

ЗАЩИЩЕННАЯ СИСТЕМА УПРАВЛЕНИЯ  
БАЗАМИ ДАННЫХ «ЈАТОВА»

Руководство по настройке. Часть 9.  
Обфускация кода PL/spgSQL.  
Компонент «PLsPgSQL»

643.72410666.00067-07 98 01-09

Листов 38

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## АННОТАЦИЯ

В документе приведены сведения необходимые для установки и эксплуатации компонента «PLsPgSQL» (далее по тексту – «компонент» либо «PLsPgSQL»).

Настоящее руководство предназначено для администраторов систем управления базами данных (СУБД).

Администратор СУБД «Jatoba» должен иметь навыки по работе с СУБД PostgreSQL или защищенной СУБД «Jatoba» (ООО «Газинформсервис»).



Все примеры в данном документе приведены для СУБД «Jatoba» версии ядра 5.x, для других версий все шаги выполняются аналогично, разница состоит в именах директорий.

Например, СУБД «Jatoba» версии 4.x по умолчанию устанавливается в директорию:

- ОС Windows – «C:\Program Files\GIS\Jatoba\4\bin»;
- ОС Linux – «/usr/jatoba-4/bin».

Для СУБД «Jatoba» используется версия компонента — 2.0.1



### **Важная информация**

Для сертифицированной версии СУБД «Jatoba» поддерживается работа только на ОС, указанных в формуляре на поставку!

Степени важности примечаний, применяемые в документе:



**Важная информация** – указания, требующие особого внимания



**Дополнительная информация** – указания, позволяющие упростить работу с изделием

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

## СОДЕРЖАНИЕ

1. Назначение компонента.....	4
1.1. Условия применения.....	4
2. Ограничения.....	6
3. Установка и использование компонента.....	7
3.1. Установка компонента в сегменте разработки и в промышленном сегменте.....	8
3.2. Установка криптопровайдера «КриптоПро CSP» .....	9
3.3. Установка библиотеки «КриптоПлатформа» (gis-cryptoplatform17) .....	15
3.4. Установка приложения «Litoria Desktop 2».....	15
3.5. Выпуск сертификата .....	16
3.5.1. Первый запуск приложения «Litoria Desktop2».....	16
3.5.2. Запуск приложения «Litoria Desktop 2» от имени и с правами пользователя .....	16
3.5.3. Формирование сертификата.....	17
3.5.4. Импорт сертификата в личное хранилище пользователя ОС .....	22
3.6. Установка компонента «PLsPgSQL» .....	25
3.6.1. Установка пакета «jatoba5-plspgsql».....	25
3.6.2. Редактирование конфигурационного файла «postgresql.conf».....	28
3.6.3. Установка расширения.....	28
3.7. Описание утилиты wrpssql.....	29
3.7.1. Установка прав на каталоги .....	29
3.7.2. Процедура обфускации .....	30
3.7.3. Перемещение преобразованной функции в промышленный сегмент .....	33
4. Сообщения об ошибках .....	35
4.1. Ошибка установки пакета «jatoba<ver>-plspgsql».....	35
4.2. Ошибка запуска приложения «Litoria Desktop 2» .....	35
Перечень сокращений.....	37

## 1. НАЗНАЧЕНИЕ КОМПОНЕНТА

PL/spgSQL – процедурный язык PL/pgSQL с дополнительными функциями безопасности.

Расширение PL/spgSQL добавляет в СУБД новый язык plspgsql, обеспечивающий разработчику дополнительные функции безопасности при создании хранимых процедур.

В состав СУБД «Jatoba» включена утилита обфускации wrlpgsql, которая создает в dst-dir обфусцированную версию src-dir с заменой процедур и функций с языка 'plpgsql' на 'plspgsql'.

Утилита обфускации использует серверную часть СУБД для обфускации отдельных SQL-команд.

### 1.1. Условия применения

Компонент «PLsPgSQL» может использоваться совместно с СУБД «Jatoba» версий 1.x – 4.x и выше в ОС, указанных в таблице 1.1.

Таблица 1.1 – Перечень поддерживаемых ОС

№	Наименование ОС	Серверная часть	Клиентская часть	Docker (ver.)	Сертификат ФСТЭК	
					№ серт.	Дата выдачи
1	Windows 10	X	X	—	—	—
2	Windows 11	X	X	—	—	—
3	Windows Server 2016	X	X	—	—	—
4	Windows Server 2019	X	X	—	—	—
5	Windows Server 2022	X	X	—	—	—
6	Astra Linux 1.7 Special Edition Смоленск (x86-64)	X	X	20.10.2	2557	30.01.2012
7	Astra Linux 1.8 (x86-64)	X	X	—	—	—
8	Astra Linux 2.12 Common Edition Орел (x86-64)	X	X	—	—	—
9	Debian 10	X	X	24.0.2	—	—
10	Debian 11	X	X	24.0.2	—	—
11	Debian 12	X	X	24.0.2	—	—
12	Альт 8 СП	X	X	20.10.11	3866	10.08.2018
13	Альт 10 СП	X	X	20.10.11	3866	10.08.2018
14	Альт 9.1 Server	X	X	—	—	—
15	Альт 10 Server	X	X	23.0.1	—	—
16	Ubuntu 20.04	X	X	24.0.2	—	—
17	Ubuntu 22.04	X	X	24.0.2	—	—
18	Ubuntu 24.04	X	X	24.0.2	—	—
19	ОСНОВА2	X	X	20.10.5	4381	31.03.2021

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

№	Наименование ОС	Серверная часть	Клиентская часть	Docker (ver.)	Сертификат ФСТЭК	
					№ серт.	Дата выдачи
20	РЕД ОС 7.3 Муром	X	X	20.10.1	4060	12.01.2019
21	РЕД ОС 8	X	X	—	—	—
22	РОСА 7.9	X	X	—	—	—
23	РОСА 12.4	X	X	—	—	—
24	RedHat Enterprise Linux 8	X	X	—	—	—
25	Oracle Linux 8.4	X	X	—	—	—

## 2. ОГРАНИЧЕНИЯ

При работе со скрытым кодом необходимо учитывать следующее:

- сокрытие усложняет восстановление исходного кода, но пароли и другие конфиденциальные данные не должны храниться в коде;
- исходный код должен быть `pgplsql`, преобразование процедур и функций других языков не поддерживается;
- скрытый код не может компилироваться в экземплярах СУБД отличных от СУБД «Jatoba»;
- в коде, к которому будет применяться сокрытие, не могут использоваться подстановочные переменные.

### 3. УСТАНОВКА И ИСПОЛЬЗОВАНИЕ КОМПОНЕНТА

Процесс сокрытия исходных текстов, процедур и функций в СУБД «Jatoba» отображен на рисунке 3.1.

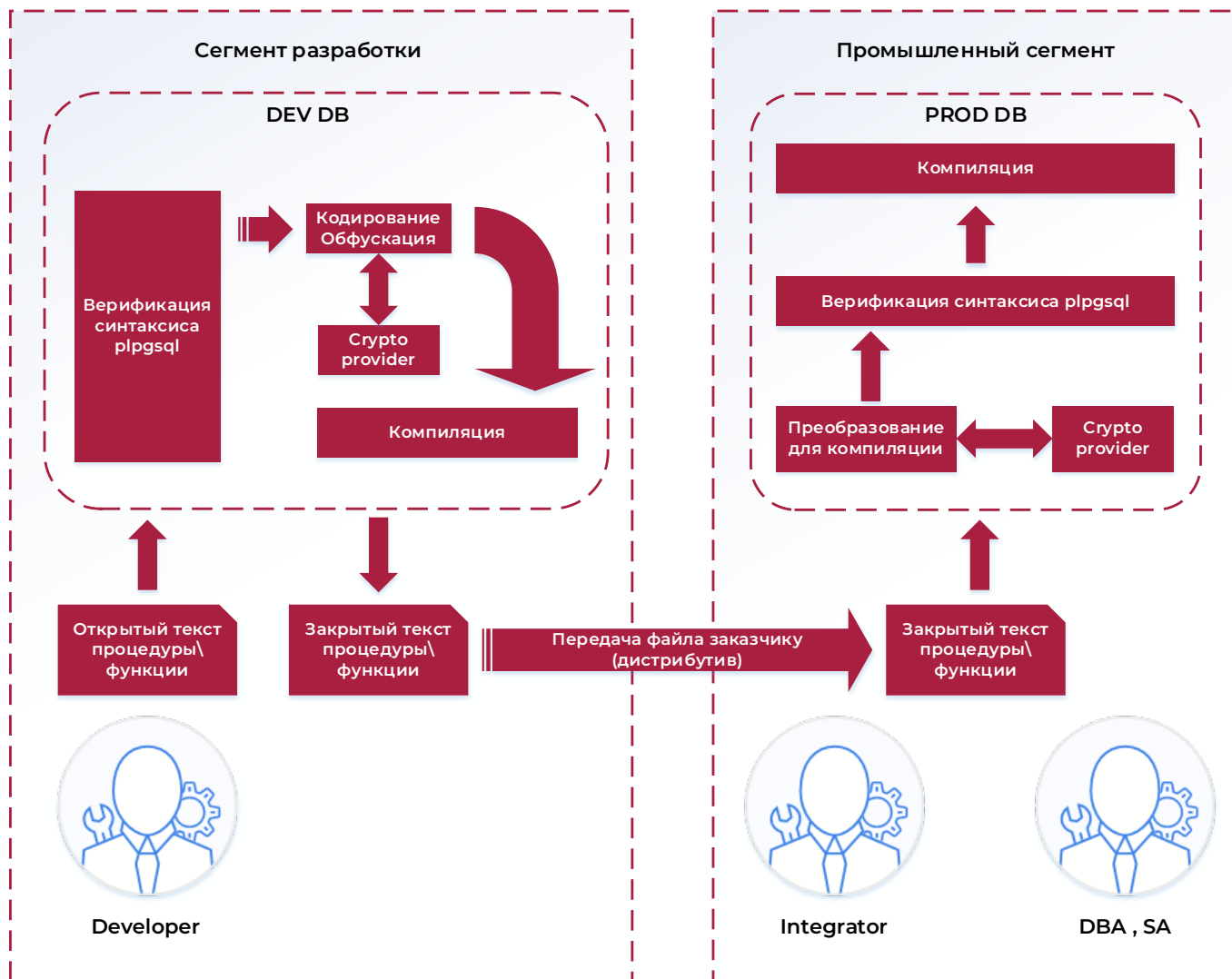


Рисунок 3.1 – Процесс сокрытия процедур и функций в СУБД «Jatoba»

В общем цикле операций выделяются следующие этапы:

- 1) Разработка автором ПО кода процедур и функций в штатном режиме.

С исходным кодом процедур и функций на языке `pgsql` разработчики работают в удобном для них виде без использования механизмов сокрытия.

- 2) Подготовка релиза к выпуску.

При подготовке кода к поставке или сборке релиза необходимо использовать СУБД «Jatoba» с соответствующими утилитами преобразования кода `pgsql`. Каждый

подготовленный файл преобразуется специальной утилитой, на выходе получается файл с pgsql, где текст преобразован в нечитаемую форму и не подлежит изменению.

3) Установка в промышленную среду.

Дистрибутив с преобразованными файлами передается службе эксплуатации заказчика. Файл может быть скомпилирован без раскрытия текста на СУБД «Jatoba».

4) Эксплуатация.

После компиляции текст файла находится в СУБД в преобразованном виде и не подлежит изменению, при этом полностью сохраняет работоспособность и соответствует изначальной логике производителя.

**3.1. Установка компонента в сегменте разработки и в промышленном сегменте**

Процесс установки компонента идентичен для всех сегментов и состоит из следующих этапов:

1) Установка СУБД.

Первоначально СУБД «Jatoba» устанавливается в двух сегментах.

Более подробно процесс установки СУБД описан в документе 643.72410666.00067-07 97 01 «Защищенная система управления базами данных «Jatoba». Руководство по установке».

2) Установка криптопровайдера «КриптоПро CSP».

3) Установка библиотеки «КриптоПлатформа» (gis-cryptoplatform17).

4) Выпуск сертификата.

Сертификаты выпускаются для каждого из участников процесса обфускации кода.

В приведенном описании, сертификат формируется с помощью ПК «Litoria Desktop 2»<sup>1</sup>. Допустим выпуск сертификатов и ключей любым другим доступным образом, при соблюдении характеристик и требований приведенных в п. 3.5.

– Установка пакета «jatoba<ver>-plpgsql»;

---

<sup>1</sup> Не входит в комплект поставки СУБД «Jatoba»



Пакет использует внешние зависимости и для удовлетворения их требуется установка следующих внешних продуктов:

1) `lsb-cryptossp` – криптопровайдер «КриптоПро CSP» версии 5.0. Данный пакет распространяется в виде самостоятельного DEB/RPM дистрибутива. Для установки необходимо обратиться к документации на официальном сайте разработчика КриптоПро – для доступа требуется зарегистрированный пользователь по правилам сайта <https://cryptopro.ru/user>);

Установка пакета описана в разделе 3.2 настоящего документа.

2) `gis-cryptoplatform17` – библиотека «КриптоПлатформа» версия 1.7.3-4. Данный пакет распространяется в составе дистрибутива СУБД «Jatoba» и устанавливается автоматически по зависимостям при установке пакета `jatoba<ver>-plpgsql`. Продукт поставляет ООО «Газинформсервис».

Установка пакета описана в разделе 3.3 настоящего документа.

– Редактирование конфигурационного файла `postgresql.conf`;

Вносятся параметры выпущенных сертификатов для каждого экземпляра СУБД.

– Установка расширения.

### **3.2. Установка криптопровайдера «КриптоПро CSP»**

Сборки продуктов загружаются с официального сайта ООО «Крипто-Про» <https://www.cryptopro.ru/>.

На сайте ООО «Крипто-Про» перейти по следующему пути:

Скачать – КриптоПро CSP – Сертифицированные версии – «Крипто Про CSP 5.0» – для – «Unix» – КриптоПро CSP 5.0 для Linux (x64, deb)

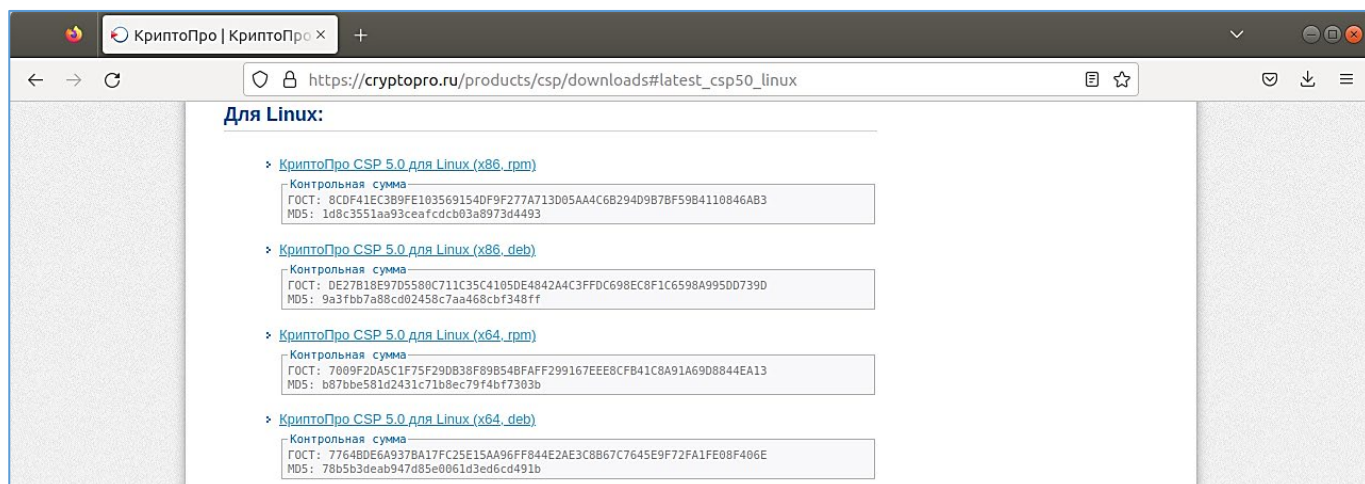


Рисунок 3.2 – Страница загрузки

Распаковать полученный архив linux-amd64\_deb.tgz, например, в каталог «install» и назначить права на выполнение скрипта «install\_gui.sh» командой:

```
chmod +x /<PathToCryptoPro>/install_gui.sh
```

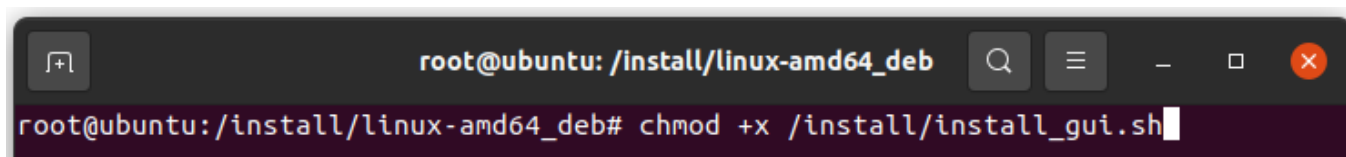


Рисунок 3.3 – Команда установки прав выполнения «install\_gui.sh»

После чего запустить скрипт командой:

```
./install_gui.sh
```

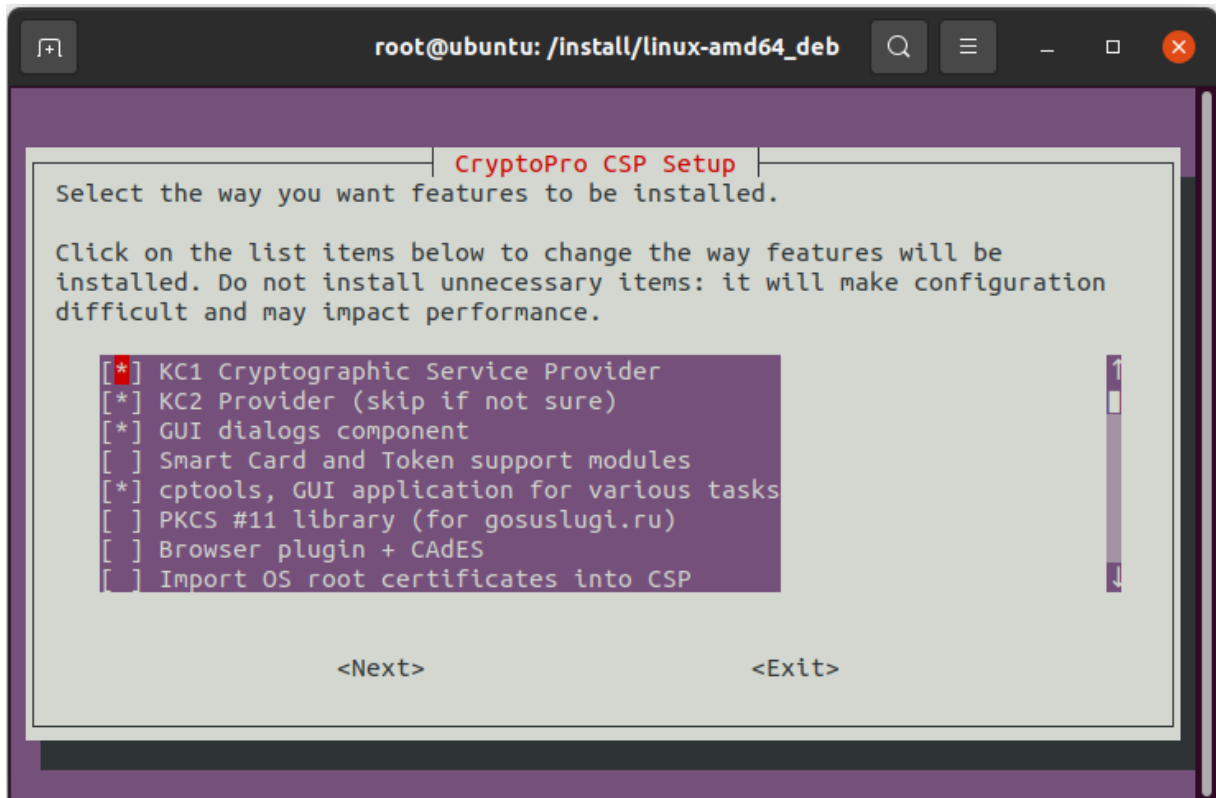


Рисунок 3.4 – Выбор компонент для установки криптопровайдера

В окне установить флаги:

- KC1 Cryptographic Service Provider;
- KC2 Provider (skip if not sure);
- GUI dialogs component;
- cptools, GUI applications for various tasks.

Подтвердить установку компонент нажав кнопку «Next».

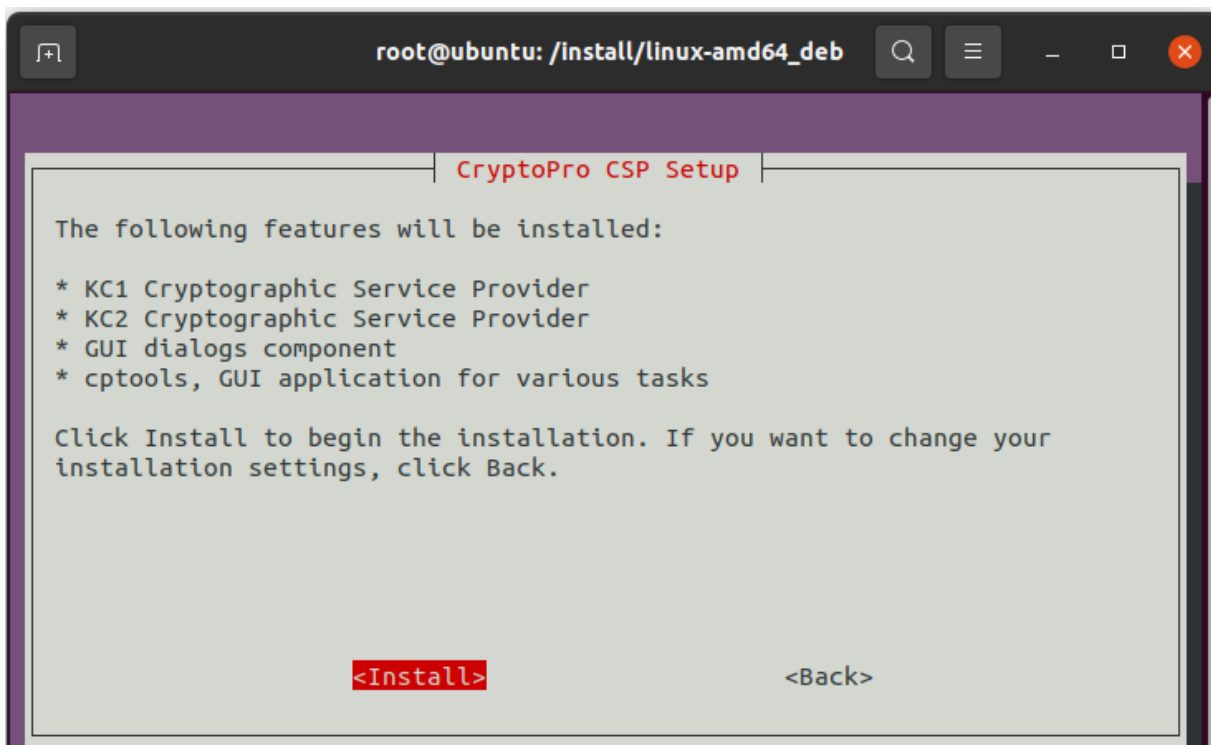


Рисунок 3.5 – Окно подтверждения установки компонент

По окончании процесса установки нажмите кнопку «ОК».

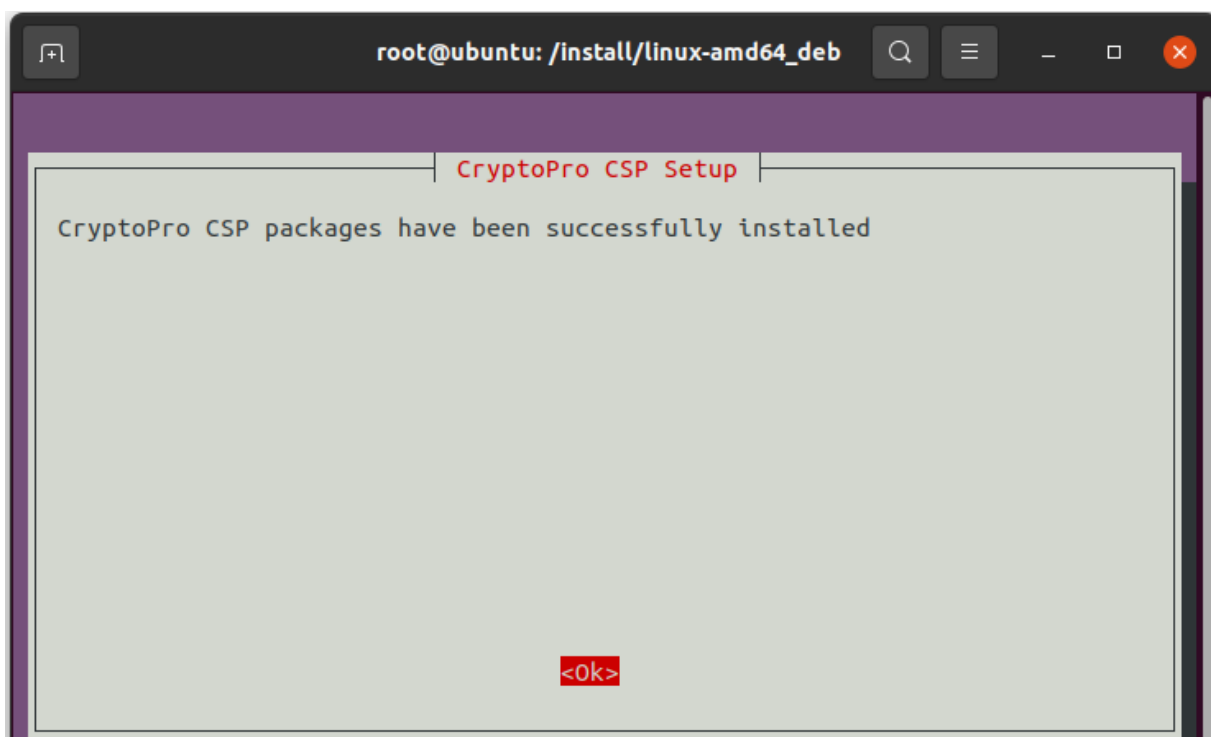


Рисунок 3.6 – Окно окончания установки

После установки компонент будет предложено активировать лицензию. На данном этапе возможно либо ввести ключ лицензии, либо пропустить этап.

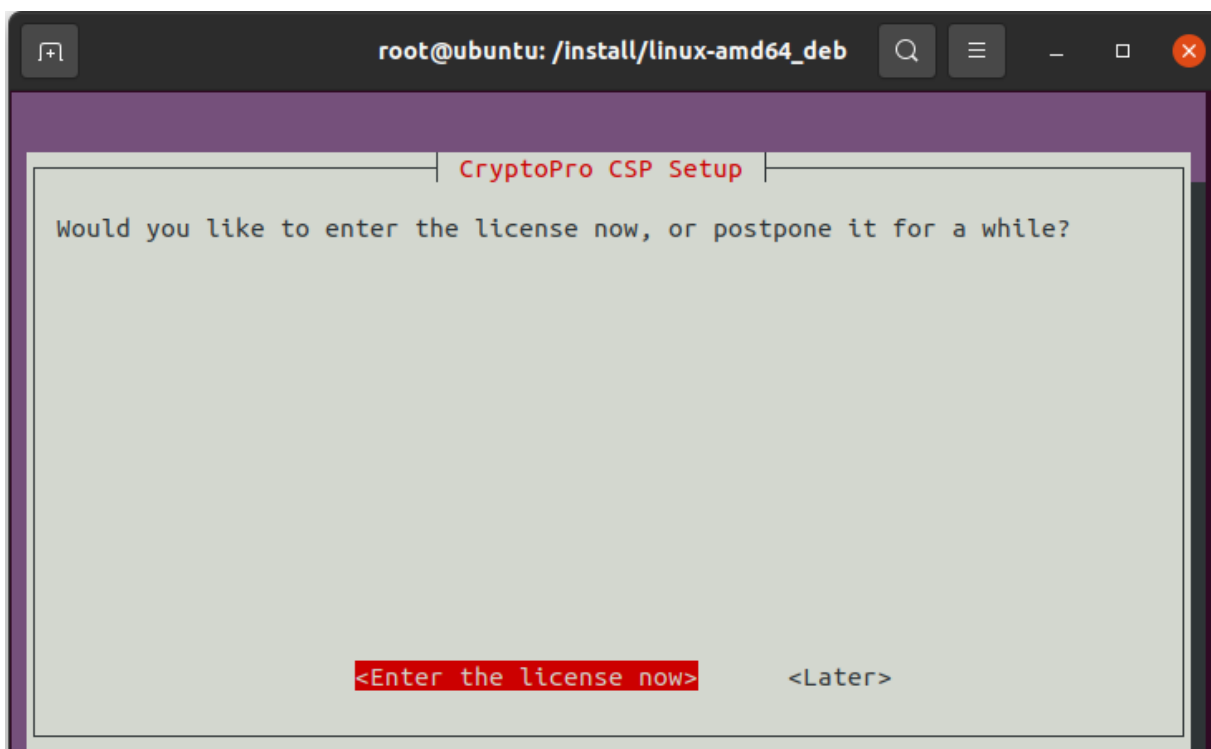


Рисунок 3.7 – Окно проверки статуса лицензии

В следующем окне требуется нажать кнопку «Exit» для выхода из скрипта установки.

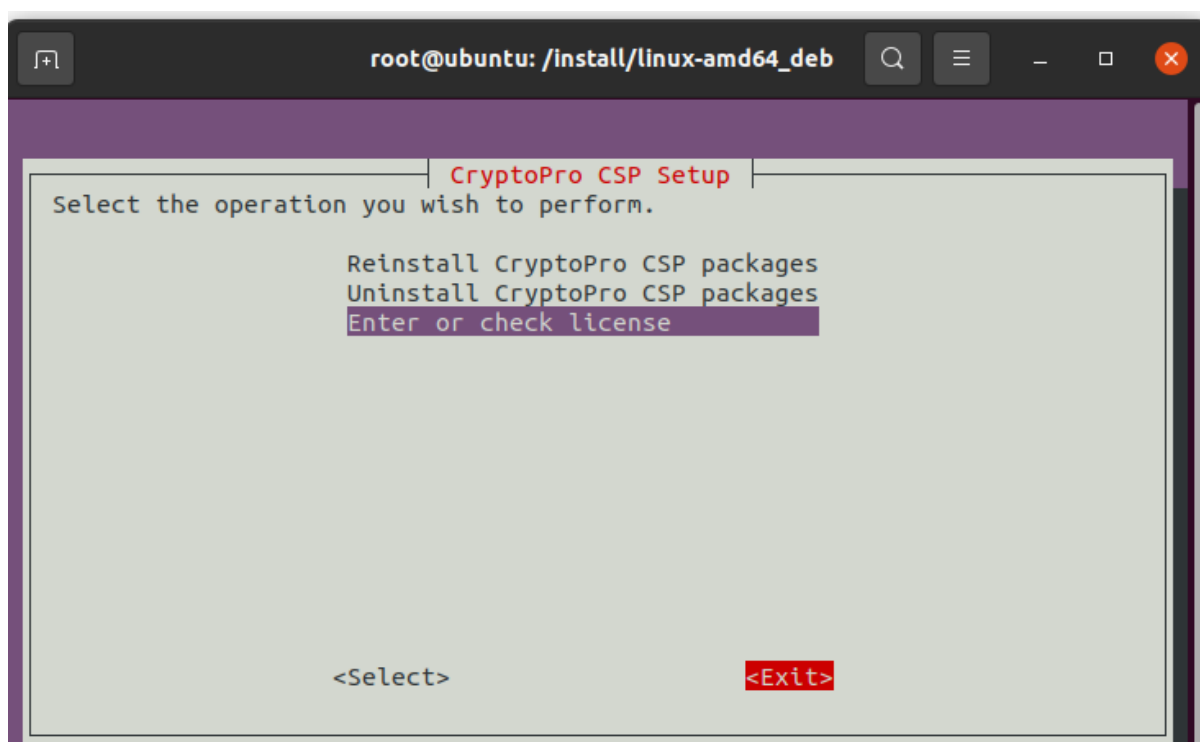


Рисунок 3.8 – Выход из скрипта установки

Далее активируется биологический ДСЧ.

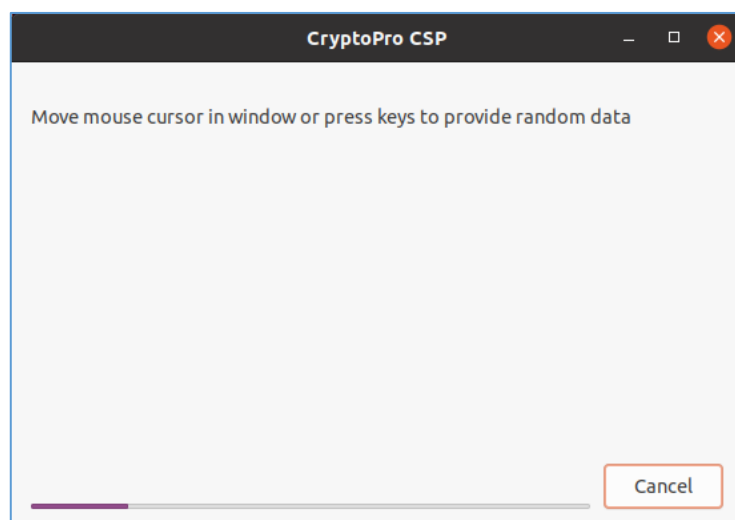


Рисунок 3.9 – Датчик случайных чисел

В приложении «Инструменты КриптоПро» на вкладке «Общее» можно проверить статус лицензии.

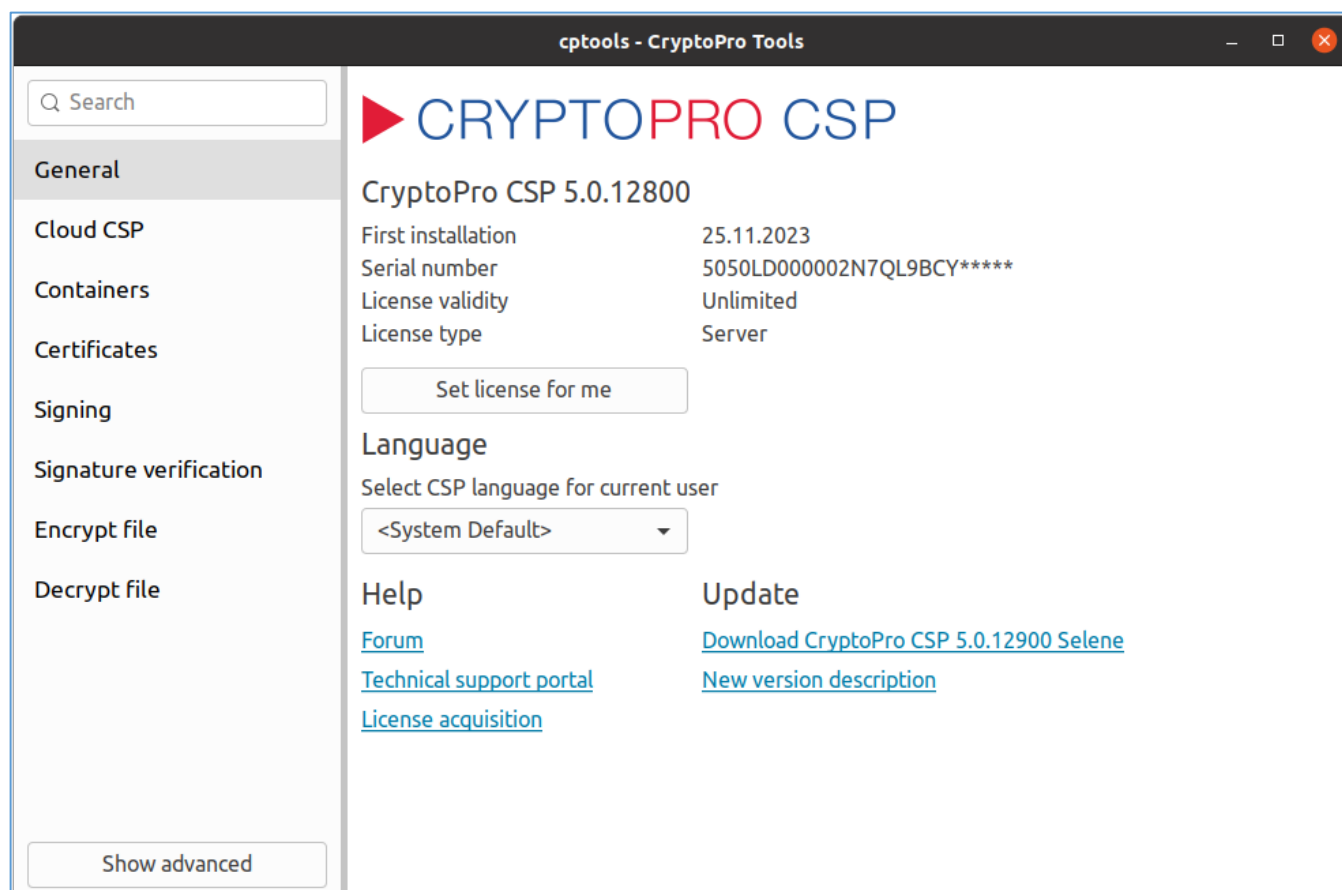


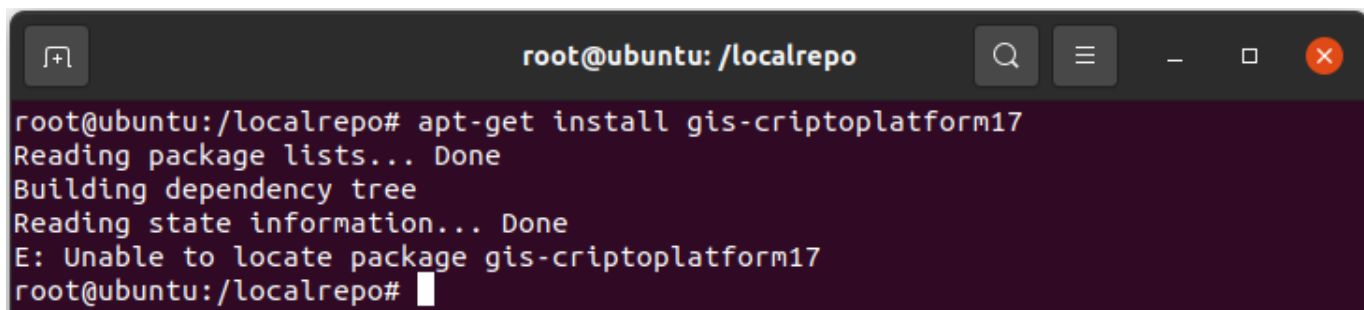
Рисунок 3.10 – Проверка статуса лицензии «В приложении «Инструменты КриптоПро»

На данном шаге установка криптопровайдера закончена.

### 3.3. Установка библиотеки «КриптоПлатформа» (gis-cryptoplatform17)

Установка библиотеки «КриптоПлатформа» выполняется командой:

```
apt-get install gis-cryptoplatform17
```



```
root@ubuntu: /localrepo
root@ubuntu:/localrepo# apt-get install gis-cryptoplatform17
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package gis-cryptoplatform17
root@ubuntu:/localrepo#
```

Рисунок 3.11 – Установка библиотеки «КриптоПлатформа»

### 3.4. Установка приложения «Litoria Desktop 2»

Имеющийся архив с дистрибутивом графического интерфейса «Litoria Desktop 2» распаковать в директорию.



```
<- /install/Litoria
.n          Name
/..
litoria_2.9.6-1_amd64.deb
*litoria_install.sh
```

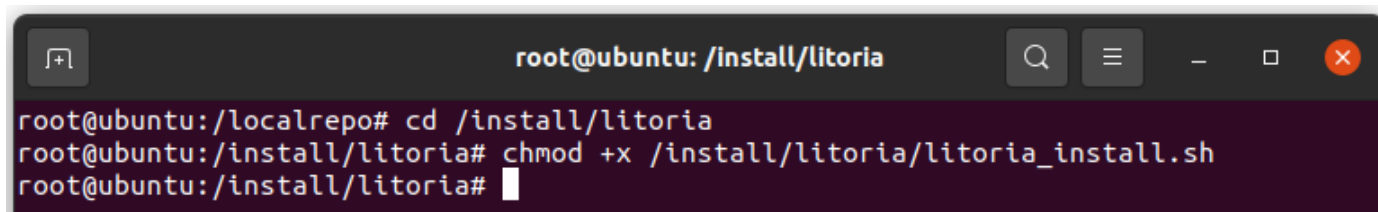
Рисунок 3.12 – Структура каталога

Установить права для выполнения скрипта командой:

```
chmod +x /<PathToCryptoPro>/litoria_install.sh
```

В рассматриваемом примере команда будет следующей:

```
chmod +x /install/litoria_install.sh
```

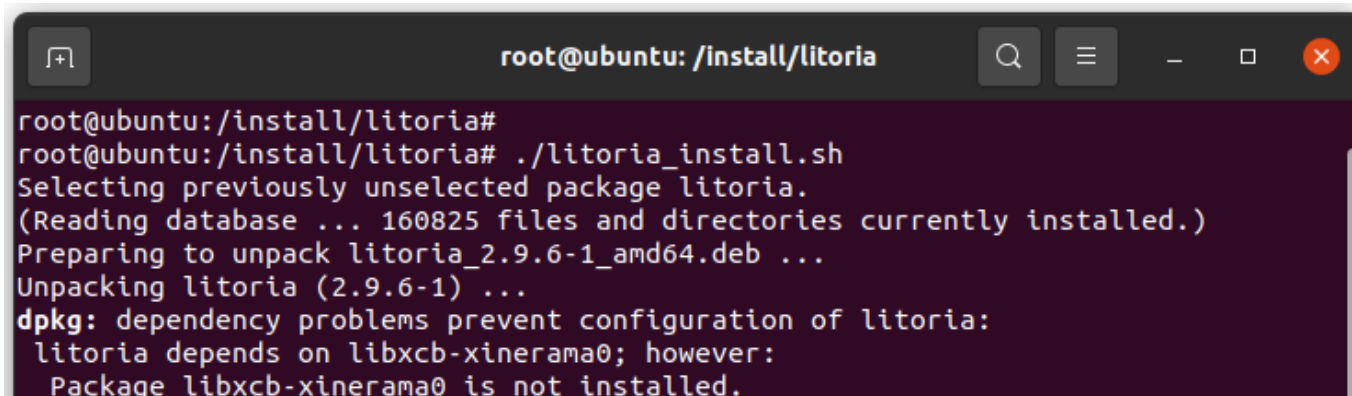


```
root@ubuntu: /install/litoria
root@ubuntu:/localrepo# cd /install/litoria
root@ubuntu:/install/litoria# chmod +x /install/litoria/litoria_install.sh
root@ubuntu:/install/litoria#
```

Рисунок 3.13 – Команда установки прав на выполнение скрипта

После чего запустить скрипт установки графического интерфейса «Litoria Desktop 2»:

```
./litoria_install.sh
```



```
root@ubuntu:/install/litoria#  
root@ubuntu:/install/litoria# ./litoria_install.sh  
Selecting previously unselected package litoria.  
(Reading database ... 160825 files and directories currently installed.)  
Preparing to unpack litoria_2.9.6-1_amd64.deb ...  
Unpacking litoria (2.9.6-1) ...  
dpkg: dependency problems prevent configuration of litoria:  
 litoria depends on libxcb-xinerama0; however:  
  Package libxcb-xinerama0 is not installed.
```

Рисунок 3.14 – Установка графического интерфейса «Litoria Desktop 2»

### 3.5. Выпуск сертификата

Пакет устанавливается в сегменте разработки и в промышленном сегменте.

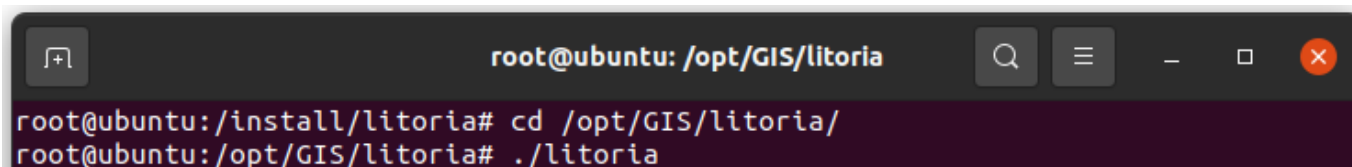
#### 3.5.1. Первый запуск приложения «Litoria Desktop2»

Первый запуск приложения должен выполняться от имени и с правами пользователя «root», необходимо перейти в каталог:

```
/opt/GIS/litoria
```

запустить клиента:

```
cd /opt/GIS/litoria/  
./litoria
```



```
root@ubuntu:/install/litoria# cd /opt/GIS/litoria/  
root@ubuntu:/opt/GIS/litoria# ./litoria
```

Рисунок 3.15 – Первый запуск приложения

В открывшемся окне приложения установить тип лицензии.

#### 3.5.2. Запуск приложения «Litoria Desktop 2» от имени и с правами пользователя

Приложение должно запускаться от имени и с правами пользователя, для которого будет формироваться сертификат.

Для этого требуется:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------



- Выйти из текущего сеанса.
- Войти в ОС от требуемого пользователя.

В представляемом примере используется пользователь «postgres».

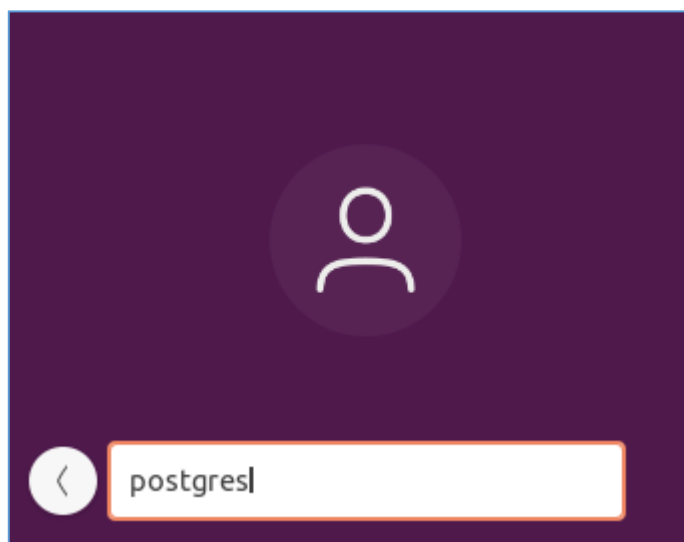


Рисунок 3.16 – Вход в ОС от имени пользователя «postgres»

- Запустить приложение «Litoria Desktop 2».

Приложение «Litoria Desktop 2» может быть запущено двумя способами:

- 1) Через терминал используя команду:

```
cd /opt/GIS/litoria/  
./litoria
```

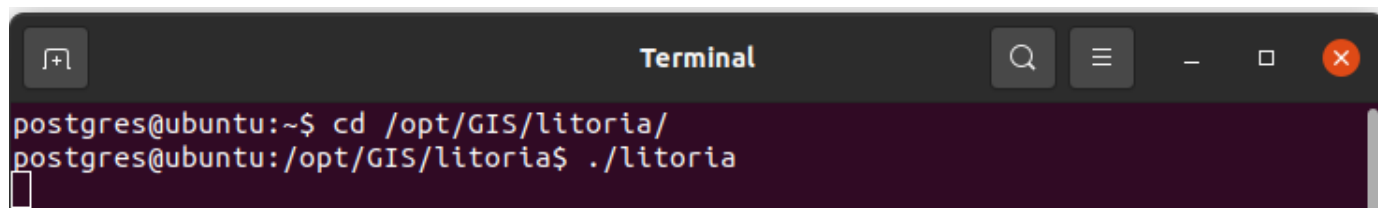


Рисунок 3.17 – Запуск приложения «Litoria Desktop 2» от имени пользователя

- 2) Через список установленного ПО.

### 3.5.3. Формирование сертификата

В меню «Сертификаты» (Certificates) выбрать вкладку «Запрос на сертификат» (Request) и в окне «Криптопровайдер» (Cryptography tool) установить параметры, приведенные в таблице 3.1.

Таблица 3.1 – Значения устанавливаемые в вкладке «Запрос на сертификат»

Область поля			Тип поля
Поле (RU)	Поле (ENG)	Значение	
<b>«Криптопровайдер» (Cryptography tool)</b>			
Криптосредство	Name	Crypto_Pro GOST R 34. 10-2012 Cryptographic Service Provider	Выпадающий список
Алгоритм	Algorithm	GOST R 34. 10-2012 256	По умолчанию
Длина ключа	Key length	512	По умолчанию
<b>Контейнер для ключевой пары (Key pair container)</b>			
Сгенерировать	Generate		флаг
ПИН-код	PIN-code	12345678	
Создать экспортируемый контейнер	Mark as exportable	флаг	флаг
<b>Информация о владельце ключа (Subject key information)</b>			
Шаблон	Template	Пустой (Empty)	Выпадающий список
<b>Information</b>			
Общее имя	Common name	devel	Текстовое поле

The screenshot shows the 'Cryptography tool' window in Litoria Desktop 2. The window is divided into several sections:

- Cryptography tool:** Name is 'Crypto-Pro GOST R 34.10-2012 Cryptographic Service Provider', Algorithm is 'GOST R 34.10-2012 256', and Key length is '512'.
- Key pair container:** Name is 'Generate', PIN-code is '12345678', and 'Mark as exportable' is checked.
- Subject key information:** Template is 'Empty', Common name is 'devel', and other fields like Surname, Given name, Initials, email, INN(TIN), INNLE, and OGRN are empty.

At the bottom, there are tabs for 'Information', 'Settings', and 'Signature'. A 'Create' button is located at the bottom right.

Рисунок 3.18 – Окно формирования сертификата

В области поля «Информация о владельце ключа» (Subject key information) перейти в вкладку «Параметры» (Settings) и установить параметры, приведенные в таблице 3.2.

Таблица 3.2 – Значения, устанавливаемые в области поля «Информация о владельце ключа»

Область поля			
Поле (RU)	Поле (ENG)	Значение	Тип поля
<b>Информация о владельце ключа (Subject key information)</b>			
<b>Использование ключа</b>			
Подпись данных	Digital signature		флаг
Неотрекаемость	Non-repudiation		флаг
Шифрование ключа	Key encryption		флаг
Шифрование данных	Data encryption		флаг
<b>Назначение сертификата</b>			
Цифровая подпись	Digital signature		флаг

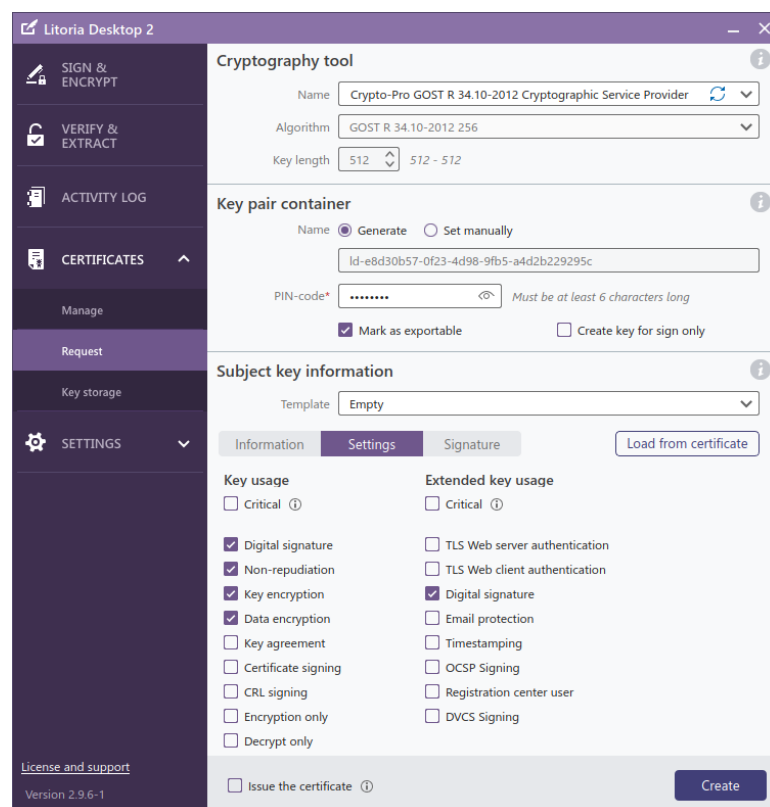


Рисунок 3.19 – Параметры в области поля «Информация о владельце ключа»

После установки параметров нажмите кнопку «Создать» (Create) и генерируем случайную последовательность движениями мышки или клавишами на клавиатуре.

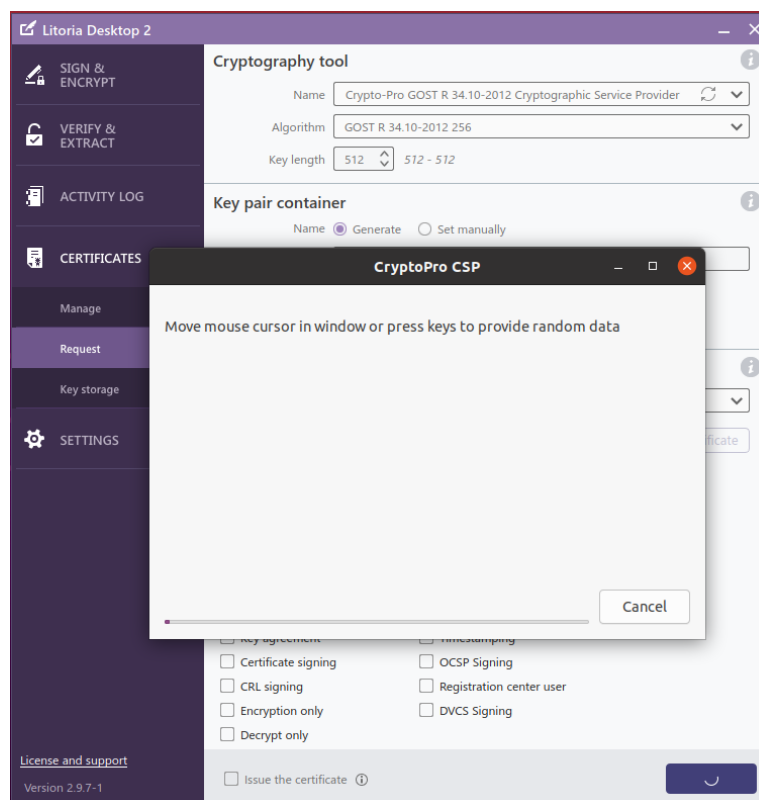


Рисунок 3.20 – Генератор ДСЧ

В результате получаем запрос на сертификат.

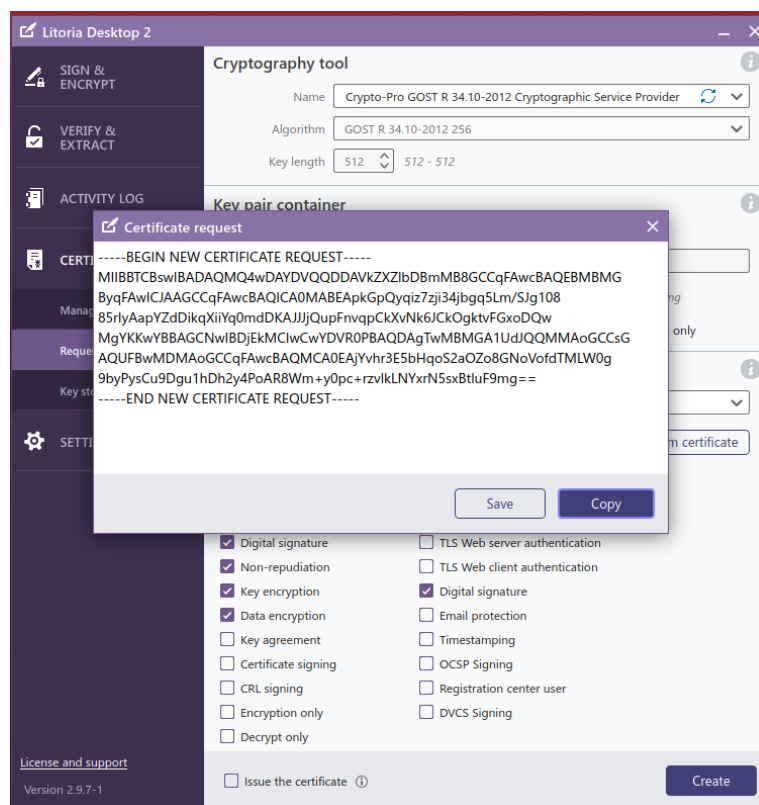


Рисунок 3.21 – Запрос на сертификат

Копируем получившийся текст нажатием кнопки «Копировать» (Copy).

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------



В рассматриваемом примере используется тестовый удостоверяющий центр компании «Газинформсервис».

Для формирования сертификата в промышленной среде, следует использовать реальный удостоверяющий центр.

С помощью браузера открываем страницу тестового удостоверяющего центра по адресу: <http://testca.gaz-is.ru/#>

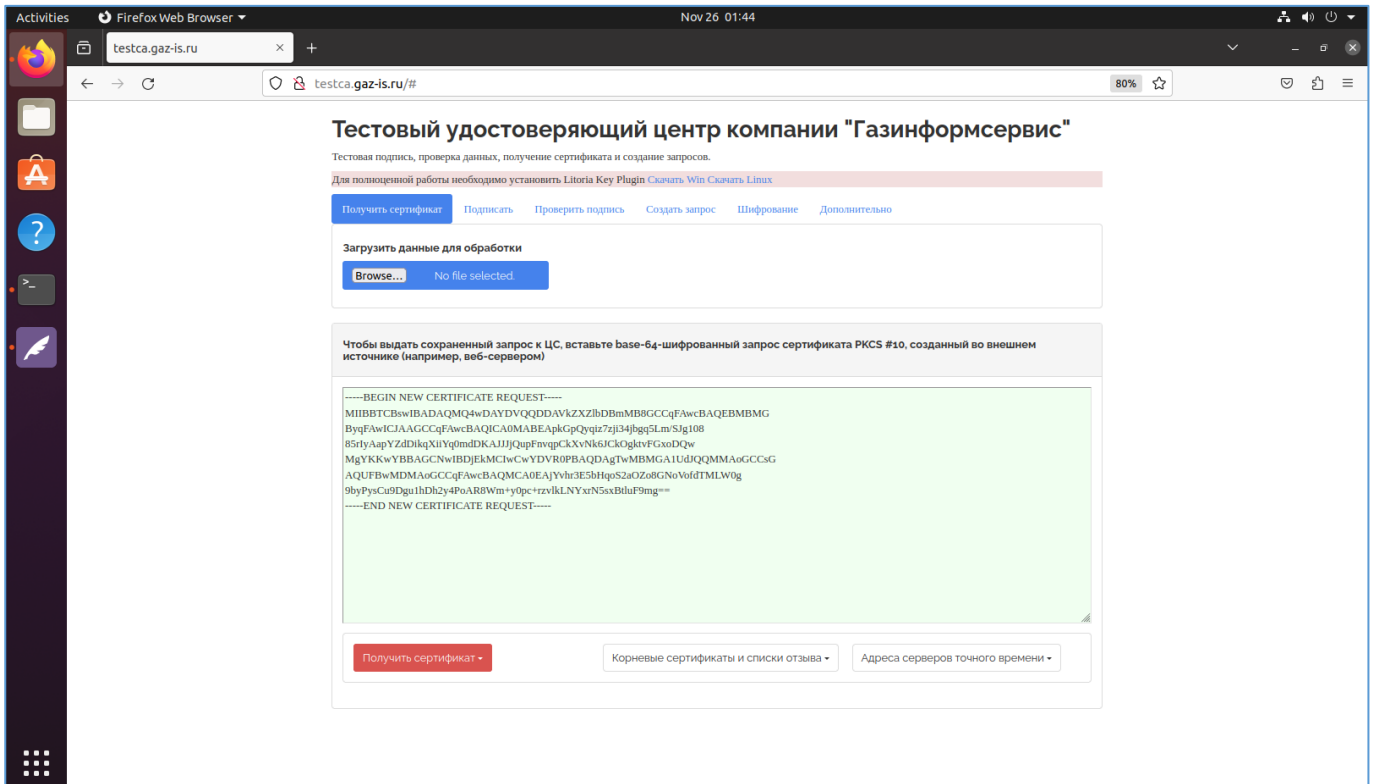


Рисунок 3.22 – Окно тестового УЦ

В окне УЦ необходимо вставить скопированный запрос на сертификат.

При нажатии кнопки «Получить сертификат», выбрать пункт «Получить сертификат 2012 гост».

После этого сертификат загрузится в стандартную директорию для загрузок.

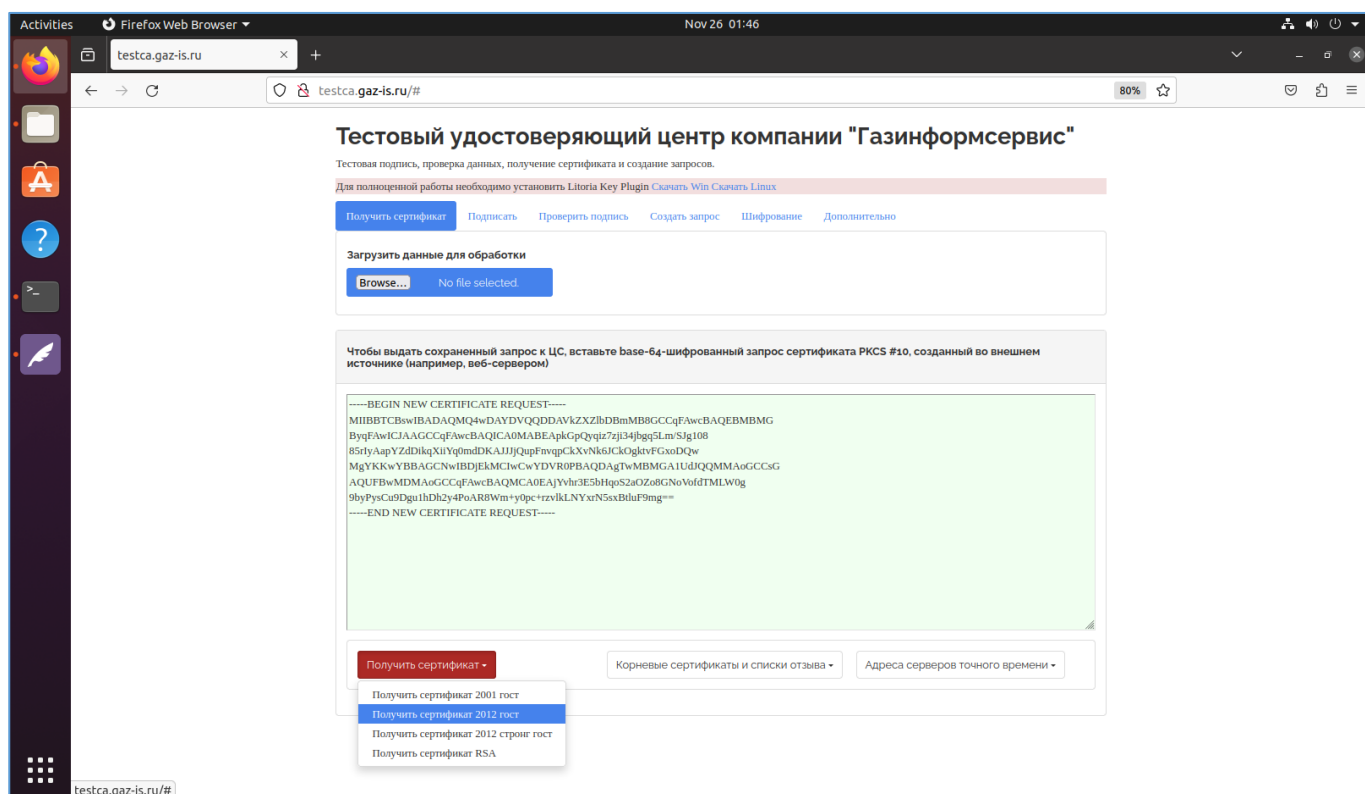


Рисунок 3.23 – Получение сертификата

### 3.5.4. Импорт сертификата в личное хранилище пользователя ОС

В приложении «Litoria Desktop 2» требуется перейти на вкладку «Управление» (Manage) и импортировать сертификат. Из директории загрузки сертификата возможно его перетащить мышкой в окно импорта сертификата.

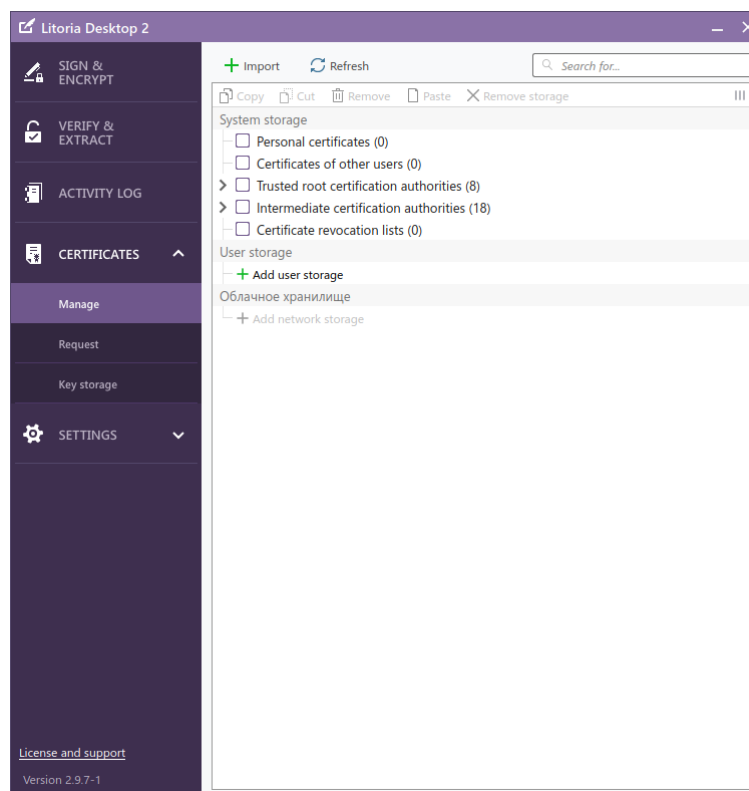


Рисунок 3.24 – Вкладка «Управление»

Импортированный сертификат необходимо перетащить мышкой в раздел «Личные сертификаты» (Personal certificates), после чего нажать «Установить» (Install).

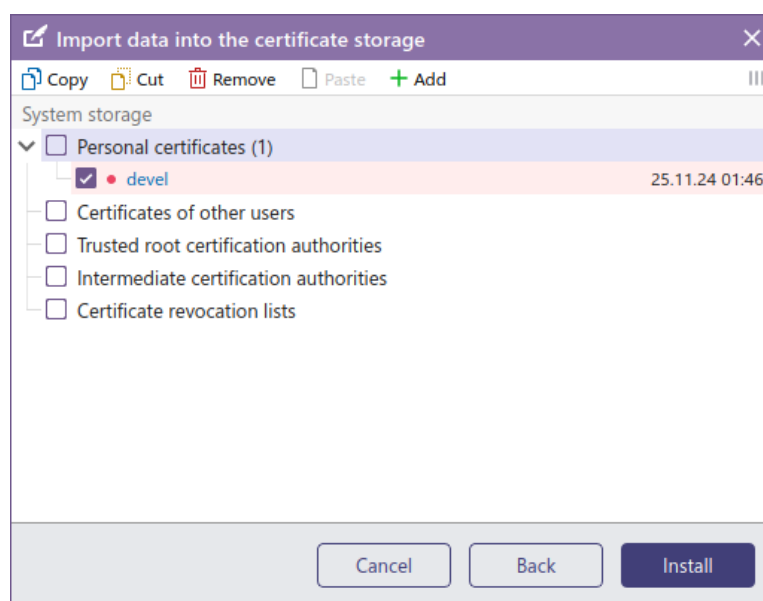


Рисунок 3.25 – Размещение сертификата в разделе «Личные сертификаты»

После установки в общем списке выбрать требуемый сертификат, для подтверждения подлинности нажать кнопку, отмеченную на рисунке 3.26.

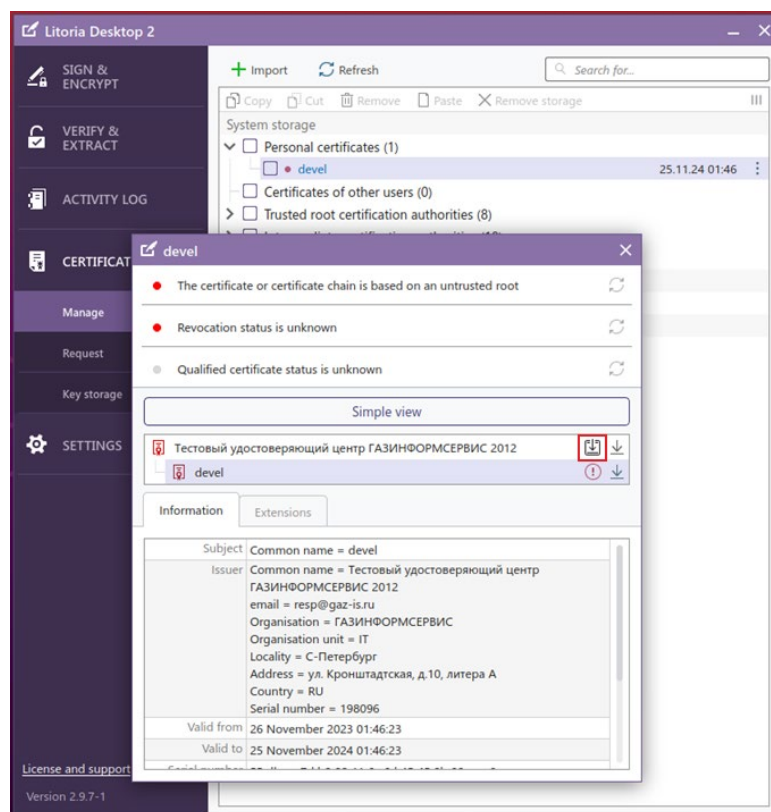


Рисунок 3.26 – Импорт сертификата в личное хранилище пользователя

В всплывающем окне подтвердить размещение сертификата в личном хранилище пользователя.

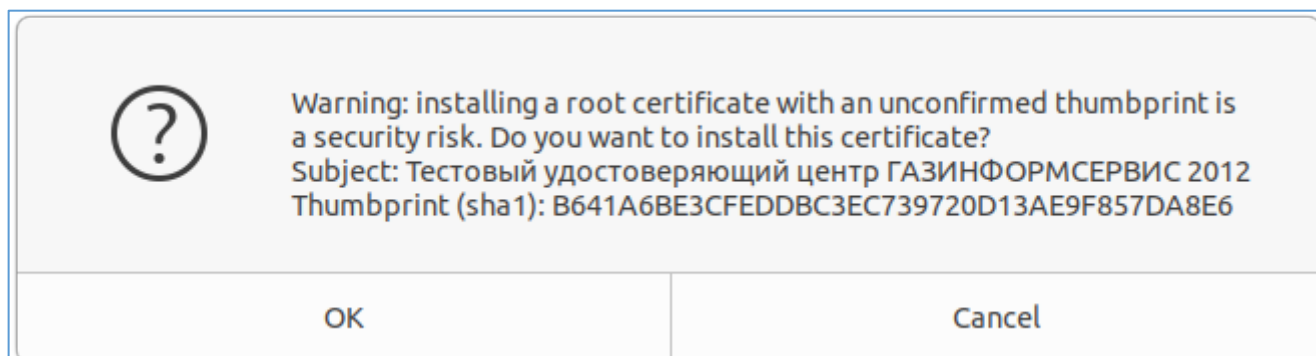


Рисунок 3.27 – Окно подтверждения



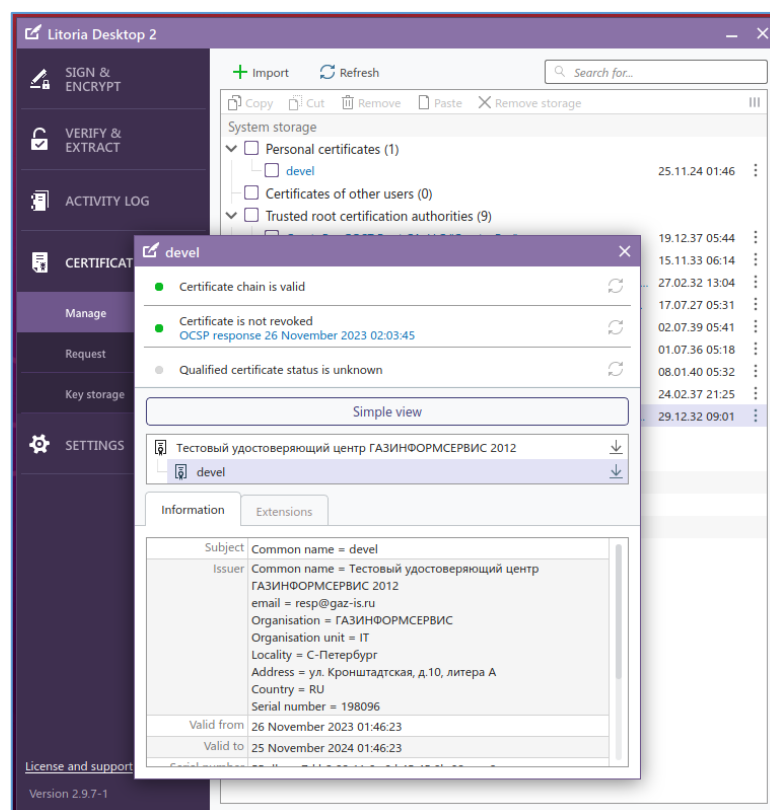


Рисунок 3.28 – Завершение импорта сертификата

На данном шаге формирование и импорт сертификата закончено.

### 3.6. Установка компонента «PLsPgSQL»

Компонент «PLsPgSQL» устанавливается в сегменте разработки и в промышленном сегменте.

#### 3.6.1. Установка пакета «jatoba5-plspgsql»

После того как, установлен криптопровайдер (см. п. 3.2) и криптоплатформа (см. п.3.3), становится доступной установка пакета компонента «PLsPgSQL».

Установку компонента возможно провести двумя способами:

- 1) установка из локального репозитория (CDROM) – производится из файлов, записанных на компакт-диск или скопированных с него;
- 2) установка непосредственно из deb/rpm-файлов – производится опционально, по усмотрению пользователя.

Компонент выполнен в виде отдельного deb или rpm-пакета. Установка компонента осуществляется средствами пакетного менеджера ОС. Для разных типов пакетных менеджеров команда установки немного отличается. Ниже приведены основные типы:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

– для систем на основе пакетного менеджера APT (к таким системам относятся все ОС семейства Debian, использующие deb-пакеты) команда установки следующая:

```
apt-get install jatoba<ver>-plspgsql
```

– для систем на основе пакетных менеджеров YUM/DNF (к таким системам относятся все ОС семейства RedHat и вышедшие из нее, использующие rpm-пакеты) команда установки следующая:

```
yum install jatoba<ver>-plspgsql
```

Отдельного уточнения требуют операционные системы ALT Linux и openSUSE.

– ALT Linux использует пакетный менеджер APT, но распространяется в виде rpm-пакетов и для нее команда установки выглядит аналогично Debian:

```
apt-get install jatoba<ver>-plspgsql
```

– openSUSE также распространяется в виде rpm-пакетов, но использует собственный пакетный менеджер zypper, для нее команда установки выглядит следующим образом:

```
zypper install jatoba<ver>-plspgsql
```

Более подробно процесс установки компонента описан в документе 643.72410666.00067-07 97 01 «Защищенная система управления базами данных «Jatoba». Руководство по установке».

Установка компонента в составе других версий СУБД «Jatoba» осуществляется аналогично. Отличие будет только в номере версии СУБД, в составе которой он распространяется. Например, jatoba5-plspgsql и т.п.

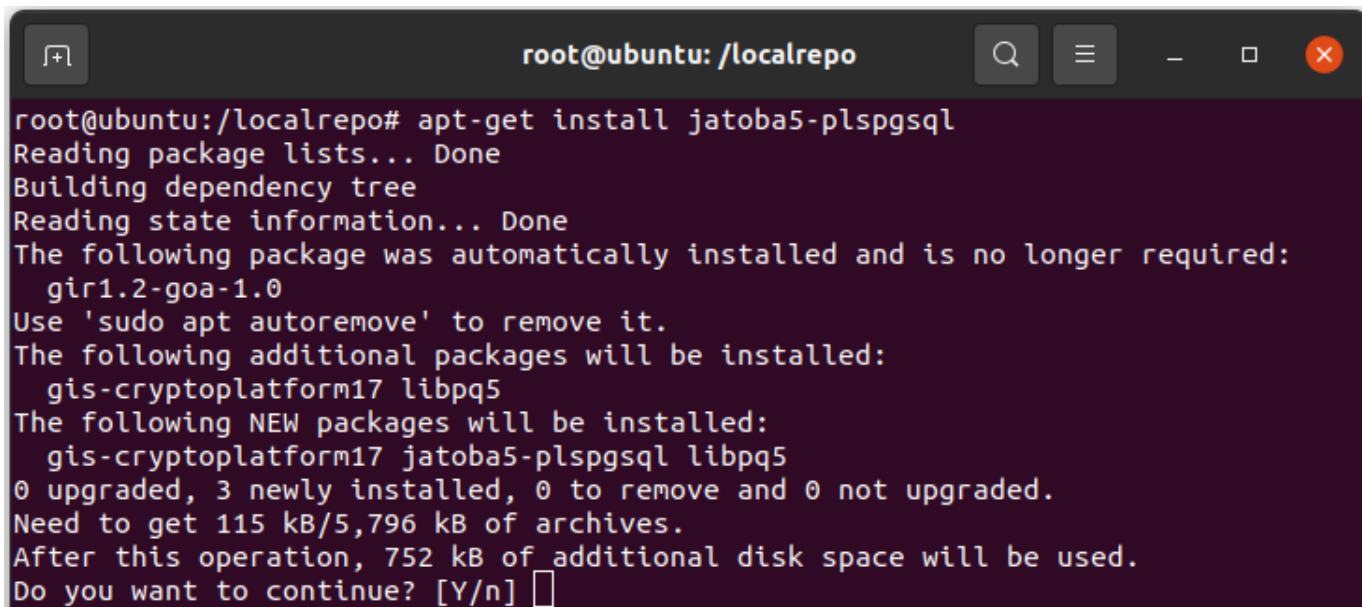
Удаление модуля также осуществляется средствами пакетного менеджера ОС. Вместо команды install нужно использовать соответствующую данному пакетному менеджеру команду удаления (remove, purge, erase и т.п.).

Для получения детальной информации по пакетному менеджеру рекомендуется обратиться к документации по ОС.

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

В рассматриваемом примере установка пакета под ОС Ubuntu выполняется командой:

```
apt-get install jatoba5-plspgsql
```

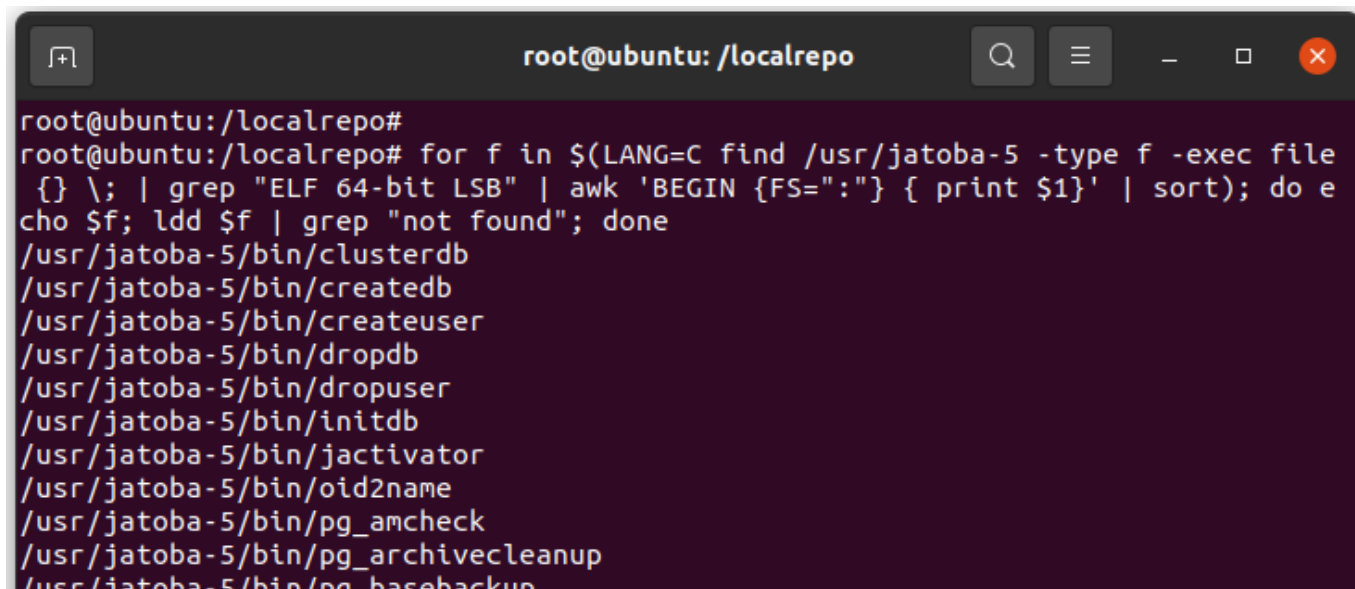


```
root@ubuntu: /localrepo
root@ubuntu:/localrepo# apt-get install jatoba5-plspgsql
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  gir1.2-goa-1.0
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  gis-cryptoplatfom17 libpq5
The following NEW packages will be installed:
  gis-cryptoplatfom17 jatoba5-plspgsql libpq5
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 115 kB/5,796 kB of archives.
After this operation, 752 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Рисунок 3.29 – Установка пакета «jatoba5-plspgsql»

Необходимо убедиться, что отсутствуют ошибки зависимостей, выполнив команду:

```
for f in $(LANG=C find /usr/jatoba-5 -type f -exec file {} \; |
grep "ELF 64-bit LSB" | awk 'BEGIN {FS=":"} { print $1}' |
sort); do echo $f; ldd $f | grep "not found"; done
```



```
root@ubuntu: /localrepo
root@ubuntu:/localrepo#
root@ubuntu:/localrepo# for f in $(LANG=C find /usr/jatoba-5 -type f -exec file
{} \; | grep "ELF 64-bit LSB" | awk 'BEGIN {FS=":"} { print $1}' | sort); do e
cho $f; ldd $f | grep "not found"; done
/usr/jatoba-5/bin/clusterdb
/usr/jatoba-5/bin/createdb
/usr/jatoba-5/bin/createuser
/usr/jatoba-5/bin/dropdb
/usr/jatoba-5/bin/dropuser
/usr/jatoba-5/bin/initdb
/usr/jatoba-5/bin/jactivator
/usr/jatoba-5/bin/oid2name
/usr/jatoba-5/bin/pg_amcheck
/usr/jatoba-5/bin/pg_archivecleanup
/usr/jatoba-5/bin/pg_basebackup
```

Рисунок 3.30 – Проверка отсутствия зависимостей

### 3.6.2. Редактирование конфигурационного файла «postgresql.conf»

В конфигурационный файл «postgresql.conf» в раздел «CUSTOMIZED OPTIONS», добавляются два параметра:

- plpgsql.cert\_cn – имя созданного сертификата;
- plpgsql.cert\_pincode – пин-код от сертификата.

Для каждой из СУБД устанавливаются собственные параметры.

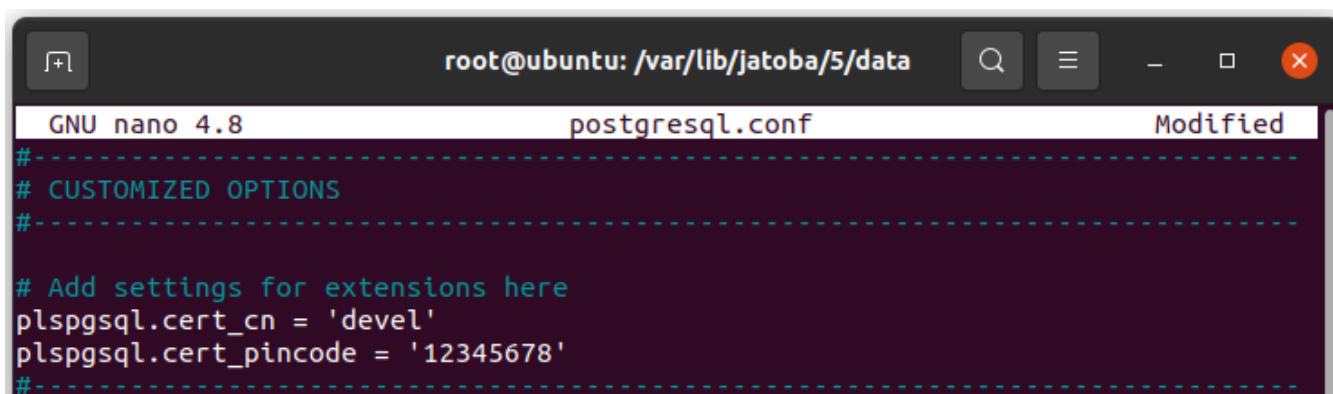


Рисунок 3.31 – Параметры сертификата в конфигурационном файле postgresql.conf

### 3.6.3. Установка расширения

Установка расширения в СУБД должна выполняться от имени и с правами пользователя, для которого был создан сертификат.

Авторизовавшись в СУБД требуется выполнить SQL-команду:

```
CREATE EXTENSION plpgsql
```

```

postgres@ubuntu:~$ psql
Password for user postgres:
psql (15.5)
Type "help" for help.

postgres=# create extension plpgsql;
CREATE EXTENSION
postgres=# \dx

               List of installed extensions
  Name      | Version | Schema  | Description
-----+-----+-----+-----
 plpgsql    | 1.0     | pg_catalog | PL/pgSQL procedural language
 plpgsql    | 2.0     | pg_catalog | PL/pgSQL procedural language with security fe
atures
(2 rows)

postgres=#
  
```

Рисунок 3.32 – Установка расширения

### 3.7. Описание утилиты wplpgsql

Одним из важных элементов функции сокрытия исходных текстов процедур и функций в СУБД «Jatoba» является утилита wplpgsql.

Утилита имеет модульную архитектуру и различные механизмы. Для преобразования текстов процедур и функций данные механизмы подключаются к утилите как модули расширения. Основным условием к механизмам преобразования является использование обратимых алгоритмов, так как на этапе установки в эксплуатируемую среду утилита производит валидацию синтаксиса после обратного преобразования текста.

Данная утилита разработана в рамках продукта СУБД «Jatoba» компанией ООО «Газинформсервис».

#### 3.7.1. Установка прав на каталоги

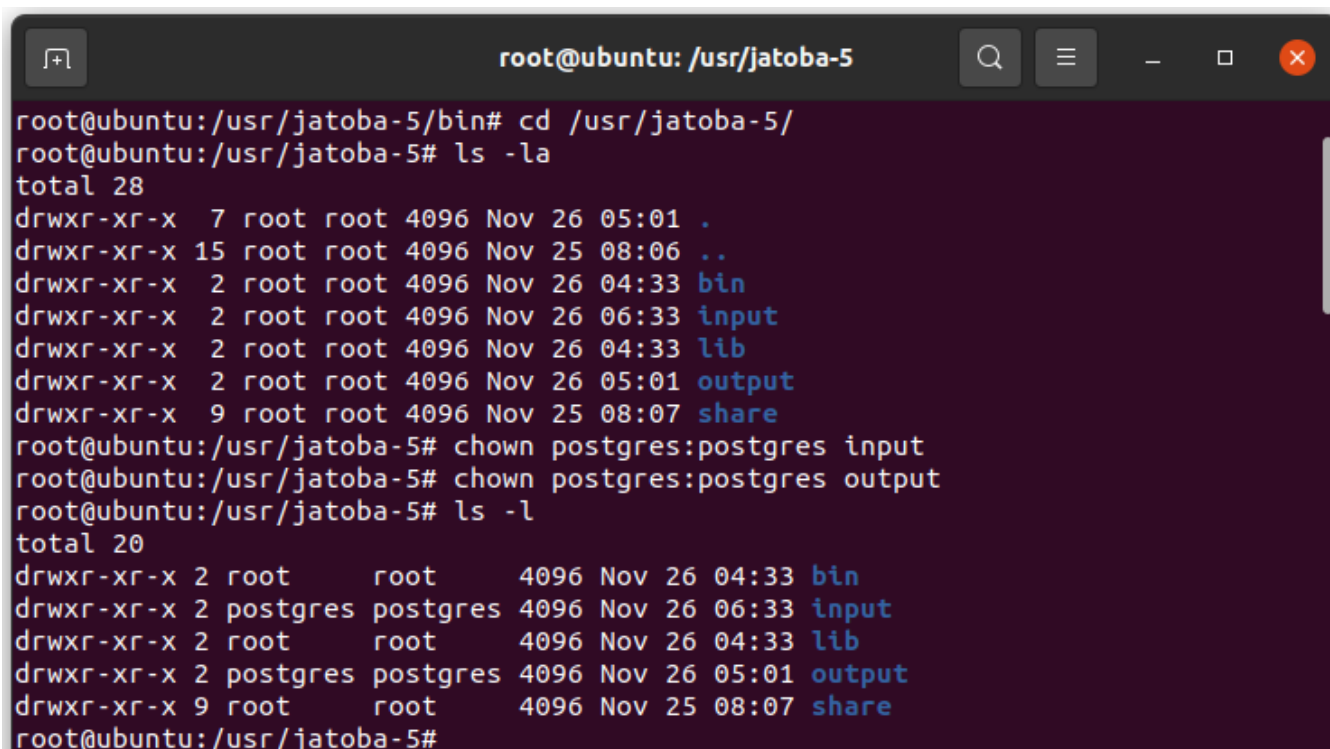
После установки расширения дополнительная функциональность с использованием специальной утилиты wplpgsql становится доступной.

В каталоге /usr/jatoba-5 создайте 2 подкаталога:

- input;
- output.

Для каталогов обязательно следует установить права доступа для пользователя от имени и с правами которого будет проводится процедура обфускации. Права устанавливаются командой в терминале ОС от пользователя «root»:

```
chown postgres:postgres input
chown postgres:postgres output
```



```
root@ubuntu: /usr/jatoba-5
root@ubuntu:/usr/jatoba-5/bin# cd /usr/jatoba-5/
root@ubuntu:/usr/jatoba-5# ls -la
total 28
drwxr-xr-x  7 root root 4096 Nov 26 05:01 .
drwxr-xr-x 15 root root 4096 Nov 25 08:06 ..
drwxr-xr-x  2 root root 4096 Nov 26 04:33 bin
drwxr-xr-x  2 root root 4096 Nov 26 06:33 input
drwxr-xr-x  2 root root 4096 Nov 26 04:33 lib
drwxr-xr-x  2 root root 4096 Nov 26 05:01 output
drwxr-xr-x  9 root root 4096 Nov 25 08:07 share
root@ubuntu:/usr/jatoba-5# chown postgres:postgres input
root@ubuntu:/usr/jatoba-5# chown postgres:postgres output
root@ubuntu:/usr/jatoba-5# ls -l
total 20
drwxr-xr-x 2 root      root      4096 Nov 26 04:33 bin
drwxr-xr-x 2 postgres postgres 4096 Nov 26 06:33 input
drwxr-xr-x 2 root      root      4096 Nov 26 04:33 lib
drwxr-xr-x 2 postgres postgres 4096 Nov 26 05:01 output
drwxr-xr-x 9 root      root      4096 Nov 25 08:07 share
root@ubuntu:/usr/jatoba-5#
```

Рисунок 3.33 – Установление прав на директории

### 3.7.2. Процедура обфускации

К примеру, исходный текст сохранен в файле /usr/jatoba-5/output/jatoba\_secret\_function.sql со следующим содержимым:

```
CREATE OR REPLACE FUNCTION jatoba_secret_function()
RETURNS text AS $$
BEGIN
    RETURN 'secret text';
END;
$$ LANGUAGE plpgsql;
```

```

root@ubuntu: /usr/jatoba-5
GNU nano 4.8      jatoba_secret_function.sql
CREATE OR REPLACE FUNCTION jatoba_secret_function()
RETURNS text AS $$
BEGIN
    RETURN 'secret text';
END;
$$ LANGUAGE plpgsql;
  
```

Рисунок 3.34 – Исходный текст сохранен в файле /usr/jatoba-5/output/jatoba\_secret\_function.sql



Процедура написана на языке plpgsql, который присутствует в СУБД «Jatoba» по умолчанию.

Процедура обфускации выполняется при помощи утилиты «wplpgsql», расположенной по пути /usr/jatoba-5/bin.

Команда обфускации имеет синтаксис:

```
./wplpgsql -i <path to input directory> -o <path to output directory> -m file
```

С параметрами, приведенными в таблице 3.3.

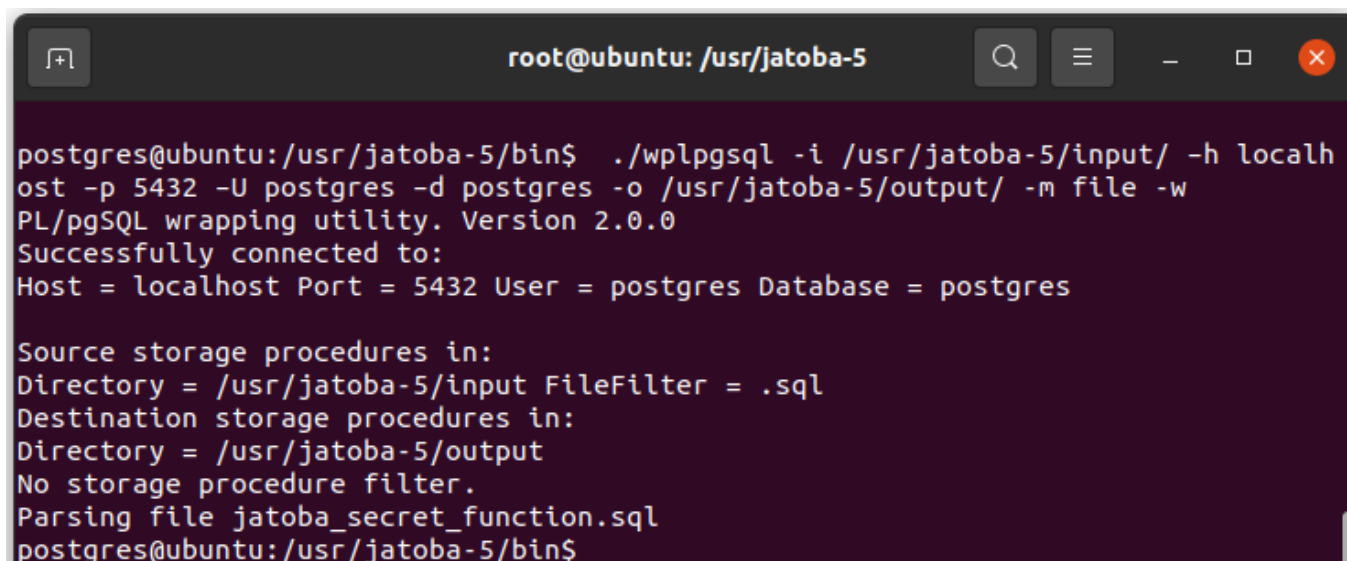
Таблица 3.3 – Параметры команды обфускации

№	Параметр	Значение параметра
1	-i	входящая директория «input»
2	<path to input directory>	Путь к директории
3	-h	Хост подключения
4	-p	Порт подключения
5	-U	Пользователь для подключения к СУБД
6	-d	БД подключения
7	-o	Исходящая директория «output»
8	-m	Режим работы. Имеет два значения: <ul style="list-style-type: none"> <li>file – обфускация в файле</li> <li>database – обфускация в БД</li> </ul>
9	-w	Не запрашивать пароль
10	-W	Запрос пароля

Преобразование читаемого исходного текста в файл с выполнением преобразования в нечитаемый текст в рассматриваемом примере выполняется командой:

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------

```
./wplpgsql -i /usr/jatoba-5/input/ -h localhost -p 5432 -U  
postgres -d postgres -o /usr/jatoba-5/output/ -m file -w
```



```
root@ubuntu: /usr/jatoba-5  
postgres@ubuntu:/usr/jatoba-5/bin$ ./wplpgsql -i /usr/jatoba-5/input/ -h localhost -p 5432 -U postgres -d postgres -o /usr/jatoba-5/output/ -m file -w  
PL/pgSQL wrapping utility. Version 2.0.0  
Successfully connected to:  
Host = localhost Port = 5432 User = postgres Database = postgres  
  
Source storage procedures in:  
Directory = /usr/jatoba-5/input FileFilter = .sql  
Destination storage procedures in:  
Directory = /usr/jatoba-5/output  
No storage procedure filter.  
Parsing file jatoba_secret_function.sql  
postgres@ubuntu:/usr/jatoba-5/bin$
```

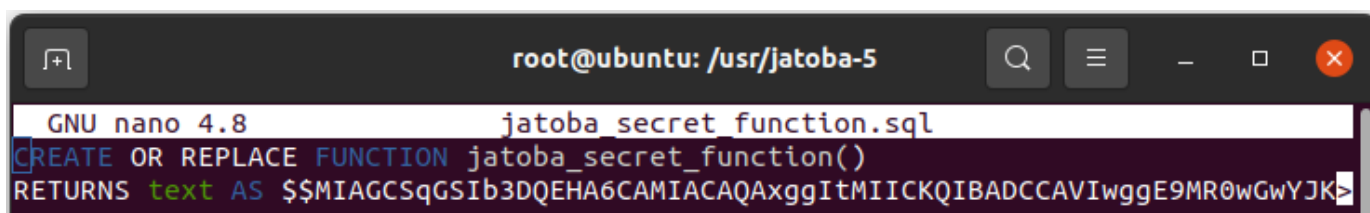
Рисунок 3.35 – Выполнение команды обфускации

Утилита:

- возьмет функцию из исходного каталога /usr/jatoba-5/input/;
- преобразует текст функции;
- преобразованный файл с тем же наименованием поместит в конечный/исходящий каталог /usr/jatoba-5/output/.

В результате преобразования содержание файла «jatoba\_secret\_function.sql» имеет следующий вид:

```
CREATE OR REPLACE FUNCTION jatoba_secret_function()  
RETURNS text AS $$  
CkJFR0lOIAogICAgUkVUVVJOICdzZWNYZXQgdGV4dCc7CkVORDsK$$  
LANGUAGE plpgsql;
```



```
GNU nano 4.8 jatoba_secret_function.sql  
CREATE OR REPLACE FUNCTION jatoba_secret_function()  
RETURNS text AS $$MIAGCSqGSIB3DQEHAA6CAMIACAQaxggItMIICKQIBADCCAVIwggE9MR0wGwYJK>
```

Рисунок 3.36 – Содержание преобразованного файла «jatoba\_secret\_function.sql»





Утилита `wrlpgsql` работает со вложенными структурами каталогов – реализована поддержка пакетного преобразования файлов.

### 3.7.3. Перемещение преобразованной функции в промышленный сегмент

Переместить полученный файл «`jatoba_secret_function.sql`» на эксплуатируемую БД и скомпилировать эту функцию, используя стандартную утилиту работы с БД, выполнив команду в консоли ОС:

```
psql -d postgres -h localhost -p 5432 -U postgres <
/usr/jatoba-5/output/jatoba_secret_function.sql
```

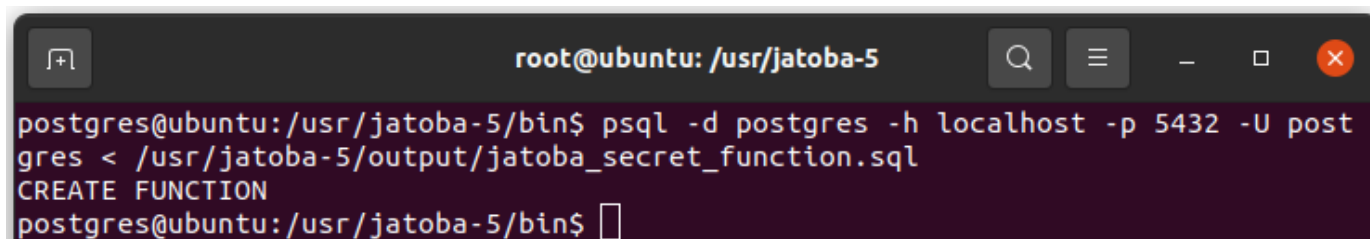


Рисунок 3.37 – Команда экспорта

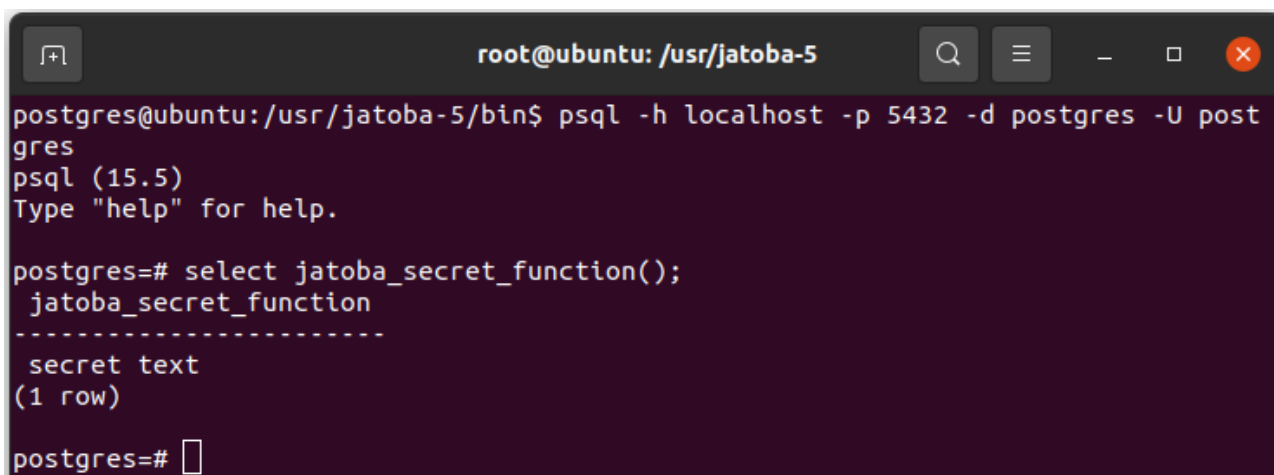


В эксплуатируемой БД для экспорта функции должно быть установлено расширение «`plpgsql`» (см. п. 3.6.3).

В результате выполнения на эксплуатируемой СУБД в БД «`postgres`» появится эквивалентная функция.

Выполнить запрос к созданной функции SQL-командой:

```
select jatoba_secret_function();
```



```
root@ubuntu: /usr/jatoba-5
postgres@ubuntu:/usr/jatoba-5/bin$ psql -h localhost -p 5432 -d postgres -U postgres
psql (15.5)
Type "help" for help.

postgres=# select jatoba_secret_function();
 jatoba_secret_function
-----
 secret text
(1 row)

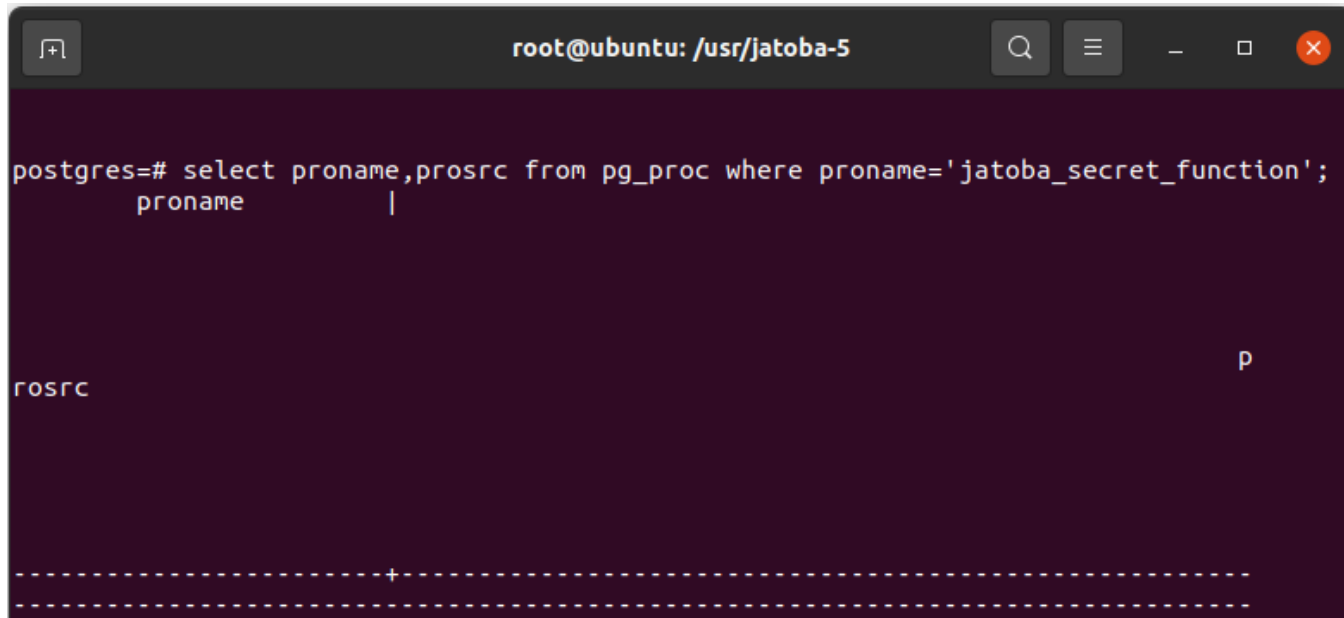
postgres=#
```

Рисунок 3.38 – Запрос функции

Результат соответствует ожидаемому, функция сохранила изначальную логику.

Проверить исходный текст, который может получить пользователь с соответствующими привилегиями выполнив SQL-команду:

```
SELECT proname,prosrc from pg_proc where
praname='jatoba_secret_function';
```



```
root@ubuntu: /usr/jatoba-5
postgres=# select proname,prosrc from pg_proc where proname='jatoba_secret_function';
 proname      |
-----+-----
 prosrc      | p
```

Рисунок 3.39 – Проверка исходного текста

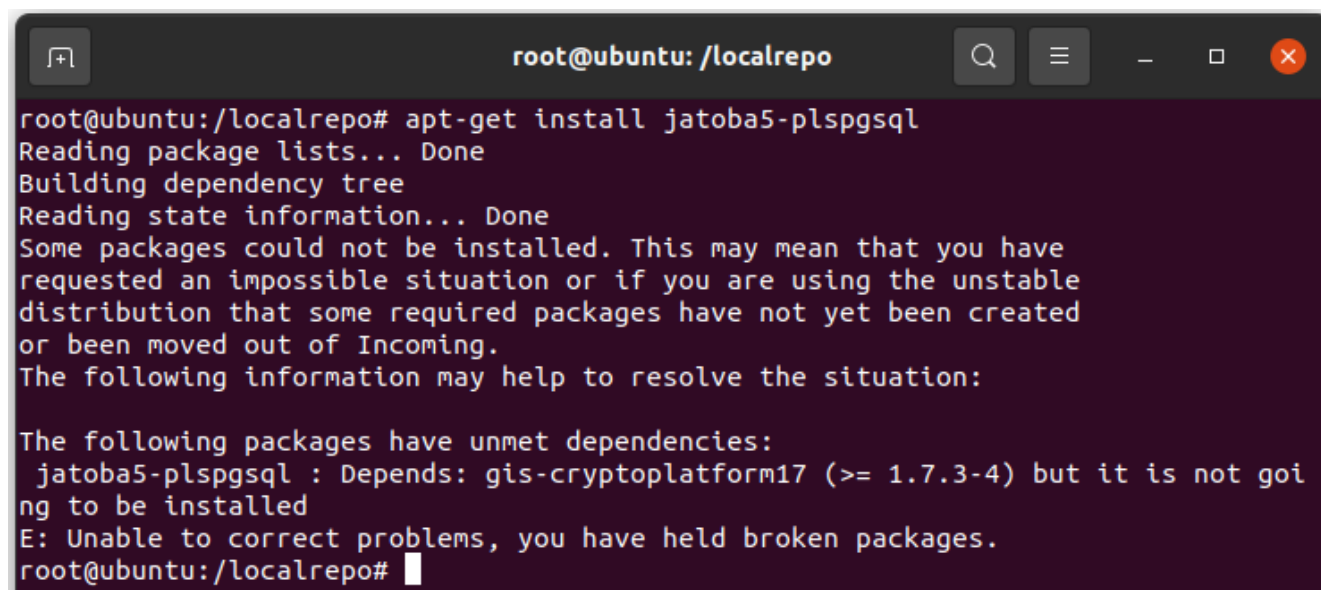
## 4. СООБЩЕНИЯ ОБ ОШИБКАХ

### 4.1. Ошибка установки пакета «jatoba<ver>-plspgsql»

При выполнении команды установки пакета компонента «PLsPgSQL»

```
apt-get install jatoba<ver>-plspgsql
```

может возникнуть ошибка установки.



```
root@ubuntu: /localrepo
root@ubuntu:/localrepo# apt-get install jatoba5-plspgsql
Reading package lists... Done
Building dependency tree
Reading state information... Done
Some packages could not be installed. This may mean that you have
requested an impossible situation or if you are using the unstable
distribution that some required packages have not yet been created
or been moved out of Incoming.
The following information may help to resolve the situation:

The following packages have unmet dependencies:
 jatoba5-plspgsql : Depends: gis-cryptoplatform17 (>= 1.7.3-4) but it is not going to be installed
E: Unable to correct problems, you have held broken packages.
root@ubuntu:/localrepo#
```

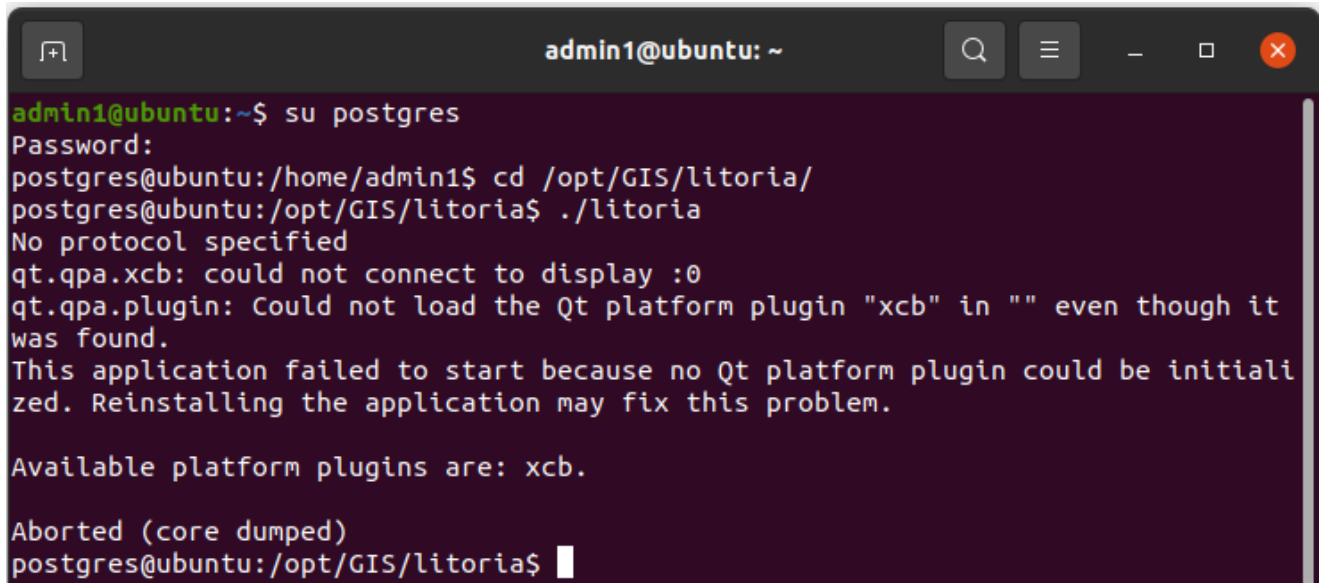
Рисунок 4.1 – Ошибка установки

Это ошибка последовательности установки компонента.

Исправить ошибку возможно установкой библиотеки «КриптоПлатформа» (gis-cryptoplatform17), описанной в п. 3.3 настоящего документа.

### 4.2. Ошибка запуска приложения «Litoria Desktop 2»

Ошибка запуска приложения «Litoria Desktop 2» возникает при попытке его запуска из терминала не от имени текущего пользователя.



```
admin1@ubuntu: ~  
admin1@ubuntu:~$ su postgres  
Password:  
postgres@ubuntu:/home/admin1$ cd /opt/GIS/litoria/  
postgres@ubuntu:/opt/GIS/litoria$ ./litoria  
No protocol specified  
qt.qpa.xcb: could not connect to display :0  
qt.qpa.plugin: Could not load the Qt platform plugin "xcb" in "" even though it  
was found.  
This application failed to start because no Qt platform plugin could be initiali  
zed. Reinstalling the application may fix this problem.  
  
Available platform plugins are: xcb.  
  
Aborted (core dumped)  
postgres@ubuntu:/opt/GIS/litoria$
```

Рисунок 4.2 – Ошибка запуска

Ошибка устраняется путем:

- выхода из текущего сеанса в ОС;
- авторизации в ОС от требуемого пользователя;

как описано в п. 3.5.2 настоящего документа.

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

SQL	–	Structured Query Language – язык структурированных запросов
БД	–	База данных
СУБД	–	Система управления базами данных
ДСЧ	–	Датчик случайных чисел

## Лист регистрации изменений

№ изменения: _____	Подпись отв. лица: _____	Дата внесения изм: _____
--------------------	--------------------------	--------------------------