ФГБОУ ВО «Московский технологический университет»

Институт комплексной безопасности и специального приборостроения

Кафедра КБ-1 «Защита информации»

**Дисциплина:** «Безопасность систем баз данных»

**Отчет по лабораторной работе № 5**

**Тема:** «Шифрование данных»

**Вариант задания № 14**

**Выполнил:**

Студент группы БББО-05-20

Кутьин Захар Сергеевич

**Проверил:**

Войтенков Д.В.

**Учебная цель занятия:** Углубить теоретические знания и выработать практические умения в области настройки безопасности компонента Database Engine с помощью среды управления Management Studio и инструкций языка Transact-SQL.

**Место проведения занятия:** компьютерная аудитория.

**Учебно-материальное обеспечение:**

1. Методическая разработка.
2. Компьютерный класс с ПЭВМ.
3. SQL Server 2022 Developer или Enterprise.

**Теоретические сведения**

Прозрачное шифрование данных (TDE) шифрует файлы данных SQL Server, База данных SQL Azure и Azure Synapse Analytics. Это так называемое шифрование неактивных данных, то есть файлов данных и журналов.

Функция прозрачного шифрования данных выполняет шифрование и дешифрование ввода-вывода в реальном времени для файлов данных и журналов. Шифрование использует ключ шифрования базы данных (DEK). Загрузочная запись базы данных хранит ключ для доступности во время восстановления. DEK является симметричным ключом. Он защищен сертификатом, который хранится в базе данных сервера master, или асимметричным ключом, который защищает модуль расширенного управления ключами.

API защиты данных Windows (DPAPI) находится в корне дерева шифрования, защищает иерархию ключей на уровне компьютера и используется для защиты главного ключа службы (SMK) для экземпляра сервера базы данных. SMK защищает главный ключ базы данных (DMK), который хранится на уровне пользовательской базы данных, а также защищает сертификаты и асимметричные ключи. Эти ключи, в свою очередь, защищают симметричные ключи, которые защищают данные. TDE использует аналогичную иерархию вплоть до сертификата. При использовании TDE DMK и сертификат должны храниться в master базе данных. Новый ключ, используемый только для TDE и называемый ключом шифрования базы данных (DEK), создается и сохраняется в пользовательской базе данных.

На рисунке 1 ниже показана архитектура прозрачного шифрования данных. При использовании прозрачного шифрования данных в базе данных SQL пользователь может настраивать только элементы уровня базы данных (ключ шифрования базы данных и фрагменты ALTER DATABASE).

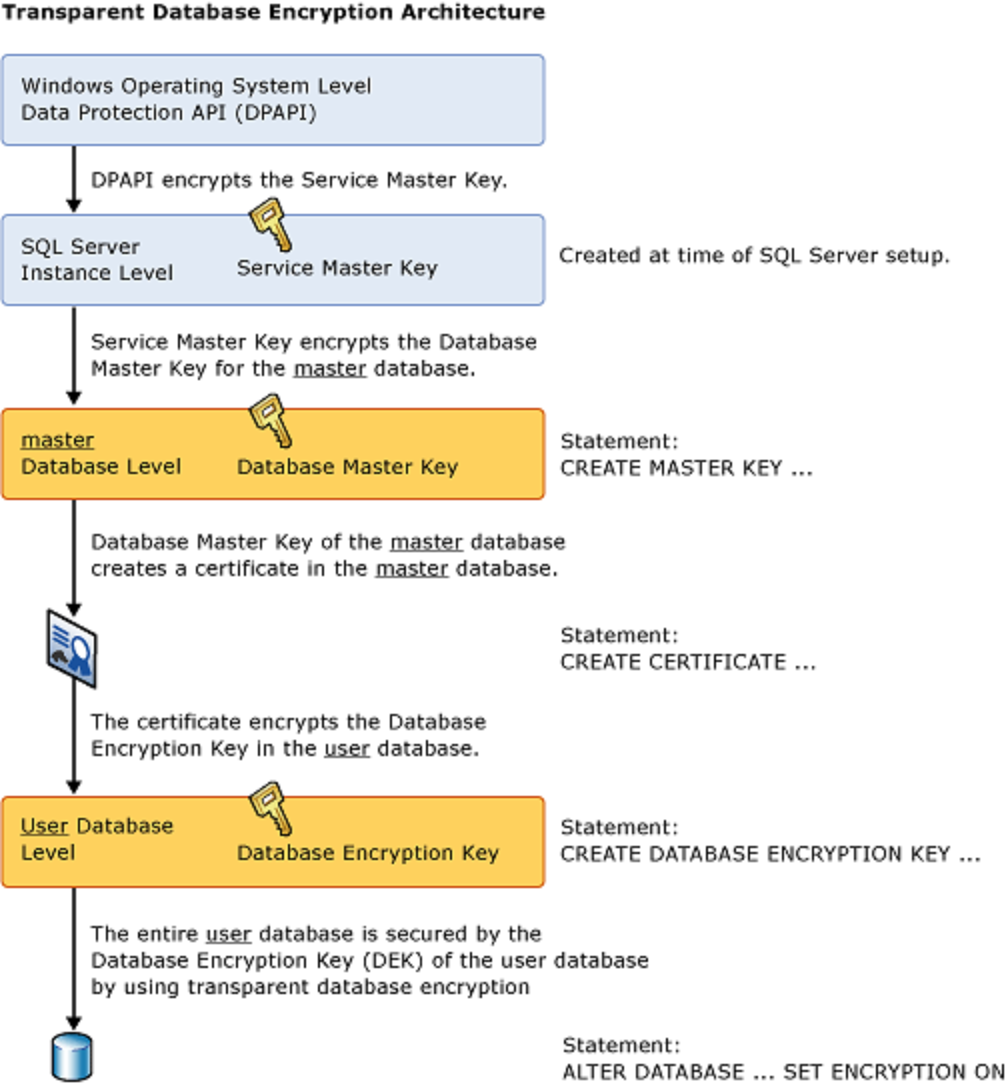


Рисунок 1 – архитектура прозрачного шифрования данных.

**Задание № 1**

Создать главный ключ БД master.

**Ход выполнения задания**

Процесс создания главного ключа БД master с использованием инструкции языка Transact-SQL показан на рисунке 2.

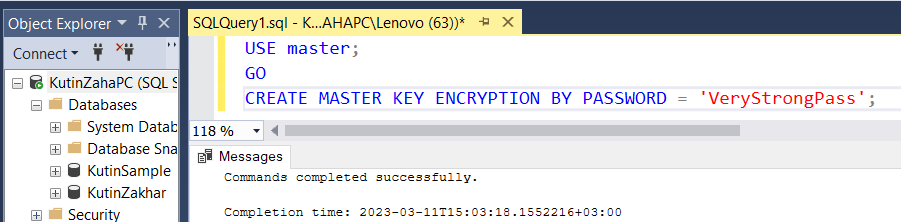


Рисунок 2 – Создание главного ключа БД master.

**Задание № 2**

Создать или импортировать сертификат, закрытый ключ которого должен быть зашифрован главным ключом БД master.

**Ход выполнения задания**

Процесс создания сертификата с использованием инструкции языка Transact-SQL показан на рисунке 3.

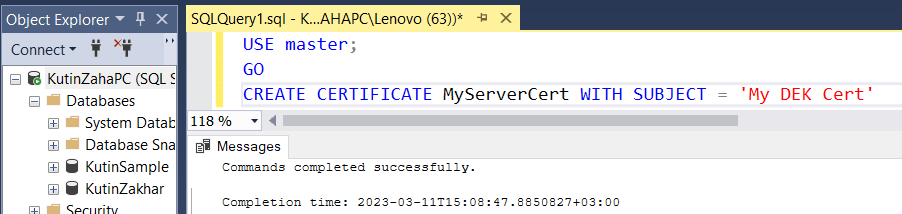


Рисунок 3 – Создание сертификата.

**Задание № 3**

Создать в базе данных, которую мы собираемся шифровать, Database Encryption Key (DEK). DEK шифруется сертификатом, который мы создали на предыдущем шаге.

**Ход выполнения задания**

Процесс создания Database Encryption Key (DEK) с использованием инструкции языка Transact-SQL показан на рисунках 4-5.

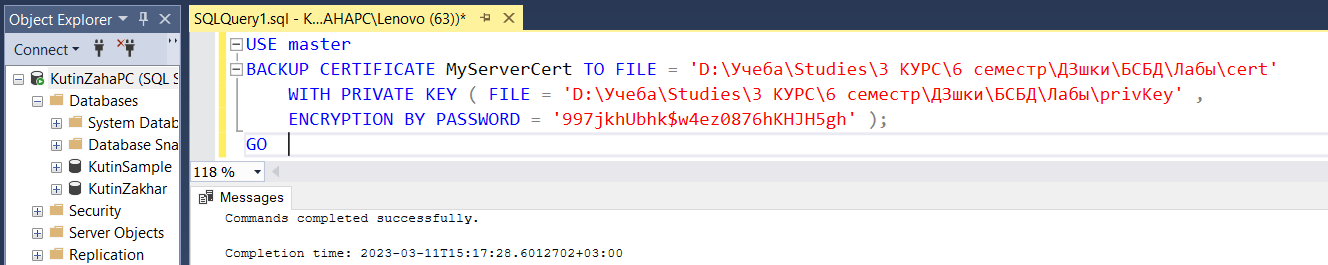
****

Рисунок 4 – Backup с экспортом сертификата и закрытого ключа.

Далее был создан Database Encryption Key, что показано на рисунке 5.

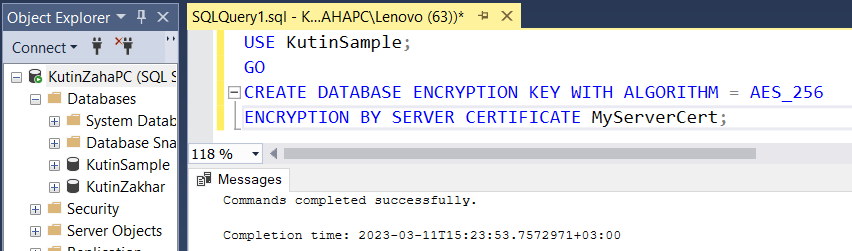


Рисунок 5 – Создание Database Encryption Key.

**Задание № 4**

Проверить, что Database Encryption Key (DEK) действительно создан. Это возможно с помощью системного представления sys.dm\_database\_encryption\_keys.

**Ход выполнения задания**

Проверка, что Database Encryption Key (DEK) действительно создан с использованием инструкций языка Transact-SQL представлена на рисунке 6.

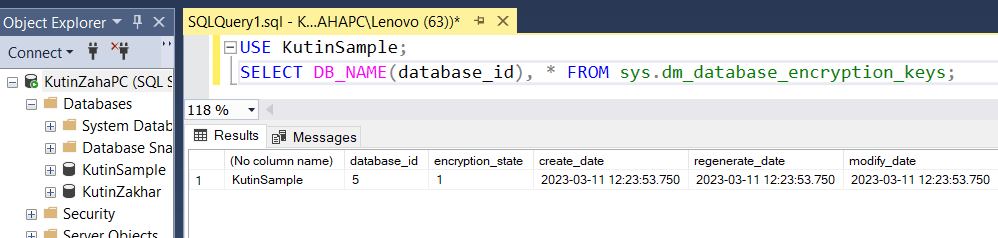


Рисунок 6 – Проверка, что Database Encryption Key (DEK) создан.

**Задание № 5**

Включить шифрование базы данных. Проверить, что процесс шифрования базы данных завершен.

**Ход выполнения задания**

Включение шифрования базы данных с использованием инструкций языка Transact-SQL представлено на рисунке 7.

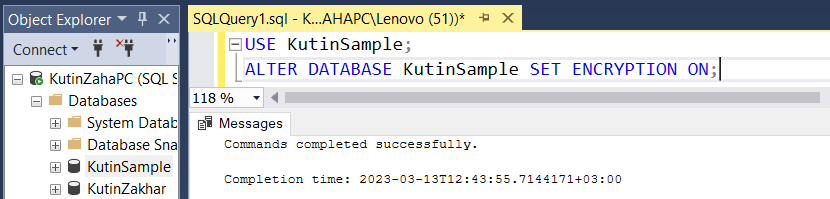


Рисунок 7 – Включение шифрования базы данных

Проверка того, что процесс шифрования базы данных завершен с использованием инструкций языка Transact-SQL представлена на рисунке 8.

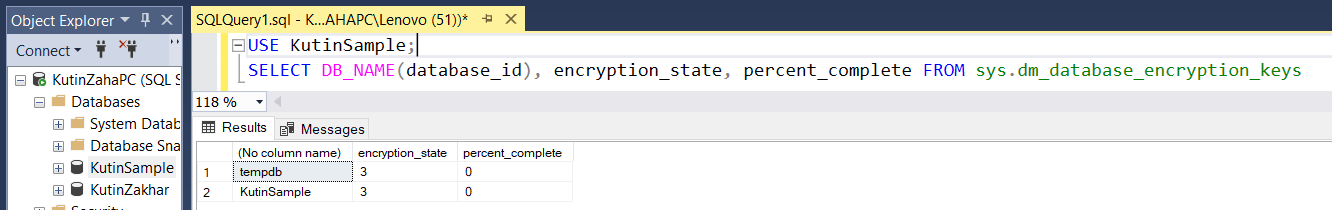


Рисунок 8 – Проверка того, что процесс шифрования базы данных завершен

В столбце encryption\_state содержится информация о текущем состоянии базы данных. В контексте Transparent Data Encryption (TDE) БД может находиться в одном из следующих состояний:

0 - Database Encryption Key (DEK) не создан.

1 - Database Encryption Key (DEK) создан, но база данных не зашифрована.

2 - Выполняется первоначальное шифрование.

3 - База данных зашифрована.

4 - Идет смена ключа.

5 - Идет расшифровка.

**Задание №6**

Выполнить шифрование столбца таблицы с использованием процедуры на языке Transact-SQL. Проверить шифрование столбца таблицы.

**Ход выполнения задания**

Выполнение шифрования столбца emp\_fname таблицы KutinEmployee с использованием процедуры на языке Transact-SQL показано на рисунках 9-11.

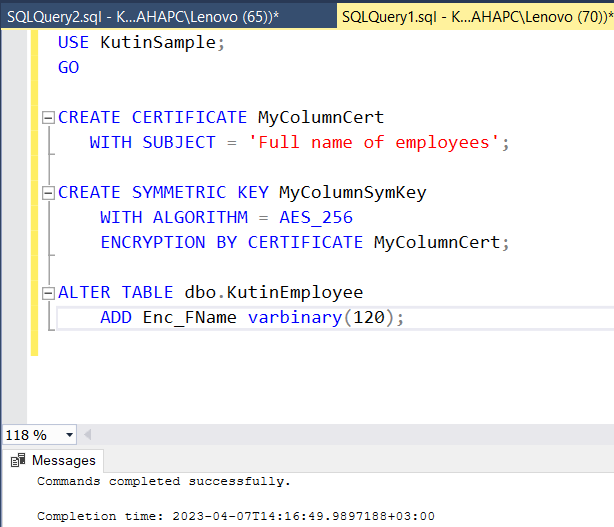


Рисунок 9 - Создание сертификата, симметричного ключа для шифрования и колонки для хранения зашифрованных данных

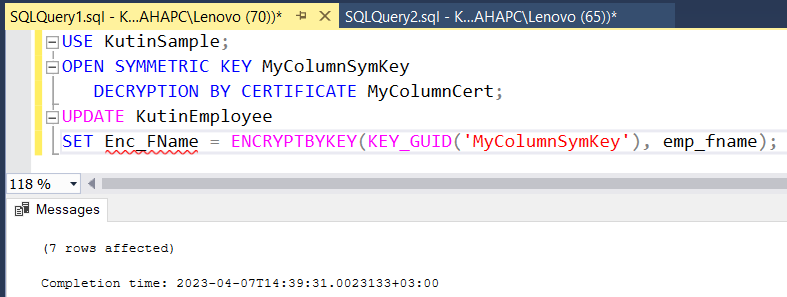


Рисунок 10 - Шифрование значения столбца emp\_fname с помощью симметричного ключа с сохранением результата в столбце Enc\_FName

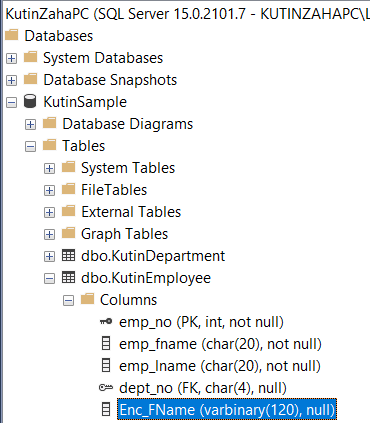


Рисунок 11 - Проверка, что что столбец с зашифрованными данными создан

Проверка шифрования столбца emp\_fname таблицы KutinEmployee с использованием инструкций языка Transact-SQL показана на рисунке 12.

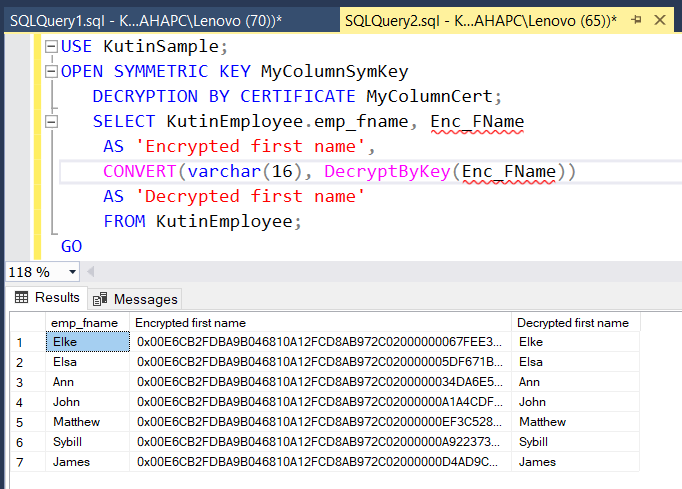


Рисунок 12 - Проверка шифрования столбца emp\_fname таблицы KutinEmployee

**Выводы по проделанной работе**

В результате выполнения лабораторной работы были решены следующие задачи:

- углублены теоретические знания в области настройки безопасности компонента Database Engine;

- освоены приемы настройки безопасности компонента Database Engine с помощью среды управления Management Studio;

- выработаны практические умения настройки безопасности компонента Database Engine с помощью инструкций языка Transact-SQL.