Hybrid Cloud

Introduction to IBM MQ Advanced Message Security (MQ AMS)

Carl Farkas
IBM Europe zHybrid Cloud Integration Consultant
Paris, France
Internet : farkas @ fr.ibm.com
Notes : Carl Farkas/France/IBM @ IBMFR

Agenda

- What is MQ AMS?
- Key features
- Pre-requisites and runtime environment
- Logical architecture
- Components
- Installation & configuration
- Summary

Why use message-level security?

- MQ networks : difficult to prove security of messages
  - Against message injection / message modification / message viewing
  - Prevalence of sub-contractors
  - Increasing levels of partnerships

- More and more data subject to standards compliance
  - Credit card data protected by PCI-DSS
  - Confidential government data
  - Regional legislation such as GDPR

- Base IBM MQ provides message encryption when the MQ messages *are in transit over channels*.

  But without AMS, MQ messages have never been encrypted while they are sitting in the queues with standard MQ!

General Data Protection Regulation (GDPR)

- The General Data Protection Regulation(GDPR) was published on 16th April 2016, and will be immediately applicable after a 2 year transition period on **25 May 2018** to any organization which operates in the EU market
- Establishes data privacy as a fundamental right. All economic operators, including economic operators, including micro, small and medium-sized enterprises must have **processes, technology, and automation to effectively protect personal data*.**
- Introduces cross-industry **72 hours breach reporting** to regulators and without undue delay to individuals with associated risk of severe reputational harm.
- Non-compliance has the potential to lead to huge fines of **up to €20m or 4% of total annual worldwide turnover.**

*Article 4 paragraph 1
"Personal data" means any information relating to an identified or identifiable natural person ("data subject");

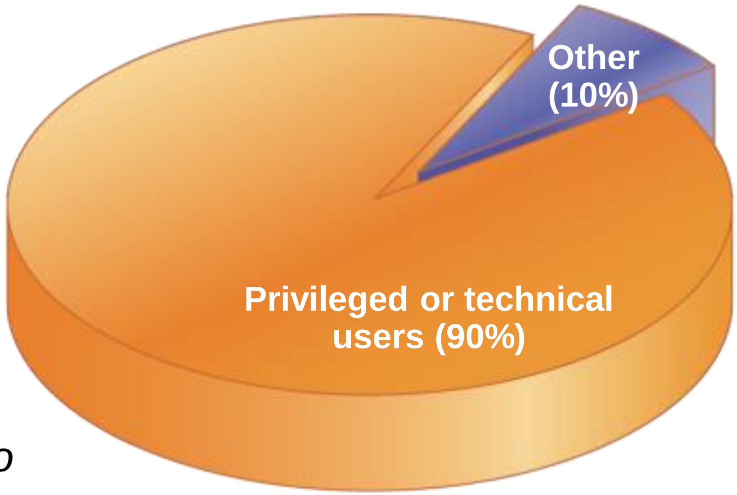
The *"thief"* is inside the gate!

The enemy is us:

- **90% of insider incidents are caused by privileged or technical users**
- **Most are inadvertent violations of:**
  - Change management process
  - Acceptable use policy
  - Account management process
- **Others are deliberate, due to:**
  - Revenge (84%)
  - "Negative events" (92%)
- **Regardless, too costly to ignore:**
  - Internal attacks cost 6% of gross annual revenue or 9 dollars per employee per day

"*Among 874 incidents, as reported by companies to the Ponemon Institute for its recent 2016 Cost of Data Breach Study, 568 were caused by employee or contractor negligence; 85 by outsiders using stolen credentials; and 191 by malicious employees and criminals*." – Tripwire security, April 2017

Who Causes Internal Incidents?

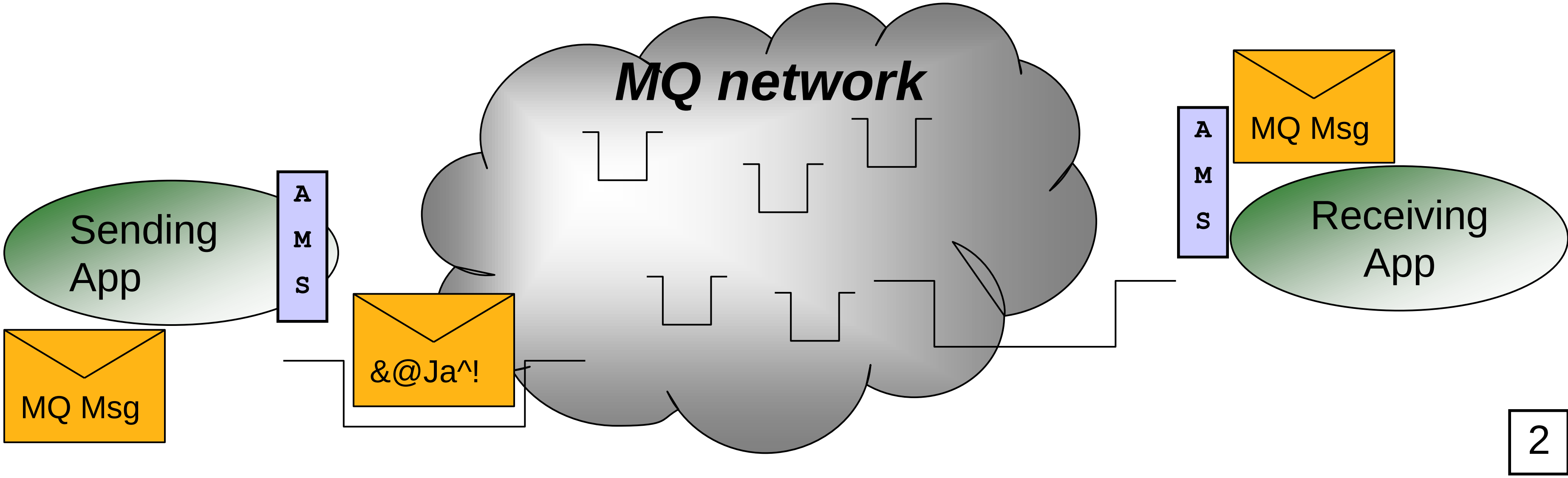


Sources: Forrester research, IdM Trends 2006; USSS/CERT Insider Threat Survey 2005/6; CSI/FBI Survey, 2005; National Fraud Survey; CERT, various documents.

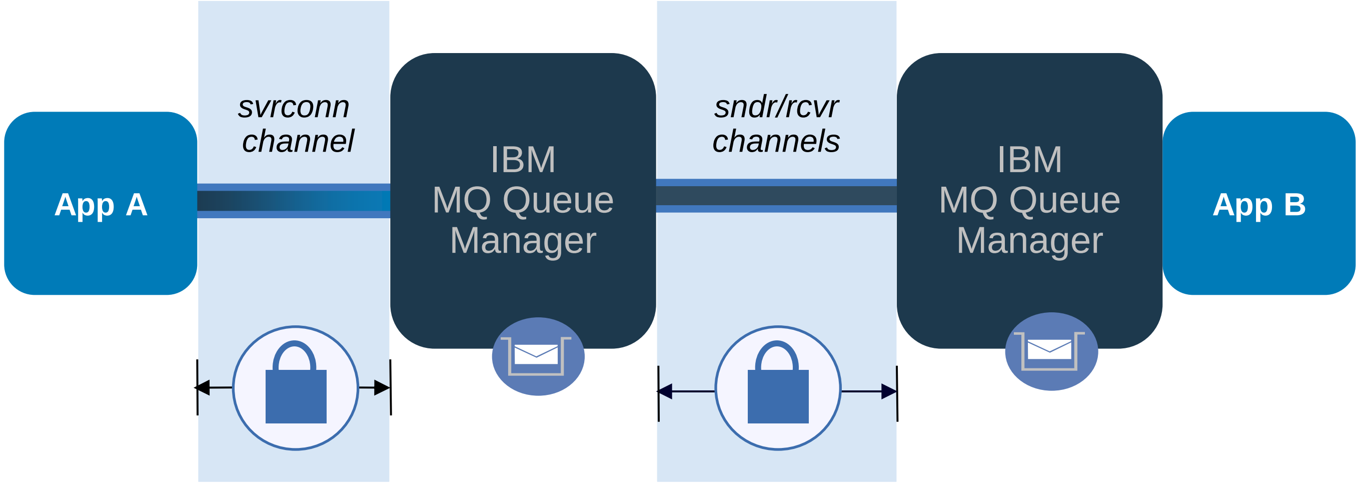
What is MQ AMS?

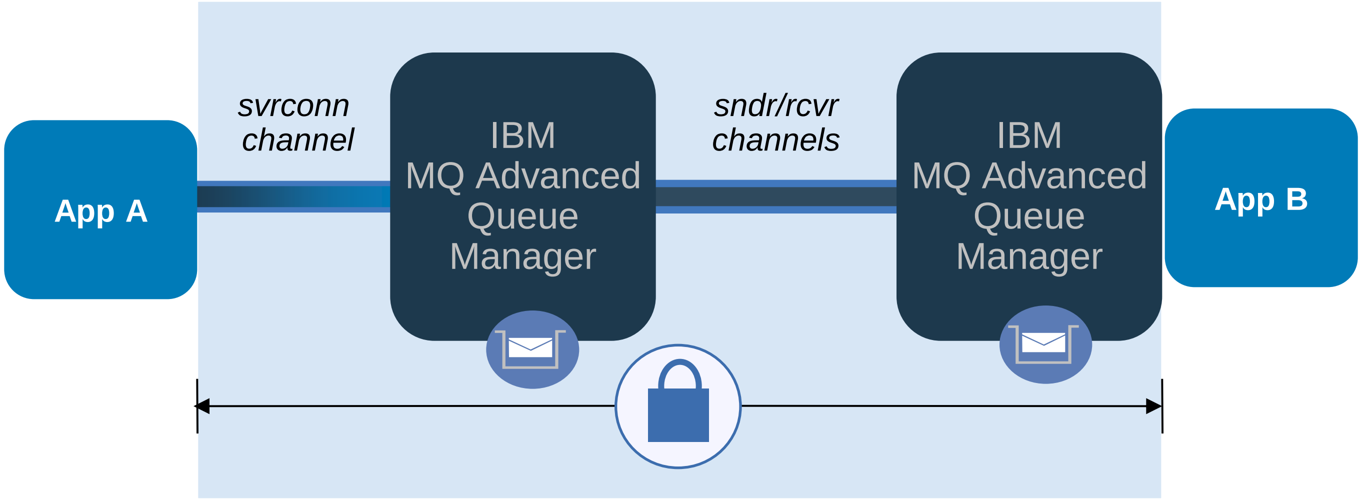
MQ Advanced Message Security

- Provides security for MQ messages, *end-to-end* with *no application changes*
- It is a simple “add-on” product that enhances IBM MQ
- Security policies are used to define the security level required which leverage X.509 certificates



MQ Channel SSL vs. MQ AMS protection

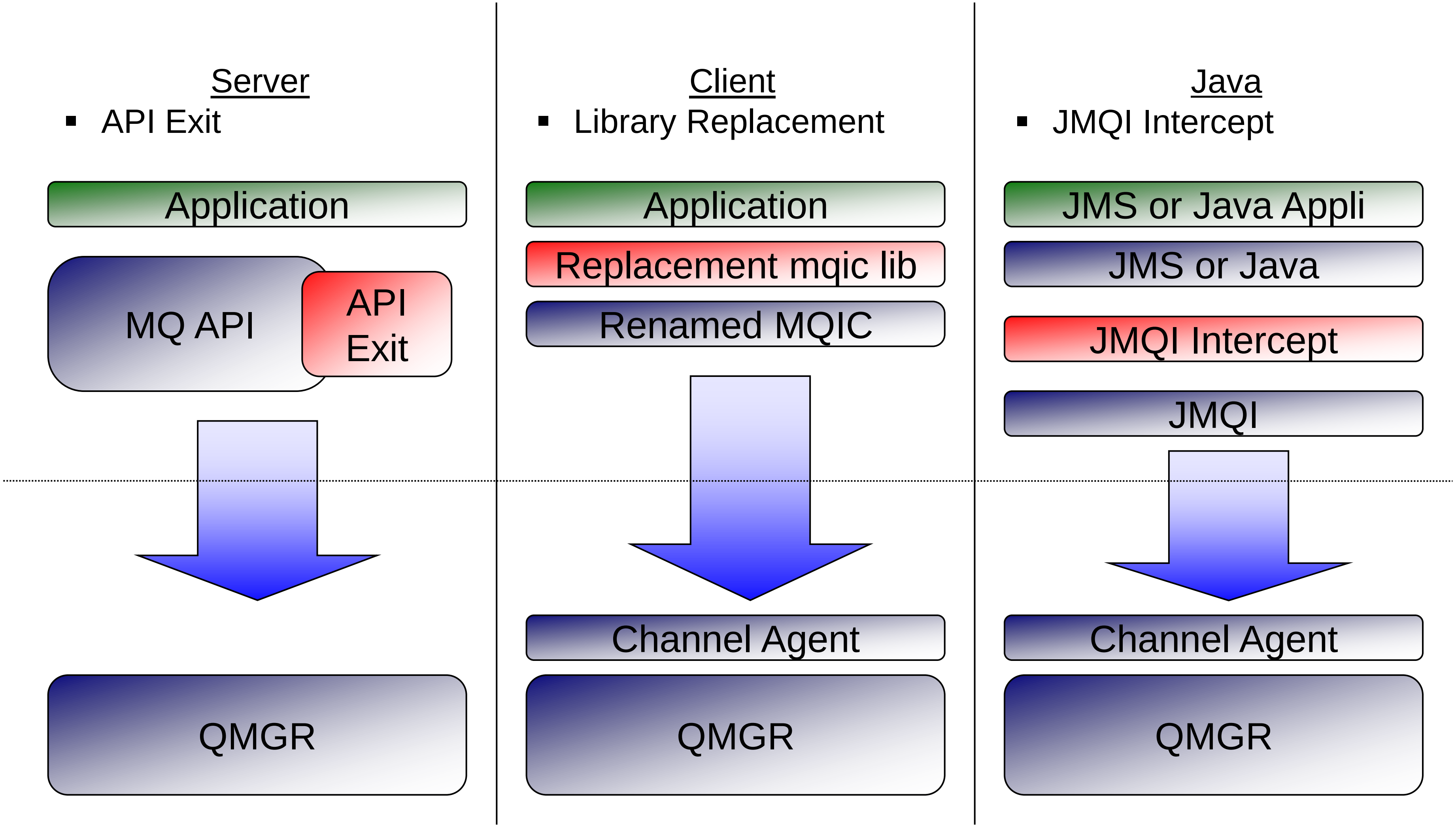




AMS Key Features

- Secures sensitive or high-value MQ messages
  - Privacy via message content encryption
  - It leverages digital certificates (X.509) and Public Key Infrastructure (PKI) to protect MQ messages
- Detects and removes rogue or unauthorized messages before they are processed by receiving applications
  - Authentication via certificate *above and beyond* operating system or MQ authentication
  - Authorization to queue *above and beyond* MQ OAM or SAF
- Verifies that messages are not modified in transit
  - Message Integrity via digital signature of message content
- Protects messages not only when they flow across the network but when they are at rest in queues
- Messages from existing MQ applications are transparently secured using “interceptors”
  - No application changes are necessary
- No pre-requisite products other than MQ
- Supports all current MQ versions (some restrictions apply)

Interceptors

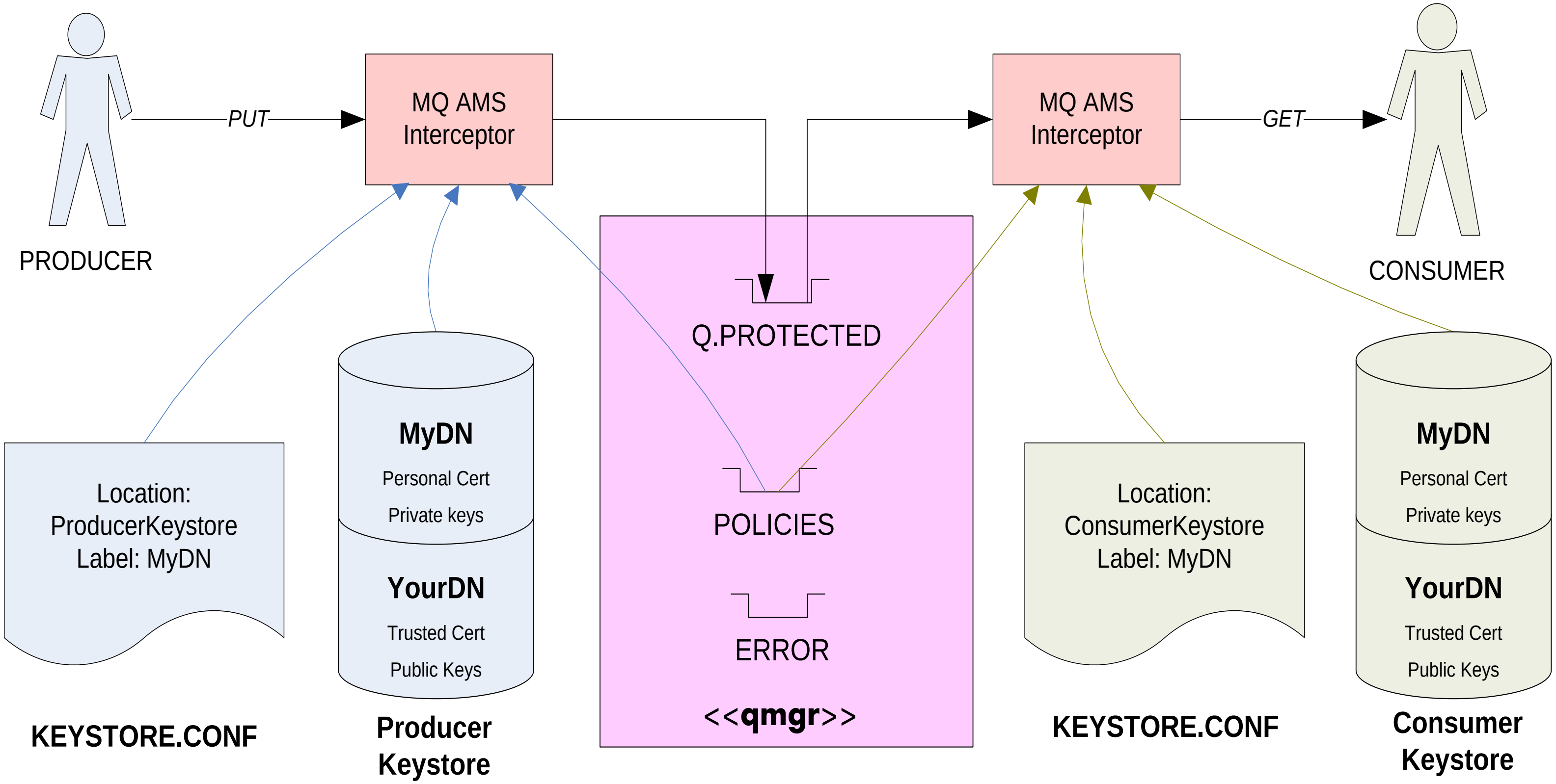


Environments supported

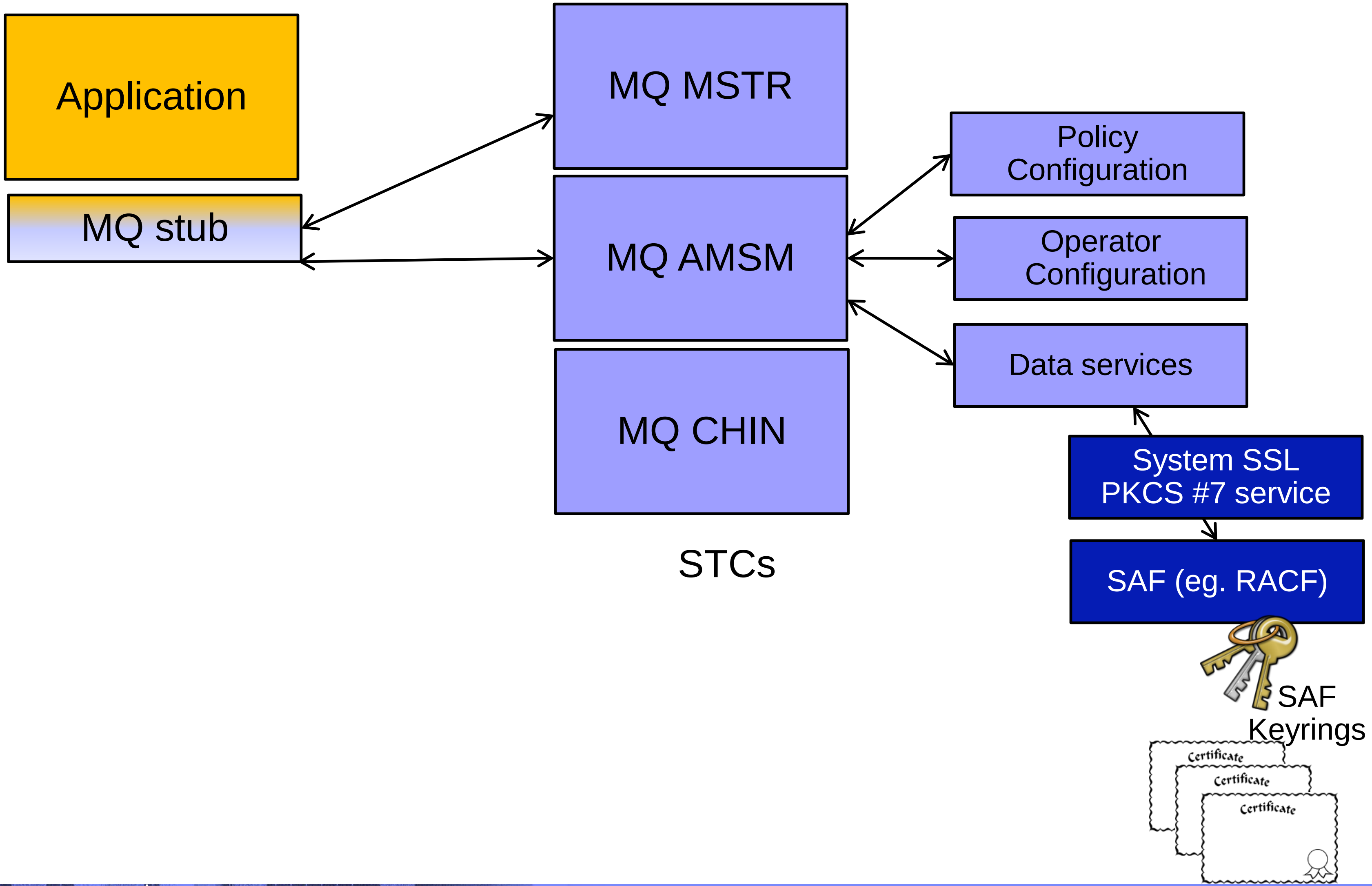
- MQ AMS functionality is implemented in “interceptors”
  - There are no long running processes or daemons (except in z/OS)
  - Existing MQ applications **do not require changes**
- Three interceptors are provided:

1. **MQ Server interceptor** for local (bindings mode) MQI API and Java applications.
   - Implemented as standard QM API exit on distributed, and “private” API exit on z/OS
   - Note that MQ v7 is required for the AMS MQ Explorer plugin
2. **MQ Client API interceptor** for remote (client mode) MQ API applications.
   - MQ AMS interceptor imbedded in MQ client code
3. **MQ Java client interceptor** for remote (client mode) MQ JMS and MQ classes for java applications (J2EE and J2SE).
   - MQ AMS interceptor imbedded in MQ java client code.
   - Requires MQ Java v7 minimum. Note: non-IBM Java supported as of MQ AMS v9.

Logical Architecture Design – Distributed Platforms



AMS task – z/OS



2

Message protection policies

- Created or updated or removed by command ‘`setmqspl`’
  - – or by MQ AMS plug-in for MQ Explorer (GUI)
- Policies are stored in queue `'SYSTEM.PROTECTION.POLICY.QUEUE'`
- Each protected queue can have only one policy
- For distributed queuing, protect the queue locally (source QM) as well as the remotely (target QM)
- Three types of policies:
  - – Message Integrity policy – digital signature for authentication and tamper protection
  - – Message Privacy policy – Integrity, plus message encryption
  - – Message Confidentiality policy (new in MQv9) – encryption only
- Display policies with command ‘`dspmqspl`’
- “Compromised messages” in queue `'SYSTEM.PROTECTION.ERROR.QUEUE'`
- Extra queue on z/OS `'SYSTEM.PROTECTION.SYNC.QUEUE'`

1

Message integrity policy definition

- There are two message signing algorithms: SHA and MD5
- The list of authorized signers is optional
  - If no authorized signers are specified then any application can sign messages.
  - If authorized signers are specified then only messages signed by these applications can be retrieved.
  - Messages from other signers are sent to the error queue
- On z/OS, same setmqspl program and parms used as SYSIN DD for PGM=CSQ0QUTIL

Syntax:

```
setmqspl
-m <queue_manager>
-p <protected_queue_name>
-s <SHA1,256,384,512 | MD5>
-a <Authorized signer DN1>
-a <Authorized signer DN2>
    :
```

Example:

```
setmqspl -m MYQM
-p MY.Q.INTEGRITY
-s SHA256
-e NONE
-a "CN=cfarkas,O=ibm,C=FR"
```

Message privacy and confidentiality policy definition

- Message privacy requires that encrypted messages are also signed.
- The list of authorized signers is optional.
- It is mandatory to specify at least one message recipient
- Messages retrieved by un-authorized recipients cause messages to be sent to the SYSTEM.PROTECTION. ERROR.QUEUE.
- Note "-c" requests Confidentiality, and number of key reuses; new with MQv9

Syntax:

```
setmqspl
-m <queue_manager>
-p <protected_queue_name>
-s <SHA1,256,384,512 | MD5>
-e <RC2,DES,3DES,AES128,AES256>
-a <Authorized signer DN1>
-a <Authorized signer DN2>
-r < Message recipient DN1>
-r < Message recipient DN2>
-c < key reuse count>
```

Example:

```
setmqspl -m MYQM
-p MY.Q.PRIVACY
-s SHA1
-e AES128
-a "CN=carl,O=ibm,C=US"
-r "CN=ginger,O=catunion,C=JP"
-r "CN=saadb,OU=WBI,O=IBM,C=FR"
```

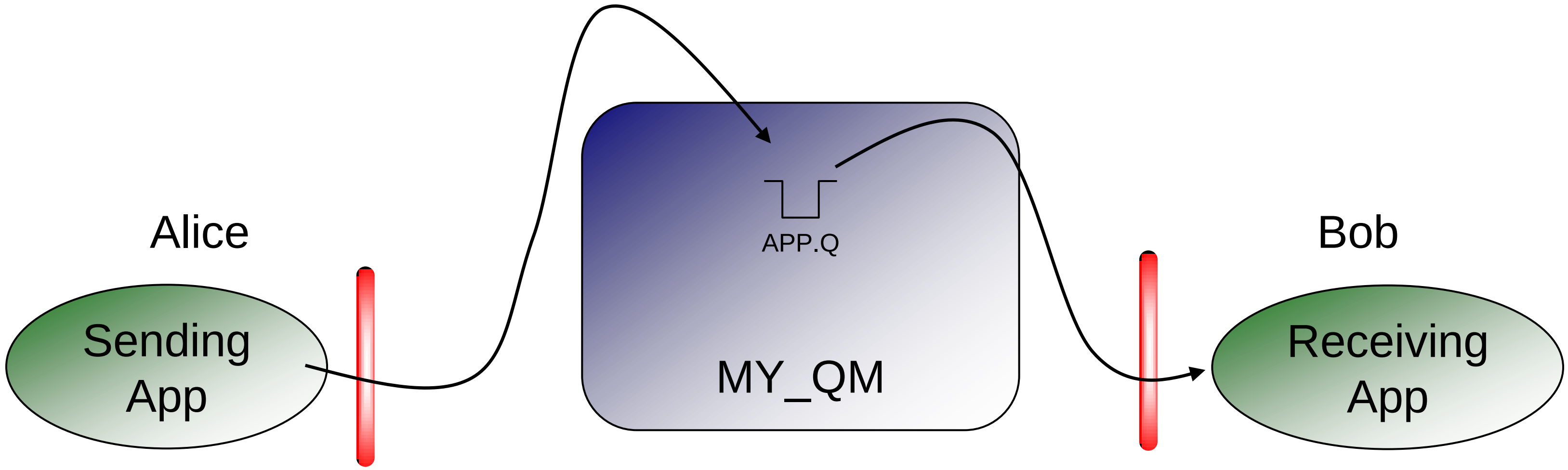
Keystores and X.509 certificates

- **Each MQ application** producing or consuming protected messages **requires access to a keystore** that contains a personal X.509 (v2/v3) certificate and the associated private key.
- The keystore and certificate is accessed by the MQ AMS interceptors.
- The keystore must contain trusted certificates to validate message signers or to obtain the public keys of encrypted message recipients
- Keystore can be the same as that used for MQ SSL
- Several types of keystore are supported (Distributed): CMS, JKS and JCEKS.
- On Distributed MQ, the IBM Key Management (iKeyman, part of GSKit) is provided to create and do simple management of local keystores
- On z/OS, standard SAF product (eg. RACF) used to create certificates which are SAF-managed and must be on a keyring named “`drq.ams.keyring`”
- 3rd party software is available to provide more robust, industrialisation of keystore maintenance.

MQ AMS configuration file (distributed AMS)

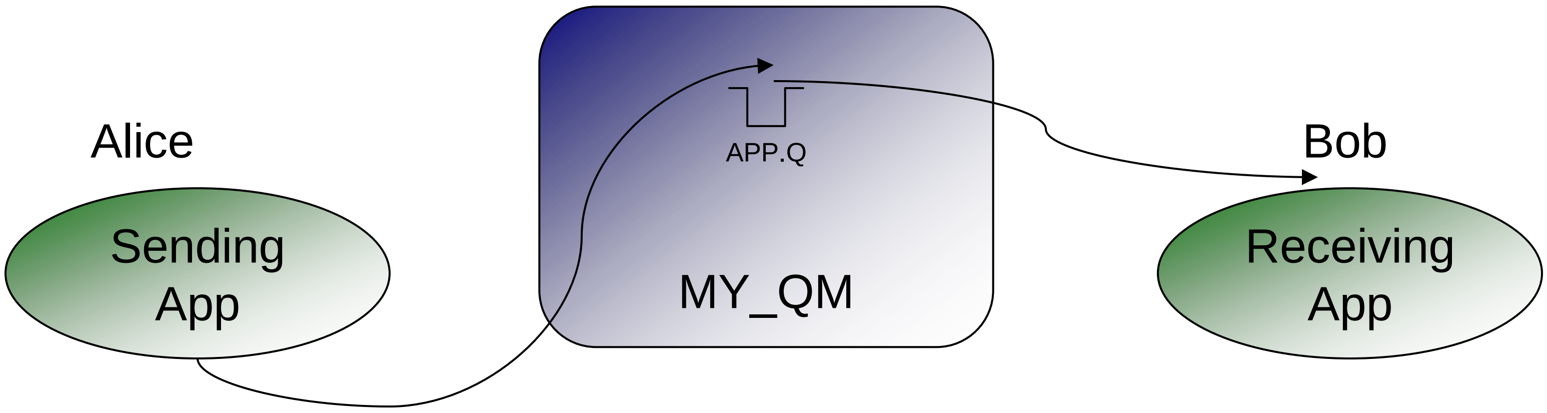
- MQ AMS interceptors require a configuration file, eg. `KEYSTORE.CONF`, which contains:
  - Type of keystore: CMS, JKS, JCEKS
  - Location of the keystore.
  - Label of the personal certificate.
  - Passwords to access keystore and private keys (or `.sth` stash for CMS format)
- Interceptors locate the configuration file using one of the following methods:
  - Environment variable MQS_KEYSTORE_CONF=<path to conf file>.
  - Checking default locations and file names.
    - ✓ Platform dependent. For example in UNIX: “$HOME/.mqs/keystore.conf”

IBM MQ AMS – Steps to get AMS running

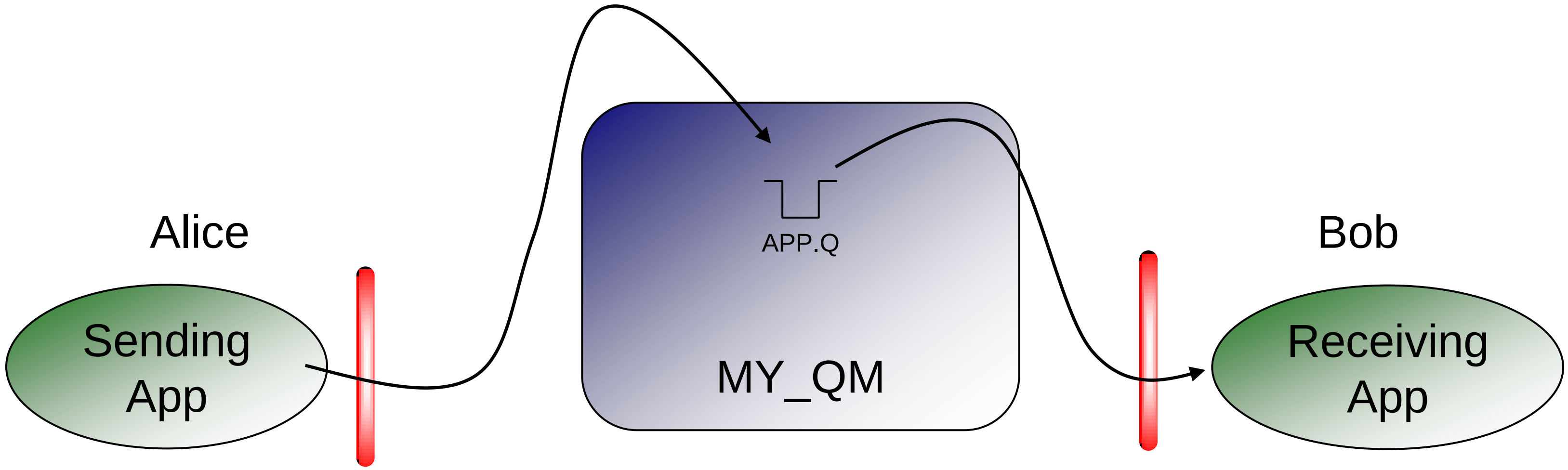


1. Install AMS Interceptor
2. Enable AMS
3. Setup keystores with public / private key pairs
4. Configure protection policy for the queue (setmqspl)

IBM MQ AMS – install/config example, 0

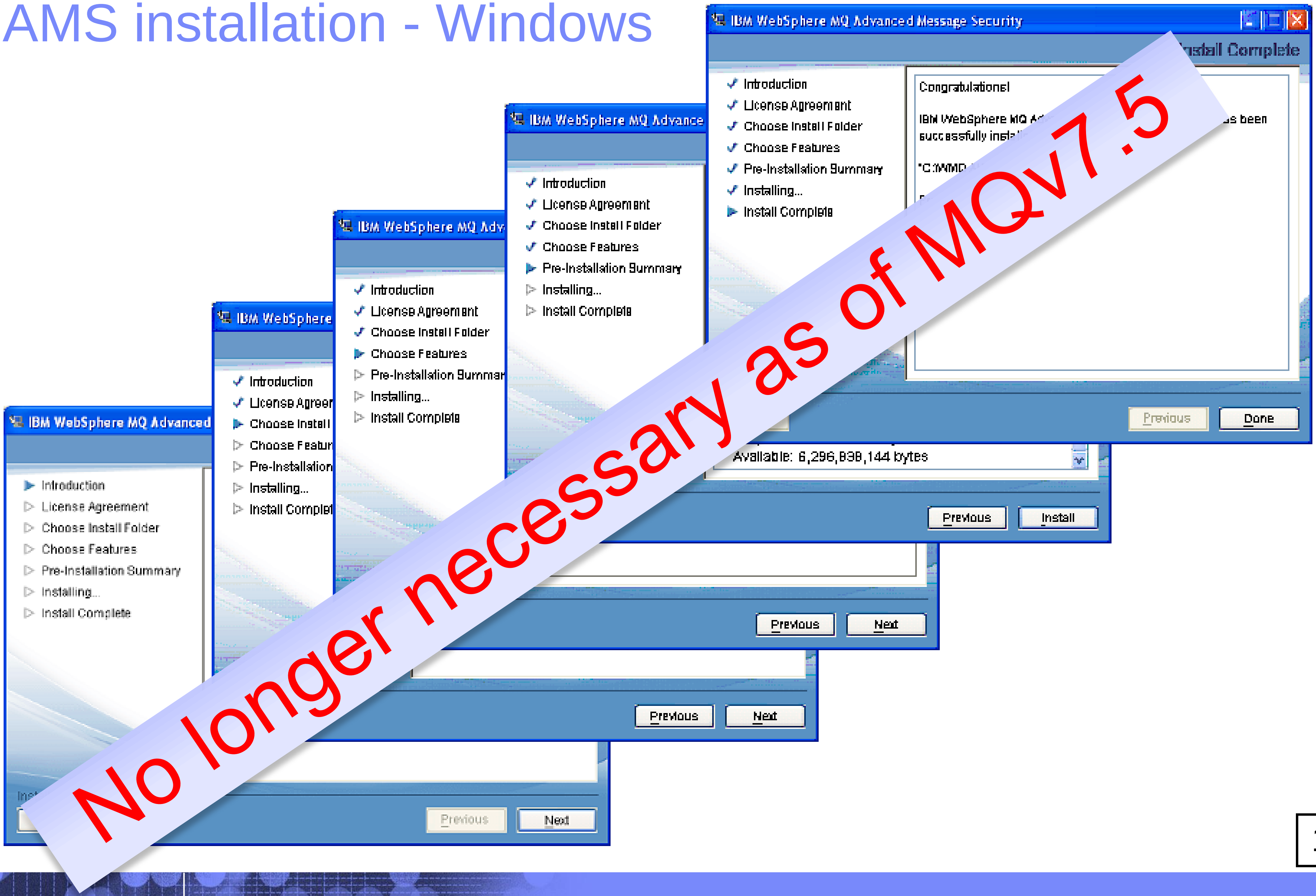


IBM MQ AMS – install-run example, step 1



1.Install AMS Interceptor

AMS installation - Windows



Congratulations!

IBM WebSphere MQ A... ...s been successfully insta...

"C:\WMQ...

Available: 6,296,838,144 bytes

No longer necessary as of MQv7.5

1

AMS installation - z/OS

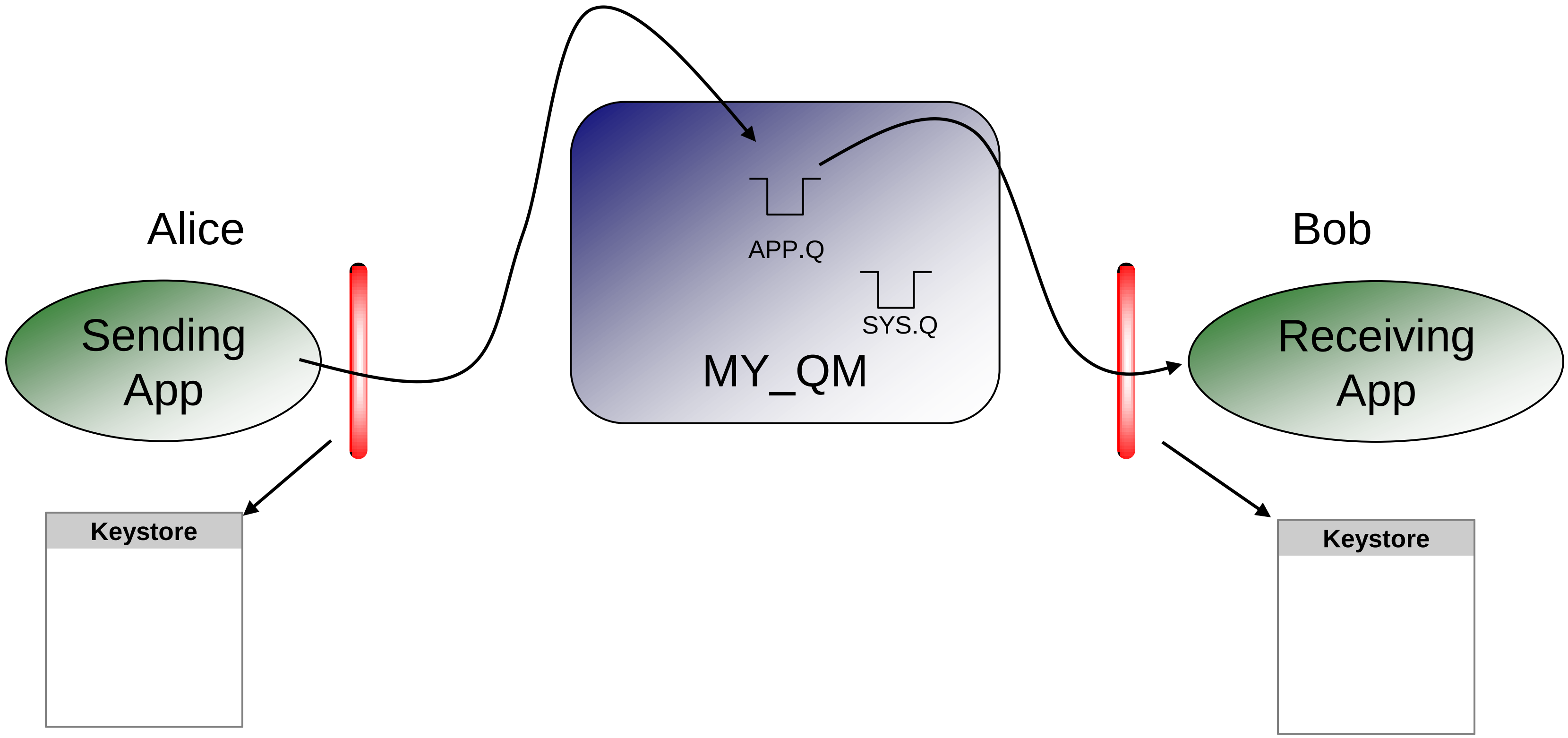
- SMP/E installation
  - Program 5655-AM9, FMID HAMS900
  - Requires 20 tracks (Target), 20 tracks (Distribution)
- Post-installation tasks
  - Update LPA for AMS module CSQ0DRTM in SCSQLINK
  - Update Authorized Program Facility (APF) list for SDRQAUTH
  - Update Program Properties Table (PPT) update for CSQ0DSRV
  - Possible update to DIAG member for allocating in user storage key
  - Update MQ MSTR procedure definition to include AMS
  - Create MQ AMSM procedure definition
- SAF (eg. RACF) definitions for the STC
  - Set up the userid(s) for the STCs, give SAF permissions, RACDCERT defs*
- SAF definitions for userids that will be putting & getting protected messages
  - An OMVS segment associated with their userid (or set default with FACILITY class, BPX.DEFAULT.USER)
  - SAF UPDATE permission for the FACILITY class, IRR.DIGTCERT.LISTRING

For a step by step guide and more details, see the MQ KC “q119090_”

* I can provide you with an example of the RACF definitions

1

IBM MQ AMS – install-run example, step 2



1. Install AMS Interceptor
2. Enable AMS

AMS enable - distributed AMS

1. Enable AMS system queues

```
runmqsc MY_QM < "C:\WMQ AMS\bin\defineqs.mqs"
DEFINE QLOCAL(SYSTEM.PROTECTION.POLICY.QUEUE) MAXDEPT                    )
    MAXMSGL(4194304) DEFSOPT(SHARED) SHARE DEFPSIST
DEFINE QLOCAL(SYSTEM.PROTECTION.ERROR.QUEUE) MA                 99999)
    MAXMSGL(4194304) DEFSOPT(SHARED) SHARE DE
All valid MQSC commands were process
```

2. Activate AMS interceptors

```
cfgmqs -enable -server MY_QM
DRQDT3052I   The IBM IBM MQ Adv              ge Security server interceptor has
   been enabled successfully
```

3. Set up Environment varia   to AMS key database configuration

```
MQS_KEYSTORE_CONF              uff\Carl\keystore.conf
```

4. Create the AMS   ase configuration file, eg. `C:\AMSStuff\Carl\keystore.conf`

```
cms.keysto              uff/Carl/carlkey
cms.ce              rl_Cert
```

No longer necessary as of MQv7.5

AMS enable - z/OS

- Create MQ objects for AMS (queues)

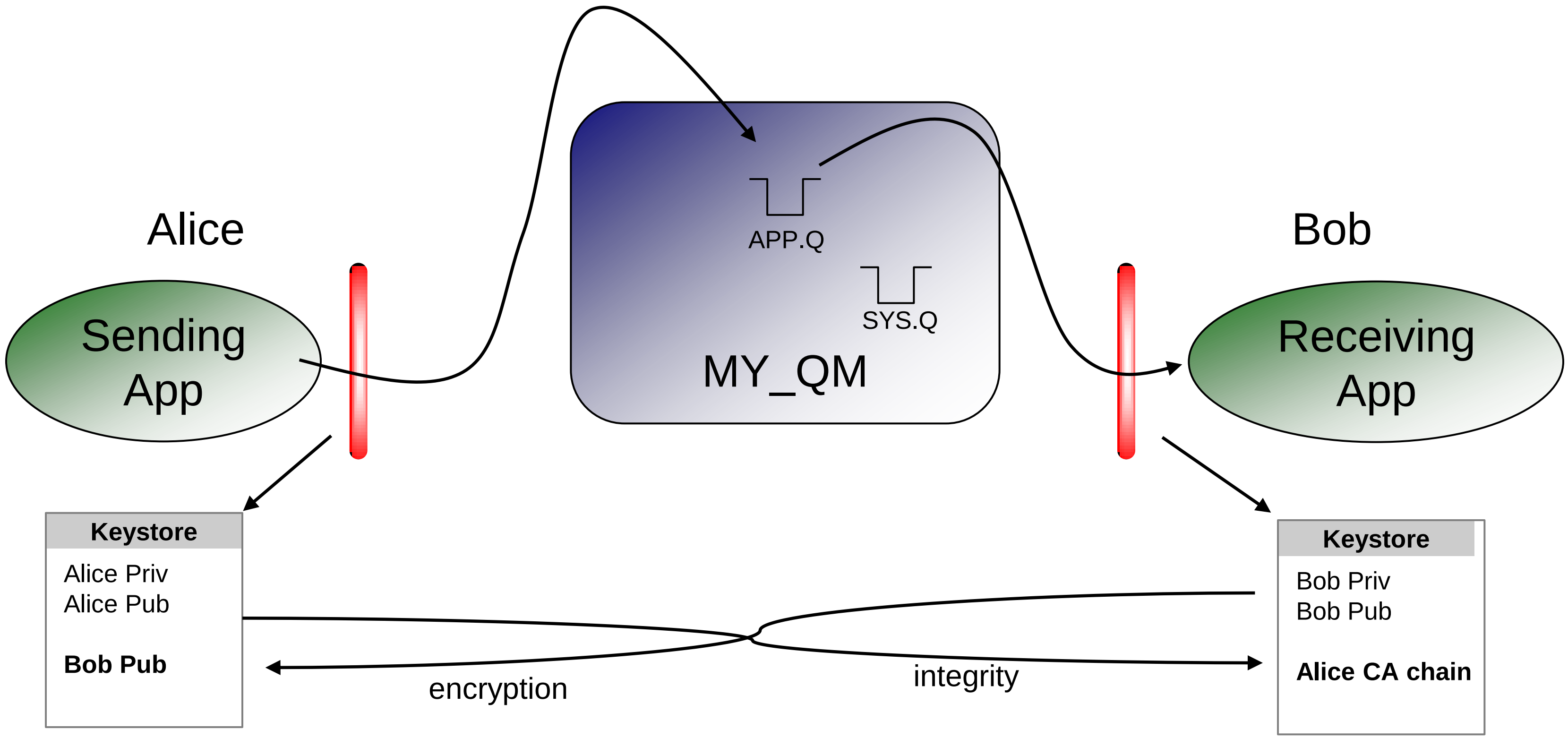
```
//CFAMSQM JOB 'Make MQ AMS queues',CLASS=A,MSGLEVEL=(1,1),
// NOTIFY=&SYSUID
/*JOBPARM SYSAFF=ZT01
//**************************************************************
//*    Define MQ Advanced Message Service (AMS) system queues
//**************************************************************
//STEP1    EXEC PGM=CSQUTIL,PARM='QZ0X',REGION=1M
//* STEPLIB  DD  DSN=WMQ.V9R0M0.SCSQAUTH,DISP=SHR
//*          DD  DSN=WMQ.V9R0M0.SCSQANLE,DISP=SHR
//OUTDEF   DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//SYSIN    DD  *
COMMAND DDNAME(CMDDEF) FAILURE(CONTINUE)
/*
//CMDDEF   DD  DSN=WMQ.V9R0M0.SCSQPROC(CSQ4INSM),DISP=SHR
//
```

MQ supplies the AMS object definitions

- Update MQ zPRM SPLCAP

```
    :
SMFSTAT=YES,        GATHER SMF STATS        *CBF*       X
SPLCAP=YES,         MESSAGE ENCRYPTION REQUIRED *CBF* X
STATIME=30,         STATISTICS RECORD INTERVAL (MIN)   X
    :
```

IBM MQ AMS – install-run example, step 3

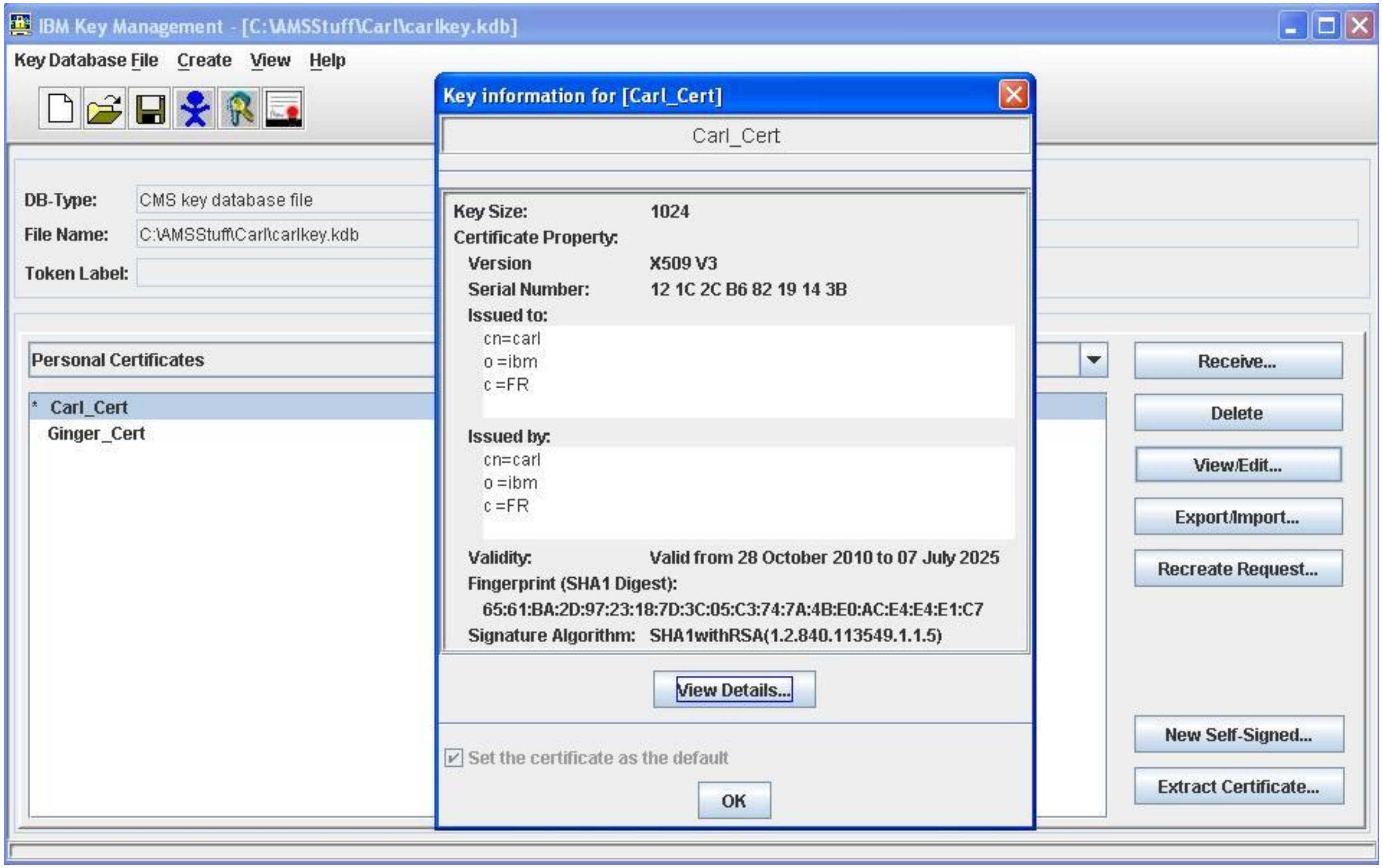


1. Install AMS Interceptor
2. Enable AMS
3. Setup keystores with public / private key pairs
   a) copy sender's CA key chain to receiver's keystore for integrity
   b) copy receiver's public key to sender's keystore for encryption

2

Certificate management

- IBM MQ supplies iKeyman with GSKit
- Line-mode commands also available (eg. gsk7capicmd)
- On z/OS, RACF commands perform certificate management



DB-Type: CMS key database file

File Name: C:\AMSStuff\Carl\carlkey.kdb

Token Label:

* Carl_Cert

Ginger_Cert

Carl_Cert

Key Size: 1024

Certificate Property:

Version X509 V3

Serial Number: 12 1C 2C B6 82 19 14 3B

Issued to:
cn=carl
o =ibm
c =FR

Issued by:
cn=carl
o =ibm
c =FR

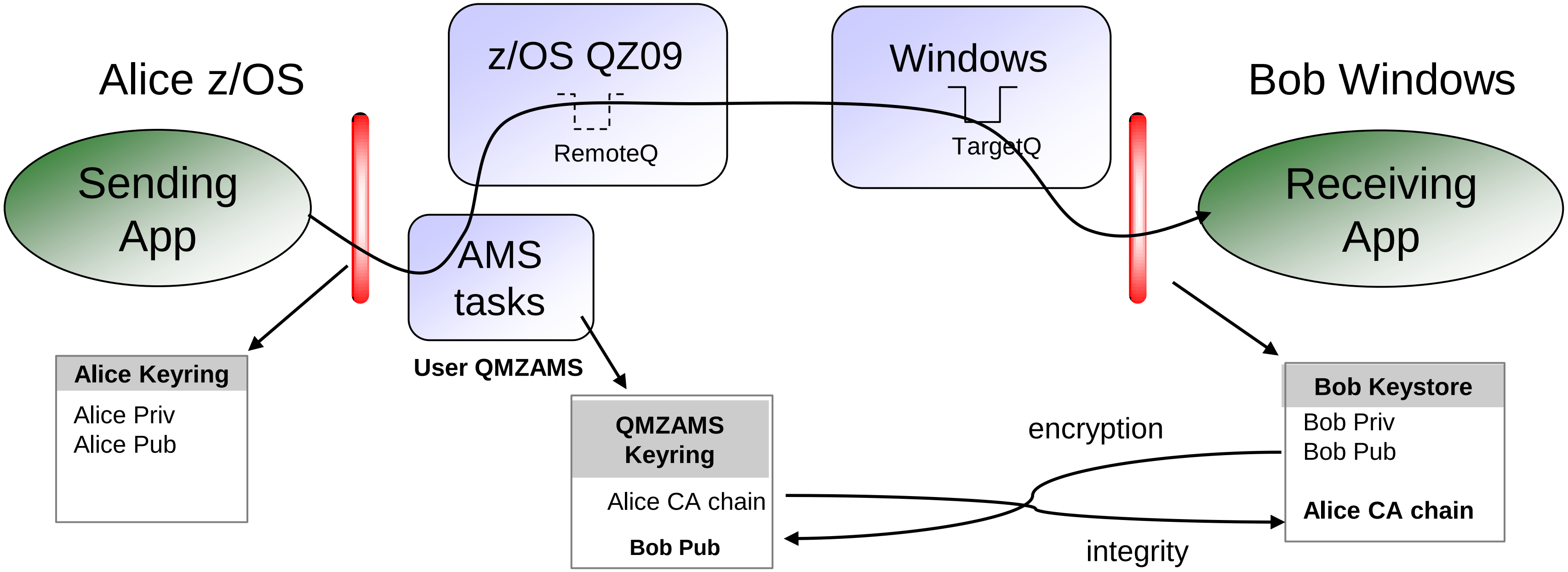
Validity: Valid from 28 October 2010 to 07 July 2025

Fingerprint (SHA1 Digest):
65:61:BA:2D:97:23:18:7D:3C:05:C3:74:7A:4B:E0:AC:E4:E4:E1:C7

Signature Algorithm: SHA1withRSA(1.2.840.113549.1.1.5)

☑ Set the certificate as the default

IBM MQ AMS – Setup z/OS -> Windows



1. Install AMS Interceptor
2. Enable AMS
3. Setup keystores/keyrings with public / private key pairs
   a) copy sender’s CA key chain to receiver’s keystore for integrity
   b) copy receiver’s public key to sender’s keyring for encryption

For several step-by-step examples for z/OS, see the MQ KC at “q119340_”

2

RACF on z/OS (sender side)

```
//SYSTSIN  DD *
```

Integrity

```
 RACDCERT ID(QMZAMS) ADDRING(drq.ams.keyring)

/* Create a CA certificate */
RACDCERT CERTAUTH GENCERT -
  SUBJECTSDN( CN('AMS CertAuth') O('IBM') ) WITHLABEL('AMSCA') -
  KEYUSAGE(CERTSIGN) TRUST NOTAFTER(DATE(2018/12/31) )

/* Connect the CA certificate to the AMSD Data task STC */
RACDCERT ID(QMZAMS) CONNECT(CERTAUTH LABEL('AMSCA') -
   RING(drq.ams.keyring)

 RACDCERT EXPORT( LABEL('AMSCA') ) CERTAUTH FORMAT(CERTB64) -
    DSN(‘ALICE.AMSCA.CERT')

/* Make a keyring for a userid that will be a MQPUTer or MQGETer */
RACDCERT ID(ALICE) ADDRING(drq.ams.keyring)

/* Create a Certificate for a MQPUTer or MQGETer id */
RACDCERT ID(ALICE) GENCERT -
 SUBJECTSDN( c('FR') O('IBM France') CN(‘Alice on Z') ) -
 WITHLABEL(‘ALICEONZ') SIGNWITH(CERTAUTH LABEL('AMSCA') )  -
 NOTAFTER(DATE(2018/12/31) ) -
 KEYUSAGE(HANDSHAKE DATAENCRYPT DOCSIGN)

  RACDCERT ID(ALICE) CONNECT(ID(ALICE) LABEL(‘ALICEONZ') -
 RING(drq.ams.keyring) DEFAULT USAGE(PERSONAL))
```

```
  RACDCERT ID(QMZAMS) ADD(‘ALICE.BOBPUB.CERT') TRUST -
    WITHLABEL(‘BOBPUB.CERT')
  RACDCERT ID(QMZAMS) CONNECT( ID(QMZAMS) LABEL(‘BOBPUB.CERT') -
    RING(drq.ams.keyring) USAGE(SITE) )

 SETROPTS RACLIST(FACILITY) REFRESH
```

Privacy

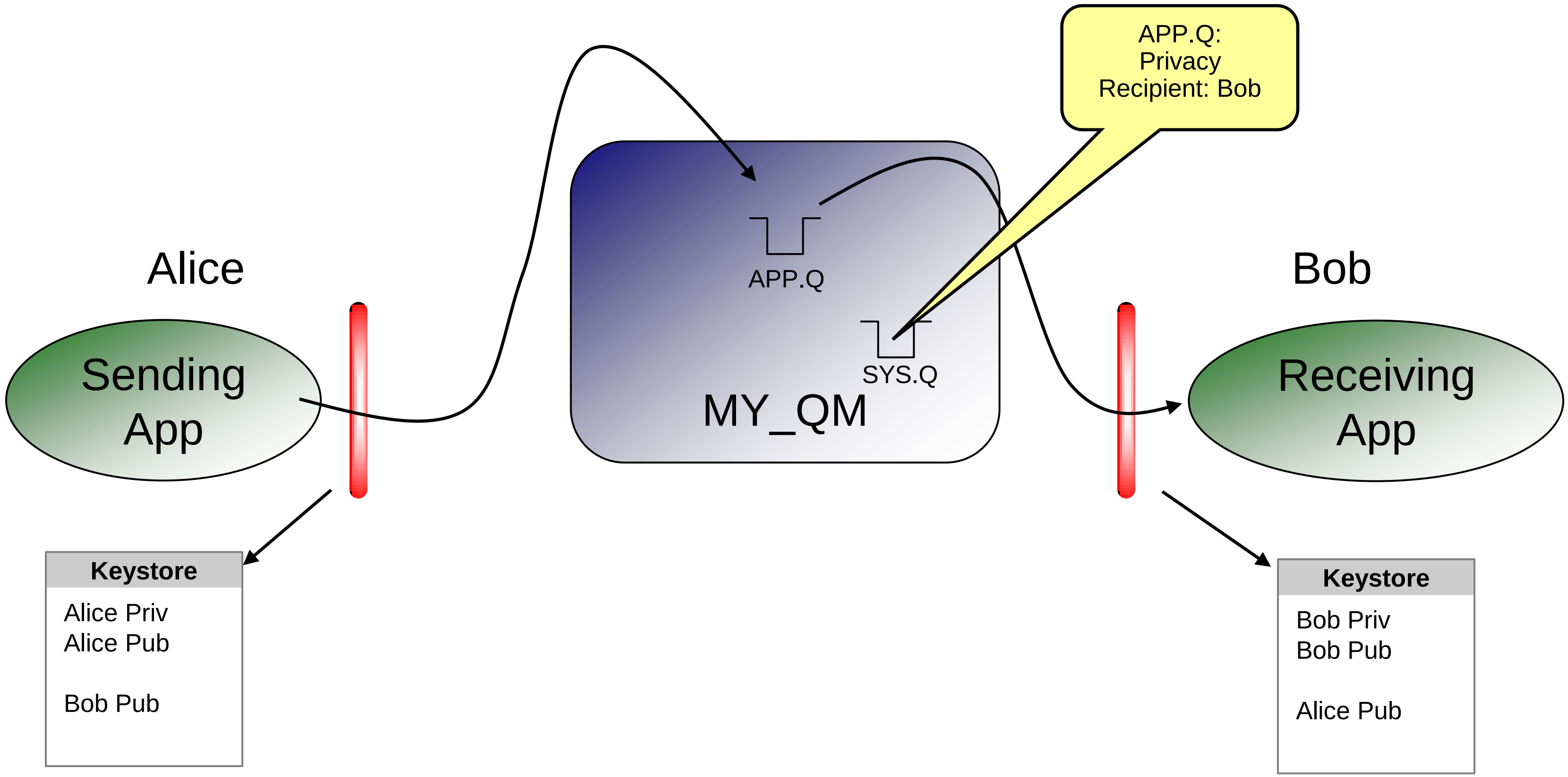
Data task on sender uses CA to validate

PUTer userid on sender uses certif to sign

Export this CA so others can check my certif

Import the Public certif of the Receiver so I can encrypt using this

IBM MQ AMS – install-run example, step 4

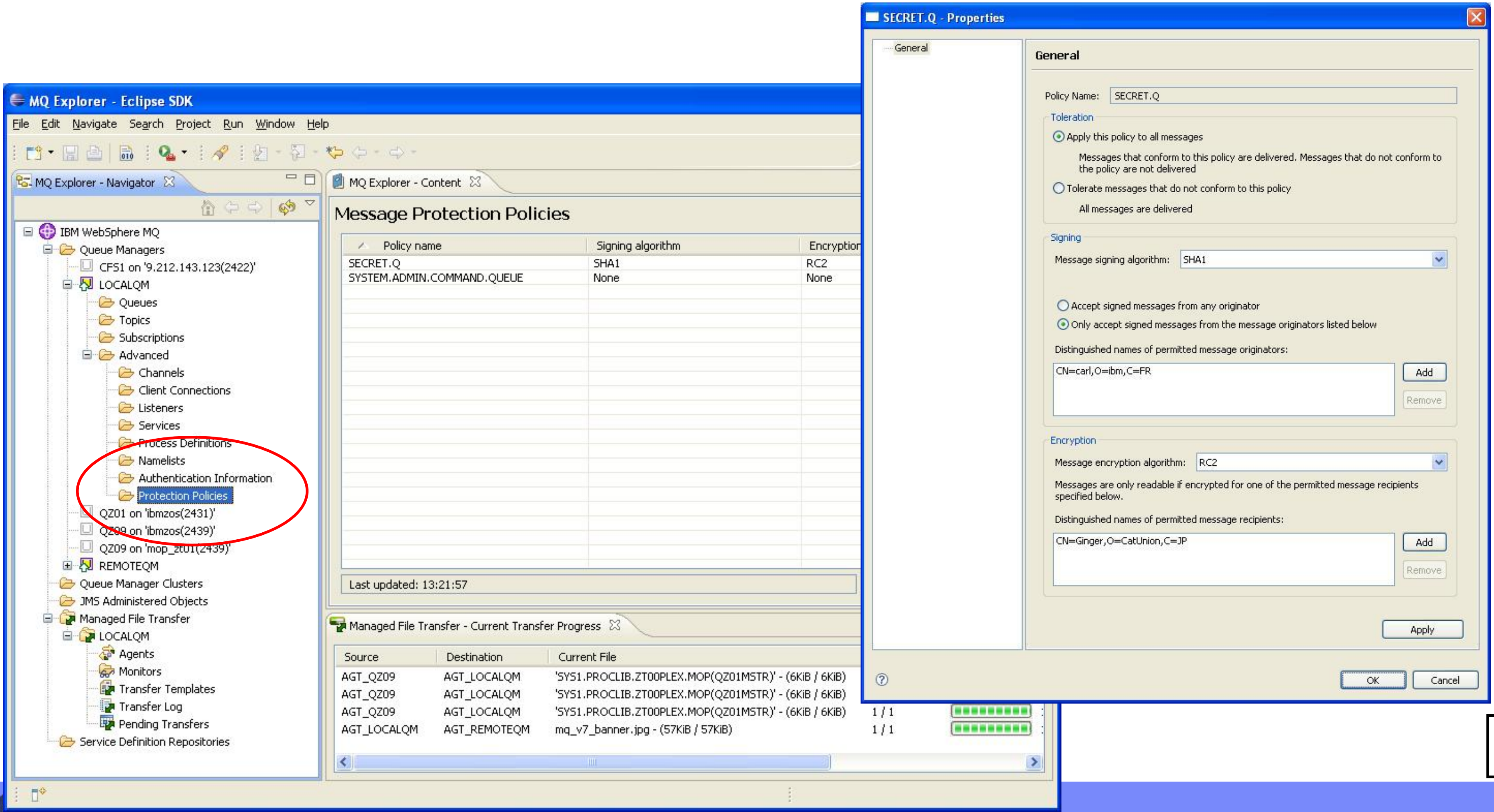


1. Install AMS Interceptor
2. Enable AMS
3. Setup keystores with public / private key pairs
4. Configure protection policy for the queue (setmqspl)

Queue policy definition

- **Use either GUI or line-mode**

```
setmqspl -m LOCALQM -p SECRET.Q -s SHA1 -a
  "CN=carl,O=ibm,C=FR" -e RC2 –r “CN=Ginger,O=CatUnion,C=JP”
```



Message Protection Policies

Policy name	Signing algorithm	Encryption
SECRET.Q	SHA1	RC2
SYSTEM.ADMIN.COMMAND.QUEUE	None	None

Last updated: 13:21:57

Managed File Transfer - Current Transfer Progress

Source	Destination	Current File		
AGT_QZ09	AGT_LOCALQM	'SYS1.PROCLIB.ZT00PLEX.MOP(QZ01MSTR)' - (6KiB / 6KiB)		
AGT_QZ09	AGT_LOCALQM	'SYS1.PROCLIB.ZT00PLEX.MOP(QZ01MSTR)' - (6KiB / 6KiB)		
AGT_QZ09	AGT_LOCALQM	'SYS1.PROCLIB.ZT00PLEX.MOP(QZ01MSTR)' - (6KiB / 6KiB)	1 / 1	🟩
AGT_LOCALQM	AGT_REMOTEQM	mq_v7_banner.jpg - (57KiB / 57KiB)	1 / 1	🟩

General

Policy Name: SECRET.Q

Toleration

● Apply this policy to all messages

Messages that conform to this policy are delivered. Messages that do not conform to the policy are not delivered

○ Tolerate messages that do not conform to this policy

All messages are delivered

Signing

Message signing algorithm: SHA1

○ Accept signed messages from any originator

● Only accept signed messages from the message originators listed below

Distinguished names of permitted message originators:

CN=carl,O=ibm,C=FR

Add Remove

Encryption

Message encryption algorithm: RC2

Messages are only readable if encrypted for one of the permitted message recipients specified below.

Distinguished names of permitted message recipients:

CN=Ginger,O=CatUnion,C=JP

Add Remove

Apply

OK Cancel

1

AMS CSQ0QUTIL commands on z/OS

```
//CFAMSAD JOB 'Make MQ AMS queues',CLASS=A,MSGLEVEL=(1,1),
// NOTIFY=&SYSUID
/*JOBPARM SYSAFF=ZT01
//******************************************************************
//*    Administer MQ Advanced Message Service (AMS)                *
//******************************************************************
//*
//CSQ40CFG EXEC PGM=CSQ0UTIL,
//           PARM='ENVAR("_CEE_ENVFILE_S=DD:ENVARS") /'
//STEPLIB  DD DSN=WMQ.V9R0M2.SCSQANLE,DISP=SHR
//         DD DSN=WMQ.V9R0M2.SCSQAUTH,DISP=SHR
//ENVARS   DD DSN=WBICFG.QZ00.SCSQPROC(CSQ40ENV),DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN    DD * setmqspl -m QZ09
   -p TO.SECRET.FROMZ
   -s SHA1
   -e RC2 -r "CN=carl,O=ibm,C=FR"
/*
//
```

Execute AMS admin commands

Point to parameters

CSQ40ENV

```
_AMS_MSG_LEVEL=*.v
GSK_TRACE_FILE=/u/farkas/AMSstuff/gsktrace
GSK_TRACE=0xff
_AMS_MAX_THREADS=100
_AMS_SMF_TYPE=180
_AMS_SMF_AUDIT=failure
TZ=CUT0GDT
GSK_CRL_SECURITY_LEVEL=HIGH
```

Error handling

- AMS returns a Rc=2063 if the application tries to access (MQGET) a message for which it is not authorized

```
c:\result>amqsgbr SECRET.Q LOCALQM
Sample AMQSGBR0 (browse) start
LOCALQM
MQGET ended with reason code 2063
Sample AMQSGBR0 (browse) end
```

- The event is also logged in the <AMS installation>\log\*.log file

- For destructive MQGET requests, the message is also transferred to the `SYSTEM.PROTECTION.ERROR.QUEUE.` The original message remains there with a DLQ header for administrative handling.

Encrypted message examples

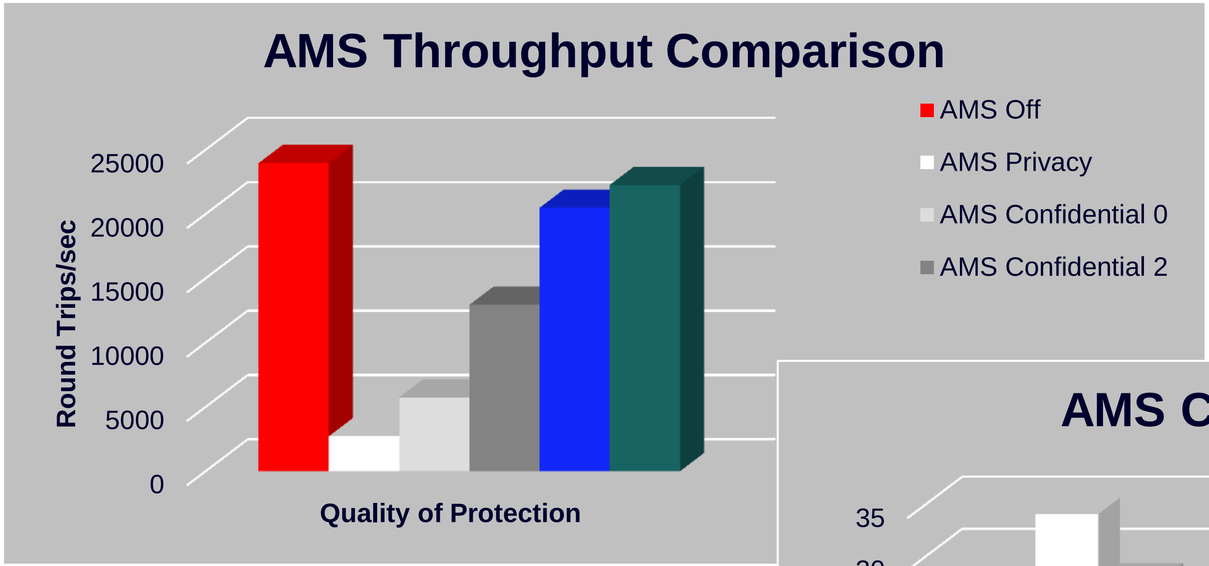
Encrypted message (via Q alias)

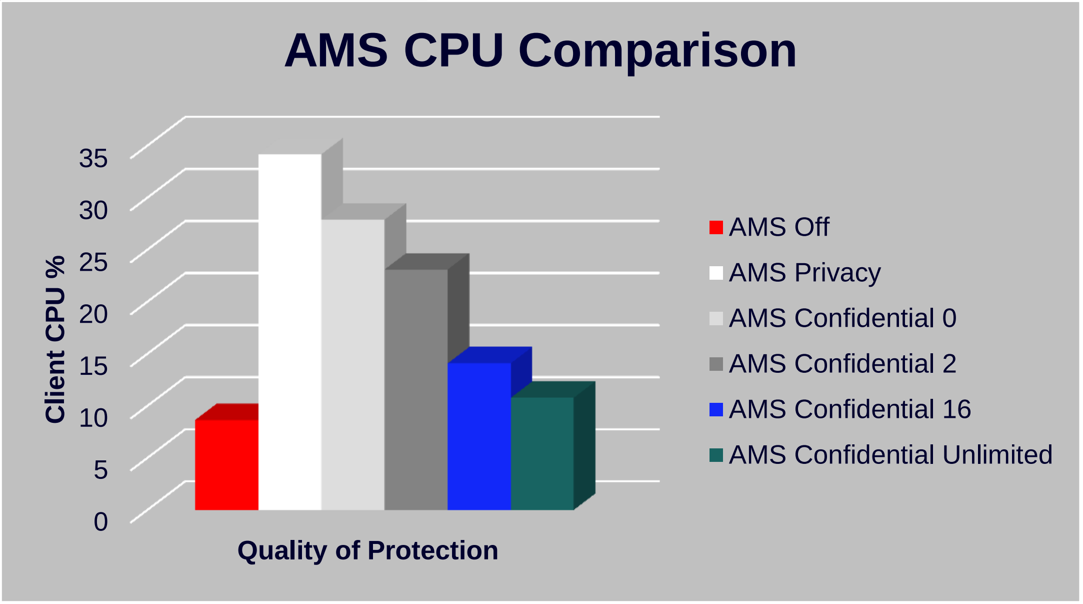
```
c:\result>amqsbcg SECRET.Q.ALIAS LOCALQM

AMQSBCG0 - starts here
**********************

 MQOPEN - 'SECRET.Q.ALIAS'


 MQGET of message number 1
****Message descriptor****

  StrucId  : 'MD  '  Version : 2
  Report   : 0  MsgType : 8
  Expiry   : -1  Feedback : 0
  Encoding : 546  CodedCharSetId : 437
  Format : '        '
  Priority : 0  Persistence : 0
  MsgId : X'414D51204C4F43414C514D2020202020D403CF4C201A0402'
  CorrelId : X'000000000000000000000000000000000000000000000000'
  BackoutCount : 0
  ReplyToQ       : '                                                '
  ReplyToQMgr    : 'LOCALQM                                         '
  ** Identity Context
    :
****   Message      ****

 length - 1242 bytes

00000000:  5044 4D51 0200 0200 6800 0000 6800 0000 'PDMQ....h...h...'
00000010:  0800 0000 B501 0000 1100 0000 0000 0000 '................'
00000020:  4D51 5354 5220 2020 0000 0000 0000 0000 'MQSTR   ........'
00000030:  0000 0000 0000 0000 0000 0000 0000 0000 '................'
00000040:  0000 0000 0000 0000 0000 0000 0000 0000 '................'
00000050:  0000 0000 0000 0000 0000 0000 0000 0000 '................'
00000060:  0000 0000 0000 0000 3082 046E 0609 2A86 '........0é.n.   `
00000070:  4886 F70D 0107 03A0 8204 5F30 8204 5B02 'Hå.....áé._0é.[.'
00000080:  0100 3181 D730 81D4 0201 0030 3D30 3131 '..1ü.0ü....0=011'
00000090:  0B30 0906 0355 0406 1302 4A50 3111 300F '.0   ..U....JP1.0.
000000A0:  0603 5504 0A13 0843 6174 556E 696F 6E31 '..U....CatUnion1'
000000B0:  0F30 0D06 0355 0403 1306 4769 6E67 6572 '.0...U....Ginger'
000000C0:  0208 C1C7 B970 3999 57D6 300D 0609 2A86 '.....p9ÖW.0..
000000D0:  4886 F70D 0101 0105 0004 8180 649E 822A 'Hå........üÇd.é*'
000000E0:  C090 A27B 16BE E9BD 916C 8F50 C239 5B9E '.Éó{..Θ.ælÅP.9[.'
000000F0:  5C87 4F22 2A9F 0839 6B9D C11C 27B9 53D3 '\çO"*ƒ.9k¥..'.S.'
00000100:  2AC3 C929 B5D8 FB71 4D2B 8F39 A8B2 381D '*..)...qM+Å9¿.8.'
00000110:  31C8 C29D 7608 0891 D6B8 744B 8012 A9DF '1..¥v..æ..tKÇ...'
```

DLQ of encrypted message (via Q alias)

```
c:\result>amqsbcg SYSTEM.PROTECTION.ERROR.QUEUE LOCALQM

AMQSBCG0 - starts here
**********************

 MQOPEN - 'SYSTEM.PROTECTION.ERROR.QUEUE'


 MQGET of message number 1
****Message descriptor****

  StrucId  : 'MD  '  Version : 2
  Report   : 0  MsgType : 8
  Expiry   : -1  Feedback : 0
  Encoding : 546  CodedCharSetId : 437
   :
****   Message      ****

 length - 1398 bytes

00000000:  444C 4820 0100 0000 0F08 0000 5345 4352 'DLH ........SECR'
00000010:  4554 2E51 0000 0000 0000 0000 0000 0000 'ET.Q............'
00000020:  0000 0000 0000 0000 0000 0000 0000 0000 '................'
00000030:  0000 0000 0000 0000 0000 0000 4C4F 4341 '............LOCA'
00000040:  4C51 4D20 2020 2020 2020 2020 2020 2020 'LQM             '
00000050:  2020 2020 2020 2020 2020 2020 2020 2020 '                '
00000060:  2020 2020 2020 2020 2020 2020 2202 0000 '            "...'
00000070:  B501 0000 2020 2020 2020 2020 0B00 0000 '....        ....'
00000080:  433A 5C57 4D51 5637 5C62 696E 5C61 6D71 'C:\WMQV7\bin\amq'
00000090:  7370 7574 2E65 7865 2020 2020 3230 3130 'sput.exe    2010'
000000A0:  3131 3135 3136 3236 3533 3530 5044 4D51 '11151626535 0PDMQ'
000000B0:  0200 0200 6800 0000 6800 0000 0800 0000 '....h...h.......'
000000C0:  B501 0000 0600 0000 0000 0000 4D51 5354 '............MQST'
000000D0:  5220 2020 0000 0000 0000 0000 0000 0000 'R   ............'
000000E0:  0000 0000 0000 0000 0000 0000 0000 0000 '................'
000000F0:  0000 0000 0000 0000 0000 0000 0000 0000 '................'
00000100:  0000 0000 0000 0000 0000 0000 0000 0000 '................'
00000110:  0000 0000 3082 045E 0609 2A86 4886 F70D '....0é.^.   *åHå..'
00000120:  0107 03A0 8204 4F30 8204 4B02 0100 3181 '...áé.O0é.K...1ü'
00000130:  D730 81D4 0201 0030 3D30 3131 0B30 0906 '.0ü....0=011.0
00000140:  0355 0406 1302 4A50 3111 300F 0603 5504 '.U....JP1.0...U.'
00000150:  0A13 0843 6174 556E 696F 6E31 0F30 0D06 '...CatUnion1.0..'
00000160:  0355 0403 1306 4769 6E67 6572 0208 C1C7 '.U....Ginger....'
00000170:  B970 3999 57D6 300D 0609 2A86 4886 F70D '.p9ÖW.0..   *åHå..'
```

Performance – AMS Confidentiality vs Privacy





- Tests on Distributed MQ
- 2K Persistent Message
- 20 Requesters

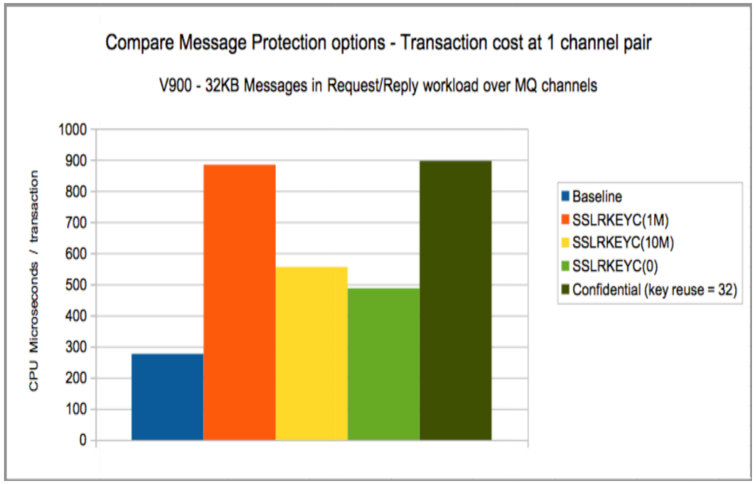
For additional performance figures, see IBM MQ SupportPac MP1J (v8) and GitHub for AMS z/OS (v9.0.1)

AMS z/OS performance with v9.0.1

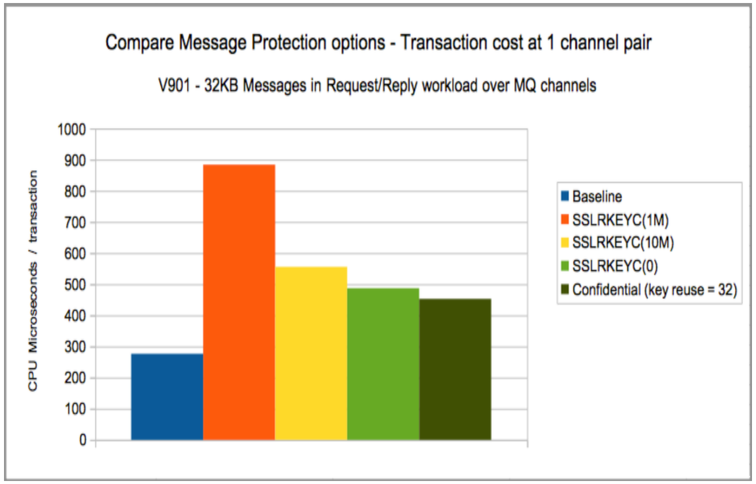
Highlight from https://ibm-messaging.github.io/mqperf/V901.pdf

"Comparing AMS on V901 with AMS on V900:

- *Transactions protected by AMS Integrity policies can see a cost of less than half of similar transactions run against V900, with a throughput improvements in excess of 30%.*
- *Transactions protected by AMS Privacy policies can achieve a cost of 40% those of similar transactions in V900, with a throughput improvement in excess of 30%.*
- *Transactions protected by AMS Confidential policies can achieve a transaction cost of 16% those of similar transactions in V900, with a throughput improvement in excess of 5 times."*



MQ for z/OS 9.0.0



MQ for z/OS 9.0.1

Known limitations today

- **Pub/Sub** is not supported today
- **Channel data conversion** is not supported
- **Distribution lists** are not supported
- **IMS Bridge** not supported (nor IMS programs in SRB mode)
- Non-threaded applications using API exit on HP-UX
- AMS does not protect MQ **message properties** (introduced in MQv7)
- IBM MQ classes for .NET in a **managed mode** not supported
- Note that AMS increases message length
  – New Message Size = 1280 + [old msg length] + (200 x [# of recipients])
- AMS usage will increase CPU requirements
- Some additional restrictions for older MQ versions

See IBM MQ Knowledge Center "q014670_" for full list of restrictions

Summary

IBM MQ Advanced Message Security

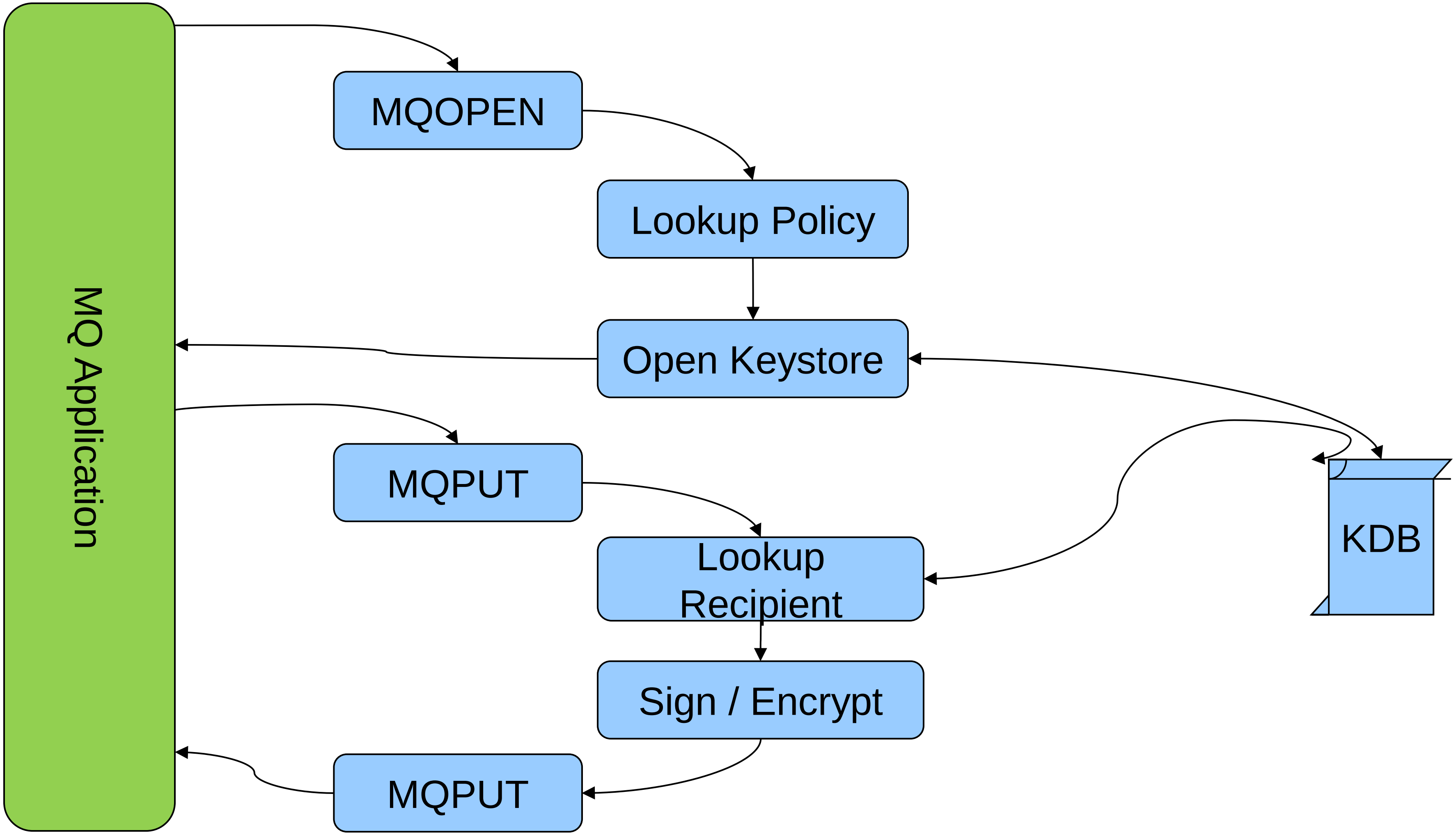
- Additional message protection above and beyond standard MQ security
- Offers various levels of message protection
- For all supported versions of MQ
- Supports MQ Server, MQ Client and JMS
- “Light weight” product - No pre-requisites, easy installation, easy configuration
- Existing MQ applications do not require changes

Backup

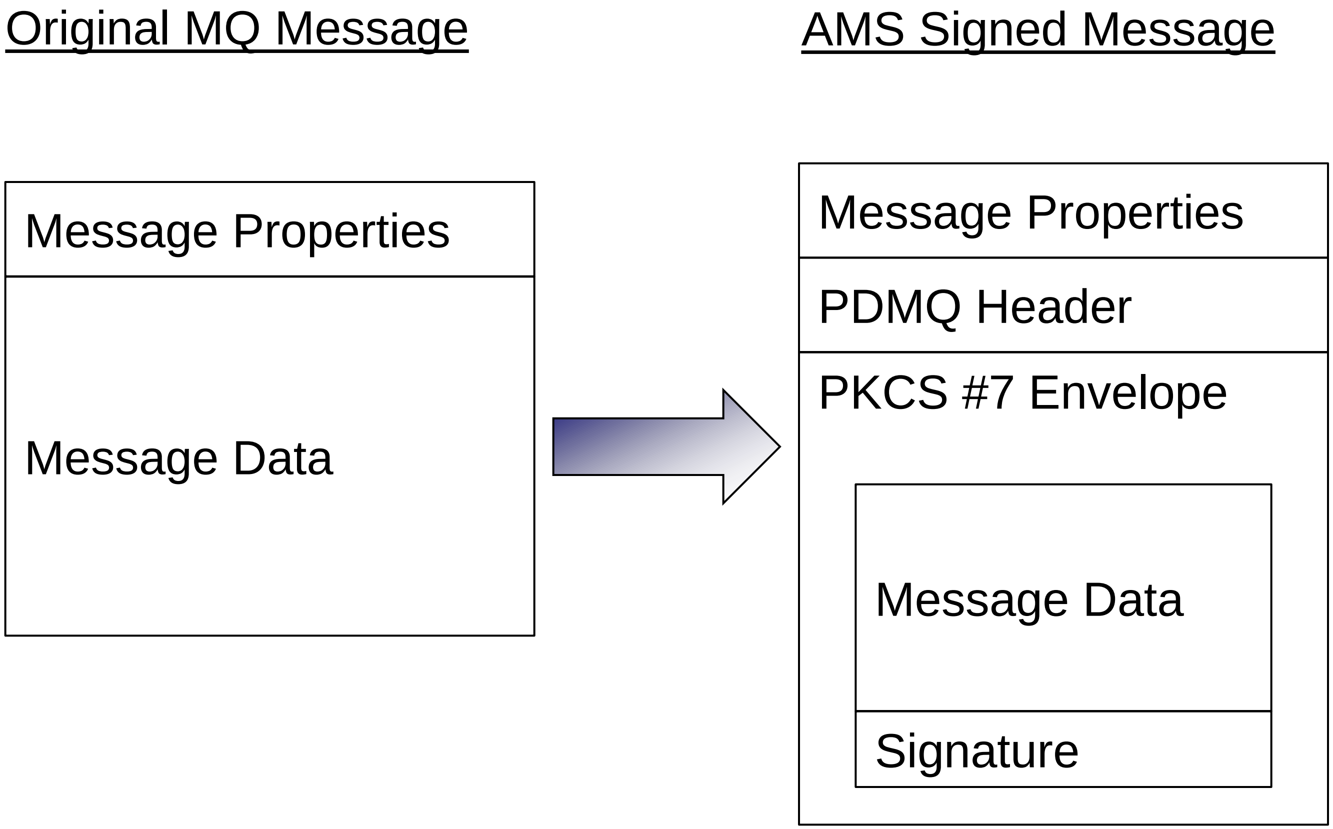
Bibliography

- IBM MQ Knowledge Center, AMS at "q014580_"
- Program Directory for IBM IBM MQ Advanced Message Security for z/OS (GI13-0559)
- IBM MQ AMS Administration Guide (GC34-7142)
- Secure Messaging Scenarios with WebSphere MQ (SG24-8069)
- SA22-7687 z/OS V1R13.0 Security Server RACF Command Language Reference
- SC24-5901 z/ OS Cryptographic Services System Secure Sockets Layer (SSL) Programming.
  - For GSK runtime error codes, you can also see online version at http://pic.dhe.ibm.com/infocenter/zos/v1r12/topic/com.ibm.zos.r12.csf/csf.htm

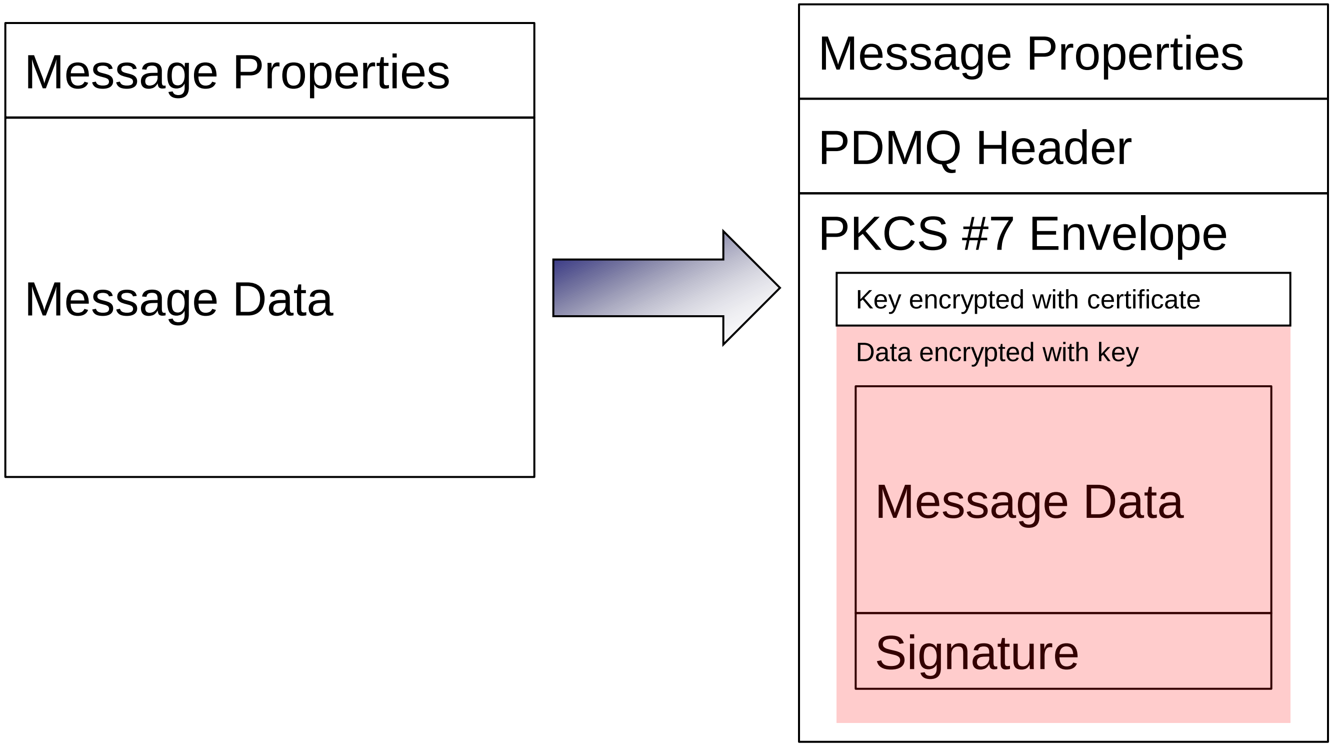
MQ AMS Process



IBM MQ AMS : Integrity Message Format



IBM MQ AMS : Privacy Message Format



Original MQ Message

Message Properties

Message Data

AMS Encrypted Message

Message Properties

PDMQ Header

PKCS #7 Envelope

Key encrypted with certificate

Data encrypted with key

Message Data

Signature