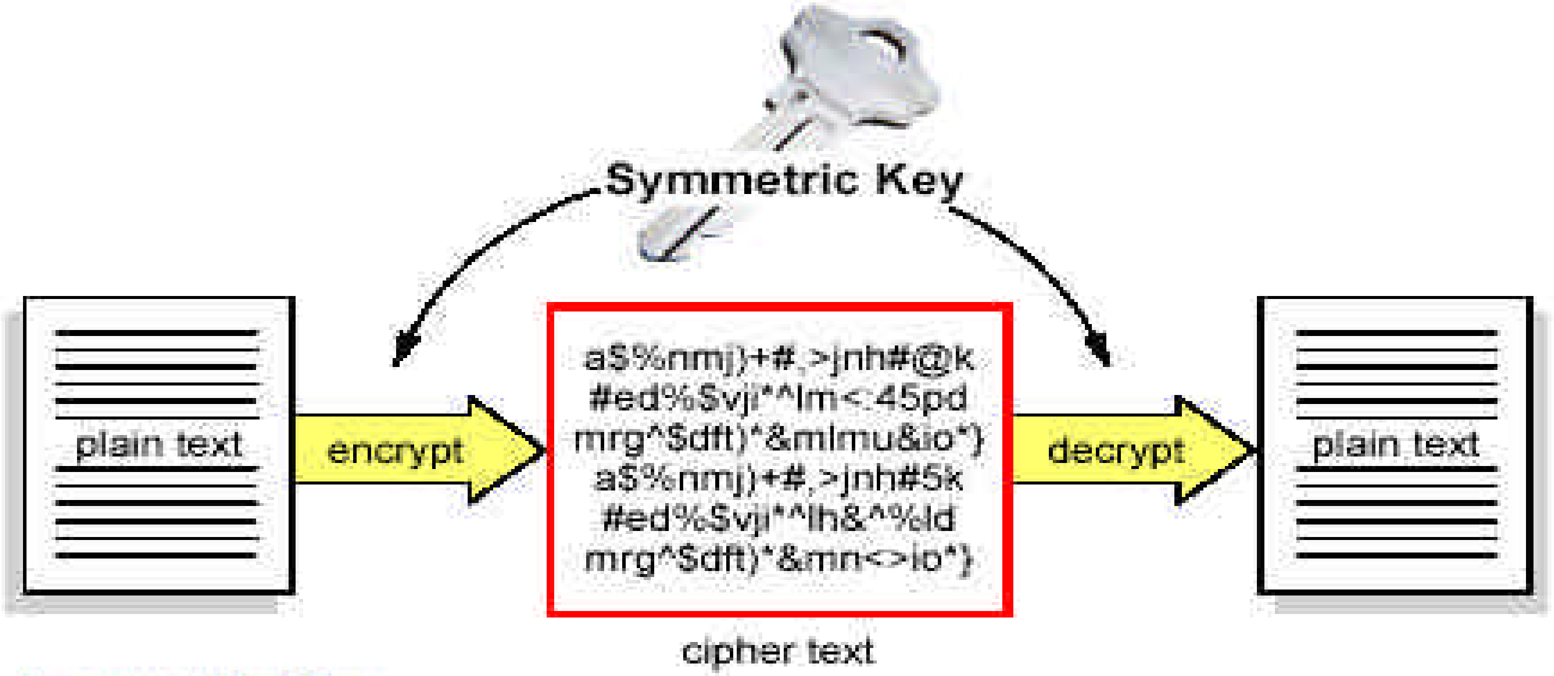
IBM Software Group

SSL Basics

Russ Stancliffe

WebSphere® Support Technical Exchange

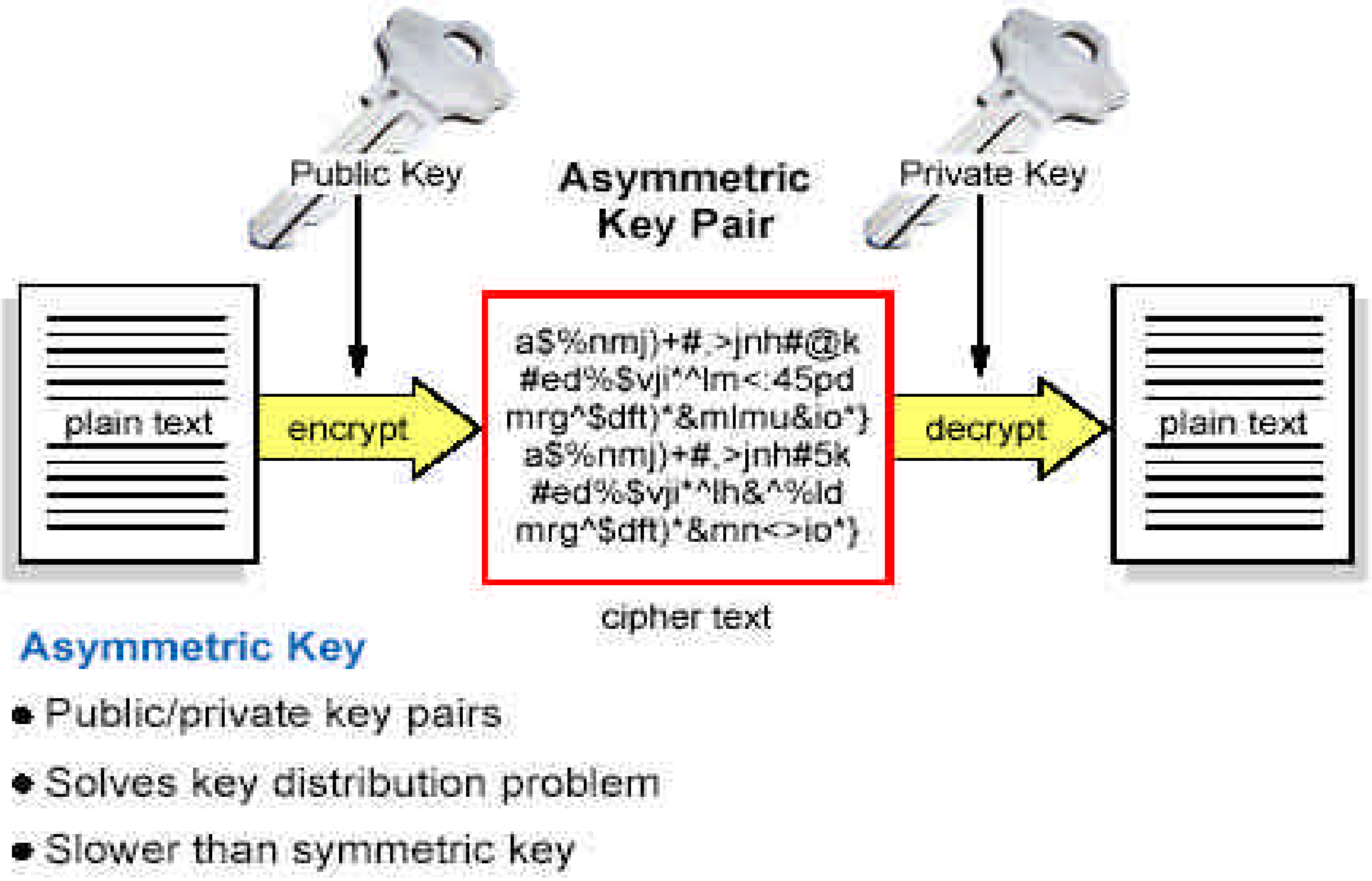
Symmetric



Symmetric Key

- Relatively fast
- Both sender and receiver use the same key
- Key distribution problem

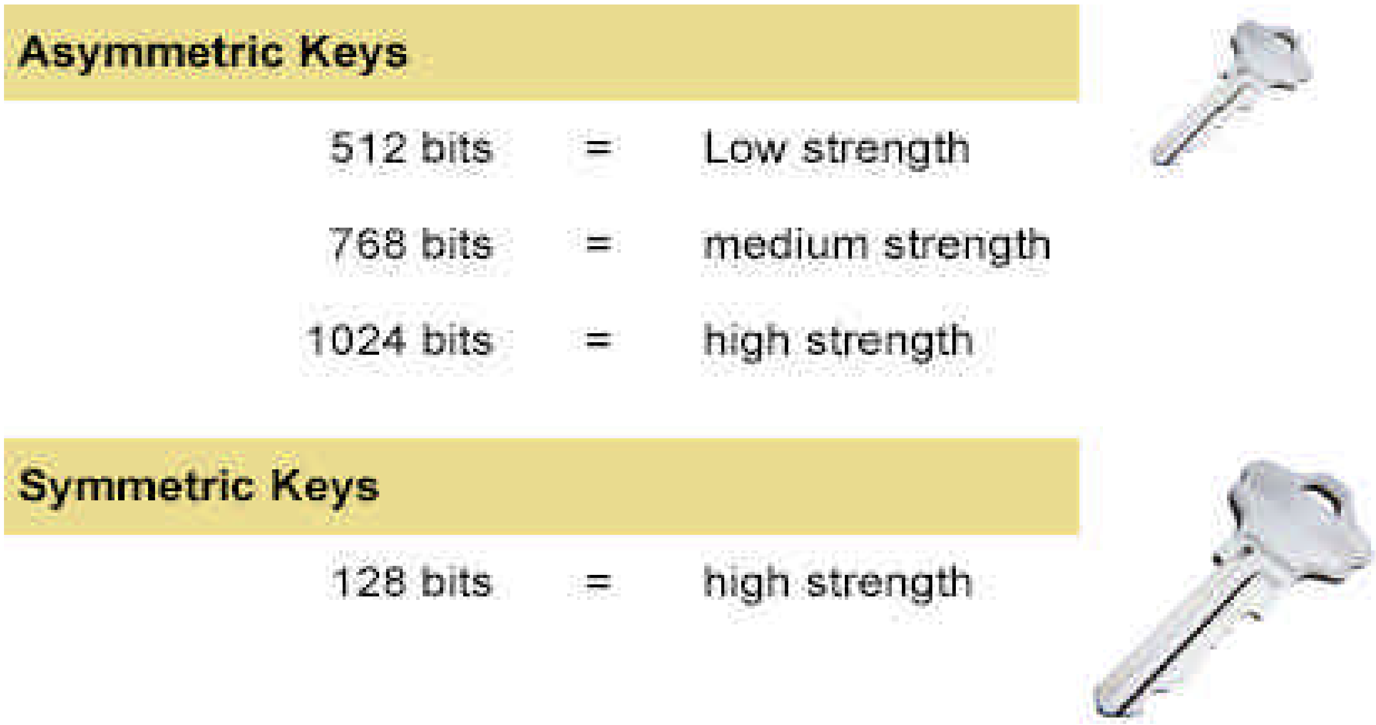
Asymmetric



Asymmetric Key

- Public/private key pairs
- Solves key distribution problem
- Slower than symmetric key

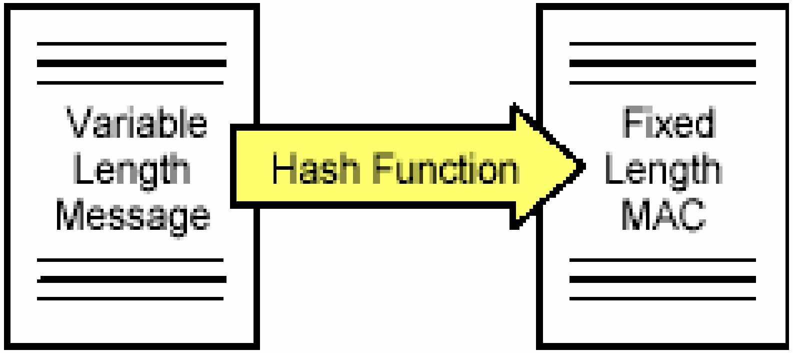
Keys



Asymmetric Keys		
512 bits	=	Low strength
768 bits	=	medium strength
1024 bits	=	high strength

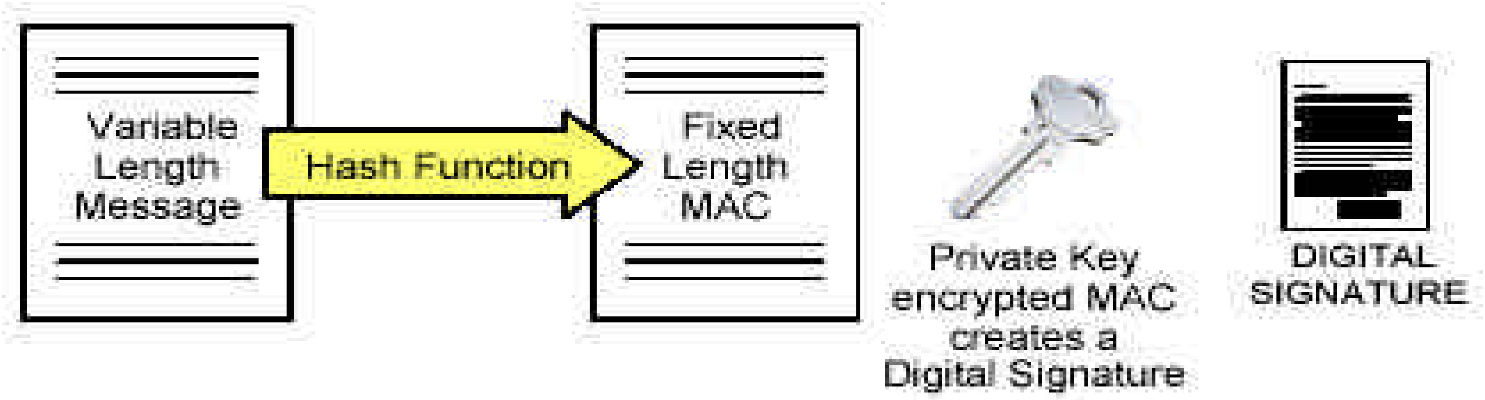
Symmetric Keys		
128 bits	=	high strength

Message Digests



- Hash function computes the Message Digest or Message Authentication Code (MAC)
- Easy to compute
- Very difficult to reverse
- MAC is sent with the message to expose tampering

Digital Signatures



Variable Length Message

Hash Function

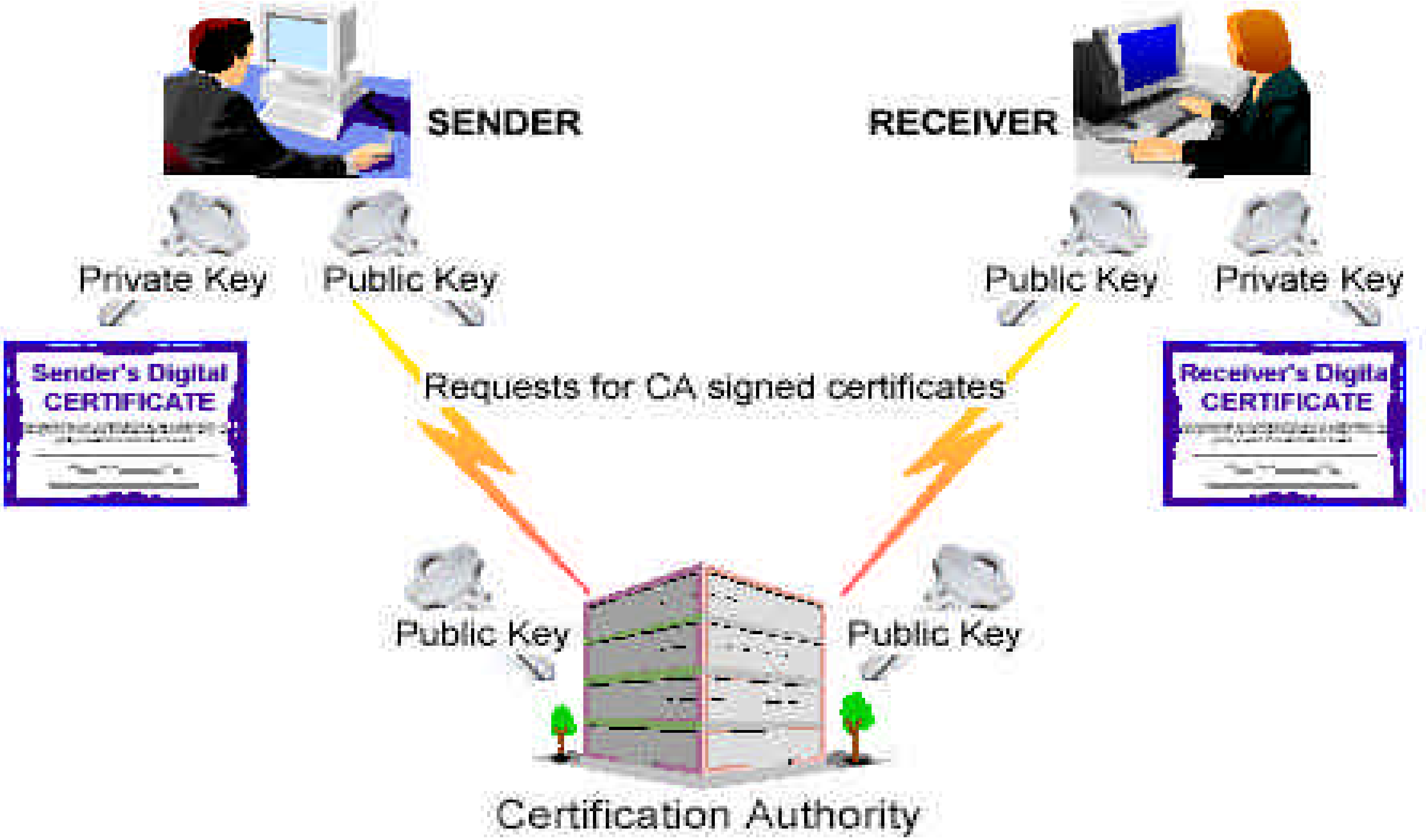
Fixed Length MAC

Private Key encrypted MAC creates a Digital Signature

DIGITAL SIGNATURE

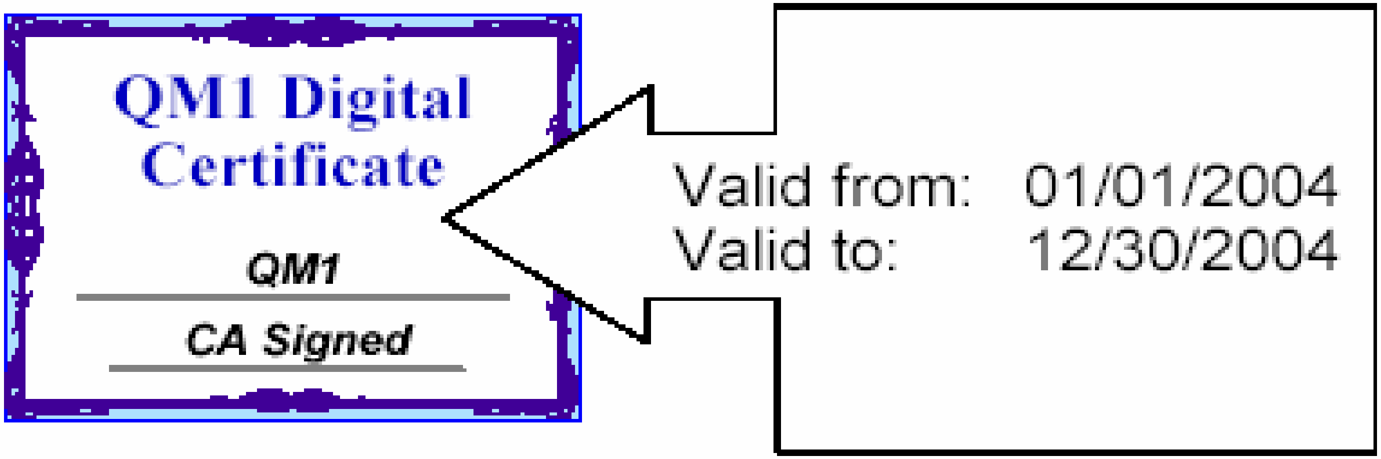
- Sender creates digital signature with private key
- Sends digital signature with the message
- Receiver decrypts the MAC with the sender's public key
- Receiver recomputes the MAC from the message received and verifies both MACs are the same
- If they match, then sender is verified and the message was not tampered with

Digital Certificates



Certificate Revocation Lists

- What happens if a Certificate is no longer trusted?



QM1 Digital Certificate

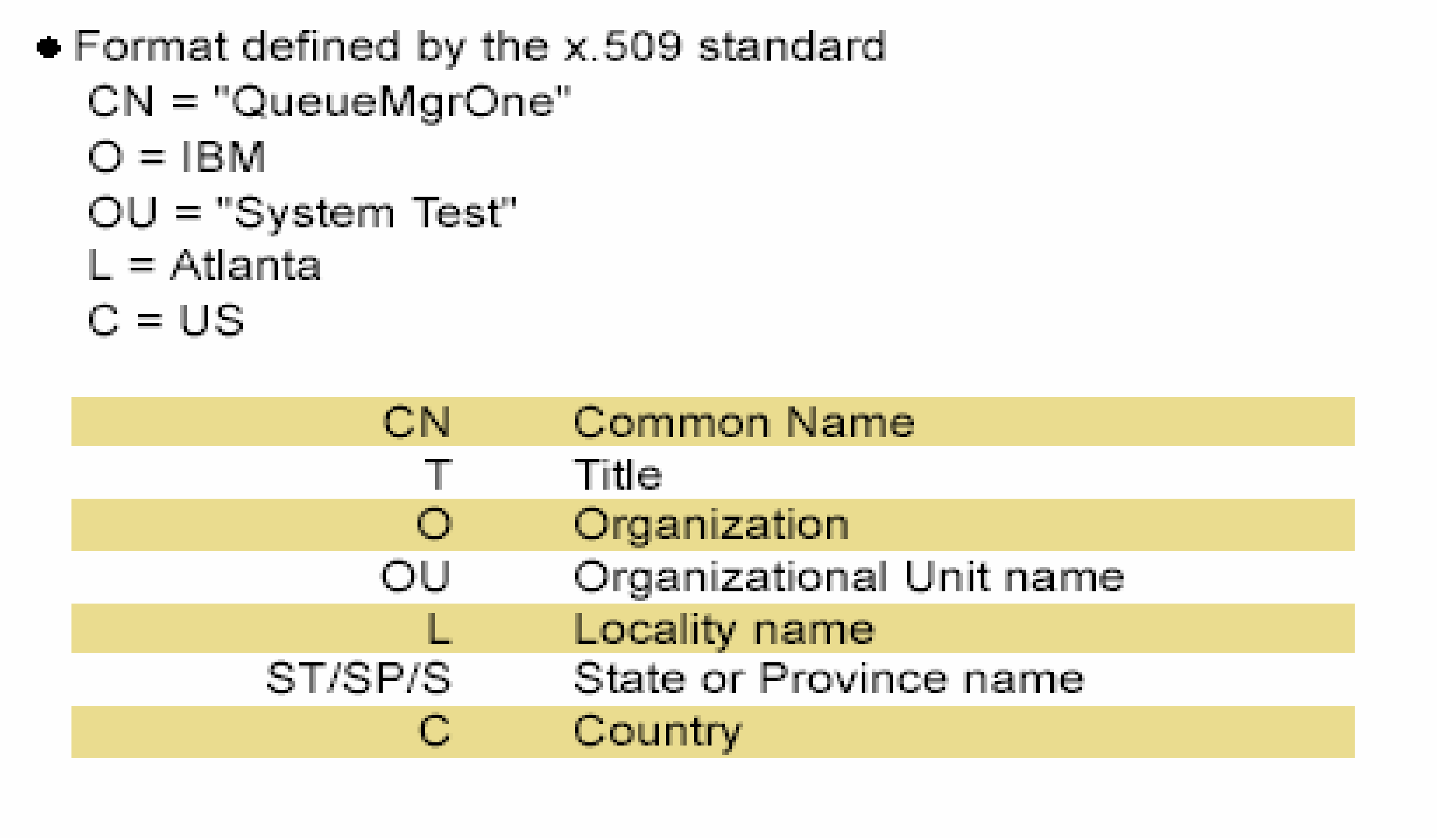
QM1

CA Signed

Valid from: 01/01/2004
Valid to: 12/30/2004

- Certification Authority revokes it on Certificate Revocation List (CRL)
- Checking CRL is optional

Distinguished Name

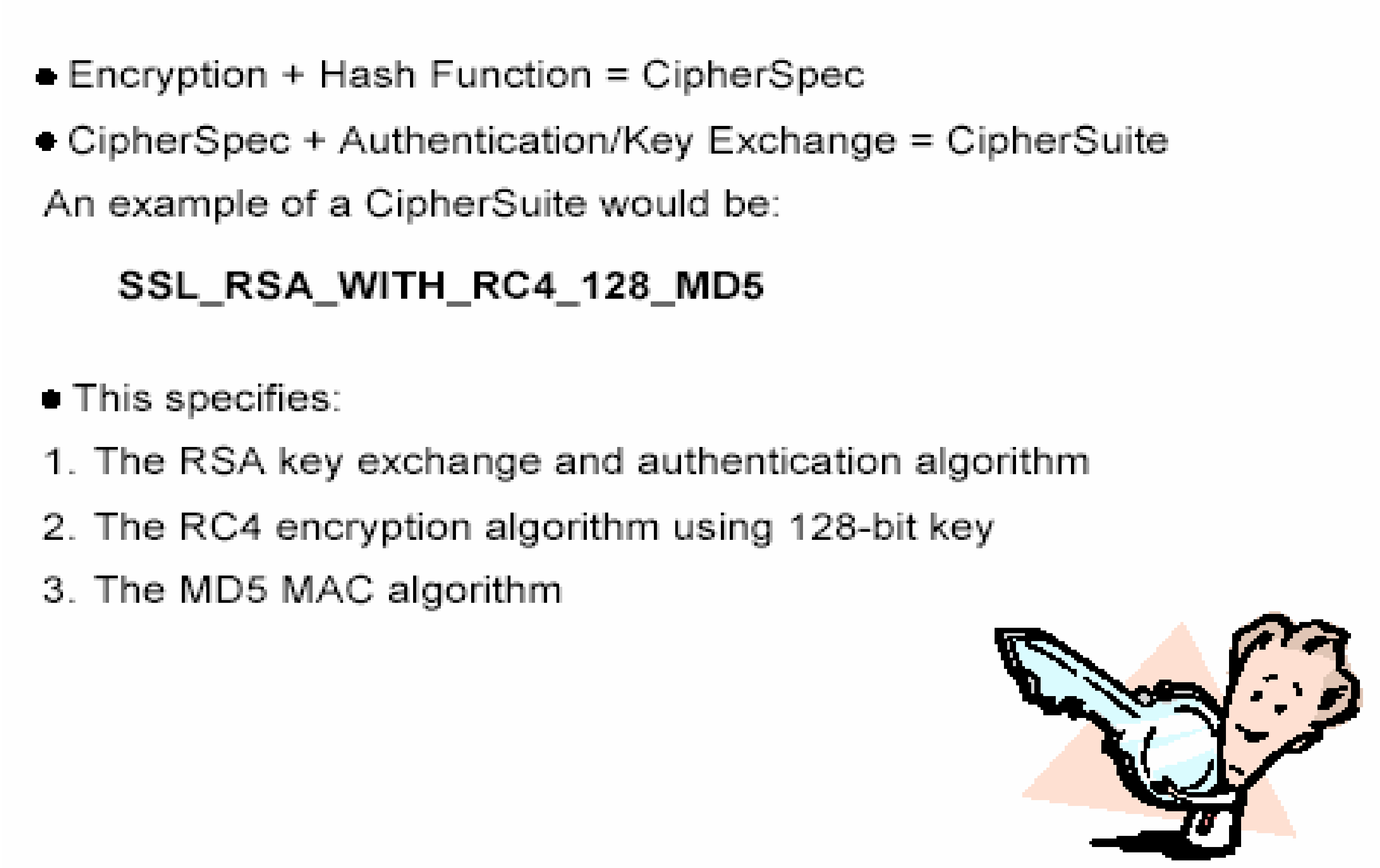


- Format defined by the x.509 standard

  CN = "QueueMgrOne"
  O = IBM
  OU = "System Test"
  L = Atlanta
  C = US

CN	Common Name
T	Title
O	Organization
OU	Organizational Unit name
L	Locality name
ST/SP/S	State or Province name
C	Country

SSL Terms



- Encryption + Hash Function = CipherSpec
- CipherSpec + Authentication/Key Exchange = CipherSuite

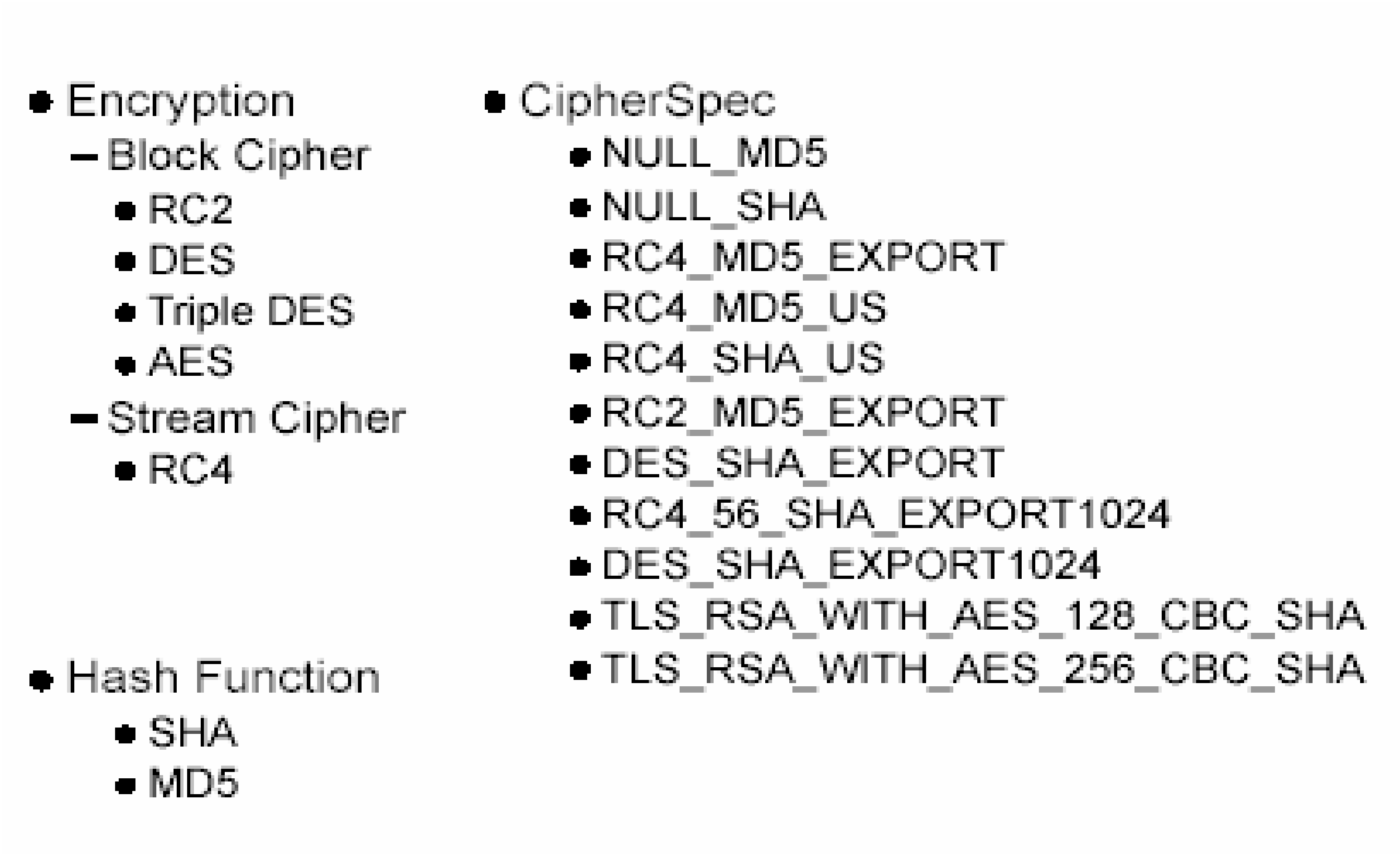
An example of a CipherSuite would be:

SSL_RSA_WITH_RC4_128_MD5

- This specifies:

1. The RSA key exchange and authentication algorithm
2. The RC4 encryption algorithm using 128-bit key
3. The MD5 MAC algorithm

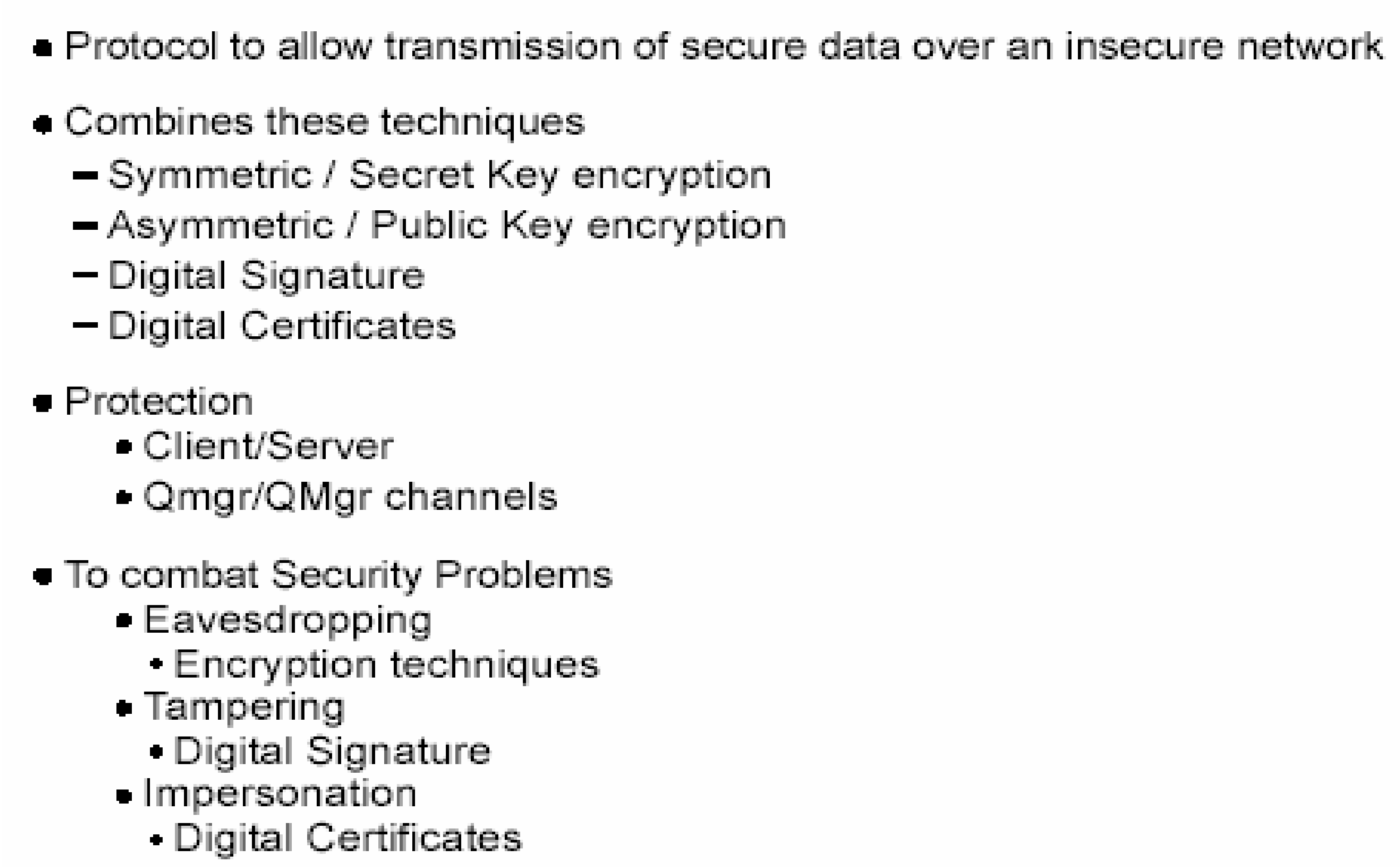
CipherSpecs



- Encryption
  - Block Cipher
    - RC2
    - DES
    - Triple DES
    - AES
  - Stream Cipher
    - RC4
- Hash Function
    - SHA
    - MD5

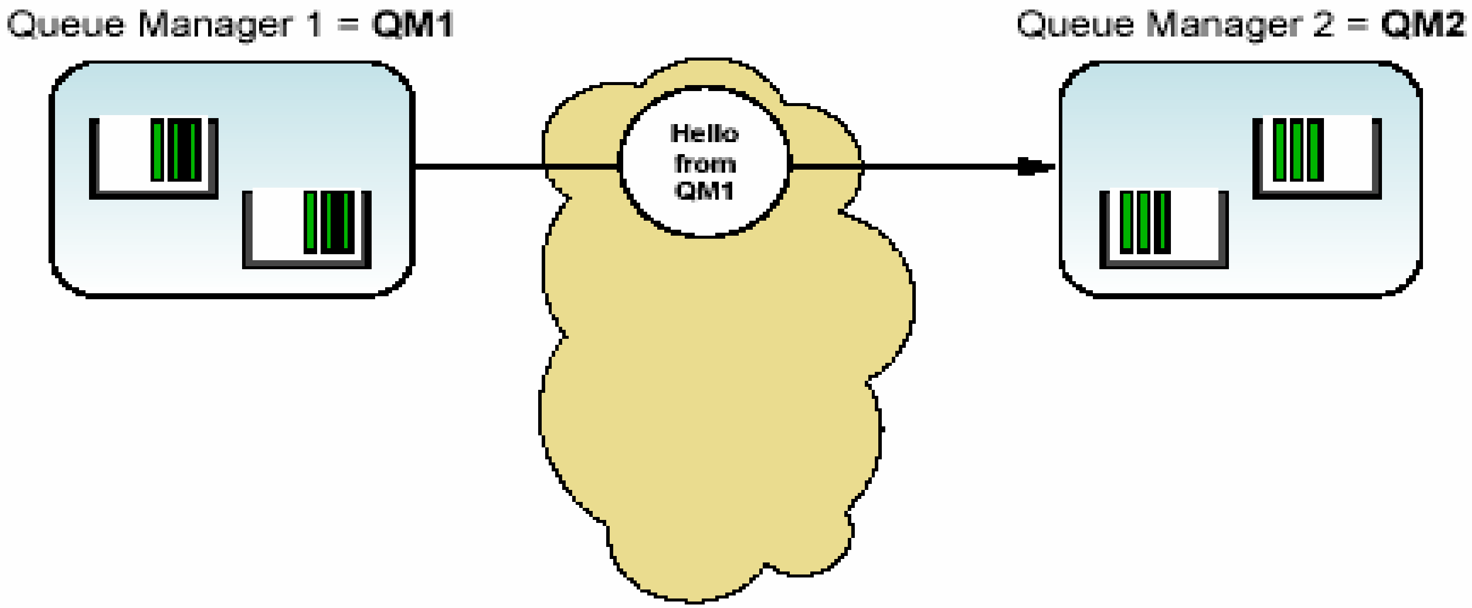
- CipherSpec
    - NULL_MD5
    - NULL_SHA
    - RC4_MD5_EXPORT
    - RC4_MD5_US
    - RC4_SHA_US
    - RC2_MD5_EXPORT
    - DES_SHA_EXPORT
    - RC4_56_SHA_EXPORT1024
    - DES_SHA_EXPORT1024
    - TLS_RSA_WITH_AES_128_CBC_SHA
    - TLS_RSA_WITH_AES_256_CBC_SHA

Secure Sockets Layer



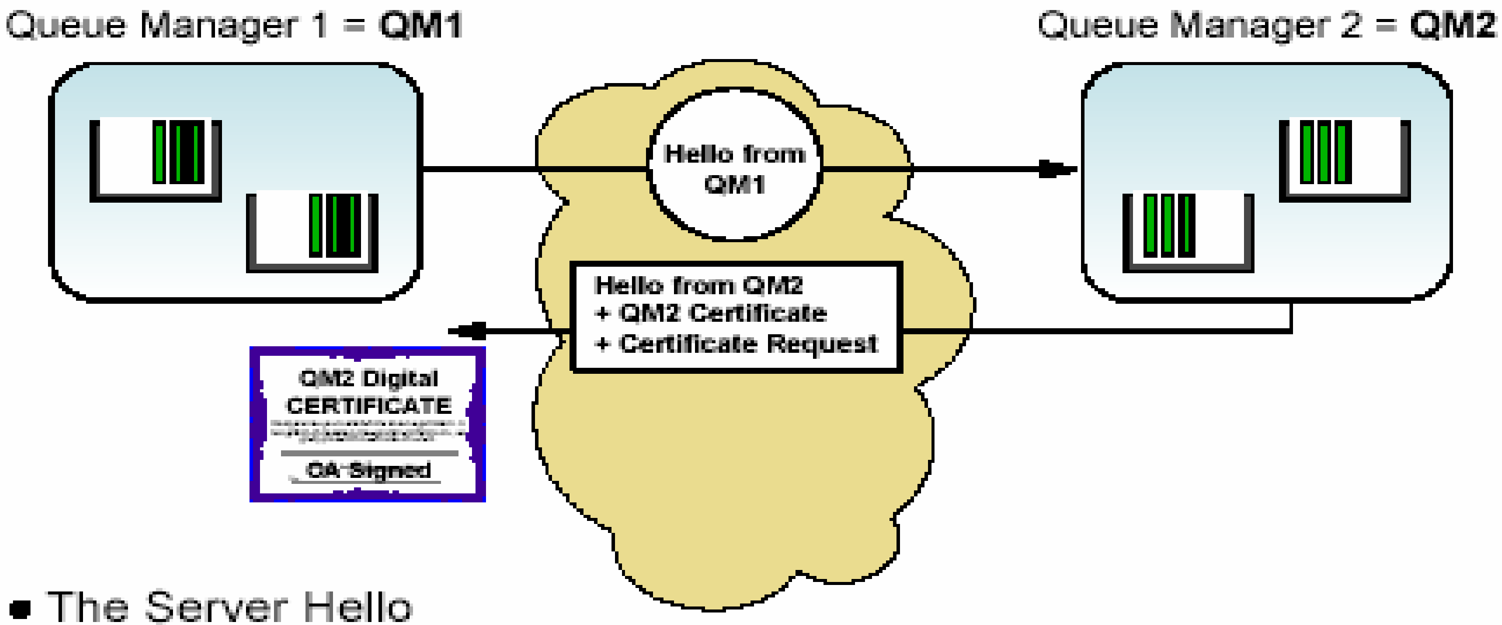
- Protocol to allow transmission of secure data over an insecure network
- Combines these techniques
  - Symmetric / Secret Key encryption
  - Asymmetric / Public Key encryption
  - Digital Signature
  - Digital Certificates
- Protection
  - Client/Server
  - Qmgr/QMgr channels
- To combat Security Problems
  - Eavesdropping
    - Encryption techniques
  - Tampering
    - Digital Signature
  - Impersonation
    - Digital Certificates

SSL Handshake (1 of 6)



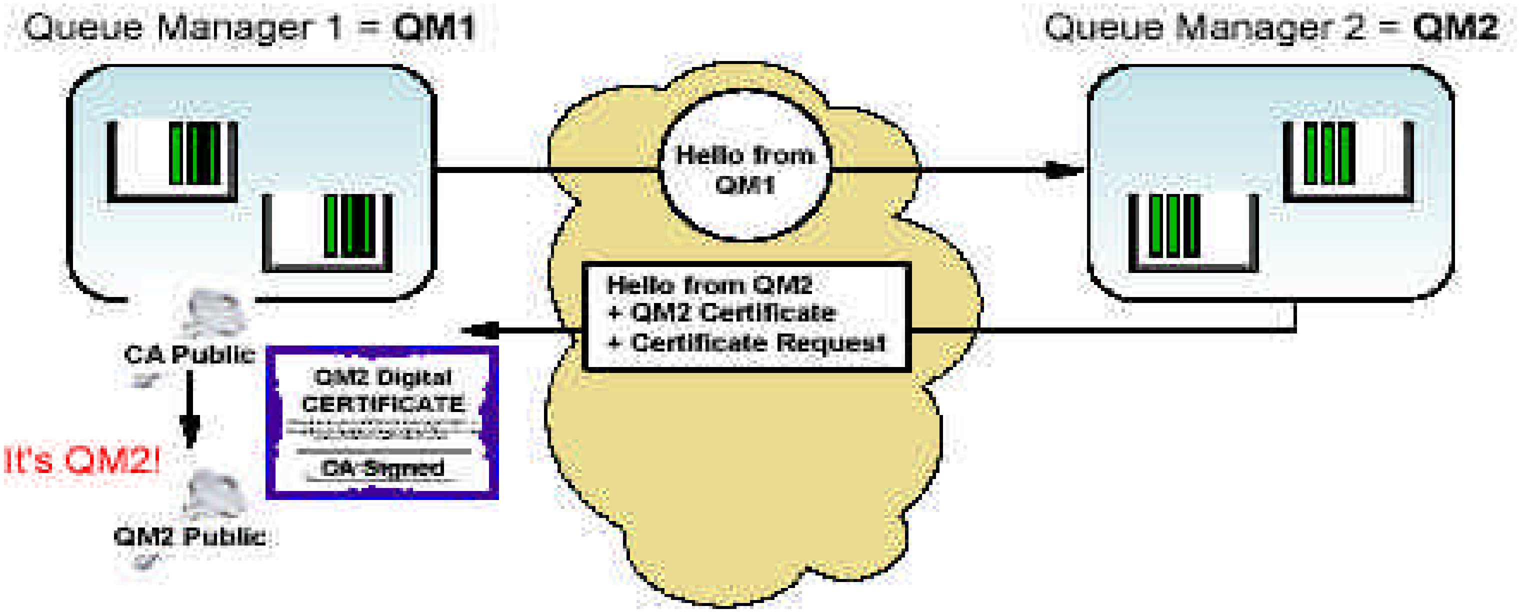
- The Client Hello
  - QM1 sends QM2 some random text
  - Also sends what CipherSpecs and compression methods it can use
  - QM1 is the client

SSL Handshake (2 of 6)



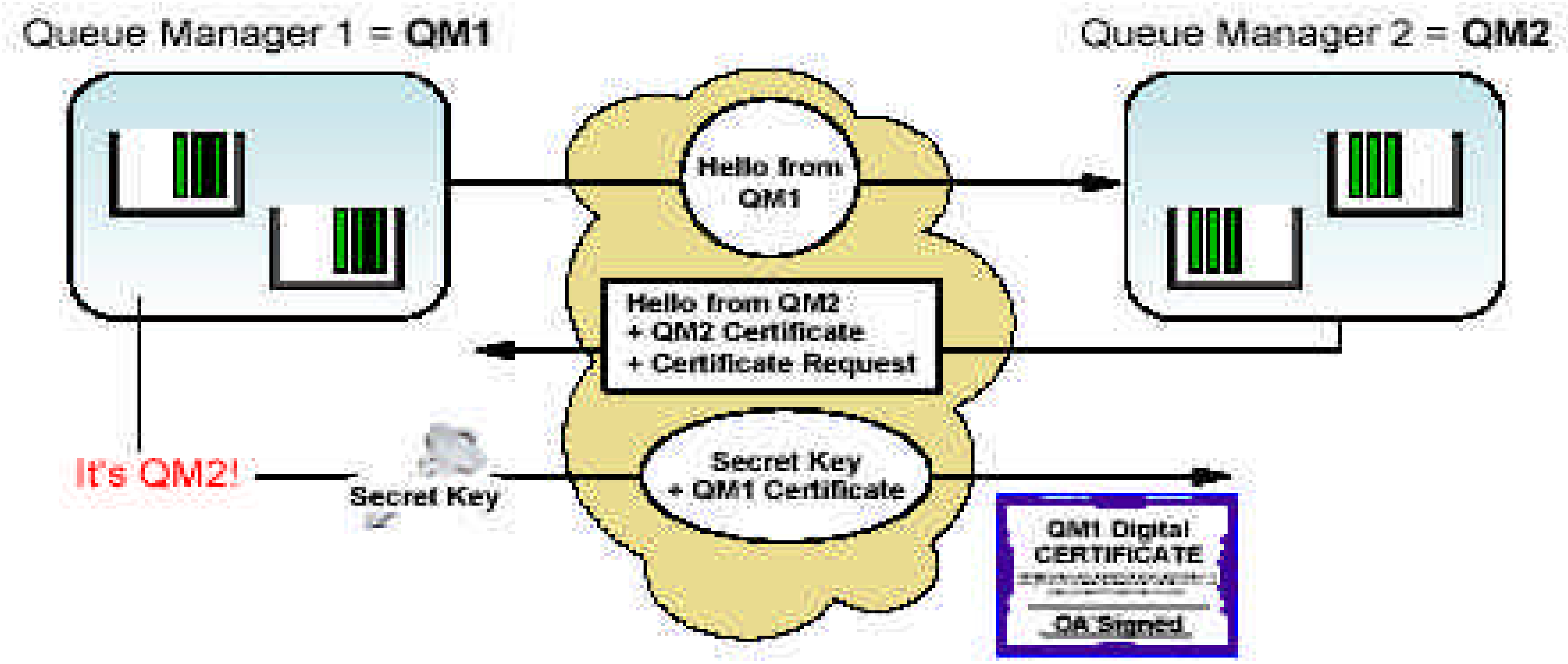
- The Server Hello
  - QM2 sends QM1 some random text
  - QM2 chooses the CipherSpec and compression method to be used, from QM1's list
  - The Server Certificate
  - The Client Certificate Request

SSL Handshake (3 of 6)



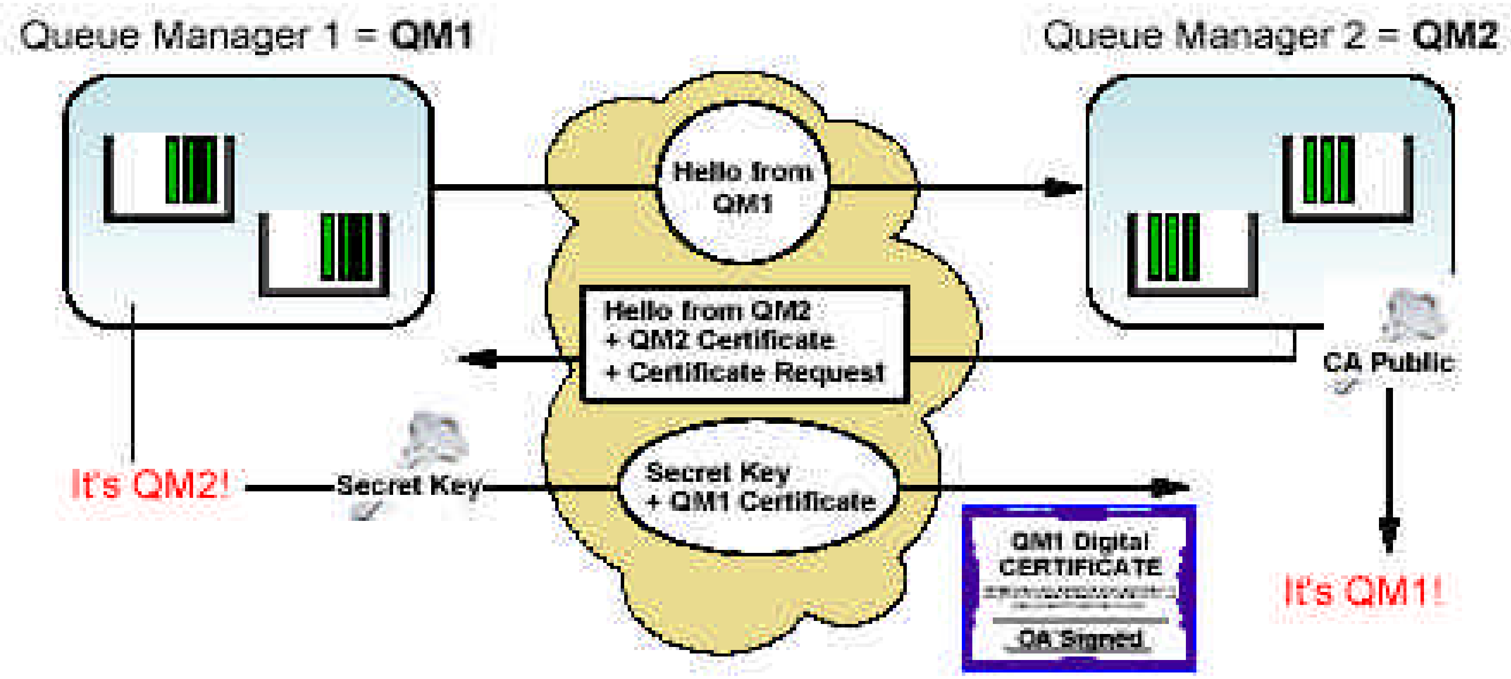
- Verify Server Certificate
  - Check validity period
  - Decrypt using CA's Public Key - Verifies that CA is trusted
  - Check Domain Name and/or Distinguished Name
  - Also receives QM2's Public Key

SSL Handshake (4 of 6)



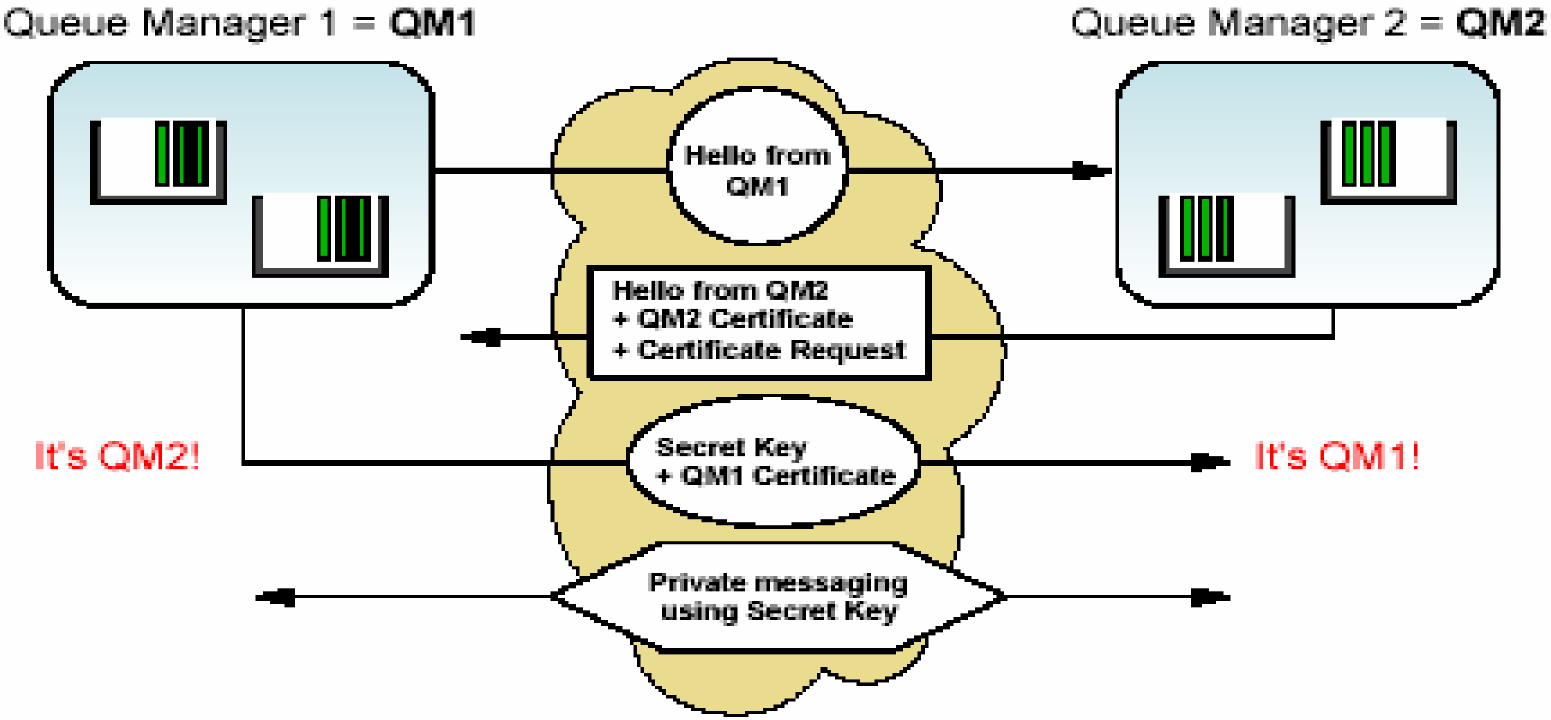
- Client Key Exchange
  - QM1 sends QM2 the Secret Key to use
  - This is encrypted with QM2's Public Key
  - Also sends QM1's Certificate

SSL Handshake (5 of 6)



- Verify Client Certificate
- Decrypt using CA's Public Key

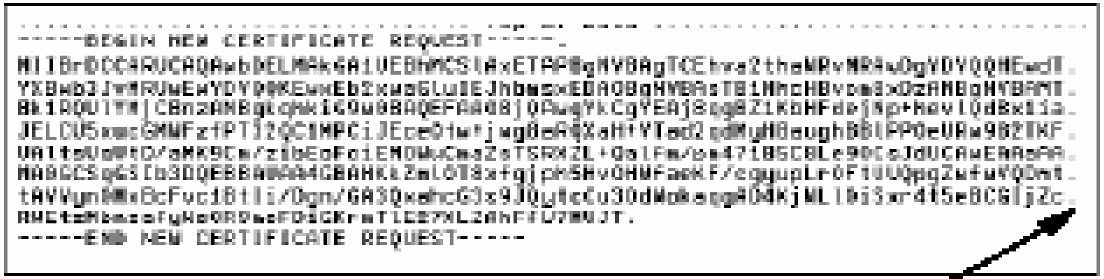
SSL Handshake (6 of 6)



- Send information using agreed Secret Key
  - Randomly generated one-time key
- This is now a secure line

FTP Problems

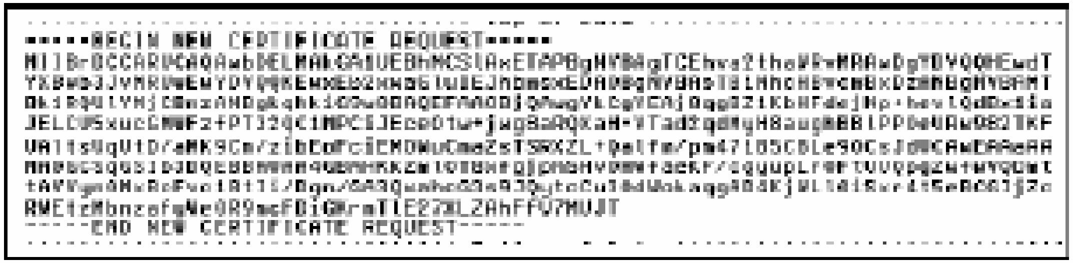
Bad Certificate Request



```
-----BEGIN NEW CERTIFICATE REQUEST-----.
MIIBr0CCAhUCAQAwbDELMAkGA1UEBhMCSlAxETAPBgNVBAgTCEhva2thaWRvMRAwDgYDVQQHEwdT.
YXBwb3JvMRUwEwYDVQQKEwxEb2xwa6LuIEJhbmsxEDAOBgNVBAsTB1MhcHBvcm3xDzANBgNVBAMT.
Bk1AQU1TMjCBnzANBgkqhkiG9w0BAQEFAA0BjQAwgYkCgYEAj8qgBZ1KbHFdejNp+hevlQdBxiia.
JELCU5xucGMWFzfPTJ2QC1MPCiJEceO1w+jwg0eRQXaH+YTad2gdMyH8aughBBlPPOeURw982TKF.
UA1tsUqUtD/aMK9Cm/zibEoFciEMDWuCmaZsTSRXZL+Qa1Fm/pm47185CBLe90CsJdUCAwEAAaAA.
MA0GCSqGSIb3DQEBBAUAA4GBAHKkZmLOT8xfgjpn5HvOHWFaeKF/egyupLr0F1UUQpgZufuYQDmt.
tAVVyn4MxBcFvc1Bt1i/Dgn/GA3QxehcG3x9J0ytcCu30dWokagg404KjWL1Ai3xr415eBCG1jZc.
RWEtzMbnzafyNe9R9mcFDiGKrmT1E27XL2AhFfU7MUJT.
-----END NEW CERTIFICATE REQUEST-----
```

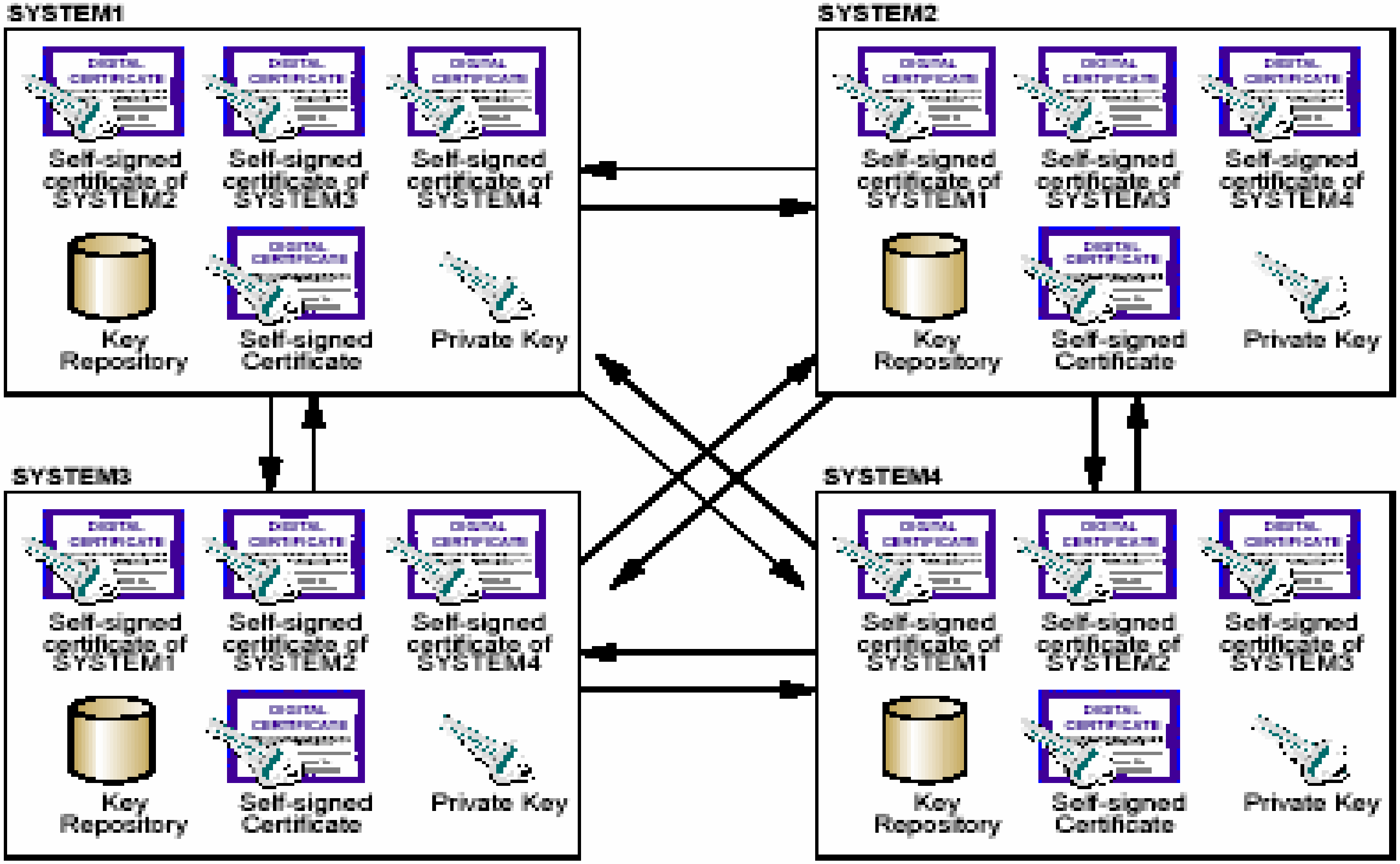
- Binary versus ASCII
- Carriage Returns
- May need HEX edit

Good Certificate Request



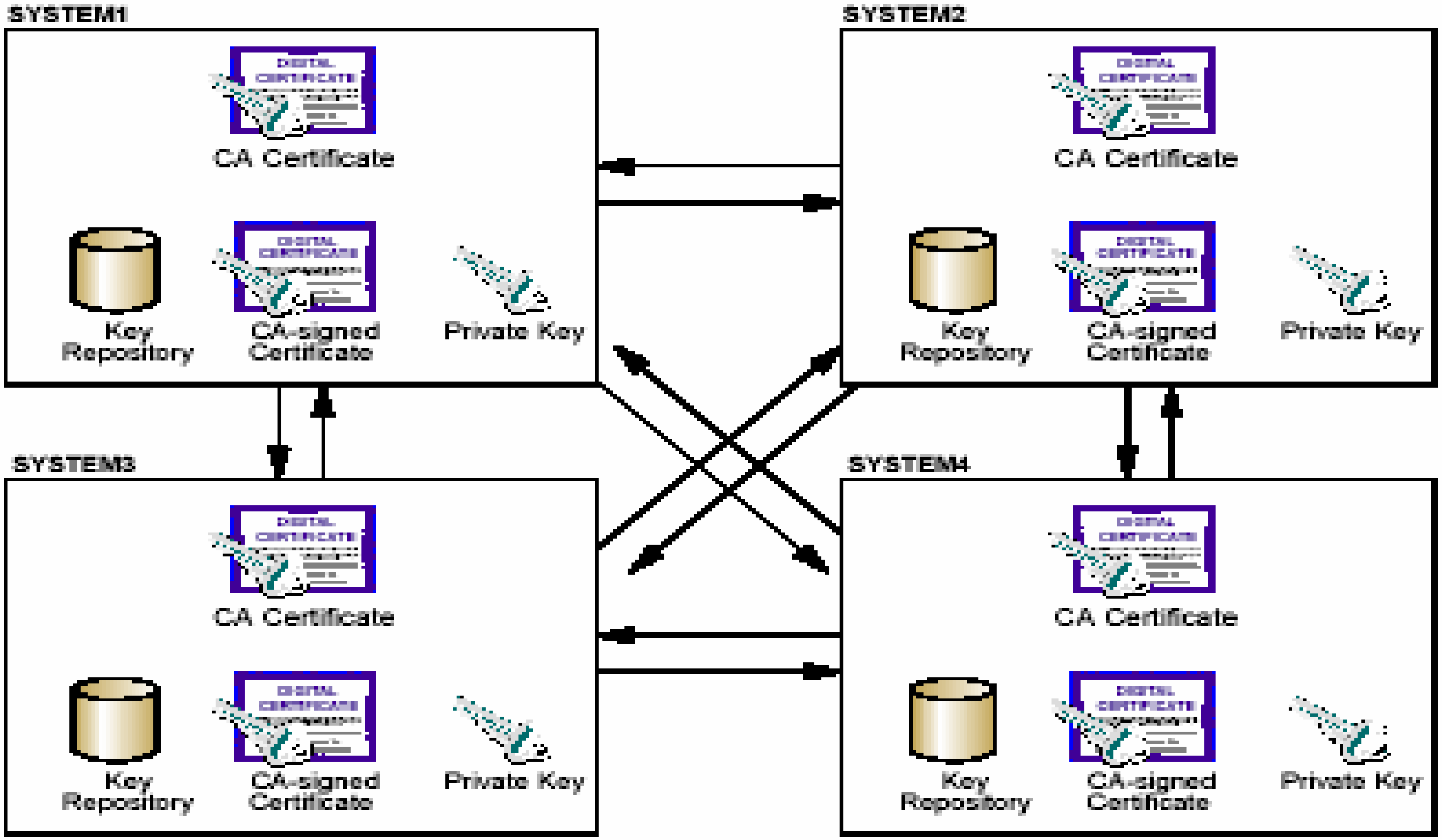
```
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBr0CCAhUCAQAwbDELMAkGA1UEBhMCSlAxETAPBgNVBAgTCEhva2thaWRvMRAwDgYDVQQHEwdT
YXBwb3JvMRUwEwYDVQQKEwxEb2xwa6LuIEJhbmsxEDAOBgNVBAsTB1MhcHBvcm3xDzANBgNVBAMT
Bk1AQU1TMjCBnzANBgkqhkiG9w0BAQEFAA0BjQAwgYkCgYEAj8qgBZ1KbHFdejNp+hevlQdBxiia
JELCU5xucGMWFzfPTJ2QC1MPCiJEceO1w+jwg0eRQXaH+YTad2gdMyH8aughBBlPPOeURw982TKF
UA1tsUqUtD/aMK9Cm/zibEoFciEMDWuCmaZsTSRXZL+Qa1Fm/pm47185CBLe90CsJdUCAwEAAaAA
MA0GCSqGSIb3DQEBBAUAA4GBAHKkZmLOT8xfgjpn5HvOHWFaeKF/egyupLr0F1UUQpgZufuYQDmt
tAVVyn4MxBcFvc1Bt1i/Dgn/GA3QxehcG3x9J0ytcCu30dWokagg404KjWL1Ai3xr415eBCG1jZc
RWEtzMbnzafyNe9R9mcFDiGKrmT1E27XL2AhFfU7MUJT
-----END NEW CERTIFICATE REQUEST-----
```

Using Self-Signed Certificates


SYSTEM1
Self-signed certificate of SYSTEM2
Self-signed certificate of SYSTEM3
Self-signed certificate of SYSTEM4
Key Repository
Self-signed Certificate
Private Key
SYSTEM2
Self-signed certificate of SYSTEM1
Self-signed certificate of SYSTEM3
Self-signed certificate of SYSTEM4
Key Repository
Self-signed Certificate
Private Key
SYSTEM3
Self-signed certificate of SYSTEM1
Self-signed certificate of SYSTEM2
Self-signed certificate of SYSTEM4
Key Repository
Self-signed Certificate
Private Key
SYSTEM4
Self-signed certificate of SYSTEM1
Self-signed certificate of SYSTEM2
Self-signed certificate of SYSTEM3
Key Repository
Self-signed Certificate
Private Key
DIGITAL CERTIFICATE

- All Certificates on all systems
- Difficult to manage
- OK for testing purposes

Using CA Certificates



- CA Certificates on all systems
- Personal CA-signed Certificate
- Much easier to manage

Additional WebSphere Product Resources

- Discover the latest trends in WebSphere Technology and implementation, participate in technically-focused briefings, webcasts and podcasts at: http://www.ibm.com/developerworks/websphere/community/
- Learn about other upcoming webcasts, conferences and events: http://www.ibm.com/software/websphere/events_1.html
- Join the Global WebSphere User Group Community: www.websphere.org
- Access key product show-me demos and tutorials by visiting IBM Education Assistant: http://www.ibm.com/software/info/education/assistant
- Learn about the Electronic Service Request (ESR) tool for submitting problems electronically: http://www.ibm.com/software/support/viewlet/ESR_Overview_viewlet_swf.html
- Sign up to receive weekly technical My support emails: http://www.ibm.com/software/support/einfo.html
- Attend WebSphere Technical Exchange conferences or Transaction and Messaging conference: http://www.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=page&c=a0011317

Questions and Answers