Secure Your Messages with IBM MQ Advanced Message Security

Robert Parker
parrobe@uk.ibm.com

Agenda

- Message Level Security
- Digital Cryptography 101 (Keys, hashes, Alice & Bob)
- WebSphere MQ Advanced Message Security Introduction
- Administration
- Architecture
- Behaviour
- Performance
- Implementing AMS

Message Level Security – Where to use it?

- “Valuable” messages
  - In flight on the network
  - At rest, on disk
  - Monitoring and control messages

- Large networks, difficult to prove security of messages
  - Injection
  - Modification
  - Unauthorized viewing

Message Level Security – Where to use it?

- Data subject to standards compliance (PCI, HIPAA, etc)
  - Credit card data protected by PCI
  - Confidential government data
  - Personal information e.g. healthcare
  - Data at rest, administrative privileges, etc

Message Level Security - Requirements

- 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 control messages, validate the sender

- Assurance that messages can only be viewed by intended recipient(s)
  - When sending confidential information

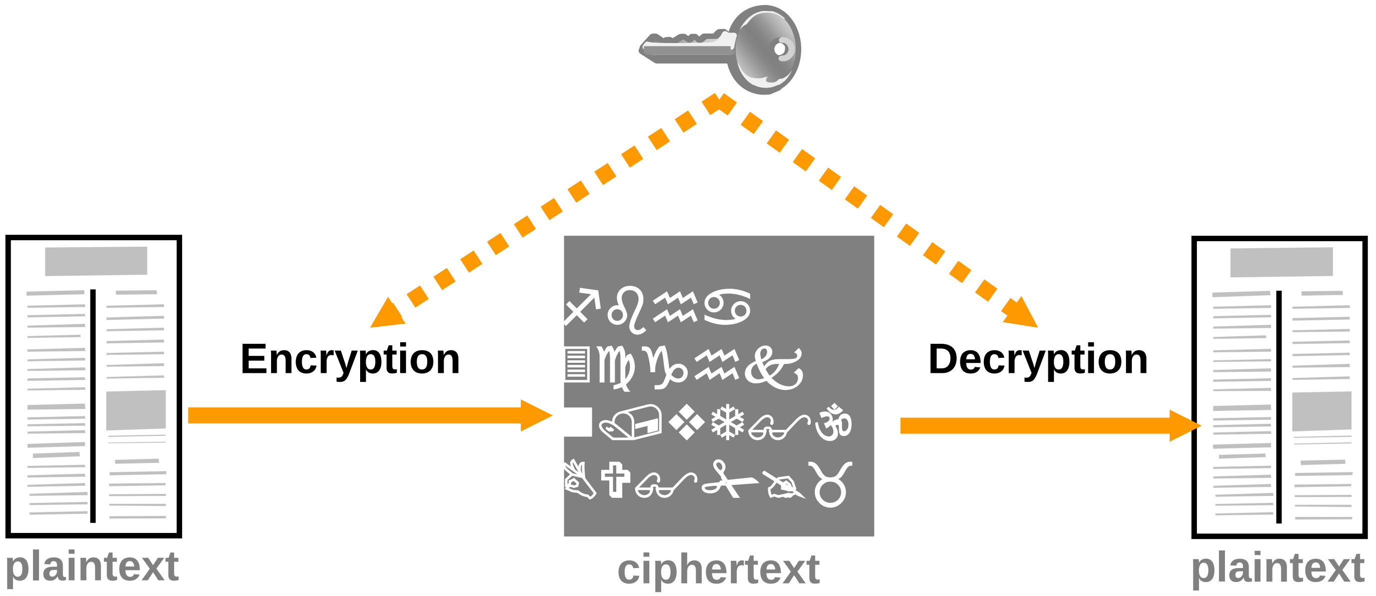
Digital Cryptography 101

Cryptography Choices

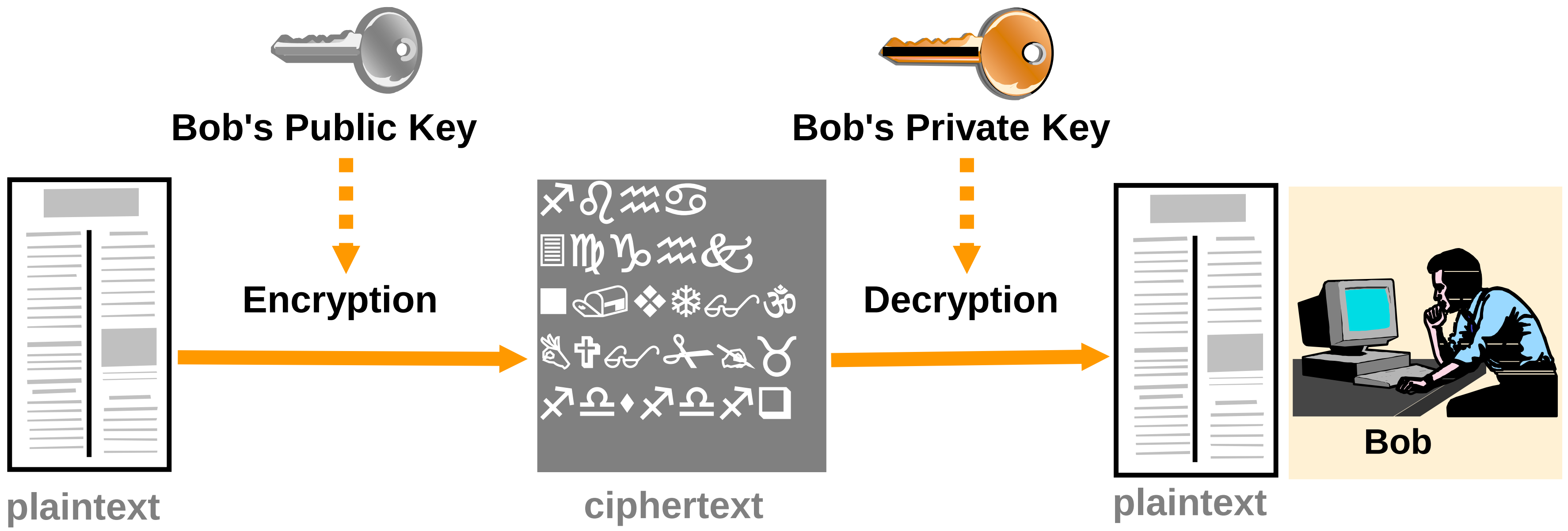
- Symmetric Key
  - Single secret key
  - Relatively fast
  - Poses key distribution challenges when faced with large numbers of senders/receivers
  - The key has to be known by the sender and receiver

- Asymmetric Keys
  - Private & Public key pairing
  - Message encrypted with one key can only be decrypted by the other one
  - Slower than symmetric key cryptography
  - Asymmetric Keys can be used to solve the key distribution challenges associated with symmetric keys

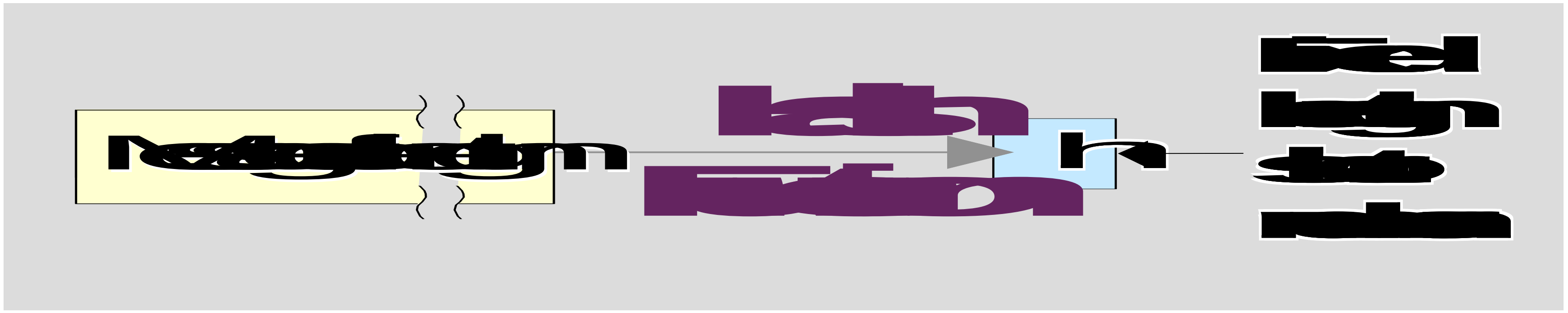
Symmetric Key Cryptography



Asymmetric Key Cryptography

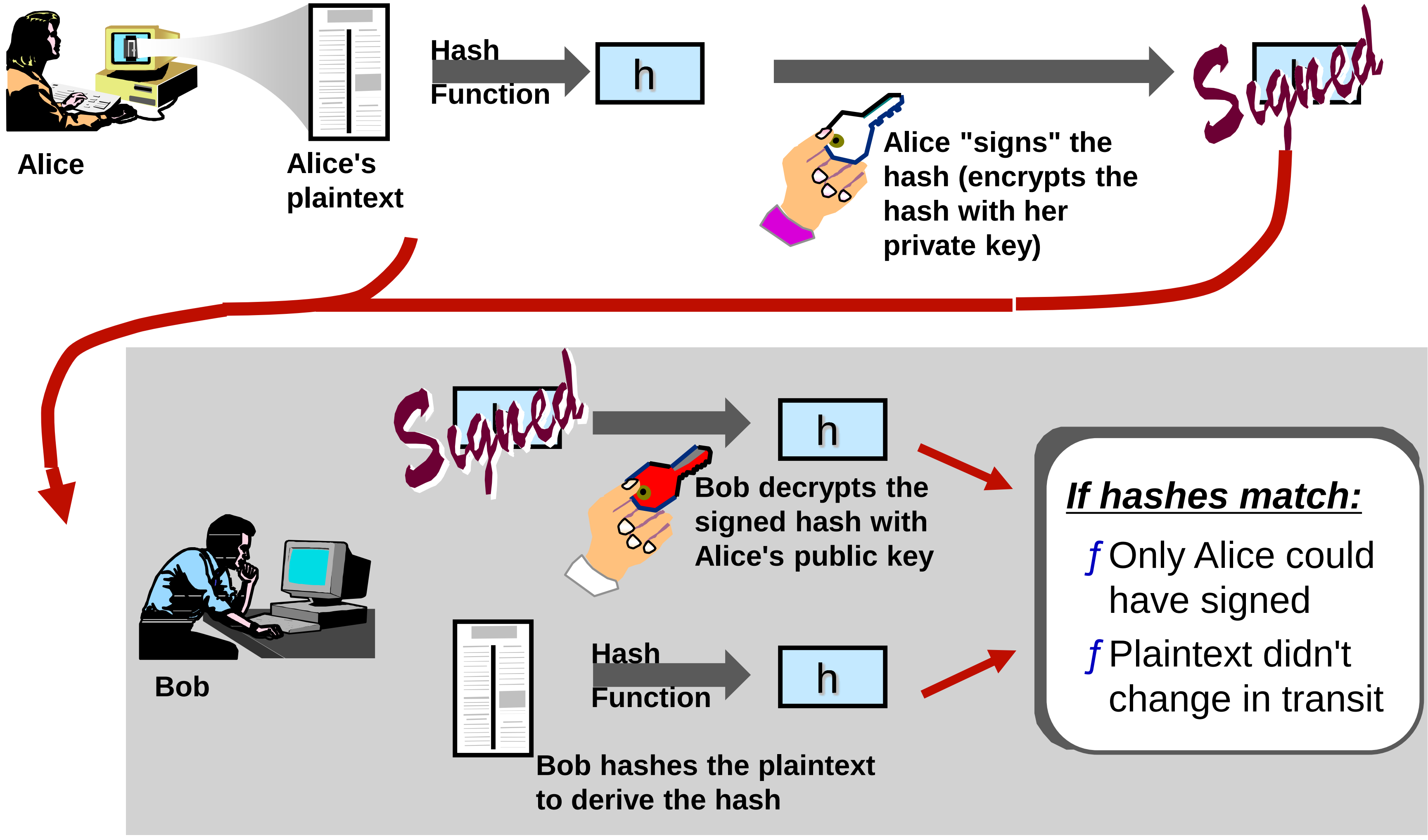


Hash Functions



- Hash Function
  - Computes the message MAC (Message Authentication Code)
  - Easy to compute
  - Very difficult to reverse
  - Computationally infeasible to find two messages that hash to the same value

Digital Signatures

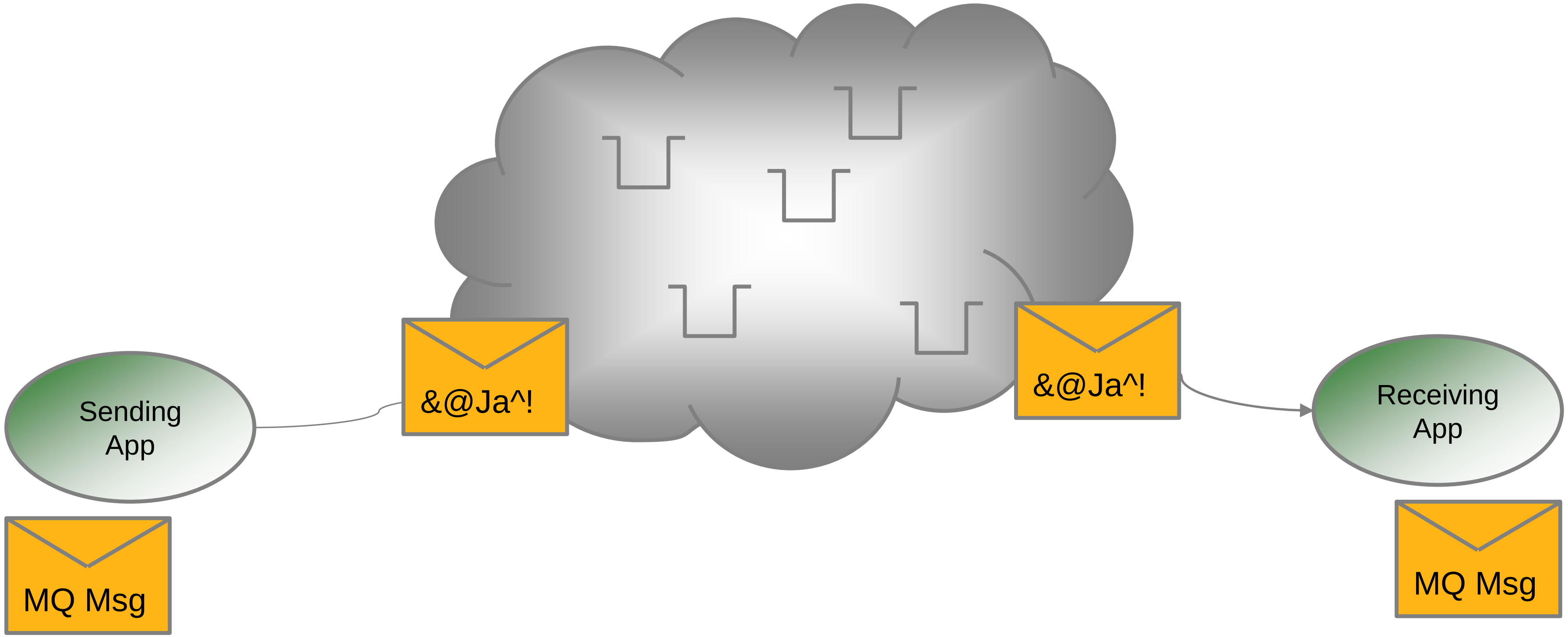

Alice
Alice's plaintext
Hash Function
h
Signed
Alice "signs" the hash (encrypts the hash with her private key)
Signed
h
Bob decrypts the signed hash with Alice's public key
Bob
Hash Function
h
Bob hashes the plaintext to derive the hash

If hashes match:

- Only Alice could have signed
- Plaintext didn't change in transit

AMS Introduction

WebSphere MQ Advanced Message Security



WebSphere MQ Advanced Message Security – Key points

- Provides additional security to that provided by base MQ
- End-to-end security, message level protection
  - A security policy defines what protection should be applied to messages
  - AMS intercepts messages at “endpoints” and applies the policy
  - Well suited to point to point, can also protect publish/subscribe but...
  - … have to know the identity of the intended recipients ahead of operation
- Asymmetric cryptography used to protect each message
  - Integrity Policies prove message origin, content not changed
  - Privacy policies as per integrity plus each message encrypted with unique key

WebSphere MQ Advanced Message Security – Key points

- Non-invasive
  - No code changes or re-linking of applications
- Administrative interfaces for policy management
  - Command line
  - MQ Explorer (Security Policies - now a default plugin)

WebSphere MQ Advanced Message Security – Security Features

- AMS is an optional component of MQ, not a replacement to base MQ security

- WebSphere MQ base
  - Authentication (Local OS user id, SSL peer and CHLAUTH for channels)
  - Authorization (OAM and CHLAUTH on distributed, RACF on z/OS)
  - Integrity (SSL for channels)
  - Privacy (SSL for channels)

- WebSphere MQ Advanced Message Security
  - Integrity (End-to-end digital signing of messages)
  - Privacy (End-to-end message content encryption)

WebSphere MQ Advanced Message Security – Limitations

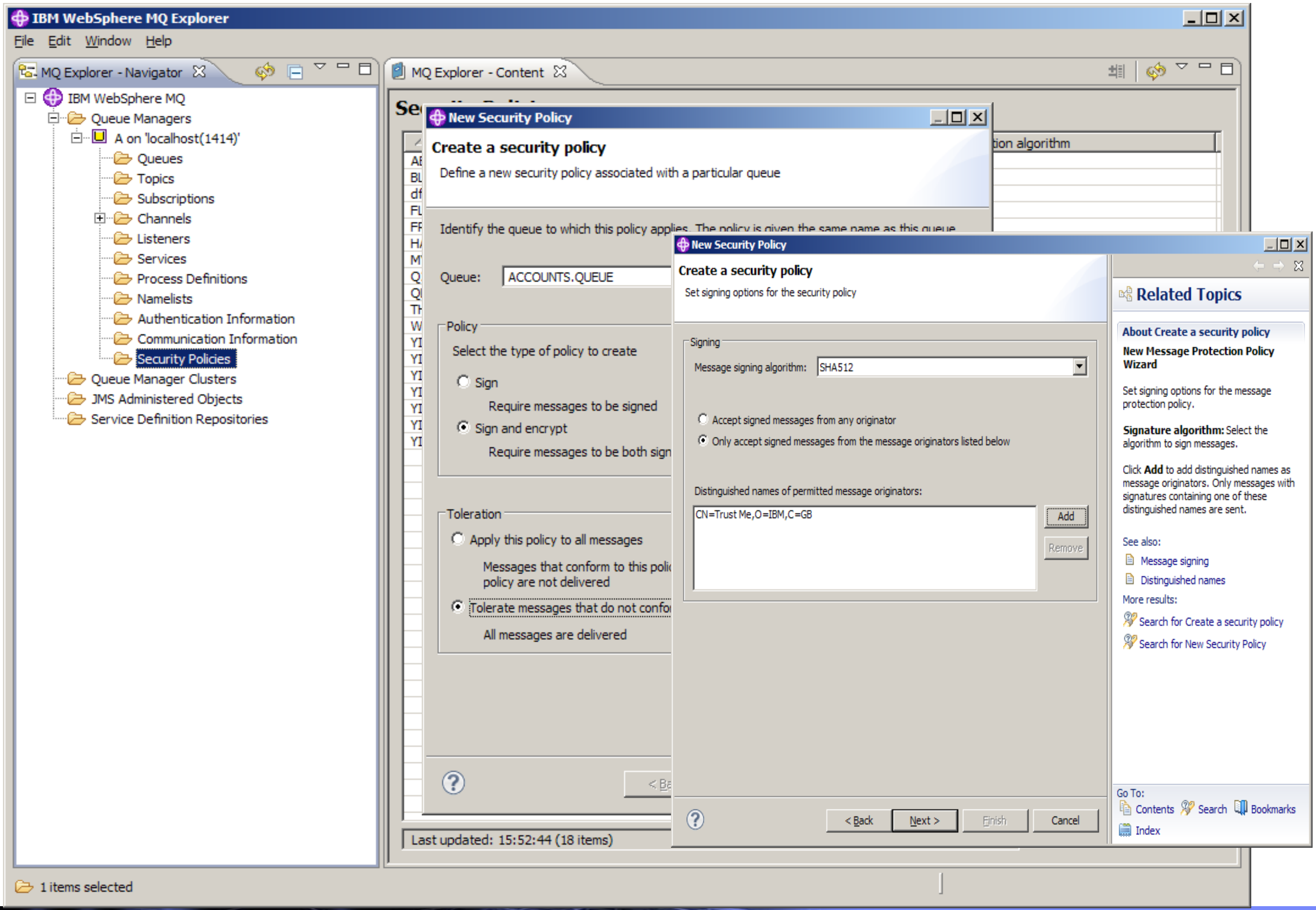
- The following MQ Options are not supported with AMS
  - Publish/Subscribe
  - Channel Data Conversion
  - Distribution lists

Administration

WebSphere MQ Advanced Message Security – Commands

- Command line tools
  - **setmqspl** : Set message protection policy
    - -m Queue manager
    - -p Policy name (matches queue name used in application)
    - -s Signing algorithm (MD5, SHA1, SHA256, SHA384, SHA512)
    - -a Authorised signers (Signed messages - DN list)
    - -e Encryption algorithm (RC2, DES, 3DES, AES128, AES256)
    - -r Message recipients (Encrypted messages - DN list)
  - **dspmqspl** : Display message protection policies
    - -m Queue manager
    - [-export]
    - [-p Policy name]

WebSphere MQ Advanced Message Security Security Policies in MQ Explorer



IBM WebSphere MQ Explorer

File Edit Window Help

MQ Explorer - Navigator

- IBM WebSphere MQ
  - Queue Managers
    - A on 'localhost(1414)'
      - Queues
      - Topics
      - Subscriptions
      - Channels
      - Listeners
      - Services
      - Process Definitions
      - Namelists
      - Authentication Information
      - Communication Information
      - Security Policies
  - Queue Manager Clusters
  - JMS Administered Objects
  - Service Definition Repositories

1 items selected

MQ Explorer - Content

Last updated: 15:52:44 (18 items)

New Security Policy

Create a security policy

Define a new security policy associated with a particular queue

Identify the queue to which this policy applies. The policy is given the same name as this queue

Queue: ACCOUNTS.QUEUE

Policy

Select the type of policy to create

( ) Sign

Require messages to be signed

[✓ Sign and encrypt]

Require messages to be both sign

Toleration

( ) Apply this policy to all messages

Messages that conform to this poli policy are not delivered

[✓ Tolerate messages that do not confo]

All messages are delivered

New Security Policy

Create a security policy

Set signing options for the security policy

Signing

Message signing algorithm: SHA512

( ) Accept signed messages from any originator

[✓ Only accept signed messages from the message originators listed below]

Distinguished names of permitted message originators:

CN=Trust Me,O=IBM,C=GB

Add | Remove

< Back | Next > | Finish | Cancel

Related Topics

About Create a security policy

New Message Protection Policy Wizard

Set signing options for the message protection policy.

Signature algorithm: Select the algorithm to sign messages.

Click **Add** to add distinguished names as message originators. Only messages with signatures containing one of these distinguished names are sent.

See also:

Message signing

Distinguished names

More results:

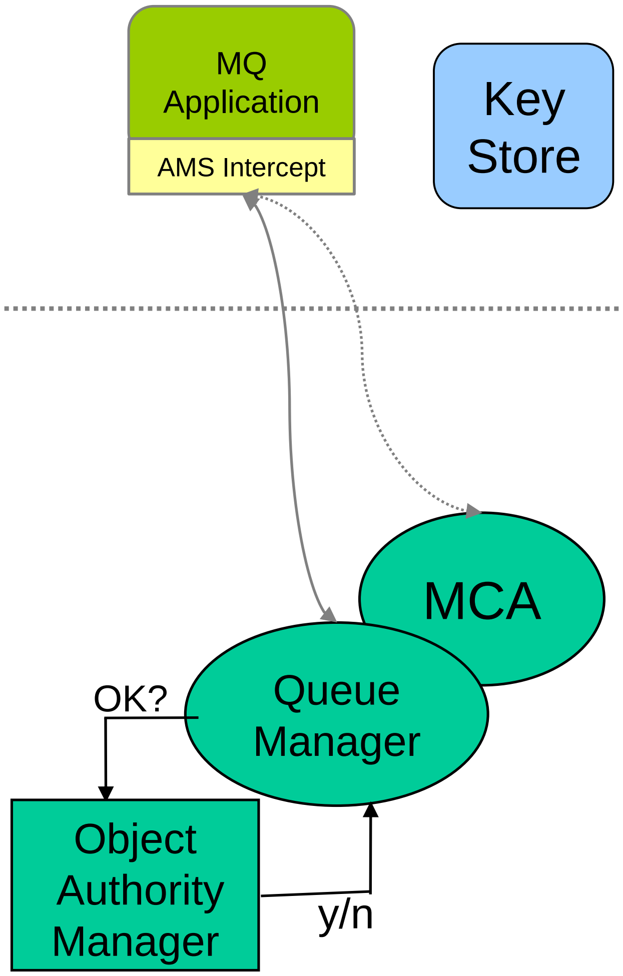
Search for Create a security policy

Search for New Security Policy

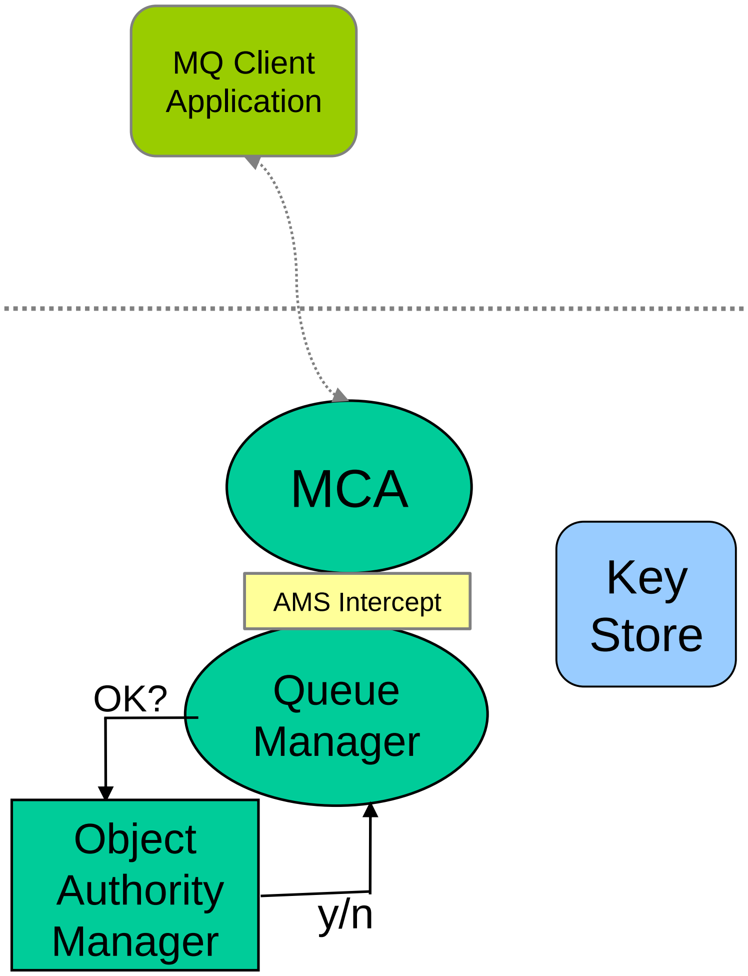
Go To: Contents Search Bookmarks Index

Architecture

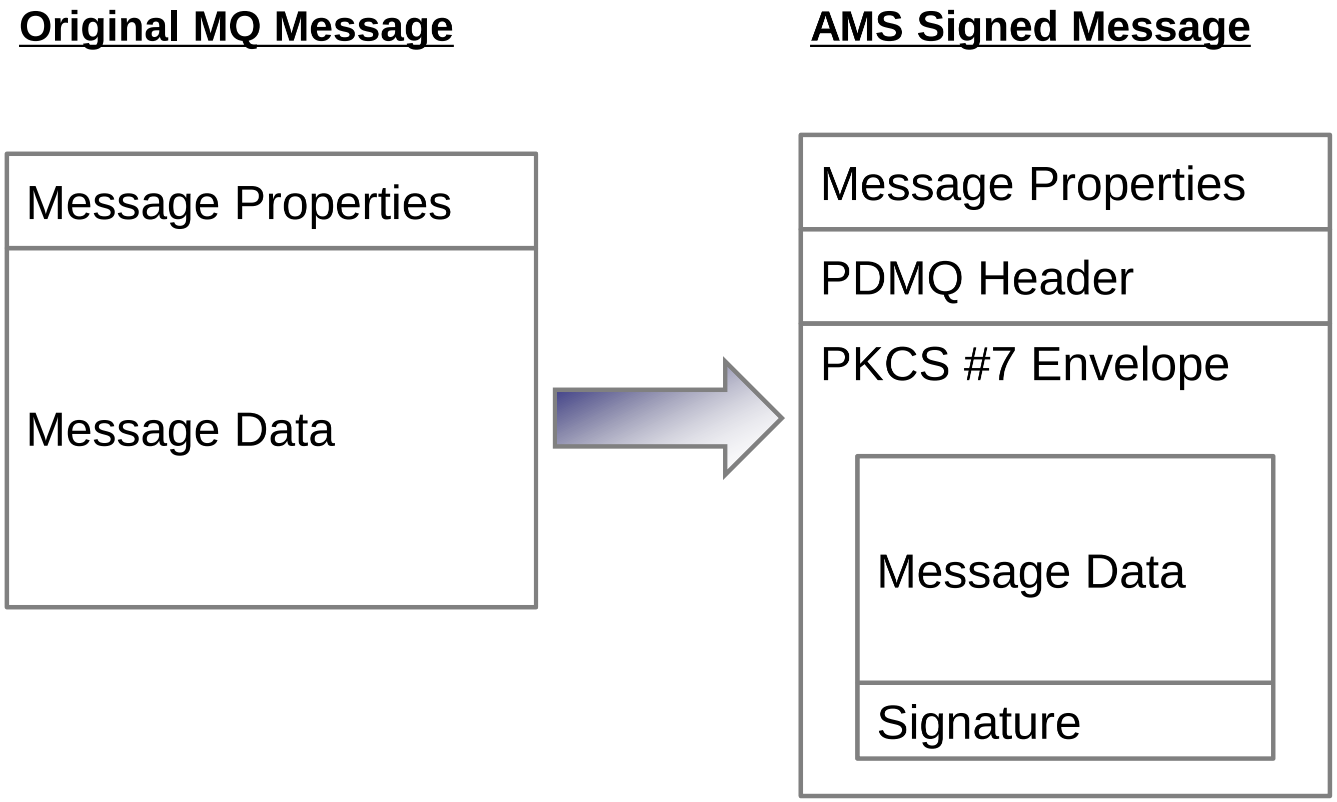
WebSphere MQ Advanced Message Security - Architecture



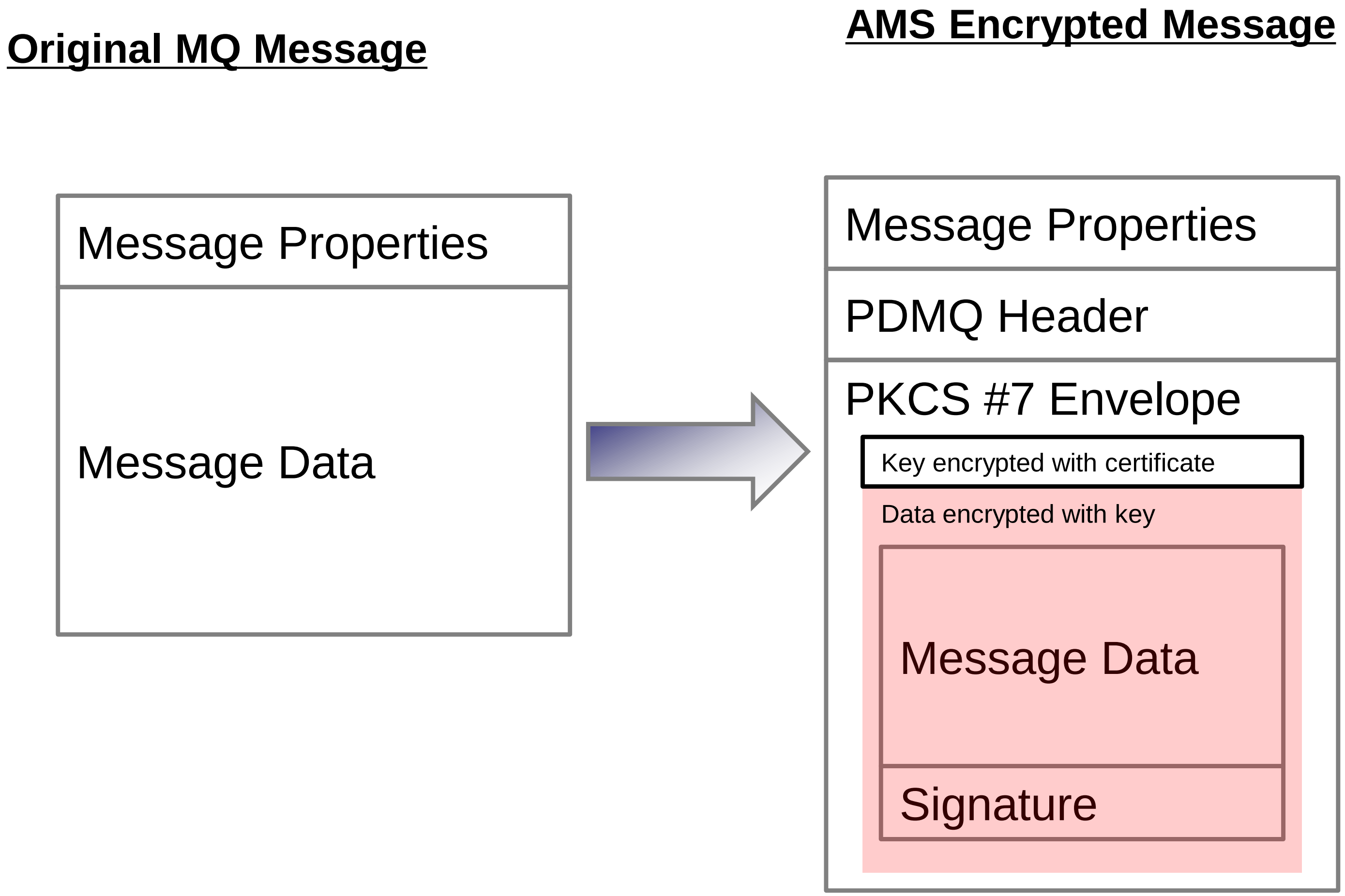
WebSphere MQ Advanced Message Security – Architecture (MCA Interception)



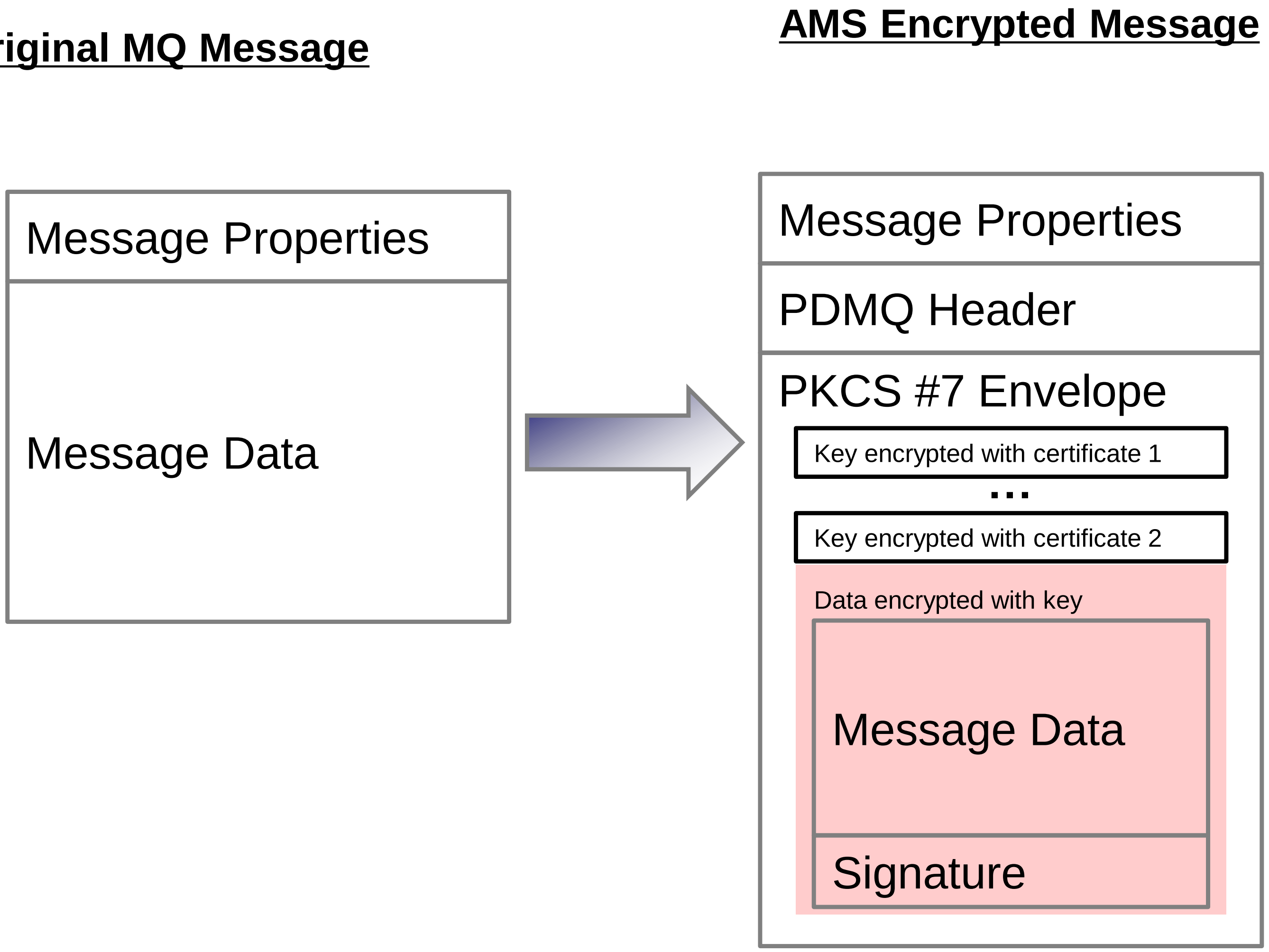
WebSphere MQ AMS – Signed Message Format (Integrity Policy)



WebSphere MQ AMS – Encrypted Message Format (Privacy Policy)



WebSphere MQ AMS – Encrypted Message Format (Privacy Policy)

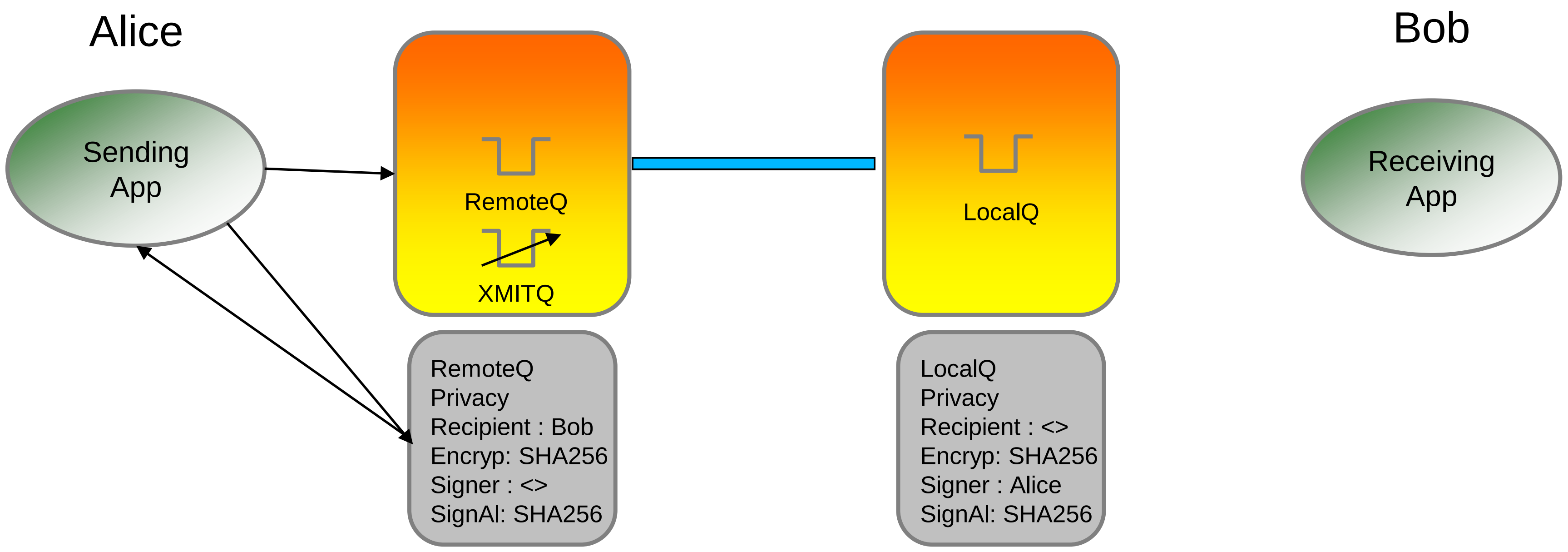


Behaviour

When will my message be protected?

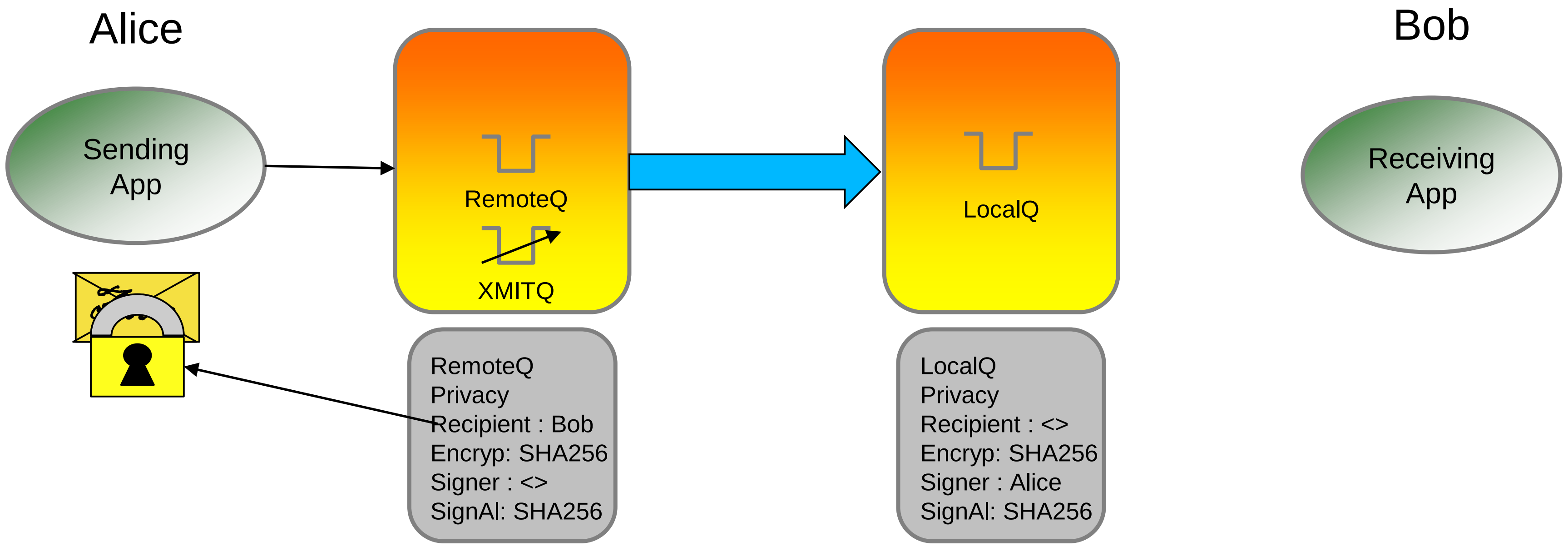
- Messages are protected when they are created
  - Level of protection depends on Policy: None, Integrity, Privacy
  - Policies apply to all Queue Types: Remote, Alias, Local
- During MQOPEN call, policies are queries
  - Look for policies named the same as the Object being opened.
- Once protected, the message retains the policy for it’s lifetime.
- At MQPUT:
  - If there is a policy (regardless of type) we sign the message data
  - If it is a privacy policy we encrypt for the specified recipients
- At MQGET
  - If there is a privacy policy we will decrypt the using our certificate or error
  - If there is a policy we check the message was signed by a signer listed in the policy

When will my message be protected?



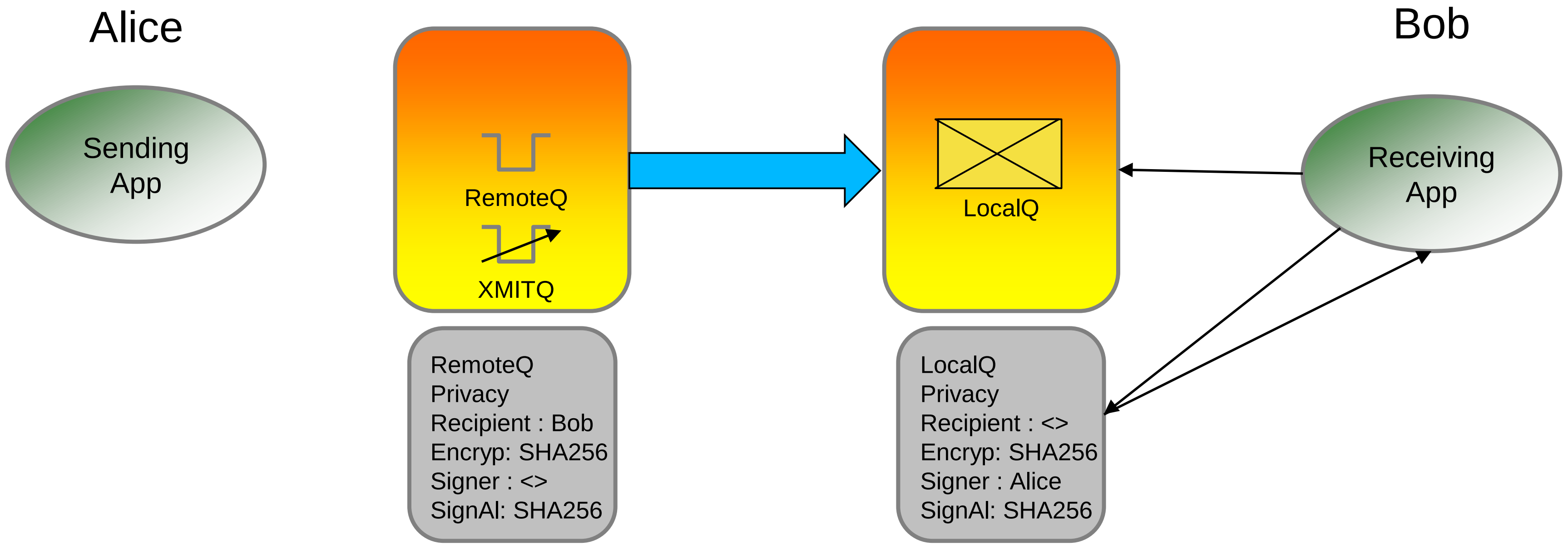
1. Alice’s Application Calls MQOPEN on RemoteQ
2. MQOPEN Queries for Policy called RemoteQ and passes info back

When will my message be protected?



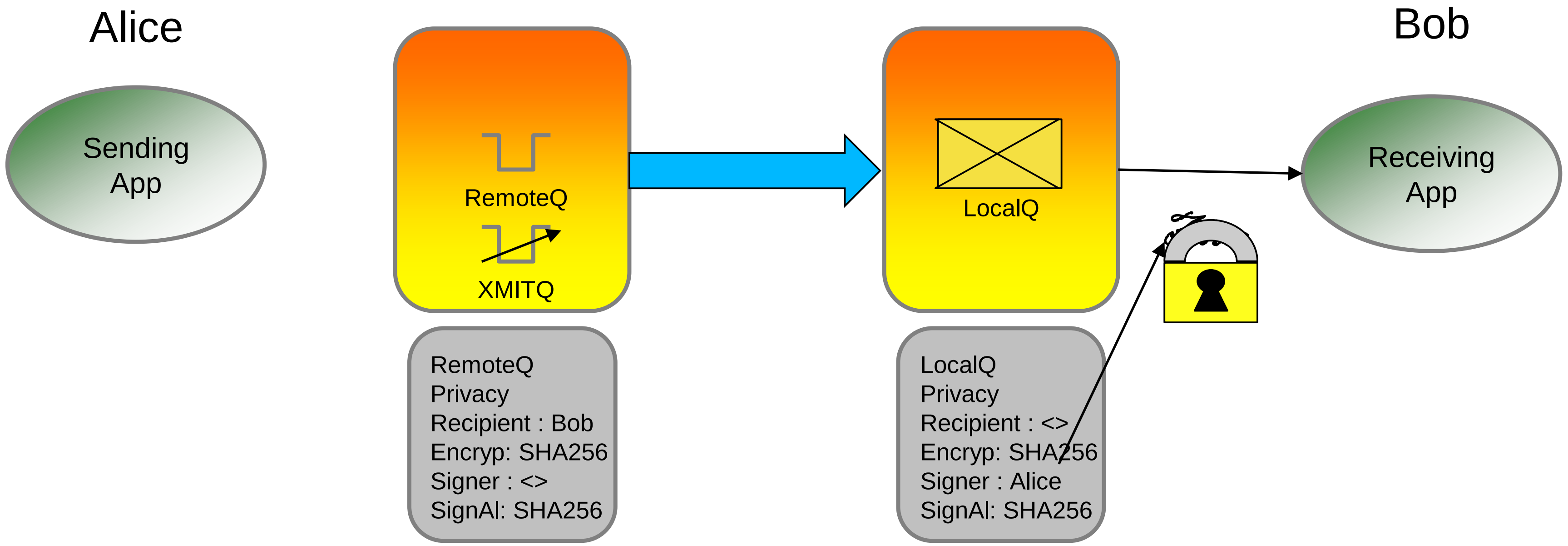
3. Alice issues a MQPUT to RemoteQ
   a) Because there is a policy AMS signs the message data
   b) If the policy is a Privacy policy it also encrypts it for the recipients
4. The message is put to RemoteQ and flows over to the LocalQ

When will my message be protected?



5. Bob Issues an MQOPEN call to LocalQ
6. MQOPEN queries for any policies called LocalQ and returns the info

When will my message be protected?



RemoteQ
Privacy
Recipient : Bob
Encryp: SHA256
Signer : <>
SignAl: SHA256

LocalQ
Privacy
Recipient : <>
Encryp: SHA256
Signer : Alice
SignAl: SHA256

7. Bob Issues MQGET
   a) Checks the Encryption Algorithm used is same or stronger
   b) Checks Bob can decrypt the message
   c) Checks the Signing Algorithm used is same or stronger
   d) Checks the message was from an authorised signer listed in the policy

8. Bob reads his message

Error conditions

- Several scenarios where something could go wrong:
  - Putting to a protected Queue without Client AMS setup
  - GET/BROWSE a message you are not a recipient for
  - GET/BROWSE a message signed by someone not authorized
  - GET/BROWSE a message that has NOT been protected (got onto Q via AliasQ/RemoteQ etc)
  - Signing or encryption Algorithm in message is weaker than policy dictates during GET/BROWSE
  - Do not have correct certificates for the all listed Recipients
  - Misspelt Distinguished names for Authorized Signers or Recipients
  - Recipient does not have the signers certificate
  - Unlike SSL/TLS - full trust chain is not supplied. E.g. Signer cert, Intermediate CA cert, CA cert, etc
  - Error with Key Store configuration – Key Store Permissions, stanzas, etc

- What happens depends on operation being performed:
  - MQPUT – 2063 Error returned and message not accepted
  - MQGET – 2063 Error returned and message is moved to SYSTEM.PROTECTION.ERROR Queue
  - MQBROWSE – 2063 Error returned
  - Key Store related problems 2035

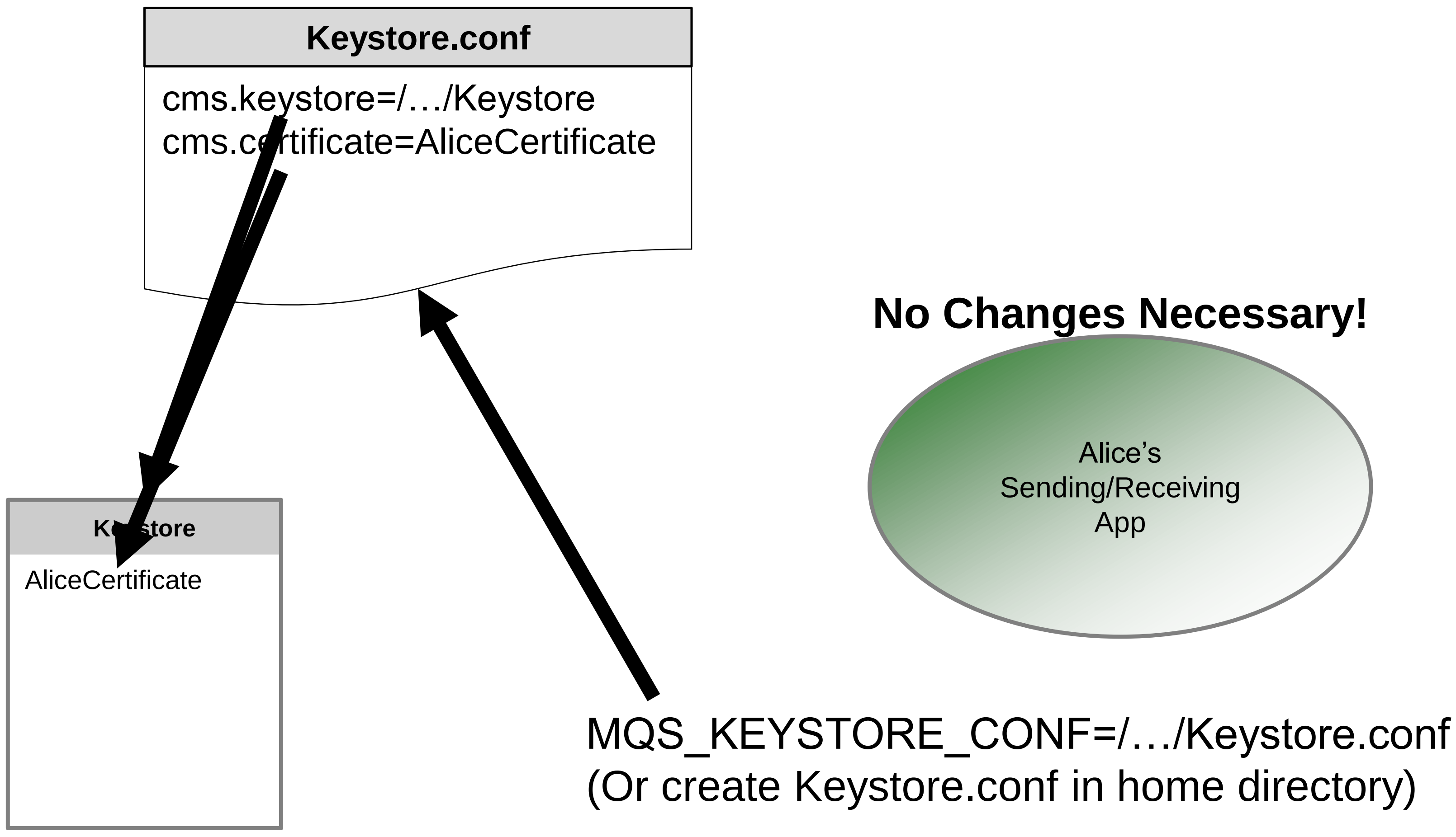
Performance

Performance considerations

- As with all cryptographic operations - there is a decrease in performance
- No official figures to performance impact. Varies wildly by application
  - 1 message per second -> 1 message per second
  - 500 messages per second -> 400~ message per second
  - 10,000 messages per second -> 500~ message per second
  - (Actual figures are likely to vary wildly depending on numerous reasons)
- Privacy Policies affect performance more than Integrity Policies

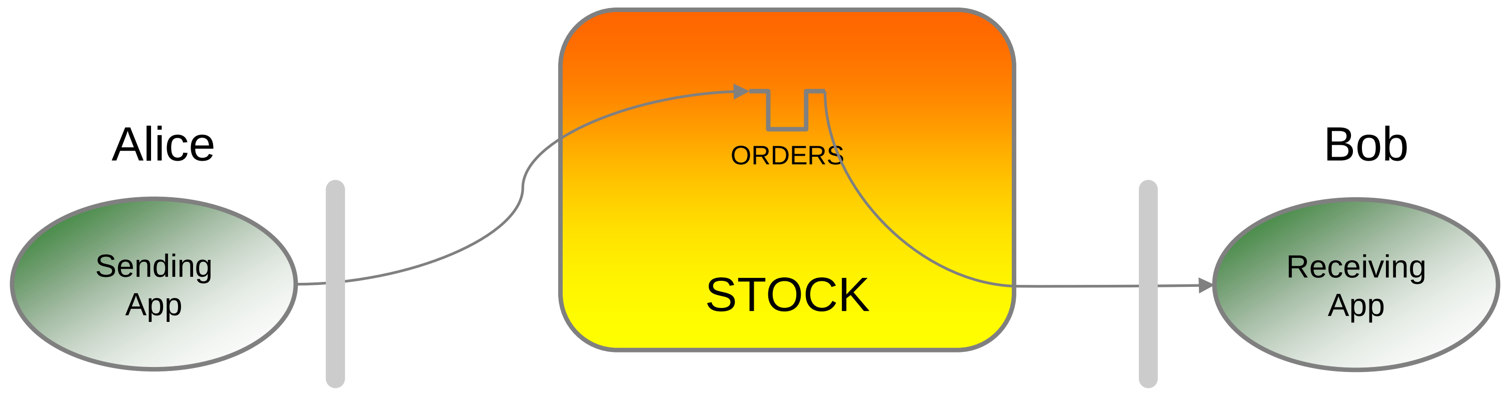
Implementation

Implementing AMS – Application Changes

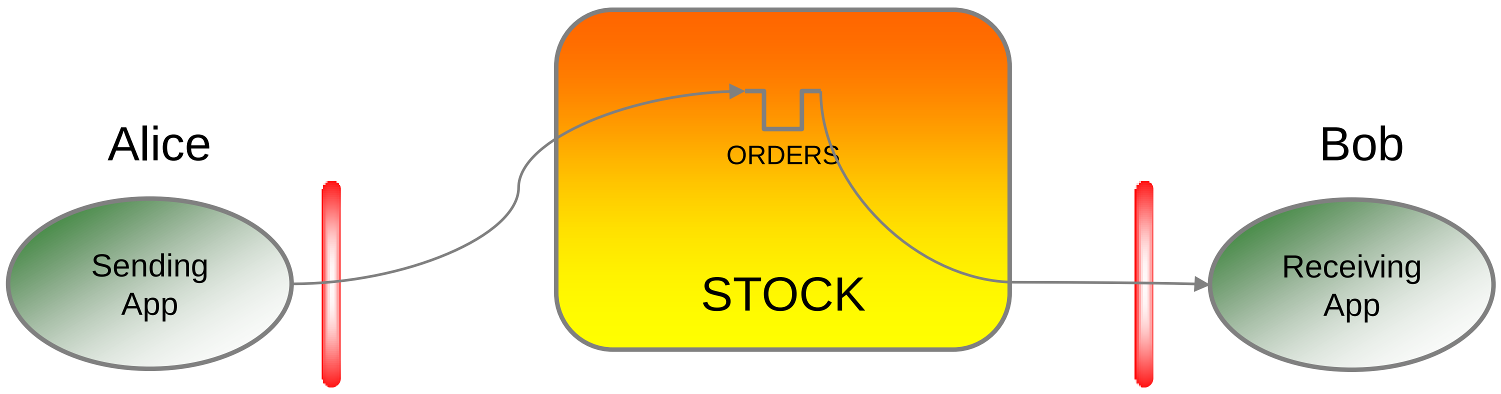


MQS_KEYSTORE_CONF=/…/Keystore.conf
(Or create Keystore.conf in home directory)

How to secure an existing MQ application – No protection

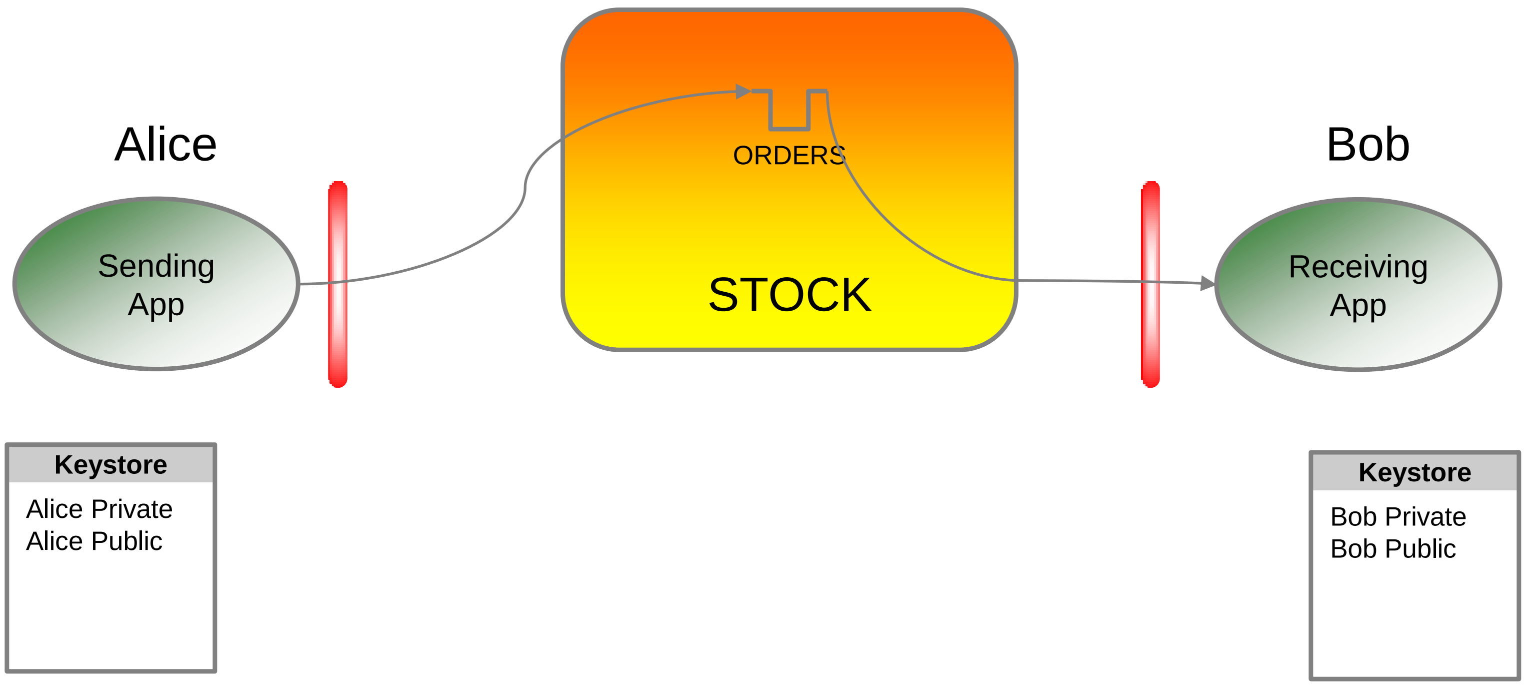


How to secure an existing MQ application - SPLCAP(ENABLED)



1. Install WebSphere MQ AMS Component on server

How to secure an existing MQ application – Assign Certificates



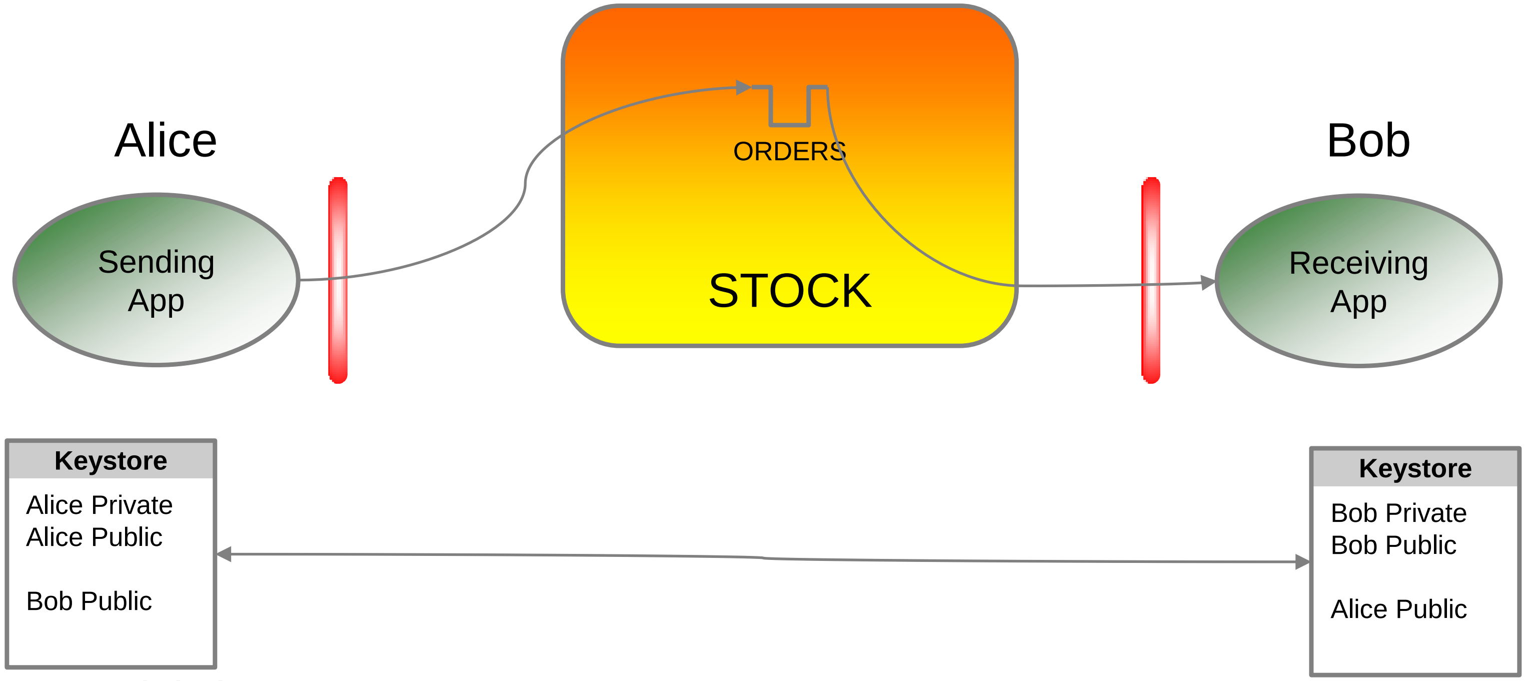
1. Install WebSphere MQ AMS Component on server
2. Create certificates (public / private key pairs)

How to secure an existing MQ application – Assign Certificates

- Key Store and certificate creation using runmqckm, runmqakm or strmqikm
- **runmqakm –keydb –create –db Alice.kdb –pw passw0rd –stash**
- **runmqakm –keydb –create –db Bob.kdb –pw passw0rd –stash**

- **Runmqakm –cert –create –db Alice.kdb –stashed –dn CN=ALICE,O=IBM,C=UK –label AliceCert**
- **Runmqakm –cert –create –db Bob.kdb –stashed –dn CN=BOB,O=IBM,C=UK –label BobCert**

How to secure an existing MQ application – Exchange Public Key

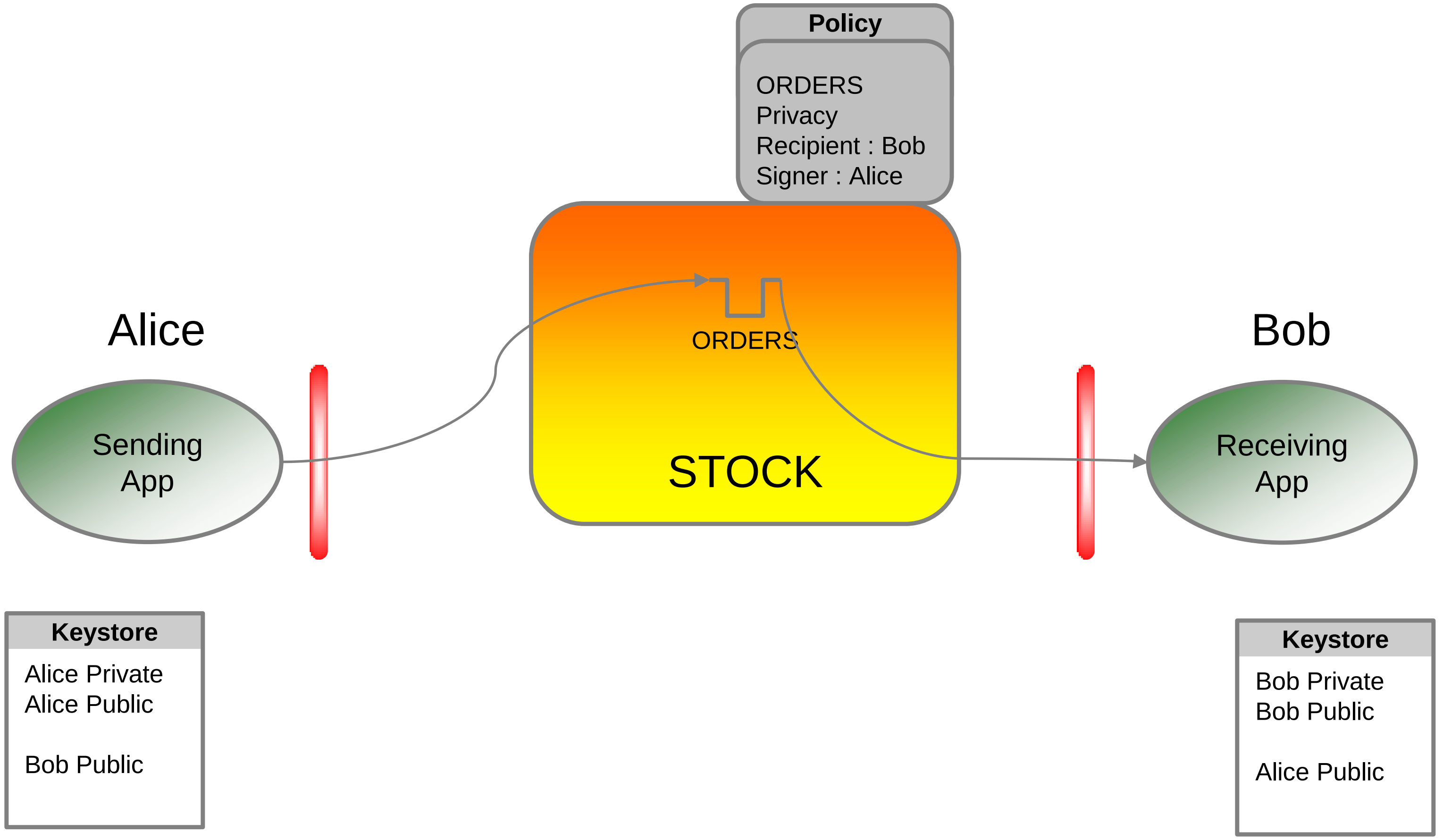


3.Exchange public keys

How to secure an existing MQ application – Exchange Public Key

- Extract and Exchange certificates using runmqckm, runmqakm or strmqikm
- **runmqakm –cert –extract –db Bob.kdb –stashed –label BobCert –target bob.cer**
- **runmqakm –cert –extract –db Alice.kdb –stashed –label AliceCert –target alice.cer**
- **Runmqakm –cert –add –db Alice.kdb –stashed –file bob.cer –label BobCert**
- **Runmqakm –cert –add –db Bob.kdb –stashed –file alice.cer –label AliceCert**

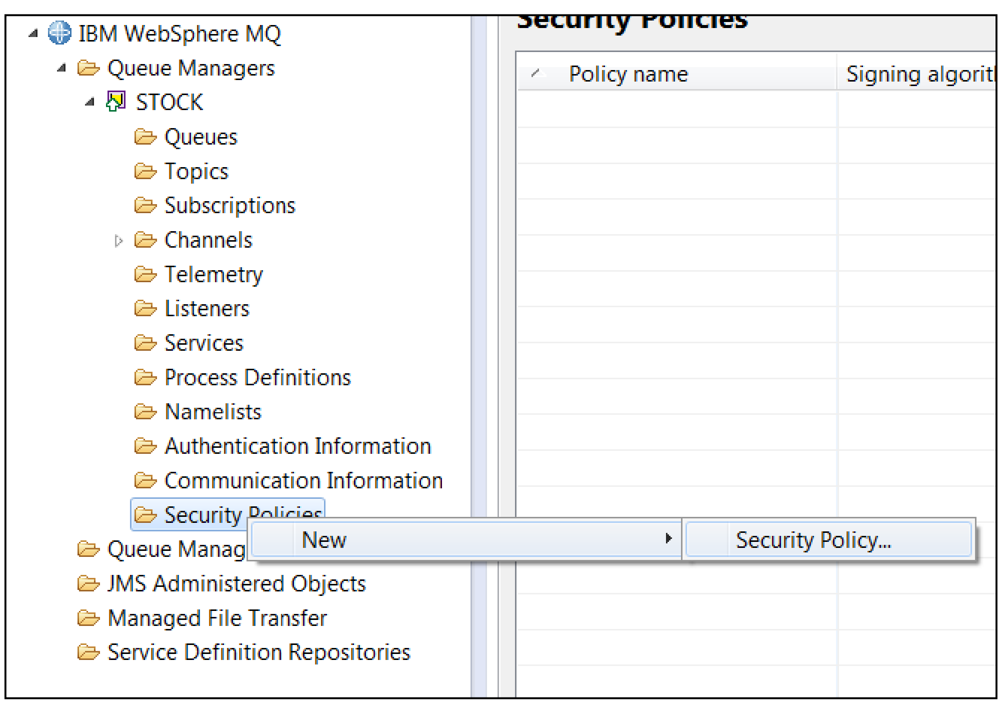
How to secure an existing MQ application – Set security policy

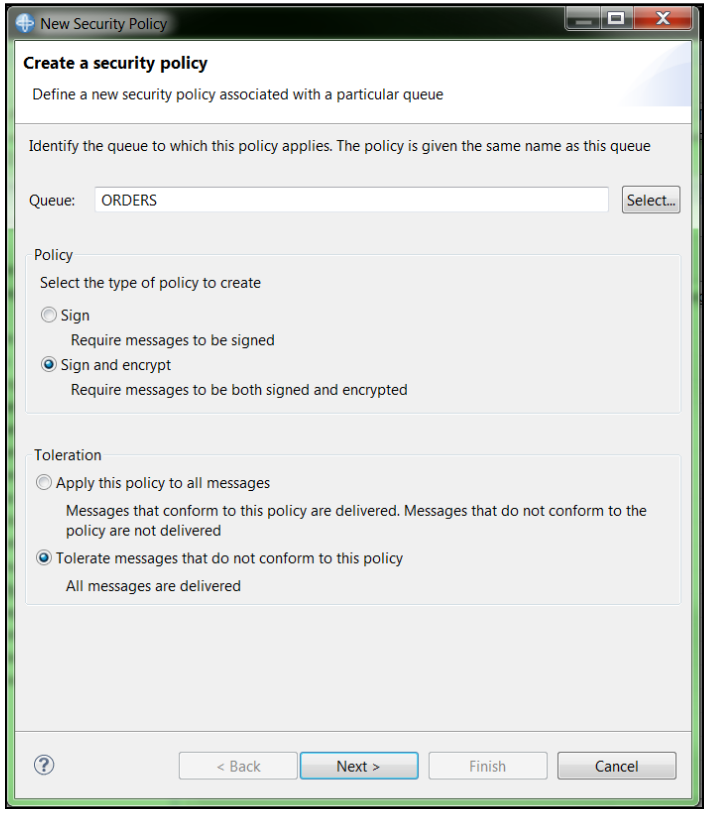


3. Exchange public keys
4. Define security policy for the queue

How to secure an existing MQ application – Set security policy

- Set Security Policy using setmqspl or MQ Explorer
- **Setmqspl –m STOCK –p ORDERS –s SHA256 –a “CN=ALICE,O=IBM,C=UK” –e AES256 –r “CN=BOB,O=IBM,C=UK”**




New Security Policy

Create a security policy

Define a new security policy associated with a particular queue

Identify the queue to which this policy applies. The policy is given the same name as this queue

Queue: ORDERS

Policy

Select the type of policy to create

○ Sign
Require messages to be signed

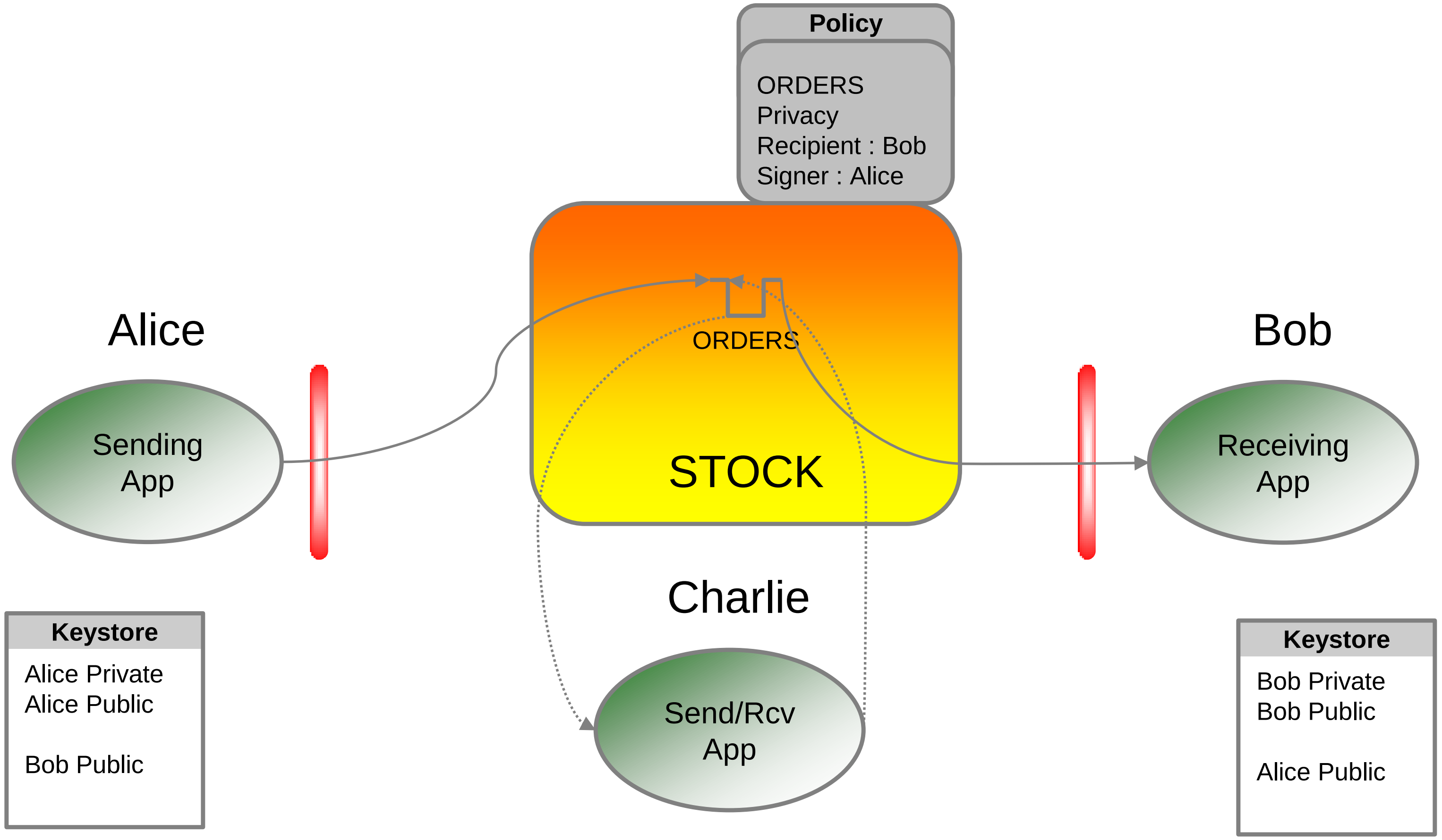
● Sign and encrypt
Require messages to be both signed and encrypted

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

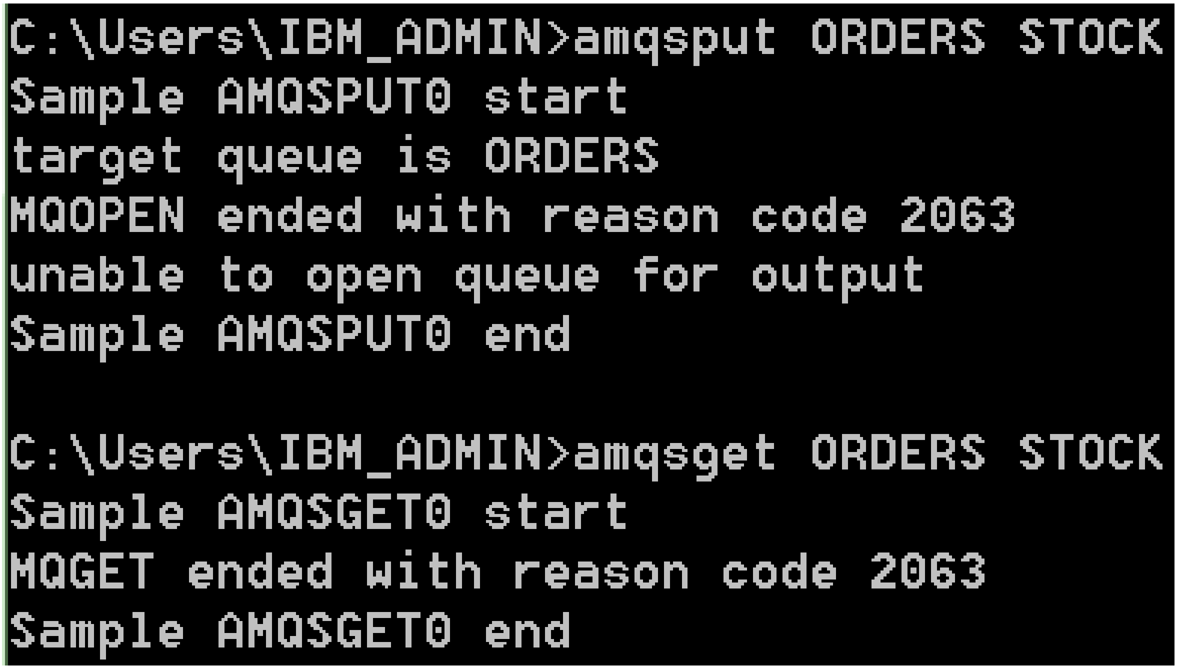
How to secure an existing MQ application – Privacy & Integrity



5.Messages can only be viewed by Bob, Bob will only accept messages from Alice

How to secure an existing MQ application – Privacy & Integrity

- When Charlie attempts to put or get a message – 2063 MQRC_SECURITY_ERROR



```
C:\Users\IBM_ADMIN>amqsput ORDERS STOCK
Sample AMQSPUT0 start
target queue is ORDERS
MQOPEN ended with reason code 2063
unable to open queue for output
Sample AMQSPUT0 end

C:\Users\IBM_ADMIN>amqsget ORDERS STOCK
Sample AMQSGET0 start
MQGET ended with reason code 2063
Sample AMQSGET0 end
```

Thank you very much.

Robert Parker

IBM
IBM MQ Security Development
parrobe@uk.ibm.com