Transparent Data Encryption for NON-CDB and Multitenant Architecture

Timo Giese, DOAG Security Day 2019

Agenda

1 About Me

2 About Us

3 Legal Framework

4 Licensing

5 TDE History

6 TDE Architecture

7 Configuration

8 Administration

9 Backup & Recovery

10 Upgrade

11 Conclusion

Agenda

About Me

- Oracle DBA since 2005
- Oracle High Availability Specialist
- Twitter: @mbe7

Agenda

1 About Me

2 About Us

3 Legal Framework

4 Licensing

5 TDE History

6 TDE Architecture

7 Configuration

8 Administration

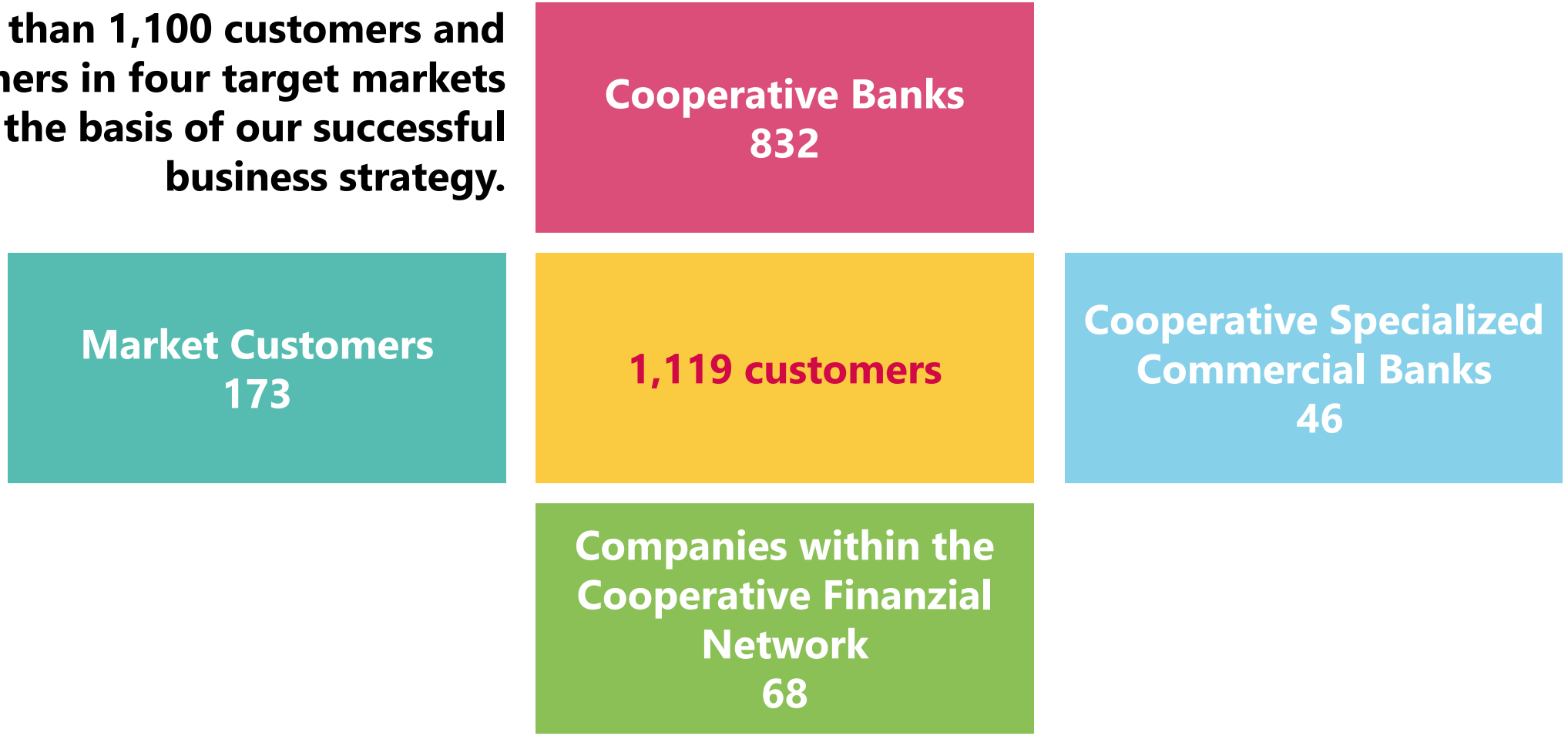
9 Backup & Recovery

10 Upgrade

11 Conclusion

Diverse Target Markets

More than 1,100 customers and owners in four target markets are the basis of our successful business strategy.



Status: 31 Dec. 2018

Focusing on customers in 4 target markets

We provide services to customers in four target markets – Our focus is on our owners in cooperative banking - Volksbanken und Raiffeisenbanken

Cooperative Financial Network*

Cooperative Banks: Volksbanken Raiffeisenbanken

Founding principle

Companies within the Cooperative Financial Network

R+V · DZ BANK Die Initiativbank · eGo Credit · DZ HYP · Schwäbisch Hall · MünchenerHyp · VR Payment · VR Smart Finanz · DZ PRIVATBANK · Union Investment · DG VERLAG

Integration promoters

Cooperative Specialized Commercial Banks

deutsche apotheker- und ärztebank · PSD Bank · EDEKA · KD-BANK Bank für Kirche und Diakonie · BIB FAIR BANKING · Bank für Kirche und Caritas · DARLEHNSKASSE MÜNSTER EG · DKM · Sparda-Bank · Pax-Bank · GLS Bank · Evangelische Bank

One IT in the CFN

Market Customers

IMERKUR IBANK · IBB Internationales Bankhaus Bodensee AG · die Bayerische · NRW.BANK · wüstenrot · SÜDWESTBANK · DONNER & REUSCHEL PRIVATBANK SEIT 1798 · IBANKSCHILLING · ELLWANGER & GEIGER · equensWorldline · Fürst Fugger Privatbank · NIBC · KFW · FLESSABANK · AKTIVBANK AG · MLP · FCA BANK

Focused Market Expansion

* Volksbanken Raiffeisenbanken Cooperative Financial Network (CFN)
(Genossenschaftliche FinanzGruppe Volksbanken Raiffeisenbanken)

Our Locations

Five locations in Germany ensure high security and best customer contact.

- **Karlsruhe and Münster**
  Board, Administration, Product Management, Development, Production, Sales, Customer Service and Printing
- **München**
  Product Management, Development, Sales and Customer Service
- **Berlin**
  Sales and Printing
- **Frankfurt**
  Registered Office and Sales

Agenda

1 About Me
2 About Us
3 Legal Framework
4 Licensing
5 TDE History
6 TDE Architecture
7 Configuration
8 Administration
9 Backup & Recovery
10 Upgrade
11 Conclusion

Legal Framework - GDPR

- Encrypt personal data
- Encrypt communication between all components and servers
- Restrict access to authorized people only

→ Reduce risk of data breach

→ Reduce risk of fines in the future

→ Lost of encrypted storage device is not neccessarily considered as data breach which has to be reported to protection authorities

Agenda

1 About Me
2 About Us
3 Legal Framework
4 Licensing
5 TDE History
6 TDE Architecture
7 Configuration
8 Administration
9 Backup & Recovery
10 Upgrade
11 Conclusion

Licensing

Feature	SE2	EE	EE-ES	DBCS SE	DBCS EE	DBCS EE-HP	DBCS EE-EP	ExaCS	Notes
Colum-Level Encrpytion	N	Y	Y	N	N	Y	Y	Y	EE and EE-ES requires ASO
Tablespace Encryption	N	Y	Y	Y	Y	Y	Y	Y	EE and EE-ES requires ASO
Advanced Security	N	Y	Y	N	N	Y	Y	Y	
Keystore for each PDB	N	N	Y	N	Y	Y	Y	Y	

- 19c Database Licensing Information User Manual

https://docs.oracle.com/en/database/oracle/oracle-database/19/dblic/Licensing-Information.html#GUID-0F9EB85D-4610-4EDF-89C2-4916A0E7AC87

Licensing
What's included in the Advanced Security Option?

- Transparent Data Encryption (TDE) for tablespaces and columns (including Oracle SecureFiles)
- DataPump Export File encryption
- RMAN backup encryption to disk
- TDE master key storage in an Oracle Wallet or external Hardware Security Module
- Data Redaction of sensitive data returned to applications (Full, Partial, Regular Expression, and Random techniques)

- **Network encryption (native network encryption and SSL/TLS) and strong authentication services (Kerberos, PKI, and RADIUS) are no longer part of Oracle Advanced Security and are available in all licensed editions of all supported releases of Oracle Database.**

Agenda

1 About Me
2 About Us
3 Legal Framework
4 Licensing
5 TDE History
6 TDE Architecture
7 Configuration
8 Administration
9 Backup & Recovery
10 Upgrade
11 Conclusion

TDE History I

- belongs to the Advanced Security Option (ASO)
- ASO first introduced with Oracle 8i
- TDE introduced with Oracle 10g R2
  - Table and Column Encryption only !
- **11gR1**
  - Tablespace Encryption possible
  - Usage of Hardware Security Modules (HSM)
- **11gR2**
  - Support for Intel AES-NI (AES-New Instructions)
  - Unified Master Key for Column and Tbs Encryption
  - Reset (rekey) of Unified Master Key possible
  - Unified Master Key can be stored in HSM
- **11gR2 (11.2.0.3)**
  - Hardware Acceleration added for Solaris Sparc
- **12c (12.1.0.1)**
  - Unified Key Management Interface (use of „administer key management for all tasks)
  - Introduction of „syskm“ role
  - Deprecation of PKI usage for TDE
- **12c (12.1.0.2)**
  - Support for Oracle Key Vault

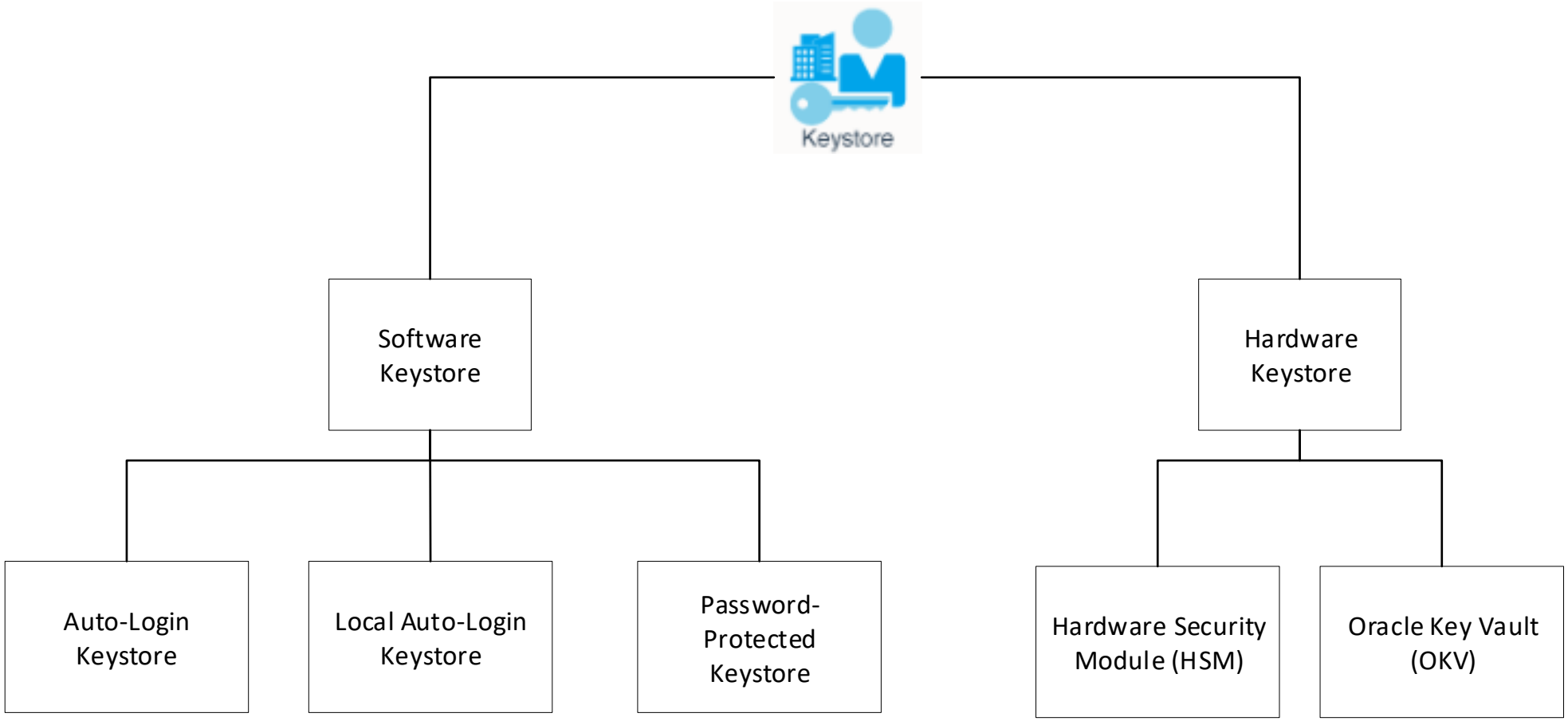
TDE History II

- **12c (12.2.0.1)**
  - Encrypt existing Tablespaces (online and offline)
  - New encryption algorithms (ARIA,GOST,SEED)
  - Ability to force Software Keystore Operations (i.e. rotating keystore pw, backup keystore, …)
  - Ability to use External Store for Software Keystore Passwords
  - Specify Oracle Key Vault as Keystore
  - Encrypt System, Sysaux and Undo-Tbs
- **18c**
  - Ability to create a user-defined Master Encrption Key (outside the database environment)
  - Create Keystore for each PDB
  - Encrypt sensitive data in the data dictionary
- **19c**
  - Transparent Online Conversion Support for Auto-Renaming in Non-OMF File Mode
  - Additional Algorithms for Offline Tablespace encryption (AES192,AES256,ARIA,GOST,3DES)
  - Access to Oracle-Managed Tablespaces even if Keystore is closed

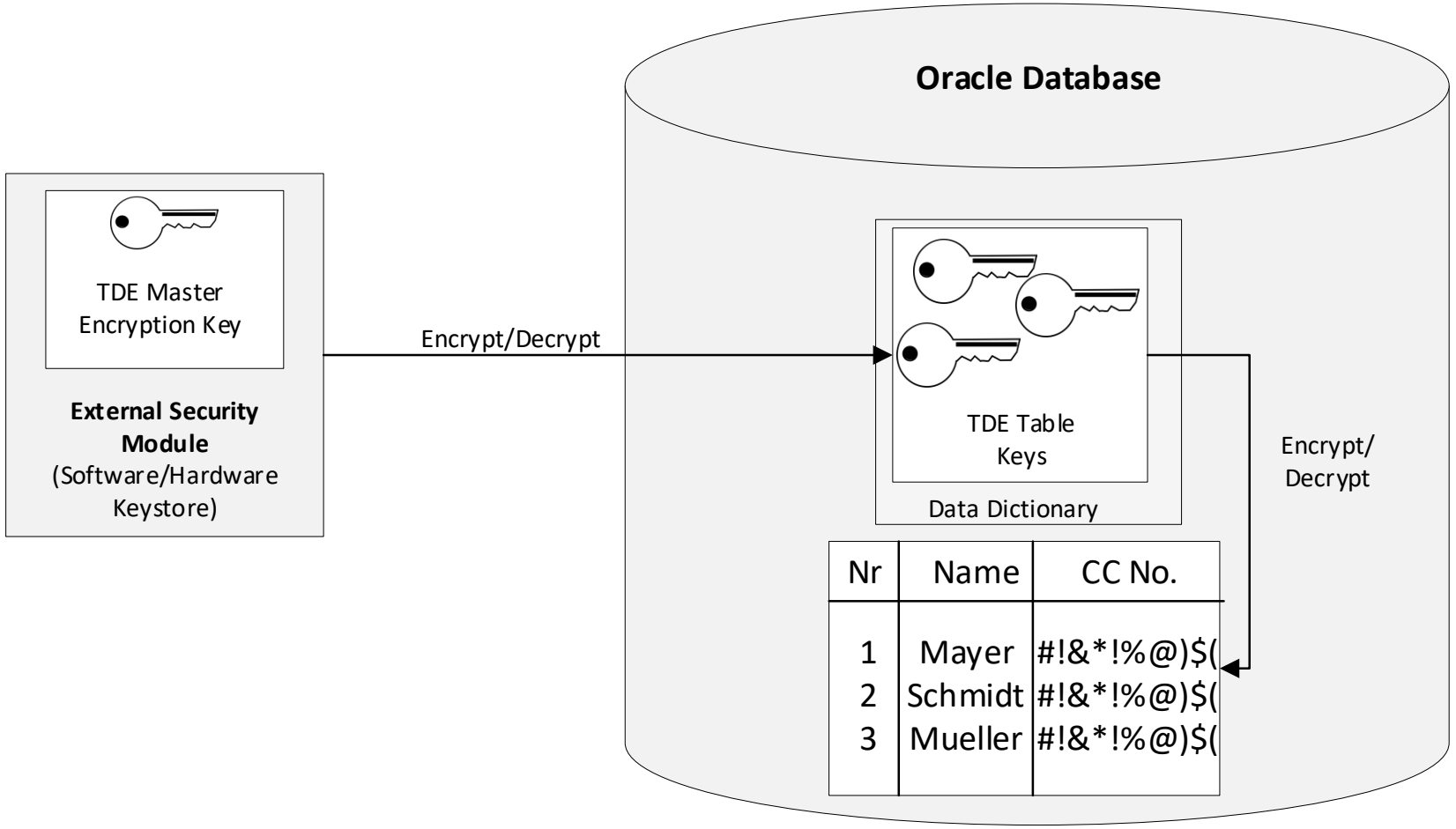
Agenda

1 About Me
2 About Us
3 Legal Framework
4 Licensing
5 TDE History
6 TDE Architecture
7 Configuration
8 Administration
9 Backup & Recovery
10 Upgrade
11 Conclusion

TDE Architecture
Types Of Keystore



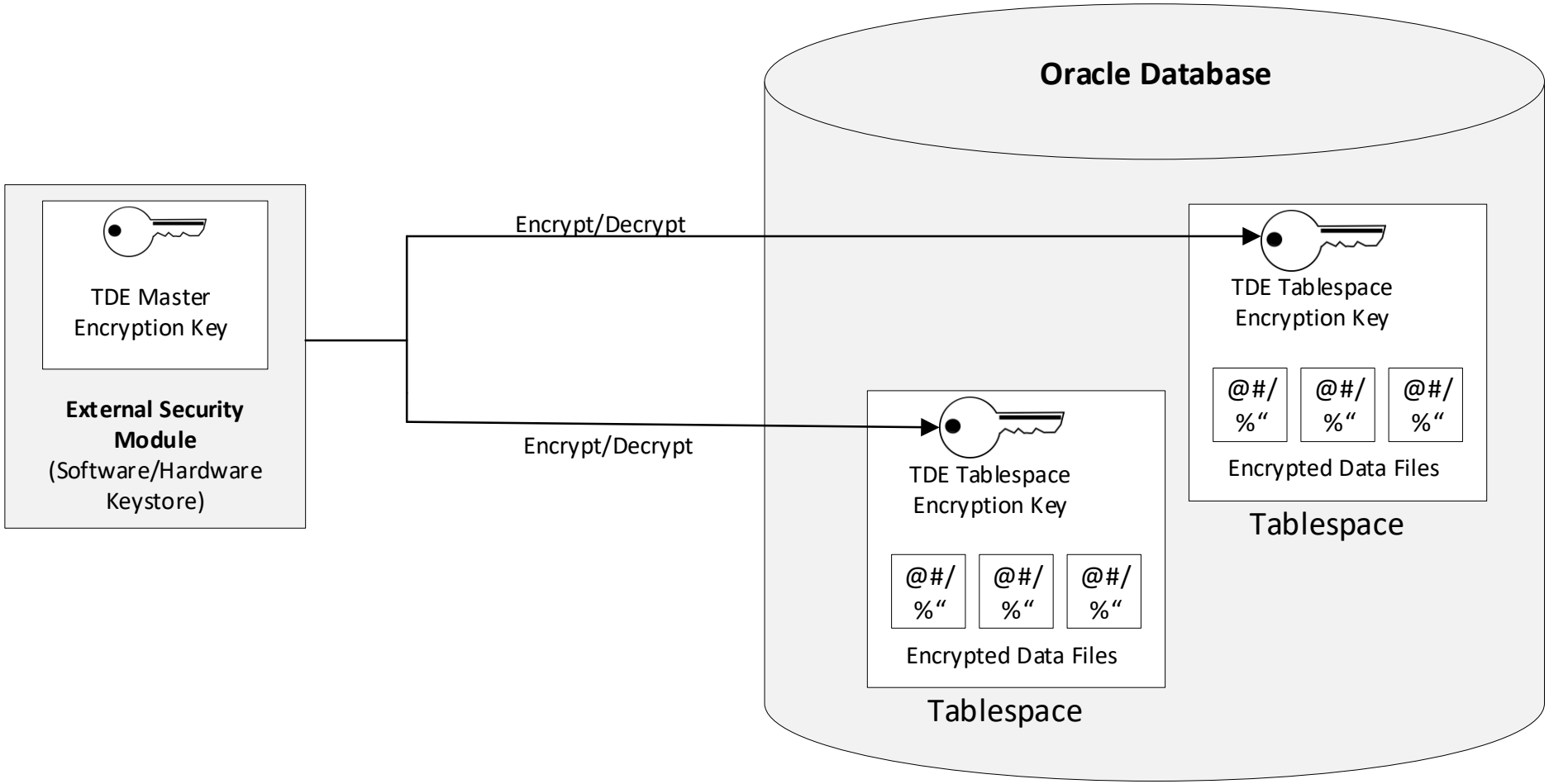
TDE Architecture
Column Encryption I



TDE Architecture
Column Encryption II

- TDE Master Key Stored in external security module
- Usage of a single Table to store all TDE Keys
  - Data Dictionary Table: ENC$
  - Column for keys: COLKLC
- Each TDE Table Encryption Key is encrypted with the TDE master encryption key
- Encryption is transparent to the application

TDE Architecture
Tablespace Encryption I



TDE Architecture
Tablespace Encryption II

- TDE Master Key Stored in external security module
- Each TDE Tablespace Encryption Key is encrypted with the TDE master encryption key
- All Data including Redo and Undo Data is encrypted
- Encrypted Data is protected during JOIN and SORT operations (Data is save if moved to temporary Tablespace)
- Encryption is transparent to the application

Agenda

1 About Me
2 About Us
3 Legal Framework
4 Licensing
5 TDE History
6 TDE Architecture
7 Configuration
8 Administration
9 Backup & Recovery
10 Upgrade
11 Conclusion

Configuration
TDE Encryption Algorithms

Encryption Algorithms (19c)	Key Length	Annotations
Advanced Encryption Standard (AES)	128, 192, 256 bits	128 (Tbs default), 192 (Column default)
Triple Data Encryption Standard (TDES)	168 bits	
ARIA (Korea)	128, 192, 256 bits	>= 12.2
SEED (Korea)	128 bits	>= 12.2
Gost (Russia)	256 bits	>= 12.2

Configuration
TDE Support

Database Technology	Integration
High-Availability	RAC, Dataguard
Backup and Restore	RMAN, OSB
Export and Import	Datapump Export and Import
Database Replication	GoldenGate
Pluggable Database	Multitenant
Engineered Systems	Smart Scan
Storage Management	ASM, ACFS
Data Compression	Standard-, Advanced-, and Hybrid Columnar Compression

Configuration
TDE Column Encryption I – Data Types

Data Type	Maximum Size	Annotations
CHAR	1932 bytes	
VARCHAR2 (legacy)	3932 bytes	
VARCHAR2 (extended)	32,699 bytes	>= 12c
NVARCHAR2 (legacy)	1966 bytes	
NVARCHAR2 (extended)	16,315 bytes	>=12c
NCHAR	966 bytes	
RAW (extended)	32,699 bytes	>=12c

Configuration
TDE Column Encryption II – Data Types

Data Type	Maximum Size	Annotations
BINARY_DOUBLE		
BINARY_FLOAT		
DATE		
INTERVAL DAY TO SECOND		
INTERVAL YEAR TO MONTH		
LOBS		Internal LOBs and SECUREFILE only
NUMBER		
RAW		
TIMESTAMP		Inlcudes WITH (LOCAL) TIME ZONE

Configuration
TDE Column Encryption III – Restrictions

- **Encryption and Decryption done at the SQL Layer**

  - No Index Types other than B-tree
  - No Range scan search through an index
  - No External large objects (BFILE)
  - No Synchronous Change Data Capture
  - No Transportable Tablespaces
  - No usage of traditional exp/imp utilities
  - No usage of encrypted columns in foreign key constraints !!!

Configuration
TDE Tablespace Encryption - Restrictions

- **Encryption and Decryption done while reading and writing the data**

  - No Data Type Restrictions
  - No Index Type Restrictions
  - No usage of traditional exp/imp utilities → use Datapump impdb and expdb
  - If encrypting SYSTEM/SYSAUX, TEMP or UNDO Tablespace
    → NEVER CLOSE KEYSTORE MANUALLY !!!

Configuration
TDE Keystore Location

- Specify the place where to find the Keystore with the Master Encryption Key
- Table also displays precedence where to search the keystore first

Location	Parameter	Version	Annotations
Database Instance	WALLET_ROOT TDE_CONFIGURATION	18c,19c	
sqlnet.ora	ENCRYPTION_WALLET_LOCATION	10g,11g,12.1,12.2,18c,19c	deprecated in 19c
sqlnet.ora	WALLET_LOCATION	10g,11g,12.1	not recommended
ORACLE_BASE/admin/DB_UNIQUE_NAME/wallet	-	10g,11g,12.1,12.2,18c	
ORACLE_HOME/admin/DB_UNIQUE_NAME/wallet	-	10g,11g,12.1,12.2	

Configuration
TDE Keystore Location – sqlnet.ora

- Example for ENCRYPTION_WALLET_LOCATION

```
ENCRYPTION_WALLET_LOCATION=
 (SOURCE=
  (METHOD=FILE)
   (METHOD_DATA=
    (DIRECTORY=/app/oracle/admin/$ORACLE_SID/wallet/)))
```

- Example for WALLET_LOCATION

```
WALLET_LOCATION=
 (SOURCE=
  (METHOD=FILE)
   (METHOD_DATA=
    (DIRECTORY=/app/oracle/admin/$ORACLE_SID/wallet/)))
```

Configuration
Options To Choose In Multitenant Environments I

- **United Mode** (>=18c)
  - One Keystore and TDE Master Key for CDB-Root and all PDBs
  - Same Management for Multitenant as before 18c
- **Isolated Mode** (>=18c)
  - Master Encryption Key for each PDB
  - Keystores (Column and Tablespace) individually per PDB

- Supported for Software-, Hardware- and Oracle Key Vault Keystores
- United and Isolated Mode can be mixed
  - Configure United Mode in CDB-Root
  - Configure Isolated Mode for individual PDBs

Configuration
United Mode I

- **United Mode (CDB-Root)**

  - Set WALLET_ROOT (defines the directory where the TDE Master Key Wallet resides)
  - Restart Database
  - Set TDE_CONFIGURATION parameters (Type of keystore used, i.e. Software or Hardware)
  - Create and open the Keystore
  - Create master encryption key

- **United Moded (PDB)**

  - Open Keystore
  - Set Masterkey for each PDB

Configuration
United Mode II

- Create Wallet Directory for CDB-Root and all PDBs

```
mkdir $ORACLE_BASE/admin/db_unique_name/tde
```

- Set WALLET_ROOT Parameter

```
alter system set WALLET_ROOT="$ORACLE_BASE/admin/db_unique_name" scope=spfile;
```

- Restart Database
- Set TDE_CONFIGURATION Parameter
  - Values: FILE, OKV, HSM

```
alter system set tde_configuration="keystore_configuration=file";
```

Configuration
United Mode III

- Check Configuration

```
show parameter wallet_root
show parameter tde_configuration
select con_id, keystore_mode from v$encryption_wallet;
```

```
    CON_ID KEYSTORE
---------- --------
         1 NONE
         2 UNITED
         3 UNITED
         4 UNITED
         5 UNITED
```

Configuration United Mode IV

- **Setup No-Auto-Login Keystore**
  - Check if everything is set correctly

```
select con_id,wrl_parameter, status from v$encryption_wallet;
```

  - Create Keystore

```
administer key management create keystore [ ,keystore_location‘]
  identified by mySuperSecretPassword;
```

  - Open Keystore in CDB-Root

```
administer key management set keystore open identfied by "mySuperSecretPassword";
```

  - Setup PDB to use Keystore
    - check if PDB is opened Read/Write
    - open keystore in the PDB

```
alter session set container=PDB1;
administer key management set keystore open identfied by "mySuperSecretPassword";
```

  - Create TDE Master Encryption Key
    - Connect to CDB-Root or PDB
    - Create Keys

```
administer key management set key [force keystore]
  identified by 'mySuperSecretPassword'
  with backup using 'my_key_backup';
```

Configuration
United Mode V

- **Setup Auto-Login Keystore**
  - Make keystore an auto-login keystore
  - Open Auto-Login Keystore
  - Check Status

```
administer key management create auto_login keystore from
  keystore  '<keystore_location>' identified by mySuperSecretPassword;

administer key management set keystore open container=all;
```

Configuration NON-CDB

- Configuration is quite equal to „United Mode“
- execute all steps in the NON-CDB Instance

DEMO

Configuration Isolated Mode I

- **in CDB-Root**
  - Set WALLET_ROOT (defines the directory where the TDE Master Key Wallet resides)
  - Restart Database
  - Set TDE_CONFIGURATION parameters
  - Create and open the Keystore
  - Create master encryption key

- **in isolated PDB**
  - Set TDE_CONFIGURATION parameters
  - Create Keystore
  - Open Keystore
  - Create PDB specific Master Encryption Key

Configuration Isolated Mode II

- Create Wallet Directory for CDB-Root

```
mkdir $ORACLE_BASE/admin/db_unique_name/tde
```

- Set WALLET_ROOT Parameter

```
alter system set WALLET_ROOT="$ORACLE_BASE/admin/ db_unique_name" scope=spfile;
```

- Bring Database to „MOUNT“
- Set TDE_CONFIGURATION Parameter in PDB!!!

```
alter system set tde_configuration="keystore_configuration=file;Container=PDBX";
```

```
ERROR at line 1:
ORA-32017: failure in updating SPFILE
ORA-12754: Feature 'Per-PDB TDE keystore' is disabled due to missing capability
'Runtime Environment'.
```

→ CLOUD_ONLY (MOS 2489529.1)

Configuration
Isolated Mode III

- open CDB and PDBs (run in CDB and each PDB)

- Create Keystore

```
administer key management create keystore [ ,keystore_location‘]
  identified by myPDBSecretPassword;
```

- Create Master Key

```
administer key management set key [force keystore]
  identified by 'myPDBSecretPassword'
  with backup using 'my_master_pdb_key_backup';
```

- Make Keystore an auto-login keystore

```
administer key management create auto_login keystore from
  keystore  '<keystore_location>' identified by myPDBSecretPassword;
```

- Open Keystore and verify status

```
administer key management set keystore open identfied by "myPDBSecretPassword";

select status from v$encryption_wallet;

select masterkey_activated from v$database_key_info;
```

Configuration
Table Column Encryption I

- Column Encryption (Default and Non-Default Algorithm)

```
CREATE TABLE employee (
     first_name VARCHAR2(128),
     last_name VARCHAR2(128),
     empID NUMBER,
     salary NUMBER(6) ENCRYPT);
```

```
CREATE TABLE employee (
     first_name VARCHAR2(128),
     last_name VARCHAR2(128),
     empID NUMBER ENCRYPT NO SALT,
     salary NUMBER(6) ENCRYPT USING 'AES256');
```

- Attention with the Datatypes !!!
- Add Encrypted Column to Existing Table

```
ALTER TABLE employee ADD (ssn VARCHAR2(11) ENCRYPT);
```

- Encrypt unencrypted column

```
ALTER TABLE employee MODIFY (first_name ENCRYPT);
```

Configuration
Table Column Encryption II

- Create Index on encrypted column
  - Encrypted Column must be encrypted without „Salt“
    → ERROR: *ORA-28338: cannot encrypt indexed column(s) with salt*

```
CREATE TABLE employee (
first_name VARCHAR2(128),
last_name VARCHAR2(128),
empID NUMBER ENCRYPT NO SALT,
salary NUMBER(6) ENCRYPT);

CREATE INDEX employee_idx on employee (empID);
```

- Remove Salt from encrypted column

```
ALTER TABLE employee MODIFY (first_name ENCRYPT NO SALT);
```

Configuration
Tablespace Encryption I

- Encrypt new and existing Tablespaces (Default and Non-Default Algorithm)
  - Set Parameter COMPATIBLE to minimum 11.2.0.0 (not all Algorithms supported !!!)

```
CREATE TABLESPACE encrypt_ts_1
 DATAFILE '/home/user/oradata/encrypt_ts_1.dbf'
 SIZE 150M
 ENCRYPTION ENCRYPT;
```

```
CREATE TABLESPACE encrypt_ts_2
  DATAFILE '$ORACLE_HOME/dbs/encrypt_ts_2.dbf' SIZE 1M
  ENCRYPTION USING 'AES256' ENCRYPT;
```

- Encrypt Future Tablespaces automatically (>= 12.2)
  - Values: CLOUD_ONLY (default), ALWAYS, DDL

```
ALTER SYSTEM SET ENCRYPT_NEW_TABLESPACES = ALWAYS;
```

Configuration
Tablespace Encryption II

- Encrypt/Decrypt existing Tablespace
  - Online Encryption (>= 12.2)

```
ALTER TABLESPACE users ENCRYPTION ONLINE [ USING 'AES192' ] ENCRYPT [ FILE_NAME_CONVERT = ('users.dbf', 'users_enc.dbf') ] ;
```

  - Online Decryption (>= 12.2)

```
ALTER TABLESPACE users ENCRYPTION ONLINE DECRYPT [ FILE_NAME_CONVERT = ('users_enc.dbf', 'users.dbf') ] ;
```

  - Offline Encryption

```
ALTER TABLESPACE users OFFLINE NORMAL;

Backup Tablespace !!!

ALTER TABLESPACE users ENCRYPTION OFFLINE ENCRYPT;

ALTER TABLESPACE users ONLINE;
```

Offline Decryption

```
ALTER TABLESPACE users OFFLINE NORMAL;

ALTER TABLESPACE users ENCRYPTION OFFLINE DECRYPT;

ALTER TABLESPACE users ONLINE;
```

Configuration
Tablespace Encryption III

- **Encrypt Oracle Maintained Tablespaces (SYSTEM, SYSAUX, UNDO)**
  - Offline Encryption (>= 12.2)

```
STARTUP MOUNT

ADMINISTER KEY MANAGEMENT SET KEYSTORE OPEN IDENTIFIED BY keystore_password;

ALTER TABLESPACE SYSTEM ENCRYPTION OFFLINE ENCRYPT;

ALTER DATABASE OPEN READ WRITE;
```

- **Encrypt Temporary Tablespaces**
  - Drop Temporary Tablespace
  - Create new encrypted Temporary Tablespace (only default algorithm supported !!!)

DEMO

Agenda

1 About Me
2 About Us
3 Legal Framework
4 Licensing
5 TDE History
6 TDE Architecture
7 Configuration
8 Administration
9 Backup & Recovery
10 Upgrade
11 Conclusion

Administration Views

View	Description
ALL_ENCRYPTED_COLUMNS	Encryption information for tables accessible by current user
DBA_ENCRYPTED_COLUMNS	Encryption information for all encrypted columns
USER_ENCRYPTED_COLUMNS	Encryption information for current user schema table columns
DBA_TABLESPACE_USAGE_METRICS	Tablespace usage metrics
V$CLIENT_SECRETS	Properties of the keys stored in the keystore
V$DATABASE_KEY_INFO	Shows default encryption keyd and key activestate for the database
V$ENCRYPTED_TABLESPACES	Displays encrypted Tablespaces
V$ENCRYPTION_KEYS	Information about Master Encryption Keys
V$ENCRYPTION_WALLET	Status and Location of the Keystore
V$WALLET	Metadata for a PKI Certificate

Administration
Managing Keystore I

- **Change Password Protected Software Keystore Password**
  - Online operation
  - no impact to encrypt/decrypt TDE operations while changing password
  - Needs „Administer Key Management“ or „SYSKM“ Privilege
  - Keystore must be open

```
ADMINISTER KEY MANAGEMENT ALTER KEYSTORE PASSWORD
  [FORCE KEYSTORE]
  IDENTIFIED BY
  old_password SET new_password
  [WITH BACKUP [USING 'backup_identifier']];
```

- **Backup Keystore**

```
ADMINISTER KEY MANAGEMENT BACKUP KEYSTORE USING 'Backup_20190315' FORCE KEYSTORE IDENTIFIED BY keystore_password;
```

Administration
Managing Keystore II

- **Change Software Keystore Location**
  - Backup Keystore
  - Close Keystore
  - update sqlnet.ora or WALLET_ROOT in the Database Instance
  - move Keystore with file system commands to new location

```
ADMINISTER KEY MANAGEMENT BACKUP KEYSTORE
  USING 'hr.emp_keystore'
  FORCE KEYSTORE
  IDENTIFIED BY
  software_keystore_password TO '/etc/ORACLE/KEYSTORE/DB1/';
```

- Keystore can also be moved from ASM to Filesystem

Administration
Managing Keystore III

- **Merging Keystores into new one**
  - Needs „Administer Key Management“ or „SYSKM“ Privilege

```
ADMINISTER KEY MANAGEMENT MERGE KEYSTORE 'keystore1_location'
  [IDENTIFIED BY software_keystore1_password] AND KEYSTORE 'keystore2_location'
  [IDENTIFIED BY software_keystore2_password]
  INTO NEW KEYSTORE 'keystore3_location'
  IDENTIFIED BY software_keystore3_password;
```

- **Merging Keystore into existing one**

```
ADMINISTER KEY MANAGEMENT MERGE KEYSTORE 'keystore1_location'
  [IDENTIFIED BY software_keystore1_password]
  INTO EXISTING KEYSTORE 'keystore2_location'
  IDENTIFIED BY software_keystore2_password
  [WITH BACKUP [USING 'backup_identifier]];
```

Administration
Managing Master Encryption Key I

- **Change Master encryption key**

  - per default system generated random value
  - can be changed (REKEY)
  - Master Key is the same for Column and Tablespace encryption
  - Not possible for Auto-Login Keystore (as they don't have a password) !!!
  - BUT: If there is an Auto-Login Keystore and a Password Keystore, REKEY operation is possible

```
ADMINISTER KEY MANAGEMENT SET [ENCRYPTION] KEY
  [FORCE KEYSTORE]
  IDENTIFIED BY [EXTERNAL STORE | keystore_password]
  [WITH BACKUP [USING 'backup_identifier']];
```

Administration
Managing Master Encryption Key II

- **Export Master Encryption Key**

  - Key

```
ADMINISTER KEY MANAGEMENT EXPORT ENCRYPTION KEYS
  WITH SECRET "export_secret"
  TO 'file_path'
  [FORCE KEYSTORE]
  IDENTIFIED BY [EXTERNAL STORE | keystore_password]
  [WITH IDENTIFIER IN 'key_id1', 'key_id2', 'key_idn' | (SQL_query)];
```

- **Import Master Encryption Key**

```
ADMINISTER KEY MANAGEMENT IMPORT KEYS
  WITH SECRET "import_secret"
  FROM 'file_name'
  [FORCE KEYSTORE]
  IDENTIFIED BY [EXTERNAL STORE | keystore_password]
  [WITH BACKUP [USING 'backup_identifier']];
```

Administration
Table Column Encryption

- **Change Encryption Key**

```
ALTER TABLE employee REKEY;
```

- **Change Encryption Key and Encryption Algorithm**

```
ALTER TABLE employee REKEY USING '3DES168';
```

Administration
Tablespace Encryption

- **Change Encryption Key**

  – COMPATIBLE >= 12.2.0.0
  – Database open in READ/WRITE
  – Master Encryption Key is open
  – KEY_VERSION Column of Tablespace in View V$ENCRYPTED_TABLESPACES is ‚NORMAL‘

```
ALTER TABLESPACE users ENCRYPTION [ USING 'AES192' ] REKEY [ FILE_NAME_CONVERT = ('users.dbf', 'users_enc.dbf') ];
```

Agenda

1 About Me
2 About Us
3 Legal Framework
4 Licensing
5 TDE History
6 TDE Architecture
7 Configuration
8 Administration
9 Backup & Recovery
10 Upgrade
11 Conclusion

Backup

Application Data	Backup with RMAN Compression	Backup with RMAN encryption	Backup with RMAN compression and encryption
Not encrypted	Data compressed	Data encrypted	Data compressed first, then encrypted
Encrypted with TDE column encryption	Data compressed; encrypted columns are treated as if they were not encrypted	Data encrypted; double encryption of encrypted columns	Data compressed first, then encrypted; encrypted columns are treated as if they were not encrypted; dobule encryption of encrypted columns
Encrypted with TDE tablespace encryption	Encrypted tablespaces are decrypted, compressed and reencrypted	Encrypted tablespaces are passed through to the backup unchanged	Encrypted tablespaces are decrpyted; compressed and reencrypted

```
rman target /
  set encryption on;
  backup [as compressed backupset] database;
```

Recovery

- Encrypted Tablespace
  - Keystore must be open (ORA-19913: unable to decrypt backup) !!!

```
rman target /
# sql 'alter system set encryption wallet open identified by mySecretKey';
sql 'administer key management set keystore open identified by mySecretKey';
restore tablespace securetbs;
recover tablespace securetbs;
alter tablespaces securetbs online;
```

- Encrypted Columns in a table

```
rman target /
restore table my_encrypted_table;
recover table my_encrypted_table [until time auxiliary destination '<PATH_TO_AUX>' ];
```

DEMO

Agenda

1 About Me
2 About Us
3 Legal Framework
4 Licensing
5 TDE History
6 TDE Architecture
7 Configuration
8 Administration
9 Backup & Recovery
10 Upgrade
11 Conclusion

Upgrade

- NON-CDB and Single-Tenant/Multitenant (all together)
  - DBUA, dbupgrade and catctl.pl supported
  - If using sqlnet.ora, copy to new ORACLE_HOME
  - Use „real“-Path in sqlnet.ora to Wallet-Location if Upgrading from 11g
  - Wallet has to be opened (0RA-28365: wallet is not open) → Use Auto-Login Wallet for Upgrades
  - When no Auto-Login Wallet is present, open Wallet (in MOUNT) before Upgrade

Upgrade

- „New World“: PDB Upgrade with Unplug/Plug

  - Export Master Key from Source CDB/PDB
  - Unplug PDB
  - Plug-In PDB in new CDB-Instance
  - Create or Import TDE Master Key into new CDB-Keystore
  - Open PDB in Upgrade Mode and run „dbupgrade“-Utility
  - Run postupgrade tasks

Agenda

1 About Me
2 About Us
3 Legal Framework
4 Licensing
5 TDE History
6 TDE Architecture
7 Configuration
8 Administration
9 Backup & Recovery
10 Upgrade
11 Conclusion

Conclusion

- If possible start using TDE with Database Version >= 12.2
  - Nearly single Command „Administer Key Management“ to do all TDE Administration Tasks
  - Most of the Tasks can be performed online
  - Oracle Maintained Tablespaces can also be encrypted
- Backup your Encryption Wallets !!!
- Use separate Wallet for each Database (NON-CDB and Multitenant United Mode)
- Always keep a Backup of your Wallet (old Master Keys are kept in Wallet after Rekey-Operation)
- Take a fresh Backup of the Wallet after each REKEY operation
- When possible use SYSKM-Privilege to separate duties
- Check your Backup & Recovery processes

Links

- ASO-Whitepaper: https://www.oracle.com/a/tech/docs/dbsec/aso/advanced-security-wp-19c.pdf
- TDE FAQ: https://www.oracle.com/database/technologies/faq-tde.html
- TDE 11.2: https://docs.oracle.com/cd/E11882_01/network.112/e40393/asotrans.htm#ASOAG600
- TDE 12.1: https://docs.oracle.com/database/121/ASOAG/asopart1.htm#ASOAG600
- TDE 12.2: https://docs.oracle.com/en/database/oracle/oracle-database/12.2/asoag/asopart1.html
- TDE 18c: https://docs.oracle.com/en/database/oracle/oracle-database/18/asoag/asopart1.html
- TDE 19c: https://docs.oracle.com/en/database/oracle/oracle-database/19/asoag/introduction-to-oracle-advanced-security.html#GUID-81BF34FA-6044-47D4-BF31-12DEF178BA6B
- TDE Upgrade: https://mikedietrichde.com/2018/01/16/database-upgrade-and-tde-things-to-know/
- TDE Upgrae: https://blog.dbi-services.com/12cr2-upgrade-by-remote-clone-with-tde-in-dbaas/

Thank you very much for your attention!

FIDUCIA GAD
ZUKUNFTSERFAHREN