Encryption in Oracle GoldenGate

by Ahmed Baraka

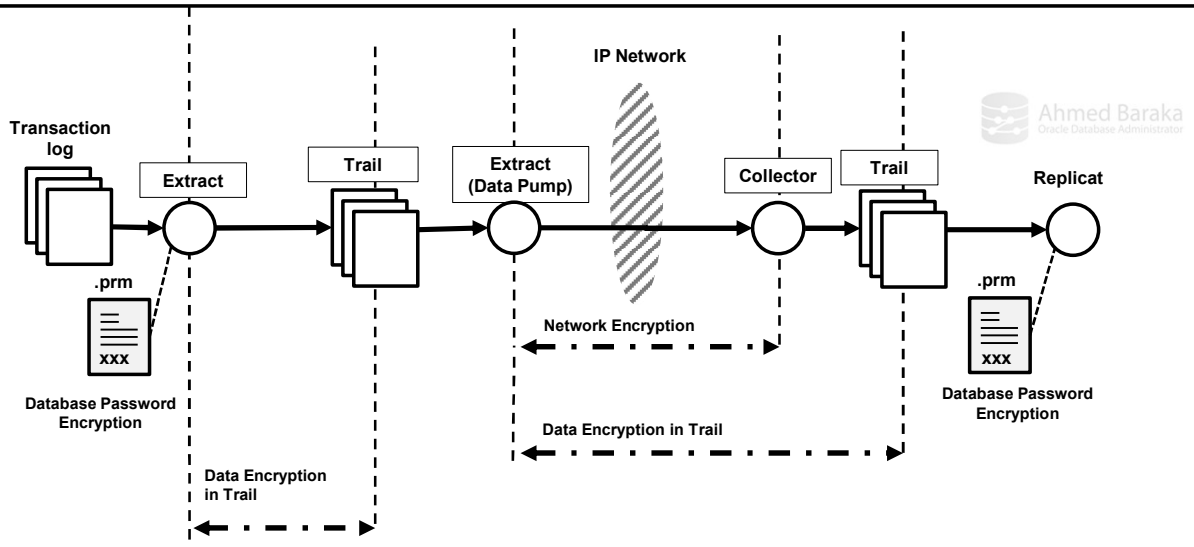
Introduction to Oracle Data Guard
In this lecture, we are going to talk about the basic concepts of Oracle Data Guard

Objectives

By the end of this lecture, you should be able to:

- Use the Encryption Key method and the Wallet method to encrypt the network packets, trail files, and database login credentials

Encryption in Oracle GoldenGate



Oracle GoldenGate Encryption: Overview

- **What to encrypt?**
  - TCP/IP messages
  - The data in the trail files
  - Database login credentials
- **Encryption methods:**
  - ENCKEYS
  - Wallet-based

Oracle GoldenGate Encryption: Overview

- Supported encryption standard ciphers:
  - Advanced Encryption Security ciphers: AES-128, AES-192, AES-256
  - Blowfish (must be used on iSeries, z/OS, and NonStop)

Encrypting Messages with ENCKEYS Method

1. (optional) use keygen utility to generate random hex keys

```
./keygen <key_length> <number_of_keys>
```

2. Enter key names and values in the file ENCKEYS in the Oracle GoldenGate directory. Copy the file to all systems.
3. Configure the Extract parameter file:

```
RmtHost dest, Port 7809, Encrypt AES256, KeyName mykey
```

4. Configure a static Server Collector:

```
server -p 7809 -Encrypt AES256 -KeyName mykey
```

Encrypting the Data with ENCKEYS Method

1. Generate an encryption key and store it in the ENCKEYS file.
2. For the primary Extract, add the following before the trail file (`EXTTRAIL`, `RMTTRAIL`, `EXTFILE`, and `RMTFILE`):

```
ENCRYPTTRAIL AES192, KEYNAME mykey
```

Decrypting the Data with ENCKEYS Method

- Network encrypted data is automatically decrypted at the destination by the collector
- Data encrypted in trail must be decrypted by the Replicat before it is applied to the destination. Use `DECRYPTTRAIL`.

```
DECRYPTTRAIL AES192
```

- Data pump cannot process encrypted data before decrypting it first.

Encrypting a Password with ENCKEYS Method

1. Generate an encryption key and store it in the ENCKEYS file.
2. Issue the `ENCRYPT PASSWORD` command:

```
Encrypt Password <password> <encrypt_type>
 EncryptKey {<key_name> | DEFAULT}
```

```
ENCRYPT PASSWORD mypassword AES256 ENCRYPTKEY mykey1
```

3. Specify the encrypted password in the parameter file (cont..)

Using Passwords Encrypted with ENCKEYS

- Database login in parameter files:

```
USERID <user>, PASSWORD <password>, <algorithm>
 ENCRYPTKEY {<keyname> | DEFAULT}
```

- Database login in GGSCI:

```
DBLOGIN USERID <user>, PASSWORD <password>, <algorithm>
 ENCRYPTKEY {<keyname> | DEFAULT}
```

- Complete list options in documentation “Administering Oracle GoldenGate for Windows and UNIX 12c”

About Oracle GoldenGate Wallet

- Introduced in Oracle GoldenGate release12c
- Contains master keys
- Is used in encrypting the TCP/IP messages and the data in the trial
- Platform-independent format
- Creates keys used with AES
- Encrypting data with a master key and wallet is not supported on the iSeries, z/OS or NonStop platforms

Wallet-related Commands in GGSCI


```
Create Wallet
Open Wallet
Add MasterKey {mykey}
Renew MasterKey {mykey}
Info MasterKey { ALL | mykey }
Delete MasterKey ALL
Purge Wallet
```

Encrypting Data using the Wallet

1. Create a master-key wallet (open it, if there is only already):

```
CREATE WALLET
```

2. Add a master key to the wallet:

```
ADD MASTERKEY
```

3. Obtain the current version and its AES hash value:

```
INFO MASTERKEY
INFO MASTERKEY VERSION <version>
```

4. Copy the wallet file to all other systems

Encrypting Data using the Wallet (cont)

5. Encrypt the data and /or the TCP/IP messages:
   - To encrypt data across TCP/IP: in the parameter file of the Data Pump:

```
RMTHOSTOPTIONS host, MGRPORT port, ENCRYPT {AES128
| AES192 | AES256 | BLOWFISH}
```

   - To encrypt the data, in the parameter file of the primary Extract and the Data Pump, add the following parameter statement before the trail or file configuration:

```
ENCRYPTTRAIL {AES128 | AES192 | AES256 | BLOWFISH}
```

Encrypting Data using the Wallet (cont)

6. Decrypt the data
   - Replicat decrypts the data automatically without any parameter input.
   - If you want the Data Pump to decrypt the data before it writes it to the remote trail files:

```
DECRYPTTRAIL
```

Changing the Master Key

1. Stop the Extract
2. Wait till the Replicat applies all pending data

```
SEND REPLICAT <replicat group name> STATUS
```

3. Stop the Replicat
4. Open the Wellet and confirm the Master Key version

```
OPEN WALLET
INFO MASTERKEY
```

Changing the Master Key (cont)

4. Renew the Master Key

```
RENEW MASTERKEY
```

5. Obtain the new generated hash value

```
INFO MASTERKEY
INFO MASTERKEY VERSION <new_version>
```

6. Update the other systems with the new wallet and hash value
7. Start the processes

About Encrypting a Password with Wallet Method

- This is a different wallet than the one used for trails and TCP/IP encryption: Credential Store
- Part of the Oracle Credential Store Framework
- The preferred method of password encryption
- It is not supported on the iSeries, z/OS, and NonStop

Creating and Populating the Credential Store

1. Create credential store and add set of credentials to it:

```
ADD CREDENTIALSTORE
ALTER CREDENTIALSTORE { ADD USER userid
  | REPLACE USER userid | DELETE USER userid }
 [PASSWORD password]
 [ALIAS alias]
 [DOMAIN domain]
```

2. Specify the alias in the parameter file (cont..)

Using the Alias in a Parameter File or Command

- Database login in parameter files:

```
USERIDALIAS <alias>
```

- Database login in GGSCI:

```
DBLOGIN USERIDALIAS <alias>
```

- Complete list options in documentation “Administering Oracle GoldenGate for Windows and UNIX 12c”

Oracle GoldenGate Encryption Summary

Encryption Purpose	Method	Implementation
Messages	ENCKEYS	`RmtHost … ENCRYPT …` `server … -Encrypt`
	Wallet	`CREATE WALLET, ADD MASTERKEY` `RMTHOSTOPTIONS`
Data in the Trail or Extract file	ENCKEYS	`ENCRYPTTRAIL and DECRYPTTRAIL`
	Wallet	`ENCRYPTTRAIL`
Database Password	ENCKEYS	`Encrypt Password` `USERID … ENCRYPTKEY`
	Wallet	`ADD CREDENTIALSTORE` `ALTER CREDENTIALSTORE ADD USER`

Summary

In this lecture, you should have learnt how to:

- Use the Encryption Key method and the Wallet method to encrypt the network packets, trail files, and database login credentials