decrypted signature to the recalculated message digest – the two should match:
 - The message text cannot have been modified since the signature was calculated
 - The signature cannot have been tampered with
 - The signature could only have been created by the partner with the matching private key

SHA: Secure Hash Algorithm – **MDn:** Message Digest n

Trust relationships via Certificate Authorities – getting my public key distributed to those who need it





SSL/TLS use of hardware crypto functions

Crypto Type	Algorithm	CPACF available only	CPACF plus Coprocessor / Accelerator available
<i>Asymmetric encrypt / decrypt</i>	RSA signature generation	In software	In coprocessor mode only. Otherwise in software (Accelerator does not support this option)
	RSA signature verification	In software	In coprocessor / accelerator
	PKA encrypt / decrypt for handshake	In software	In coprocessor / accelerator
<i>Symmetric encrypt / decrypt</i>	DES encrypt / decrypt	CPACF (non-FIPS mode only; DES not allowed in FIPS mode)	
	3DES encrypt / decrypt	CPACF	
	AES-CBC-128 encrypt / decrypt	CPACF	
	AES-CBC-256 encrypt / decrypt	In software on z9, in CPACF on z10	
<i>Symmetric authentication</i>	SHA-1 digest generation	CPACF	
	SHA-224 digest generation	CPACF	
	SHA-256 digest generation	CPACF	
	SHA-384 digest generation	In software on z9, in CPACF on z10	
	SHA-512 digest generation	In software on z9, in CPACF on z10	
	MD5	In software (non-FIPS mode only; MD5 not allowed in FIPS mode)	

Hardware support

- With AT-TLS enabled, check the TCP/IP stack SYSOUT file for details on which cryptographic algorithms are supported by your hardware:

```
System SSL: SHA-1 crypto assist is available
System SSL: SHA-224 crypto assist is available
System SSL: SHA-256 crypto assist is available
System SSL: SHA-384 crypto assist is available
System SSL: SHA-512 crypto assist is available
System SSL: DES crypto assist is available
System SSL: DES3 crypto assist is available
System SSL: AES 128-bit crypto assist is available
System SSL: AES 256-bit crypto assist is available
System SSL: ICSEF services are not available
```

Safe and Secure Transfers with z/OS FTP

z/OS FTP Server with server authentication only – client WS_FTP Pro on Windows

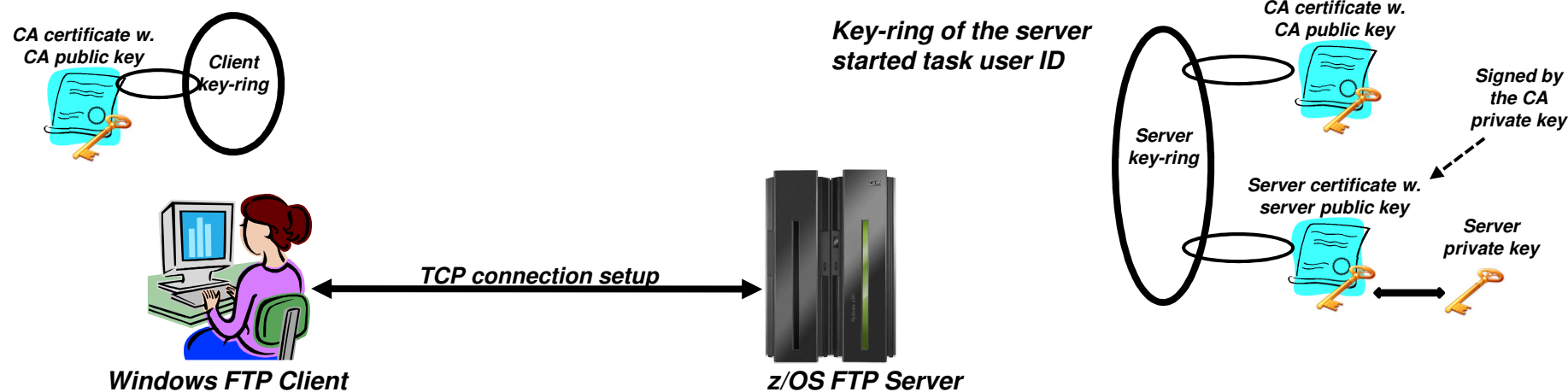
WS_FTP Professional is a product from Ipswitch File Transfer Division:

<http://www.ipswitchft.com/products/ws ftp pro/index.aspx>

This material does not in any way endorse or promote WS_FTP Professional, but merely uses it as an example of a Windows FTP client that supports SSL/TLS FTP functions.



What is needed for z/OS Server authentication only (which is sufficient for encrypted data exchange)



1. Verify server certificate has not expired
2. Verify server certificate is valid using CA's public key
3. Do optional checks on the server certificate
4. Store server's public key for later use
5. Generate symmetric key and encrypt under server's public key

TCP connection setup
Hello – I want to use SSL/TLS
Hello – OK, me too !!
And here is my server certificate
Server certificate w. server public key
Here is our secret symmetric key
Encrypted under your public key

- **CA may be an external CA, such as Verisign, or it may be an in-house CA**
 - In both cases, the CA root certificate needs to be present at both the client and the server side
- **The server certificate is signed by the CA and is stored on the server side**
 - On z/OS, this will typically be the default certificate in the server's started task user ID's key-ring in RACF
- **During SSL handshake, the server certificate (not the server private key) is sent to the client**
 - The client verifies the certificates signature using the CA public key in its copy of the CA certificate

Create self-signed root certificate for test purposes

```

RACDCERT CERTAUTH GENCERT +
  SUBJECTSDN( +
    CN('MVS098 Certificate Authority') +
    OU('Z/OS CS V1R9', 'ENS', 'AIM', 'SWG') +
    O('IBM') +
    L('Raleigh') +
    SP('NC') +
    C('US') ) +
  SIZE(1024) +
  NOTBEFORE(DATE(2010-02-01)) +
  NOTAFTER(DATE(2020-12-31)) + ←
  WITHLABEL('ABCTLS CA') +
  KEYUSAGE(CERTSIGN) +
  ALTNAME( +
    DOMAIN('mvs098.tcp.raleigh.ibm.com') )

```

Create a self-signed root certificate and a private/public key-pair:

- **CERTAUTH**
- **KEYUSAGE(CERTSIGN)**
- **Absence of a SIGNWITH option**

It can become a nightmare when these things expire, so don't create certificates with too short a time span! (Your security czar will likely have an opinion on that)

- In a production environment, you would not need a self-signed root certificate. To sign server and personal certificates, you would use your company root certificate or an external Certificate Authority.
- For testing, a self-signed root certificate is useful. It allows you to familiarize yourself with keys and certificates and allows you to thoroughly test your secure FTP setup on z/OS before deploying it in production.

Create z/OS FTP server certificate signed with your own root certificate

```
RACDCERT ID(TCPCS) GENCERT +
  SUBJECTSDN( +
    CN('MVS098 Server Certificate') +
    OU('Z/OS CS V1R11', 'ENS', 'AIM', 'SWG') +
    O('IBM') +
    L('Raleigh') +
    SP('NC') +
    C('US') ) +
  SIZE(1024) +
  NOTBEFORE(DATE(2010-02-01)) +
  NOTAFTER(DATE(2020-12-31)) +
  WITHLABEL('ABCTLS TCPSERV') +
  KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN) +
  ALTNAME( +
    DOMAIN('mvs098.tcp.raleigh.ibm.com') ) +
  SIGNWITH(CERTAUTH LABEL('ABCTLS CA'))
```

Create a server certificate signed with your own root certificate and a private/public key pair:

- *ID(userID) – the started task user ID of your FTP server*
- *KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)*
- *SIGNWITH(CERTAUTH LABEL('your root certificate'))*

- In a production environment, you would use an alternative procedure after having generated the server key pair and certificate:
 - You would generate a certificate signing request and send it to your CA
 - Your CA would process your request and create a certificate signed with the CA private key
 - You would import the signed certificate into RACF

Alternative: use an external CA to sign your server certificate

```

RACDCERT ID(TCPCS) GENCERT +
  SUBJECTSDN( +
    CN('MVS098 Server Certificate') +
    OU('Z/OS CS V1R11', 'ENS', 'AIM', 'SWG') +
    O('IBM') +
    L('Raleigh') +
    SP('NC') +
    C('US') ) +
  SIZE(1024) +
  NOTBEFORE(DATE(2010-02-01)) +
  NOTAFTER(DATE(2020-12-31)) +
  WITHLABEL('ABCTLS TCPSERV') +
  KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN) +
  ALTNAME( +
    DOMAIN('mvs098.tcp.raleigh.ibm.com') )
RACDCERT ID(TCPCS) GENREQ (LABEL('ABCTLS TCPSERV')) +
  DSN('USER1.PKITEST.SERVERS.REQ')

(**** delay here while CA processes your request ****)

RACDCERT ID(TCPCS) +
  ADD('USER1.PKITEST.SERVERS.CRT') +
  TRUST +
  WITHLABEL('ABCTLS TCPSERV')
  
```

Create a server certificate and a private/public key pair:

- *ID(userID) – the started task user ID of your FTP server*
- *KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)*

Generate a request to have the certificate signed by an external CA

- *Send the request to the CA*
- *Receive the response from the CA*

Add the signed certificate into RACF

If not already there, you also need to add the CA's root certificate to RACF as a CERTAUTH certificate

Create you z/OS server started task user ID key-ring and connect required certificates to it

```

RACDCERT CERTAUTH +
  EXPORT (LABEL ('ABCTLS CA')) +
  DSN ('USER1.ABCTLSCA.B64') +
  FORMAT (CERTB64)
RACDCERT ID (TCPCS) ADDRING (TLSRING)
RACDCERT ID (TCPCS) +
  CONNECT (CERTAUTH LABEL ('ABCTLS CA') +
  RING (TLSRING) )
RACDCERT ID (TCPCS) +
  CONNECT (LABEL ('ABCTLS TCPSERV') +
  RING (TLSRING) +
  DEFAULT)
RACDCERT ID (TCPCS) +
  LISTRING (TLSRING)
  
```

In order for the remote client to successfully authenticate server certificates that are signed with our self-signed root certificate, they need a copy of that root certificate in their local key-rings. Download as a text file to your client workstation

Create key-ring for your started task server user ID

Connect certificates to the key-ring:

- Your root certificate
- Your server certificate

Digital ring information for user TCPCS:

Ring:

>TLSRING<

Certificate Label Name	Cert Owner	USAGE	DEFAULT
ABCTLS CA	CERTAUTH	CERTAUTH	NO
ABCTLS TCPSERV	ID (TCPCS)	PERSONAL	YES

Configure you z/OS FTP server to use SSL/TLS – this example is based on AT-TLS

- Define an FTP server that supports SSL/TLS connections, but does not require it
 - It depends on the client sending an AUTH command or not
- SSL/TLS is done by ATTLS

EXTENSIONS	AUTH_TLS	; Enable TLS authentication
TLSMECHANISM	ATTLS	; Server-specific or ATTLS
SECURE_FTP	ALLOWED	; Security required/optional
SECURE_LOGIN	NO_CLIENT_AUTH	; Client authentication
SECURE_PASSWORD	REQUIRED	; Password requirement
SECURE_CTRLCONN	PRIVATE	; Minimum level of security CTRL
SECURE_DATACONN	PRIVATE	; Minimum level of security DATA
TLSRFCLEVEL	RFC4217	; SSL/TLS RFC Level supported

AT-TLS setup: Server port and keyring definitions

Modify Rule (Background Dialog)

AT-TLS rule name
 Rule name: * ABC_FTP_4021 Enable rule

Specify settings
 Traffic | Role | Key Ring | Data Endpoints | Security Level | Advanced

Use this panel to specify the traffic settings.
 Application name: * ABC_FTP4021_New

Local port
 All ports
 All ephemeral ports
 Ports: * 4021
Separate multiple ports with a co...

Remote port
 All ports
 All ephemeral ports
 Ports: *
Separate multiple po...

Indicate the TCP connect direction
 Either Inbound only Outbound only

Specify jobname and user ID
 Jobname: U

OK

Modify Rule (Foreground Dialog)

AT-TLS rule name
 Rule name: * ABC_FTP_4021 Enable rule

Specify settings
 Traffic | Role | Key Ring | Data Endpoints | Security Level | Advanced

Use this panel to specify the key ring database and certificate label to use for this rule.

Key ring database
 Use the key ring database defined for the z/OS image
 Use a Simple name (as in an SAF product or in PKCS #11 Token format):
 Key ring: * TLSRING

Use this z/OS UNIX file system key database:
 Key database: *
 Key database stash file: * or
 Key database password: *

Certificate label:

OK Cancel Help ?

AT-TLS setup: Data endpoints

Modify Rule
✕

AT-TLS rule name

Rule name: * Enable rule

Specify settings

Traffic | Role | Key Ring | **Data Endpoints** | Security Level | Advanced

Select the address groups of the host endpoints of the traffic you want to protect.

Local data endpoint

Address group

▾

IPv4 or IPv6 address, subnet or range

*

Examples: x.x.x.x, x.x.x.x/yy, x.x.x.x-y.y.y.y
x:xx, x:xx/yyy, x:xx-yyz

Remote data endpoint

Address group

▾

IPv4 or IPv6 address, subnet or range

*

Examples: x.x.x.x, x.x.x.x/yy, x.x.x.x-y.y.y.y
x:xx, x:xx/yyy, x:xx-yyz

?

AT-TLS setup: Security level

- **Type:**
 - AT-TLS
- **Encryption:**
 - 0x35 - TLS_RSA_WITH_AES_256_CBC_SHA (first choice)
- **Use TLS Version 1.0:**
 - Yes
- **Use TLS Version 1.1:**
 - Yes
- **Use SSL Version 3:**
 - Yes
- **Use SSL Version 2:**
 - No
- **Client authentication:**
 - None
- **FIPS 140 Support:**
 - Off



Adding your root certificate to WS_FTP Pro's trusted authorities

Select tools,
then
options, and
then import

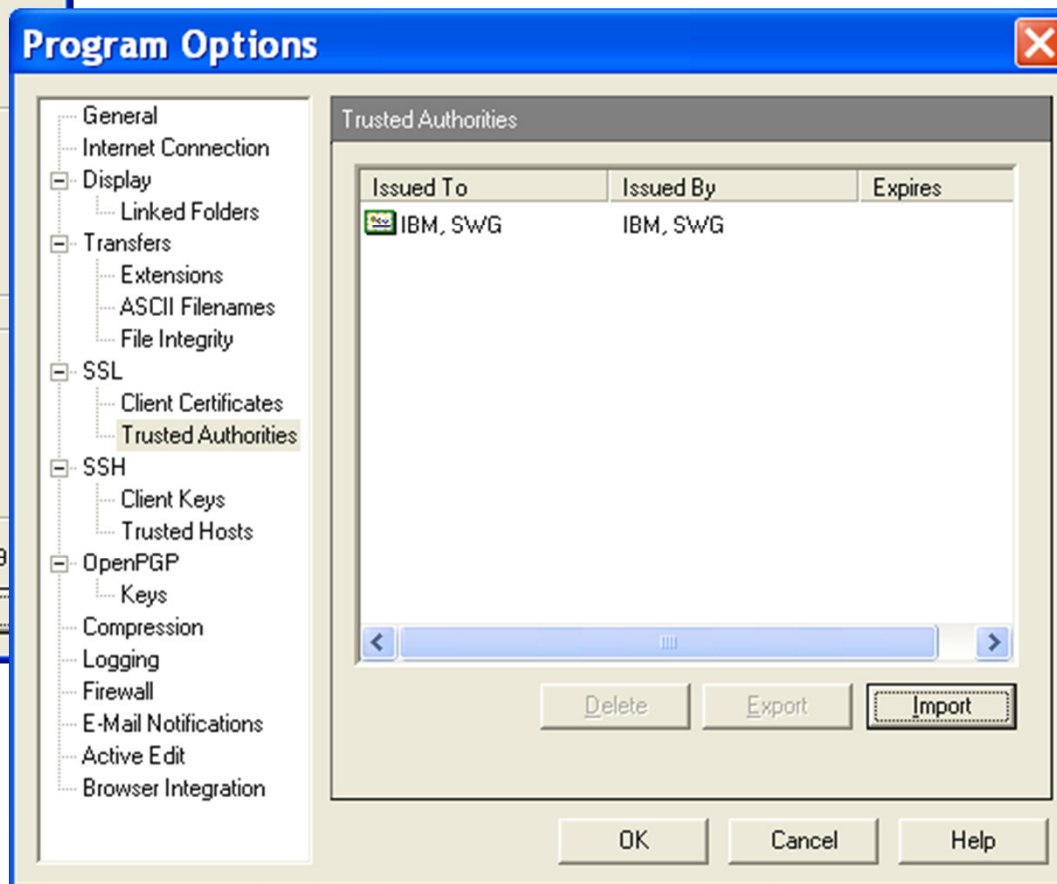
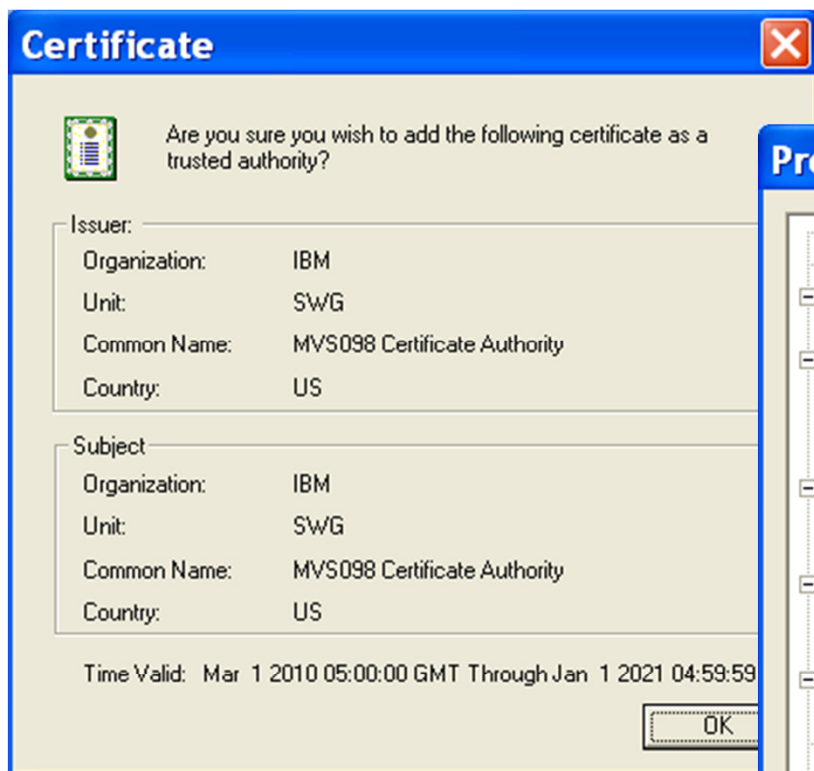
The screenshot shows the 'Program Options' dialog box in Ipswitch WS_FTP Professional. The 'Trusted Authorities' section under 'SSL' is selected. The dialog box contains a table for 'Trusted Authorities' with the following columns: 'Issued To', 'Issued By', and 'Expires'. Below the table are buttons for 'Delete', 'Export', and 'Import'. The main window shows a file explorer view of 'c:\ABC_Work\SSL' with files like 'ABC_Cert.p12'.

Issued To	Issued By	Expires

Adding your root certificate to WS_FTP Pro's trusted authorities

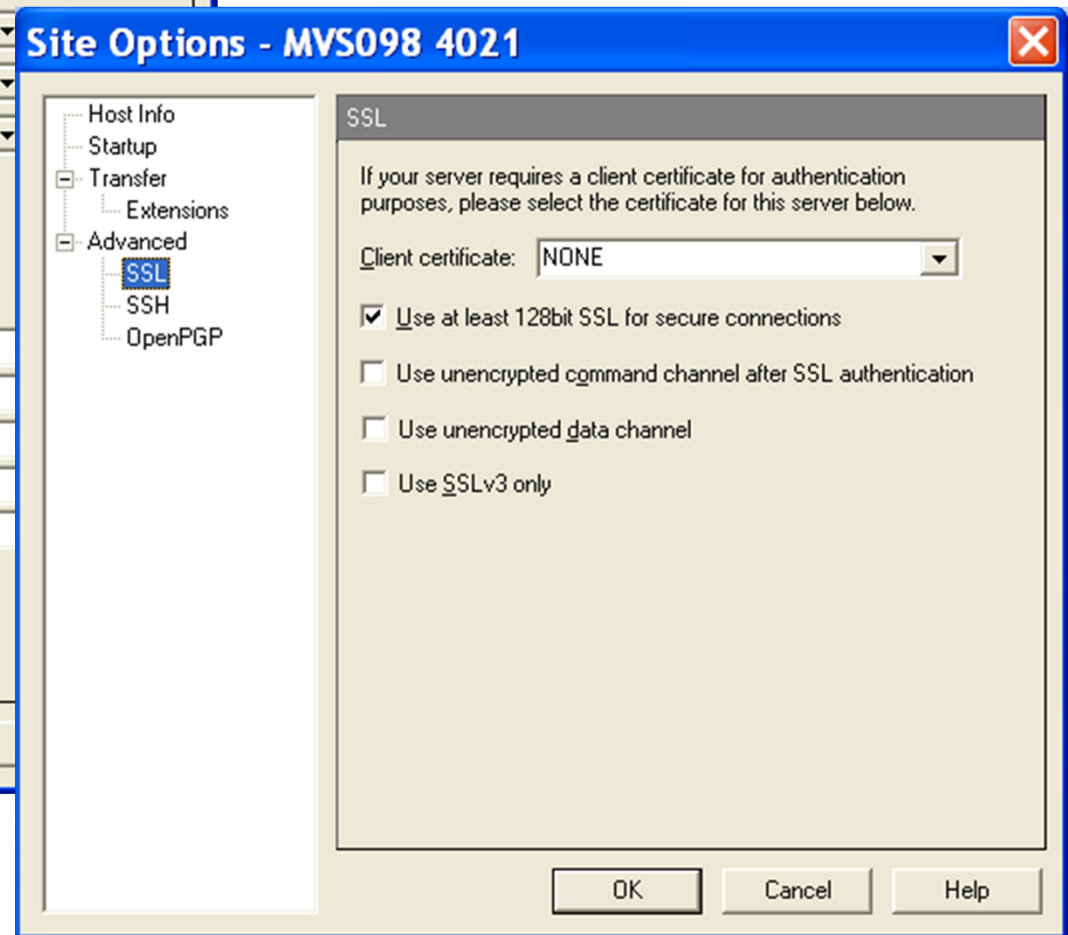
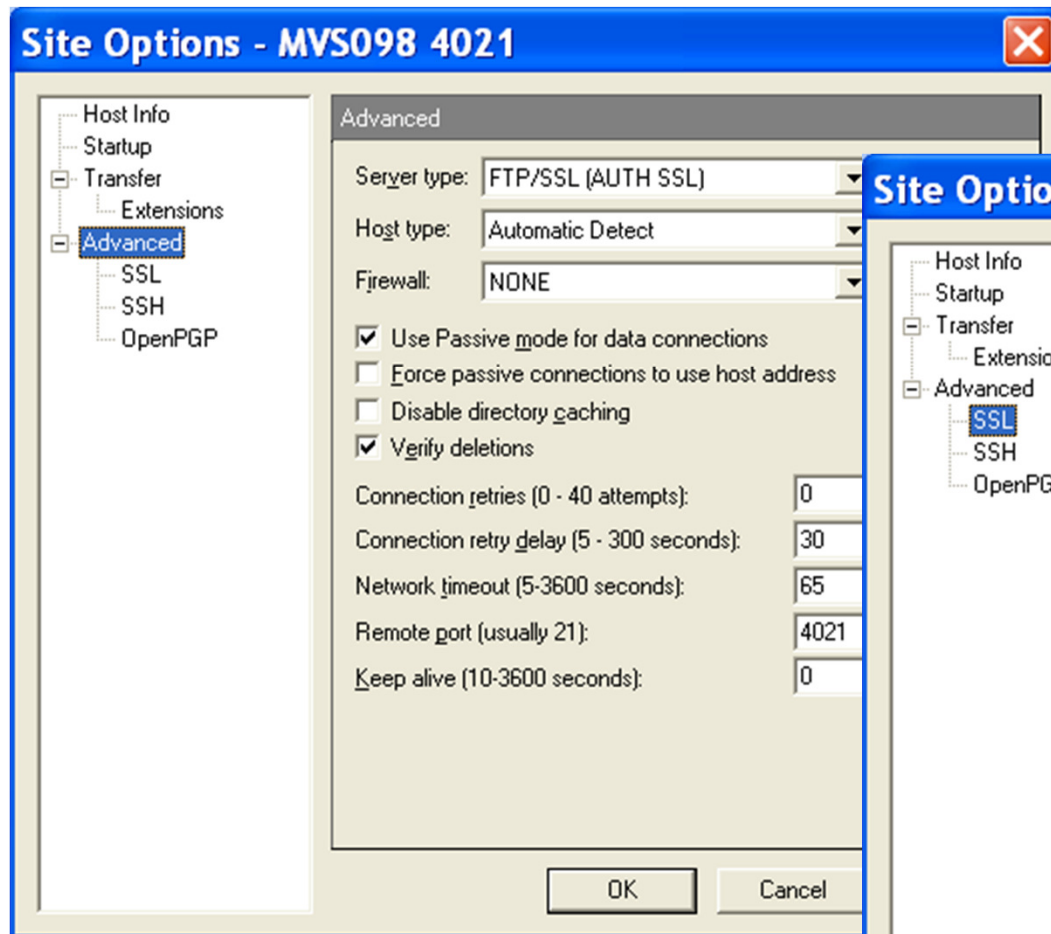
Import lets you pick a file.

Choose the one you just downloaded with your root certificate in base64 encoding.



And set up a WS_FTP Pro site for your secure z/OS FTP server port

Define a server site in WS_FTP Pro pointing to your secure z/OS FTP server.



And connect securely to a z/OS FTP server port that supports SSL/TLS

The screenshot shows the Ipswitch WS_FTP Professional interface. The address bar displays `ftp-authssl://mvs098.tcp.raleigh.ibm.com/USER1.` with the user ID `user1` and a masked password. The interface is split into two panes: 'My Computer' on the left and 'MVS098 4021' on the right. The 'MVS098 4021' pane shows a directory listing for 'USER1.' with files like 'ABCCA.B64' and 'ABCTLSCA.B64'. A status bar at the bottom of the right pane indicates 'Connected to mvs098.tcp.r 440 object(s) - 8.26 MB' with a lock icon. The 'Information Window' at the bottom shows the following log:

```

220.* Your host name is TP60ABC.raleigh.ibm.com
220.*
220 Connection will not timeout.
AUTH TLS
234 Security environment established - ready for negotiation
SSL session NOT set for reuse
SSL Session Started.
Host type [1]: Automatic Detect
USER user1
  
```

Two callout boxes provide context:

- One points to the log lines `AUTH TLS` and `SSL Session Started.`, stating: "Here you see the AUTH command and the set up of the secure connction".
- Another points to the lock icon in the status bar, stating: "This indicates you have a secure FTP session".

What if it doesn't work ?



- Make a visual drawing of where your certificates and private keys are located and what the names of the key rings and certificates are
 - Cross reference your definitions in ATTLS and the remote FTP client to those definitions
 - Make sure certificate authority certificates are stored in RACF as CERTAUTH certificates
- Check all MVS SYSLOG messages for error return codes and reason codes and dig into the documentation to try and get some info out of them
 - Remember z/OS UNIX System Services Messages and Codes is a very good friend !!
- The ATTLS component logs error messages to the z/OS UNIX syslog daemon (syslogd).
 - A syslogd rule should've been set up to direct ATTLS messages to a z/OS UNIX log file
 - ***.TCP*.daemon.* /var/syslog/logs/ATTLS.log**
 - If you are using the z/OS V1R11 syslogd ISPF browser application, search for messages in this file with a message tag of TTLS
 - Limit the search to the time window you're interested in
 - Refer to z/OS Communications Server IP Diagnosis Guide Chapter 30 for details on ATTLS error messages and codes
 - Some return codes are referred to the z/OS Cryptographic Services System Secure Sockets Layer Programming
- The FTP server also logs errors to the z/OS UNIX syslog daemon
 - ***.FTP*.daemon.* /var/syslog/logs/ftp.log**

```
00000024 Jul 30 10:41:07 MVS098/TCPCS      TCPCS      TTLS[10]: 10:41:07
TCPCS      EZD1286I TTLS Error GRPID: 00000001 ENVID: 00000001
CONNID: 0000007E LOCAL: ::0..1126 REMOTE: ::0..2252 JOBNAME:
JESES002 USERID: TCPCS RULE: ABC_NJE~2  RC: 503 Initial
Handshake 00000000 7E60A378
```

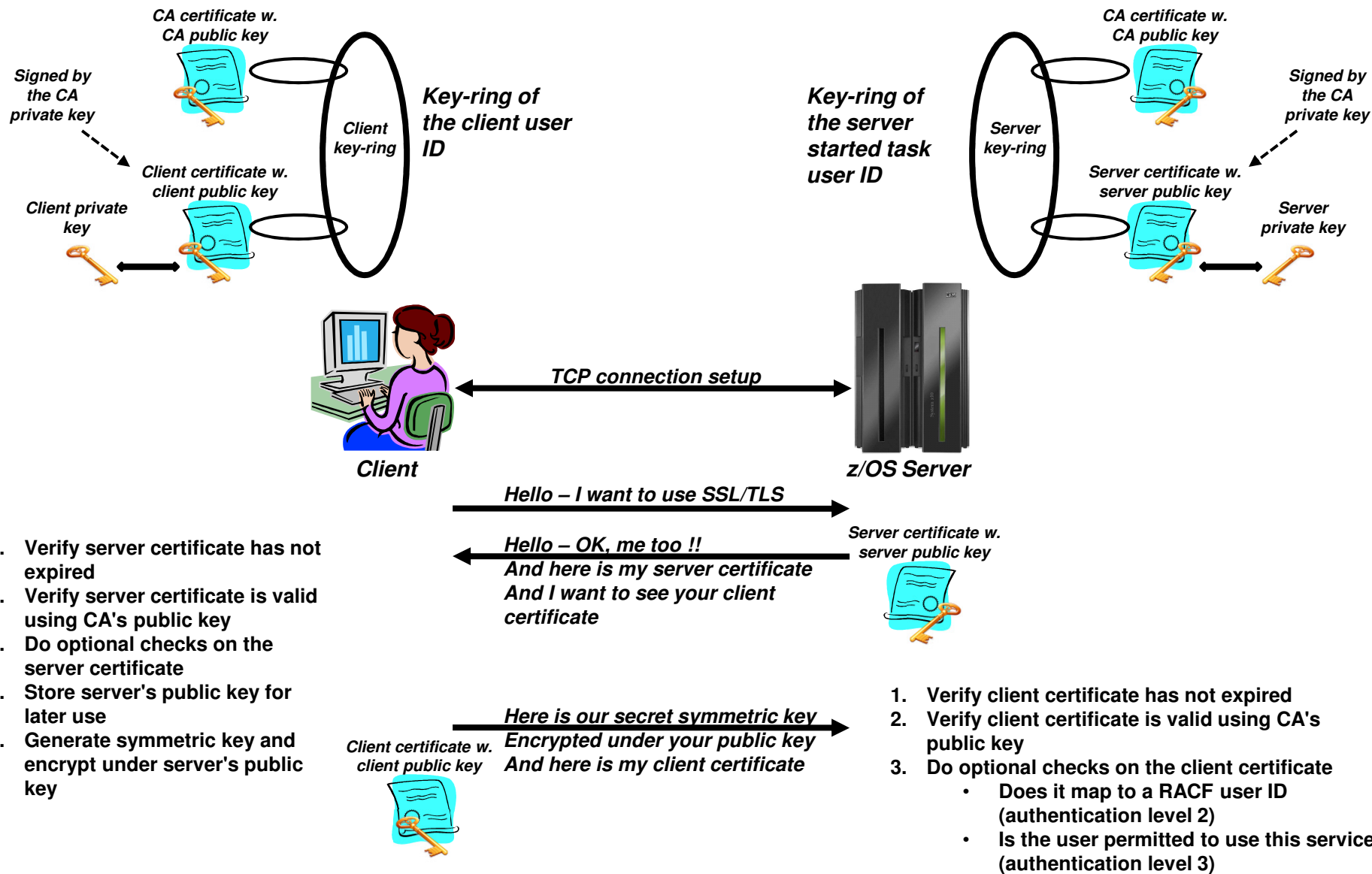


Safe and Secure Transfers with z/OS FTP

z/OS FTP Server with server
authentication and client
authentication – client WS_FTP Pro
on Windows



What is needed for z/OS Server and client authentication



1. Verify server certificate has not expired
2. Verify server certificate is valid using CA's public key
3. Do optional checks on the server certificate
4. Store server's public key for later use
5. Generate symmetric key and encrypt under server's public key

1. Verify client certificate has not expired
2. Verify client certificate is valid using CA's public key
3. Do optional checks on the client certificate
 - Does it map to a RACF user ID (authentication level 2)
 - Is the user permitted to use this service (authentication level 3)

z/OS FTP server options for authenticating an FTP client

Authentication level	FTP server SECURE_LOGIN option	Description
Level 1	REQUIRED	The authenticity and validity of the client certificate is verified against the trusted roots in the FTP server's key-ring.
Level 2	VERIFY_USER	Same as level 1 PLUS a verification that the client certificate is registered by RACF and mapped to a known RACF user ID.
Level 3	VERIFY_USER	Same as level 2 PLUS a verification that the user ID has permission to a SERVAUTH profile that represents this specific FTP server: EZB.FTP.sysname.ftpdemonname.PORTnnnnn

User private key and certificate

- Start creating a private key and certificate for your z/OS user ID
 - Connect the certificate to the user's keyring
- Export the private key and certificate into a PKCS#12 binary file format
 - Download that data set to your FTP client product

```
//ALFREDCI JOB 1,ALFRED,CLASS=A,MSGCLASS=X,NOTIFY=USER1
//*
//IEFPROC EXEC PGM=IKJEFT01,REGION=4M,DYNAMNBR=10
//SYSTSPRT DD SYSOUT=* BATCH TSO SESSION LOG
//SYSTSIN DD *
RACDCERT ID(USER1) GENCERT +
    SUBJECTSDN(CN('USER1 CERT JULY 2010') +
    OU('CS Z/OS') +
    O('IBM') +
    C('US')) +
    NOTBEFORE(DATE(2010-07-19)) +
    NOTAFTER(DATE(2020-12-31)) +
    WITHLABEL('USER1 TLS JULY 2010') +
    SIGNWITH(CERTAUTH LABEL('ABCTLS CA'))
RACDCERT ID(USER1) EXPORT (LABEL('USER1 TLS JULY 2010')) +
    DSN('USER1.USER1.DER.P12') +
    FORMAT(PKCS12DER) +
    PASSWORD('XXXXXX')
RACDCERT ID(USER1) CONNECT(LABEL('USER1 TLS JULY 2010') +
    RING(USER1RING))
RACDCERT ID(USER1) LISTRING(USER1RING)
```

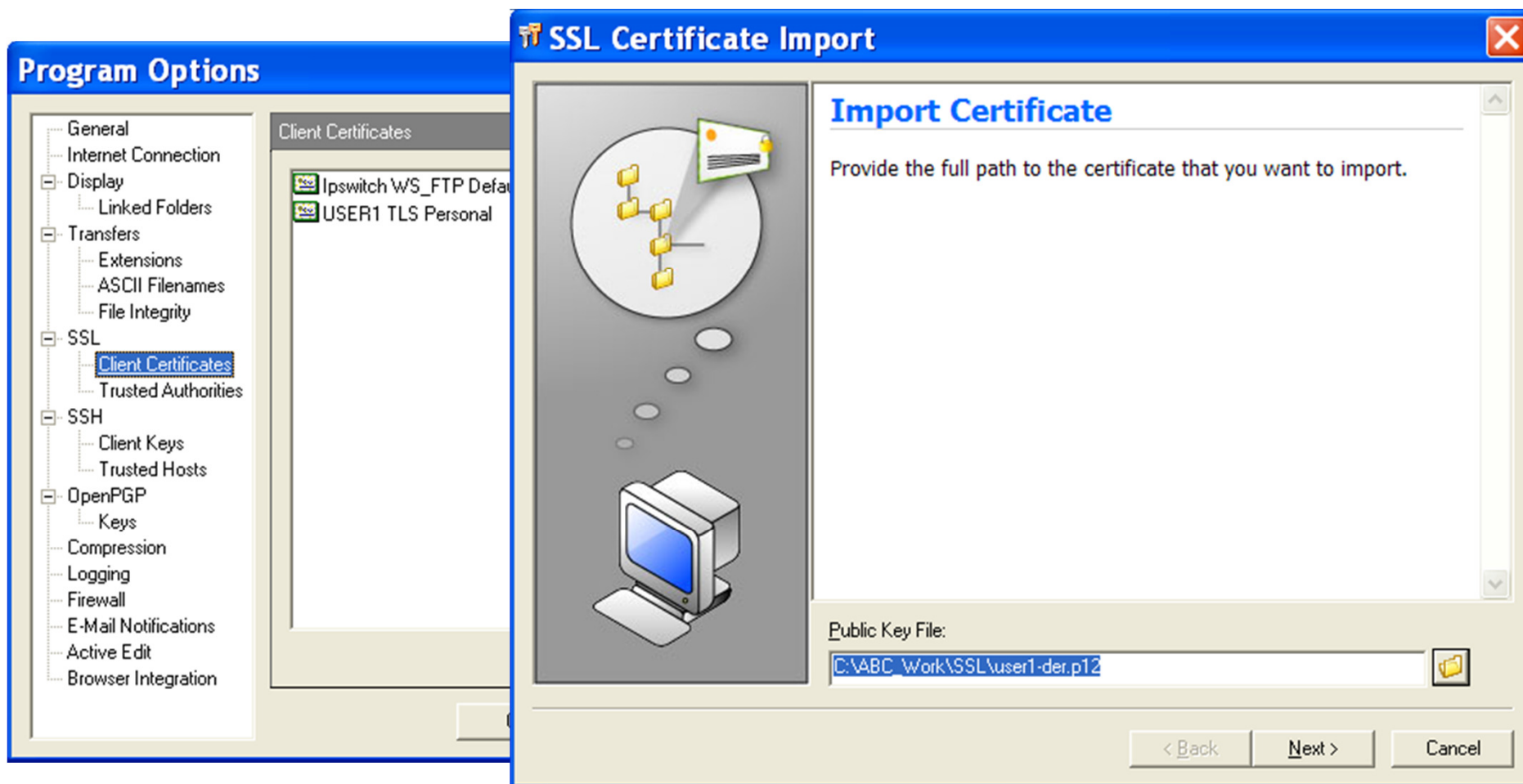
Personal certificate
and private key

Export into PKCS#12
Format with a password

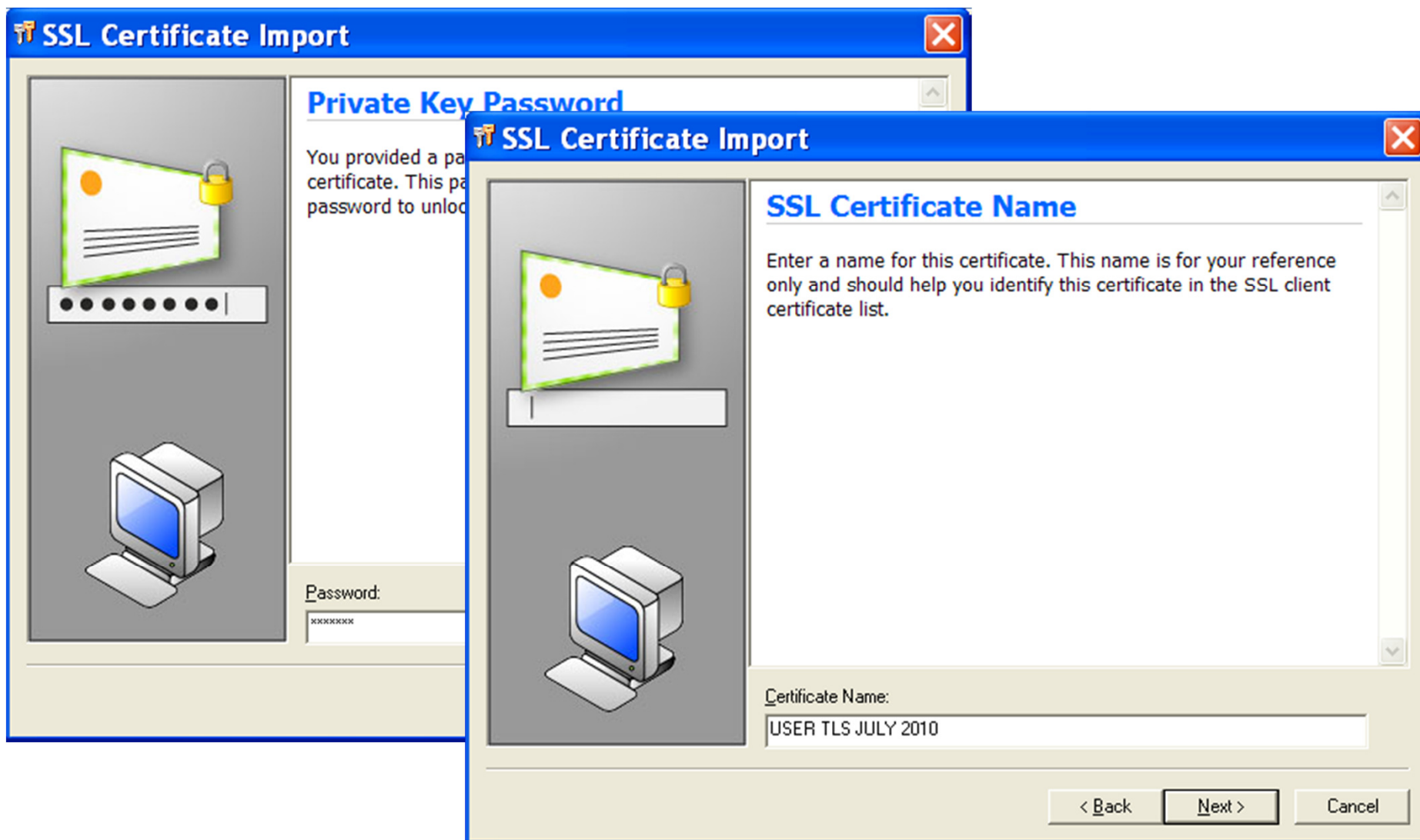
Connect to user's
Keyring in RACF

Import user's private key and certificate into WS_FTP Pro

- Import the PKCS#12 file into your secure FTP client product as a client certificate
 - Select tools, options, and Client Certificates – followed by import.

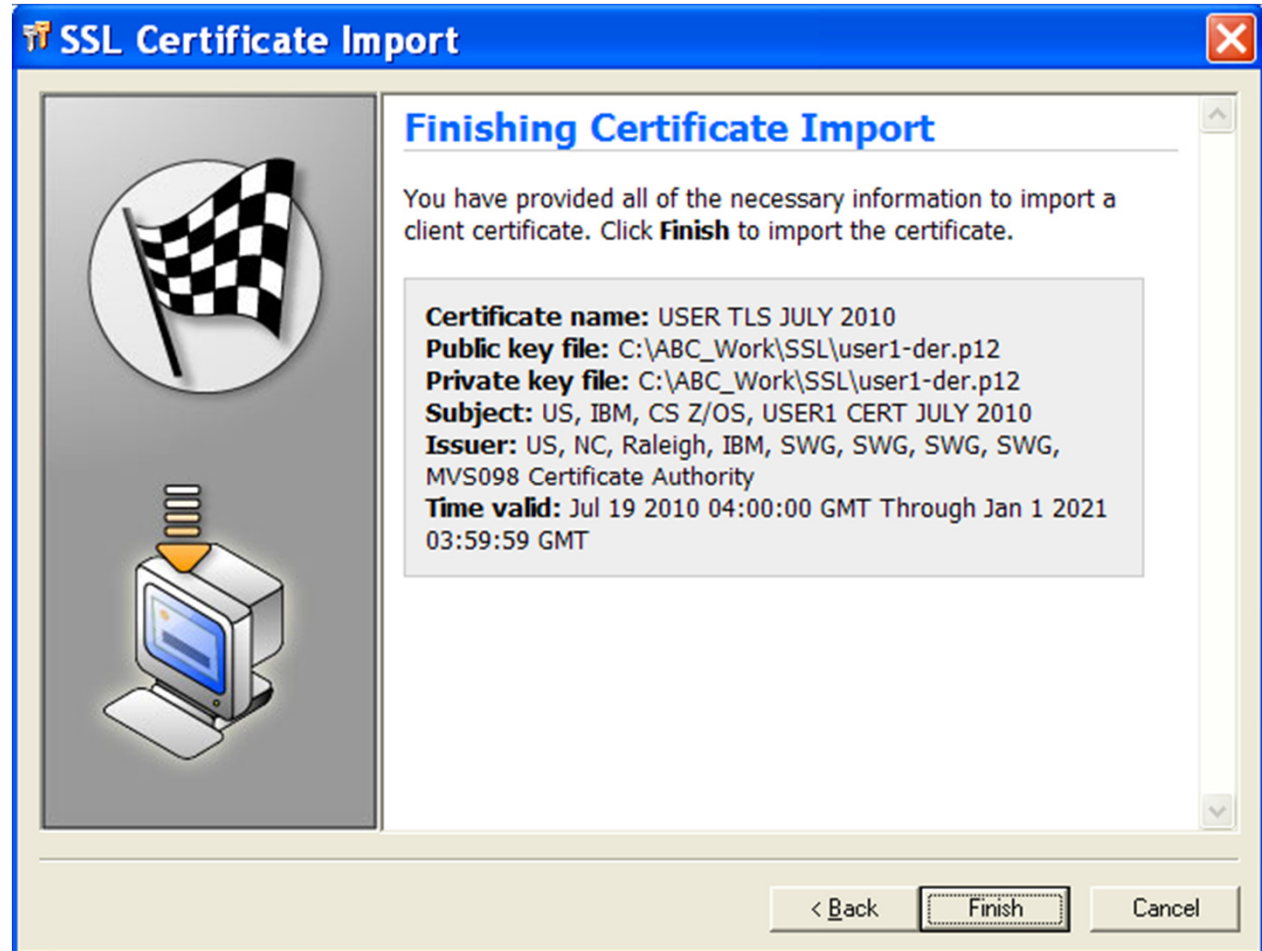


Import user's private key and certificate into WS_FTP Pro



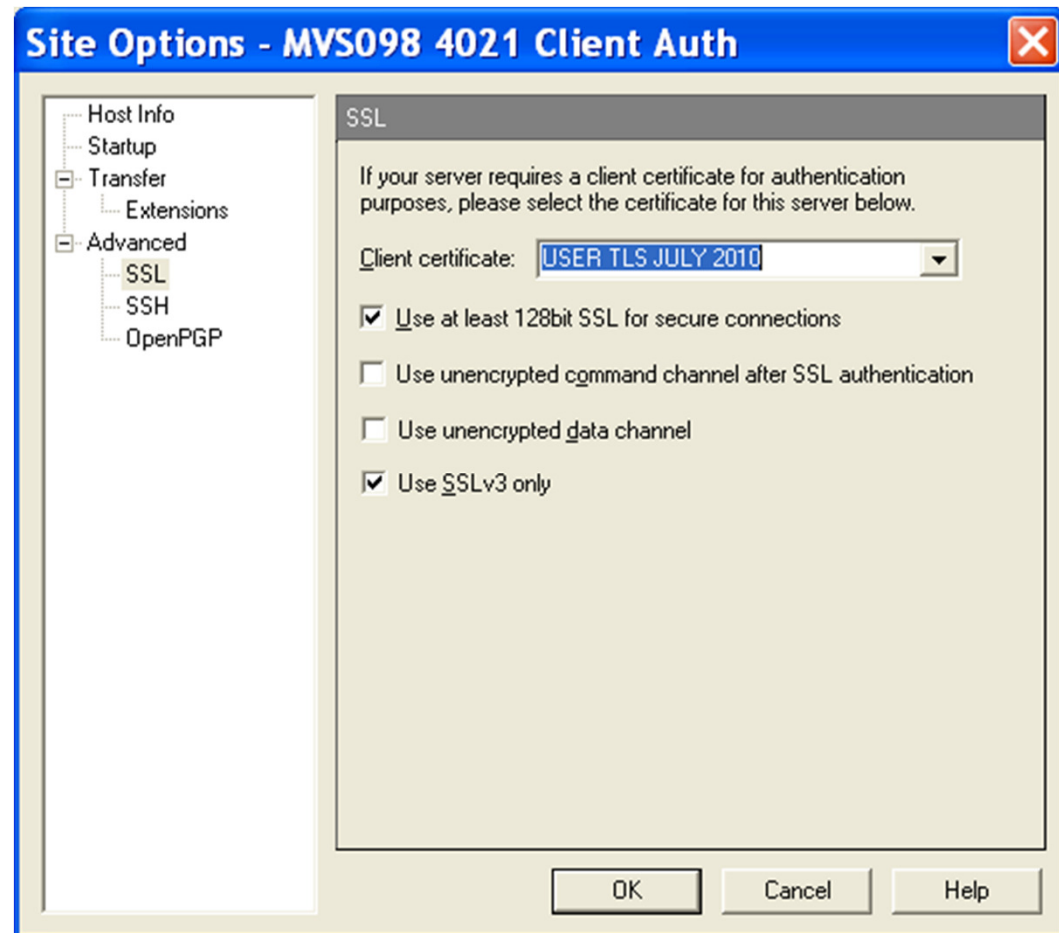
Import user's private key and certificate into WS_FTP Pro

- Verify the information is correct – and then finish the import



Set up WS_FTP Pro to use the personal certificate

- Use the WS_FTP Pro site manager to select which client certificate to use
- In this example, we use the one we just imported



z/OS FTP server setup

- Define an FTP server that requires SSL/TLS connections
 - Client must send an AUTH command or the connection will be rejected
- Require client authentication
 - Client must provide a client certificate
 - The client certificate is verified by RACF
- SSL/TLS is done by ATTLS

EXTENSIONS	AUTH_TLS	; Enable TLS authentication
TLSMECHANISM	ATTLS	; Server-specific or ATTLS
SECURE_FTP	REQUIRED	; Security required/optional
SECURE_LOGIN	VERIFY_USER	; Client authentication requirement
SECURE_PASSWORD	REQUIRED	; Password required
SECURE_CTRLCONN	PRIVATE	; Minimum level of security CTRL
SECURE_DATACONN	PRIVATE	; Minimum level of security DATA
TLSRFCLEVEL	RFC4217	; SSL/TLS RFC level

AT-TLS setup for z/OS FTP server with client and server authentication

Modify Rule

AT-TLS rule name
 Rule name: * ABC_FTP_4121 | Enable rule

Specify settings
 Traffic | Role | Key Ring | Data Endpoints | Security Level | Advanced

Use this panel to specify the traffic settings.

Application name: * ABC_FTP_Port_4121_ClientA

Local port

- All ports
- All ephemeral ports
- Ports: * 4121
Separate multiple ports with a comma

Remote port

- All ports
- All ephemeral
- Ports: *

Indicate the TCP connect direction

- Either
- Inbound only
- Outbound only

Specify jobname
 Jobname:

Modify Rule

AT-TLS rule name
 Rule name: * ABC_FTP_4121 | Enable rule

Specify settings
 Traffic | Role | Key Ring | Data Endpoints | Security Level | Advanced

Use this panel to specify the key ring database and certificate label to use for this rule.

Key ring database

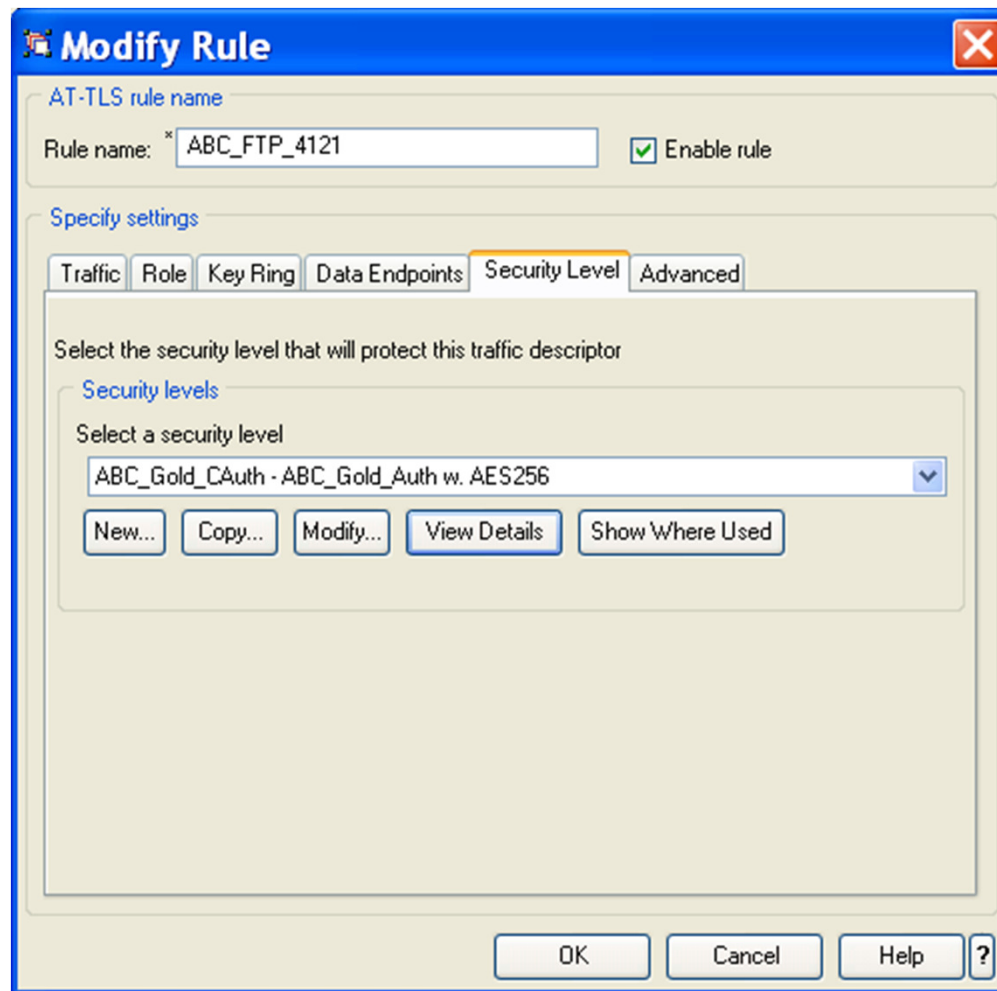
- Use the key ring database defined for the z/OS image
- Use a Simple name (as in an SAF product or in PKCS #11 Token format):
 Key ring: * TLSRING
- Use this z/OS UNIX file system key database:
 Key database: *
 Key database stash file: * or
 Key database password: *

Certificate label:

OK Cancel Help ?

AT-TLS setup for z/OS FTP server with client and server authentication

- **Type:**
 - AT-TLS
- **Encryption:**
 - 0x2F - TLS_RSA_WITH_AES_128_CBC_SHA (first choice)
- **Use TLS Version 1.0:**
 - Yes
- **Use TLS Version 1.1:**
 - Yes
- **Use SSL Version 3:**
 - Yes
- **Use SSL Version 2:**
 - No
- **Client authentication:**
 - Required
- **FIPS 140 Support:**
 - Off



And start a secure FTP session with z/OS

The screenshot displays the Ipswitch WS_FTP Professional interface. The main window shows a connection to `ftp-authssl://mvs098.tcp.raleigh.ibm.com/USER1` with the user ID `user1` and a masked password. The interface is split into two panes: 'My Computer' on the left and 'MVS098 21' on the right. The 'MVS098 21' pane shows a directory listing for 'USER1' with files like 'ABCCA.B64' and 'ABCTLSCA.B64'. The status bar at the bottom indicates 'Connected to mvs098.tcp 446 object(s) - 8.34 MB' with a lock icon, signifying a secure connection. The 'Information Window' at the bottom left shows connection logs, including '220 Connection will not timeout.', 'AUTH TLS', '234 Security environment established - ready for negotiation', 'SSL session NOT set for reuse', 'SSL Session Started', 'Host type (1): IBM MVS', 'USER user1', and '331 Send password please. PASS (hidden)'. A pink callout bubble points to the status bar with the text: 'There is no way you can see on WS_FTP Pro that you use client authentication. You can just see it is a secure connection'.

Use z/OS netstat TTLS report to see the details of the secure FTP connection

Authenticated client user is USER1

```

ConnID: 0000
JobName:      FTP40211
LocalSocket  : ::ffff:9.42.105.45..4021
RemoteSocket : ::ffff:9.37.219.139..2912
SecLevel:    SSL Version 3
Cipher:      2F TLS_RSA_WITH_AES_128_CBC_SHA
CertUserID:  USER1
MapType:     Primary
FIPS140:     Off
TTLSRule:    ABC_FTP_4021_New~2
Priority:     254
LocalAddr:   All
LocalPort:   4021
RemoteAddr:  All
RemotePortFrom: 1024          RemotePortTo: 65535
Direction:   Inbound
TTLSGrpAction: gAct1
  GroupID:    00000004
  TTLSEnabled: On
  Envfile:    /etc/attls.env
  CtraceClearText: Off
  Trace:      7
  SyslogFacility: Daemon
  SecondaryMap: Off
  FIPS140:    Off
TTLSEnvAction: eAct2~ABC_FTP4021_New
  EnvironmentUserInstance: 0
  HandshakeRole: ServerWithClientAuth
  Keyring:      TLSRING
  SSLV2:        Off
  SSLV3:        On
  TL SV1:       On
  TL SV1.1:     On
  ResetCipherTimer: 0
  
```

Client is authenticated via a SAF check

```

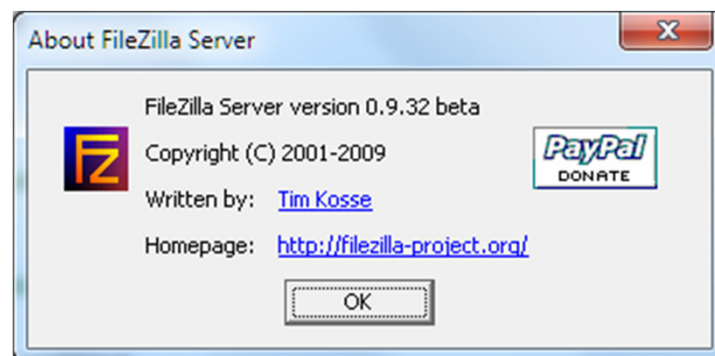
ApplicationControlled: Off
HandshakeTimeout:     10
TruncatedHMAC:        Off
ClientMaxSSLFragment: Off
ServerMaxSSLFragment: Off
ClientHandshakeSNI:   Off
ServerHandshakeSNI:   Off
ClientAuthType:       SAFCheck
CertValidationMode:   Any
TTLSConnAction: cAct2~ABC_FTP4021_New
HandshakeRole:        ServerWithClientAuth
V3CipherSuites:       2F TLS_RSA_WITH_AES_128_CBC_SHA
                       0A TLS_RSA_WITH_3DES_EDE_CBC_SHA
                       35 TLS_RSA_WITH_AES_256_CBC_SHA

CtraceClearText:      Off
Trace:                 7
ApplicationControlled: On
SecondaryMap:          On
  
```

Handshakerole indicates client authentication

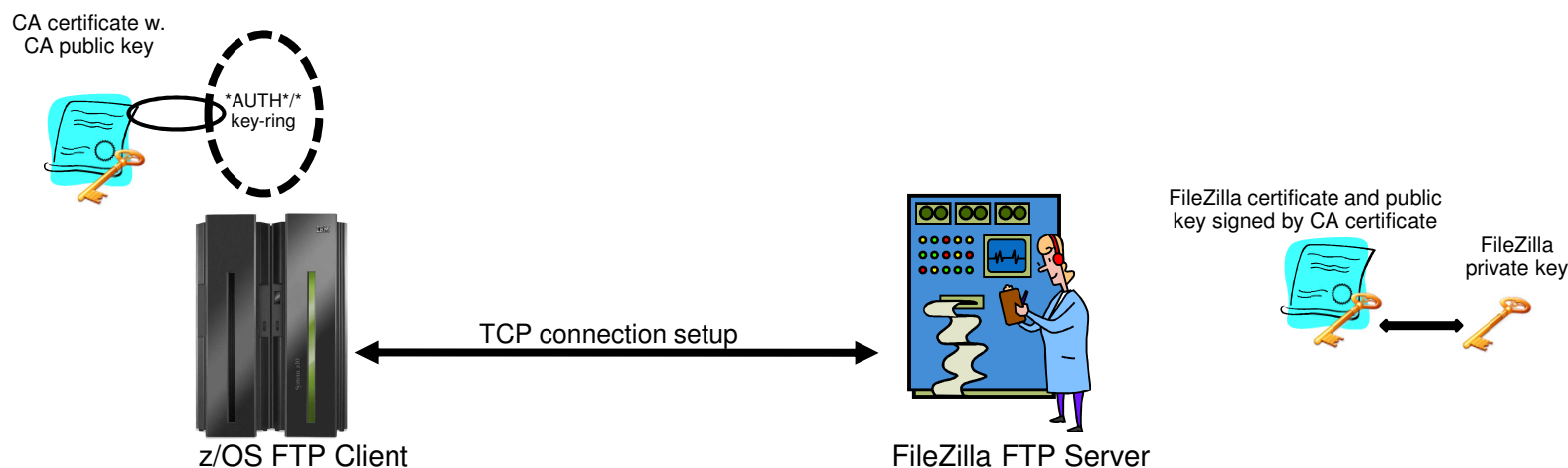
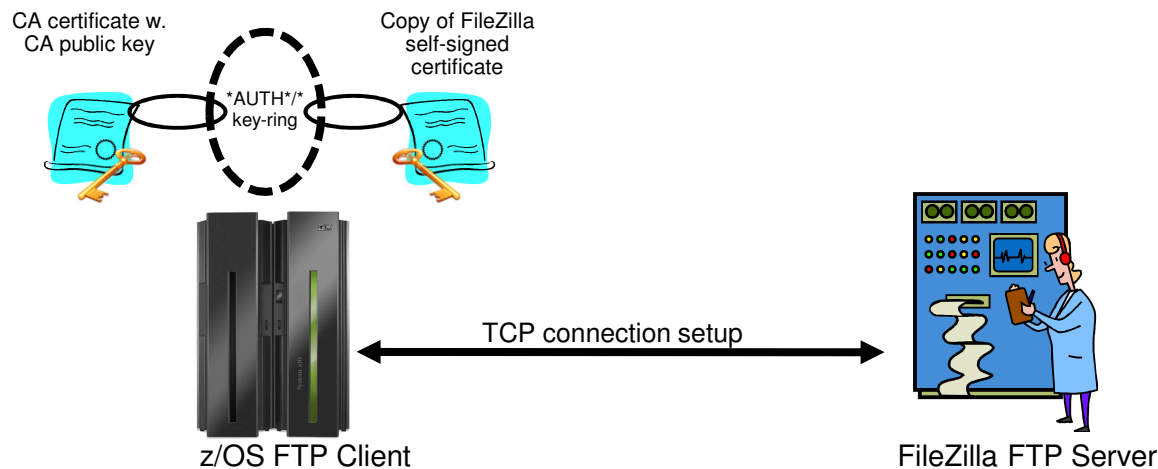
Safe and Secure Transfers with z/OS FTP

z/OS FTP client connecting to a
FileZilla Windows server with server
authentication only



z/OS FTP client to Windows FileZilla server – Two methods for certificate management

Note: The FileZilla server does not support SSL/TLS client authentication, but only server authentication

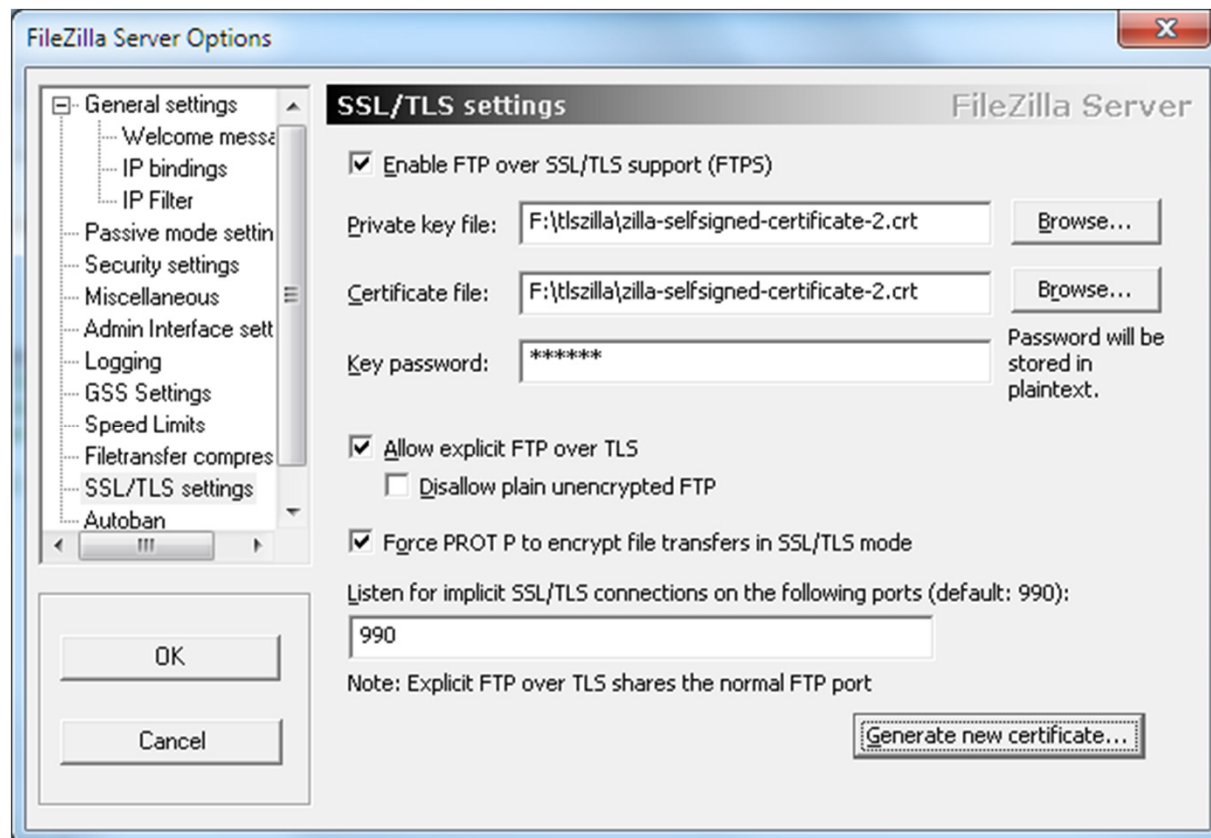


FileZilla Server - outline

- The FileZilla server works with key and certificates in a format known as PEM (Privacy Enhanced Mail)
- A PEM file has two sections:
 - A section where the private RSA key is encoded
 - A section where the certificate is encoded
- z/OS cannot generate a PEM file
 - It can however import a PEM file, but will ignore the private key section
- FileZilla can generate a self-signed certificate and private key, but cannot generate a private key and a certificate request
 - If you use this approach, you need to transfer the self-signed certificate PEM file to z/OS and import it as a CERTAUTH certificate into RACF
- If you want FileZilla to use a certificate that is signed by your root certificate, you can use the following procedure:
 - Generate the private key and certificate on z/OS
 - Export the private key and certificate into a password-protected PKCS#12 file and transfer it to Windows
 - If not already present, download and install openSSL for Windows
 - Use an openSSL command: `pkcs12 -in p12file -out pemfile -nodes -clcerts`

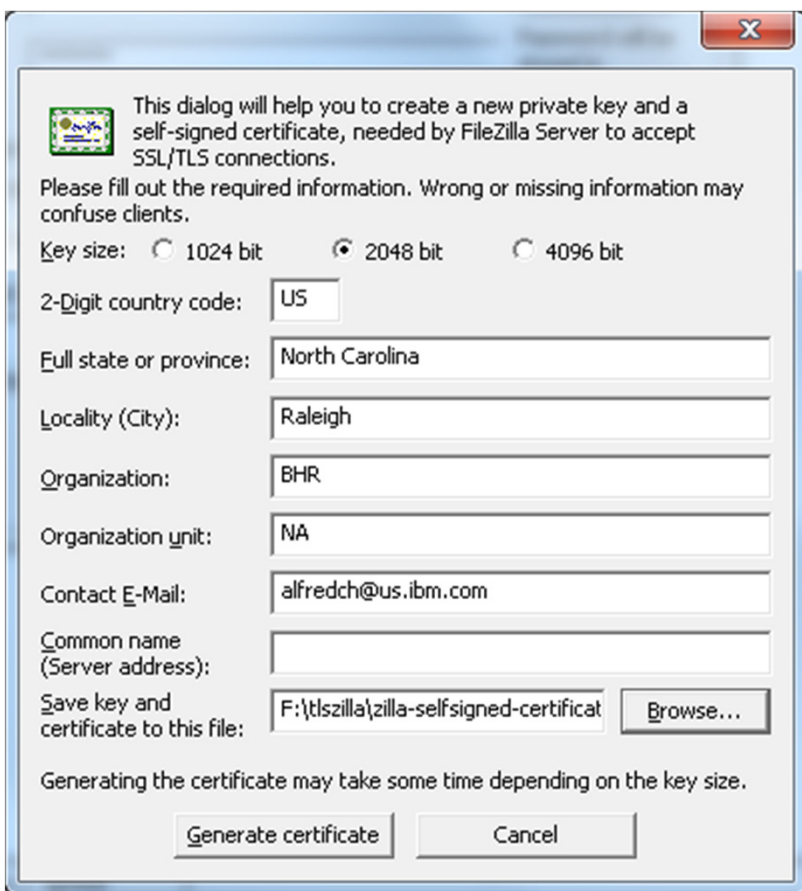
FileZilla server – method 1: self-signed certificate

- Use the 'generate new certificate' button to start creating a self-signed certificate for your FileZilla server



FileZilla server – method 1: self-signed certificate creation

- Enter the requested information including a file name to store both the private key and the certificate in



This dialog will help you to create a new private key and a self-signed certificate, needed by FileZilla Server to accept SSL/TLS connections.

Please fill out the required information. Wrong or missing information may confuse clients.

Key size: 1024 bit 2048 bit 4096 bit

2-Digit country code:

Full state or province:

Locality (City):

Organization:

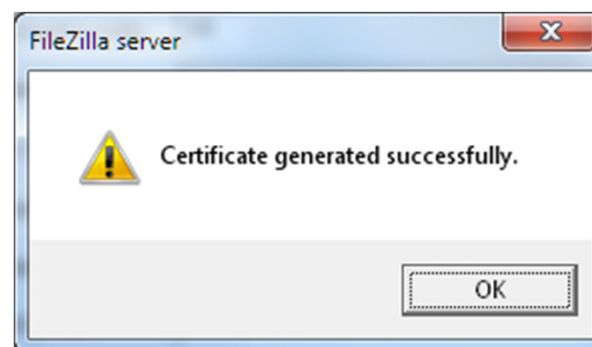
Organization unit:

Contact E-Mail:

Common name (Server address):

Save key and certificate to this file:

Generating the certificate may take some time depending on the key size.



FileZilla server – Method 1: Importing FileZilla's self-signed certificate into RACF

- Upload the FileZilla certificate file to z/OS using an ASCII transfer
 - It is in base64 encoding
- On z/OS, add the FileZilla self-signed certificate as a CERTAUTH certificate.
- No need to connect it to any specific key rings.

```
//ALFREDCI JOB 1,ALFRED,CLASS=A,MSGCLASS=X,NOTIFY=USER1
//IEFPROC EXEC PGM=IKJEFT01,REGION=4M,DYNAMNBR=10
//SYSTSPRT DD SYSOUT=*                BATCH TSO SESSION LOG
//SYSTSIN DD *
RACDCERT CERTAUTH +
      ADD('USER1.ZILLA.SELFSIGN.B64') +
      TRUST +
      FORMAT(CERTB64) +
      WITHLABEL('ABCTLS ZILLASELFSIGNED')
/*
```

```
RACDCERT CERTAUTH ADD('USER1.ZILLA.SELFSIGN.B64') TRUST FORMAT(CERTB64) WITHLABEL('ABCTLS ZILLASELFSIGNED')
IRRD113I The certificate that you are adding is self-signed. The certificate is added with TRUST status.
READY
```

FileZilla server – method 2: generate private key and certificate on z/OS

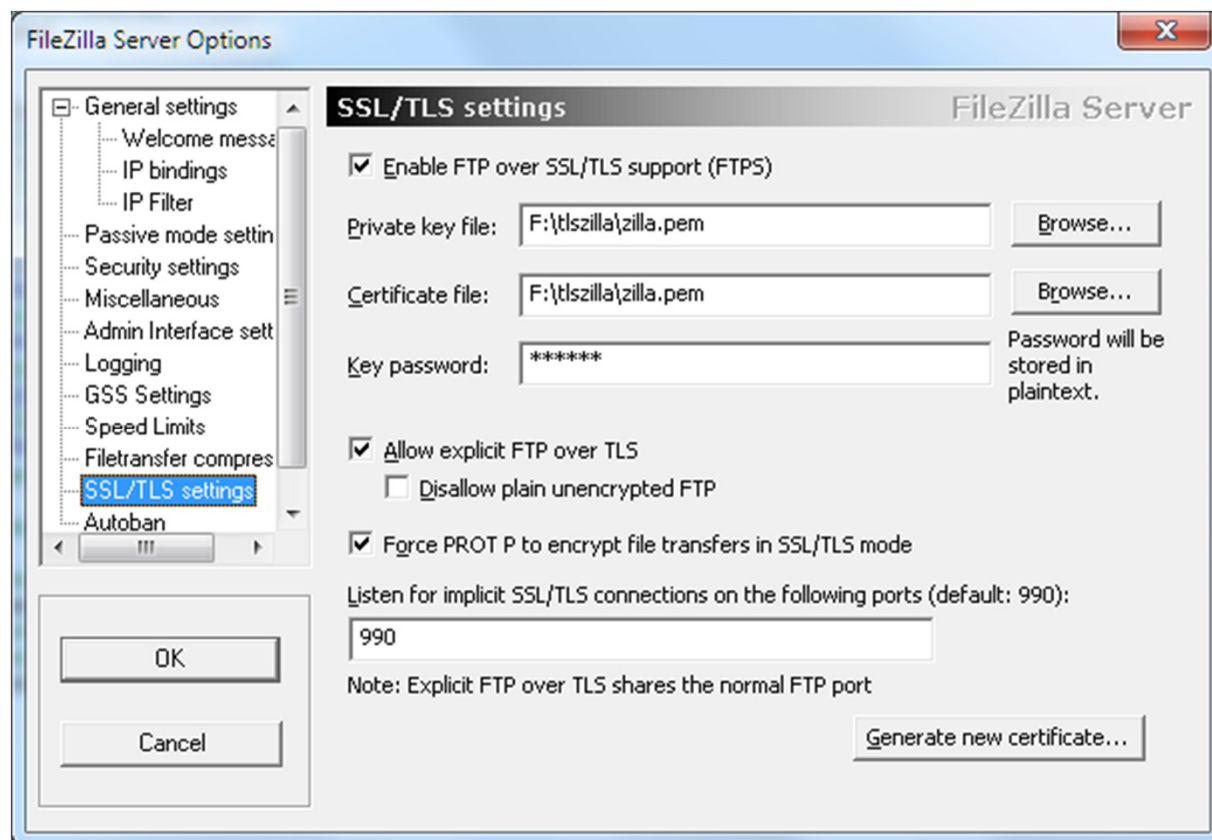
- Use GENCERT command to generate the private/public key pair, and create a certificate signed with our root certificate
- Export the private key and certificate into a password-protected PKCS#12 file

```
RACDCERT GENCERT +
  SUBJECTSDN( +
    CN('ABC FileZilla Certificate') +
    OU('Z/OS CS V1R11', 'ENS', 'AIM', 'SWG') +
    O('IBM') +
    L('Raleigh') +
    SP('NC') +
    C('US') ) +
  SIZE(1024) +
  NOTBEFORE(DATE(2010-07-18)) +
  NOTAFTER(DATE(2020-12-31)) +
  WITHLABEL('ABCTLS FILEZILLASERV') +
  KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN) +
  SIGNWITH(CERTAUTH LABEL('ABCTLS CA'))
RACDCERT EXPORT (LABEL('ABCTLS FILEZILLASERV')) +
  DSN('USER1.ZILLA.DER.P12') +
  FORMAT(PKCS12DER) +
  PASSWORD('??????')
```


FileZilla Server – method 2: Convert PKCS#12 file to PEM and update FileZilla certificate settings

- To convert from PKCS#12 to PEM, you need a copy of openssl on Windows
 - Can be obtained at <http://gnuwin32.sourceforge.net/packages/openssl.htm>
- After having installed openssl issue the following command:

```
openssl pkcs12  
-in p12file  
-out pemfile  
-nodes -clcerts
```



z/OS FTP client FTP.DATA parameters for secure connections

```

SECURE_MECHANISM  TLS                ; Name of the security mechanism
                                     ; that the client uses when it
                                     ; sends an AUTH command to the
                                     ; server.
                                     ; GSSAPI = Kerberos support
                                     ; TLS    = TLS

TLSMECHANISM      ATTLS              ; SSL/TLS implementer
                                     ; FTP    - FTP use of system SSL
                                     ; ATTLS  - the ATTLS component

SECURE_FTP        ALLOWED            ; Authentication indicator
                                     ; ALLOWED      (D)
                                     ; REQUIRED

SECURE_CTRLCONN   CLEAR             ; Minimum level of security for
                                     ; the control connection
                                     ; CLEAR      (D)
                                     ; SAFE
                                     ; PRIVATE

SECURE_DATACONN   PRIVATE           ; Minimum level of security for
                                     ; the data connection
                                     ; NEVER
                                     ; CLEAR      (D)
                                     ; SAFE
                                     ; PRIVATE

```

AT-TLS policy for secure FTP client connections from z/OS

Modify Rule

AT-TLS rule name

Rule name: *ABC-FTP-Client Enable rule

Specify settings

Traffic | Role | Key Ring | Data Endpoints | Security Level | Advanced

Use this panel to specify the traffic settings.

Application name: *ABC-FTP-Client-To-Win7

Local port

All ports

All ephemeral ports

Ports: *

Separate multiple ports with a comma

Remote port

All ports

All ephemeral ports

Ports: *21

Separate multiple ports with a comma

Indicate the TCP connect direction

Either Inbound only Outbound only

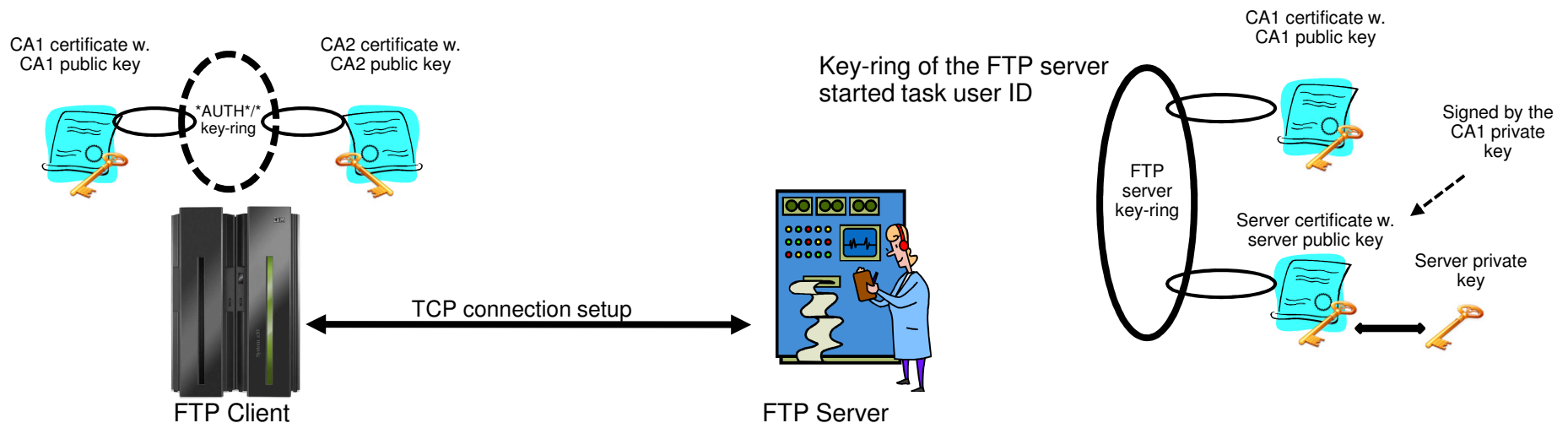
Specify jobname and user ID

Jobname: User ID:

OK Cancel Help ?

Virtual key-rings are useful when z/OS is the FTP client

- If z/OS is the FTP client, does every FTP user on z/OS have to have a key-ring with a copy of the CA certificate?
 - A few releases back, the answer was yes
 - What we call an "administratively heavy process"
 - z/OS V1R8 added support for something known as a virtual key-ring
- To have System SSL check all CERTAUTH certificates in RACF when verifying a certificate that was received during the SSL handshake, specify a key-ring in the client FTP.DATA (or matching AT-TLS definitions) as:
 - KEYRING *AUTH*/*
- If client authentication is required, the z/OS FTP user still needs his/her own key-ring



AT-TLS policy for secure FTP client connections from z/OS

Modify Rule

AT-TLS rule name

Rule name: *ABC-FTP-Client Enable rule

Specify settings

Traffic | Role | **Key Ring** | Data Endpoints | Security Level | Advanced

Use this panel to specify the key ring database and certificate label to use for this rule.

Key ring database

Use the key ring database defined for the z/OS image

Use a Simple name (as in an SAF product or in PKCS #11 Token format):

Key ring: *AUTH*

Use this z/OS UNIX file system key database:

Key database: *

Key database stash file: * or

Key database password: *

Certificate label: *

OK Cancel Help ?

AT-TLS policy for secure FTP client connections from z/OS

Modify Rule

AT-TLS rule name

Rule name: *ABC-FTP-Client Enable rule

Specify settings

Traffic | Role | Key Ring | **Data Endpoints** | Security Level | Advanced

Select the address groups of the host endpoints of the traffic you want to protect.

Local data endpoint

Address group

All_IP_Addresses

New... Copy... Modify... View Details Show Where Used...

IPv4 or IPv6 address, subnet or range

*

Examples: x.x.x.x, x.x.x.x/yy, x.x.x.x-y.y.y.y
x::x, x::x/yyy, x::x-y::y

Remote data endpoint

Address group

All_IP_Addresses

New... Copy... Modify... View Details Show Where Used...

IPv4 or IPv6 address, subnet or range

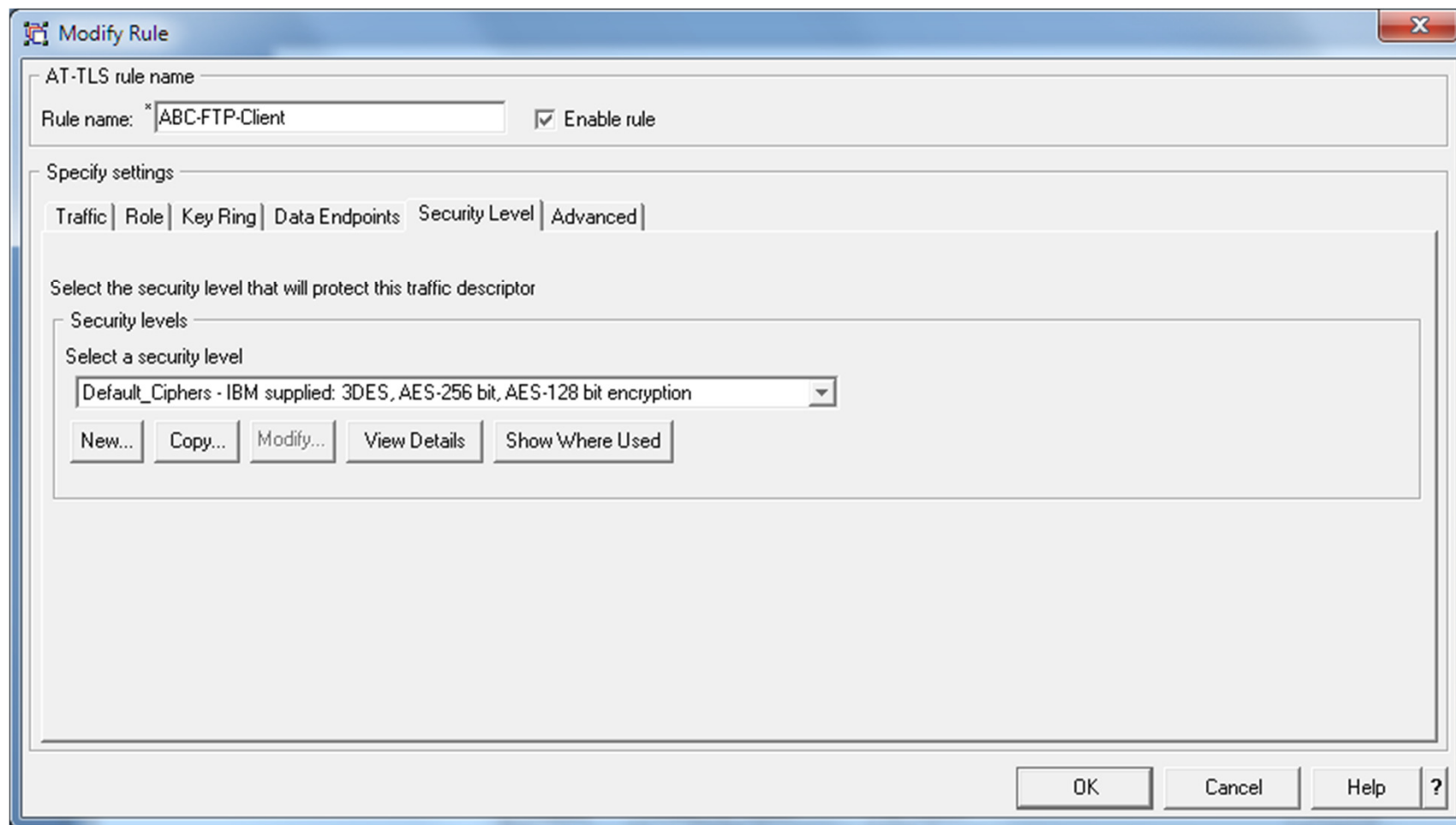
* 9.65.228.73

Examples: x.x.x.x, x.x.x.x/yy, x.x.x.x-y.y.y.y
x::x, x::x/yyy, x::x-y::y

OK Cancel Help ?

My FileZilla server IP address

AT-TLS policy for secure FTP client connections from z/OS



Sample z/OS FTP client session with secure FileZilla server

```
ftp -a tls 9.65.228.73

EZA1736I FTP
EZY2640I Using dd:SYSFTPD=USER1.FTP.DATA for local site configuration parameters
EZA1450I IBM FTP CS V1R12
EZA1466I FTP: using TCPCS
EZA1456I Connect to ?
EZA1736I 9.65.228.73 (exit
EZA1554I Connecting to: 9.65.228.73 port: 21.
220-FileZilla Server version 0.9.32 beta
220-written by Tim Kosse (Tim.Kosse@gmx.de)
220 Please visit http://sourceforge.net/projects/filezilla/
EZA1701I >>> AUTH TLS
234 Using authentication type TLS
EZA2895I Authentication negotiation succeeded
EZA1701I >>> PBSZ 0
200 PBSZ=0
EZA1701I >>> PROT P
200 Protection level set to P
EZA2906I Data connection protection is private
EZA1459I NAME (9.65.228.73:USER1):
EZA1701I >>> USER alfred
331 Password required for alfred
EZA1789I PASSWORD:
EZA1701I >>> PASS
230 Logged on
EZA1460I Command:
EZA1736I dir
EZA1701I >>> EPSV
229 Entering Extended Passive Mode (|||51309|)
EZA1701I >>> LIST
150 Connection accepted
EZA2284I drwxr-xr-x 1 ftp ftp          0 Jan 24 2010 $RECYCLE.BIN
EZA2284I drwxr-xr-x 1 ftp ftp          0 Jan 08 2010 05818b61f89fef75d8745f
EZA2284I drwxr-xr-x 1 ftp ftp          0 May 30 2009 BACKUP
EZA2284I drwxr-xr-x 1 ftp ftp          0 Dec 10 2009 CMPNENTS
. . . . .
```


Netstat TTLS report for the z/OS FTP client connection to FileZilla

```

ConnID: 00002AC
JobName: USER1
LocalSocket: 9.42.130.98..1171
RemoteSocket: 9.65.228.73..21
SecLevel: TLS Version 1
Cipher: 35 TLS_RSA_WITH_AES_256_CBC_SHA
CertUserID: N/A
MapType: Primary
FIPS140: Off
TTLSRule: ABC-FTP-Client~1
Priority: 255
LocalAddr: All
LocalPortFrom: 1024 LocalPortTo: 65535
RemoteAddr: 9.65.228.73
RemotePort: 21
Direction: Outbound
TTLSGrpAction: gAct1
  GroupID: 00000002
  TTLSEnabled: On
  Envfile: /etc/attls.env
  CtraceClearText: Off
  Trace: 7
  SyslogFacility: Daemon
  SecondaryMap: Off
  FIPS140: Off
TTLSEnvAction: eAct1~ABC-FTP-Client-To-Win7
EnvironmentUserInstance: 0
HandshakeRole: Client
Keyring: *AUTH*/*
SSLV2: Off
SSLV3: On
TLV1: On
TLV1.1: On
ResetCipherTimer: 0
ApplicationControlled: Off
HandshakeTimeout: 10
TruncatedHMAC: Off
ClientMaxSSLFragment: Off
ServerMaxSSLFragment: Off
ClientHandshakeSNI: Off
ServerHandshakeSNI: Off
ClientAuthType: Required
CertValidationMode: Any
  
```

```

TTLSConnAction: cAct1~ABC-FTP-Client-To-Win7
HandshakeRole: Client
V3CipherSuites:
  35 TLS_RSA_WITH_AES_256_CBC_SHA
  39 TLS_DHE_RSA_WITH_AES_256_CBC_SHA
  37 TLS_DH_RSA_WITH_AES_256_CBC_SHA
  38 TLS_DHE_DSS_WITH_AES_256_CBC_SHA
  36 TLS_DH_DSS_WITH_AES_256_CBC_SHA
  0A TLS_RSA_WITH_3DES_EDE_CBC_SHA
  16 TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA
  10 TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA
  13 TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
  0D TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA
  2F TLS_RSA_WITH_AES_128_CBC_SHA
  33 TLS_DHE_RSA_WITH_AES_128_CBC_SHA
  31 TLS_DH_RSA_WITH_AES_128_CBC_SHA
  32 TLS_DHE_DSS_WITH_AES_128_CBC_SHA
  30 TLS_DH_DSS_WITH_AES_128_CBC_SHA

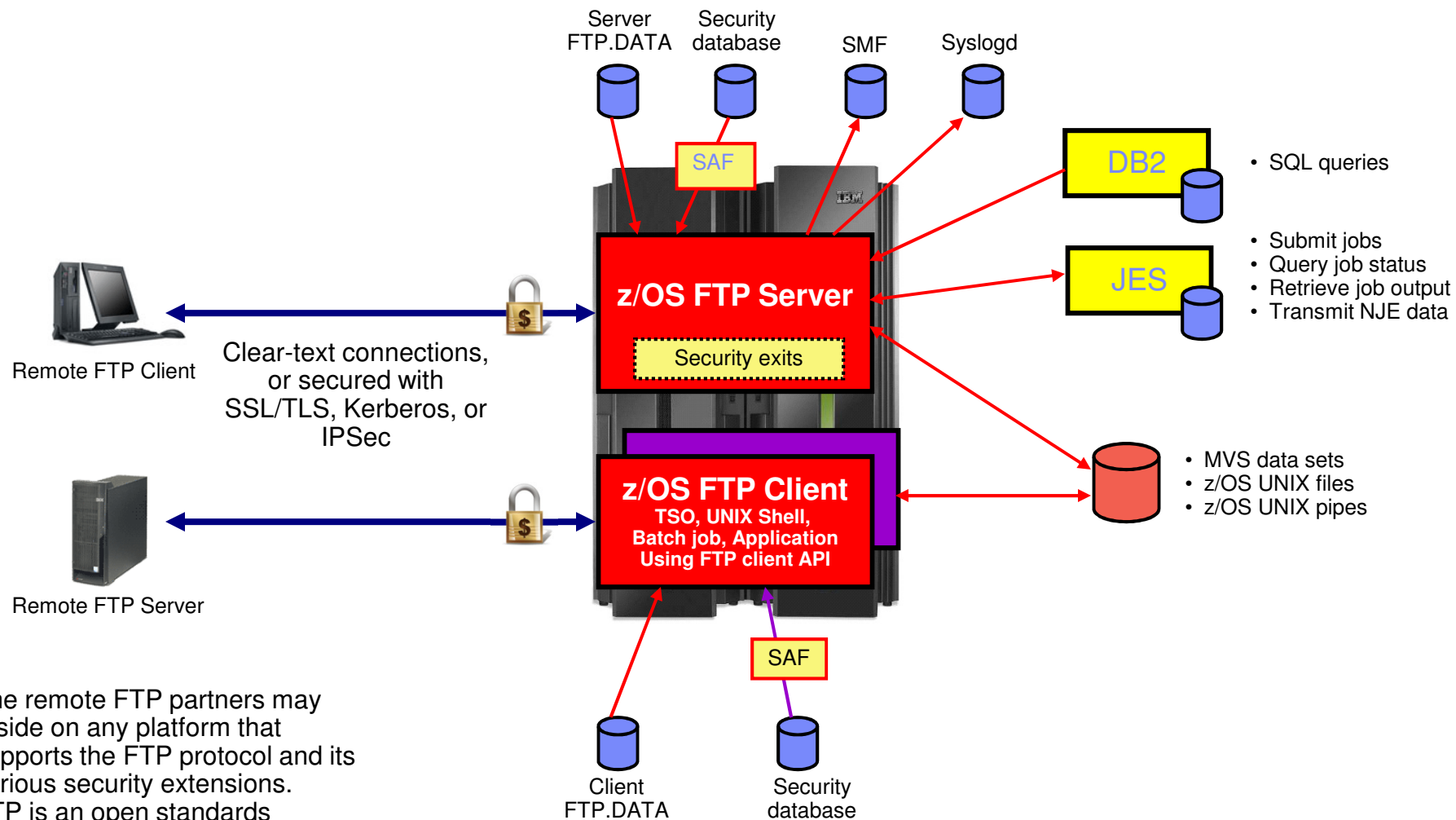
  CtraceClearText: Off
  Trace: 7
  ApplicationControlled: On
  SecondaryMap: On
  
```

Safe and Secure Transfers with z/OS FTP

Appendix A: z/OS FTP – local security



z/OS FTP – the big picture



The remote FTP partners may reside on any platform that supports the FTP protocol and its various security extensions. FTP is an open standards protocol.

Securing the local z/OS FTP server

1. **Basic platform security setup is a pre-requisite**

- Users defined with proper MVS data set access protection
- z/OS UNIX files defined with proper owning user and group along with user/group/world access permissions
- Etc.

2. **FTP server-specific SAF resource definitions**

- Via SERVAUTH resource profiles

3. **Security-related options in the server's FTP.DATA configuration file**

- Controlling various aspects of how the FTP server reacts to selected requests, such as a request for anonymous access

4. **Optional security exits in the FTP server**

- Can be implemented to provide vary granular levels of controls in the FTP server

Selected SAF resource definitions in the SERVAUTH class

- **EZB.PORTACCESS.sysname.tcpname.port_safname**
 - Controls ability for a started task user ID to establish itself as a server on the matching port number in the TCP/IP Profile port reservation section

- **EZB.FTP.sysname.ftpdname.PORTxxxxx**
 - Controls ability to log into an FTP server (control port number) based on the SAF user ID that is being used to log in
 - Initially used for SSL/TLS connections if SECURE_LOGIN VERIFY_USER was coded in the FTP server's FTP.DATA
 - Can be enforced for all types of connections by coding VERIFYUSER TRUE in the server's FTP.DATA - (This support was added in z/OS V1R10)

- **EZB.FTP.sysname.ftpdname.SITE.DUMP** and **EZB.FTP.sysname.ftpdname.SITE.DEBUG**
 - Provides ability to restrict usage of SITE DUMP and DEBUG commands (commands may generate large amount of output)

- **EZB.FTP.sysname.ftpdname.ACCESS.HFS**
 - Provides ability to generally restrict FTP user access to the z/OS UNIX file system

Selected security options in the FTP server's FTP.DATA

▪ **ANONYMOUS**

- Controls the ability to log into your FTP server as an anonymous user
- If the ANONYMOUS option is not included in the server's FTP.DATA, anonymous access is disabled
- Disabled by default – keep it that way, unless you have specific need for it.
 - If you do enable ANONYMOUS, make sure to change the default value of 1 on the ANONYMOUSLEVEL option to 3
 - Also, verify the settings of all the options that start with ANONYMOUS.. – there are a total of 8 including the ANONYMOUS option itself
 - Use the supplied shell script to build a specific z/OS UNIX file system directory structure for anonymous access
 - EMAILADDRCHECK is a syntax check only of the entered email address

▪ **DEBUGONSITE and DUMPSITE**

- Controls the ability to enable dump and debug SITE command options
- If you set these to TRUE, make sure you define the corresponding SERVAUTH profiles so only authorized users can issue these two SITE command options

▪ **PORTCOMMAND, PORTCOMMANDPORT, PORTCOMMANDIPADDR, and PASSIVEDATACONN**

- Control the ability of your FTP server to participate in three-way proxy mode.

Selected security options in the FTP server's FTP.DATA - continued

▪ **REPLYSECURITYLEVEL**

- Controls how much identification information is sent on the initial 220 greeting message from the FTP server, and also how much detail is returned when MVS data set contention occurs.
- Default is no restrictions (level 0).
- If your auditors request you to send as little information as possible, use a setting of 1 on this option
 - Level 0: 220-FTPABC1 IBM FTP CS V1R11 at MVS098, 16:42:51 on 2009-05-24.
 - Level 1: 220-IBM FTP, 16:45:57 on 2009-05-24.

▪ **ACCESSERRMSG**

- To prevent details of failed log in attempts to be returned to the FTP client user, set this option to FALSE (which is the default).
- You may change it to TRUE in an internal-only shop if you want your users to receive details about their failed log in attempt.

▪ **SECURE_...**

- There are a number of options that start with SECURE_ - they are all used to control the ability of the FTP server to accept secure connections (SSL/TLS or Kerberos)

Selected security options in the FTP server's FTP.DATA - continued

▪ **VERIFYUSER**

- Discussed earlier – extends SAF check of all users' ability to connect to the server's control port number
 - EZB.FTP.sysname.ftpddaemonname.PORTxxxxx

▪ **PASSIVEDATAPORTS**

- Controls which range of port numbers the server may use for passive mode data connections
- Can be very useful if there are filtering firewalls in-between the FTP client and the FTP server

If it is a few years ago you created your server's FTP.DATA data set, I recommend recreating it based on the FTPSDATA member in hlq.SEZAINST – many new options have been added over the last releases and all are included in this sample member for documentation purposes.



FTP server security exit points – extending FTP server security

FTP Server	FTCHKIP	Accept/reject connections based on client and server IP address and port information
	FTCHKPWD	Accept/reject login based on client user ID and/or password
	FTCHKCMD	Accept/reject/modify individual FTP commands and their arguments
	FTCHKJES	Accept/reject submission of a job based on analyzing records of job to be submitted
	FTPPOSTPR	Initiate file transfer post processing based on result of file transfer
	FTPSMFEX	Accept/reject writing of old SMF118 records (no longer recommended)

- If these exits routines are present they will be loaded and called at the defined exit points
- The FTCHKIP exit is called by the FTP daemon, while the others are called by the FTP server (after the new address space has been created)
- The command check routine is the most widely used. It has information about the current command from the client, what the current working directory is, what file-type we are using, etc. It may reject the command or it may modify the command options, such as the file or data set name on a STOR or RETR command. If it does reject the command, it can also return the text that will be returned to the client in the 500 reply
- The FTCHKCMD exit executes under the logged in user's user ID. Installation-defined SAF resource definitions can be checked in that routine if needed
- The exits are normally coded in assembler, but we have seen examples where they were coded in C.

FTP server security exit details

Exit point	Called by	Called when	Main input	Possible actions
FTCHKIP	Daemon address space	When control connection is being accepted by the FTP daemon	Client and server IP addresses and ports	Accept or reject connection setup
FTCHKPWD	Server address space	When the client user sends the PASS command	IP addresses and ports, client user ID and password	Accept or reject login request
FTCHKCMD	Server address space	For every command received over the control connection	IP addresses and ports, client user ID, directory type, file type, current directory, and the FTP command and arguments	Accept, reject, or modify the FTP command
FTCHKJES	Server address space	For every record in a job that is being submitted to JES	IP addresses and ports, the full JES input record	Accept or reject the job submission
FTPOSTPR	Server address space	For every completed file transfer operation	IP addresses and ports, plus details about the completed file transfer	Initiate post processing

Samples for all in hlq.SEZAINST

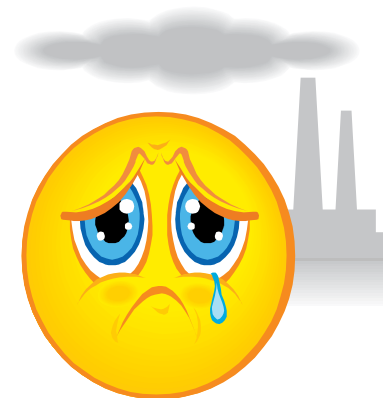
Securing the local z/OS FTP client

- Basic platform security setup is a pre-requisite
 - Users defined with proper MVS data set access protection
 - z/OS UNIX files defined with proper user/group/world access permissions
 - Etc.

- FTP server-specific SAF resource definitions
 - None for the FTP client

- Security-related options in the client's FTP.DATA
 - Not really any

- Optional security exits
 - No exit points in the z/OS FTP client (but requirement to have one has been dutifully noted)



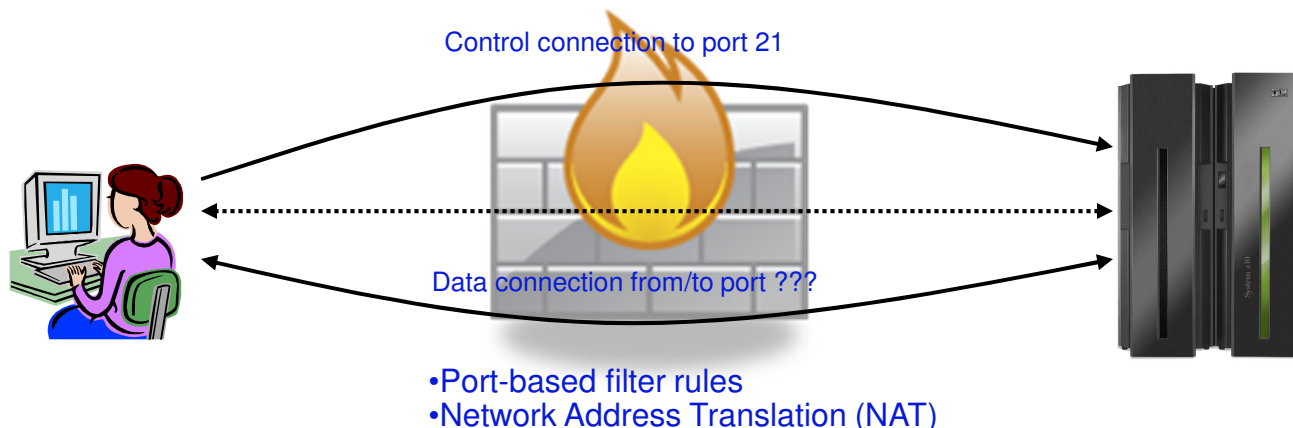
There really isn't much you can do in this area short of protecting the FTP client program itself.

Safe and Secure Transfers with z/OS FTP

Appendix B: Secure FTP: network traversal challenges and solutions



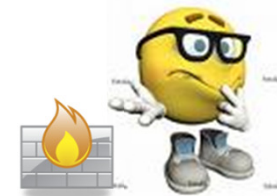
Firewalls and FTP



- **Port-based filter rules – in particular dynamic port rules**
 - FTP control connection is no problem - pre-defined server port number (default 21)
 - Data connection port number (or direction) is not pre-defined, but dynamically negotiated between the FTP client and server
 - The firewall does “deep inspection” (peeks into) the FTP control connection to learn about the negotiated ports and the direction for the data connection

- **NAT**
 - FTP control connection is no problem – only IP headers need translation
 - PORT command and PASV reply refers to local (intranet) IP addresses
 - Firewall needs to do “deep inspection” of the FTP control connection to locate and modify the IP address information in the PORT command and the PASV reply

Deep inspection and data modification is impossible when the data on the FTP control connection is secured through encryption and message integrity checking at the end points.



So what if I need both FTP security and firewalls?



I am a "box-in-the-middle" who wants to inspect the data in those IP packets !

No encryption:

SrcIP	DestIP	SrcPort	DestPort	Data
192.168.100.1	192.168.1.1	50001	80	POST / HTTP/1.1 ... <soapenv:Envelope ...



WSS encryption:

SrcIP	DestIP	SrcPort	DestPort	Data
192.168.100.1	192.168.1.1	50001	80	POST / HTTP/1.1 ... <soapenv:Envelope ... <xenc:EncryptedData ... ^%\$##%/%/%%



SSL/TLS encryption:

SrcIP	DestIP	SrcPort	DestPort	Data
192.168.100.1	192.168.1.1	50002	443	@%\$#* &^!:"J)*GVM-><



IPSec encryption:

SrcIP	DestIP	SrcPort	DestPort	Data
192.168.100.1	192.168.1.1	>:!"	*&hU\$\$\$\$	@%\$#dd*&^s^!:"J)*bGVM-> (*hgvvv<



IP header encryption varies based on transport/tunnel mode, and AH/ESP protocol

Your network/our security engineer! czar!

- **No firewalls – no problems**

- Dream on ...



- **No FTP security, but firewalls**

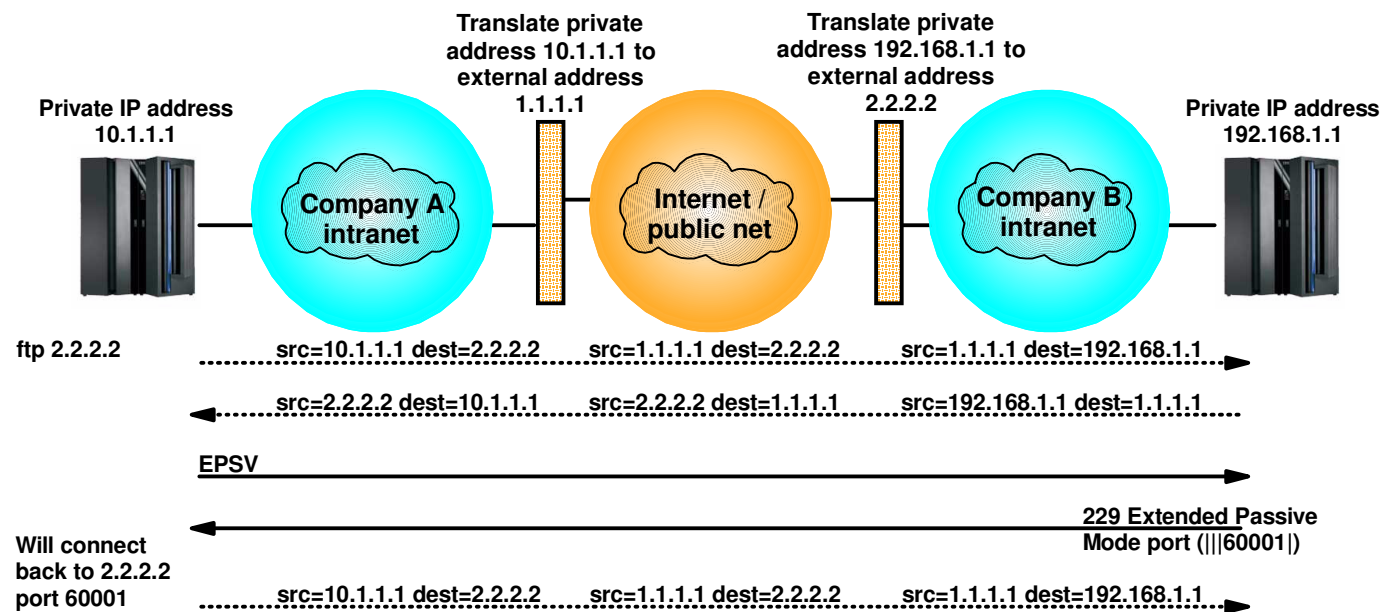
- Firewalls manage port filtering by deep inspection
 - Firewalls manage NAT by deep inspection and modification of data on the control connection

- **FTP security, and firewalls**

- Requires a bit of ingenuity !!!!
 - See the following pages.

RFC 2428: FTP Extensions for IPv6 and NATs

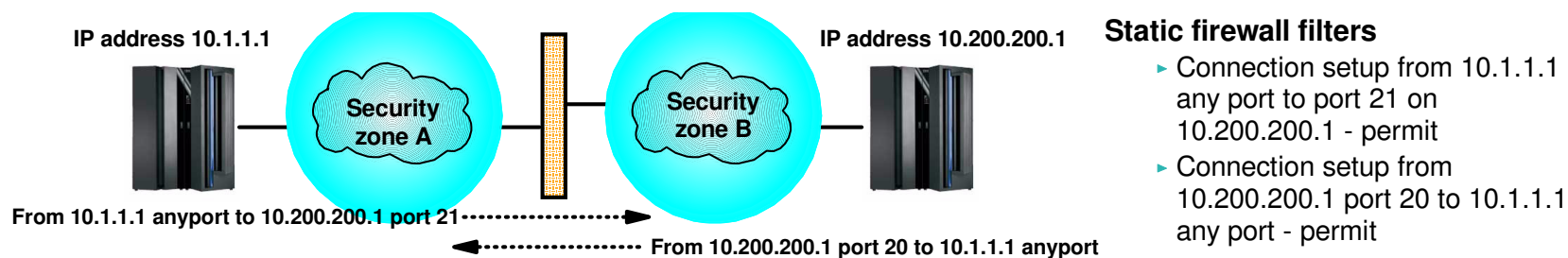
- Extended passive mode (EPSV) will solve NAT problems for secure FTP sessions
 - If using z/OS FTP client to a server that does not support EPSV, code `PASSIVEIGNOREADDR TRUE` in the FTP client's `FTP.DATA`
- The EPSV reply does not include an IP address, but only a port number
 - The FTP client will connect to the same IP address it used for the control connection
- The EPSV and the accompanying extended port command (EPRT) are also used to enable IPv6 support in FTP
 - Used with IPv4, the EPSV command provides NAT firewall relief



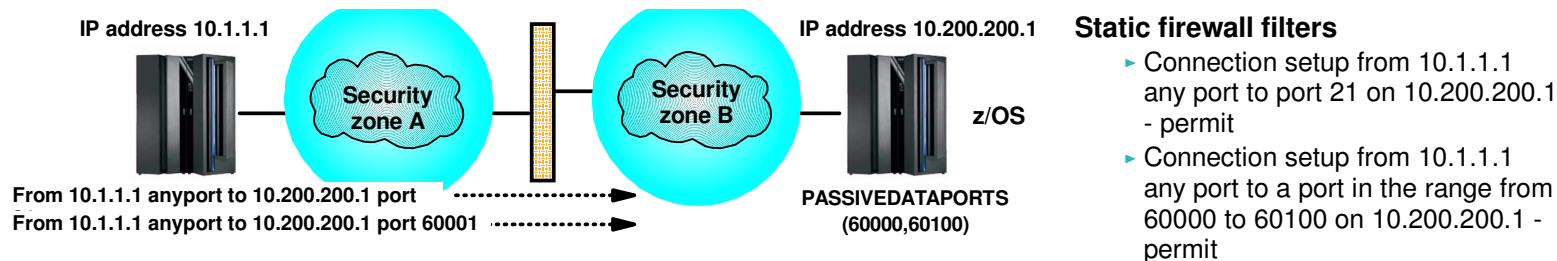
RFC 2428 does not help with dynamic port-based filter rules in firewalls!

How to deal with static port-based filters in firewalls

- If you are able to use active mode FTP, the firewall filters can sometimes be managed:
 - The control connection is permitted inbound to port 21
 - The data connection is permitted outbound from port 20
 - Will work for both standard active mode (PORT) and extended active mode (EPRT)

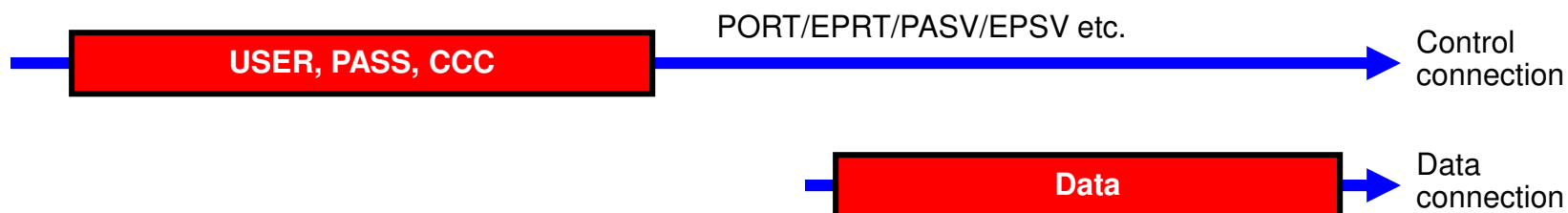


- If you use passive mode FTP, and your server is a z/OS FTP server, you can predefine a range of port numbers to be used for passive mode data connections
 - The control connection is permitted inbound to port 21
 - The data connection is permitted inbound to a port in a pre-defined range
 - Will work for both standard passive mode (PASV) and extended passive mode (EPSV)



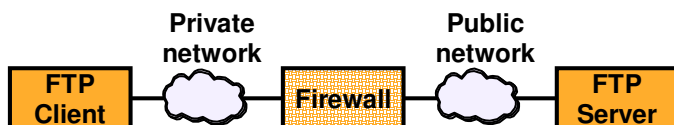
How to deal with dynamic port-based filters in firewalls

- When using dynamic filters, the firewall enables (permits) ports based on IP address and/or port number information in the PORT/EPRT command or the PASV/EPSV reply
 - The original FTP SSL/TLS draft RFC stated that the FTP control connection always had to be encrypted!
 - The final RFC (RFC 4217 "Securing FTP with TLS") relaxes on this requirement and implements a new Clear Command Channel (CCC) FTP command



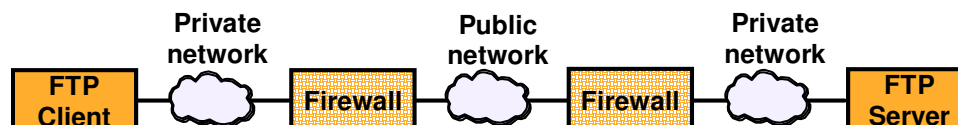
- Both the FTP client and server need to support the CCC command according to RFC 4217
 - Not all FTP clients and servers that support FTP SSL/TLS support the CCC command
 - z/OS added full support for the CCC command in z/OS V1R9 (both z/OS FTP client and server)
 - APAR PK26746 supplied this function for the z/OS FTP client in fall 2006 (back to z/OS V1R4)
 - For those products that claim support, some interoperability issues have been observed !
 - If you have problems getting CCC to work, try to specify TLSRFCLEVEL CCCNONOTIFY instead of TLSRFCLEVEL RFC4217 (applies to both z/OS FTP server and client)
 - CCCNONOTIFY supports a pre-RFC4217 level of the CCC command processing, which some FTP implementations are based upon
 - z/OS FTP server must have SECURE_CTRLCONN CLEAR configured to accept a CCC command
- In general, the CCC command is a solution that solves SSL/TLS-enabled FTP issues with both NAT firewalls and filtering firewalls

FTP and firewall topologies – part 1 of 2



NAT, no or minimal filtering

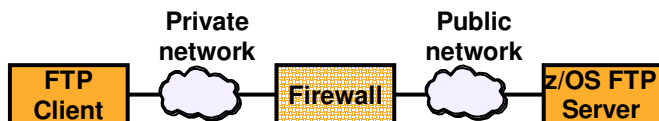
- ✓ Normal passive mode (PASV) will usually work in such a topology.
- ✓ Extended passive mode (EPSV) will also work, but is not generally required.



NAT, no or minimal filtering

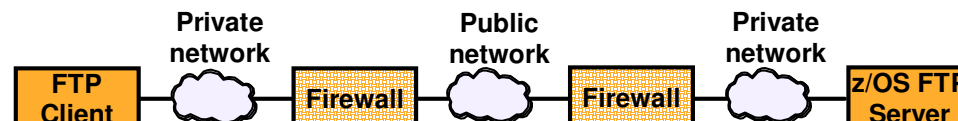
NAT, no or minimal filtering

- ✓ If your partner secure FTP product supports extended passive mode - use extended passive mode (EPSV) from the FTP client.
- ✓ If the FTP client is on z/OS (V1R11) and the partner secure FTP server product does not support EPSV, configure the PASSIVEIGNOREADDR option at your z/OS FTP client to simulate EPSV processing.



NAT, static filtering

- ✓ Use the PASSIVEDATAPORTS option on the z/OS FTP server to predefine a range of port numbers the z/OS FTP server may use for data connections.
 - ✓ Other FTP servers may have similar configuration capabilities
- ✓ Have your firewall administrator add static filter rules for the passive data port range.
- ✓ Normal passive mode (PASV) will usually work in such a topology, but extended passive mode (EPSV) can also be used if supported by the FTP client.

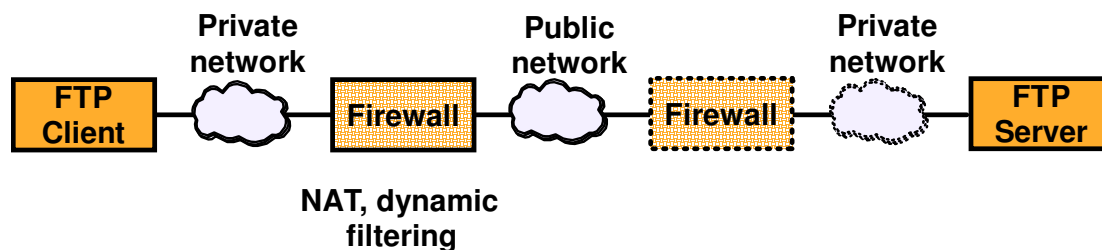


NAT, static filtering

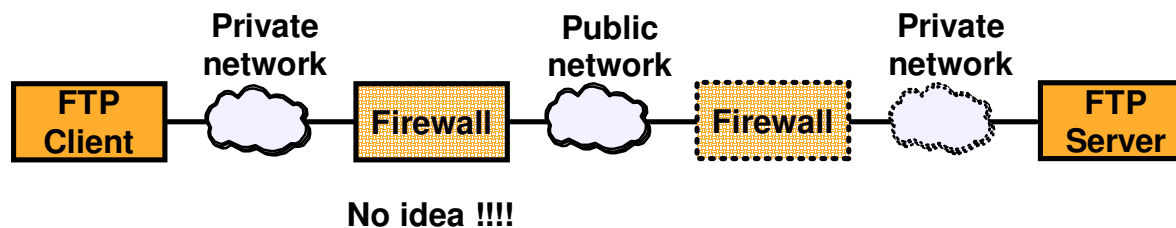
NAT, static filtering

- ✓ Use the PASSIVEDATAPORTS option on the z/OS FTP server to predefine a range of port numbers the z/OS FTP server may use for data connections.
 - ✓ Other FTP servers may have similar configuration capabilities
- ✓ Have your firewall administrator add static filter rules for the passive data port range.
- ✓ In this case, you must use extended passive mode.
- ✓ If the FTP client does not support extended passive mode, you will likely not get this scenario to work.
- ✓ If the FTP client is on z/OS (V1R11) and the partner secure FTP server product does not support EPSV, configure the PASSIVEIGNOREADDR option at your z/OS FTP client to simulate EPSV processing.

FTP and firewall topologies – part 2 of 2



- ✓ Use the CCC command from the FTP client.
- ✓ You will most likely not get this scenario to work without the CCC command support.



- ✓ Use the CCC command from the FTP client.
- ✓ You will most likely not get this scenario to work without the CCC command support

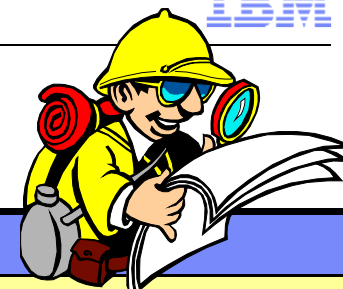
Why it may still fail ..



- Some firewalls are known to apply various validity checks on the FTP control connection data stream.
 - One known check is a check to verify that all interactions on the FTP control connection are terminated with an ASCII new-line (NL) character.
 - Most of those checks will fail when the control connection is secured with SSL/TLS since the data is encrypted.
 - If despite following the above guidelines, you run into problems establishing SSL/TLS secure FTP sessions through firewalls, verify with your firewall administrators whether your firewalls implement such checks on the FTP control connection, and consider disabling those checks.

- Other firewalls are known to disable active mode data connections by default and will block all active mode data connections.
 - Use passive or extended passive mode FTP instead.

- Finally, many firewalls monitor activity on TCP connections and will terminate connections that are idle for a certain period of time.
 - While a large data transfer occurs over an FTP data connection, the FTP control connection is idle.
 - To avoid having firewalls terminate idle FTP connections, consider using the z/OS FTP option FTPKEEPALIVE for the control connection and DATAKEEPALIVE for the data connection.

For more information



URL	Content
http://www.twitter.com/IBM_Commserver 	IBM Communications Server Twitter Feed
http://www.facebook.com/IBMCommserver 	IBM Communications Server Facebook Fan Page
http://www.ibm.com/systems/z/	IBM System z in general
http://www.ibm.com/systems/z/hardware/networking/	IBM Mainframe System z networking
http://www.ibm.com/software/network/commserver/	IBM Software Communications Server products
http://www.ibm.com/software/network/commserver/zos/	IBM z/OS Communications Server
http://www.ibm.com/software/network/commserver/z_lin/	IBM Communications Server for Linux on System z
http://www.ibm.com/software/network/ccl/	IBM Communication Controller for Linux on System z
http://www.ibm.com/software/network/commserver/library/	IBM Communications Server library
http://www.redbooks.ibm.com	ITSO Redbooks
http://www.ibm.com/software/network/commserver/zos/support/	IBM z/OS Communications Server technical Support – including TechNotes from service
http://www.ibm.com/support/techdocs/atmastr.nsf/Web/TechDocs	Technical support documentation from Washington Systems Center (techdocs, flashes, presentations, white papers, etc.)
http://www.rfc-editor.org/rfcsearch.html	Request For Comments (RFC)
http://www.ibm.com/systems/z/os/zos/bkserv/	IBM z/OS Internet library – PDF files of all z/OS manuals including Communications Server

For pleasant reading