How to enable MLE in a BigFix deployment

Authors: DAVIDE COSENTINO and ADOLFO BURGIO

Introduction

Reports from the BigFix clients can contain sensitive data that you want to secure, especially when traversing from clients to relays to the server through the internet. Does BigFix include any capability to achieve it? Of course, and the feature is called "**Message Level Encryption"**. Message Level Encryption (MLE in short) allows BigFix clients to secure the upstream data by leveraging on an RSA public key infrastructure.

The enablement of MLE happens at two levels:

- **Server side**: Generate a private key by using the BESAdmin tool on the Server and automatically propagate the server's public key to the clients (through the masthead).
- **Client side**: Run a dedicated BigFix task.

Figure 1 shows the typical flow of MLE:

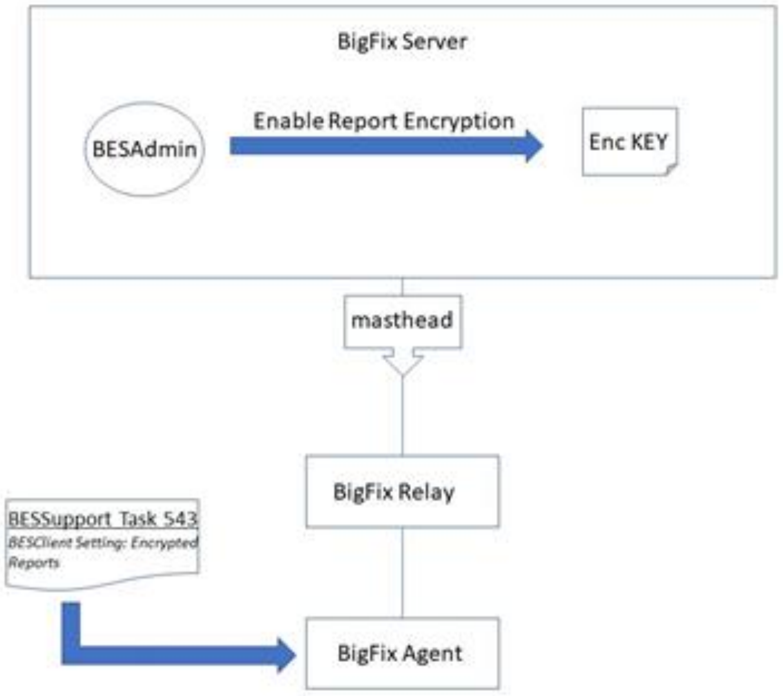


Fig. 1 MLE enablement main flow

Enabling MLE on the server side

First off, create a key on the server to prepare it for decryption. You can do this by using either a CLI or through GUI (the latter onlyon Windows). The key is saved in the following folder on the server:

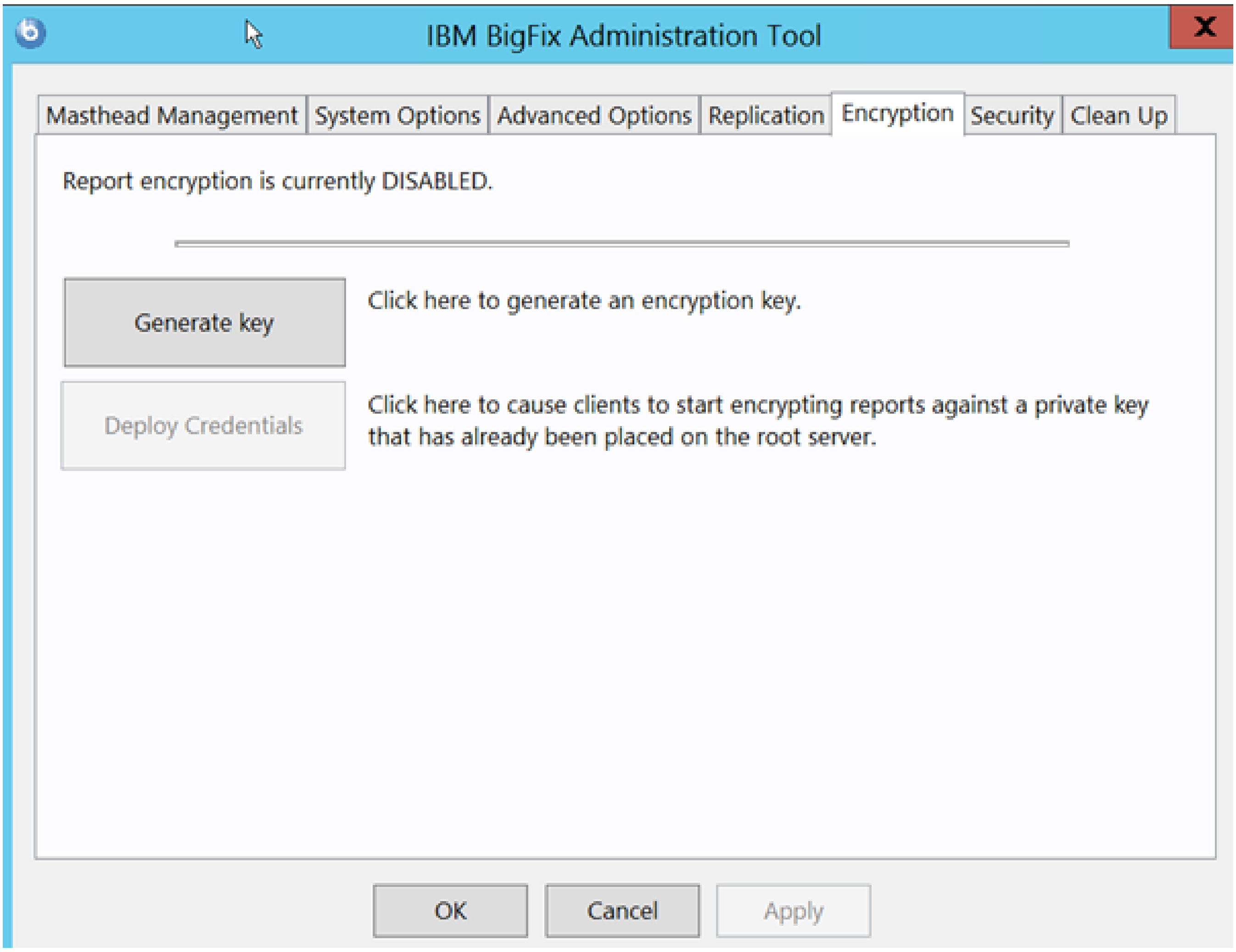
<BESServer installation dir>/Encryption Keys

If you wish to use the CLI, run any of the following commands:

LINUX: *./BESAdmin.sh -reportencryption -sitePvkLocation="<license.pvk path>" -sitePvkPassword=<license password> -generatekey -deploynow=yes*

WINDOWS: *BESAdmin.exe /hideUI /reportencryption /sitePvkLocation: <license.pvk path> /sitePvkPassword:<license password> /generatekey /deploynow=yes*

If using the BESAdmin GUI (Windows-only), open the "Encryption" tab and click "Generate Key" .

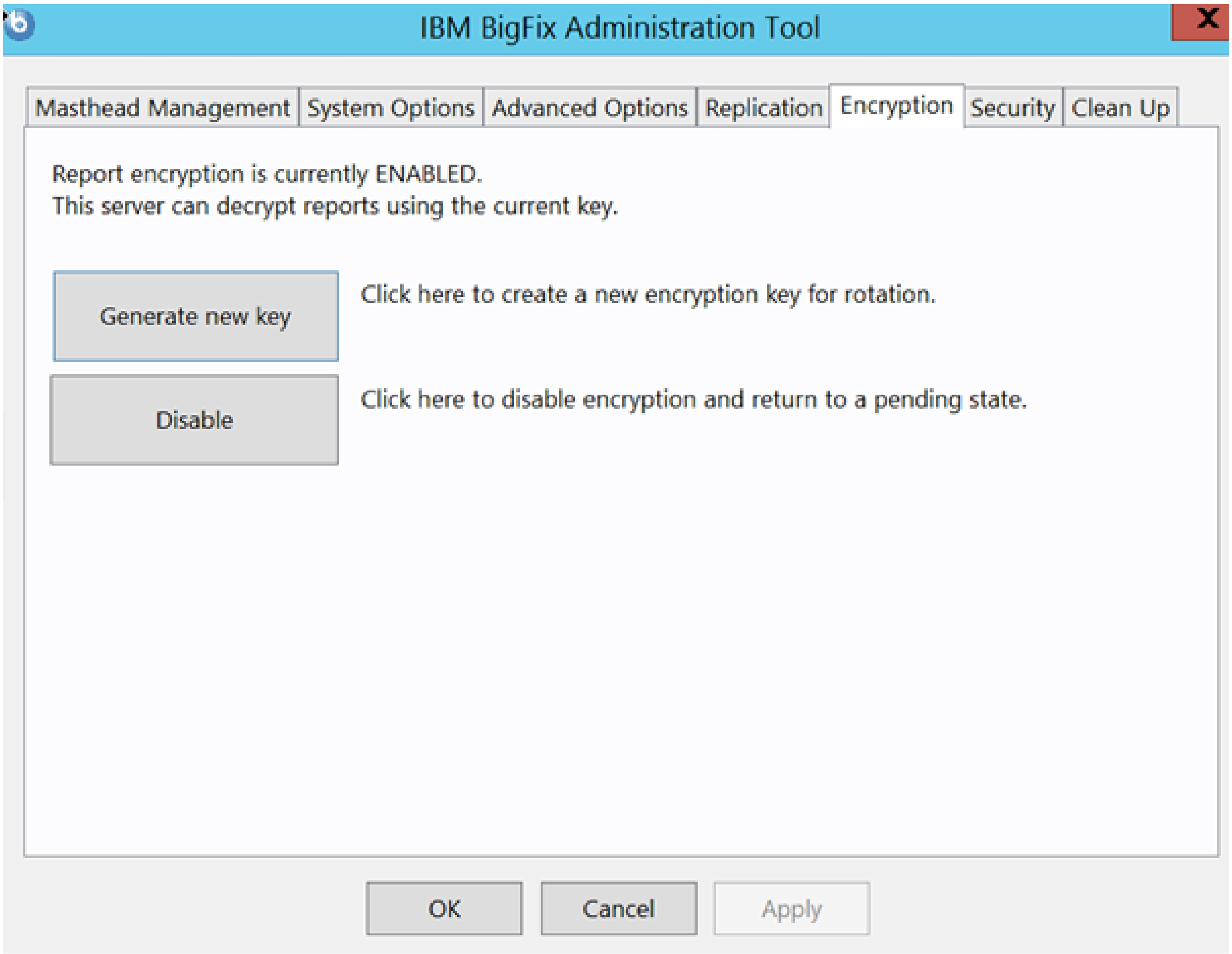


Report encryption is currently DISABLED.

Click here to generate an encryption key.

Click here to cause clients to start encrypting reports against a private key that has already been placed on the root server.

Fig.2 Make the Server ready for MLE by clicking on "Generate Key"

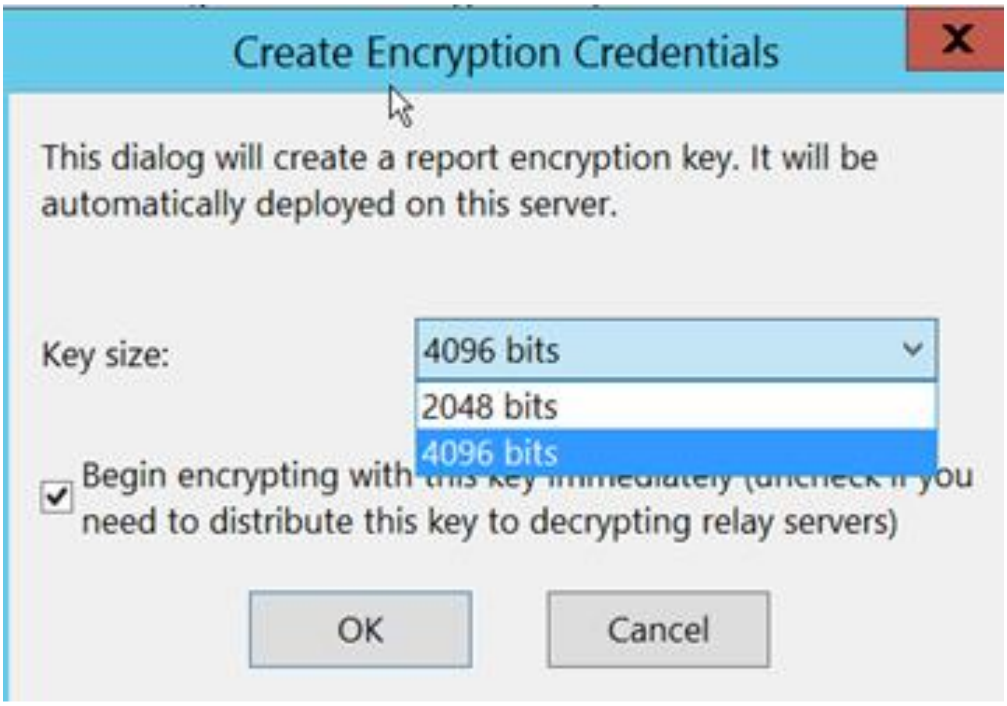


Report encryption is currently ENABLED.
This server can decrypt reports using the current key.

Click here to create a new encryption key for rotation.

Click here to disable encryption and return to a pending state.

Fig. 3 Encryption is now enabled and the Server is ready for MLE



Create Encryption Credentials

This dialog will create a report encryption key. It will be automatically deployed on this server.

Key size: 4096 bits

☑ Begin encrypting with this key immediately (uncheck if you need to distribute this key to decrypting relay servers)

Fig. 4 Choose the complexity of the encryption

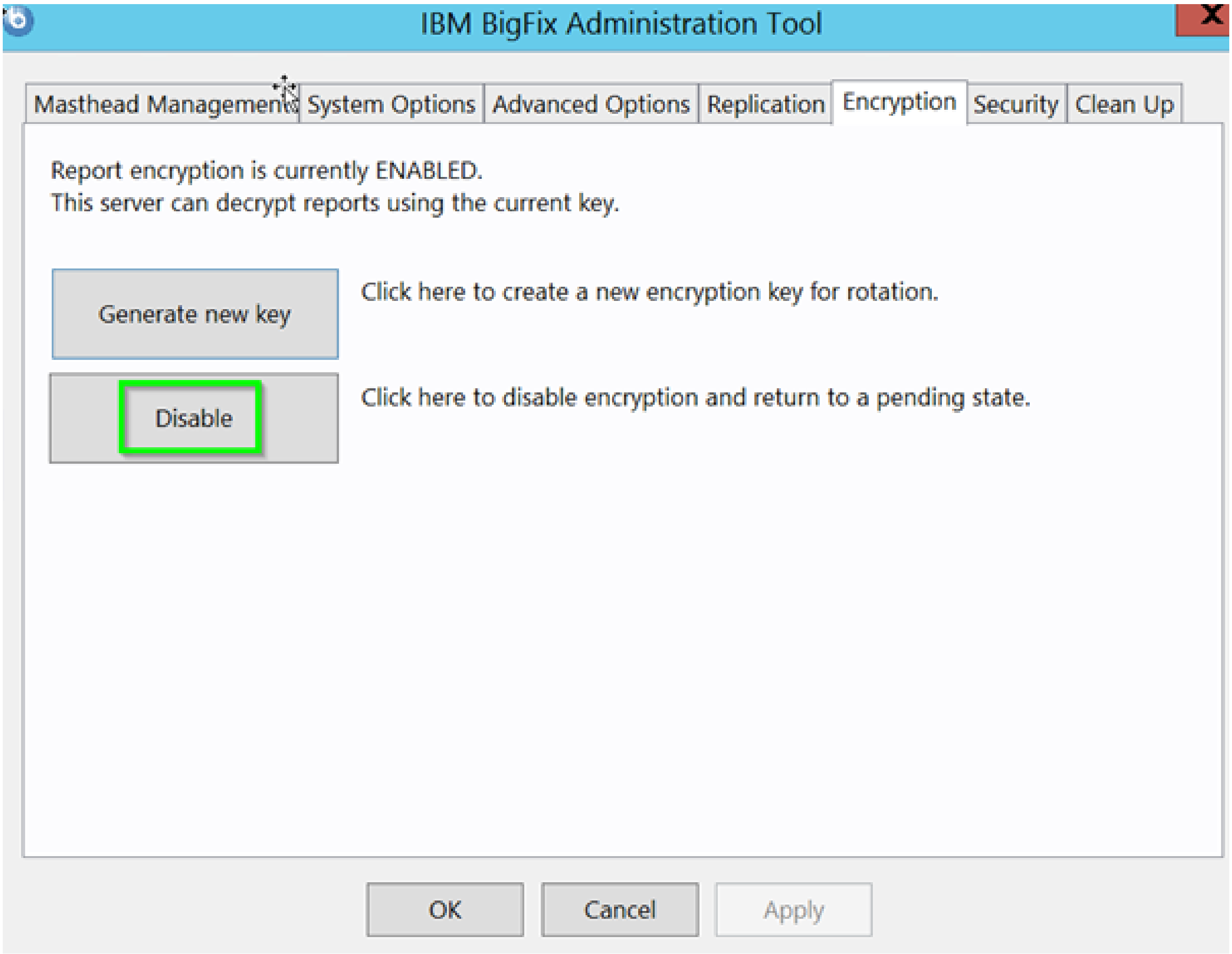
To disable the encryption, run the following commands in a CLI:

LINUX: *./BESAdmin.sh -reportencryption -sitePvkLocation="<license.pvk path>" -sitePvkPassword=<license password> -disable*

WINDOWS: *BESAdmin.exe /hideUI /reportencryption /sitePvkLocation: <license.pvk path> /sitePvkPassword:<license password> /disable*

This action disables the report encryption and deletes the key.

On the BESAdmin GUI, go to the "Encryption" tab and click "Disable":



Report encryption is currently ENABLED.
This server can decrypt reports using the current key.

Click here to create a new encryption key for rotation.

Click here to disable encryption and return to a pending state.

Fig. 5 Click "Disable" to disable the encryption

Enablement on the client side

What are my prerequisites?

Private key created on the server and public key (automatically) propagated to the clients.

You still need to mandate the encryption on each client of interest. Run the following task available on the BES Support site: *BESClient Setting: Encrypted Reports (ID 543)*

Choose any of the following three options to establish the behavior on the client side:

1. **Require**: if the BES Client is set to require encryption to send reports, the BES Client will not send any reports if no encryption certificate could be found, or if the parent relay does not support receipt of encrypted documents.
2. **If possible**: if the BES Client is set to use encryption if possible, the BES Client will attempt to use encryption but will send its reports in cleartext if encryption is impossible

3. **Disable**: if the BES Client is set to disable encryption, the BES Client will not encrypt its reports to the BES Server even if encryption is possible.

Performance and Scalability considerations

To support MLE, BigFix Server requires additional resources so that it can process encrypted reports received from the clients.

Server hardware recommendations are as follows:

Additional CPU	Additional Memory	Additional Storage
+2	+4GB	n/a

Fig. 6 Performance and Scalability

These hardware additions provide for more processing headroom in multi core environments. For more detail on BigFix capacity planning, the BigFix Capacity Planning Guide is available here.

If your deployment is very large, or you are using an encryption key strength of 4096 bits, BigFix highly recommends configuring one or more decrypting top level Relays (with 2-4 CPU cores each) to help distribute the additional processing load. The reports delivered to the server from the decrypting relays are in cleartext format.

Troubleshooting on the Server side

When things do not work as expected on the server side, what can you do to troubleshoot?

Enable the Encrypted Carbon Copy feature; to do so, add the following setting to the Global Option section of client configuration:

[Software\BigFix\EnterpriseClient\GlobalOptions]

EncryptedCarbonCopyPath = <carbon copy dir>

Verify that:

- Encrypted reports are copied in to the *<carbon copy dir>* with ".enc" extension. Each encrypted file starts with the following tag: ##SE001

The Decrypting Relay feature

How do you eliminate the CPU load from the main server? It's pretty straight-forward - enable a Relay to decrypt reports and pass them on to the Server. A Decrypting Relay decrypts all the client reports as applicable and forwards them to the main server. Unless you have thousands of clients in your deployment or if your main Server CPU load is too high for any reasons, you don't need a Decrypting Relay.

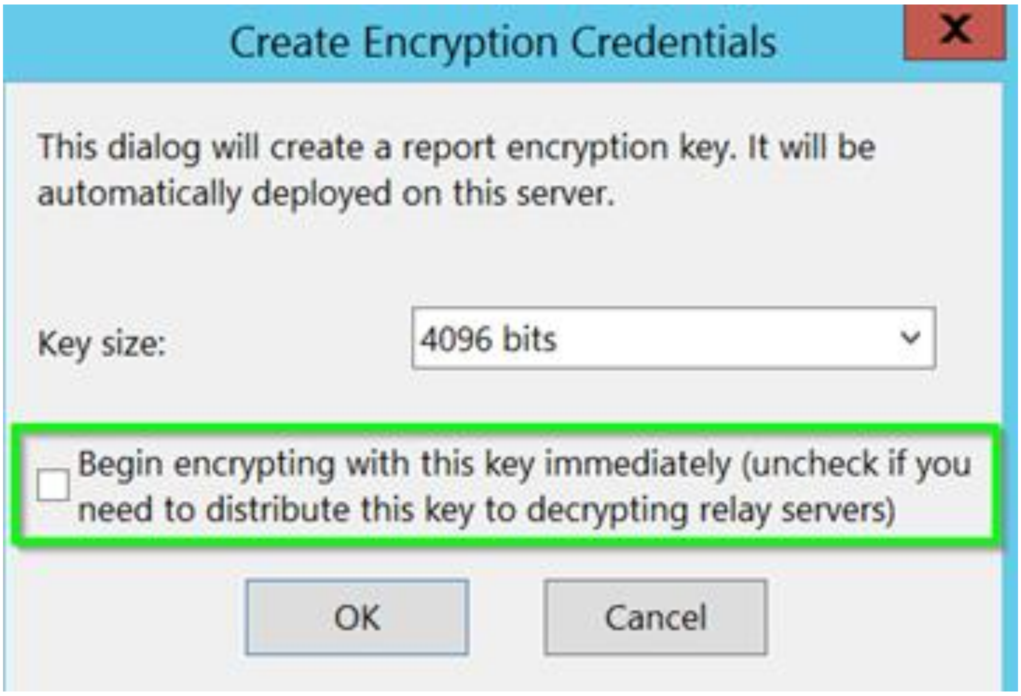
To enable a Decrypting Relay, run the following command:

Note that it's the same BESAdmin command as described above, but with *deploynow set to no*.

LINUX: *./BESAdmin.sh -reportencryption -sitePvkLocation="<license.pvk path>" -sitePvkPassword=<license password> -generatekey -deploynow=no*

WINDOWS: *BESAdmin.exe /hideUI /reportencryption /sitePvkLocation: <license.pvk path> /sitePvkPassword:<license password> /generatekey /deploynow=no*

On the GUI (Windows only), clear the “Begin encrypting with this key immediately” check box.



This dialog will create a report encryption key. It will be automatically deployed on this server.

Key size: 4096 bits

□ Begin encrypting with this key immediately (uncheck if you need to distribute this key to decrypting relay servers)

Fig.7 "Delayed" MLE to allow for Decrypting Relay enablement

This option helps you prevent clients from starting to send encrypted reports before the relays are ready to receive them. During this period, the key gets copied in to the following folder in the relay machine(s):

<BES Relay Root Path>/Encryption Keys

Restart the relay service and yes, you are good to go now!

You are free to decide how to copy the key from the Server over to the relays. Bear in mind though that it’s a private key, and hence you should be extra careful while copying.

Further Reading

In the event further reading is desired on BigFix or MLE itself, the following technical resources are available.

BigFix Platform Documentation: URL

BigFix MLE Overview: URL

Authors' Bio

***Adolfo Burgio** is a Senior Software Engineer at Bigfix Team. After 5 years on the Z/OS platform he joined the Bigfix project and took several responsibilities on automation team. He currently works as automation designer and developer on this team, being responsible of all the automation suites of the project*

***Davide Cosentino** is a BigFix Performance and Scalability Engineer at HCL. After 17 years at IBM where he was in Software Product Development covering roles like Coder, Tester and Level 3 Support Engineer for many different deliverables, he joined HCL in 2018. He's now part of the Performance and Scalability Team since 5 years, with major expertise in the BigFix Platform and the WebUI.*