IBM MQ for z/OS v9.1.3 Advanced Message Security Interception on server to server message channels

Tony Sharkey
Published on 07/08/2019

With the announcement of IBM MQ V9.1.3, the Advanced Message Security Interception on server to server message channels feature became available to z/OS-based queue managers.

What is Advanced Message Security (AMS) Interception on server to server message channels?

The IBM Knowledge Center provides an overview of this new AMS Interception on server to server message channels feature.

For the in-depth performance report, see "AMS Interception on server-server message channels"

The performance report suggests a variety of use cases for AMS Interception on server to server message channels, but for this blog we focus on:

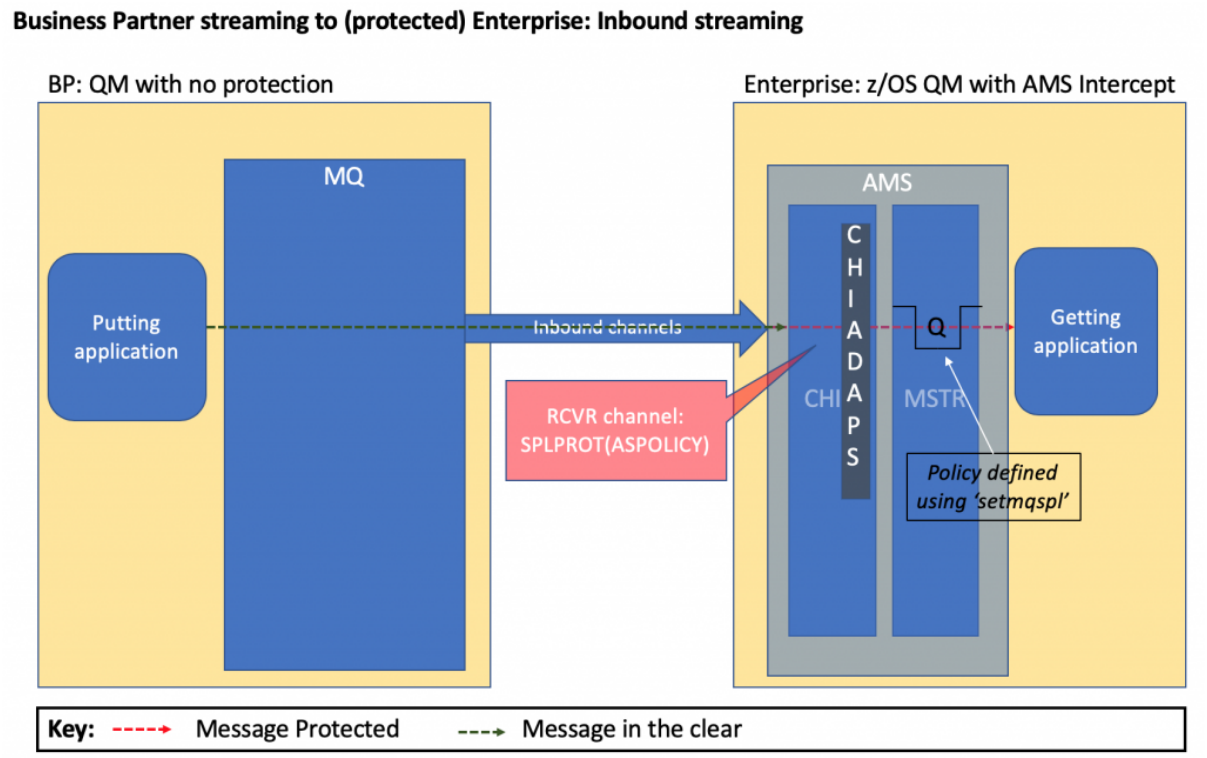
- Allowing the Enterprise to protect their data using AMS qualities of protection without mandating all of their business partners apply AMS protection or even the same AMS quality of protection.

Please note that AMS Interception on Server to Server Message Channels is not the same, and should not be confused with AMS MCA Interception, which is a distributed-only feature and can be used to selectively enable policies to be applied for server connection-type channels.

Business partner streaming data to AMS Interception-protected Enterprise

This scenario aims to demonstrate the impact to a business partner streaming messages to an Enterprise-hosted queue manager configured with AMS Interception on z/OS.

The configuration can be represented thus:



Business partner to AMS Interception-protected Enterprise

In these measurements, the business partner is running on a distributed non-AMS enabled queue manager.

The putting application on the distributed partner is attempting to put messages at a rate of 10,000 2KB non-persistent messages per second.

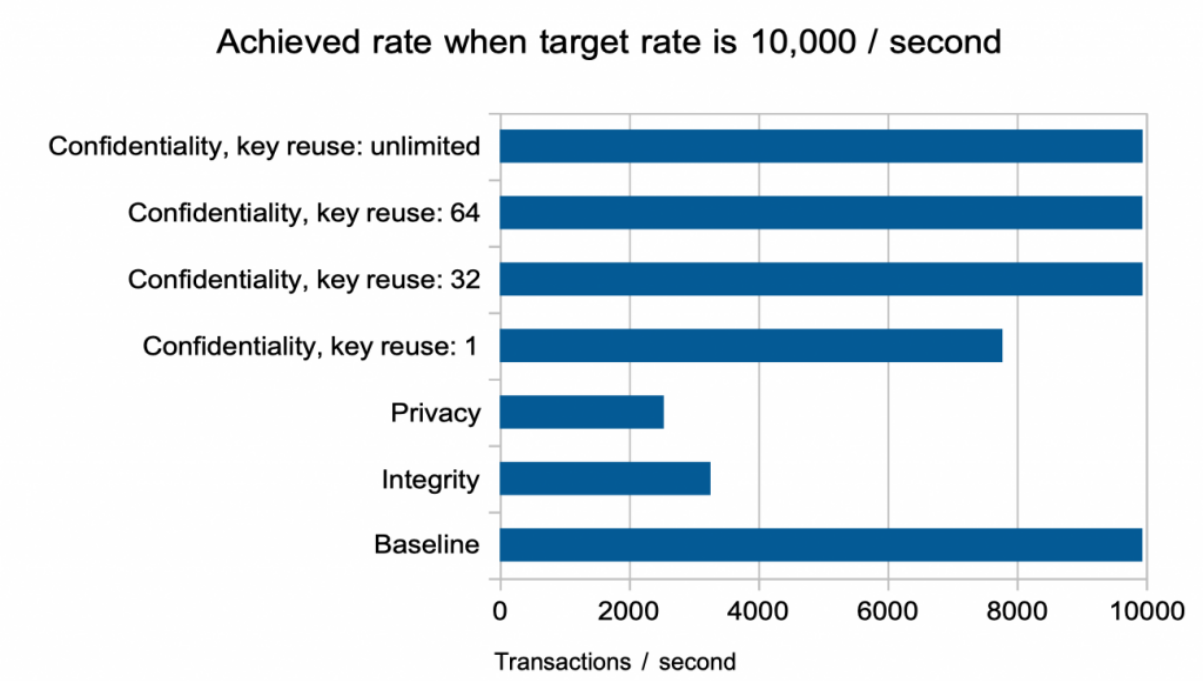
These messages flow over a sender-receiver type channel pair to the AMS-enabled queue manager, where the channel initiator applies AMS protection to the message being put on the target queue.

The authorized application gets and unprotects the message, discarding it after use.

The types of AMS protection used in these measurements are:

- None (baseline)
- Integrity, message is signed with SHA256
- Privacy, message is signed with SHA256 and encrypted with AES256
- Confidentiality, message is encrypted with AES256 and key reused:
  - 1 time
  - 32 times
  - 64 times
  - unlimited.

The first chart shows the achieved throughput rate on the AMS-enabled system.



Business Partner to AMS Intercept-enabled Enterprise – achieved transaction rate

This chart shows that the target rate of 10,000 messages per second was not sustainable in 3 of the 7 configurations, namely Integrity, Privacy and Confidentiality with a key reuse of 1.

When the target rate of 10,000 was not sustainable, the distributed queue manager saw a backlog of messages building up on the queue. In addition, the time spend on the transmission queue (XQTIME) is significantly higher when sending message to the AMS message channel intercept-enabled queue manager.

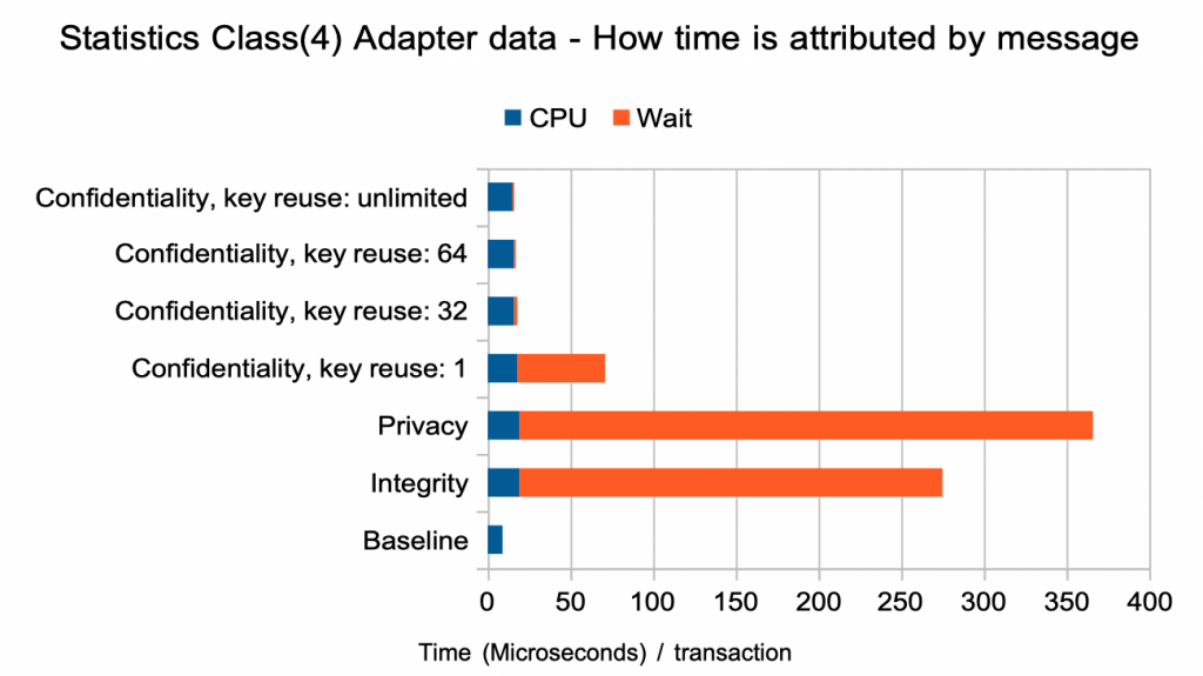
To prevent the putting application failing, 2 changes were made:

1. The MAXDEPTH attribute on the queue on the distributed queue manager was altered to have a higher value than the default of 5000.
2. The application was altered to wait-then-retry when the MQRC_Q_FULL was returned from the MQPUT call.

The backlog of messages on the distributed queue manager was due to the increased time in the adapter task on the AMS-enabled queue manager, which occurred whilst applying protection to the inbound message.

A secondary effect occurred on the z/OS queue manager for the Confidentiality configuration where key reuse of 1 was specified. In this instance, the rate at which the messages were encrypted and put to the queue out-paced the rate at which the getting task was able to remove the messages from the queue. This resulted in page set I/O, and eventually the rate of the arriving messages was slowed to allow for immediate writes to page set. This could have been alleviated with larger buffer pools on the z/OS queue manager.

The second chart represents the class(4) statistics data for the adapter task for each configuration:



Business Partner to AMS Intercept-enabled Enterprise – Statistics Class(4) adapter data

In the above chart, the integrity, privacy and confidentiality with key-reuse 1 configurations show significant time in wait-state. This time spent waiting means the adapter task is blocked from processing other work.

These are the same 3 configurations that were unable to sustain 10,000 messages per second.

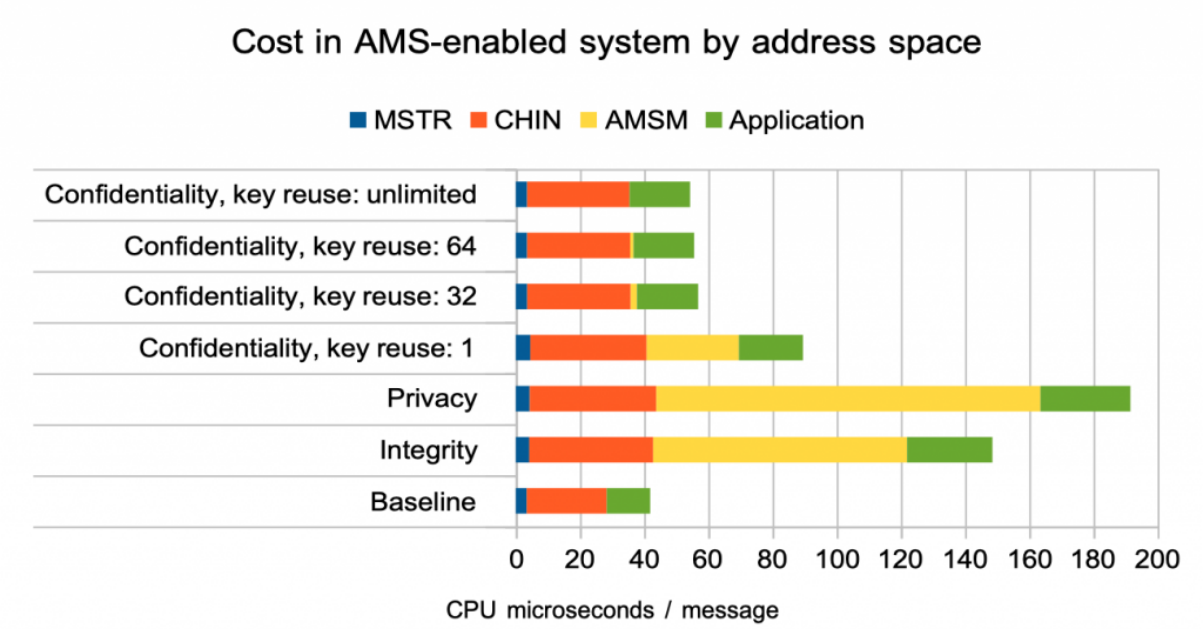
The wait time is calculated from the average elapsed time minus the average CPU time per adapter request.

In this scenario, for the *privacy* and *integrity* configurations, this wait time is largely spent in the cryptographic certificate processing, some of which is charged to the AMSM address space. The *confidentiality key reuse 1* configuration sees wait time, partly due to the immediate writes to page set and partly due to waiting for cryptographic certificate processing.

Note that encryption is performed on CPACF (CP Assist for Cryptographic Functions) and is recorded as CPU time, which is why the wait time on the confidentiality configurations reduces as the key reuse increases.

The final chart in this blog shows the cost per message in the MQ address spaces – MSTR, CHIN and AMSM, as well as the cost in the getting application.

Note that the getting application contains minimal processing apart from the MQGET, so the impact of AMS protection appears more significant than in an application where MQ forms 10% of the processing.



Business Partner to AMS Intercept-enabled Enterprise – breakdown of transaction cost on Enterprise

For the 3 configurations where the target rate of 10,000 messages per second was not achieved, the queue manager and channel initiator cost per message are higher than when the target rate was achieved. This is in part due to the address spaces not being driven as efficiently at the lower messaging rate and in part due to the additional cost of AMS protection, amounting to 10-14 CPU microseconds per message.

When the channel initiator is processing AMS Confidentiality messages at the same rate as unprotected messages, there is an increase in cost per message of 7 CPU microseconds.

The application address space costs were impacted by the type of protection:

- Integrity added 13 CPU microseconds per MQGET
- Privacy added 14 CPU microseconds per MQGET
- Confidentiality added 5 CPU microseconds per MQGET.

Summary

The use case detailed above is just one example of AMS Interception on server to server message channels and aims to demonstrate that applying this new function to your queue manager can affect the queue managers at both ends of the MQ channels.

AMS Interception on server to server message channels provides a solution to a specific problem, for example:

- Ensuring an Enterprise or line of business can protect their MQ data at rest without mandating their partner(s), whether internal or external, implement the same quality of protection, or at the same point in time.

The impact of AMS Interception on z/OS will vary depending on whether you are currently AMS-enabled or not.

It is worth considering whether the increased flexibility from AMS Interception on z/OS outweighs the lower AMS end-to-end protection costs.

Implementing AMS Interception on z/OS may require reviewing your CHIADAPS and CHIDISPS settings on the proposed queue managers in order to minimize the impact on any non-AMS Interception on z/OS workloads.

Using AMS Interception on z/OS may increase the load on your cryptographic hardware – review the RMF Cryptographic report to determine there is sufficient capacity available for the expected increase in cryptographic work.

Applying AMS protection over server-to-server channels, whether in an end-to-end or a message channel interception configuration, will add both cost and latency to the end-to-end transaction. In a system that is constrained for CPU or cryptographic resource, the use of AMS Interception on z/OS could be a significant factor in a change in the behavior of the workload.

Given the impact that AMS protection over server-to-server message channels can have on throughput and latency, it is always worth reviewing your settings for:

- Maximum queue depth
- Maximum message length – AMS protected messages will be larger than their unprotected counterparts.
- Expiry – with the increased latency from AMS Interception on z/OS, are messages going to be expired before they can reach their target destination?

Lastly in the summary, ensure the user ID for the channel initiator applying or removing AMS protection is authorized to perform this processing.

Finally

For the AMS Interception on server to server message channel performance report and indeed other IBM MQ performance reports, keep an eye on the “MQ for z/OS” section on the MQ performance github page.