#SHAREorg

SHARE

WebSphere MQ
Advanced Message Security

Mark Taylor
IBM Hursley

March 2012
Session 10539

Universal Messaging Backbone

*Dynamic network that delivers the **data** you require from wherever it resides to wherever you want it in whatever way you want it at whatever time you want it*

Universal Messaging

1. Best Delivery
- Choice of service
- Resilience, Integrity, Security
- Throughput, Latency
- High availability

2. Anything Anywhere
- Any skills
- Any traffic
- Any language
- Any environment
- Any platform

3. Scale Dynamically
- Start small
- Grow incrementally
- Stretch elastically
- Scale admin

SHARE in Atlanta 2012

Securing the UMB

- Traditionally, WMQ offers:
  - Integration with operating system security e.g. file/directory/user access
  - Object-level access security via the Object Authority Manager
  - Channel encryption
  - Channel authorisation with certificates
- Some applications require higher degrees of security for message data, for example where regulatory compliance rules apply
- Useful to offer an extension to the MQ family offering this capability
  - Aim to be non-invasive to applications
  - Simple to install
  - Straightforward to configure
  - Use industry standards for encryption

Universal Messaging

SHARE in Atlanta 2012

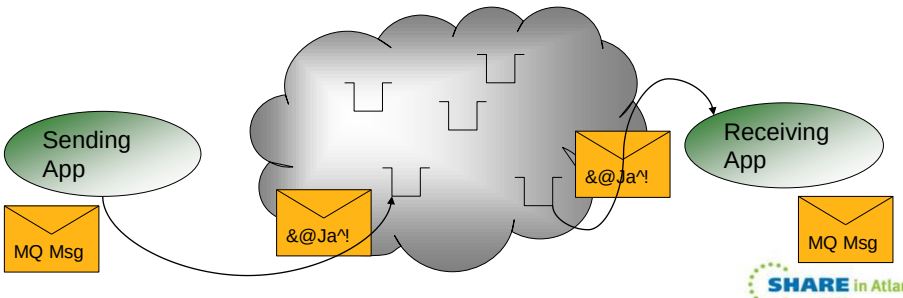
WMQ Advanced Message Security

- New product - WMQ Advanced Message Security V7.0.1
  - Available Oct 8, 2010
- Enhances WMQ security processing
  - Provides additional security services over and above base QM
  - Designed to assist with requirements such as PCI DSS compliance
- Application ---> Application protection for point-to-point messaging
  - Sometimes called “end-to-end” or “message-level” protection
- Simplifies regulatory compliance (PCI, HIPAA, etc.) for audit & privacy
- Protects messages even when messages are “at rest”
  - Messages protected from original putter to final getter
  - Messages protected when on a queue and in logs

SHARE in Atlanta 2012

Message Level Protection

- Enables secure message transfers at application level
- Assurance that messages have not been altered in transit
  - When issuing payment information messages, ensure the payment amount does not change before reaching the receiver
- Assurance that messages originated from the expected source
  - When processing messages, validate the sender
- Assurance that messages can only be viewed by intended recipient(s)
  - When sending confidential information.



Which Messages are Secured

- Not all messages are equal
- May have …
  - Command and control scenarios
  - Unimportant "status update"
  - Data subject to auditory controls
  - Data subject to standards compliance
  - Credit card data protected by PCI
  - Confidential government data
- Expectation that only limited queues are protected on each qmgr
- System architecture designs need to consider message content

WMQ AMS - Key Features

- Secures sensitive messages
- Detects and removes rogue or unauthorized messages before they are processed by receiving applications
- Verifies that messages are not modified in transit from queue to queue
- Protects messages not only when they flow across the network but when they are at rest in queues
  - Cannot view message contents in logs or queues
- Messages from existing applications are transparently secured
  - No changes needed to existing applications
- Industry standard asymmetric cryptography used to protect messages
  - Uses Public Key Infrastructure (PKI) to protect messages
  - Uses digital certificates (X.509) for applications

SHARE in Atlanta 2012

WMQ AMS – Simplicity and Integration

- No prereq products
  - Significantly simplified installation and configuration compared to predecessor product
  - Up and running in minutes …
- Works in conjunction with SSL
  - Can choose to use either or both depending on your requirements
- Works in conjunction with WMQ authorisation model (OAM and SAF)
- No changes required to WMQ applications
  - Works with local applications and clients, including Java
  - Support for WMQ V6 and V7
- No changes required to existing object definitions
- Fine-grained policies to define which queues are protected and how
- Administratively controlled policies
  - Command line
  - MQ Explorer

SHARE in Atlanta 2012

Platforms supported

- HP-UX Itanium
- HP-UX PA-RISC
- Linux for System p
- Linux for System x (32 bit and 64-bit)
- Linux for System z
- Solaris for Intel X86 (64-bit)
- Solaris for Sun SPARC
- AIX for System p
- Windows (32-bit and 64-bit)

- z/OS
  - CICS Bridge, IMS Bridge, IMS SRB apps are not supported

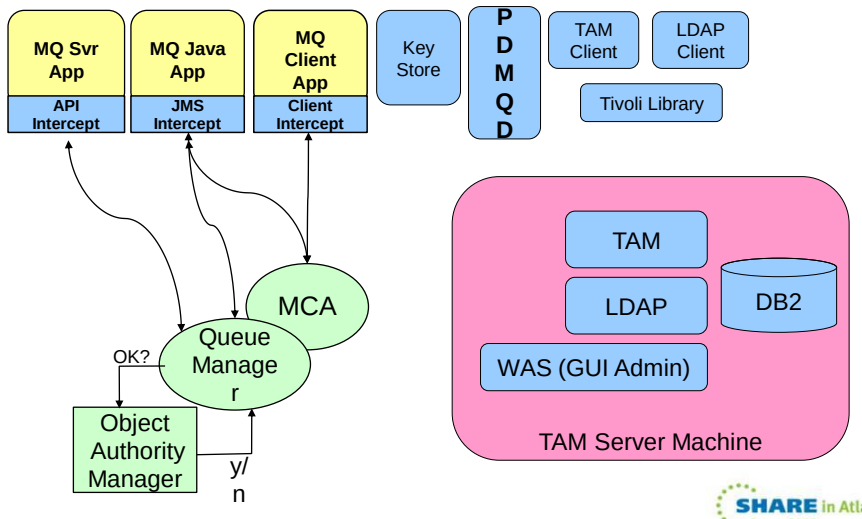
- Supports MQ6, MQ7, MQ7.1 queue managers (JMS requires V7 jars)

WMQ vs WMQ AMS

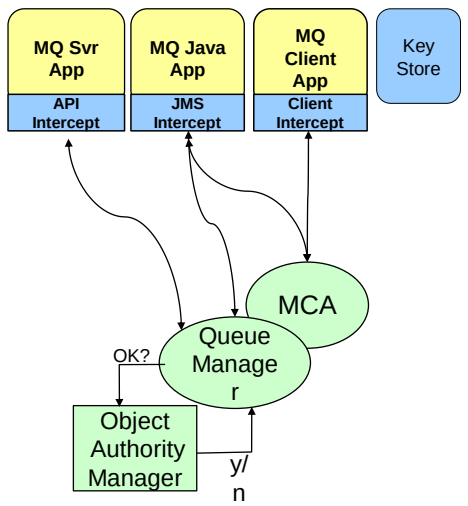
- WebSphere MQ
  - Authentication (OS for local apps or peer authenticated SSL for client apps)
  - Authorisation (OAM on distributed, SAF on z/OS)
  - Auditing (event messages)
  - Integrity (SSL for channels)
  - Privacy (SSL for channels)

- WebSphere MQ AMS
  - As above, additionally:
  - Integrity (Digital signature of message content)
  - Privacy (Message content encryption)

Architecture Changes - From ESE



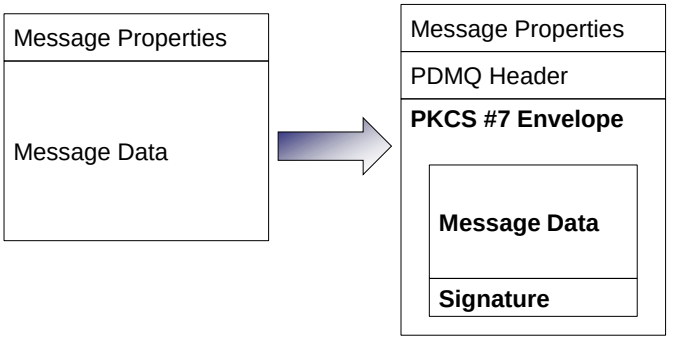
Architecture Changes - To AMS



WMQ AMS: Signed Message Format

- Original MQ Message

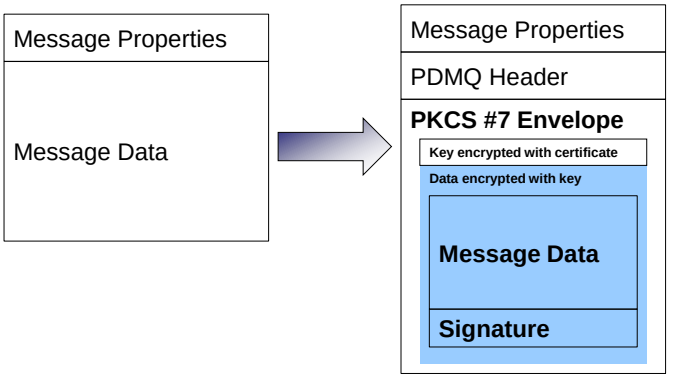
AMS Signed Message



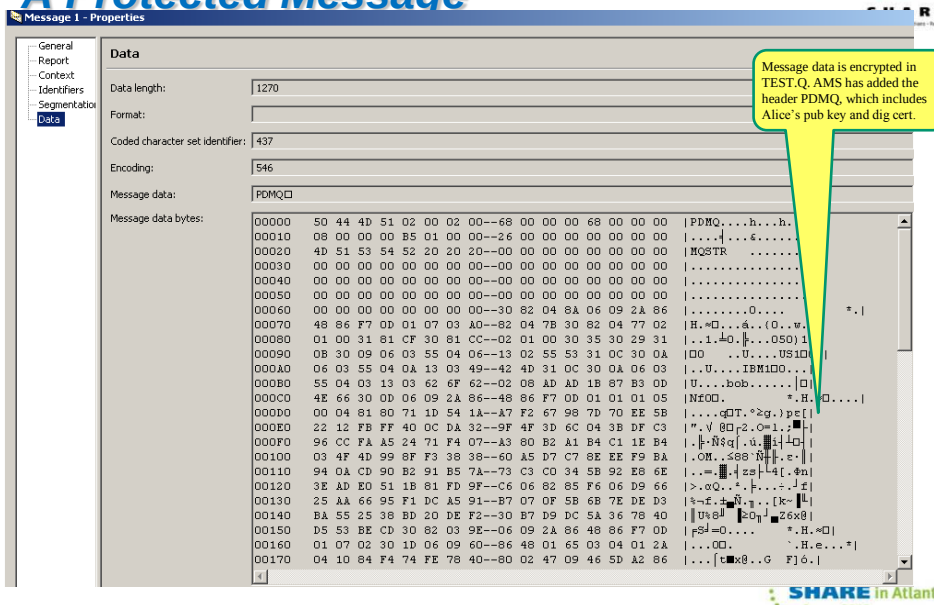
WMQ AMS: Privacy Message Format

- Original MQ Message

AMS Encrypted Message



A Protected Message



SHARE in Atlanta 2012

Protected Messages

- New message size is approximately …
  - 1280 + Original Length + (200*Recipient Count) bytes
- May affect max lengths configured on queues and channels
- Data conversion done by queue manager after protection removed
- Bad messages sent to SYSTEM.PROTECTION.ERROR.QUEUE
  - Sender did not have the authority to write to the queue
  - Sender's certificate was not valid
  - AMS was unable to decrypt the message
  - A policy mismatch occurred. For example, the sender used integrity instead of the expected quality of protection of privacy, or used the wrong algorithm
  - The message was sent without expected AMS protection
- Messages moved here have a DLH attached
  - So standard dead-letter handlers can process them

SHARE in Atlanta 2012

Message Protection Policies - Overview

- Created or updated or removed by command 'setmqspl'
  - Or by AMS plug-in for MQ Explorer (GUI)
- Policies are stored on queue SYSTEM.PROTECTION.POLICY.QUEUE
- Each protected queue can have only one associated policy
- Display policies with command 'dspmqspl'
  - Can be displayed in "setmqspl" format for easy backup/restore
- Applied based on queue name as opened by application
  - can deal with alias and remote queues

SHARE in Atlanta 2012

Message Protection Policies - Detail

- 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
- If encryption set to NONE, then only signing is done
- Toleration flag (-t) assists with phased introduction of AMS

```
setmqspl
-m <queue_manager>
-p <protected_queue_name>
-s <SHA1 | MD5>
-e <encryption algorithm>
-a <Authorized signer DN1>
-a <Authorized signer DN2>
-r <Message recipient DN1>
-r <Message recipient DN2>
-t <0|1>
```

SHARE in Atlanta 2012

Message Protection Policies - Example

- This policy enforces privacy protection (signature and encryption) for messages put on queue Q.PRIVACY in queue manager QM
- The message signing algorithm is SHA1.
- The message encryption algorithm is AES128
- Two message recipients are listed using their certificates DN

- Messages retrieved by un-authorized recipients sends messages to SYSTEM.PROTECTION.ERROR.QUEUE

```
setmqspl
-m QM
-p Q.PRIVACY
-s SHA1
-e AES128
-r 'CN=pdmqss,O=tivoli,C=US'
-r 'CN=Vicente
  Suarez,OU=ISSW,O=IBM,
  L=Hursley,C=GB'
```

Publish/Subscribe with AMS

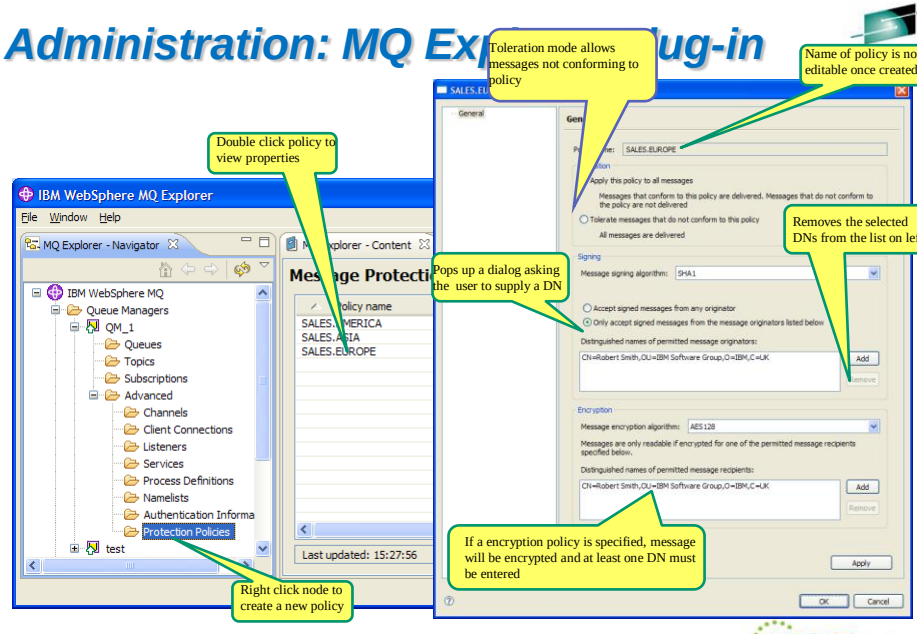
- AMS does not directly support MQv7 publish/subscribe features

- Main reason for this is the decoupling of publisher from subscriber
  - The publisher does not know who the recipients are going to be
  - Dynamic changes to subscription list
  - Only the queue manager knows – and does not have access to publisher's certificates

- However, a degree of support is possible
  - Use QALIAS to point to a TOPIC
  - Set a policy on the QALIAS that lists all authorised subscribers
  - More like a distribution list but OK for some scenarios

- Question: what would user requirements be for greater pub/sub?
  - Signed messages only?
  - Using qmgr credentials sometimes, but not publisher?
  - How dynamic?

Administration: MQ Exp ug-in



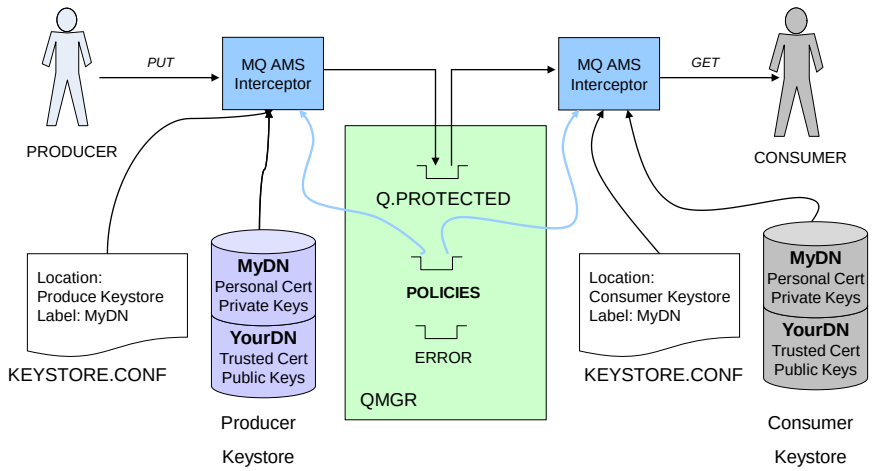
Keystores and X.509 Certificates

- 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 must also contain trusted certificates to validate message signers or to obtain the public keys of encrypted message recipients
- Several types of keystore are supported: CMS, JKS and JCEKS.

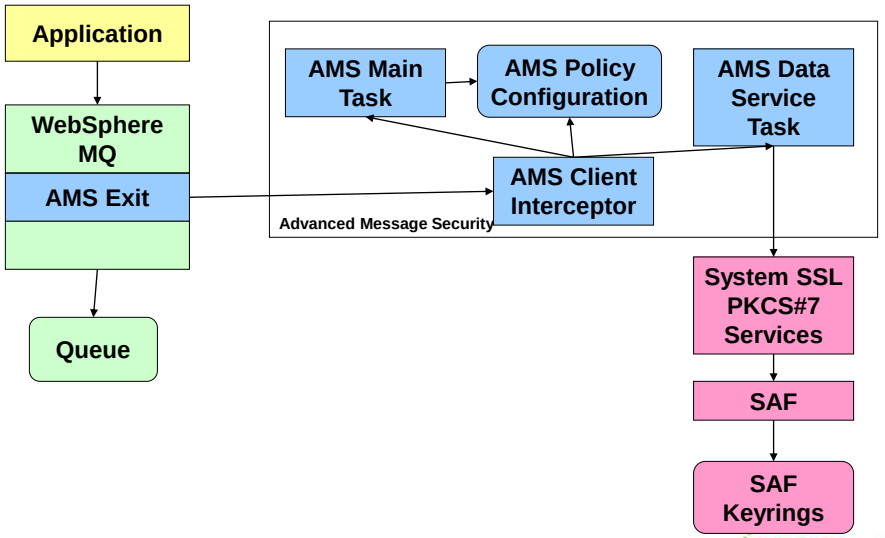
AMS Configuration Files

- Each user of AMS requires a configuration file.
  - Type of keystore: CMS (for C programs) and JKS, JCEKS (for Java)
  - Location of the keystore
  - Label of the personal certificate
  - Passwords to access keystore and private keys
    - Password can be encrypted in the configuration file
- Configuration file located using one of the following methods:
  - Environment variable MQS_KEYSTORE_CONF=<path to conf file>
    - MQS_KEYSTORE_CONF=C:\Documents and Settings\Bob\AMS\keystore.conf
  - Checking default locations and file names
    - Platform dependent. For example in UNIX: "$HOME/.mqs/keystore.conf"
- Configuration file should be secured with OS permissions
- Also a configuration file ("routing file") for logging and tracing

Architecture - Distributed Platforms



Architecture – z/OS



AMS Interceptors

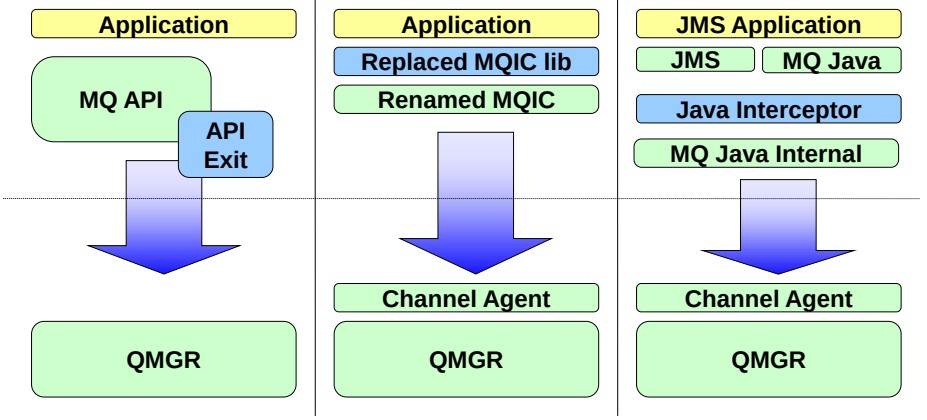
- AMS functionality is implemented in interceptors.
  - There are no long running processes or daemons (Except in z/OS).
- Existing MQ applications do not require changes.
- Three interceptors are provided
  - Server interceptor for local (bindings mode) MQI API and Java applications.
    - Implemented as queue manager API exit.
  - MQI API client interceptor for remote (client mode) MQ API applications.
    - MQ AMS interceptor imbedded in MQ client code.
  - 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.
    - MQ V7.0 java client required.
    - SupportPac MQC7 WebSphere MQ V7.0 clients.
- Scripts provided to install and configure these interceptors
  - For example, update qm.ini for the API Exit

Interceptors

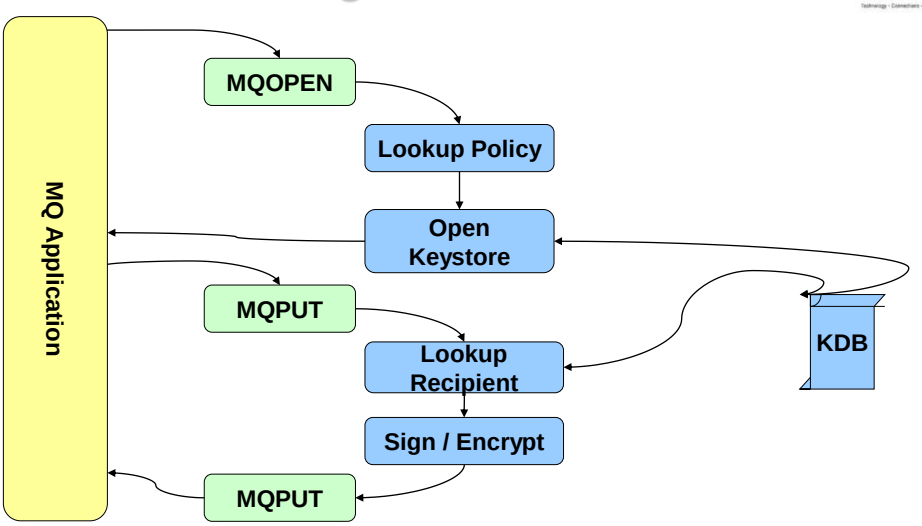
Server
- API Exit

Client
- Library Replacement

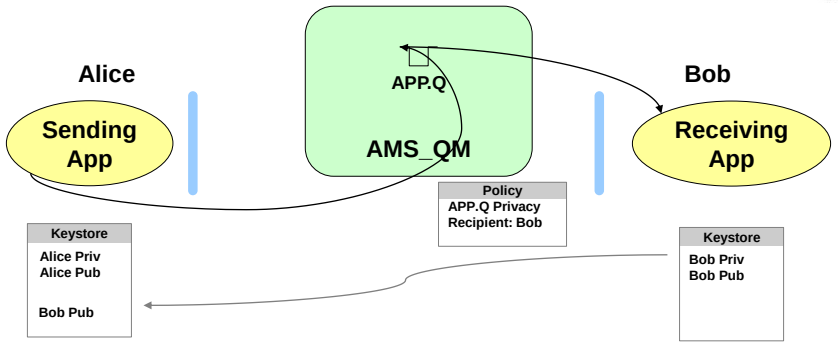
JMS and Java
- Java Interceptor



AMS Processing

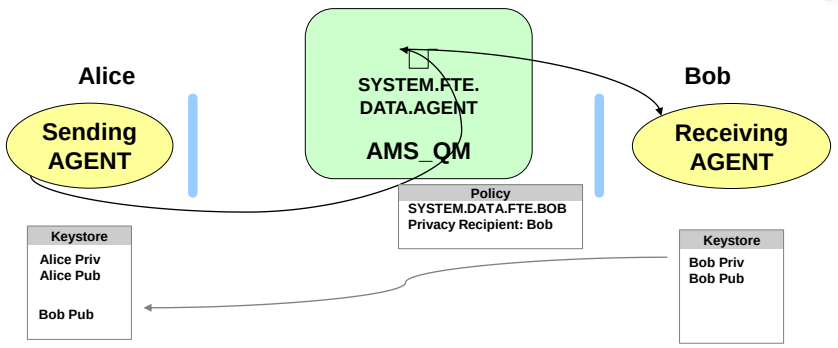


WMQ AMS Deployment



1.Install AMS Interceptor
2.Create public / private key pairs
3.Copy recipient's public key
4.Define protection policy for queues

WebSphere MQ AMS and FTE



1.Install AMS Interceptor
2.Create public / private key pairs
3.Copy recipient's public key
4.Define protection policy for queues

Using Message Broker with AMS

- Remember that messages can only be read by authorised applications
- If MB used purely as a router, then it does not need to decrypt messages
  - Can do true end-to-end protection
  - MQ Input and Output queues do not need policy settings
- If MB does work based on message content, or changes content, then it has to be considered an endpoint for AMS
  - "End-to-middle" protection
  - Still achieves goal of no unprotected message data on queues or in logs
- Many MB scenarios only have MQ on one side of a flow
  - Security for other protocols can be done by MB eg WS-Security

Responding to Regulatory Compliance

Large Food & Drug Retailer in North America	➢ Company had exposure to loss of customer personal healthcare information and personal credit card data ➢ A level 1 retailer with large volumes of personal data to deal with the need to secure their systems across multi-channels

Solution:
- Implementing WMQ AMS for encryption of data at rest in queues.
- WebSphere DataPower XS40 for firewall and data encryption for data in motion.

Solution Benefit:
- No need to modify applications, able to leave existing systems intact and add security updates quickly at the same time as continuing normal operation.
- By encrypting the data and limiting access to the applications the possibility of personal data being stolen and will be minimized.

V7.0.1.2 Enhancements

- Available January 2012
- Supports WMQ V7.1
  - Extends WMQ V7.1 Application Activity Trace to show applied AMS policy
- Supports SHA-2 Digest algorithms
- Provides Command and Configuration Events for Policy changes
  - Audit trail of who has changed configuration

SOA Sandbox for AMS discovery

- Try AMS and see what it can do for you
  - http://www.ibm.com/developerworks/downloads/soasandbox/mqsecurity.html
- SOA Sandbox main page for offerings designed to give you hands-on experience of various IBM products without having to install them
  - http://www.ibm.com/developerworks/downloads/soasandbox/.html

3

Summary

- WebSphere MQ Advanced Message Security V7.0.1
- Simplifies regulatory compliance
- Provides additional security over and above base MQ
- Complements (does not replace) existing MQ security
- Works with all levels of MQ in service (MQ 6 & 7)
- Does not require application changes
- Policies applied on individual queues