**Санкт-Петербургский политехнический университет Петра Великого**

**Институт кибербезопасности и защиты информации**

**ЛАБОРАТОРНАЯ РАБОТА №5**

**Шифрование в СУБД**

по дисциплине «Основы построения защищенных баз данных»

Выполнил

студент гр. 4841001/00101 *<подпись>* А.С. Петушков

Проверил

преподаватель *<подпись>* М.А. Полтавцева

Санкт-Петербург  
2022

**Цель работы**

Получить навыки обеспечения конфиденциальности баз данных методом шифрования.

**Задачи работы**

Часть 1. *(Теоретическая, в отчете не отражается)*

1. Изучите такие характеристики СУБД, на которой выполняется лабораторная работа, как:
   1. Режимы шифрования и их соответствие уровням шифрования (см. лекционный материал);
   2. Объекты, которые могут быть зашифрованы;
   3. Алгоритмы (типы алгоритмов) применяемых для шифрования данных;
   4. Ограничения режимов шифрования;
2. Рассмотрите возможности по шифрованию данных при передаче.
3. Рассмотрите поддержку шифрования в системном каталоге – хранение метаданных и доступ к ним.

Часть 2. *(Практическая)*

1. Для своего варианта выполнения лабораторных работ определите приватные (требующие шифрования):
   1. По крайней мере, одно отношение;
   2. По крайней мере, 2 столбца разного типа данных в отношении, в котором также будут присутствовать незашифрованные данные.
2. Реализовать шифрование в СУБД (для всех допустимых в бесплатной версии режимах), включая:
   1. Шифрование столбцов таблицы (см.п.1)
   2. Шифрование всей таблицы (см.п.1)
   3. Шифрование файлов данных на диске (и только)
   4. Шифрование данных при передаче.
   5. Сквозное шифрование всех объектов, включая данные в памяти (если возможно).
3. При использовании штатных режимов шифрования, укажите, какие задачи из перечисленных они позволяют решить.
4. Укажите для каждой из задач 2.a – 2.e возможно ли ее решение с разным уровнем конфиденциальности (для пользователей с разным доступом). Реализуйте примеры, где это возможно.
5. Для каждого практического примера оцените изменение производительности и объема данных, проведя для измерения производительности 6-10 тестов на объеме данных, обеспечивающем должный уровень нагрузки (задержку не менее нескольких секунд).

**Ход работы**

СУБД PostgreSQL предоставляет следующие возможности шифрования (представлены в таблице 1).

Таблица 1 – Возможности шифрования

|  |  |  |  |
| --- | --- | --- | --- |
| Режим | Уровень | Объекты | Алгоритмы |
| Шифрование столбцов | СУБД | Столбцы | Хэширование: MD5, SHA1/224/256/384/512, DES  Шифрование: **AES128**/192/256, 3DES, CAST5 |
| Шифрование паролей | СУБД | Пароли | **MD5**, SCRAM |
| Шифрование раздела данных | Хранения | Данные на диске | Определяется на уровне файловой системы |
| Шифрование данных при передаче по сети | Приложения | Данные, передавае-мые по сети | TLS, GSSAPI; также можно применять stunnel и SSH |
| Шифрование на стороне клиента | Приложения | Данные типа bytea | Определяется пользователем |
| Шифрование всей таблицы | Не предоставляется | | |
| Сквозное шифрование всех объектов | Не предоставляется | | |

Для примера определим приватное отношение: это будет

"Documents"

защищаемые столбцы в нем –

"type"

"number"

**Пример 1 – Шифрование столбцов таблицы**

Для шифрования столбцов применяется расширение pgcrypto, включаемое запросом

CREATE EXTENSION pgcrypto;

Для зашифрования данных используется функция pgp\_sym\_encrypt(), для расшифрования – pgp\_sym\_decrypt(). По умолчанию используется алгоритм AES128.

Таким образом, создание тестовой записи, удовлетворяющее установленным критериям защиты, будет выглядеть так:

INSERT INTO "Documents" (

"id"

, "type"

, "number"

, "id\_teachers"

) VALUES (

0

, pgp\_sym\_encrypt('Паспорт', 'passw')

, pgp\_sym\_encrypt('4004 078567', 'passw')

, 1

);

Расшифрование будет выглядеть так:

SELECT

"id"

, pgp\_sym\_decrypt("type"::bytea, 'passw')

, pgp\_sym\_decrypt("number"::bytea, 'passw')

, "id\_teachers"

FROM "Documents";

На рисунке 1 представлено зашифрованное содержимое таблицы после добавления записи.

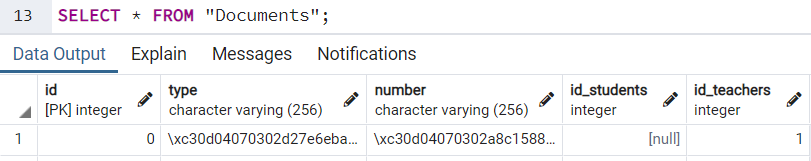


Рисунок 1 – Зашифрованные данные

На рисунке 2 представлен результат расшифрования.

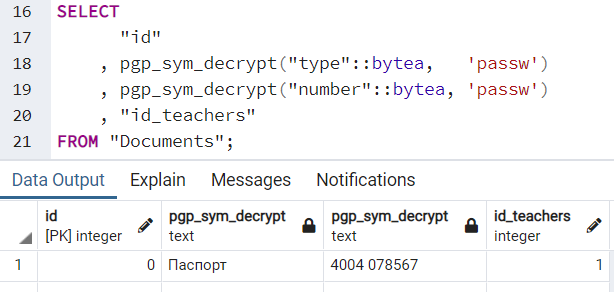


Рисунок 2 – Расшифрованные данные

**Пример 2 – Шифрование данных при передаче по сети**

Для шифрования данных при передаче по сети будет применяться SSH. Для этого не требуется никаких действий с самой СУБД, т.к. это уровень приложения.

Например, при использовании оболочки pgAdmin необходимо настроить подключение по туннелю SSH, как показано на рисунке 3, и ввести пароль при подключении к базе данных.

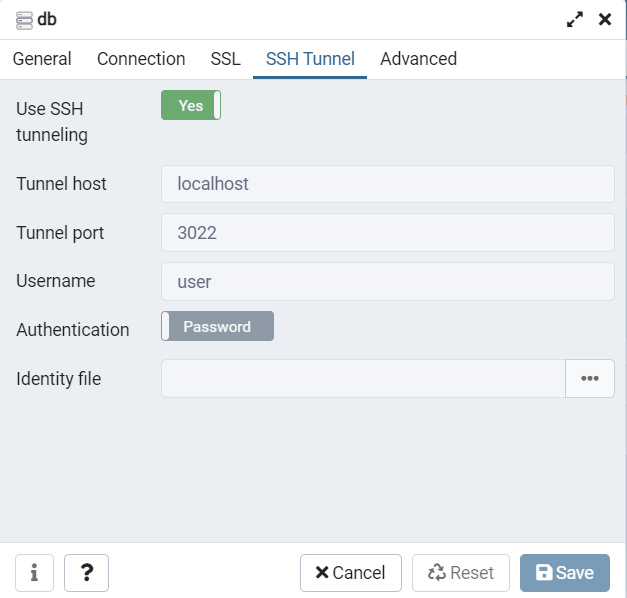


Рисунок 3 – Настройка SSH Tunnel в pgAdmin

При просмотре трафика утилитой Wireshark было выяснено, что открытый трафик не присутствует.

**Пример 3 – Шифрование раздела данных**

Для шифрования данных на диске будет применяться encfs. Для этого не требуется никаких действий с самой СУБД, т.к. это уровень хранения.

Сначала требуется установить encfs

# apt install encfs

Необходимо создать две директории: одна из них будет содержать зашифрованные с помощью encfs данные, а другая будет точкой монтирования, содержать расшифрованные данные, доступные только пользователю, который осуществил расшифрование, введя пароль.

В PostgreSQL директория, содержащая данные БД, называется base/.

Для начала СУБД было остановлено, все данные из директории base/ перенесены во временную копию, затем эта директория была очищена (она будет использоваться как точка монтирования). Также создана директория .base/ для хранения зашифрованных данных.

Далее произведена инициализация encfs. Она происходит при первом запуске операции расшифрования

postgres$ encfs ~/11/main/.base ~/11/main/base

На этом этапе encfs предлагает выбрать конфигурацию (использована конфигурация по умолчанию) и задать пароль.

Теперь расшифрованные файлы из директории .base/ (рисунок 4) отображаются в base/ (рисунок 5) и могут использоваться СУБД.



Рисунок 4 – Содержимое зашифрованной директории



Рисунок 4 – Расшифрованные файлы

После применения данного метода защиты СУБД работает корректно, время работы не поменялось (т.к. во время функционирования файлы расшифрованы), размер содержимого также не изменился.

**Сравнение производительности**

Для подключения к базе данных без шифрования использовался протокол telnet, c шифрованием – ssh. В обоих случаях использовалась утилита psql на удаленном сервере.

Также при тестировании шифрования столбца для подключения использовался протокол telnet.

Создано по 1000 зашифрованных и незашифрованных строк. Производилось по 10 измерений. Время замерялось средствами psql.

Размер базы данных отображался с помощью команды psql

\l+ postgres

Для режима шифрования при передаче данных размер БД не выводился, т.к. в этом нет смысла.

В таблице 2 приведены результаты, полученные при тестировании производительности рассмотренных примеров.

Таблица 2 – Производительность

|  |  |  |
| --- | --- | --- |
| Режим | Среднее время выполнения запроса, мс | Размер базы данных, Кб |
| Без шифрования | 4.672 | 9245 |
| Шифрование столбцов | 6226.693 | 9565 |
| Шифрование при передаче данных | 4.865 | -- |

Видно, что **шифрование столбцов** влечет за собой огромные накладные расходы.

Шифрование с помощью ssh **при передаче данных** не является ресурсозатратным.

**Шифрование раздела данных** с помощью encfs (не отражено в таблице) показало отличный результат, никак не повлияв на работу СУБД.

**Вывод**

В результате выполнения данной лабораторной работы были получены навыки обеспечения конфиденциальности баз данных методом шифрования на примере СУБД PostgreSQL.

Были изучены все возможности шифрования, предоставляемые данной СУБД, а также на практике применено шифрование столбцов, шифрование при передаче данных и шифрование раздела данных. Исследована производительность.