

---

# IBM Planning Analytics

## Encryption at Rest

---

Last Updated:

**June 2019**

Created By:

**Andreas Kugelmeier**

Executive Consultant, FOPM

Planning Analytics Architect

IBM Data and AI Expert Labs

Mobile Phone: +1-215-384-7302

Email: [kugelmeier@us.ibm.com](mailto:kugelmeier@us.ibm.com)

**Document Version History**

Date	Version	Author	Description
08/28/2018	0.9	Andreas Kugelmeier	Draft Version
09/13/2018	1.0	Andreas Kugelmeier	Version 1.0 (corrections and enhancements to documentation, additional examples, additional chapter on migration guidance)
09/13/2018	1.1	Andreas Kugelmeier	Addtl. Information on Master Key extraction/backup/restoration
01/04/2019	1.2	Andreas Kugelmeier	Misc. addtl. information
01/22/2019	1.35	Andreas Kugelmeier	- Add recommendation to save data and delete old transaction logs prior to encryption - Updates to section on backup and restoration of master keys
02/19/2019	1.4	Andreas Kugelmeier	- Addtl. Information on Encryption at Rest on PA Cloud
06/27/2019	1.5	Andreas Kugelmeier	- Add 'Useful Links' section 1.2 - In Section 1.2, add link to DB documentation on keystore management - Remove section 6 (Important Links) (replaced by section 1.2)

**Table of Contents**

**1. Introduction to Encryption at Rest ..... 4**

1.1 TM1 Database Encryption at Rest: FAQ ..... 5

1.2 Useful Links ..... 5

**2. Encryption Keys ..... 6**

2.1 Encryption key management system ..... 6

2.2 The Data Encryption Key (DEK) ..... 6

2.3 The Master Key (MK)..... 7

**3. Encryption and Decryption..... 8**

3.1 Encryption and Decryption Methods ..... 8

3.2 The TM1Crypt Utility ..... 8

    3.2.1 Location..... 8

    3.2.2 Syntax..... 8

    3.2.3 Examples..... 11

**4. TM1 database object migrations ..... 13**

4.1 API-based migrations ..... 13

4.2 Metadata-based migrations..... 13

4.3 Physically migrating encrypted files between environments ..... 13

**5. Key Backup ..... 14**

5.1 DEK Backup ..... 14

5.2 MK Backup ..... 14

## Notices & Disclaimers

Copyright © 2018 by International Business Machines Corporation (IBM). No part of this document may be reproduced or transmitted in any form without written permission from IBM.

### **U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.**

Information in these presentations and papers (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. THIS document is distributed "AS IS" without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity. IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

### **Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.**

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular purpose.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

IBM, the IBM logo, ibm.com, Aspera®, Bluemix, Blueworks Live, CICS, Clearcase, Cognos®, DOORS®, Emptoris®, Enterprise Document Management System™, FASP®, FileNet®, Global Business Services®, Global Technology Services®, IBM ExperienceOne™, IBM SmartCloud®, IBM Social Business®, Information on Demand, ILOG, Maximo®, MQIntegrator®, MQSeries®, Netcool®, OMEGAMON, OpenPower, PureAnalytics™, PureApplication®, pureCluster™, PureCoverage®, PureData®, PureExperience®, PureFlex®, pureQuery®, pureScale®, PureSystems®, QRadar®, Rational®, Rhapsody®, Smarter Commerce®, SoDA, SPSS, Sterling Commerce®, StoredIQ, Tealeaf®, Tivoli®, Trusteer®, Unica®, urban{code}®, Watson, WebSphere®, Worklight®, X-Force® and System z® Z/OS, are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at: [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

## 1. Introduction to Encryption at Rest

In its default state (and as per a common default practices for database objects), the data objects in a TM1 database data directory and the TM1 database log files are not encrypted.

In the unencrypted state, the restoration of a TM1 database is possible just based on data directory files and - where applicable - transaction logs. Additionally, unencrypted TM1 objects can be read using a text editor.

*Encryption at Rest* provides the capability to encrypt TM1 objects (database files) such that

- The TM1 database content cannot be read 'at rest' (without logging onto the database)
- An encrypted TM1 database can only be started up if the TM1 Server has access to the encryption keys used to encrypt the database (i.e. one needs the encryption keys to restore / start up a database)

Note:

- The TM1-specific Encryption at Rest feature discussed in this paper is not supported for TM1 servers that are using 'replication and sync'.
- The TM1-specific Encryption at Rest feature discussed in this paper is not supported for IBM Planning Analytics on Cloud. IBM Planning Analytics Cloud uses a different technology to implement Encryption at Rest: IBM Planning Analytics Cloud is using IBM Cloud Block Storage to implement Encryption at Rest at the storage device level<sup>1</sup>. Encryption at Rest is applied
  - for the shared folder (where the TM1 database directory and transaction log resides), and
  - for all backups are encrypted at rest.<sup>2</sup>Furthermore, the content store for Planning Analytics Workspace and the content store for Cognos Analytics reporting both leverage database level encryption at rest.

---

<sup>1</sup> <https://console.bluemix.net/docs/infrastructure/FileStorage/index.html#getting-started-with-file-storage>

<sup>2</sup> Because IBM Planning Analytics Cloud leverages Encryption at Rest at the file storage level, the implementation of Encryption at Rest on IBM Planning Analytics Cloud is transparent to PA Cloud processes and application. It follows that encryption/decryption is managed implicitly in the PA Cloud, i.e. customers do not need to (nor can) manage encryption/decryption themselves.

## 1.1 TM1 Database Encryption at Rest: FAQ

Q: Is the data encrypted in memory?

A: No. While TM1 is an in-memory database, the TM1 data 'at rest' resides in the TM1 Database (file) objects. The database files will be encrypted. Upon loading a database objects into memory (and upon startup of the database), the file contents are decrypted into memory.

Q: When the database starts up, will the files/objects be decrypted?

A: No. The files stay encrypted. Their content is decrypted by the TM1 server upon access, and the decrypted content is loaded into memory.

Q: What happens when I create new objects / save data to disk?

A: The files are being (re)encrypted by TM1 as TM1 saves the object(s) to disk. Examples: A SaveDataAll will lead to the in-memory data to be encrypted and written to disk. When you create a TI process and save it, the \*.pro file object will be encrypted. But when you 'open' the TI with a TM1 client, the TI will appear in its unencrypted state.

Q: So since the files are not decrypted upon startup, can I start an encrypted database?

A: Yes. That is the behavior with encryption at Rest. Note however that the TM1 server needs the encryption keys to start up an encrypted database. Without the keys, an encrypted database cannot be started. => An encrypted database without both encryption keys is not usable

Q: Do I need to perform extra steps to stop/start an encrypted TM1 Database?

A: No, decryption to memory and (re-)encryption to disk occur automatically by TM1 upon loading/saving of TM1 objects.

Q: How does this affect Backups?

A: you need to both backup the database encryption key and the master key associated with the database. See <Key Backup> for details.

Q: How does this affect Migrations?

A: file-based migrations require decryption and re-encryption. API-based migrations do not require any additional steps. See <TM1 database object migrations> for details.

## 1.2 Useful Links

Planning Analytics documentation on:

- [TM1 Server data encryption at rest](#)
- [Run the TM1Crypt Utility](#)

General information on GSKit and keystore management, from the DB2 documentation:

- [A primer for managing PKCS12 keystores](#)

Importing and exporting keys (from the IBM Http Server documentation):

- [Importing and Exporting keys from/into the IBM GSKit using the command line](#)

## 2. Encryption Keys

### 2.1 Encryption key management system

TM1 server uses a two-tier key management system to encrypt/decrypt server data, where the first tier includes a Data Encryption Key (DEK) to encrypt data, and where the second tier uses a Master Key (MK) to encrypt the DEK. The encrypted DEK is hence not sufficient to decrypt a TM1 database. Decryption occurs by

- 1) Unencrypting the DEK (at decryption runtime) by using the MK
- 2) Using the unencrypted DEK to decrypt the TM1 Database objects

Note: you must back up BOTH your master key and data encryption key as part of your regular TM1 backup and restore procedure. If you lose your master key, you cannot restore the master key and will be unable to access the encrypted TM1 data.

### 2.2 The Data Encryption Key (DEK)

On Encryption, the DEK is generated and stored on-disk in a directory (within the TM1 database directory) called **}key**:

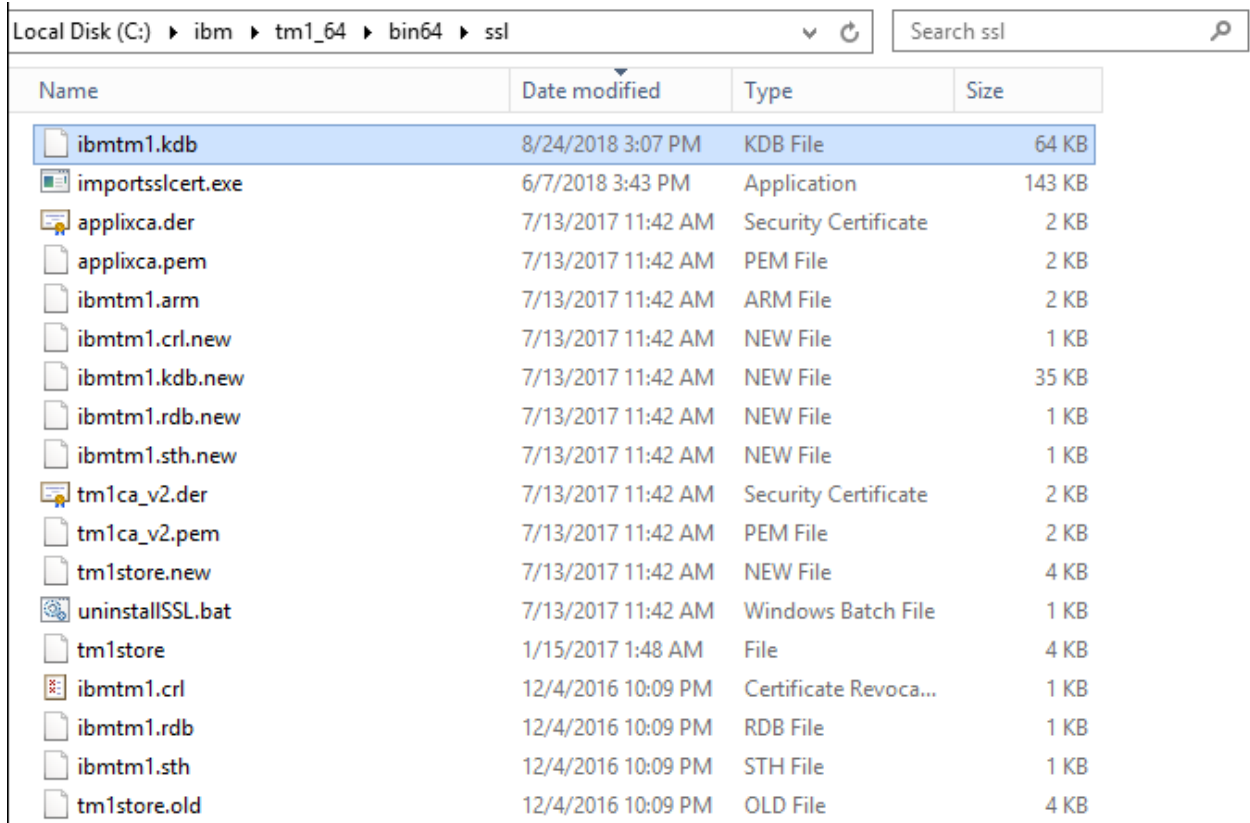
}Dimensions}subs	8/24/2018 3:07 PM	File folder
}DimensionSecurity}vues	8/24/2018 3:07 PM	File folder
}Externals	3/1/2018 11:33 AM	File folder
}Groups}subs	8/24/2018 3:07 PM	File folder
}key	8/24/2018 3:07 PM	File folder
}PerfCubes}subs	8/24/2018 3:07 PM	File folder
}Processes}subs	8/24/2018 3:07 PM	File folder
}ProcessSecurity}vues	8/24/2018 3:07 PM	File folder
}StatsByCube}vues	8/24/2018 3:07 PM	File folder

The **}key** directory includes the 'current' encryption key (if the database is encrypted) plus - in **}key\_backup** - the previously used data encryption keys:

TM1Data ▶ }key		Search }key
Name	Date modif...	Type
}key_backup	8/24/2018 ...	File folder
ibm_tm1_dek_v1_projectbasedplanning_20180824190700.dat	8/24/2018 ...	DAT File

### 2.3 The Master Key (MK)

The master key is generated on encryption and then stored in an IBM Global Security Kit (GSKit) store (ibmtm1.kdb). The keystore database (ibmtm1.kdb) can be found in the ssl sub-directory in either the <TM1InstallDirectory>/bin64 or <TM1InstallDirectory>/bin (depending on 64- or 32-bit version):



Name	Date modified	Type	Size
ibmtm1.kdb	8/24/2018 3:07 PM	KDB File	64 KB
importsslcert.exe	6/7/2018 3:43 PM	Application	143 KB
applixca.der	7/13/2017 11:42 AM	Security Certificate	2 KB
applixca.pem	7/13/2017 11:42 AM	PEM File	2 KB
ibmtm1.arm	7/13/2017 11:42 AM	ARM File	2 KB
ibmtm1.crl.new	7/13/2017 11:42 AM	NEW File	1 KB
ibmtm1.kdb.new	7/13/2017 11:42 AM	NEW File	35 KB
ibmtm1.rdb.new	7/13/2017 11:42 AM	NEW File	1 KB
ibmtm1.sth.new	7/13/2017 11:42 AM	NEW File	1 KB
tm1ca_v2.der	7/13/2017 11:42 AM	Security Certificate	2 KB
tm1ca_v2.pem	7/13/2017 11:42 AM	PEM File	2 KB
tm1store.new	7/13/2017 11:42 AM	NEW File	4 KB
uninstallSSL.bat	7/13/2017 11:42 AM	Windows Batch File	1 KB
tm1store	1/15/2017 1:48 AM	File	4 KB
ibmtm1.crl	12/4/2016 10:09 PM	Certificate Revoca...	1 KB
ibmtm1.rdb	12/4/2016 10:09 PM	RDB File	1 KB
ibmtm1.sth	12/4/2016 10:09 PM	STH File	1 KB
tm1store.old	12/4/2016 10:09 PM	OLD File	4 KB

The master key can be rotated for added security: When a master key is rotated, the DEK is decrypted by using the previous master key and then encrypted with the new master key. During a rotation, the DEK is backed up in a }key\_backup subdirectory; located in the }key directory. Older master keys are persisted in the keystore in case a model restoration is required later.

### 3. Encryption and Decryption

#### 3.1 Encryption and Decryption Methods

Methods for running encryption and decryption include

- a) The TM1 API
- b) The TM1Crypt utility

In its current release, this document describes how to use the TM1Crypt Utility for encryption and decryption

#### 3.2 The TM1Crypt Utility

##### 3.2.1 Location

The TM1Crypt utility, tm1crypt.exe, is installed in the directory:

PA\_install\_directory\bin (32-bit) or PA\_install\_directory\bin64 (64-bit)

##### 3.2.2 Syntax

Run the TM1Crypt utility from a command prompt with the following syntax:

tm1crypt.exe [<cmd\_parm> <connect\_parm> <password\_parm>]

You can provide parameters with constant values in a configuration file when you run tm1crypt.

##### 3.2.2.1 Command parameters

Parameter	Value	Description
-i	filespec	Name of the file that contains default configuration parameters. Parameters specified in this file are used, unless overridden by parameters provided on the command prompt. If no path is specified, the TM1 Server directory is assumed. If -i is not specified, then other parameters must be specified to provide the process name, TM1 Server, and so on.
-connect	string	This parameter can be used to specify a section in the configuration file that contains parameters used to make server connections, such as user, pwd, or CAMnamespace.
-logpath	string	Enables logging and specifies location of log.
-action	string	1 [default] - Generate encrypted password and key file
		2 - Encrypt server model
		3 - Decrypt server model
		4 - Encrypt file
		5 - Decrypt file
		6 - Rotate server key
-keyfile	string	Name of the file generated containing key. If no keyfile is specified the default is tm1key.dat.



-outfile	string	Name of file generated encrypted password. If no outfile is specified the default is tm1cipher.dat.
-filesrc	string	Source file to perform conversion on (decryption or encryption). Source is replaced with converted data unless file destination is provided for parameter -filedest:
-filedest	string	Destination folder for converted files. If not specified, source files are replaced with converted data!
-filetype	string	1 [default] - TM1 object file (if no file type specified, will decrypt/encrypt data directory, transaction logs, and audit logs)
		2 - Transaction log
		3 - Audit log
-minsbeforeshutdown		Time before performing a TM1 Database shutdown when encrypting or decrypting a server model. If not specified, shutdown occurs immediately. Shutdown has to occur (and will occur automatically) only when encrypting or decrypting an entire model with actions 2 or 3, not when targeting specific files for encryption (migrate in) or decryption (migrate out).
-validate		Validate key file.
-help		Display help documentation including parameters and descriptions.
?		Display a synopsis of command line parameters.

### 3.2.2.2 Connect Parameters

Connect parameters are common across TM1 components and can be defined in their own section of a configuration file to reuse them.

Parameter	Value	Description
-adminhost	string	TM1 admin host
-server	string	TM1 Server name
-user	string	TM1 or Cognos Access Manager (CAM) username, depending on the type of authentication that is used by the TM1 Server.
-securitymode		Security mode used to connect to the TM1 Server. The mode must match the value in the TM1 Server configuration file.
-retryattempts		Number of attempts to connect to the TM1 Server.
-retryinterval		Time in seconds to retry connection to the TM1 Server.
-keystorefile	filespec	The full path of the key database file that contains the trusted certificate authorities.
-keystashfile	filespec	The full path of the file that contains the password that is used to access the key database file.
-FIPSOperationMode	1 2 3	Indicates FIPS mode of operation.
		FIPS_MODE = 1 (default)
		FIPS_APPROVED = 2
		FIPS_NONE = 3
-CAMNamespace	id	The ID of the Cognos Access Manager (CAM) namespace. This parameter is the namespace ID, not the namespace name.

### 3.2.2.3 Password Parameters

Passwords are either prompted for on the command line or supplied by using an encrypted file provided by the passwordfile parameter.

Parameter	Value	Description
-pwd	string	Password for the username given in the -user parameter, in clear text. For greater security, the password can be specified in an encrypted file using the -passwordfile parameter. <b>This parameter is ignored on the command line. You are prompted for the password or the user specified in the connection string or connection config file.</b>
-passwordfile	filespec	Filename of the file containing the encrypted password for the user specified by -user. If no path is specified, the TM1 Server directory will be assumed. When this option is used, you cannot use -pwd.
-passwordkeyfile	filespec	If the passwordfile parameter is given, a key file is also required to decrypt the password. The password file and key file can be created using the TM1Crypt tool.

### 3.2.2.4 TM1Crypt configuration file

```
[tm1crypt]
#connect=ConnectParams
#retryattempts=3
#retryinterval=3

### Actions ###
##1 - OPERATION_CRYPT_PWD
##2 - OPERATION_ENCRYPT_MODEL
##3 - OPERATION_DECRYPT_MODEL
##4 - OPERATION_ENCRYPT_FILE
##5 - OPERATION_DECRYPT_FILE
##6 - OPERATION_ROTATE_KEY
#action=

### File Types
##1 - Object File //default
##2 - Transaction Log
##3 - Audit Log
#filetype=

### Valid path for logs files
#logpath=

### Path to file source and destination
#filesrc=
#filedest=

#adminhost=
#server=
#user=
#camnamespace=
```

### 3.2.3 Examples

#### 3.2.3.1 Encrypting TM1 Model (Database Transaction, Audit & Log Files)

Recommendation: To avoid lengthy encryption of large transaction logs, it is recommended to

- a) run a SaveDataAll, and
- b) remove all 'archived' transaction logs

prior to encrypting an entire model

```
c:\ibm\tm1_64\bin64>tm1crypt -i c:\copypath\tm1crypt.config -action 2
Password: *****
Verify: *****
```

After issuing the above command, both DEK and MK are generated/updated, the TM1 database is encrypted and the TM1 Server process is shut down.

#### 3.2.3.2 Decrypting TM1 Model (Database Transaction, Audit & Log Files)

```
c:\ibm\tm1_64\bin64>tm1crypt -i c:\copypath\tm1crypt.config -action 3
Password: *****
Verify: *****
```

After issuing the above command, the DEK is moved to the }key\_backup folder (sub-directory of }key), decrypted and the TM1 Server process is shut down.

#### 3.2.3.3 Encrypting Transaction Log Files

```
c:\ibm\tm1_64\bin64>tm1crypt -i c:\copypath\tm1crypt.config -action 2 -filetype 2
Password: *****
Verify: *****
```

After issuing the above command, both DEK and MK are generated/updated, the TM1 transaction log files are encrypted and the TM1 Server process is shut down.

Sample content of an encrypted transaction log file:

```
#LOG_FORMAT=1
#LOGID=2
#LOGIV=WBR4qZfBdR2MCwE//ouduA==
", "20180827194545", "20180827194545", "0ZMNB7pUymjjT41n8NTEyQ==", "RxPhNdhKTVyFJONZ2xOvAw==", "jwYy2Gmtb0rCbQhC
TL+NtA==", "h63i7Q7njoN9BSAeJh8c4g==", "tNNmFUxkg3W0sPvUhsSYXMIauI9rkF05NUfs2npymKo=", "UwWUQmMLxWzhA73FyaL
2Bw==", "tPo506ExnQnSbl1Z+1XWDNxsMj0SQY8zrX3o7Vdsvs=", ""
"3", "20180827194647", "20180827194647", "0ZMNB7pUymjjT41n8NTEyQ==", "si3q34XOumIM28jbonPnug==", "IX0Qvk1f4fhtGGYjm
7T6bw==", "PVLO/gojTdK+oYmpp/6xOQ==", "03ebnm0Prr5H2VI9L2sOGn6pr2WDNCLdFchd6IIBzY0=", "Sud+gTzZDhHkIKKnN2tD5
w==", "RnQV4TI2eNGSPI+VGyFA1A==", "/NSc1FXdBAR8vrMHGyJjKg==", "Rf6atiu2oZUS79gzcsMIkoIZ8dz0/PP8OoCk9LcfKg=", "iPY
WIYR4n4YeEJQyWpXchQ==", "Am0HCD8njIRf5IB00hWOHA==", "3mVHtqv3+M3i8f2lvhQarQ==", "XsoOulyOwPpT8iB2zYp3bA==", ""
#"3", "20180827194647", "Change set 3 complete : 1"
#"3", "20180827194647", "Change set 3 complete : 1"
```

#### 3.2.3.4 Decrypting Transaction Log Files

```
c:\ibm\tm1_64\bin64>tm1crypt -i c:\copypath\tm1crypt.config -action 3 -filetype 2
Password: *****
Verify: *****
```

### 3.2.3.5 Decrypting specific Files

```
c:\ibm\tm1_64\bin64>tm1crypt -i c:\copypath\tm1crypt.config -filetype 1 -action 5 -filesrc
<TM1DataDirectory>\<TM1Object> -filedest <TargetPathForDecryptedFiles> -logpath
<LoggingDirectory>
```

For example:

```
tm1crypt -i c:\copypath\tm1crypt.config -filetype 1 -action 5 -filesrc
C:\TM1Databases\Profitability\TM1Data\CustomerProfitability.cub -filedest
C:\TM1Databases\Profitability\unencrypted -logpath c:\copypath
Password: *****
Verify: *****
```

Will decrypt the CustomerProfitability cube to a separate directory called 'unencrypted' and log the action in a log file that is placed in the c:\copypath directory

### 3.2.3.6 Encrypting specific Files

```
c:\ibm\tm1_64\bin64>tm1crypt -i c:\copypath\tm1crypt.config -filetype 1 -action 4 -filesrc
<SourcePathForUnencryptedFiles>\<unencryptedTM1Object> -filedest <TM1DataDirectory> -logpath
<LoggingDirectory>
```

For example:

```
tm1crypt -i c:\copypath\tm1crypt.config -filetype 1 -action 4 -filesrc
C:\TM1Databases\Profitability\unencrypted\CustomerProfitability.cub -filedest
C:\TM1Databases\Profitability\TM1Data -logpath c:\copypath
Password: *****
Verify: *****
```

Will encrypt the CustomerProfitability cube and place it into the TM1 data directory and log the action in a log file that is placed in the c:\copypath directory.

## 4. TM1 database object migrations

### 4.1 API-based migrations

Encryption at Rest does not affect API-based migrations, because API-based migrations connect to source and target environments as a TM1 client and then will - in the context of the migration action - create or update the target object in the target system via use of API calls.

### 4.2 Metadata-based migrations

Encryption at Rest does not affect metadata-based migrations, as such migrations are based on executing metadata-based procedures on the target system to (re-)create or update objects as needed.

### 4.3 Physically migrating encrypted files between environments

Encryption at Rest does affect scenarios where objects are physically migrated between environments.<sup>3</sup> Because encryption and decryption occur via two keys which generally are specific to (a) the TM1 Server Installation (such as the PROD environment vs. the DEV environment for example) and (b) the specific TM1 Database (different databases in each environment), encrypted objects cannot be shared (as in migrated) between two environment without (i) decryption in the source environment followed by (ii) encryption into the target environment.

The corresponding process follows the examples for Decrypting specific Files & Encrypting specific Files:

- a) Decrypt migration objects to a migration folder, using the methodology described in Decrypting specific Files
- b) Encrypt the migration files to the target environment, using the methodology described in Encrypting specific Files

Multiple files can be processed by bundling the corresponding TM1Crypt commands into a batch file. For migrating an entire DEV or TST or Staging database to a new target environment, the fastest approach is to decrypt the entire DB (i.e. using -action 3), then copy the directory to the target data directory folder to then encrypt the entire new target database.

---

<sup>3</sup> Note that such physical object migrations are only recommended in certain cases, API-based migration tools as available from IBM and from 3<sup>rd</sup> party solution vendors are generally preferred. See <[IBM Cognos TM1 & Planning Analytics Code Migrations, Separation of Roles, Duties & Procedures: Scenarios, Options & Proven Practices](#)> for more information.

## 5. Key Backup

### 5.1 DEK Backup

Backup the content of the }Key directory (see <The Data Encryption Key (DEK)>)

### 5.2 MK Backup

Option 1: **Backup the entire GSKit database = Backup ibmtm1.kdb** (see <The Master Key (MK)>)

Option 2: Export the MK from the GSKit database:

**Before working with / testing Master Key backup procedures, please back-up the ibmtm1.kdb! (Option 1)**

- i) On the TM1 Server, open a command prompt **as an administrator** and change the directory to the <TM1InstallDirectory>/bin64 for 64-bit or <TM1InstallDirectory>/bin for 32bit installations
- ii) Now run the GSKit command line tool (see [Managing Certificates with IBM GSKit](#)) using the following syntax:

```
gsk8capicmd_64 -cert -list -db ./ssl/ibmtm1.kdb -stashed
```

You should see an output like:

```
c:\ibm\tm1_64\bin64>gsk8capicmd_64 -cert -list -db ./ssl/ibmtm1.kdb -stashed
Certificates found
* default, - personal, ! trusted, # secret key
!   tm1ca_v2
!   applixca
*-  ibmtm1_server
-   tm1svr_v2
-   tm1adminsvr_v2
-   tm1svr
-   tm1adminsvr
#   ibm_tm1_mk_v1_projectbasedplanning_20180824183803
#   ibm_tm1_mk_v1_projectbasedplanning_20180824184217
#   ibm_tm1_mk_v1_projectbasedplanning_20180824184554
#   ibm_tm1_mk_v1_projectbasedplanning_20180824185330
#   ibm_tm1_mk_v1_projectbasedplanning_20180824185639
#   ibm_tm1_mk_v1_projectbasedplanning_20180824190700
```

with

```
ibm_tm1_mk_v1_<TM1Database>_<DateTimeStamp>
```

= the keys generated for your tm1 database(s). In the above example, keys were generated for just one database, the database with the name 'ProjectBasedPlanning'.

- iii) Export the desired key using the following syntax

```
c:\ibm\tm1_64\bin64>gsk8capicmd_64.exe -secretkey -extract -label <KeyName>
-db ./ssl/ibmtm1.kdb -stashed -target ./ssl/<KeyName>_key.backup -format binary
```

(\_key.backup or any other naming suffix and extension to label the extracted file as a key backup)

like

```
c:\ibm\tm1_64\bin64>gsk8capicmd_64.exe -secretkey -extract -label
ibm_tm1_mk_v1_projectbasedplanning_20180824190700
```

```
-db ./ssl/ibmtm1.kdb -stashed -  
target ./ssl/ibm_tm1_mk_v1_projectbasedplanning_20180824190700_key.backup -format binary
```

iv) To add a key (from a backup)

```
c:\ibm\tm1_64\bin64>gsk8capicmd_64.exe -secretkey -add -label <KeyName> -db ./ssl/ibmtm1.kdb -  
stashed -file ./ssl/<KeyName>_key.backup -format binary
```

like

```
c:\ibm\tm1_64\bin64>gsk8capicmd_64.exe -secretkey -add -label  
ibm_tm1_mk_v1_projectbasedplanning_20180824190700 -db ./ssl/ibmtm1.kdb -stashed -file  
./ssl/ibm_tm1_mk_v1_projectbasedplanning_20180824190700_key.backup -format binary
```

v) To remove a key (for testing only):

```
c:\ibm\tm1_64\bin64>gsk8capicmd_64.exe -cert -delete -label <KeyName> -db ./ssl/ibmtm1.kdb -  
stashed
```

like

```
c:\ibm\tm1_64\bin64>gsk8capicmd_64.exe -cert -delete -label  
ibm_tm1_mk_v1_projectbasedplanning_20180824190700 -db ./ssl/ibmtm1.kdb -stashed
```