How to Use Oracle* 11g Transparent Data Encryption with Intel® AES-NI

Abstract

This guide outlines a short test case on how to use Intel® AES-NI with Oracle* Transparent Data Encryption (TDE). Intel AES New Instructions (AES-NI) was first introduced in March 2010 and are also on the Intel® Xeon® processor codenamed Westmere-EX. These new instructions provide hardware cryptographic acceleration, making the AES computations in Oracle TDE faster and stronger.
TDE is part of Oracle Advanced Security which is an option that can be purchased with Oracle Database Enterprise Edition (ODBEE). Oracle's TDE capabilities make encrypting sensitive data in application table columns, or application tablespaces seamless as the cryptographic operations are performed by the database kernel. This, and the built-in key management, dramatically lowers the cost and complexity of database encryption. Oracle implemented the Intel AES-NI hardware encryption acceleration into their database software using the Intel Integrated Performance Primitives crypto library. With ODBEE 11.2.0.2, Intel AES-NI is automatically detected and used for decryption by default. TDE with ODBEE 11.2.0.2 supports tablespace encryption using Intel AES-NI. Column wise encryption with Intel AES-NI is currently not supported. The test case outlined measures timing for decryption of a 1 million row database.

1.1 Software Installation

The test case documented in this paper was done using Oracle Enterprise Edition 11g Release 2 with patchset 1 (ODBEE 11.2.0.2). The first step in the test case is to install the required software. Links are provided below to install the database software together with Oracle Enterprise Linux. When ODBEE is purchased, such as from oraclestore.oracle.com, there will be a customer support identification code. Save this code as it is needed to install patches.

1) Install Oracle Enterprise Linux 5U5 (64 bit) from http://edelivery.oracle.com/linux.

2) Install ODBEE patchset 1 (11.2.0.2 from 9/13, complete build with Intel AES-NI hardware acceleration for decryption)
http://www.oracle.com/technetwork/database/enterprise-edition/downloads/index.html.

3) The patch that enables encryption with AES-NI is located at
https://updates.oracle.com/download/10080579.html. The support ID code will be needed from the original purchase in step 2.

1.2 General Flow of a Test Case

The test case follows a similar flow of events as follows:

1. An Oracle Wallet file location is specified. Then a TDE master encryption key is created. This can be done using SQL* Plus command line or Oracle Enterprise Manager. This document does so using the command line.
2. Once the Oracle Wallet is created and the TDE master encryption key is added, it needs to be opened for use against the database.
3. Create the encrypted tablespace and populate with contents. When the encrypted tablespace is created, a tablespace key is automatically created. The TDE master key is used to encrypt the tablespace encryption key, which in turn is used to encrypt and decrypt data in the tablespace.

1.3 Test Case: Encrypt 1M Row Database, Test Encryption, & Time Access

This test case creates a 1 million row encrypted table from the all_objects system table. Then it tests to see the data is indeed encrypted by trying to access the data using the strings command. Finally, the time it takes to count the number of rows while decrypting is measured. To see acceleration benefit of Intel AES-NI, encryption can be timed with the patch (with Intel AES-NI) and without the patch (without Intel AES-NI).

1.3.1 Wallet set up

Before attempting to create an encrypted tablespace, a wallet must be created to hold the encryption key. Rather than using the default database wallet, Oracle recommends a separate wallet for transparent data encryption functionality by specifying the ENCRYPTION_WALLET_LOCATION parameter in the sqlnet.ora file. To accomplish this, please add the following entry into the sqlnet.ora file and make sure the specified directory has been created. The default location for the wallet is ($ORACLE_BASE/admin/$ORACLE_SID/wallet) where $ORACLE_SID is <db_name from init.ora>.

```
 ENCRPTION_WALLET_LOCATION=
(SOURCE=(METHOD=FILE)(METHOD_DATA=
   (DIRECTORY=/u01/app/oracle/admin/DB10G/wallet/)))
```

1.3.2 Test case in sqlplus

1. In Linux window, login into sqlplus

```
>sqlplus / as sysdba
```

2. Set TDE master encryption key

```
SQL> alter system set encryption key identified by "intelpassword"; authenticated by
"intelpassword";
```

3. If after shutting down/starting up the database, you will need to run this command to open the encryption wallet

```
SQL> alter system set encryption wallet open identified by "intelpassword"; alter database
open;
```

4. Set timing on

```
SQL> set timing on;
```

5. Clear up existing tablespace and then create new encrypted tablespace. Tablespace key will be created automatically and multiple data files could exist within a tablespace.

```
SQL>drop tablespace ENC_TS including contents and datafiles;
SQL>create tablespace ENC_TS datafile ‘/u01/app/oracle/admin/orcl/encrypted_ts.dbf’ size 2G
encryption using ‘AES256’ default storage(encrypt);
```

6. Populate the tablespace with data from the internal all_objects system table.

```
SQL>drop table fts_table purge;
SQL>create table fts_table tablespace ENC_TS as select * from all_objects where
rownum<=1000;
```

7. Make a bigger 1 million rows table by performing a Cartesian join.

```
SQL>drop table big_table purge;
SQL>create table big_table tablespace ENC_TS as select * from (select*from fts_table)a, (select
rownum from fts_table)b;
```

8. To make sure the data is in fact encrypted, in a Linux window run the following string command. The command won't be able to read the encrypted data within the encrypted_ts.dbf file because it is encrypted.

```
>cd /u01/app/oracle/admin/orcl
>strings encrypted_ts.dbf | grep Timothy
```

9. Returns number of rows from big_table. Decryption with AES-NI is on by default for the table scan.

```
SQL>select count(*) from big_table;
```

1.3 Summary

The test case shows how an Oracle Database can be encrypted and read (decrypted) using Oracle TDE optimized with Intel AES-NI. Using Oracle Database 11g Enterprise Edition version 11.2.0.2 and simple SQL commands a TDE master key was created. The TDE master key encrypts the tablespace key which in turn encrypts/decrypts the table. It was also shown how to time the decryption of a 1 million row table.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

I**ntel may make changes to specifications and product descriptions at any time, without notice. All products, dates, and figures specified are preliminary based on current expectations, and are subject to change without notice.**

Performance tests and ratings are measured) more efficiently **using specific computer systems and/or components and reflect the approximate performance of Intel products as measured by those tests. Any difference in system hardware or software design or configuration may affect actual performance.** Intel®

Westmere and other code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user

Intel, Xeon and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

***Other names and brands may be claimed as the property of others.**

Copyright ©2010 Intel Corporation.