Protecting messages at rest

63SV_Jonathan_Rumsey
Published on 08/01/2018 / *Updated on 18/01/2018*

I frequently get drawn into discussions around the subject of protecting message data, more often than not this starts off with a high level requirement to encrypt message data whilst it is "at rest". The encryption requirement is usually qualified by ensuring that message data is readable only to applications that have a need to access that data. There are several different approaches that could be used to fulfill these types of requirement, but which one is right for your messages ?
Disk or filesystem-level encryption can look attractive in providing "at rest" encryption, but how do these stack up against an end-to-end security model such as IBM MQ Advanced Message Security (AMS) ?
In this article, let us take a look at what states a message could be held in a messaging environment and what solutions could be applied.
Let us also consider that messages are not always going to be held in the "at rest" state and consider the benefits of deploying an "end-to-end" security model.

Message state

Message data can be considered to be in one of three states;

- "in use" (in main memory)
- "in transit" (traversing a network) or
- "at rest" (held in persistent storage, such as disk)

In use

Message data that is "in use" in a messaging environment could be either in memory in a messaging application program that is producing or consuming messages, but also temporarily in a buffer in the message broker awaiting activity. For example in IBM MQ, whilst a queue remains open by an application or by a channel agent, an in memory message buffer is allocated by the queue manager to allow efficient access to messages on the queue, without needing to go to disk each time a message is produced or consumed by an application.

In transit

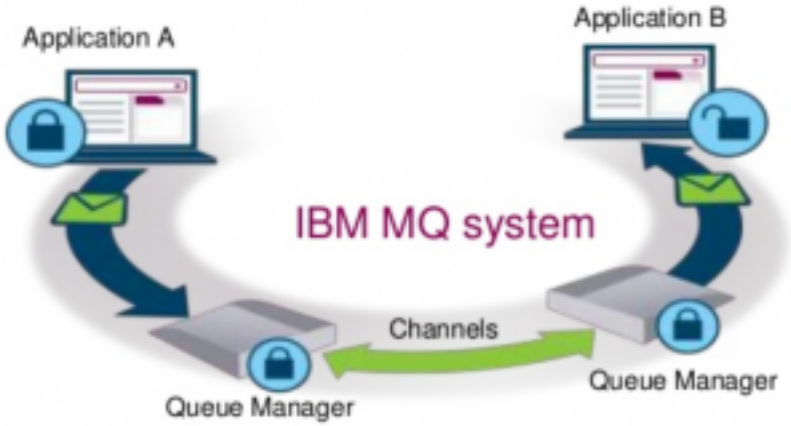
Message data that is "in transit" covers any network based transfers between messaging applications and brokers and broker to broker transfers. Protecting message data whilst it is in transit, using encryption and message digests is typically implemented in the presentation layer of the OSI model using transport layer security TLS, at a level sandwiched between TCP/IP and the application protocol being carried. The requirements around which protocol version of TLS and what encryption and hash algorithms must be used when message data is in transit is usually a prescriptive requirement.

At rest

In an distributed messaging environment, for example IBM MQ, message data will exist in all three states and it would be typical for message data to remain in the "at rest" state for a significant period of time whilst its being held in a queue. There is always a possibility that message data will be written to disk, regardless of message type. For persistent messages writing the message to disk is a relatively direct assumption, but non-persistent messages may also need to be written to disk too, for example when storage in a queue buffer is exceeded messages will spill to a queue file. Whilst it may be possible to reduce the chances of a queue spilling from memory to disk by increasing queue buffer sizes, this is not guaranteed to prevent non-persistent messages from ever being written to disk.
In addition to individual queue files, message data may also be written to recovery logs, trace or FFST dumps and so there are multiple potential locations on disk where message data may be written.

Protection

Full disk and filesystem-level encryption solutions both provide encryption for data at rest, full disk encryption as the name suggests encrypts at a disk partition level, whilst in contrast, filesystem-level can encrypt at the granularity of a file or directory level. In both solutions data is encrypted as it is written and decrypted as it is read. With filesystem-level encryption the finest level of granularity of an encryption key is a single file.
IBM MQ Advanced Message Security (AMS) is part of IBM MQ Advanced, it offers a policy driven end-to-end security model that continually protects message data whilst it is "in use", "in transit" and "at rest". Messages can be encrypted for one or more intended recipients and in addition a digital signature can be added to each message to prove the authenticity of a message which could be used to provide a non-repudiation solution. The finest level of granularity of an encryption key with AMS is an individual message.



IBM MQ Advanced Message Security (AMS) and disk/filesystem operate at different levels and can be layered together.
Lets look again at the three states and see how these protection solutions apply;

In use

Message data that is "in use" will need to be readable by the applications that are intended to consume messages.
Message data that is in memory is beyond the boundary of disk/filesystem-level encryption as data is always unprotected as files are successfully read from the disk/filesystem. Taking

IBM MQ as an example, this would mean when using disk/filesystem encryption that data is always in plaintext whilst it is “in use” and any user that has authority to a queue or an alias, including administrators, would be able to view all messages or inject new messages onto the queue. If a message needed to be moved to a dead-letter queue, for example if a destination queue were to be full, this would also mean the rerouted message would be in plaintext. The effectiveness of this solution relies on strong authentication and restrictive access controls.
IBM MQ Advanced Message Security (AMS) offers some key advantages over disk/filesystem level encryption here as it will only ever decrypt message data back into plaintext at an endpoint, that is within the memory address space of a consuming application and then only if the application can present a private key that matches one of the intended recipients of the message. A further advantage over disk/filesystem encryption is that once a message reaches the intended consuming application any digital signature that was required by the policy can be checked to prove the origin and authenticity of the message.

In transit

All modern messaging environments will offer the ability to configure transport level encryption, it would be incredibly unlikely for messages that have requirements for encryption whilst at rest to not require protection whilst in transit.
Messages that are in transit are not protected by disk/fileystem encryption.
IBM MQ has supported the capability of setting a single prescriptive CipherSpec (a combination of protocol, cipher & digest) for channels in base product since 2002. Mutual certificate authentication and mapping features are also configurable to provide a more flexible and stronger security over the network connection.
Whilst IBM MQ Advanced Message Security (AMS) does not involve itself in any way in the network transfer of message data, messages remain protected until they reach a consuming application and so even if messages were to be transferred over a plaintext connection the message would remain protected. Applications using AMS are also able to use the same X.509 digital certificates used for TLS mutual authentication.

At rest

There are lots of different ways of protecting message data at rest, but here we will focus on encrypting of the data to prevent unauthorized viewing, injection and modification.
As stated earlier, disk and filesystem encryption both work in similar ways, but with contrasting levels of granularity with a file being the finest level of granularity of an encryption key. Given that message data might be written to multiple queue files, including shared transmission queues and recovery logs, it would not be possible to fulfill the requirements for two disparate applications, even with finest level of granularity offered by filesystem-level encryption.
By providing a true end-to-end protection security model, messages protected by IBM MQ Advanced Message Security (AMS) remain protected throughout their entire lifecyle between producer and consumer, so no matter what queues the message travels through the queue files and recovery logs will always contain the protected copy of the message and not the plaintext data. Having the assurance that the message will remain protected between producing and consuming applications and only viewable to the intended recipients, no matter what queues or channels the messages passes through is perhaps the most compelling reason why you’d choose IBM MQ Advanced Message Security (AMS) to protect MQ messages.