IBM MQ for z/OS version 9.0.1 Performance Report

November 2016

IBM MQ Performance
IBM UK Laboratories
Hursley Park
Winchester
Hampshire
SO21 2JN

Introduction

This report focuses on performance improvements since IBM® MQ for z/OS version 9.0.0, which relate to the Advanced Message Security (AMS) component of the MQ product.

Comparisons in AMS performance will be made between versions 8.0.0, 9.0.0 and 9.0.1.

The performance data in supportPac MP1K "IBM MQ for z/OS version 9.0.0 Performance Report" is still applicable for areas of MQ performance that is not superceded by this report.

SupportPac MP16 "Capacity Planning and Tuning Guide" will continue to be the repository for ongoing advise and guidance learned as systems increase in power and experience is gained.

Objective

Examine the performance improvements of the AMS policy types, primarily concentrating on AMS Confidentiality which was introduced in IBM MQ for z/OS version 9.0 and that has been subsequently optimised in IBM MQ for z/OS version 9.0.1.

Performance Highlights

The performance of AMS in V9.0.1, as demonstrated later in this document, has improved significantly over previous versions. Here are some of the highlights:

Comparing AMS on V901 with AMS on V800:

- Transactions protected by AMS Integrity policies can see a cost of **30%** of similar transactions run against V800.
- Transactions protected by AMS Privacy policies can achieve a cost of **30%** those of similar transactions in V800.
- Transactions protected by AMS Confidential policies can achieve a transaction rate of **24 times** that of a comparable workload using V800 AMS Privacy protection.

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**.

Comparing AMS on V901 with channels protected by SSL cipher specs:

- Small messages workloads (10KB and less) protected using AMS Confidential policies can demonstrate a lower transaction cost than channels protected using SSL cipher specs that have SSLRKEYC settings such that significantly more data has flows between queue managers between key negotiations.
- Medium sized messages workloads (32-64KB) protected using AMS Confidential policies with a key reuse of 32 can achieve comparable transaction cost with long running channels protected using SSL cipher specs that negotiate the secret key only at channel start.
- Large messages workloads (1MB) protected using AMS Confidential policies can show a **17%** reduction in transaction cost over channels running SSL cipher specs where the secret key is re-negotiated at similar intervals.
- AMS policies allow the key to be renegotiated at a queue level, rather than the queue manager level attribute SSLRKEYC (SSL key reset count, which is the total number of bytes to be sent and received within an SSL conversation before the secret key is renegotiated).

  This means that the key can be renegotiated on an appropriate value for the particular workload. This may be useful if some channels send high volumes of data and some send low-volumes which may leave a secret key unchanged for long periods.
- AMS policy protection means that the message is protected from the time it is put to the queue until it is gotten by the authorised recipient and remains protected regardless of how many queue managers the message transitions through.

  By contrast a message protected by SSL channels is only encrypted for the flow across the network. Furthermore if the message transitions across multiple queue managers each with SSL enabled channels, the cost of encryption and decryption can be incurred for each transition.

Background

IBM MQ Advanced Message Security (IBM MQ AMS) provides a high level of protection for sensitive data flowing through the MQ network with different levels of protection by using a public key cryptography model.

IBM MQ version 9.0.0 supplemented the existing two qualities of protection *Integrity* and *Privacy* with a third quality of protection, namely *Confidential*.

Integrity protection is provided by digital signing, which provides assurance on who created the message, and that the message has not been altered or tampered with.

Privacy protection is provided by a combination of digital signing and encryption. Encryption ensures that message data is viewable by only the intended recipient, or recipients.

Confidentiality protection is provided by encryption only.

IBM MQ AMS uses a combination of symmetric and asymmetric cryptographic routines to provide digital signing and encryption. As symmetric key operations are very fast in comparison to asymmetric key operations, which are CPU intensive and some of the cost may be offloaded to cryptographic hardware such as Crypto Express5, this in turn can have a significant impact on the cost of protecting large numbers of messages with IBM MQ AMS.

- **Asymmetric cryptographic routines** as used by Integrity and Privacy

  For example, when putting a signed message the message hash is signed using an asymmetric key operation. When getting a signed message, a further asymmetric key operation is used to verify the signed hash. Therefore, a minimum of two asymmetric key operations are required per message to sign and verify the message data. Some of this asymmetric cryptographic work can be offloaded to cryptographic hardware.

- **Asymmetric and symmetric cryptographic routines** as used by Privacy and Confidentiality

  When putting an encrypted message, a symmetric key is generated and then encrypted using an asymmetric key operation for each intended recipient of the message. The message data is then encrypted with the symmetric key. When getting the encrypted message the intended recipient needs to use an asymmetric key operation to discover the symmetric key in use for the message. The symmetric key work cannot be offloaded to cryptographic hardware but will be performed in part by CPACF processors.

All three qualities of protection, therefore, contain varying elements of CPU intensive asymmetric key operations, which will significantly impact the maximum achievable messaging rate for applications putting and getting messages.

Confidentiality policies do, however, allow for symmetric key reuse over a sequence of messages.

Key reuse can significantly reduce the costs involved in encrypting a number of messages intended for the same recipient or recipients.

For example, when putting 10 encrypted messages to the same set of recipients, a symmetric key is generated. This key is encrypted for the first message using an asymmetric key operation for each of the intended recipients of the message.

Based upon policy controlled limits, the encrypted symmetric key can then be reused by subsequent messages that are intended for the same recipient(s). An application that is getting encrypted messages can apply the same optimization, in that the application can detect when a symmetric key has not changed and avoid the expense of retrieving the symmetric key.

In this example 90% of the asymmetric key operations can be avoided by both the putting and getting applications reusing the same key.

IBM MQ for z/OS version 9.0.1 has further optimised the performance of AMS qualities of protection, for all 3 levels, with the most optimal offering from a performance perspective being AMS Confidential.

This document aims to demonstrate the improvements in performance, comparing release-on-release improvements as well as comparing the different message protection options offered by MQ.

Scenarios

There are a number of scenarios that were used to gather the data in this report:

- **Request / Reply - Local workload:**

  A request/reply workload is run using pairs of requester and server batch tasks connected to a single MQ queue manager. Each pair of applications uses their own request and reply queues.

  The requester task puts a message to the request queue and waits for a specific response on the reply queue. Once the reply message is gotten, the requester will put another message to the request queue.

  The corresponding server task issues MQGET-with-wait calls on the request queue, gets the message to the known (and pre-opened) reply queue and the application goes back into its MQGET-with-wait call. The messages are got and put in syncpoint with 1 MQGET and 1 MQPUT per commit.

  2KB non-persistent messages are used.

- **Request / Reply - Mover workload:**

  A request/reply workload is run to move messages between 2 queue managers on separate LPARs of the same performance sysplex. There are sender-receiver channels defined in each direction.

  Comparisons of transaction cost and rate are provided for 3 configurations - no protection, channels protected by TLS encryption and messages protected by AMS policies.

  The TLS configuration uses cipher spec "ECDHE_RSA_AES_256_CBC_SHA384" and the test varies the SSL key negotiation frequency (SSLRKEYC) using values of 0, 1 and 10MB.

  All AMS policy types are used i.e. Integrity, Privacy and Confidential.

  Non-persistent messages of size 32KB are used.

- **Streaming messages between queue managers:**

  An example of streaming messages between queue managers is a scenario such as an InfoSphere Replication Server workload.

  This replication workload simulates moving data from one system to another using MQ channels. The system that sends the data uses a "capture" task to get the data and put to an MQ queue as quickly as possible. At the remote end, there is an "apply" task that gets the messages from the queues and processes them. As the data flows in a single direction, there is the potential for a build up of messages on the transmit queue as the capture task may put messages more quickly than the channel initiator can get and send the messages, for example in the event of a network delay or the apply task being slow.

  Measurements will be run using TLS cipher spec "ECDHE_RSA_AES_256_CBC_SHA384" with a range of key negotiation frequency and compared with queues protected using AMS Confidentiality policies with a range of key reuse values.

  Persistent messages of 10KB and 1MB are used.

All policies are defined such that there is only a single recipient for each message.

The signing algorithm used in the Integrity and Privacy tests is **SHA256**. The encryption algorithm used for Privacy and Confidential tests is **AES256**.

The applications in use in all measurements have minimal business logic, which means that the application cost is considerably less than would be expected in a customer transaction. This can

make the impact of AMS policies appear more significant than would be observed in a customer environment.

Measurements

Request / Reply - Local workload

This section demonstrates the improvement in performance of the 3 AMS policy types in V9.0.1 by comparing the transaction costs and rates of a simple request / reply workload using non-persistent messages.

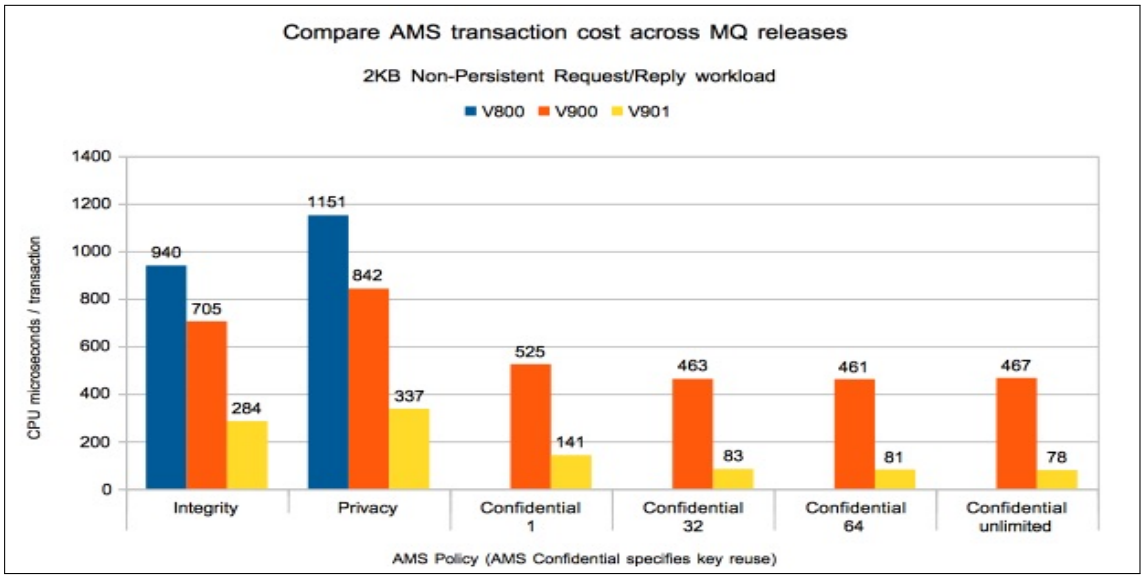
In this most simplistic scenario using small messages, when comparing the total transaction cost for V9.0.1 against V9.0.0, there are significant reductions in cost e.g.

- **Integrity**: V901 cost was 45% of the equivalent V900 measurement.
- **Privacy**: V901 cost was 40% of the equivalent V900 measurement.
- **Confidential**: V901 cost was 15-25% of the equivalent V900 measurements depending on the key reuse value.

Both Integrity and Privacy policy types showed similar percentage reductions in transaction cost for larger messages.

The improvements in the AMS Confidential performance for V901 are less marked as the message size increases, but as a guide, with 64KB messages this workload benefits from a 60% decrease in transaction cost over V900 with 4MB messages benefitting from a 30% reduction in total transaction cost.

Chart: Transaction Cost - Request/Reply - Local workload



Notes on chart:

- The costs shown are based on the total CPU costs for the queue manager, AMS region and batch application regions, using data from the RMF Workload report, and divided by the number of transactions in the measurement.
- The transaction cost for the Integrity measurements reduced by 25% between V800 and V900.
- The transaction cost for the Privacy measurements reduced by 27% between V800 and V900
- The reduction in transaction cost for the Confidential measurements depends on the key reuse value selected.
  - Low key reuse values: V901 cost is 73% less than V900 equivalent
  - Higher key reuse values (32+): V901 cost is 80% less than V900 equivalent

The 2 charts on the following page show a breakdown of the transaction cost by address space, e.g. how much of the transaction cost is charged to the queue manager, AMS and application address spaces and how this has changed between MQ V9.0.0 and V9.0.1.

Chart: V900 - Breaking down the transaction cost by address space

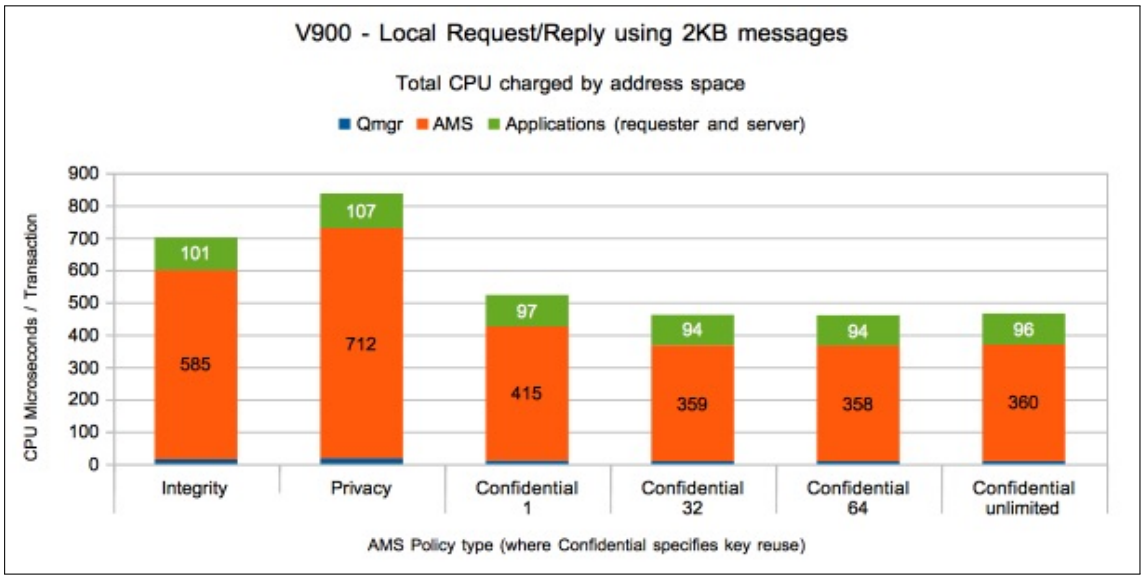
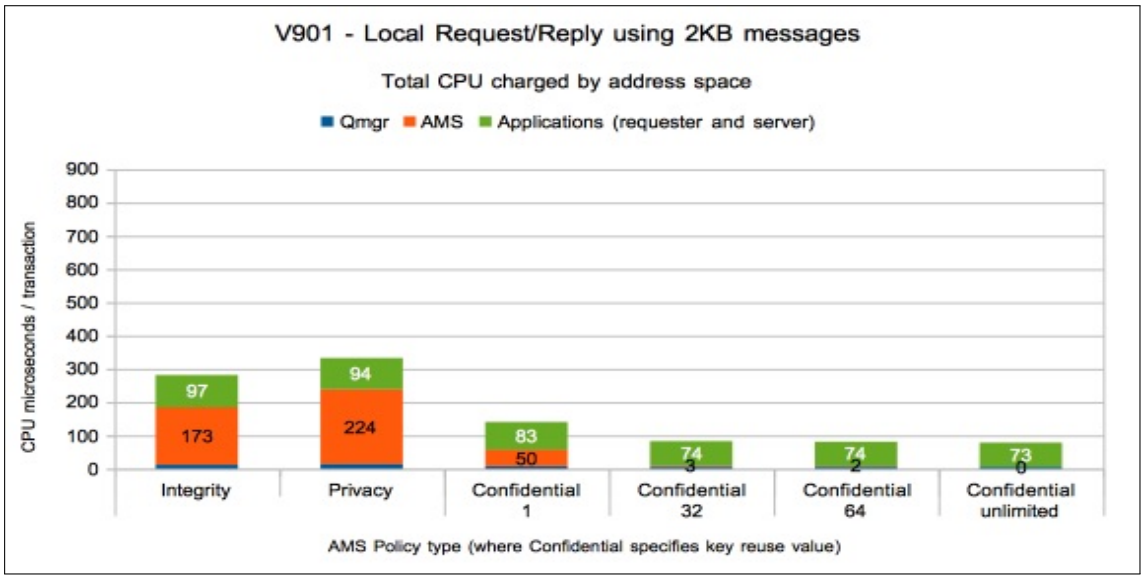


Chart: V901 - Breaking down the transaction cost by address space



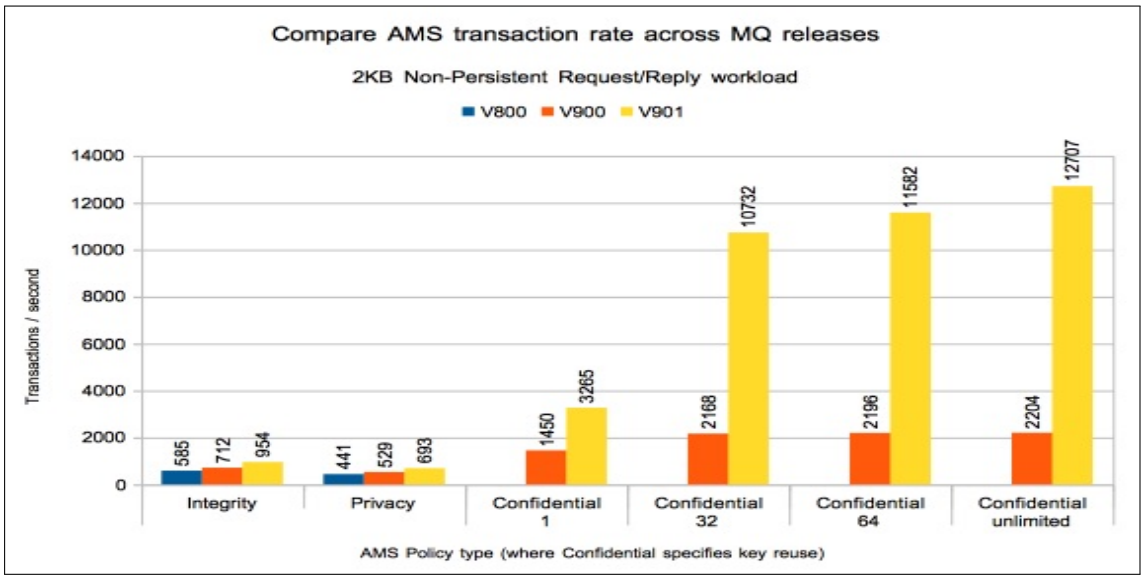
Note that both charts have the same scale on the y-axis (transaction cost).

Both AMS Integrity and Privacy see a much reduced cost in the AMS address space.

The AMS Confidential costs remain a significant proportion of the transaction cost in V900 even as the key reuse become unlimited, whereas in V901 the AMS address space become a much smaller factor in the transaction cost tending towards zero cost, particularly as the secret key is reused for more messages. Note that even with key reuse of "unlimited", there is still an impact to the application address space which is charged for the encryption and decryption of the message.

The final chart for the local request/reply workload shows the transaction rates achieved for the three AMS releases and the different policy types.

Chart: Transaction Rate - Request/Reply - Local workload



The AMS Confidential measurements show an increase in throughput of up to 5 times when moving from V900 to V901.

Smaller but still significant improvements can be seen with both AMS Integrity and Privacy qualities of protection.

Request / Reply - Mover workload

This section discusses the performance of the 3 AMS policy types, namely Integrity, Privacy and Confidential in a queue manager to queue manager environment. The Confidential measurements include key reuse values of 1, 32, 64 and unlimited.

A further comparison is made between AMS Confidential and channels that are protected using SSL ciphers where the SSL secret key negotiation (SSLRKEYC) ranges from 0 (only at channel start), to 1MB and 10MB.

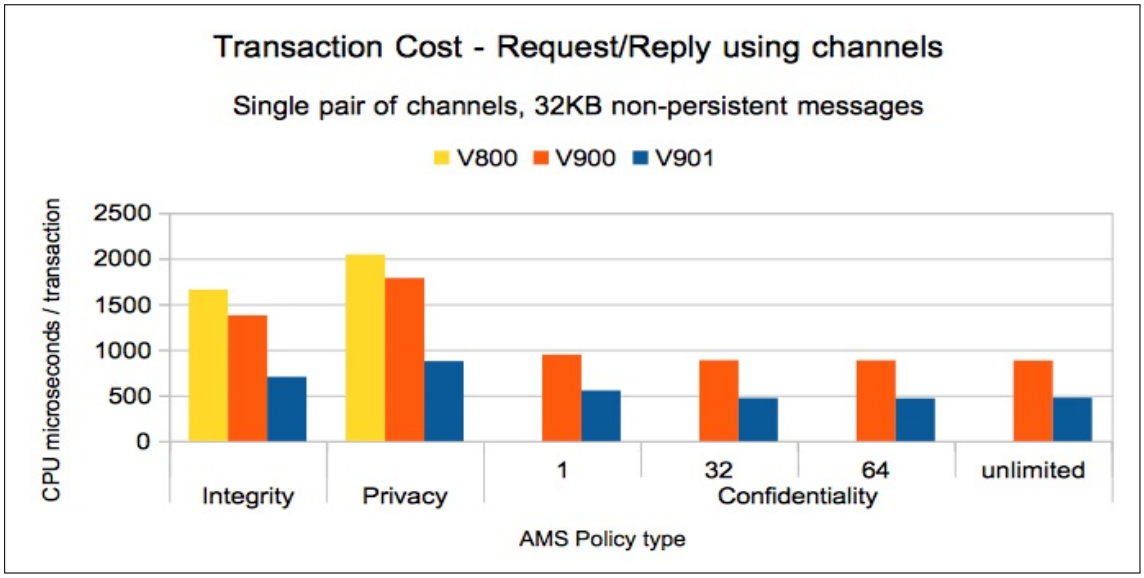
Comparing AMS Policy types

As was demonstrated in the request/reply local workload section, the improvements are not limited to AMS Confidential policies.

The following 2 charts offer a comparison in the performance of a request/reply workload using 32KB non-persistent messages between 2 queue managers on separate LPARs for MQ V8.0.0, V9.0.0 and V9.0.1.

For the purpose of clarity the measurements show the performance of a single outbound and a single inbound channel only.

Chart: Transaction Cost - Request/Reply - Mover workload

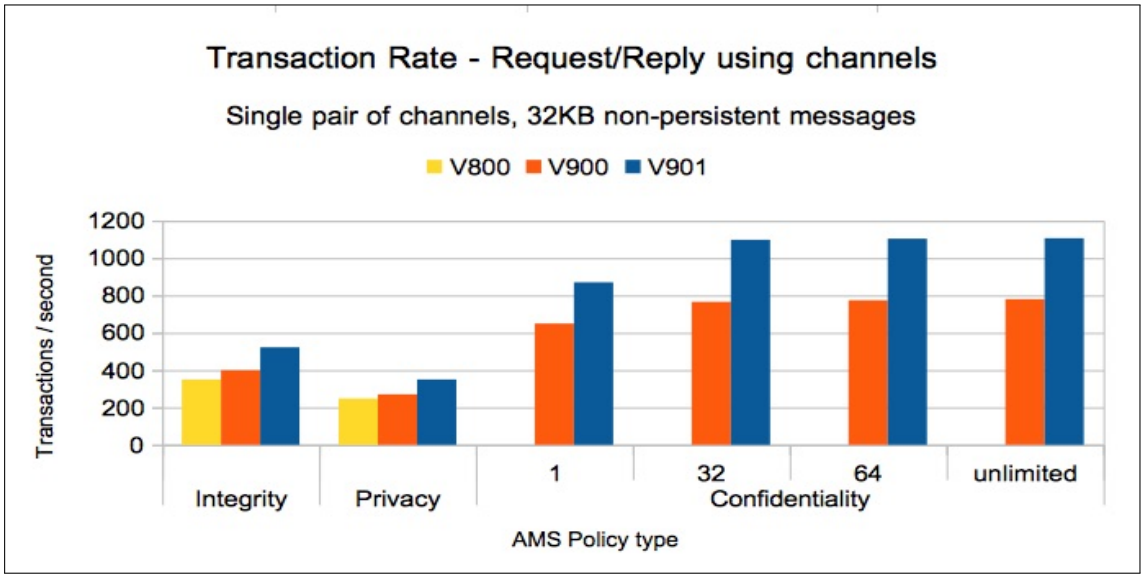


Notes on chart:

- The cost of the transaction include 2 MQPUTs with associated encryption cost and 2 MQGETs with associated decryption cost by the applications plus 2 MQGET and 2 MQPUTs by the channel initiators.
- In V901 for this workload, AMS Integrity costs have reduced by 57% compared to the equivalent V800 measurement and 49% compared to the equivalent V900 measurement.
- In V901 for this workload, AMS Privacy costs have reduced by 57% compared to the equivalent V800 measurement and 51% compared to the equivalent V900 measurement.
- In V901 for this workload, AMS Confidentiality costs have reduced by between 40 to 48% compared to the equivalent V900 measurement.

- As a guide, the equivalent measurement where no AMS policy is defined for the queues, the transaction cost is 268 microseconds.

Chart: Transaction Rate - Request/Reply - Mover workload



Notes on chart:

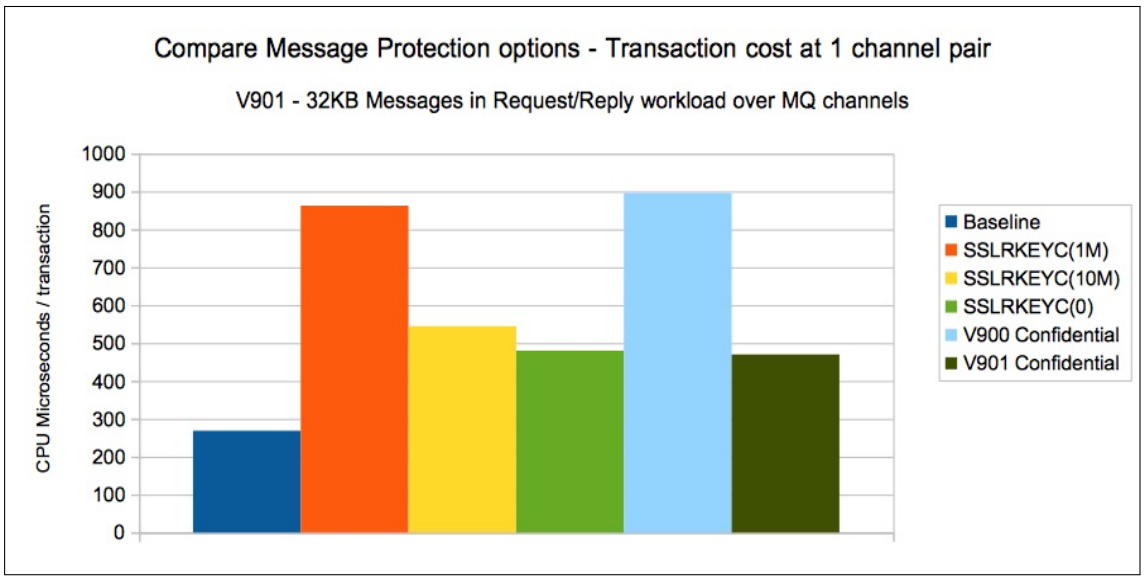
- The chart shows the transaction rate when using a single requesting task and a single server task.
- In V901 for this workload, AMS Integrity transaction rates have increased 49% compared to the equivalent V800 measurement and 30% compared to the equivalent V900 measurement.
- In V901 for this workload, AMS Privacy transaction rates have increased by 41% compared to the equivalent V800 measurement and 29% compared to the equivalent V900 measurement.
- In V901 for this workload, AMS Confidentiality transaction rates have increased by between 33 to 43% compared to the equivalent V900 measurement.
- As a guide, the equivalent measurement where no AMS policy is defined for the queues, a transaction rate of 1680 per seconds is achieved which is 52% higher than the AMS Confidential measurement with a key reuse of 64.

Comparing AMS with SSL ciphers

This section compares the performance of AMS Confidential with a key reuse of 32 against the performance of a workload with SSL cipher spec "ECDHE_RSA_AES_256_CBC_SHA384" with a range of secret key negotiation values.

The following chart offers a comparison in the transaction cost of a request/reply workload using 32KB non-persistent messages between 2 queue managers on separate LPARs and compares a baseline (no message protection) with SSL encryption against queues protected using AMS Confidential with a key reuse of 32.

Chart: Transaction Cost - Request/Reply - Mover workload



Notes on chart:

- The chart shows the total transaction cost when using a single requesting task and a single server task.
- The V900 AMS Confidential measurement shows similar transaction costs to the SSL measurement where the secret key is negotiated every 1MB (or approximately every 32 messages).
- In V901 AMS Confidential measurement shows transaction costs comparable with SSL measurements where the secret key is negotiated only at channel start (SSLRKEYC(0)).
- In V901 for this workload, AMS Confidentiality transaction rates have increased by between 33 to 43% compared to the equivalent V900 measurement.

Streaming messages between queue managers

One use of the AMS Confidential quality of protection is where data is moved between data centres, such as an IBM InfoSphere Data Replication queue replication scenario.

The channels defined between the queue managers in the two data centres may be protected using SSL ciphers but it can be less expensive to encrypt the messages using AMS Confidential-type policies.

There are a number of considerations to take into account:

- SSL secret key negotiation and the generation of the AMS secret key costs are relatively static.

  What this means is that with a key reuse of 32, the impact of AMS key generation is spread across 32 messages, whether the message is 1 byte or 100MB, which means the impact is greater for small messages. By comparison with SSLRKEYC(32MB), the impact on the message cost is very dependent on the size of the message. For example 978 messages of 10KB could flow between negotiations based on SSLRKEYC(32MB), which is 30 times more messages than if using AMS with key reuse 32.

- Whether the data flows through other queue managers, for example a gateway queue manager.

  In the SSL configuration, the secret key would be negotiated between the source queue manager and the gateway queue manager, and again between the gateway queue manager and the target queue manager. In addition, the data would be decrypted whilst at rest on the gateway queue manager.

  Using AMS Confidential, the message would be encrypted once - before it is put to the source queue manager and remain encrypted until it was successfully gotten by the application on the target queue manager.

Streaming small (10KB) messages between queue managers

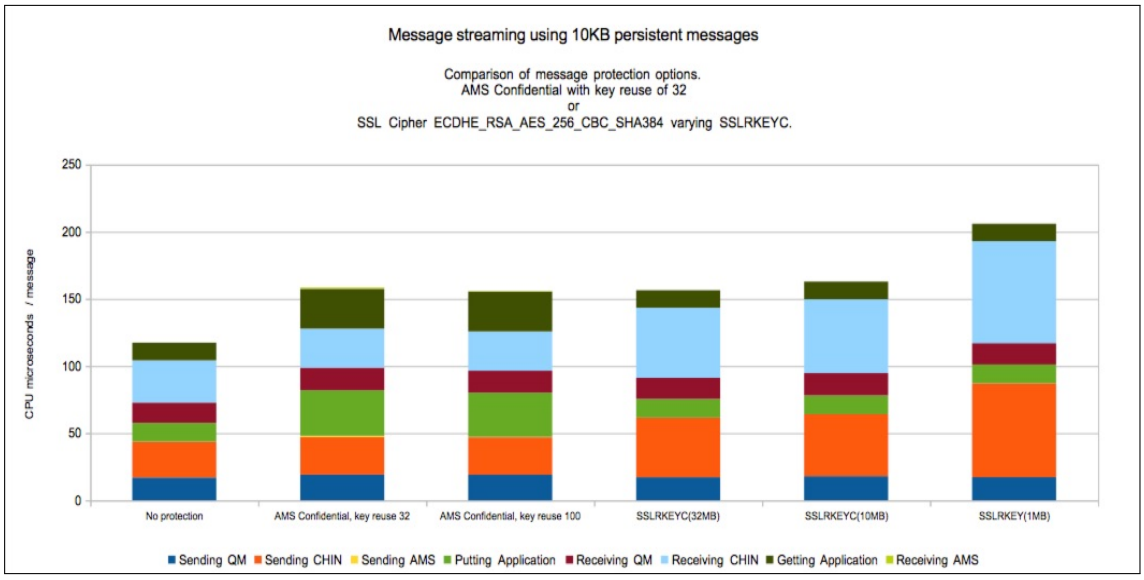
The following chart demonstrates in a streaming type environment, the total transaction cost of protecting the messages using AMS Confidential is comparable to that when protecting the messages using SSL ciphers.

In the lowest cost measurements of both SSL and AMS Confidentiality, there is a total increase in transaction cost of approximately 33% over the cost of the workload when no message or channel protection is in place.

When considering the impact of AMS Confidentiality on only the MQ address spaces, there is an increase of 3 to 5% (including only MSTR, CHIN and AMSM). The majority of the transaction cost increase is in the application regions due to encrypting and decrypting the messages.

By contrast adding SSL ciphers to the baseline measurement, results in an increase of 40% to the transaction cost in the MQ address spaces with SSLRKEYC(32MB), and significantly more with a higher key negotiation frequency.

Chart: Streaming 10KB messages between 2 queue managers



Notes on chart:

- The chart shows the cost per transaction by address space for a 10KB message streaming-type workload.
- With 10KB messages, the AMS costs are equivalent to those of the SSL protected channels.

  There is less than 2% difference in the total costs of the AMS Confidential workload (key reuse 32) and the SSL protected channel using SSLRKEYC(32MB), despite SSL transferring more than additional 3200 messages using the same key.

  Where the SSLRKEYC is set to a similar frequency to the key reuse, e.g. SSLRKEYC(1MB) and key reuse 100 for 10KB messages, the total AMS transaction cost is 25% lower.
- For the AMS measurements, an impact in transaction cost is seen:
  - primarily in the application address space, which incurs the cost of encrypting and decrypting the data
  - in the AMS address space which diminishes with an increasing key reuse value.

    - in the queue manager and channel initiator, which see a small increase due to the message size increasing due to being protected by AMS policies.

- The SSL measurements see an increase in the channel initiator address space, which includes the cost of encrypting and decrypting the data plus the cost of negotiating the secret key.

  With an increased value in the SSLRKEYC attribute, the impact on each message of negotiating the secret key can be reduced.

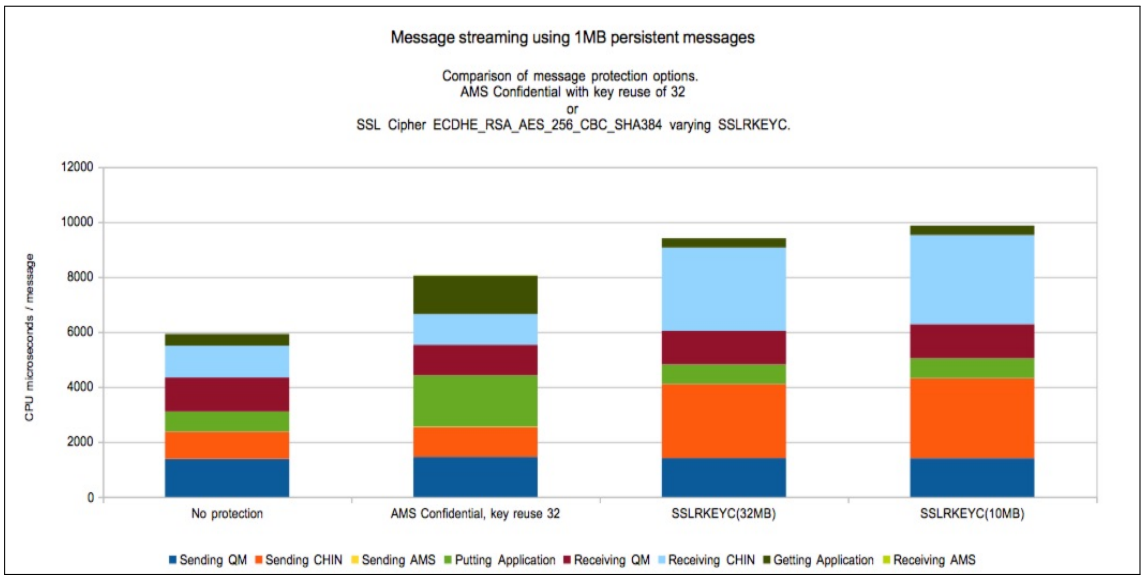
  Note, that some of the secret key negotiation cost has been offloaded to the available Crypto Express card.

Streaming large (1MB) messages between queue managers

The following chart compares the impact of AMS Confidential with key reuse of 32, against channels protected using SSL ciphers that negotiate the secret key either every 10MB or 32MB.

Where similar numbers of messages flow between key negotiations, namely AMS Confidential with key reuse 32 and SSL channels with SSLRKEYC(32MB), the AMS measurement has a transaction cost of approximately 15% less.

Chart: Streaming 1MB messages between 2 queue managers



Notes on chart:

- AMS key reuse 32 shows a 36% cost increase on the "no protection" measurement.
- SSLRKEYC(32MB) shows an additional 17% increase on the AMS Confidential measurement, or a 59% increase in the "no protection" measurement.
- SSLRKEYC(10MB) shows an additional 22% increase on the AMS Confidential measurement, or a 66% increase in the "no protection" measurement.
- AMS Confidential shows similar costs in the MSTR and CHIN address spaces to the "no protection" option, but an increase in the application address spaces.
- SSL shows an increase in the CHIN address spaces compared with both the "no protection" and the AMS Confidential measurements.

System configuration

IBM MQ Performance Sysplex running on z13 (2964-NE1) configured thus:

LPAR 1: 1-16 dedicated CP processors, 128GB of real storage.

LPAR 2: 1-3 dedicated CP processors, 32GB of real storage.

LPAR 3: 1-10 dedicated CP processors, plus 1 zIIP, 32GB of real storage.

Default Configuration:

3 dedicated processors on each LPAR, where each LPAR running z/OS v2r1 FMID HBB7790.

Coupling Facility:

- Internal coupling facility with 4 dedicated processors.
- Coupling Facility running CFCC level 21 service level 02.16.
- Dynamic CF Dispatching off.
- 3 x ICP links between each z/OS LPAR and CF

DASD:

- FICON Express 16S connected DS8870.
- 4 dedicated channel paths (shared across sysplex)
- HYPERPAV enabled.

System settings:

- zHPF disabled by default.
- HIPERDISPATCH enabled by default.
- LPARs 1 and 3 configured with different subnets such that tests moving messages over channels send data over 10GbE performance network.
    - SMC-R enabled by default between LPARs 1 and 3.
- zEDC compression available by default - exploited in V800 onwards for ZLIBFAST compression.
- Crypto Express5 card configured thus:
    - 1 x Accelerator, shared between LPARs 1 and 3.
    - 2 x Coprocessor on LPAR1.
    - 2 x Coprocessor on LPAR3.

IBM MQ trace status:

- TRACE(GLOBAL) disabled.
- TRACE(CHINIT) disabled.
- TRACE(S) CLASS(1,3,4) enabled where supported.
- TRACE(A) CLASS(3,4) enabled where supported.

General information:

- Client machines:

    - 2 x IBM SYSTEM X5660 each with 12 x 2.8GhZ Processor, 32GB memory
- Client tests used a 10GbE performance network.
- Other IBM products used:
    - CICS V6.9 CTS5.2 with latest service applied as of November 2016.
    - IMS V13.
    - IBM MQ for z/OS version 8.0 with latest service applied as of May 2016.
    - IBM MQ for z/OS version 9.0 as at GA.