



**Welcome's  
the members of the  
NY NJ Metro WebSphere  
MQSeries Users Group**



# Session Overview

---

Today's session is focused on the use of Digital Certificates and SSL with WebSphere MQSeries 5.3.

Vincent Terrone, Technologist,  
Infinity Systems Services, Inc.

[vterrone@infinte-blue.com](mailto:vterrone@infinte-blue.com)

visit us @ [www.Infinite-Blue.com](http://www.Infinite-Blue.com)



# About me

---

**Vincent Terrone, Technologist,  
Infinity Systems Services, Inc.**



**15 Years professional IT experience  
concentrated in application development  
and product development across Enterprise  
and Open Systems platforms.**

**IBM Certified for e-business –  
Solutions Designer**

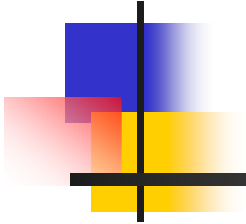




# Agenda

---

- Security Problems
- Current MQSeries Channel Security
- Cryptography
- Secure Sockets Layer
- SSL & WebSphere MQ
- Creating Certificates with OpenSSL
- Storing of Certificates for WMQ
- Websphere MQ Configuration
- Questions and Answers



# Security Problems



# Security Problems

---

- **Eavesdropping**

- How do I stop someone from seeing the information I send?

- **Tampering**

- **Impersonation**



# Security Problems

---

- **Eavesdropping**

- How do I stop someone from seeing the information I send?

- **Tampering**

- How can I detect if someone has intercepted my information and changed it?

- **Impersonation**



# Security Problems

---

- **Eavesdropping**

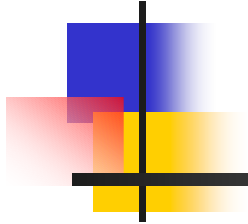
- How do I stop someone from seeing the information I send?

- **Tampering**

- How can I detect if someone has intercepted my information and changed it?

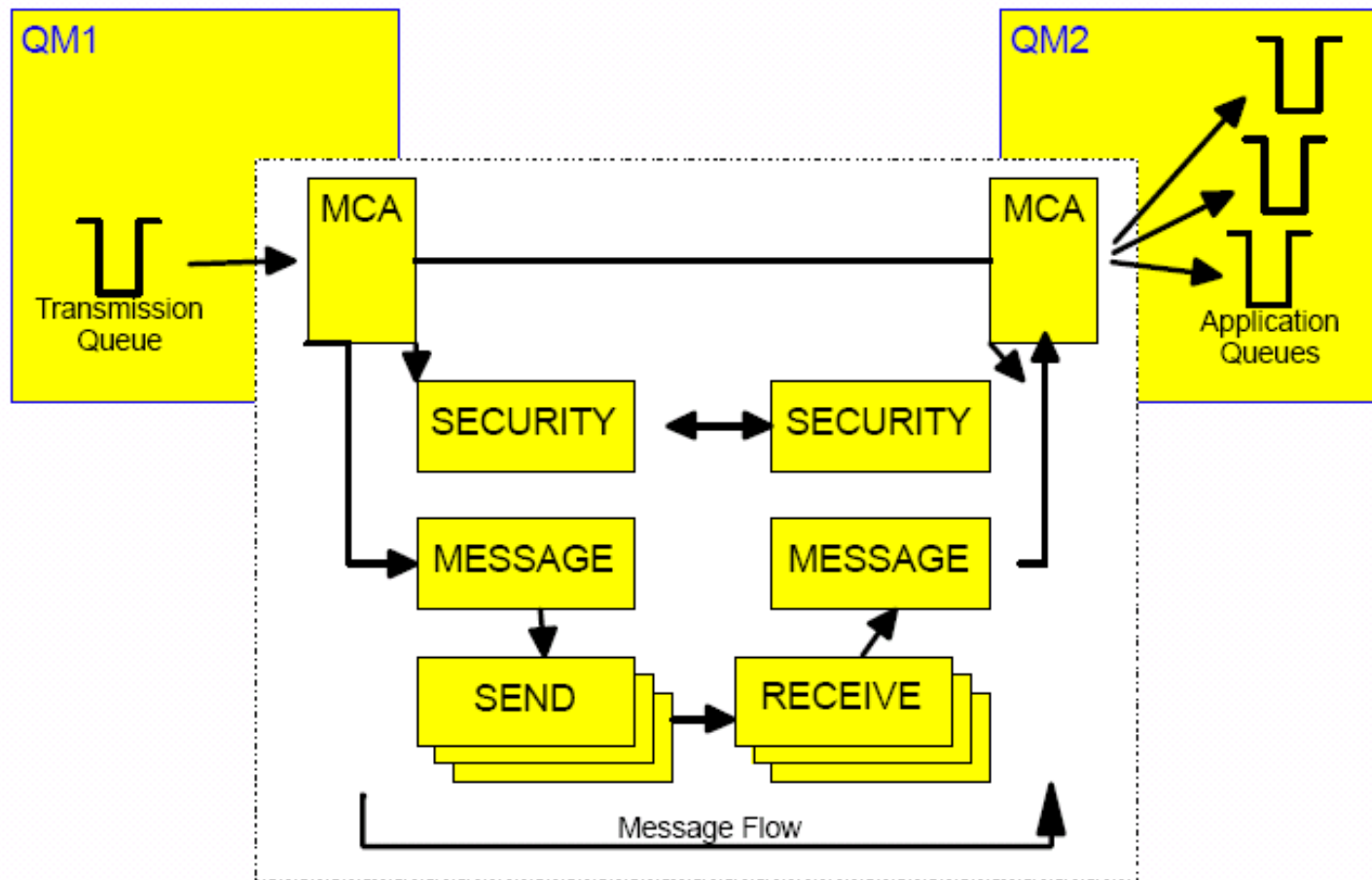
- **Impersonation**

- How can I be sure who the information is from?
- How can I be sure who I am exchanging information with?



# MQSeries Channel Security Framework

# MQ Channel Security Framework





# MQSeries Channel Exits

---

- Security Exit
  - Called after initial channel negotiation
  - Useful for partner authentication
- Message Exit
  - Operates on the full message
  - Useful for encryption of the full message before transmission
- Send/Receive Exit
  - Operates on the transmission buffer
  - Useful for compression and decompression
  - Useful for encryption
- Significant setup required for user exits
- Not 'out of the box' security
  - Does not feel integrated with the product



# Cryptography

---

**crypt·tog·ra·phy** *n.*

- 1. The process or skill of communicating in or deciphering secret writings or ciphers.**
- 2. Secret writing.**

**Source:** *The American Heritage® Dictionary of the English Language, Fourth Edition*

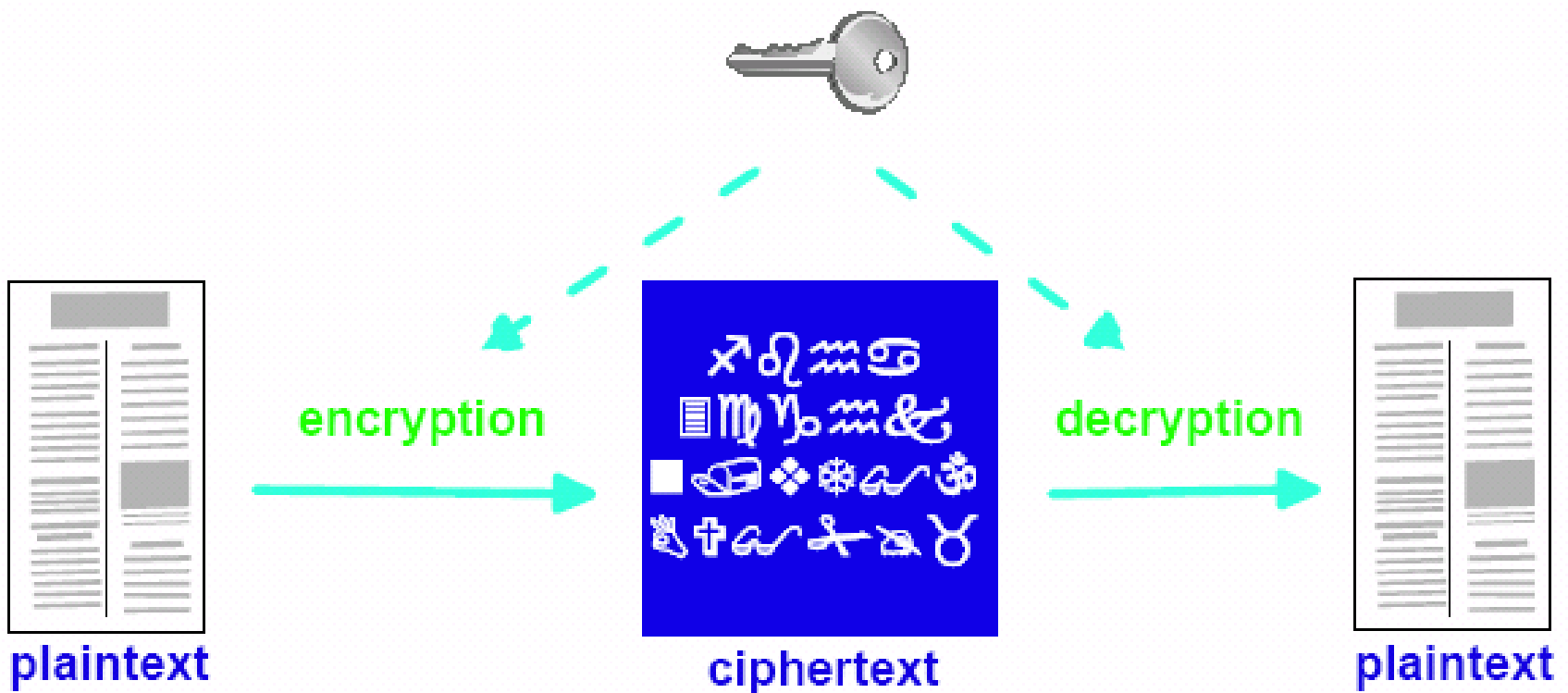


# Cryptography

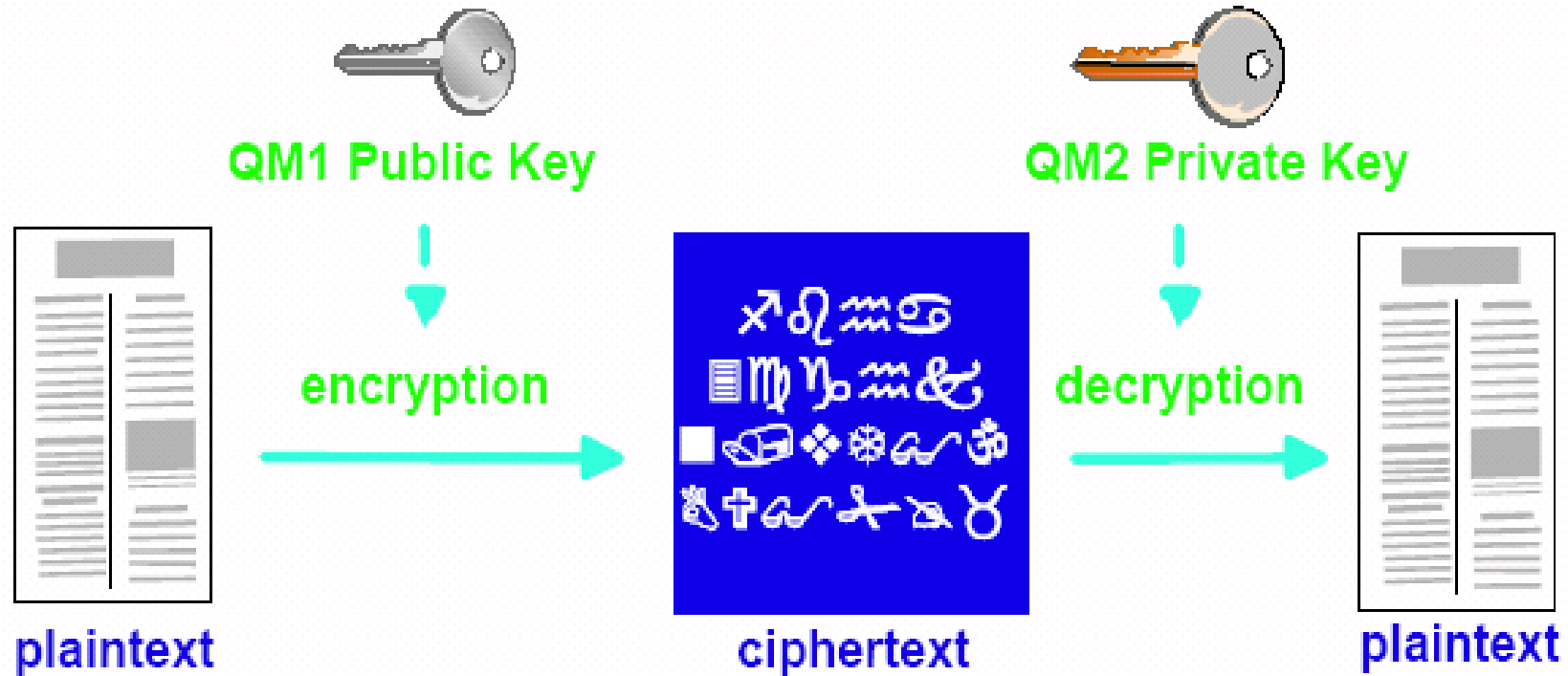
---

- Symmetric Key
- Asymmetric Key
- Hash Function
- Digital Signature
- Digital Certificates

# Symmetric Key - Secret Key



# Asymmetric Key – Public/Private Key Pairs

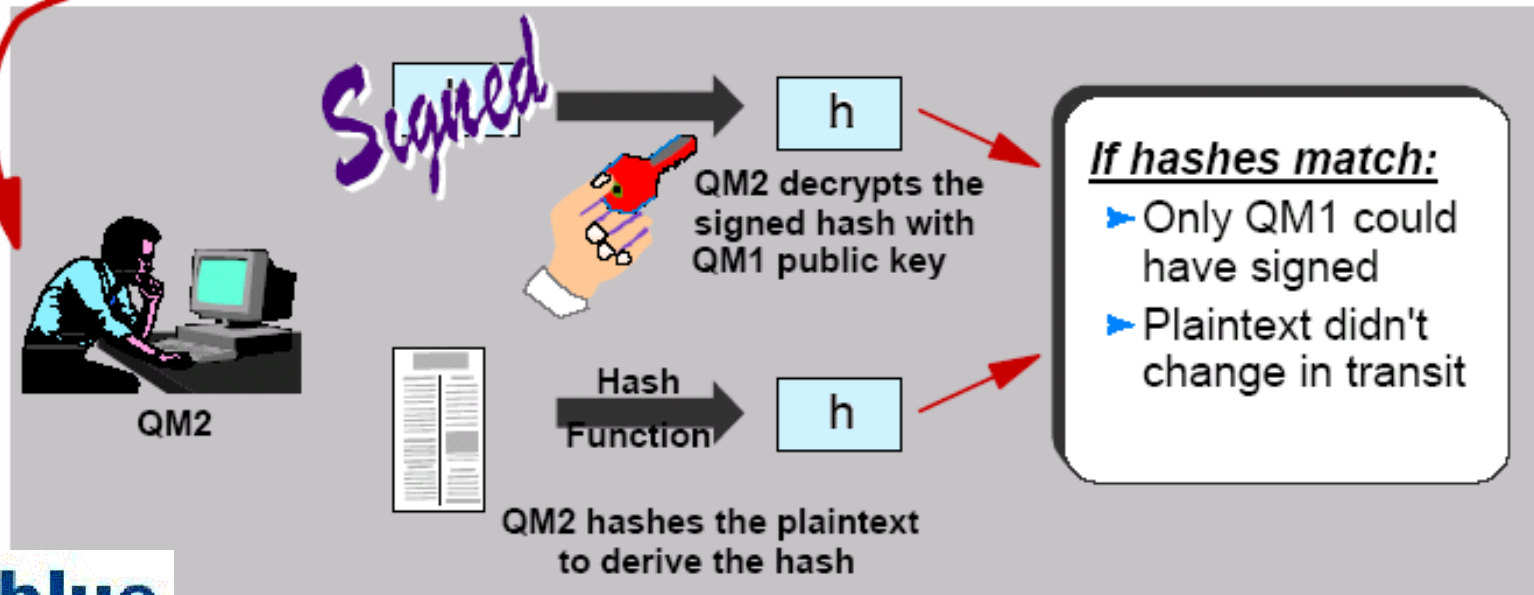
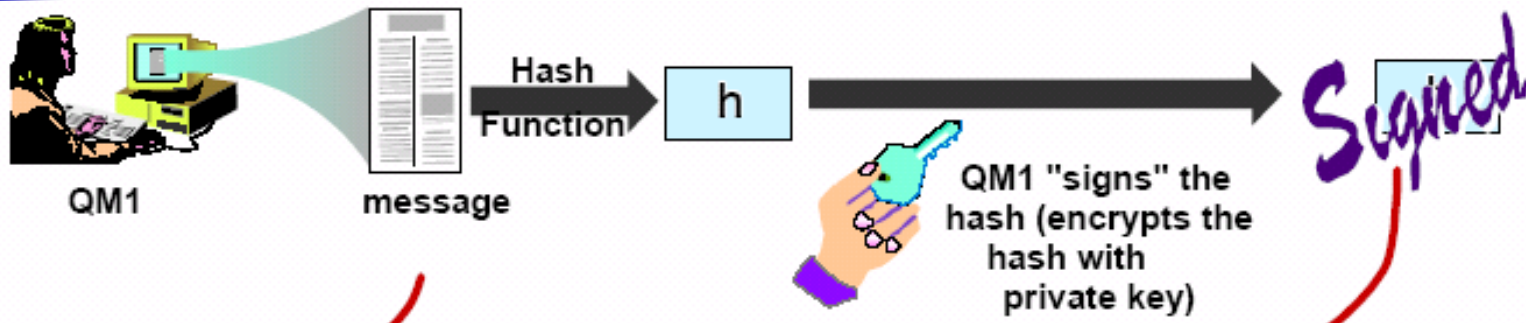


# Hash Function

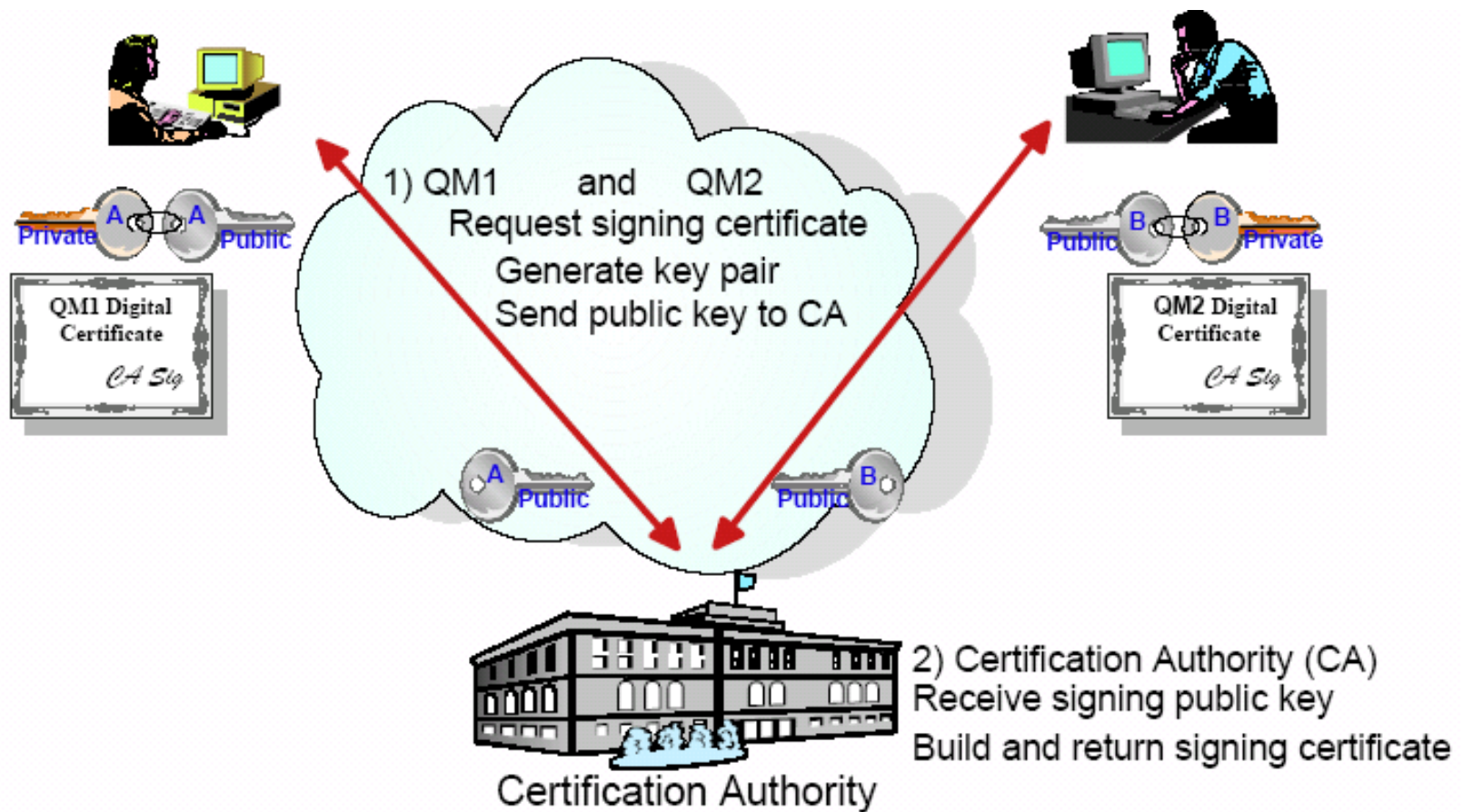


- **Computes the message digest or Message Authentication Code (MAC)**
- **Easy to compute**
- **Very difficult to reverse**
- **It should be computationally infeasible to find two messages that hash to the same thing.**

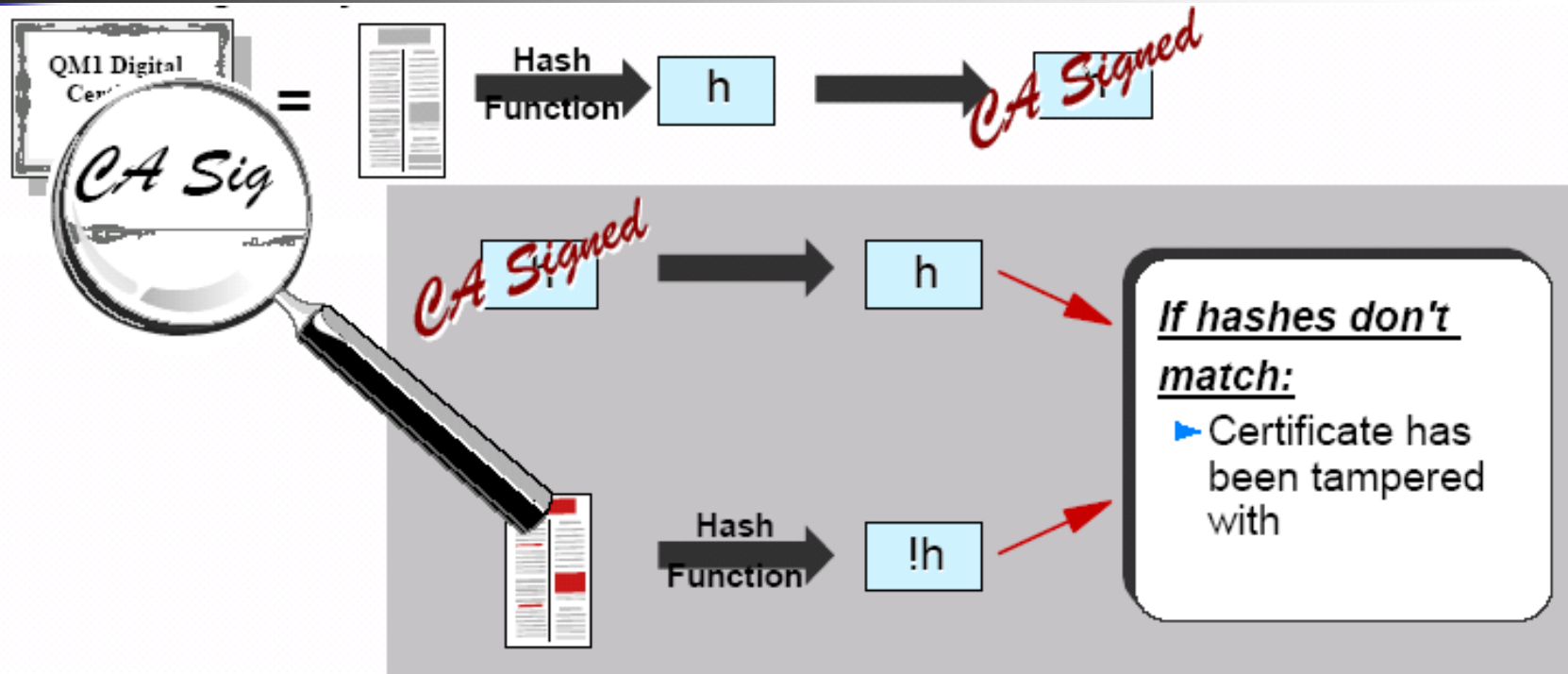
# Digital Signature



# Digital Certificate



# Trusting a Digital Certificate

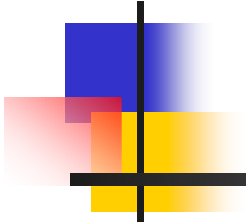


## •Digital Certificate = Plaintext

- Can be subject to tampering
- Signed by CA at creation

## •CA's Digital Signature

- Allows tampering to be detected



# Secure Sockets Layer



# Secure Sockets Layer

---

- **Protocol to allow transmission of secure data over an insecure network**
- **Combines these techniques**
  - **Symmetric / Secret Key encryption**
  - **Asymmetric / Public Key encryption**
  - **Digital Signature**
  - **Digital Certificates**
- **To combat security problems**
  - **Eavesdropping**
    - **Encryption techniques**
  - **Tampering**
    - **Digital Signature**
  - **Impersonation**
    - **Digital Certificates**



# Benefits of SSL

---

- **Provides a protocol for the function we need**
  - **Encryption**
  - **Message Integrity Checking**
  - **Authentication**
- **Supports a range of cryptographic algorithms**
- **Uses Public/Private Keys**
  - **No key distribution problem**
- **Widely accepted in the Internet community**
- **Subjected to significant testing by the hacker community**

# Certificate Revocation

- **What happens if a Certificate is no longer trusted?**

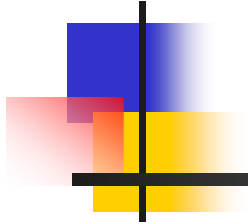


Valid From 01/04/2003  
Valid To 12/10/2003



- **Certification Authority revokes it on a Certificate Revocation List (CRL)**





# SSL & WebSphere MQ



# SSL functions & WebSphere MQ

---

- **Supported**
  - **SSL V3.0**
  - **Ability to authenticate client**
  - **Certificate Revocation Lists on LDAP servers**
- **Not Supported**
  - **List of CipherSpecs, only one must be provided**
  - **SSL session reuse**



# SSL Handshake

---

1. The client's sender channel starts the connection to the server and requests a certificate.
2. The server sends the certificate (which is encrypted using the certificate authorities key).
3. The client verifies the server's digital signature in the certificate. Now the client knows who the server claims to be.
4. (Optional) The server requests a certificate from the client and verifies the client's digital signature in the certificate. Now the server knows who the client claims to be.
5. The handshake continues with the selection of a secret key that both parties will use to sign and/or encrypt the messages.



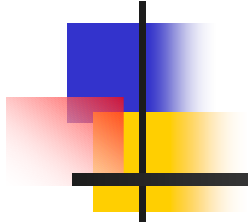
# Creating Certificates

---

- **Create internal certificates for test or intranet applications,**
  - Using RACF panels or RACDCERT commands on z/OS
  - Using IBM GSKit or iKeyMan GUI tool on Unix platforms
  - Using Windows Certificate Services on Windows
  - Using Digital Certificate Manager (DCM) on OS/400
  - **Using OpenSSL – multiplatform**

**OR**

- **Generate a certificate request to be processed by CA**
  - This request is written to a file/data set
  - Send it out to a Certification Authority
  - Receive your signed certificate from the CA



# Creating Certificates with OpenSSL



# About OpenSSL

---

- Provides a command line interface to a large number of certificate related facilities
  - Creation of your own CA
  - Generation and signing of certificates
  - Conversion between various certificate formats.
- Open source product – free to use for commercial and non-commercial
- Support multiple platform
- Certificates created are valid for use on all platforms regardless of where they are generated.
- Website : [www.openssl.org](http://www.openssl.org)



# Creation Self-Signed CA Cert

---

- **Create CA private certificate**
  - `openssl genrsa -rand -des3 -out keys/ca.key 1024`
- **Create self-sign CA certificate – public key**
  - `openssl req -new -x509 -days 1095 -key keys/ca.key -out keys/ca.cer -outform PEM`

Output from command:

Country Name (2 letter code) [AU]:**US**  
State or Province Name (full name) [Some-State]:**NY**  
Locality Name (eg, city) []:**New York**  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:**ISSI**  
Organizational Unit Name (eg, section) []:**IT**  
Common Name (eg, YOUR name) []:**ISSI-CA**  
Email Address []:**vterrone@infinite-blue.com**

**The file ca.cer is self-signed CA public certificate and ca.key is the CA private key that must be protected.**



# Creation Server Certificate

---

- **Create server private key**
  - openssl genrsa -rand -des3 -out QM1/server.key 1024
- **Create request certificate to be signed**
  - openssl req -new -days 1095 -key QM1/server.key -out QM1/server.crs

Output from command:

Country Name (2 letter code) [AU]:**US**

State or Province Name (full name) [Some-State]:**NY**

Locality Name (eg, city) []:**New York**

Organization Name (eg, company) [Internet Widgits Pty Ltd]:**ISSI**

Organizational Unit Name (eg, section) []:**IT**

Common Name (eg, YOUR name) []:**ibmwebspheremqqm1**

Email Address []:**vterrone@infinite-blue.com**

A challenge password []:**pwpass**

An optional company name []:**ISSI**



# Sign Server Certificate

---

- **Sign the server certificate with our CA certificate**
  - `openssl ca -in QM1/server.crs -out QM1/sscert.cer -keyfile keys/ca.key -cert keys/ca.cer -days 1095`

Output from command:

The Subjects Distinguished Name is as follows

```
countryName           :PRINTABLE:'US'
stateOrProvinceName  :PRINTABLE:'NY'
localityName         :PRINTABLE:'New York'
organizationName     :PRINTABLE:'ISSI'
organizationalUnitName :PRINTABLE:'IT'
commonName           :PRINTABLE:'ibmwebspheremqqm1'
emailAddress         :IA5STRING:'vterrone@infinite-blue.com'
```

Certificate is to be certified until May 26 2004 GMT (1095 days)  
Sign the certificate? [y/n]:**y**



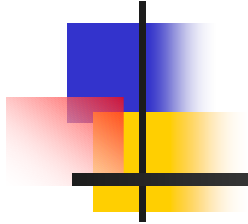
# Export Server Certificate

---

- **Export certificate in PKCS#12 format.**
  - `openssl pkcs12 -export -in QM1/sscert.cer -out QM1/sscert.p12 -inkey QM1/server.key -name ibmwebspheremqqm1`

PKCS#12 portable format for storing or transporting a user's private keys and certificates. Private and public certificates are stored in a password protected file.

**Note the file `sscert.p12` is the PKCS #12 file that will be imported into WMQ key repository.**



# Storing Certificates for WMQ



# Storing Certificates

---

- **Key database files on UNIX platforms and OS/400. IBM GSKit is used on distributed platforms to install certificates.**
- **Keyrings in RACF, ACF2 or TopSecret on z/OS. Using RACF panels or RACDCERT are used to install certificates.**
- **Certificate Stores on Windows. On Windows 2000 and above certificates are maintained in Internet Options under control panel. Key ring is created and maintained by WMQ.**
  - **Private keys are stored in the registry on Windows.**

# Queue Manager's Key Repository

- **Queue Manager Digital Certificate label**

- **ibmWebSphereMQ<QMgr Name>**
  - (mixed case) label on z/OS
- **ibmwebspheremq<qmgr name>**
  - (lower case) label on UNIX & OS/400
- Selected from a GUI on Windows

- **Digital Certificates from various Certification Authorities or your own CA.**





# Storing Certificates under Unix

---

- **Create key repository:**
  - `gsk6cmd -keydb -create -db QM1/key.kdb -pw pwdb -type cms -expire 2048 -stash`
- **Stash the password for the key repository**
  - `gsk6cmd -keydb -stashpw -db QM1/key.kdb -pw pwdb`
- **Add the self-sign CA public certificate to repository**
  - `gsk6cmd -cert -add -db QM1/key.kdb -pw pwdb -label ISSI-CA -file keys/ca.cer -format ascii`
- **Add the server certificate to the repository**
  - `gsk6cmd -cert -import -file QM1/sscert.p12 -target QM1/key.kdb -target_pw pwdb -type pkcs12`

Note: All files named key.\* and copy them to  
`/var/mqm/qmgrs/<QMgr Name>/ssl`



# Storing Certificates under zOS

---

- **Add the Certificate Authority to the supplied RACF Certificate Authority List. Note the value for ADD is the dataset the transferred CA cert is stored.**
  - `RACDCERT CERTAUTH ADD( 'TESTCERT.CA.CER' ) +  
WITHLABEL( 'ISSI-CA' ) TRUST`
- **Add the Server Certificate to RACF. Note that the value for ADD is the dataset that the transferred server cert is stored.**
  - `RACDCERT ID(QM1CHIN)+  
ADD( 'TESTCERT.SRVCERT.P12' ) +  
WITHLABEL( 'ibmWebSphereMQQM1' ) TRUST +  
PASSWORD( 'pwpass' )`
- **Create key ring need WMQ in RACF.**
  - `RACDCERT ID(QM1CHIN) ADDRING( 'QM1RING' )`



# Storing Certificates under zOS

---

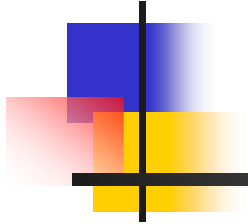
- **Connect the Certificate Authority to the key ring.**
  - RACDCERT ID(QM1CHIN) +  
CONNECT(CERTAUTH LABEL('ISSI-CA') +  
RING(QM1RING) USAGE(CERTAUTH))
- **Connect the Server Certificate to the key ring.**
  - RACDCERT ID(QM1CHIN) +  
CONNECT(ID(QM1CHIN)+  
LABEL('ibmWebSphereMQQM1') +  
RING(QM1RING) USAGE(PERSONAL))



# Storing Certificates under Windows

---

- **Installation of CA certificate:**
  - Double click on CA cert ca.cer
  - When box appears, click on Install Certificate. Follow wizard and when you get to Certificate Store be sure to place certificate in Trusted Root Certificate Authorities.
- **Installation of Server certificate:**
  - Double click on file sscert.p12
  - A wizard will start and it will ask for the password that was used when the certificate was generated. Enter that password and then check the box 'Mark the Private Key as Exportable'.



# Websphere MQ Configuration



# SSLKEYR

---

- **Install Certificate on z/OS – Option on Queue Manager**

```
ALTER QMGR SSLKEYR(QM1RING)
```

- **Install Certificate on Unix, OS/400**

```
ALTER QMGR SLKEYR('/var/mqm/qmgrs/QM1/ssl/key')
```

- **Install Certificate on Windows**

```
ALTER QMGR SLKEYR('C:\IBM\WebSphere MQ\qmgrs\QM1\ssl\key')
```

- **Note that the extension on the key database is not specified but assumed to be kdb on Unix and sto for Windows.**



# SSL Task

---

- **On z/OS only**
  - **The number of server subtasks to use for processing SSL calls.**
  - **Used to run SSL handshake and encryption calls**
  - **At least 2 required to run any SSL channels**
  - **Value of zero and SSL channels will not start.**

**ALTER QMGR SSLTASKS(8)**



# SSLCIPH

---

- **Only mandatory parameter on an SSL channel**
  - Without it channel is assumed not to be using SSL
- **Specify the CipherSpec to be used**
  - Both ends of the channel must specify the same CipherSpec
- **From a list of human-readable strings**
  - e.g. NULL\_SHA
  - TRIPLE\_DES\_SHA\_US
- **z/OS, Windows, OS/400: also SSL API numeric values**
  - Allow support of new CipherSpecs without updates to MQ Code

**SSLCIPH(RC4\_MD5\_US)**

**or**

**SSLCIPH(04)**



# SSLPEER

---

- **Specify the partner's Distinguished Name**
- **Can use wildcards**
- **Multiple Organizational Unit (OU)**
  - **Must be matched in order**

```
SSLPEER('CN="ibmwebspehremqqm1", O=ISSI')  
or  
SSLPEER('OU=IT, O=ISSI')
```



# SSLCAUTH

---

- **Client Authentication**
  - Request whether the client end is required to provide a certificate for authentication to the server.

**SSLCAUTH(REQUIRED)**  
or  
**SSLCAUTH(OPTIONAL)**

# Security Problems / Solutions

~~Security Problems~~

Solutions

Using WebSphere MQ

- **Eavesdropping**

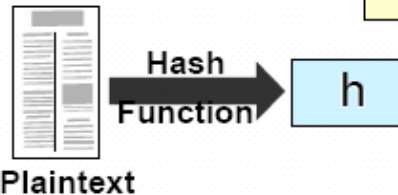
- Symmetric Key Cryptography



SSLCIPH(RC4\_MD5\_US)

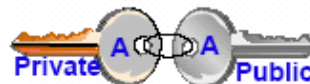
- **Tampering**

- Hash Function

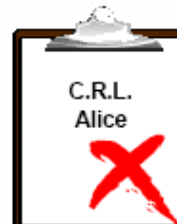


- **Impersonation**

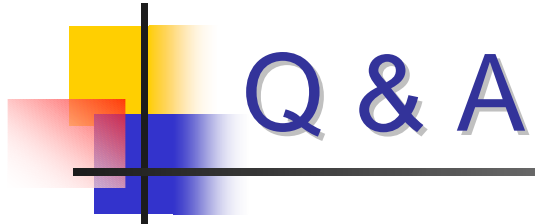
- Digital Certificates
- Asymmetric Keys
- CRL checking



SSLKEYR(QM1KEYRING)  
SSLPEER('O=IBM')  
SSLCAUTH(REQUIRED)



SSLCRLNL(LDAPNL)



Q & A

---

# Questions and Answers



In Closing

---

**Thank You  
for attending this  
portion of  
Today's Session**



# About ISSI, Inc.

---

- √ **Infinity Systems** is dedicated to meeting the needs of its clients. We provide our customers with high quality, cost effective Information Technology Services and help organizations use new technologies to attain their business goals.
- √ Our technical expertise, commitment to customer service and corporate integrity are undoubtedly why many of the nation's finest corporations look to us for their technical contracting needs.
- √ We believe that business success is based on building long lasting relationships. Our service minded approach to technical staffing has enabled us to enjoy long term business relationships with a wide ranging client base, diversified in size, geography and industry.



# Contact us @

---

Contact us at 1-646-405-9300

e-mail us at [info@infinite-blue.com](mailto:info@infinite-blue.com)

visit us at [www.infinite-blue.com](http://www.infinite-blue.com)



**infiniteblue**



## Special Thanks

---

We would like to acknowledge and thank Mr. Morag Hughson, at IBM Hursley, U.K. for providing additional content that was incorporated into today's presentation.

Today's presentation and Morag Hughson's complete presentation will be made available at the following sites [www.nynjmq.org](http://www.nynjmq.org) and [www.infinite-blue.com](http://www.infinite-blue.com)