Шифрование баз данных в Firebird

Александр Пешков

Firebird Foundation

2017

Регулярные встречи коммьюнити Firebird

- 2 марта 2017 года - «Все о шифровании». Скачайте материалы.
- 9 декабря 2016 года - «Предварительный обзор Firebird 4 и Отказоустойчивые решения на Firebird». Скачать материалы.
- Не хотите пропустить следующую встречу с коммьюнити? Подпишитесь на нашу рассылку (на любой странице www.ibase.ru, в окошке-слайдере справа).

Шифрование баз данных в Firebird

- История вопроса
  - Существовала (закрытая #ifdef) с IB 6.01.
    - Не было поддержки шифрации ранее незашифрованной БД
    - Передача секретного (!!!) ключа шифрации с клиента в DPB
  - Заново реализована в FB 3
    - Шифрование / дешифрация БД средствами SQL
    - Шифрация «на лету» в выделенном служебном потоке
    - Управление ключами с помощью специального плагина (тип KeyHolder)

Шифрование баз данных в Firebird

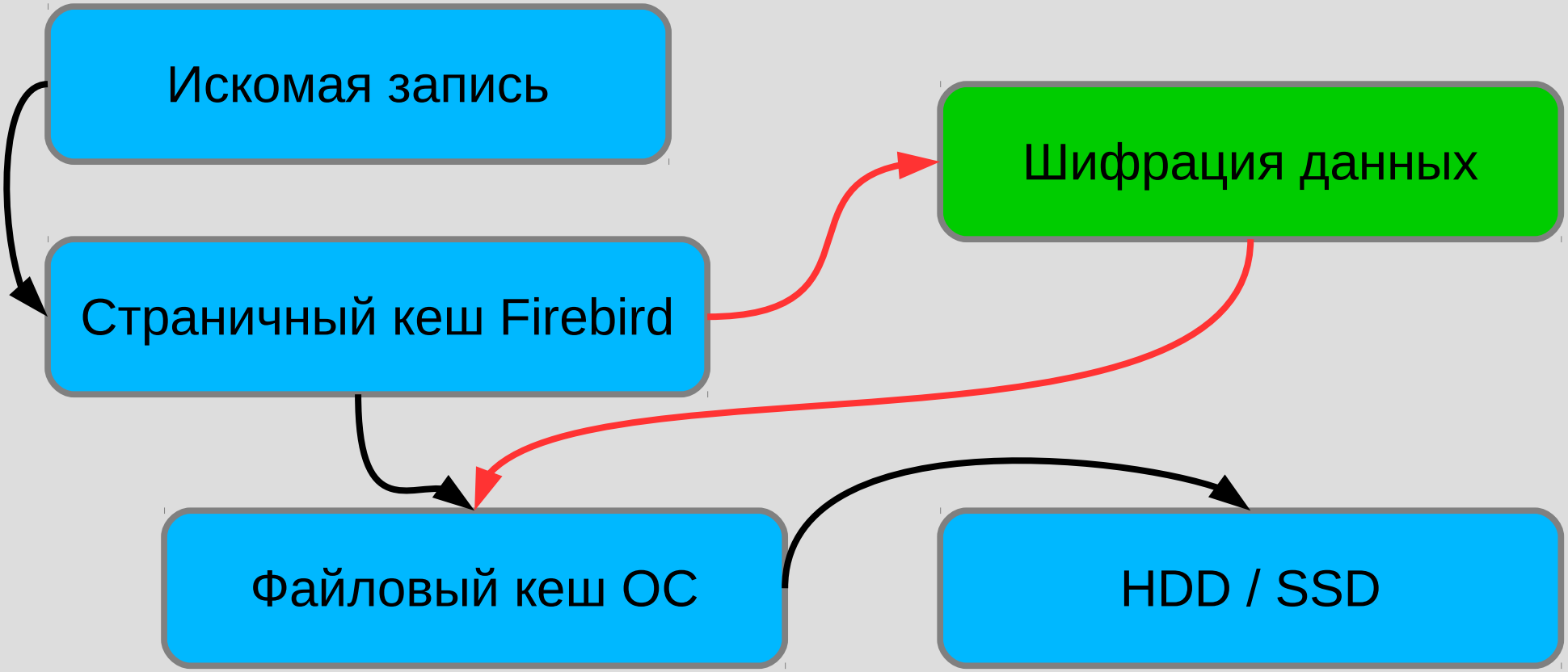
- В сравнении с размещением файла базы данных на зашифрованном диске
  - Поддержка шифрации баз данных, предназначенных для передачи для дальнейшего использования другому лицу
  - Не требует offline-периода для шифрации (дешифрации) БД

Шифрование баз данных в Firebird

- Что и как шифруется
  - Страницы данных, blob и индексов (кроме заголовка страницы) — зашифрованы, включая системные таблицы
  - Вспомогательные страницы (PIP, TIP и тп.) - незашифрованы
  - Проверка корректности ключа производится с помощью сравнения хеша фиксированных данных с образцом хранящимся в заголовке
  - Особо важные данные (контрольный хеш, флаги шифрации и тп.) защищены дополнительной зашифрованной контрольной суммой

Шифрование баз данных в Firebird

- На каком этапе происходит шифрация



Шифрование баз данных в Firebird

- Не рекомендуется:
  - Защита файла БД от копирования с сервера (по сети)
- Решение:
  - Настройка прав доступа к файлам на сервере

  - Share \\server\c - Full Control
  - Все работают с логином Administrator (или члены группы с подобными правами)

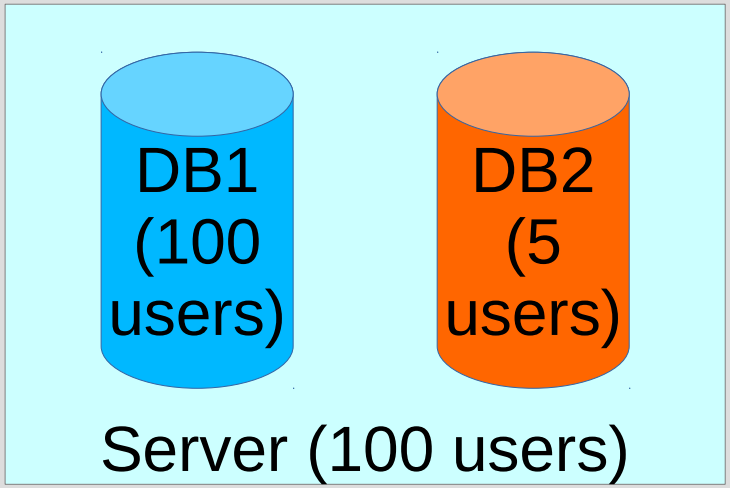
Шифрование баз данных в Firebird

- Не рекомендуется:
  - Защита файла БД от копирования с сервера (по сети)
- Решение:
  - Настройка прав доступа к файлам на сервере

  - В FB 2.1 Windows trusted authentication превращала всех членов группы Domain Admins в SYSDBA
  - FB 2.5 добавлена группа RDB$ADMIN и контроль за её назначением Domain Admins (по многочисленным требованиям пользователей с некорректной настройкой прав доступа в ОС)

Шифрование баз данных в Firebird

- Не рекомендуется:
  - Ограничить доступ к конкретной БД
- Решение FB3:
  - Использовать отдельную security database
- Pre-FB3 (шифрация?)



Шифрование баз данных в Firebird

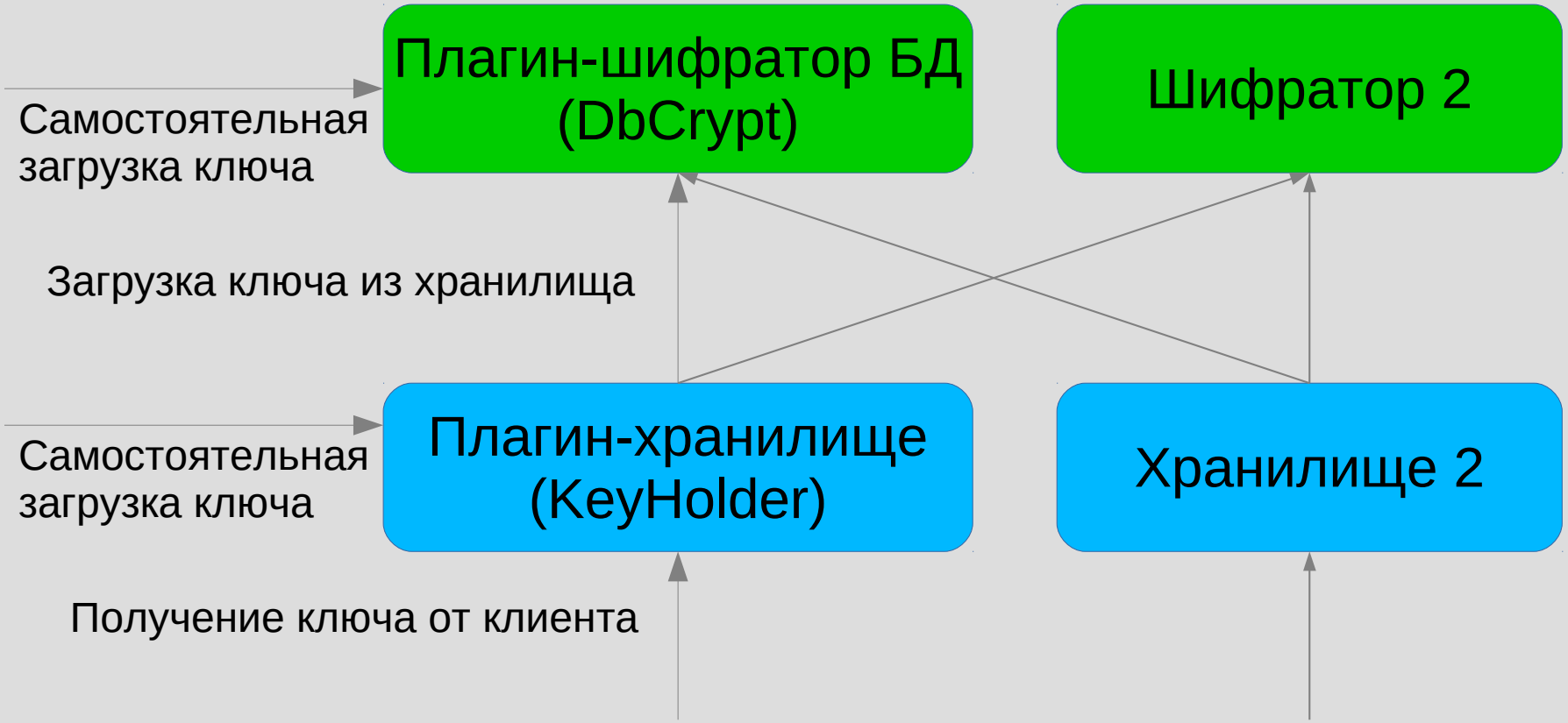
- Рекомендуемое использование
- Защита БД, предназначенных на продажу
  - Пред-заполненных существенными данными (справочные системы и тп)
  - Имеющих нетривиальную бизнес-логику в метаданных
- Защита БД от физического похищения (диск или даже сервер целиком)

Шифрование баз данных в Firebird

- Где может храниться ключ шифрации
- БД распространяемые за плату
  - В специализированном клиентском ПО
    - Доступ к БД возможен только из этого ПО
  - Поддержка “режима разработчиков” - неограниченный доступ к ключам и БД
- БД защищённые от физической кражи
  - В любом надёжном месте (отдел безопасности)
    - Доступ к БД возможен с любого клиента, включая универсальные средства типа утилит firebird

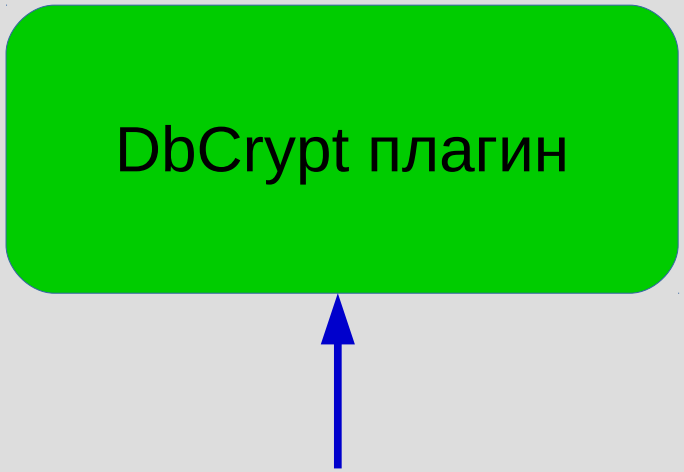
Шифрование баз данных в Firebird

- Как ключ попадает в плагин ?



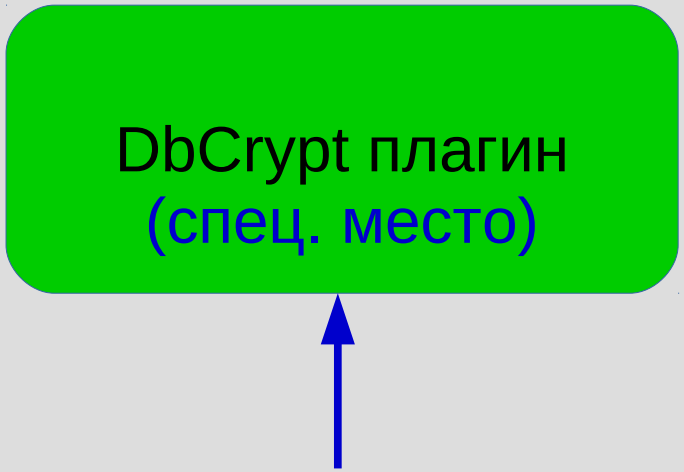
Шифрование баз данных в Firebird

- Возможные источники ключей



Шифрование баз данных в Firebird

- Возможные источники ключей



Шифрование баз данных в Firebird

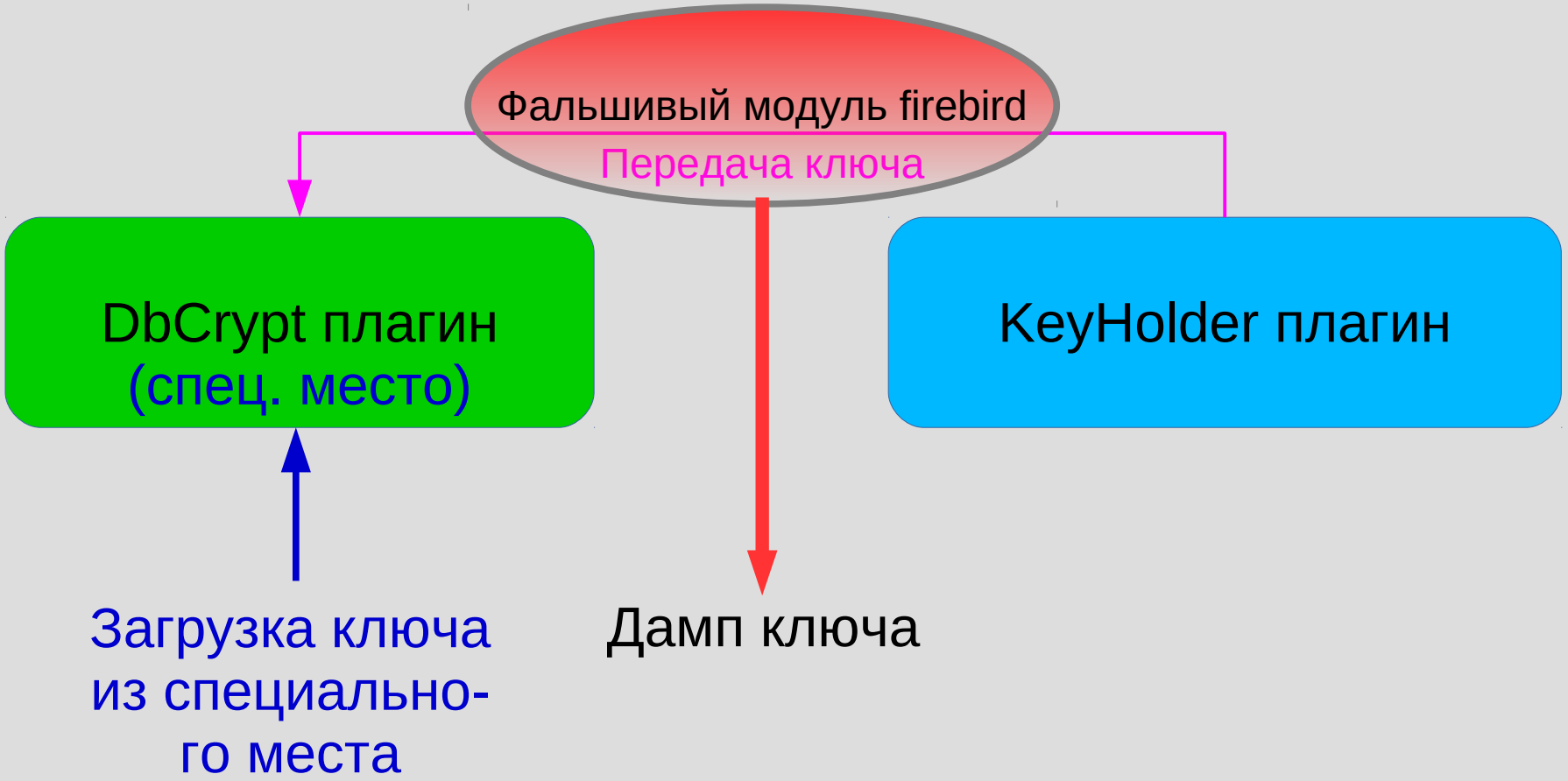
- Возможные источники ключей

Шифрование баз данных в Firebird

- Возможные источники ключей

Шифрование баз данных в Firebird

- Возможные источники ключей



Шифрование баз данных в Firebird

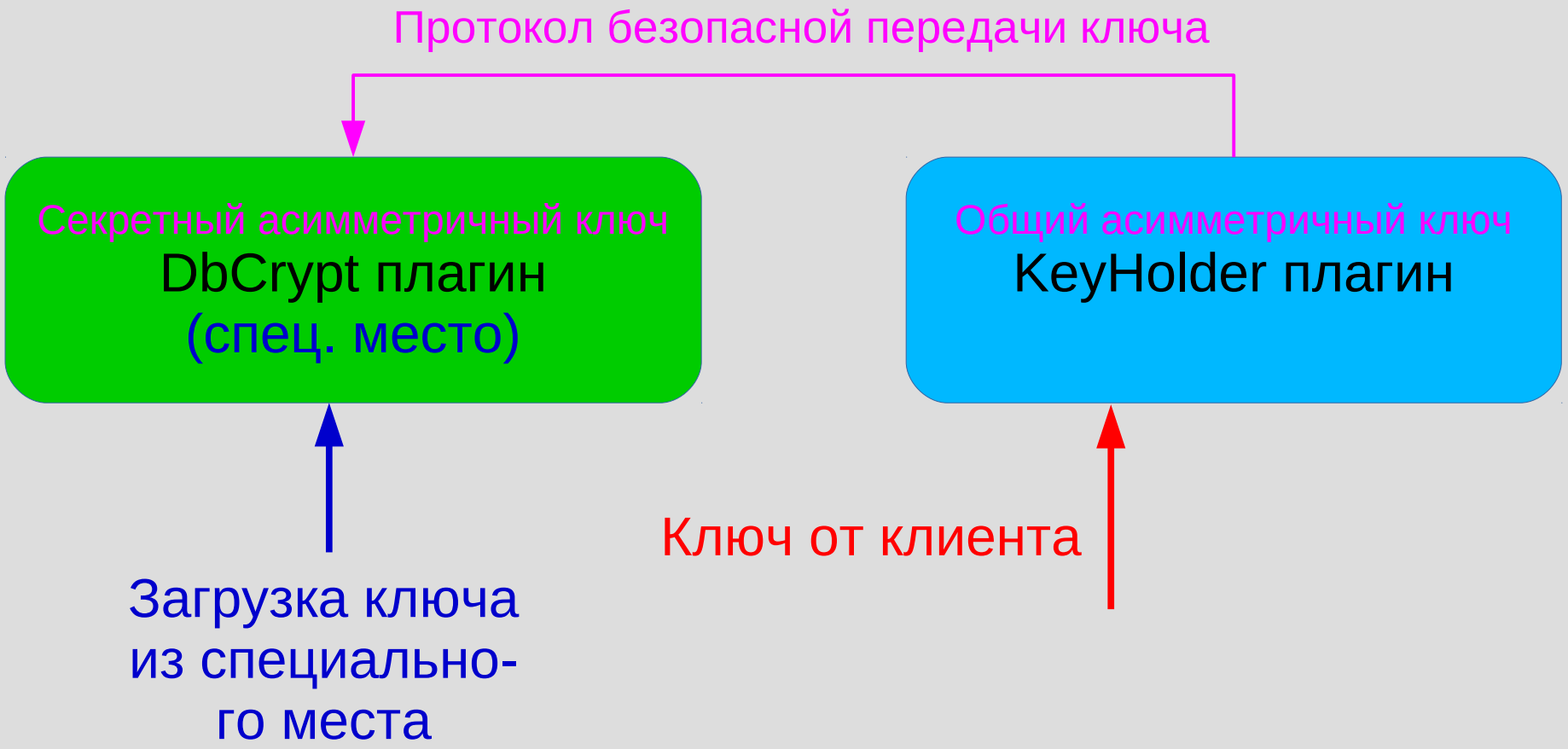
- Упрощённый протокол безопасной передачи ключа
  - DbCrypt плагин => KeyHolder плагин:
    - Прошу передать мне ключ
  - KeyHolder плагин:
    - Шифрует ключ (с учётом данных от DbCrypt)
  - KeyHolder плагин => DbCrypt плагин:
    - Зашифрованный ключ
  - DbCrypt плагин:
    - Дешифрует и проверяет ключ
    - Готов к работе

Шифрование баз данных в Firebird

- Возможные источники ключей

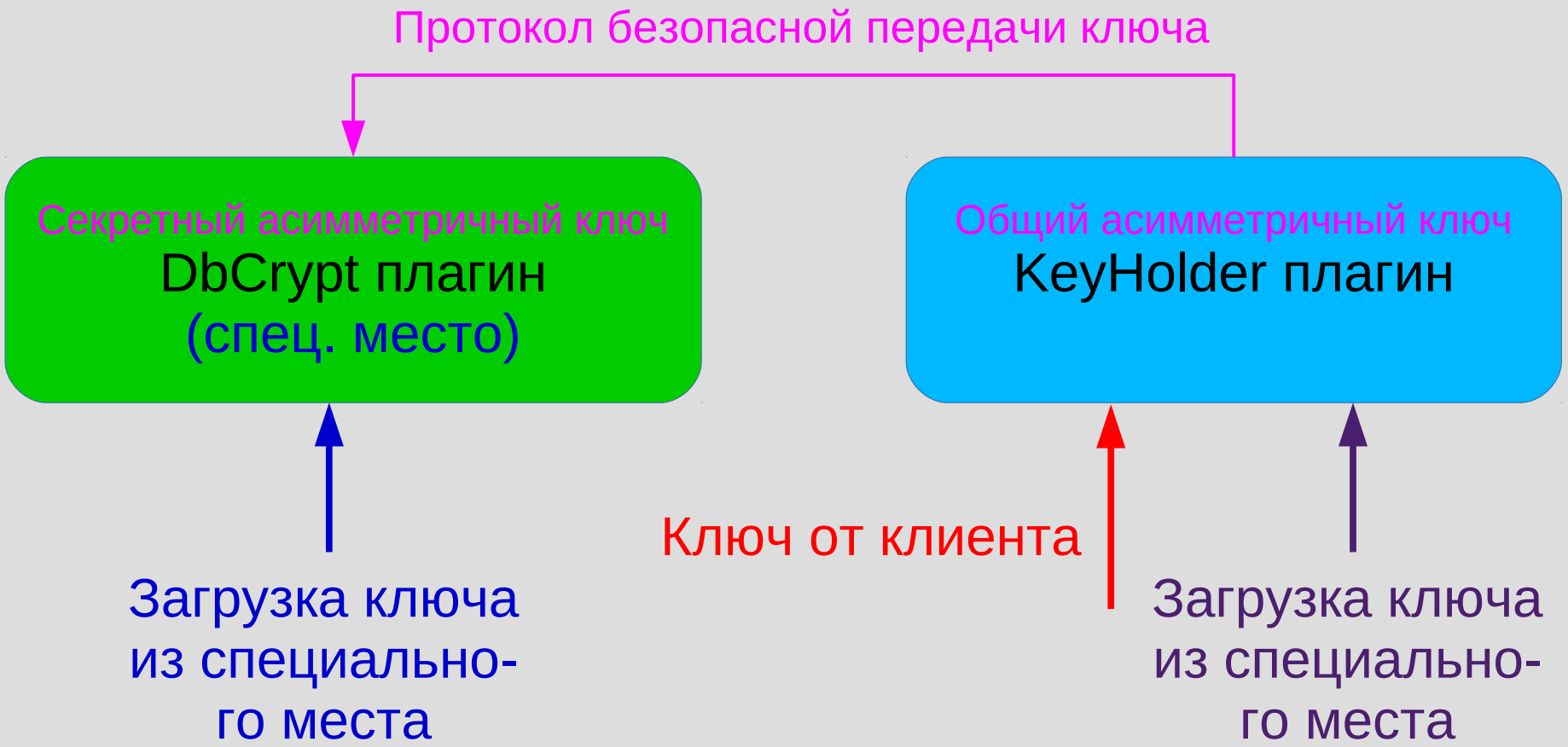
Шифрование баз данных в Firebird

- Возможные источники ключей



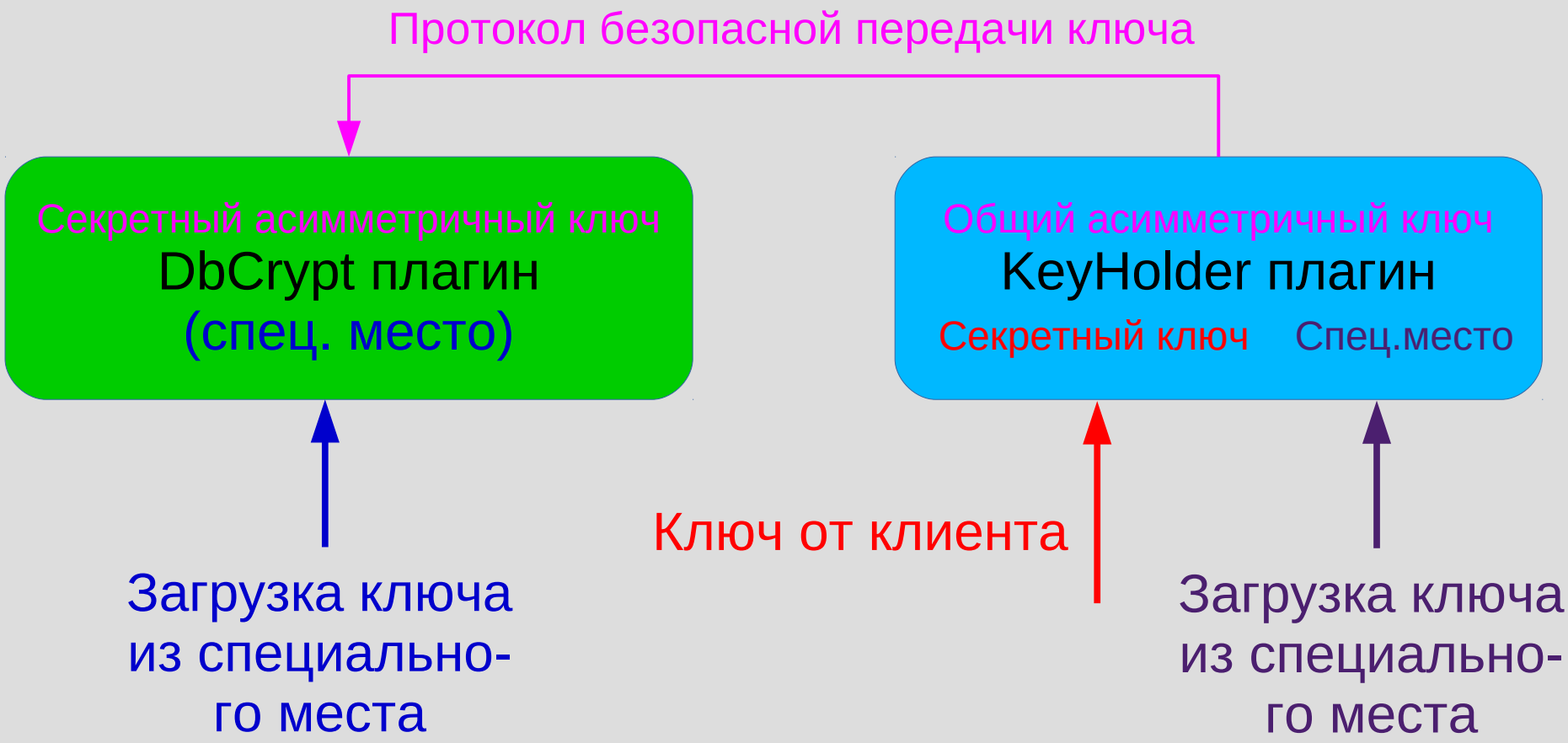
Шифрование баз данных в Firebird

- Возможные источники ключей



Шифрование баз данных в Firebird

- Возможные источники ключей



Шифрование баз данных в Firebird

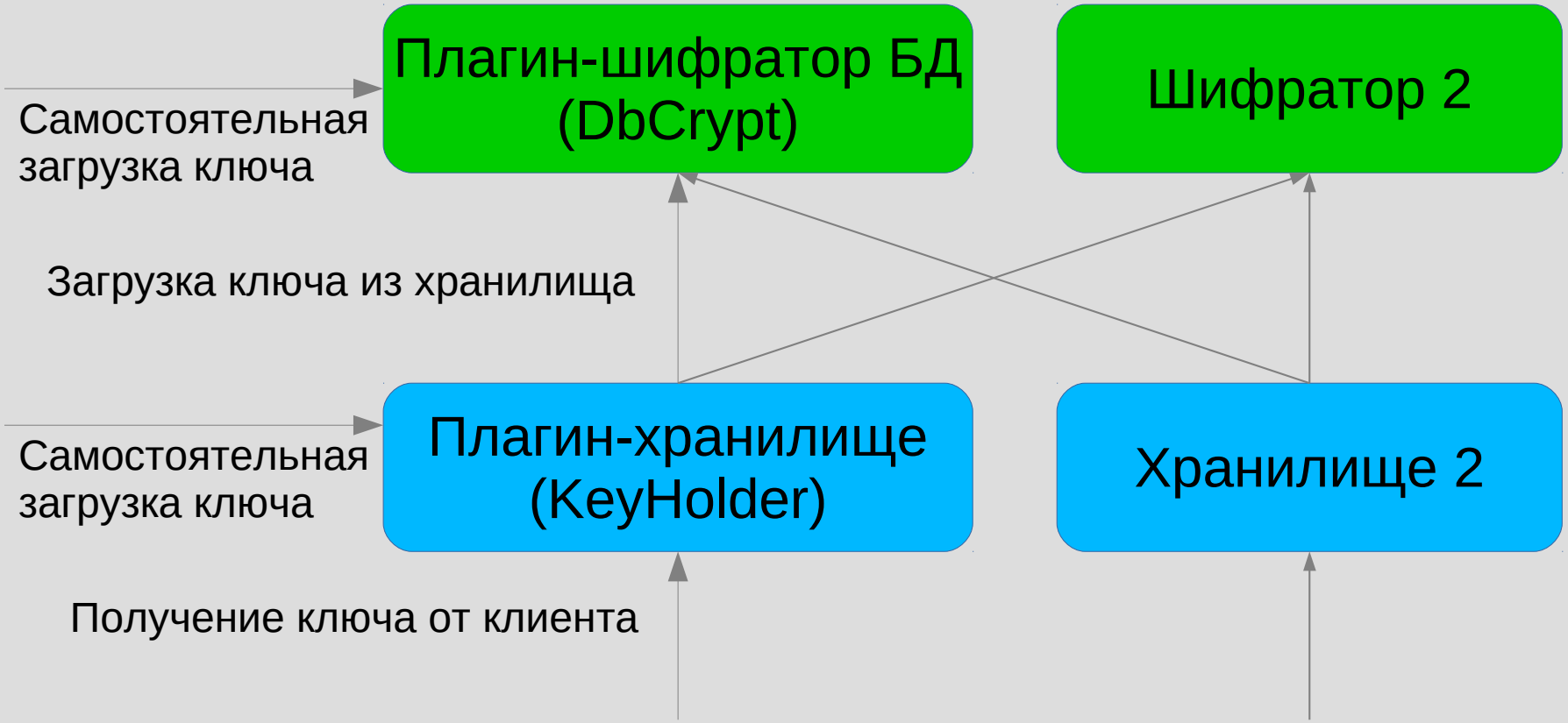
- Написание плагинов
- DbCrypt:

```
void setInfo(Status status,
      DbCryptInfo info); // вспомогательная информация
void setKey(Status status,
      uint length, KeyHolder sources[ ],
      string keyName);  // инициализация

void encrypt(Status status,  // основные вызовы
      uint length, const void* from, void* to);
void decrypt(Status status,
      uint length, const void* from, void* to);
```

Шифрование баз данных в Firebird

- Как ключ попадает в плагин ?



Шифрование баз данных в Firebird

- Написание плагинов
- DbCryptInfo: // информационный интерфейс
  string getDatabaseFullPath(Status status);

Шифрование баз данных в Firebird

- Написание плагинов
- KeyHolder:

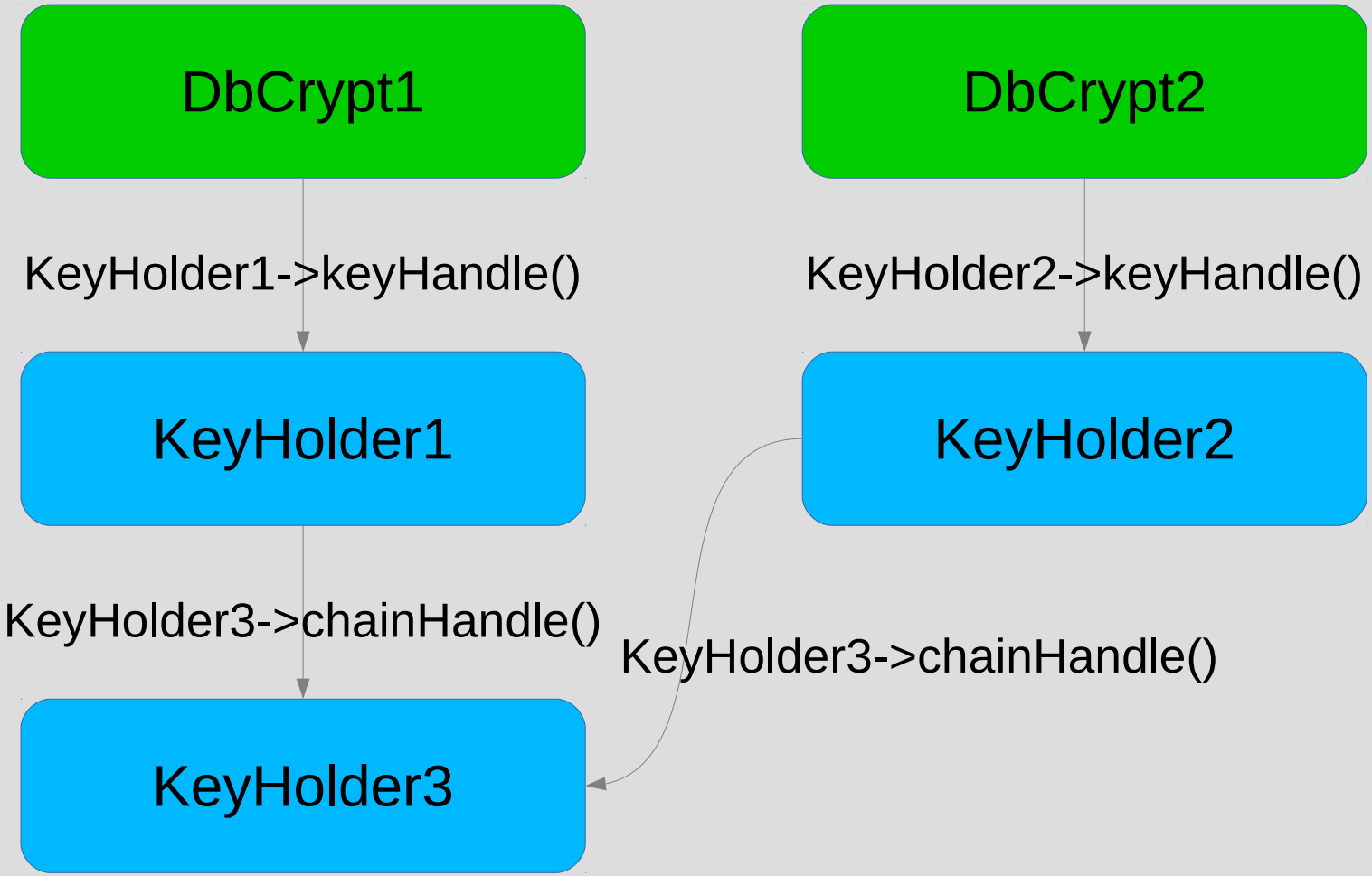
```
int keyCallback(Status status,
      CryptKeyCallback callback);  // инициализация

CryptKeyCallback keyHandle(Status status,
      string keyName);         // запрос от DbCrypt
CryptKeyCallback chainHandle(Status status);
                  // запрос от следующего KeyHolder в цепочке

boolean useOnlyOwnKeys(Status status);
```

Шифрование баз данных в Firebird

- Как ключ попадает в плагин ?



Шифрование баз данных в Firebird

- Написание плагинов
- CryptKeyCallback:

```
uint callback(uint dataLength, const void* data,
        uint bufferLength, void* buffer);
```

```
CryptKeyCallback cb = keyHolder->keyHandle(….);
if (callback)
    answerLen = cb->callback(requestLen, request,
        bufferLength, buffer);
    …
    answerLen = cb->callback(requestLen2, request2,
        bufferLength, buffer);
```

Шифрование баз данных в Firebird

- Написание плагинов
- KeyHolder:

```
int keyCallback(Status status,
        CryptKeyCallback callback);  // инициализация

CryptKeyCallback keyHandle(Status status,
        string keyName);              // запрос от DbCrypt
CryptKeyCallback chainHandle(Status status);
                    // запрос от следующего KeyHolder в цепочке

boolean useOnlyOwnKeys(Status status);
```

Classic/SuperClassic — не влияет
Super — принудительный контроль корректности ключа для каждого клиента базы данных

Шифрование баз данных в Firebird

- Шаг 1 – выбор плагина
  - Нет плагинов с открытым кодом – проблема с управлением ключами
    - Самостоятельное написание плагина
    - Использование готового плагина с закрытым кодом

Шифрование баз данных в Firebird: защита от кражи

- Шаг 2 – проверка работоспособности плагина на копии базы данных

  - SQL оператор:

    Alter database encrypt with “PluginName”
  - Или:

    Alter database encrypt with “PluginName” key “Name”

  Использование и назначение имени ключа полностью зависит от плагина

Шифрование баз данных в Firebird: защита от кражи

- Шаг 3 – резервное копирование

  Все данные будут перезаписаны в зашифрованном виде!!!

Шифрование баз данных в Firebird: защита от кражи

- Шаг 4 – выбираем период минимальной нагрузки и шифруем базу данных

  - Не производить копирование во время шифрации!
  - Используем таблицы мониторинга чтобы следить сколько осталось шифровать:

```
SQL: select mon$crypt_page * 100.0 / mon$pages as
  Percent from mon$database

gstat -e db_name
```

Шифрование баз данных в Firebird: передача БД

- Шаг 2 – интеграция специализированного клиентского ПО с выбранным плагином
  - Реализация интерфейса CryptKeyCallback
  - Регистрация его в fbclient:
    - OO API: Provider::setDbCryptCallback(status, cb)
    - fb_database_crypt_callback(status_vector, cb)

Шифрование баз данных в Firebird: передача БД

- Шаг 3 – проверка работоспособности программного комплекса в целом

Использование ваших обычных методов тестирования, но с зашифрованной БД

Шифрование баз данных в Firebird: передача БД

- Шаг 4 (при необходимости) – проверка обновления базы у клиента с шифрацией

  При установке версии специализированного ПО у клиента должна быть произведена шифрация базы данных

Шифрование баз данных в Firebird

- Ограничения при работе с зашифрованной базой данных

  - API — работает полностью, есть особенности в работе с security db
  - Утилиты firebird - работают все кроме gstat, есть особенности резервного копирования
    - В gstat работают только 2 ключа – -h / -e (статистика количества зашифрованных/нешифрованных страниц)

Шифрование баз данных в Firebird

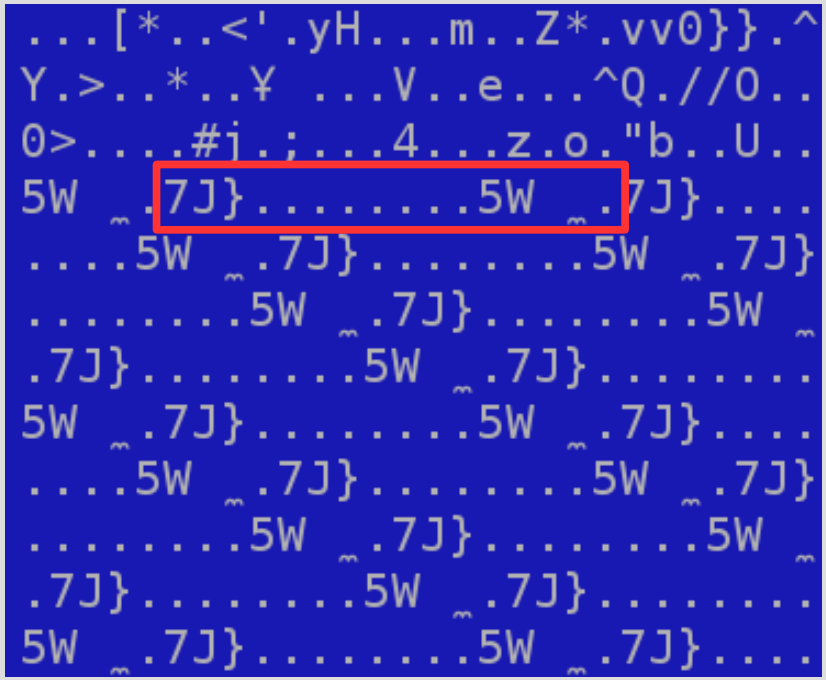
- Ограничения при работе с зашифрованной базой данных
  - Резервное копирование
    - gbak: необходима шифрация копии БД (file.gbak) вручную
    - nbackup: после шифрации следует снять полную (уровень 0) копию

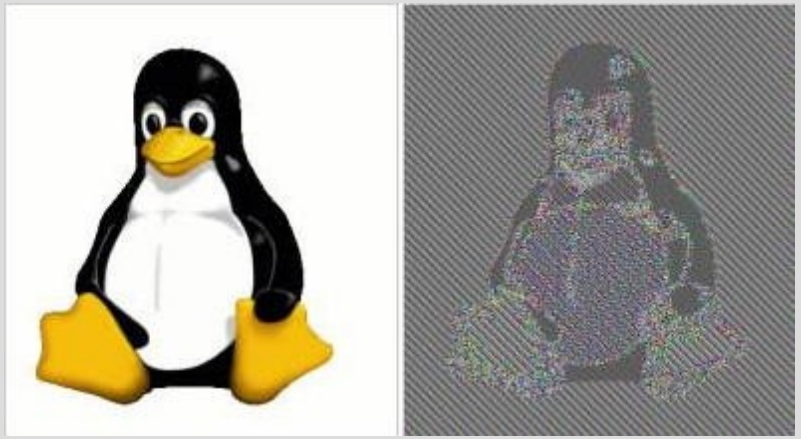
Шифрование баз данных в Firebird

- Ограничения при работе с зашифрованной базой данных
  - Работа с зашифрованной security db
    - Размещение списка пользователей в одной базе вместе с данными (self security db)
    - Специальный параметр конфигурации (CryptSecurityDatabase)
    - Повышенные требования к шифрации ключа передаваемого от клиента

Шифрование баз данных в Firebird

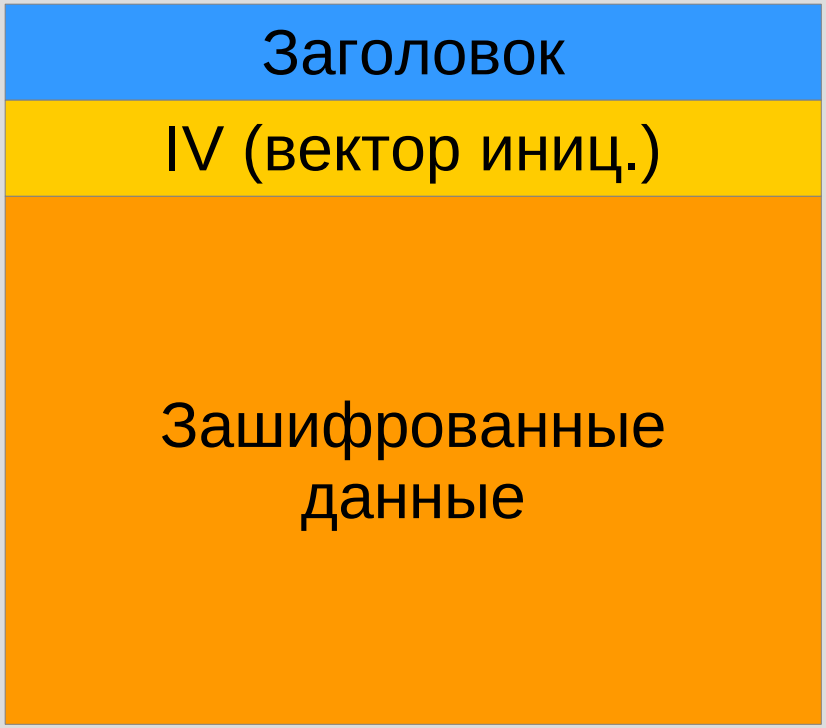
- Особенность некоторых алгоритмов
  - Размер шифрованной страницы == начальному
  - Использование режима ECB (напр. в AES)
  - Заметные на глаз повторы на пустых страницах





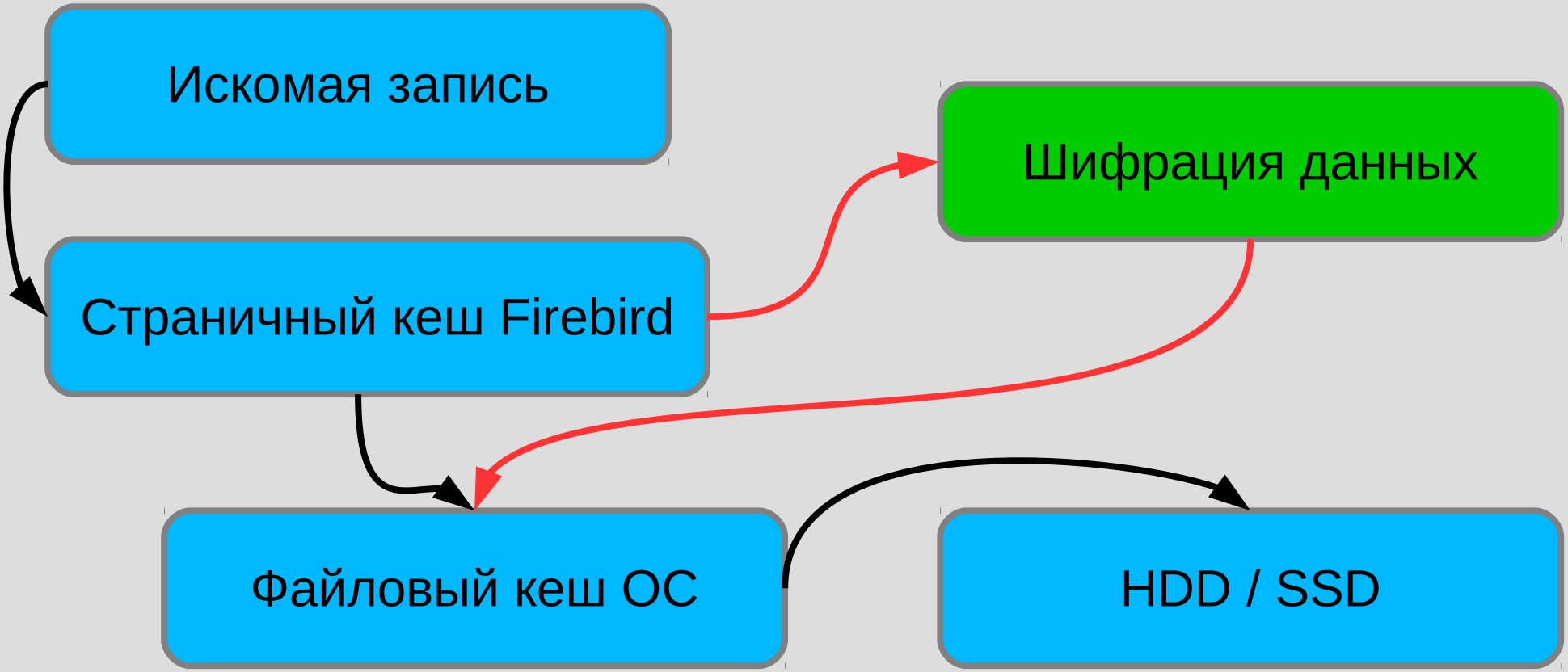
Шифрование баз данных в Firebird

- Возможные решения
  - Использование других алгоритмов
  - Резервирование места на странице для IV на этапе создания БД (будущие версии FB)



Шифрование баз данных в Firebird

- Производительность



Шифрование баз данных в Firebird

- Производительность

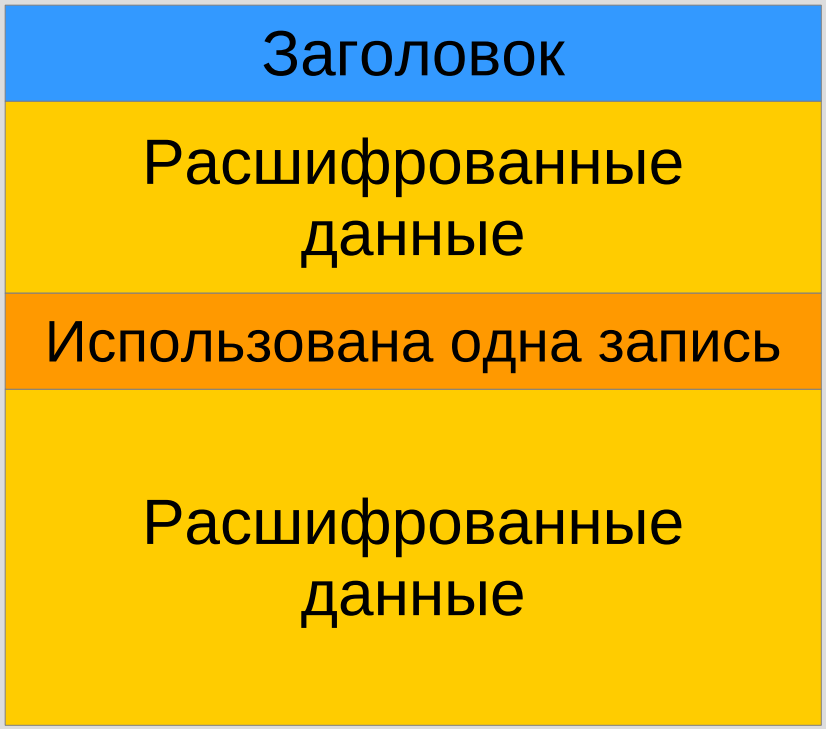
  Время выборки записи равно:
  - время поиска записи в кеше (плюс - если не найдена)
  - время чтения страницы из кеша файловой системы и дешифрации (плюс - если не найдена)
  - время чтения страницы с диска

Шифрование баз данных в Firebird

- Производительность
- Сравнение выборки по индексу и полной

Последовательная выборка

Выборка по индексу



Шифрование баз данных в Firebird

- Производительность (рабочая станция)
  - 8 ядер CPU (AMD FX-8120)
  - RAM 8 Gb
  - SATA
  - 4 коннекта, 1 минута TPC-C
  - AES, реализация OpenSSL
  - Кеш по умолчанию 16 Mb (< DB size)

(tpmC, TPC-C Throughput)

Forced writes	Обычный	Шифрация	Потери %%
On	984	740	25%
Off	27062	18453	32%

Шифрование баз данных в Firebird

- Производительность (рабочая станция)
  - 8 ядер CPU (AMD FX-8120)
  - RAM 8 Gb
  - SATA
  - 4 коннекта, 1 минута TPC-C
  - AES, реализация OpenSSL
  - Увеличенный кеш 320 Mb (> DB size)

(tpmC, TPC-C Throughput)

Forced writes	Обычный	Шифрация	Потери %%
On	1036	882	15%
Off	27793	19170	31%

Шифрование баз данных в Firebird

- Производительность (сервер)
  - 24 (12 with HT) ядра CPU
  - RAM 32 Gb
  - SSD
  - 100 коннектов, 90 минут
  - AES, реализация OpenSSL
  - Размер кеша 6Gb (> DB size)

(operations / minute)

Forced writes	Not encrypted	Encrypted	Performance loss
On	4491	4152	8%
Off	4346	4183	4%

Шифрование баз данных в Firebird

- Производительность начальной шифрации базы даных (рабочая станция)
  - Монопольная работа с БД
  - Кеш по умолчанию

Страниц (8Kb) в секунду

Forced writes	Шифрация
On	3964
Off	6378

Вопросы?