



CRYPTOMATHIC



CYBERARK®

CYBERARK PAS CRYPTO SERVICE GATEWAY INTEGRATION GUIDE

Name of Company: Cryptomathic

Website: <https://www.cryptomathic.com/>

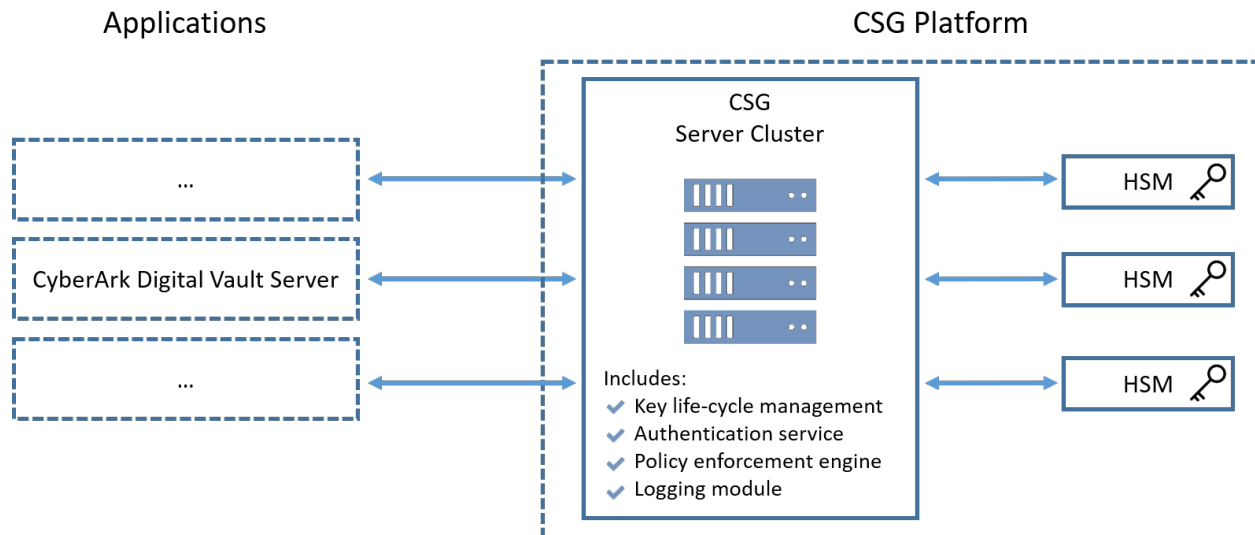
Name of Product: Crypto Service Gateway (CSG)

Version: 3.17

Date: 07/11/23

INTRODUCTION

CSG (Crypto Service Gateway) is a central cryptographic platform that simplifies application integration while ensuring the highest availability and utilization of HSMs. CSG shares HSM resources between applications, allowing central policy enforcement and management of HSMs. With CSG, a business can assert total control over its crypto estate, delivering increased efficiency, cost savings and confident compliance.



KEY BENIFITS

Benefits of CSG:

- Reduces costs through shared infrastructure and increased HSM utilization
- Centralizes policy and control over all crypto operations; keeping crypto decisions in the hands of the security team
- Enables complete central management and monitoring over entire HSM real-estate
- Provides easy to read audit logs for proof of compliance
- Simple-to-use API delivers increased development velocity and reduced time to market

Benefits of integrating CSG with CyberArk Vault:

- Introduces an additional layer of security in terms of key usage protection.
- Takes advantage of the High Availability Architecture that CSG provides.
- Organisations can manage CyberArk keys through the same cryptographic platform as their other applications.

REQUIREMENTS

The following requirements must be fulfilled for the integration to be successful.

- CSG must be installed and be fully operational, with HSMs attached (please see the “Cryptomathic Crypto Service Gateway Installation, Configuration and Management” manual for detailed steps.)
- CyberArk Digital Vault Server must be installed and be fully operational

PREPARING CRYPTOMATIC CSG

Before the integration can be performed the following configuration of the Cryptomathic Crypto Service Gateway must be performed:

- Client API binding for access
- Certificate based authentication must be set up
- Crypto policy must allow access
- Configuration of PKCS#11 library

These items are covered in the following subsections.

PREREQUISITES

As the communication between CyberArk and CSG is TLS protected it is required to establish CSG Server and CyberArk Client TLS certificates before the configuration of CSG can be started. The certificates must fulfil standard TLS requirements and the Client Certificate must furthermore contain a DN that is unique across all the CSG Client Applications. The DN is used to distinguish the individual application accessing CSG.

ACCESS CONFIGURATION

For CyberArk to be able to access the CSG, a Client API binding must be configured through the CSG Administration Client.

Name:	TLS SAPI Certificate	
Host:	<Hostname>	
Port:	9997	
Transport Protocol:	SAPI ▼	
Transport Layer Security:	Mutual authentication ▼	
Check CRL	☐	
User Authentication Scheme:	Certificate	
Key Store:	Show certificate	<button>Choose File</button>
Password:		
Trust Store:	<button>Download</button>	<button>Choose File</button>

The following settings are especially relevant for the CyberArk integration:

- Host: The hostname of the CSG server access point.
- Port: The port of the CSG server access point.
- Transport Protocol: SAPI or RESTful protocol.
- Transport Layer Security: Must be set to “Mutual authentication”
- Key Store: Must contain the CSG server TLS keys + certificate
- Trust Store: Must contain trust relation to the CyberArk client TLS certificate (Root, intermediate or leaf certificate)

For further details please refer to the “Crypto Service Gateway Installation, Configuration and Maintenance” manual.

AUTHENTICATION

To be able to authenticate the CyberArk client, certificate-based authentication must be configured through the CSG Administration Client.

The screenshot shows a configuration window with the following fields and values:

- Name:** LOCAL:CyberArk
- User Authentication Scheme:** Certificate (selected from a dropdown menu)
- Password:** (empty text field)
- Re-enter:** (empty text field)
- Associations:** A table with two columns: Subject and Issuer.

Subject	Issuer
cyberark	Intermediate CA

Below the table, the full DN is displayed: Subject: CN=cyberark, Issuer: CN=Intermediate CA

The following setting must be set:

- **Name:** An application reference (CSG Username) in the form of “LOCAL:<application name>”
- **User Authentication Scheme:** Must be set to “Certificate”
- **Associations:** The DN of the CyberArk client TLS certificate must be added as association. This is done by uploading the CyberArk client TLS certificate.

For further details please refer to the “Crypto Service Gateway Installation, Configuration and Maintenance” manual.

AUTHORIZATION

For the CyberArk client to be authorized to perform cryptographic operations the CSG Username must be added to the CSG Crypto Policy.

The following is an example of the section to include in the CSG Crypto Policy, which contains the required attributes:

```
[USER LOCAL:CyberArk]
TicketValidity = 1 day
ExtendMode = Never
MustBeLogged = false
```

For the CyberArk integration only the MustBeLogged attribute is relevant. This specifies whether cryptographic operations are permitted if usage logging cannot be performed (for example in cases when no disk space is available). This should be set to true if required for demonstrating compliance, otherwise set to false.

The other attributes are required as part of the “User” section, but they do not apply to this type of integration and the value can therefore be ignored.

For further details on the CSG Crypto Policy format please refer to the “Crypto Service Gateway System Configuration” manual.

For further details on updating the CSG Crypto Policy please refer to the “Crypto Service Gateway Installation, Configuration and Maintenance” manual.

PKCS#11 LIBRARY CONFIGURATION

As the communication between CyberArk and CSG is based on PKCS#11, the CSG PKCS#11 library must first be installed and configured on the CyberArk host. To do so the PortalPKCS11.dll and PortalPKCS11.properties files, which are placed within the *Integration\APIs\Api.Pkcs11* folder of the CSG installation package, must be installed to the CyberArk host machine and be accessible by the CyberArk service. Please note that the service user must have read and write permissions to this location.

The PortalPKCS11.properties file must then be updated to include CSG communication settings, including authentication keys and certificates. The following is an example of the configuration file:

```
log.folder.path=C:/portalpkcs11/logs
log.level=1
delay.between.auto.recovery.attempts=300
authentication.mode=CERTIFICATE
client.certificate.path=C:/portalpkcs11/cyberark_combined.pem
portal.server.list=node1
portal.node1.uri=127.0.0.1:9997
portal.node1.certificate.path=C:/portalpkcs11/combined_root_inter.pem
portal.node1.concurrent.clients=2
timeout.connection=15000
timeout.command=60000
timeout.keep.alive=600000
```

The important settings are:

- *authentication.mode*: Must be set to CERTIFICATE
- *client.certificate.path*: An absolute path to a PEM encoded file containing the client certificate and encrypted private key used for mutual TLS.
- *portal.<node>.uri*: “hostname:port” corresponding to the configured CSG client API binding.
- *portal.<node>.certificate.path*: An absolute path to a PEM encoded certificate chain used to verify the TLS certificate of the CSG server. Please note that this must contain the root and if applicable the intermediate certificate.

For further details please refer to the “Crypto Service Gateway PKCS11” manual.

VAULT CONFIGURATION

1. To interact with CSG the Firewall must be configured to allow communication to the remote CSG server. In DBParm.ini, configure the AllowNonStandard FWAddresses parameter to open the Firewall and allow access to the CSG server, as shown in the following example:

```
AllowNonStandardFWAddresses=[<CSG-IP>],Yes,<CSG Client API Port>:inbound/tcp,<CSG Client API Port>:outbound/tcp
```

2. Configure the CSG PKCS#11 provider DLL and specify it in the PKCS11ProviderPath parameter in DBParm.ini, as shown in the following example:

```
PKCS11ProviderPath=<path to the CSG PortalPKCS11.dll >
```

3. As CSG implements and supports HSM failover and recovery on its own the vault functionality for reconnecting an HSM on error codes must be disabled. This is done by ensuring that the following parameter in DBParm.ini is set blank:

```
ReconnectHSMOnErrorCodes=
```

4. Save DBParm.ini and close it.
5. Define the passphrase to be used by the Vault to unlock the CyberArk Client TLS key, when accessing CSG:
From a command line, run the following command, specifying your own passphrase that will be used to access the Server key. The passphrase cannot begin with "/":

```
CAVaultManager SecureSecretFiles /SecretType HSM /Secret <TLS private key passphrase>
```

6. Restart the PrivateArk Server to apply the new Firewall rules.
7. Shutdown the PrivateArk Server.

LOAD THE SERVER KEY IN CSG

The following process installs and stores the Server key in CSG. Once this process is complete, the server key is stored as non-exportable key in CSG and will be used by the Vault.

Install key on CSG

1. Make sure that the Vault Server is not running.
2. Load the Server key to CSG:
 - a. From a command line, run the following command:

```
CAVaultManager.exe LoadServerKeytoHSM /WrapKey
```

This will generate a new key pair. The public key will be used to encrypt the server key, and the private will decrypt it on the HSM device.

3. Make sure that the result confirms that the Server key has been loaded to CSG.
4. In DBParm.ini, change the value of the ServerKey parameter as follows:

```
ServerKey=HSM
```

5. Start the PrivateArk Server and make sure you can log on to the Vault.

The Server key has been successfully moved to CSG and will be used for all relevant CyberArk Vault operations.

GENERATE THE SERVER KEY IN CSG

1. Make sure that the Vault Server is not running.
 2. Run the **CAVaultManager** command to generate the server key in CSG:
- ```
GenerateKeyOnHSM /ServerKey
```
3. The above command will generate a new key for the Vault server and store it in CSG and will return the key generation keyword. For example: HSM#5
  4. Each time a key generation is created, the keyword allocated is one number higher than the current server key generation specified in DBParm.ini. The HSM can store up to 255 key generations, after which key generation numbering will begin again at one. In order to create additional key generations successfully, users have to manually delete the first generation of the server key, otherwise an error will be returned. If the ServerKey parameter in the CAVaultManager command specifies a path instead of an HSM keyword, the first key generation will be created, i.e., HSM#1.

3. Re-encrypt the Vault data and metadata with the newly generated keys on the HSM.

```
ChangeServerKeys PathToKeys PathToEmergencyFile HSMKeyword
```

Running the ChangeServerKeys will change the encryption keys that will be used for the Vault server.

4. For example, the following command will re-encrypt the Vault data and metadata with the encryption keys in 'K:\PrivateArk\Keys', and the 'HSM#1' key will be used as the server key.

```
ChangeServerKeys K:\PrivateArk\Keys K:\PrivateArk\Keys\VaultEmergency.pass HSM#1
```

4. Open DBParm.ini and in the ServerKey parameter specify the value of the key generation version that was generated and specified in the output of the CAVaultManager command above, as shown in the following example.

```
ServerKey=HSM#1
```

5. Start the Vault server and make sure you can log onto the Vault.

## PARTNER CONTACT INFO

|                   |       |                                                                                    |
|-------------------|-------|------------------------------------------------------------------------------------|
| Business Contact  | Name  | Johannes Lintzen                                                                   |
|                   | Email | <a href="mailto:jo.lintzen@cryptomathic.com">jo.lintzen@cryptomathic.com</a>       |
|                   | Tel   |                                                                                    |
| Technical Contact | Name  | Glen Leonhard                                                                      |
|                   | Email | <a href="mailto:glen.leonhard@cryptomathic.com">glen.leonhard@cryptomathic.com</a> |
|                   | Tel   |                                                                                    |
| Support Contact   | Name  | Jonas Hartvig Søholm                                                               |
|                   | Email | <a href="mailto:jonas.soholm@cryptomathic.com">jonas.soholm@cryptomathic.com</a>   |
|                   | Tel   |                                                                                    |