# Зашифрование в базе данных

**PostgreSQL** — это мощная реляционная система управления базами данных с открытым исходным кодом, которая поддерживает множество возможностей для обеспечения безопасности данных. Она может работать как локально, так и в распределенной сети, обеспечивая высокую производительность и надежность.

**Принципы работы:**

* PostgreSQL может работать локально или по сети, поддерживая взаимодействие через клиент-серверную архитектуру.
* Клиенты взаимодействуют с сервером базы данных через стандартные протоколы (например, TCP/IP), отправляя запросы на выполнение SQL-команд.

**Виды шифрования в PostgreSQL**

1. **Шифрование на уровне диска (Full Disk Encryption):** Используется для защиты данных в случае физического доступа к носителям. Реализуется на уровне файловой системы с использованием таких инструментов, как dm-crypt и LUKS на Linux, что позволяет шифровать весь диск или раздел.
2. **Шифрование на уровне сетевого взаимодействия:** PostgreSQL поддерживает шифрование всех данных, передаваемых между клиентом и сервером, с использованием SSL или GSSAPI, что защищает от перехвата данных в процессе передачи.
3. **Шифрование столбцов с использованием pgcrypto:** Позволяет шифровать определенные столбцы таблиц с помощью встроенных функций, таких как pgp\_sym\_encrypt() и pgp\_sym\_decrypt(). Это удобно, когда необходимо защитить только определенные поля, например, номера кредитных карт.
4. **Прозрачное шифрование данных (TDE):** Прозрачное шифрование данных (Transparent Data Encryption, TDE) в PostgreSQL шифрует данные на уровне хранения (данные на диске) и предотвращает несанкционированный доступ в случае кражи или потери носителя. TDE шифрует файлы данных, журналы изменений (WAL) и временные файлы, обеспечивая безопасность на всех этапах хранения данных. Однако в стандартной версии PostgreSQL эта возможность отсутствует и доступна только через сторонние расширения, такие как pg\_tde.

**Реализация прозрачного шифрования**

Для реализации прозрачного шифрования можно использовать расширение pg\_tde, которое позволяет шифровать данные всего кластера или отдельных таблиц. Процесс включает следующие шаги:

1. Настройка параметров в postgresql.conf:

* Определение местоположения хранилища ключей (keystore\_location).
* Указание алгоритма шифрования, например, AES256.

1. Создание мастер-ключа: С помощью функции pgx\_set\_master\_key создается мастер-ключ для шифрования данных.
2. Перезапуск сервера: После настройки ключа и параметров необходимо перезапустить сервер для активации шифрования.
3. Создание и шифрование таблиц: Шифрование можно применять к таблицам и индексам, сохраняя их данные в зашифрованном виде на диске.

**Реализация шифрования с помощью pgcrypto**

**1) Установка расширения:**

CREATE EXTENSION IF NOT EXISTS pgcrypto;

**2) Шифруем столбец Password в таблице Users:**

UPDATE public."Users"

SET "Password" = pgp\_sym\_encrypt("Password", 'your\_secret\_key');

**3) Для вставки новых зашифрованных данных:**

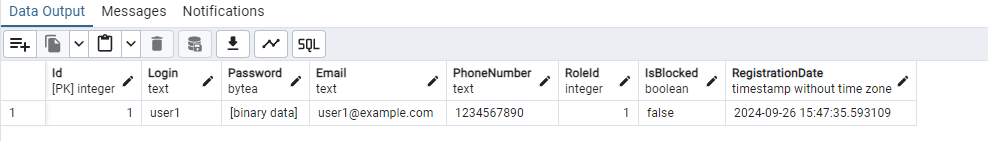
INSERT INTO public."Users" ("Login", "Password", "Email", "PhoneNumber", "RoleId", "IsBlocked", "RegistrationDate")

VALUES ('new\_user', pgp\_sym\_encrypt('user\_password', 'your\_secret\_key'), 'email@example.com', '1234567890', 1, FALSE, CURRENT\_TIMESTAMP);

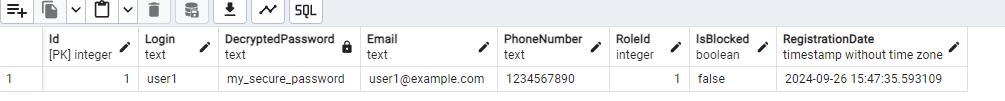
**4) Для расшифровки данных при выборке:**

SELECT "Id", "Login", pgp\_sym\_decrypt("Password", 'your\_secret\_key') AS "DecryptedPassword", "Email", "PhoneNumber", "RoleId", "IsBlocked", "RegistrationDate" FROM public." Users ";

Зашифрованные данные:



Расшифрованные данны:

****

Столбец для шифрования должен быть байтовым.