|  |  |  |
| --- | --- | --- |
|  |  | **УТВЕРЖДАЮ**  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  …….. |
|  |  | ……… |
|  |  | «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

**Система криптографической защиты информации в базах данных**

**Подсистема криптографической защиты данных**

**в**

**Описание программного обеспечения**

**СКЗИ «Крипто БД»**

СОДЕРЖАНИЕ

[1 Перечень принятых сокращений 3](#_Toc28001295)

[2 Общие сведения 4](#_Toc28001296)

[2.1 Полное наименование системы и ее условное обозначение 4](#_Toc28001297)

[2.2 Шифр темы или шифр (номер) договора 4](#_Toc28001298)

[2.3 Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты 4](#_Toc28001299)

[2.4 Перечень документов, на основании которых создается система, кем и когда утверждены эти документы 4](#_Toc28001300)

[2.5 Плановые сроки начала и окончания работы по созданию системы 4](#_Toc28001301)

[2.6 Сведения об источниках и порядке финансирования работ 4](#_Toc28001302)

[2.7 Порядок оформления и предъявления заказчику результатов работ по созданию системы (ее частей), по изготовлению и наладке отдельных средств (технических, программных, информационных) и программно-технических (программно-методических) комплексов системы 4](#_Toc28001303)

[3 Назначение и цели создания системы 6](#_Toc28001304)

[3.1 Назначение системы 6](#_Toc28001305)

[4 Условия применения 8](#_Toc28001306)

[4.1 Требования к аппаратному обеспечению 8](#_Toc28001307)

[4.2 Требования к программному обеспечению 8](#_Toc28001308)

[4.3 Ограничения 9](#_Toc28001309)

[5 Основные характеристики 10](#_Toc28001310)

[5.1 Пригодность 10](#_Toc28001311)

[5.2 Способность к взаимодействию 10](#_Toc28001312)

[5.3 Защищенность 10](#_Toc28001313)

[1.1. Согласованность 12](#_Toc28001314)

[1.2. Надежность 12](#_Toc28001315)

[5.4 Практичность 13](#_Toc28001316)

[6 Общие сведения 14](#_Toc28001317)

[6.1 Язык программирования 14](#_Toc28001318)

[6.2 Управление конфигурацией 14](#_Toc28001319)

[6.3 Реализованные алгоритмы 14](#_Toc28001320)

[6.4 Использованное внешнее ПО 15](#_Toc28001321)

[6.5 Способ создания дистрибутива 15](#_Toc28001322)

[6.6 Сборка дистрибутива 15](#_Toc28001323)

[7 Структура Крипто БД 16](#_Toc28001324)

[7.1 Структура Крипто БД и связь со средой 16](#_Toc28001325)

[7.2 Описание элементов 16](#_Toc28001326)

[8 Сервер Крипто БД 19](#_Toc28001327)

[8.1 Состав 19](#_Toc28001328)

[8.2 Состояния сервера Крипто БД 19](#_Toc28001329)

[1.3. Включение сервера Крипто БД 20](#_Toc28001330)

[1.4. Запуск сервера Крипто БД 21](#_Toc28001331)

[1.5. Вычисления КШД 22](#_Toc28001332)

[9 Репозиторий 24](#_Toc28001333)

[9.1 Объекты БД 24](#_Toc28001334)

[9.2 Схема владельца 24](#_Toc28001335)

[9.3 Создание объектов системы на сервере базы данных. 25](#_Toc28001336)

[9.4 Подготовка и создание пакетов и триггеров. 25](#_Toc28001337)

[9.5 Установка начальных данных. 26](#_Toc28001338)

[10 Выполнение функций 28](#_Toc28001339)

[10.1 Шифрование данных 28](#_Toc28001340)

[10.2 Преобразование таблицы 28](#_Toc28001341)

[10.3 Завершение преобразования таблицы. 38](#_Toc28001342)

[10.4 Исходные данные для расшифрования 38](#_Toc28001343)

[10.5 Шаги по преобразованию целевой таблицы. Расшифрование. 39](#_Toc28001344)

[1.6. Маскирование 41](#_Toc28001345)

[10.6 Описание решения 41](#_Toc28001346)

[10.7 Интерфейс маскирующих функций. 41](#_Toc28001347)

[10.8 Функция генерации маскирующих значений для типов NUMBER. 42](#_Toc28001348)

[10.9 Функция генерации маскирующих значений для типов DATE. 42](#_Toc28001349)

[10.10 Аудит 43](#_Toc28001350)

[11 Взаимодействие сервера Крипто БД и клиентского ПО 57](#_Toc28001351)

[11.1 Вызов и загрузка 57](#_Toc28001352)

[11.2 Входные данные 59](#_Toc28001353)

[11.3 Выходные данные 60](#_Toc28001354)

1. Перечень принятых сокращений

| Сокращение или обозначение | Расшифровка |
| --- | --- |
| BIOS | Базовая система ввода-вывода (Base Input-Output System) |
| ITU-T | Международный комитет по телекоммуникациям (International Telecommunication Union) |
| IETF | Специальная комиссия интернет разработок (Internet Engineering Task Force) |
| UDP | User Datagram Protocol (сетевой протокол) |
| XML | eXtended Markup Language |
| АС | Автоматизированная система |
| АРМ | Автоматизированное рабочее место |
| БД | База данных |
| ДСЧ | Датчик случайных чисел |
| КШД | Ключ шифрования данных |
| ЛВС | Локкальная вычислительная сеть |
| HDD, НЖМД | Накопитель на жёстких магнитных дисках |
| НСД | Несанкционированный доступ |
| ОС | Операционная система |
| ПАК | Программно-аппаратный комплекс |
| ПО | Программное обеспечение |
| ПЭВМ | Персональная электронная вычислительная машина |
| СВТ | Средства вычислительной техники |
| СКЗИ | Средство криптографической защиты информации |
| СОС (CRL) | Список отозванных сертификатов (Certificate Revocation List) |
| СУБД | Система управления базой данных |
| УЦ | Удостоверяющий Центр |
| ЭЦП | Электронная цифровая подпись |

1. Общие сведения
   1. Полное наименование системы и ее условное обозначение

**Полное наименование системы:** Система анализа и криптографической защиты информации в базах данных .

**Условное обозначение системы**: СКЗИ "Крипто БД".

* 1. Шифр темы или шифр (номер) договора

<>

* 1. Наименование предприятий (объединений) разработчика и заказчика (пользователя) системы и их реквизиты

**Заказчик:** .

**Исполнитель:** Закрытое акционерное общество «Аладдин Р.Д.» (ЗАО «Аладдин Р.Д.»).

* 1. Перечень документов, на основании которых создается система, кем и когда утверждены эти документы

1. Договор <>
2. Федеральный закон № 152-ФЗ от 27.07.06 г. "О персональных данных".
   1. Плановые сроки начала и окончания работы по созданию системы

Сроки выполнения работ определяются Договором <> на оказание услуг по созданию системы криптографической защиты информации в базах данных .

* 1. Сведения об источниках и порядке финансирования работ

Финансирование работ осуществляет Заказчик.

* 1. Порядок оформления и предъявления заказчику результатов работ по созданию системы (ее частей), по изготовлению и наладке отдельных средств (технических, программных, информационных) и программно-технических (программно-методических) комплексов системы

Результаты работ передаются заказчику в следующем составе:

* протоколы тестовых испытаний;
* ПАК модуль принятия решений;
* техническая документация;
* эксплуатационная документация.

1. Назначение и цели создания системы
   1. Назначение системы

Программный комплекс СКЗИ «Крипто БД» предназначен для обеспечения конфиденциальности и контроля целостности информации, хранящейся в таблицах баз данных СУБД Oracle посредством криптографического преобразования и имитозащиты.

Крипто БД обеспечивает:

* + идентификацию и аутентификацию пользователей:
  + на рабочей станции администратора безопасности – с использованием ключа eToken;
  + в БД Oracle – по цифровым сертификатам с использованием SSL протокола;
  + защиту данных пользователей путем шифрования таблиц/колонок БД;
  + установку и изменение настроек Крипто БД;
  + управление пользователями Крипто БД и БД Oracle и их ключами шифрования:
  + регистрацию (добавление) пользователей и их ключей шифрования,
  + удаление ключей шифрования пользователей,
  + передачу ключей шифрования в защищенную сессию Oracle;
  + хранение служебной информации:
  + алгоритмы шифрования,
  + ключи шифрования,
  + список таблиц и колонок, защищенных системой,
  + настройки аудита/мониторинга;
  + просмотр/изменение служебной информации только администратору безопасности;
  + ограничение доступа к защищенным данным в БД Oracle с помощью ключей eToken;
  + контроль целостности ПО и служебной информации Крипто БД;
  + аудит безопасности:
  + регистрацию событий Крипто БД в журнале событий,
  + просмотр собранных данных аудита.
  + защиту остаточной информации.

Крипто БД включает:

* + библиотеки криптографии – пакеты хранимых процедур, устанавливаемые на Oracle Database Server;
  + сервер Крипто БД – пакет хранимых процедур, устанавливаемый на Oracle Database Server;
  + Средства администрирования, устанавливаемые на рабочей станции администратора безопасности.
  + Библиотеки криптографии включают следующие библиотеки:
  + реализующие алгоритмы шифрования;
  + для работы с ключами шифрования;
  + администратора (контроля целостности, преобразования таблиц, прочие);
  + API пользователя.

Библиотеки криптографии обеспечивают:

* + вызов модулей, реализующих алгоритмы шифрования,
  + регистрацию событий доступа к защищенным данным и формирование событий аудита.

Сервер Крипто БД включает сервис вычисления ключей шифрования и сервис аудита.

Сервер Крипто БД обеспечивает:

* + вычисление ключей шифрования;
  + получение и передачу ключей шифрования в сессии приложения;
  + установку параметров аудита в сессии приложения.

Средства администрирования включают консоль АДМБ и сценарии работы с БД.

Консоль АДМБ обеспечивает:

* + управление ключами шифрования,
  + аудитом,
  + процессами расшифрования/зашифрования информации,
  + контроль целостности,
  + восстановление объектов базы данных.

1. Условия применения
   1. Требования к аппаратному обеспечению

Техническая архитектура для работы Крипто БД с СУБД Oracle включает компьютеры на платформе 32/64 бит:

* + сервер БД;
  + автоматизированное рабочее место (АРМ) администратора безопасности.

Конфигурация аппаратных средств АРМ администратора безопасности должна удовлетворять требованиям, изложенным в документации на операционную систему и другое устанавливаемое ПО.

В зависимости от количества объектов, целостность которых будет контролироваться Крипто БД, для хранения списка объектов требуется от 1 до 50 Мбайт дискового пространства.

Дополнительно на АРМ администратора безопасности могут быть установлены средство защиты от несанкционированного доступа (НСД) «Аккорд-АМДЗ» либо электронный замок «Соболь».

Для работы Крипто БД необходимы идентифицирующие устройства, в качестве которых используются USB-ключи или смарт-карты eToken из состава программно-аппаратного комплекса (ПАК) Электронный ключ eToken 5.

Применяемые модели ключа eToken: eToken PRO, eToken NG-OTP, eToken NG-FLASH, eToken PRO (Java), eToken GT, eToken NG-OTP (Java), eToken PRO Anywhere, eToken NG-FLASH (Java).

Для работы с USB-ключом eToken требуется минимум один свободный порт USB.

Для работы со смарт-картой eToken требуется устройство чтения/записи Athena ASEDrive IIIe v2 (USB/Keyboard).

* 1. Требования к программному обеспечению

Сервер Крипто БД функционирует в среде сервера баз данных Oracle – Oracle Database Server 9i, 10g, 11g (Personal, Standard, Enterprise editions), под управлением одной из операционных систем, указанных в эксплуатационной документации СУБД Oracle. Требования к ОС сервера БД Oracle не предъявляются.

На АРМ администратора безопасности должна быть установлена одна из следующих операционных систем (ОС):

* + Microsoft Windows XP SP2, SP31 (32/64 бит);
  + Microsoft Windows Vista SP1, SP2[[1]](#footnote-1) (32/64 бит).

На АРМ администратора безопасности также требуется предварительная установка ПО:

* + Microsoft.NET Framework 2.0;
  + Oracle Client 9i, 10g или 11g for Windows (вариант администратора);
  + ПО eToken PKI Client 5.1 SP1 из состава изделия Электронный ключ eToken 5 или SafeNet Authentication Client.

При необходимости использования сертифицированных российских средств криптографической защиты на компьютер могут быть установлены СКЗИ: Инфотекс Домен КС2, КриптоПро 3.0 или 3.6, либо МагПро CSP 2.0.

Для хранения в памяти ключа eToken ключевых контейнеров КриптоПро CSP на компьютере должен быть установлен модуль поддержки eToken для КриптоПро CSP. Программу установки модуля поддержки eToken для КриптоПро CSP можно загрузить с Web-сайта компании Аладдин Р.Д.

При использовании КриптоПро CSP 3.0 и 3.6 убедитесь в том, что в число настроенных ключевых носителей входит ваша модель ключа eToken.

*Примечание* – При возможности выбора между драйверами, поставляемыми вместе с Windows, и драйверами производителей для контроллеров устройств, на которых будут создаваться зашифрованные диски, предпочтение следует отдавать драйверам производителей.

* 1. Ограничения

Средствами Крипто БД не допускается защищать информацию, составляющую государственную тайну.

Крипто БД не позволяет шифровать колонки таблиц для следующих условий:

* + таблица создана в схеме SYS, DBSNMP или SYSTEM;
  + на основе таблицы построено материализованное представление (materialized view), таблица очередей (Advanced Queue) или журнал материализованного представления (materialized view log);
  + таблица является индекс-организованной (Index-Organized table);
  + на таблицу наложены тонкие ограничения доступа (FGAC);
  + таблица является временной (temporary table);
  + шифруемая колонка входит в ограничения целостности — первичные, внешние ключи, CHECK.

1. Основные характеристики
   1. Пригодность

Крипто БД должен обеспечивать выполнение предусмотренных функций в соответствии с Руководством администратора безопасности 46538383.50 1430 005-01 92 01.

* 1. Способность к взаимодействию

Крипто БД должен обеспечивать взаимодействие с ключом eToken для:

* + идентификации администратора по уникальному идентификатору смарт-карты – Smart Card ID ключа eToken (на АРМ администратора) и записанному в нем сертификату пользователя в виде файла (в БД Oracle);
  + аутентификации пользователя или администратора (с проверкой предъявленного пароля ключа eToken) и проверки наличия в памяти ключа eToken сертификата;
  + чтения и записи ключевых пар.
  + Крипто БД должен обеспечивать взаимодействие с:
  + ПО управления ключами eToken – eToken PKI Client и SafeNet Authentication Client;
  + ПО Oracle Database Server;
  + криптопровайдерами КриптоПро CSP, Домен КС2 и МагПро CSP;
  + средством защиты от несанкционированного доступа (НСД) «Аккорд-АМДЗ» и электронным замком «Соболь».
  1. Защищенность

*Идентификация и аутентификация*

Для идентификации пользователей Крипто БД используются сертификаты формата X.509, хранящиеся в памяти ключа eToken. Только сертификаты пользователей, зарегистрированных в репозитории Крипто БД, могут использоваться для идентификации.

Получение пользователем ключей шифрования в процессе работы с сервером Oracle и выполнения операций с зашифрованными данными обеспечивается с помощью процедуры строгой двухфакторной аутентификации пользователя с применением ключей eToken.

*Управление доступом*

Пользователь получает доступ к защищённым данным только в том случае, если он:

* + прошел аутентификацию с использованием ключа eToken;
  + зарегистрирован в репозитории Крипто БД;
  + его ключ шифрования имеет соответствующую метку безопасности.

Регистрация (добавление) пользователей и их ключей шифрования, а также их удаление должны быть обеспечены (индивидуально и в пакетном режиме, используя список отозванных сертификатов в виде файла).

*Защита данных пользователя*

Данные на защищенных дисках всегда хранятся в зашифрованном виде. Даже в случае изъятия компьютера или утери съемного диска записанные на них данные невозможно использовать.

Размеры ключей для защиты ключей шифрования:

* + закрытый ключ – 256 бит;
  + открытый ключ - 512 бит.

Размеры ключей, используемых при шифровании — 256 бит.

Для формирования случайной последовательности должна быть возможность использования датчиков случайных чисел (ДСЧ), поддерживаемых сертифицированными СКЗИ, следующих типов:

* + биологический ДСЧ;
  + «Аккорд-АМДЗ»;
  + электронный замок «Соболь».

Должно быть обеспечено управление ключевой информацией и ее безопасное хранение.

*Аудит безопасности*

Функции: организации аудита, ключи аудита, управление информацией аудита.

Подсистема аудита должна регистрировать все предусмотренные (связанные с безопасностью) события видов:

* + служебные (регистрируются всегда при задании глобального флага аудита);
  + доступ к данным (регистрируются в соответствие с настройками).

*Защита ПО*

Должны быть обеспечены контроль целостности ПО и служебной информации Крипто БД, защита данных аудита и ключей шифрования путем шифрования. Доступ на чтение зашифрованных ключей и установка ключей шифрования должен предоставляться только после проверки целостности ПО.

Информация о защищенных таблицах/колонках должна храниться в БД в схеме пользователя – Администратора безопасности.

Ключевая система СКЗИ "Крипто БД" версии 2.0 обеспечивает возможность защиты ключей шифрования асимметричным алгоритмом с использованием для каждой пользовательской копии ключа шифрования пары уникальных ключей (открытого и закрытого), создаваемых на основе принципа открытого распределения ключей.

Ключи шифрования должны храниться в БД, зашифрованные ключом Администратора безопасности.

Для легального пользователя копии ключей шифрования должны храниться зашифрованными ключом пользователя.

Должно быть обеспечено сохранение незашифрованных копий ключей шифрования в файле на рабочей станции Администратора безопасности, защищенном паролем, и их восстановление из этого файла.

Области памяти, используемые для хранения данных Крипто БД, должны быть недоступны для несанкционированного чтения/изменения.

Сбор и обработку данных аудита/мониторинга должны выполнять процессы на рабочих станциях (не на сервере БД).

Информация об установках аудита (таблица, колонка, операция, пользователи), должна быть доступна только пользователю – Администратору безопасности и процессам сбора данных аудита.

В случае невозможности сбора аудита (если аудит был назначен для данного пользователя) система должна блокировать доступ к защищенным данным.

По завершении работы Крипто БД должен выполнять очистку всех элементов оперативной памяти компьютера, которые использовались в процессе работы для хранения и обработки информации.

## Согласованность

Интеграция Крипто БД с системами PKI обеспечивается использованием сертификатов X.509 и связанных с ними криптографических ключей для защиты ключей шифрования данных и аутентификации.

Алгоритм зашифрования/расшифрования данных и вычисления имитовставки реализован в соответствии с [2], [6], [3], [4], [5].

Алгоритмы защиты ключей шифрования реализованы в соответствии с требованиями [7], [3], [4], [5].

ПК Крипто БД должен соответствовать следующим руководящим документам (РД).

* + Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации (Гостехкомиссия России, 1992) – по классу 1Г.
  + Положение о методах и способах защиты информации в информационных системах персональных данных, утвержденного приказом ФСТЭК России от 5 февраля 2011 г. № 58 – по 1 классу.
  + Защита от несанкционированного доступа к информации. Часть 1. Программное обеспечение средств защиты информации. Классификация по уровню контроля отсутствия недекларированных возможностей» (Гостехкомиссия России, 1999). – по 4 уровню контроля.

## Надежность

В случаях сбоев (отказов) программного обеспечения, а также программного обеспечения среды функционирования, сбоев (отказов) аппаратного обеспечения (в том числе источников электропитания) защищенная память должна сохранять безопасное состояние. После восстановления работоспособности все функции безопасности должны функционировать корректно.

В случае нарушения целостности программ Крипто БД должна быть предусмотрена соответствующая процедура восстановления его работоспособности.

* 1. Практичность

Удобство работы, доступность данных должны обеспечиваться следующими средствами.

Для пользователей – работа системы абсолютно прозрачна.

Для администратора безопасности – быстрый ввод в эксплуатацию, удобное управление правами пользователей (не нужен доступ к eToken пользователей, используются сертификаты X.509).

Перешифрование данных выполняется как одна операция. Не надо сначала расшифровывать данные, а затем их зашифровывать с новым ключом и/или алгоритмом шифрования.

Возможности ведения мониторинга и аудита доступа к защищённым данным.

Крипто БД можно установить с интерфейсом пользователя на русском или английском языке.

1. Общие сведения
   1. Язык программирования

При разработке Крипто БД использовались языки программирования С, C++.

* 1. Управление конфигурацией

Для управления конфигурацией используется система SVN.

Каждый элемент конфигурации обладает свойством «revision» (ревизия), определяющим его конкретное состояние. При изменении элемента конфигурации изменяется свойство «revision» для этого элемента.

Совокупность элементов конфигурации, используемых для выпуска каждой версии продукта, помечается «тегом» (tag) – специальной меткой, обозначающей определённый релиз продукта.

* 1. Реализованные алгоритмы

Алгоритм шифрования ГОСТ 28147-89 реализован в соответствии с [2] и [6] в следующих режимах:

* + режим простой замены (ECB);
  + режим гаммирования (OFM);
  + режим гаммирования с обратной связью (CFB);
  + режим сцепления блоков (CBC);
  + режим простой замены c диверсификацией ключа шифрования (ECB-UKM);
  + режим гаммирования c диверсификацией ключа шифрования (OFB-UKM);
  + режим гаммирования с обратной связью c диверсификацией ключа шифрования (CFB-UKM);
  + режим сцепления блоков c диверсификацией ключа шифрования (CBC-UKM);
  + режим простой замены с выработкой имитовставки (ECB-MAC);
  + режим гаммирования с выработкой имитовставки (Counter mode-MAC);
  + режим гаммирования с обратной связью и выработкой имитовставки (CFB-MAC);
  + режим сцепления блоков с выработкой имитовставки (CBC-MAC);
  + режим простой замены с диверсификацией ключа шифрования и выработкой имитовставки (ECB-MAC-UKM);
  + режим гаммирования с диверсификацией ключа шифрования и выработкой имитовставки (OFB-MAC-UKM);
  + режим гаммирования с обратной связью, с диверсификацией ключа шифрования и выработкой имитовставки (CFB-MAC-UKM);
  + режим сцепления блоков с диверсификацией ключа шифрования и выработкой имитовставки (CBC-MAC-UKM).
  1. Использованное внешнее ПО

При разработке было использовано ПО внешних разработчиков:

* + ATL (Acive Template Library) – набор шаблонных классов языка [C++](http://ru.wikipedia.org/wiki/C%2B%2B), разработанных компанией [Microsoft](http://ru.wikipedia.org/wiki/Microsoft) для упрощения написания [COM](http://ru.wikipedia.org/wiki/Component_Object_Model)-компонентов;
  + ATrace – библиотека для ведения файлов с трассировочной информацией.
  1. Способ создания дистрибутива

Сборки CryptoDB, а также SCCAPI (вспомогательный компонент с криптографией), проводятся из исходных текстов на специальной сборочной платформе.

**Программное обеспечение, установленное на сборочную машину, используемое при создании дистрибутива:**

* + Windows XP Professional Service Pack 3 x86.
  + Visual Build Professional 7.0.
  + Microsoft Visual Studio 2008 Professional Edition.
  + NetBeans IDE 3.5.
  + Surround SCM Client 2008.1.1 Build 12.
  + Средство создания инсталляций Wix 3.0.
  + Plug-in встраивания Wix в Microsoft Visual Studio.
  + Средство для подписи дистрибутивов «Sign Tool».
  1. Сборка дистрибутива

Сборка дистрибутива включает следующий порядок действий пользователя:

* + Открытие в Visual Build Professional файлов build.bld, соответствующих собираемым проектам.
  + Запуск процедуры создания дистрибутива путем выбора опции Build -> Start в меню Visual Build Professional (или по нажатию «горячей клавиши» F7).

Первым собирается проект SCCAPI, затем – CryptoDB. После завершения процедуры дистрибутив продукта располагается на специальном сервере в каталоге .\Project\CryptoDB\Rel\_2.0\Setup.

1. Структура Крипто БД
   1. Структура Крипто БД и связь со средой

ПК Крипто БД включает:

* + библиотеки криптографии – пакеты хранимых процедур на Oracle Database Server;
  + сервер Крипто БД – пакет хранимых процедур на Oracle Database Server;
  + Средства администрирования, устанавливаемые на рабочей станции АДМБ.

Структура Крипто БД и его взаимосвязи со средой приведена на Рис. 1.

Рис.1 Структура Крипто БД и взаимосвязь со средой

* 1. Описание элементов

### Библиотеки криптографии

Библиотеки криптографии включают библиотеки, реализующие алгоритмы шифрования, и библиотеки работы с ключами шифрования данных.

Компонент Библиотеки криптографии устанавливается на сервере БД.

### Сервер Крипто БД

Сервер Крипто БД включает сервис вычисления ключей шифрования и сервис аудита.

Сервер Крипто БД обеспечивает:

* + получение зашифрованных ключей шифрования данных от сессий пользователя;
  + расшифрование и установка ключей шифрования данных в сессии пользователей;
  + оперативное управление аудитом;
  + аутентификация сессии пользователя.

Компонент сервер Крипто БД устанавливается на сервере БД.

### Репозиторий БД

Необходимая для работы Крипто БД информация хранится в объектах – таблицах БД на сервере БД. Таблицы с данными образуют репозиторий Крипто БД.

В таблицах репозитория содержится следующая информация:

* + описание используемых алгоритмов;
  + информация о ключах шифрования;
  + копии ключей шифрования пользователей с контрольными значениями;
  + сведения о зашифрованных колонках с контрольными суммами;
  + журнал служебных событий;
  + сообщения;
  + сертификаты (в формате X.509) и идентификаторы пользователей;
  + последовательности.

Информация о размещении таблиц хранится в схеме администратора безопасности.

### Средства администрирования

Средства администрирования установлены на АРМ администратора безопасности.

Состав компонента:

* + консоль администратора безопасности;
  + сценарии работы с базой данных.

*Консоль администратора безопасности*

Консоль администратора безопасности — приложение для интерактивного выполнения следующих функций по обеспечению конфиденциальности информации, хранящейся в таблицах баз данных Oracle.

Консоль АДМБ обеспечивает выполнение следующих функций:

* + создание и смена ключей сервера базы данных;
  + генерация, смена и удаление ключей шифрования;
  + зашифрование и расшифрование информации в таблицах БД;
  + управление копиями ключей шифрования для пользователей базы данных (приложений базы данных);
  + резервное копирование и восстановление ключей шифрования;
  + проведение контроля целостности и восстановление объектов Крипто БД;
  + управление аудитом.

*Сценарии работы с базой данных*

Сценарии работы с базой данных при установке ПО помещаются в папку ***C:\Program Files\Aladdin\eToken Crypto DB\*** вместе с другими исполняемыми файлами Крипто БД***.***

В зависимости от выбранного вида установки папка может содержать следующие предназначенные для администратора БД и администратора безопасности файлы (в подпапках):

* + ***scripts\grantaccess.sql*** — сценарий назначения дополнительных прав пользователю - администратору безопасности для шифрования данных в таблицах;
  + ***scripts\revokeaccess.sql*** — сценарий изъятия дополнительных прав у пользователя - администратора безопасности после завершения процедуры шифрования данных в таблицах;
  + ***scripts\shutdown.sql*** —сценарий выключения сервера Крипто БД;
  + ***scripts\startup.sql*** —сценарий включения сервера Крипто БД;
  + ***collector\collector.ini*** — шаблон конфигурации коллекторов аудита;
  + ***collector\collector.jar*** — исполняемая библиотека коллектора аудита;
  + ***collector\colStart.bat*** — сценарий запуска коллектора аудита (для ОС Windows);
  + ***collector\colStop.bat*** — сценарий останова коллектора аудита (для ОС Windows);
  + ***collector\lib\Ojdbc14.jar*** — JDBC-драйвер для коллектора аудита.

### АРМ пользователя

ПК Крипто БД взаимодействует с ПК eToken SecurLogon для Oracle, установленным на АРМ пользователя. В памяти ключа eToken пользователя, работающего с БД Oracle, должна быть записана лицензия пользователя на использование ПК.

1. Сервер Крипто БД
   1. Состав

Сервер Крипто БД представляет собой пакет хранимых процедур (PL/SQL пакет NCRYPTOKEY), реализующий всю функциональность компонента. Выполняется на сервере БД.

Ключевая пара сервера (открытый – закрытый ключ) предназначена для выполнения криптографических операций. Пара генерируется на АРМ АДМБ, с использованием сертифицированного поставщика криптографии. Ключевая пара непосредственно после генерации зашифровывается на открытом ключе АДМБ и сохраняется в реестре Windows на АРМ АДМБ.

**Пакет хранимых процедур хэша ключа –** PL/SQL пакет (NCRYPTOSERVER), содержащий значение хэш-функции от закрытого ключа сервера Крипто БД. Используется для проверки подлинности закрытого ключа. Алгоритм хэш-функции – ГОСТ 34.11-94.

* 1. Состояния сервера Крипто БД

Состояния сервера Крипто БД и возможные переходы состояний приведены на Рис. 2, описание состояния – в Табл. 1. Переходы между состояниями – см. Табл.А- 1 (Приложение А), обрабатываемые команды - Табл.А- 2 (Приложение А), список и описание внутренних структур – Табл.А- 3 (Приложение А).

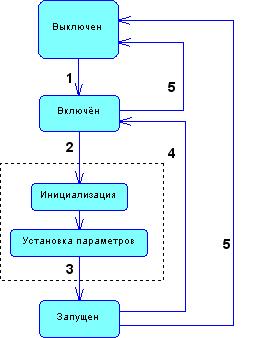
Рис. 2. Состояния сервера Крипто БД.

Табл. 1 – Описание состояний сервера Крипто БД.

| Состояние | Описание | Управляющее событие |
| --- | --- | --- |
| Выключен | Процесс не существует в контексте сервера БД | Команда «Выключить» из консоли администратора безопасности. |
| Включён | Процесс запущен. Внутренние структуры данных не инициализированы. Сервер Крипто БД готов к приёму и обработке сообщений от консоли администратора безопасности. | Выполнение sql-скрипта (вызов NCRYPTOKEY.SERVER\_STARTUP). |
| Инициализация | Установка закрытого ключа сервера Крипто БД. | Команды «Установить ключ» из консоли АДМБ. |
| Установка параметров | Установка параметров аудита. | Команда «Установить параметры аудита» из консоли АДМБ. |
| Запущен | Процесс запущен. Внутренние структуры данных инициализированы. Сервер Крипто БД готов к приёму и обработке сообщений от сессий пользователей. | Вызовы set\_dkey/set\_dkeys пакета NCRYPTO. |

## Включение сервера Крипто БД

Включение сервера Крипто БД производится вызовом процедуры пакета **NCRYPTOKEY.SERVER\_STARTUP**. Вызов осуществляется из консоли АДМБ, запущенной под учётной записью АДМБ (Рис. 3).

Вызов:

**NCRYPTOKEY.SERVER\_STARTUP();**

Входящие параметры:

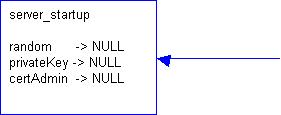
**нет**

Исходящие параметры:

**нет**

Возвращаемое значение:

**нет**

Рис. 3. Включение сервера Крипто БД.

## Запуск сервера Крипто БД

Запуск сервера Крипто БД производится с помощью консоли АДМБ. Процесс запуска представляет собой последовательность вызовов процедур/функций пакета **NCRYPTOKEY** (нумерация в соответствии с рис.3):

Инициализация.

Вызов:

**NCRYPTOKEY.SERVER\_INIT(password in varchar2, encryptedCert in raw);**

Входящие параметры:

**encryptedCert (raw)** – сертификат АДМБ, зашифрованный на пароле;

**password (varchar2)** – пароль.

Исходящие параметры:

**нет**

Возвращаемое значение:

**нет**

Процедура инициализирует переменную certAdmin значением сертификата АДМБ, расшифрованного с помощью пароля. Переменная random инициализируется случайным значением для последующей аутентификации сеанса.

Аутентификация сеанса.

Вызов:

**encryptedRandom := NCRYPTOKEY.SERVER\_SYNCRONIZE();**

Входящие параметры:

**нет**

Исходящие параметры:

**нет**

Возвращаемое значение:

**raw –** текущее значение переменной, зашифрованное на открытом ключе сертификата АДМБ**.**

Полученное значение расшифровывается закрытым ключом АДМБ на АРМ АДМБ и далее используется для аутентификации сеанса (следующей командой).

Установка закрытого ключа сервера Крипто БД (запуск).

Вызов:

**NCRYPTOKEY.SERVER\_START(random in raw, password in varchar2, encryptedKey in raw);**

Входящие параметры:

**random (raw)** – расшифрованное случайное значение (см. п.2).

**password (varchar2)** – пароль для расшифрования закрытого ключа сервера

Крипто БД.

**encryptedKey (raw)** - закрытый ключ сервера Крипто БД, зашифрованный на пароле.

Исходящие параметры:

**нет**

Возвращаемое значение:

**Нет**

Процедура аутентифицирует сеанс, сравнивая полученное значение параметра random с ранее сгенерированным (п.1) случайным значением. Если сравнение прошло успешно, переменная **privateKey** инициализируется значением закрытого ключа сервера Крипто БД, расшифрованного с помощью переданного пароля (**password**). Далее, переменной **random** присваивается новое случайное значение. Сервер Крипто БД переходит в состояние «Запущен» (**started**).

## Вычисления КШД

Вычисление КШД производится сеансами пользователей посредством клиентского ПО для Oracle. Процесс запуск представляет собой последовательность вызовов процедур/ функций пакета **NCRYPTO,** которые вызывают функцию пакета NCRYPTOKEY(нумерация в соответствии с рис.4):

1. Получение КШД.

Вызов:

**NCRYPTOKEY.DECRYPT(keyid in binary\_integer, alg in varchar2, value in raw, encrid in raw);**

Входящие параметры:

keyid (binary\_integer) – идентификатор КШД;

alg (varchar2) – идентификатор алгоритма шифрования. Используется для проверки целостности КШД.

value (raw) – КШД, зашифрованный на открытом ключе сертификата СК;

encid (raw) – идентификатор (keyid), зашифрованный передаваемым КШД по алгоритму alg.

Исходящие параметры:

**нет**

Возвращаемое значение:

**КШД**

Рис.4. Вычисление КШД

Функция расшифровывает конверт CMS, содержащий КШД, проверяет правильность расшифрования КШД и возвращает значение КШД. Проверка КШД представляет собой следующую последовательность: переданное значение encrid расшифровывается полученным КШД и соответствующим алгоритмом, полученный результат сравнивается с оригинальным значением идентификатора ключа (keyid). В случае совпадения проверка считается успешной.

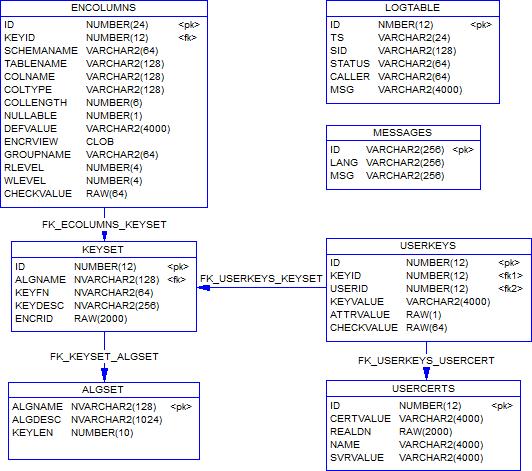
Типы данных, определяемые сервером Крипто БД, приведены в Приложении Б.

Программный интерфейс сервера Крипто БД – см. Приложение В.

1. Репозиторий
   1. Объекты БД

Схема объектов БД, составляющих репозиторий, приведена на Рис. 5.

Описание объектов БД приведено в Табл. Г- 1 – Табл. Г- 8 (Приложение А).

Рис. 5. Схема объектов БД

* 1. Схема владельца

Владельцу схемы должна быть предоставлена квота на некоторое табличное пространство, где в дальнейшем будут создаваться объекты (таблицы, индексы). Размер квоты зависит от предполагаемого количества ключей и пользователей. Для 200 ключей, 1000 зашифрованных колонок и 10000 пользователей требуется около 300М.

Схеме владельца следует назначить минимальные права и роли:

* + создание и изменение сессии;
  + создание триггеров и процедур в своей схеме;
  + создание таблиц и последовательностей в своей схеме;
  + выполнение пакета SYS.DBMS\_PIPE.

Для назначения привилегий выполнить от имени SYS команды:

grant create session, alter session to %OWNER%

grant create trigger, create procedure to %OWNER%

grant create table, create sequence to %OWNER%

grant execute on DBMS\_PIPE to %OWNER%

grant select\_catalog\_role to %OWNER%

grant select on SYS.SOURCE$ to %OWNER%

grant select on SYS.IDL\_UB1$ to %OWNER%

grant select on SYS.VIEW$ to %OWNER%

grant select on SYS.TRIGGER$ to %OWNER%

* 1. Создание объектов системы на сервере базы данных.

Объекты системы на стороне сервера создаются (пересоздаются) в схеме владельца и от имени владельца. Объекты перечислены в Приложении Г в порядке их создания.

Чтобы проверить были ли уже установлены объекты системы на указанном экземпляре базы данных, следует выполнить запрос от имени SYSTEM:

select table\_owner from all\_synonyms

where synonym\_name='NCRYPTO'

and owner = ‘PUBLIC’;

В случае непустой выборки, table\_ownerукажет на схему, где были установлены объекты системы. В этом случае следует выдается предупреждение относительно наличия установленных объектов и, возможно, наличия зашифрованных таблиц. В случае выбора пункта **Продолжить** - объекты пересоздаются, иначе – процесс установки прекращается.

* 1. Подготовка и создание пакетов и триггеров.
  + Предварительная обработка.

Тела пакетов предварительно обрабатываются утилитой **wrap,** например:

$ORACLE\_HOME\bin\wrap iname=aldncryptobody.sql edebug=wrap\_new\_sql

Файлы, полученные в результате обработки, впоследствии используются при установке системы.

Формирование триггеров на служебные таблицы.

В схеме владельца выполняется скрипт ‘triggers.sql’.

Создание публичного синонима, контекста приложения и назначение дополнительных прав.

Выполняются следующие команды от имени SYSTEM:

create or replace public synonym ncrypto for %OWNER%.NCRYPTO;

create or replace context sc\_ids using %OWNER%.SET\_SC\_IDS;

exec dbms\_java.grant\_permission( '%OWNER%','SYS:java.net.SocketPermission', '\*','connect,resolve')

exec dbms\_java.grant\_permission( '%OWNER%','SYS:java.net.SocketPermission', 'localhost:1024-', 'listen,resolve')

Проверка валидности установленных объектов.

Для получения списка установленных объектов, их типов и состояния следует выполнить запрос от имени владельца (%OWNER%):

Select object\_name,object\_type, status from all\_objects where owner='%OWNER%';

Для объектов, имеющих статус, отличный от '**VALID**' следует вывести на экран их имена и тип.

* 1. Установка начальных данных.

*Начальные данные для «eToken SafeData».*

Предполагает начальное заполнение справочника алгоритмов (**ALGSET**). Список предопределенных алгоритмов охватывает алгоритмы, предоставляемые пакетом DBMS\_OBFUSCATION\_TOOLKIT (Oracle 9.2.0.x):

insert into ALGSET values ('NULL','NO ENCRYPTION -- AUDIT ONLY',0);

insert into ALGSET values ('DES','Embedded Oracle DES',64);

insert into ALGSET values ('3DES','Embedded Oracle TripleDES',168);

commit;

и DBMS\_CRYPTO (Oracle 10g/11g):

insert into ALGSET values ('NULL','NO ENCRYPTION -- AUDIT ONLY',0);

insert into ALGSET values ('DES','Embedded Oracle 10g DES',64);

insert into ALGSET values ('3DES','Embedded Oracle 10g TripleDES',168);

insert into ALGSET values ('3DES\_2KEY','Embedded Oracle 10g TripleDES. Operates on a block 3 times with 2 keys. Effective key length of 112 bits.',168);

insert into ALGSET values ('AES128','Embedded Oracle 10g AES with 128bit key',128);

insert into ALGSET values ('AES192','Embedded Oracle 10g AES with 192bit key',192);

insert into ALGSET values ('AES256','Embedded Oracle 10g AES with 256bit key',256);

commit;

В случае возникновения ошибок – выдать диагностику сервера и продолжить установку.

*Начальные данные для Крипто БД*

Для всех поддерживаемых версий Oracle:

insert into algset values ('NULL', 'No encryption -- audit only', 0);

insert into algset values ('GOST-ECB', 'GOST 28147-89 (ECB mode)', 256);

insert into algset values ('GOST-ECB-MAC', 'GOST 28147-89 (ECB mode+Imito)', 256);

insert into algset values ('GOST-CFB', 'GOST 28147-89 (Crypto Pro CFB mode)', 256);

insert into algset values ('GOST-CFB-MAC', 'GOST 28147-89 (Crypto Pro CFB mode+Imito)', 256);

insert into algset values ('GOST-OFB', 'GOST 28147-89 (Crypto Pro OFB mode)', 256);

insert into algset values ('GOST-OFB-MAC', 'GOST 28147-89 (Crypto Pro OFB mode+Imito)', 256);

insert into algset values ('GOST-CBC', 'GOST 28147-89 (Crypto Pro CBC mode)', 256);

insert into algset values ('GOST-CBC-MAC', 'GOST 28147-89 (Crypto Pro CBC mode+Imito)', 256);

commit;

В случае возникновения ошибок – выдать диагностику сервера и продолжить установку.

1. Выполнение функций
   1. Шифрование данных

Встраивание реализуемых Крипто БД функций шифрования данных в таблицах баз данных Oracle реализовано с помощью интерфейса прикладного программирования (API). Функции API могут вызываться как непосредственно прикладным ПО (встраивание в новое прикладное ПО), так и через промежуточные представления и триггеры, автоматически создаваемые через консоль администратора безопасности (встраивание в существующее прикладное ПО без изменения его исходного кода).

Функции API Крипто БД делятся на следующие группы:

* + функции работы с репозиторием зашифрованных объектов и ключей.
  + функции работы с ключами шифрования;
  + функции зашифрования и расшифрования данных;
  + функции инициализации.
  1. Преобразование таблицы

Преобразование целевой таблицы реализуется операциями – зашифрованием и расшифрованием колонок. Каждая из операций атомарна, т.е. не допускается зашифрование колонки в целевой таблице, которая уже содержит зашифрованную колонку, и допускает расшифрование только одновременно всех зашифрованных колонок.

Исходные данные для операции зашифрования приведены в Табл. Е- 1 (Приложение Е).

Ниже приведены привилегии, которые должны быть предоставлены администратору безопасности перед операциями зашифрования или расшифрования таблиц. Список команд для предоставления привилегий вносится в скрипт, формируемый при установке ПО «Крипто БД» на рабочей станции администратора

Привилегии, необходимые для проведения операции шифрования:

- alter any table

- update any table

- create any index

- select any table

- create any table

- lock any table

**-** unlimited tablespace (или соответствующего размера квота на табличное пространство).

Приведенные в Табл. 2 шаги по преобразованию целевой таблицы выполняются для каждой из зашифровываемых колонок. Прочие колонки остаются без изменений. Порядок следования колонок в основном представлении должен соответствовать целевой таблице.

Табл. 2 – Преобразование целевой таблицы (зашифрование)

| Шаг | Реализация | Дополнительно |
| --- | --- | --- |
| Проверка возможности преобразования таблицы | См. п.п. 0. | При возникновении ошибки:  Прекратить дальнейшую обработку, выдать сообщение с соответствующей диагностикой. |
| Создание модельной таблицы (модели) | См. п.п. 0.  \*\*\*\*\*) | При возникновении ошибки прекратить дальнейшую обработку, выдать сообщение с диагностикой сервера. |
| Сохранение информации об оригинальной колонке | insert into %OWNER%.ENCOLUMNS  values (  %ECOLUMN\_ID%,  %KEYID%,  %SCHEMA%,  %ORIGTABLE%,  %COLNAME%,  %COLTYPE%,  %COL\_LENGTH%,  %COL\_PREC%,  %COL\_SCALE%,  %NULL\_NOTNULL%,  ’USERS’,  %MASK\_FMT%  ) | При возникновении ошибки выполнить:  - ROLLBACK  - действия, аналогичные п.п. 0 (обработка ошибочной ситуации).  Прекратить дальнейшую обработку, выдать сообщение с диагностикой сервера. |
| Завершение преобразования таблицы | п.п. 6.4.4. | п.п. 6.4.4. |
| Восстановление ограничений целостности | NCRYPTOUTL.RESTORE\_CONSTRAINTS  (  %SCHEMA%,  %ORIGTABLE%\_%SUFFIX%  %SCHEMA%,  %ORIGTABLE%,  1  ) | См. выше |
| Удаление целевой таблицы | для версии сервера БД 9i:  drop table %SCHEMA%.%ORIGTABLE% cascade constraint  для остальных версий:  drop table %SCHEMA%.%ORIGTABLE% cascade constraint PURGE | При возникновении ошибки выполнить:  прекратить дальнейшую обработку, выдать сообщение с диагностикой сервера. |
| Переименование таблицы | Если %ENCRYPT\_ONLY% == TRUE, выполнить  alter table %SCHEMA%.%ORIGTABLE%\_%SUFFIX% rename to %SCHEMA%.%ORIGTABLE% |  |
| Создание промежуточного представления | Если %ENCRYPT\_ONLY% == FALSE, выполнить  create or replace view %SCHEMA%.%ORIGTABLE%\_%SUFFIX%\_IM as  select  [...]  %OWNER%.ncrypto.decrypt\_%type%  (  %COLNAME%,[%COLNAME%$$1, %COLNAME%$$2,]  %ENCOLUMN\_ID%,  1  ) as %COLNAME%\_%SUFFIX%[,]  [...,]  rowid as ID$$$\_%SUFFIX%,  rownum as ROWNUM$$$\_%SUFFIX%  from %SCHEMA%.%ORIGTABLE%\_%SUFFIX% | При возникновении ошибки выполнить:  прекратить дальнейшую обработку, выдать сообщение с диагностикой сервера. |
| Создание основного представления | Если %ENCRYPT\_ONLY% == FALSE, выполнить  create or replace view %SCHEMA%.%ORIGTABLE% as  select  [...,]  %COLNAME%\_%SUFFIX% as %COLNAME%  [...]  from %SCHEMA%.%ORIGTABLE%\_%SUFFIX%\_IM | При возникновении ошибки выполнить:  - drop view %SCHEMA%.%ORIGTABLE%\_%SUFFIX%\_IM  Прекратить дальнейшую обработку, выдать сообщение с диагностикой сервера. |
| Создание триггера (insert) | Если %ENCRYPT\_ONLY% == FALSE, выполнить  create or replace trigger %SCHEMA%.ti\_%ORIGTABLE%\_%SUFFIX% instead of insert on %SCHEMA%.%ORIGTABLE%\_%SUFFIX%\_IM  for each row is  declare BV\_%COLNAME% RAW(4016);  begin  BV\_%COLNAME%:=%OWNER%.ncrypto.encrypt\_%type%(  :new.%COLNAME%\_%SUFFIX%,  %ENCOLUMN\_ID%,  2,  :new.ID$$$\_%SUFFIX%,  :new.ROWNUM$$$\_%SUFFIX%  )  insert into %SCHEMA%.%ORIGTABLE%\_%SUFFIX% ([...,]%COLNAME%[,...])  values ([...,]  BV\_%COLNAME%  [,...]);  end ti\_%ORIGTABLE%\_%SUFFIX%;  Для длинных (4000 > %FINAL\_LENGTH% > 2000) символьных (CHAR/VARCHAR/VARCHAR2/RAW) оператор insert формируется следующим образом:  insert into %SCHEMA%.%ORIGTABLE%\_%SUFFIX% ([...,]  %COLNAME%,  %COLNAME%$$1  [,...])  values ([...,]  %OWNER%.ncrypto.extract\_from\_raw(BV\_%COLNAME%,1,2000),  %OWNER%.ncrypto.extract\_from\_raw(BV\_%COLNAME%,2001,%FINAL\_LENGTH%-2000)  [,...]);  Для длинных (4000 > %FINAL\_LENGTH%) символьных (CHAR/VARCHAR/VARCHAR2/RAW) оператор insert формируется следующим образом:  insert into %SCHEMA%.%ORIGTABLE%\_%SUFFIX% ([...,]  %COLNAME%,  %COLNAME%$$1,  %COLNAME%$$2  [,...])  values ([...,]  %OWNER%.ncrypto.extract\_from\_raw(BV\_%COLNAME%,1,2000),  %OWNER%.ncrypto.extract\_from\_raw(BV\_%COLNAME%,2001, 2000),  %OWNER%.ncrypto.extract\_from\_raw(BV\_%COLNAME%,4001,%FINAL\_LENGTH%-4000)  [,...]); | Объявления  declare BV\_%COLNAME%...  делаются для каждой зашифрованной колонки.  При возникновении ошибки выполнить:  - drop view %SCHEMA%.%ORIGTABLE%  - drop view %SCHEMA%.%ORIGTABLE%\_%SUFFIX%\_IM  Прекратить дальнейшую обработку, выдать сообщение с диагностикой сервера. |
| Создание триггера (update)  \*\*\*\*) | Если %ENCRYPT\_ONLY% == FALSE, выполнить  create or replace trigger %SCHEMA%.tu\_%ORIGTABLE%\_%SUFFIX% instead of update on %SCHEMA%.%ORIGTABLE%\_%SUFFIX%\_IM  for each row  declare BV\_%COLNAME% RAW(4016);  begin  if updating (‘%COLNAME%\_%SUFFIX%’)  then  BV\_%COLNAME%:= %OWNER%.ncrypto.encrypt\_%type%  (  :new. %COLNAME%\_%SUFFIX%,  %ENCOLUMN\_ID%,  3,  :new.ID$$$\_%SUFFIX%,  :new.ROWNUM$$$\_%SUFFIX%  );  update %SCHEMA%.%ORIGTABLE%\_%SUFFIX% set  %COLNAME% = BV\_%COLNAME%  where rowid = :new.ID\_%SUFFIX%;  End if;  […]  if updating (%COLNAME%)  then  update %SCHEMA%.%ORIGTABLE%\_%SUFFIX% set  %COLNAME% = :new.%COLNAME%  where rowid = :new.ID\_%SUFFIX%;  end if;  […]  end tu\_%ORIGTABLE%\_%SUFFIX%;  Для длинных (4000 > %FINAL\_LENGTH% > 2000) символьных (CHAR/VARCHAR/VARCHAR2/RAW) оператор insert формируется следующим образом:  update %SCHEMA%.%ORIGTABLE%\_%SUFFIX% set  %COLNAME% = %OWNER%.ncrypto.extract\_from\_raw(BV\_%COLNAME%,1,2000),  %COLNAME%$$1 =  %OWNER%.ncrypto.extract\_from\_raw(BV\_%COLNAME%,2001,%FINAL\_LENGTH%-2000)  where rowid = :new.ID\_%SUFFIX%;  Для длинных (4000 > %FINAL\_LENGTH%) символьных (CHAR/VARCHAR/VARCHAR2/RAW) оператор insert формируется следующим образом:  update %SCHEMA%.%ORIGTABLE%\_%SUFFIX% set  %COLNAME% = %OWNER%.ncrypto.extract\_from\_raw(BV\_%COLNAME%,1,2000),  %COLNAME%$$1 =  %OWNER%.ncrypto.extract\_from\_raw(BV\_%COLNAME%,2001,2000),  %COLNAME%$$2 =  %OWNER%.ncrypto.extract\_from\_raw(BV\_%COLNAME%,4001,%FINAL\_LENGTH%-4000)  where rowid = :new.ID\_%SUFFIX%; | Блоки  if updated...  делаются для каждой колонки %COLNAME%.  При возникновении ошибки выполнить:  - drop view %SCHEMA%.%ORIGTABLE%  - drop view %SCHEMA%.%ORIGTABLE%\_%SUFFIX%\_IM  Прекратить дальнейшую обработку, выдать сообщение с диагностикой сервера. |
| Создание триггера (delete) | Если %ENCRYPT\_ONLY% == FALSE, выполнить create or replace trigger %SCHEMA%.td\_%ORIGTABLE%\_%SUFFIX% instead of delete on %SCHEMA%.%ORIGTABLE%\_%SUFFIX%\_IM  for each row  begin  delete from %SCHEMA%.%ORIGTABLE%\_%SUFFIX% where rowid = :old.ID\_%SUFFIX%;  %OWNER%.NCRYPTO.aud\_delete(  %ECOLUMN\_ID%,  :new.ID$$$\_%SUFFIX%  );  end td\_%ORIGTABLE%\_%SUFFIX%;  \*\*\*) | При возникновении ошибки выполнить:  - drop view %SCHEMA%.%ORIGTABLE%  - drop view %SCHEMA%.%ORIGTABLE%\_%SUFFIX%\_IM  Прекратить дальнейшую обработку, выдать сообщение с диагностикой сервера. |
| \*) Первичное зашифрование колонок производится от имени пользователя с ролью «Администратор безопасности», который не имеет доступа к зашифрованным данным ключом колонкам (см. п. FR0001- 0003 требований). В силу этого получение копии ключа шифрования администратором безопасности должно производится непосредственно из хранилища ключей шифрования (таблица USERKEYS), а установка расшифрованного значения ключа шифрования – вызовом соответствующей функции пакета из ПО утилит администратора безопасности (а не штатным методом через клиентское ПО SecurLogon for Oracle (см. ф.с. fs\_osl&de\_client.doc).  \*\*) При создании индекса сформированное по шаблону %ORIGTABLE%\_%COLNAME% имя следует ограничить 30 символами.  \*\*\*) Если в целевой таблице зашифровываются несколько колонок, то значение %ECOLUMN\_ID% для триггера выбирается для первой по порядку колонки.  \*\*\*\*) Проверка IF UPDATED и последующие действия производятся для каждой из колонок. | | |

**Ограничения зашифрования колонок**

Не могут быть зашифрованы колонки таблиц, если:

* + на таблице построен журнал материализованного представления (Materialized view log), это проверяется наличием записи в ALL\_MVIEW\_LOGS:

**select 1 from ALL\_MVIEW\_LOGS**

**where LOG\_OWNER = ‘%SCHEMA%’**

**and MASTER = ‘%ORIGTABLE%’**

* + таблица является базовой для материализованных представлений (Materialized view) и таблицы очередей (AQ table),

для AQ tables это проверяется наличием записи в ALL\_QUEUE\_TABLES:

**select 1 from ALL\_QUEUE\_TABLES**

**where OWNER = ‘%SCHEMA%’**

**and QUEUE\_TABLE = ‘%ORIGTABLE%’**

для Materialized views это проверяется наличием записи в ALL\_QUEUE\_TABLES:

**select 1 from ALL\_MVIEWS**

**where OWNER = ‘%SCHEMA%’**

**and MVIEW\_NAME = ‘%ORIGTABLE%’**

* + таблица является индекс-организованной (IOT), что проверяется по содержимому колонки IOT\_TYPE (NOT NULL)

**select 1 from ALL\_TABLES**

**where OWNER = ‘%SCHEMA%’**

**and TABLE\_NAME = ‘%ORIGTABLE%’**

**and IOT\_TYPE is NOT NULL**

* + на таблицу наложены тонкие ограничения доступа (FGAC), что проверяется наличием записи в V$VPD\_POLICY:

**select 1 from V$VPD\_POLICY**

**where OBJECT\_OWNER = ‘%SCHEMA%’**

**and OBJECT\_NAME = ‘%ORIGTABLE%’**

* + таблицы, содержащие колонки типа LONG, LONG RAW, BFILE – см. Табл. 2.
  + таблица является временной (temporary table), что проверяется по содержимому колонки DURATION (NOT NULL):

**select 1 from ALL\_TABLES**

**where OWNER = ‘%SCHEMA%’**

**and TABLE\_NAME = ‘%ORIGTABLE%’**

**and DURATION is NOT NULL**

* + таблица из схем SYS, SYSTEM – см. Табл. 2.
  + колонка входит в ключ раздела таблицы (partition) проверяется по содержимому колонки DURATION (NOT NULL):

**select 1 from ALL\_PART\_KEY\_COLUMNS**

**where OWNER = ‘%SCHEMA%’**

**and NAME = ‘%ORIGTABLE%’**

**and COLUMN\_NAME = ‘%COLNAME%’**

* + к зашифрованию предназначена колонка с данными, которые не соответствуют валидным типам данных (**‘NUMBER’,’CHAR’,’NCHAR’,’VARCHAR2’, ’NVARCHAR2’,’RAW’, ‘BLOB’, ‘CLOB’, ‘DATE') –** см. Табл. 2.

**Создание модельной таблицы.**

Выполняются следующие последовательности команд:

create table %SCHEMA%.%ORIGTABLE%\_%SUFFIX%

(

%COLNAME% %DST\_COLTYPE% (%FINAL\_LENGTH\_0%) %NULL\_NOTNULL%,

[%COLNAME%$$1 %DST\_COLTYPE% (%FINAL\_LENGTH\_1%), NULL]

[%COLNAME%$$2 %DST\_COLTYPE% (%FINAL\_LENGTH\_2%), NULL]

…

)

1. %FINAL\_LENGTH\_0% для типов %COLTYPE% == ‘CHAR’/’VARCHAR’/’VARCHAR2’/’RAW’, если %FINAL\_LENGTH% (с учетом служебной информации) не превышает 2000, устанавливается равной %FINAL\_LENGTH%.
2. Дополнительная колонка

[%COLNAME%$$1 %DST\_COLTYPE% (%FINAL\_LENGTH\_1%), NULL]

формируется только для типов %COLTYPE% == ‘CHAR’/’VARCHAR’/’VARCHAR2’/’RAW’, если %FINAL\_LENGTH% (с учетом служебной информации) превышает 2000 байт соответственно. В данном случае %FINAL\_LENGT\_0% устанавливается равной 2000, а %FINAL\_LENGT\_1% - (%FINAL\_LENGTH% - 2000).

1. Дополнительные колонки

[%COLNAME%$$1 %DST\_COLTYPE% (%FINAL\_LENGTH\_1%), NULL]

[%COLNAME%$$2 %DST\_COLTYPE% (%FINAL\_LENGTH\_2%), NULL]

формируется только для типов %COLTYPE% == ‘CHAR’/’VARCHAR’/’VARCHAR2’/’RAW’, если %FINAL\_LENGTH% (с учетом служебной информации) превышает 4000 байт соответственно. В данном случае %FINAL\_LENGT\_0% и %FINAL\_LENGT\_1% устанавливаются равными 2000, а %FINAL\_LENGT\_2% - (%FINAL\_LENGTH% - 4000).

* 1. Завершение преобразования таблицы.

Выполнить команды:

* + lock table %SCHEMA%.%ORIGTABLE% in exclusive mode NOWAIT
  + insert /\*+ Append \*/ into %SCHEMA%.%ORIGTABLE%\_%SUFFIX%
  + select %COL\_LIST% from %SCHEMA%.%ORIGTABLE%
  + commit

Опция **NOWAIT** в конструкции **lock table** указывает, что в случае если таблица читается или модифицируется кем-либо, возникнет ошибка «ORA-00054: resource busy and acquire with NOWAIT specified». В этом случае следует повторить операцию через некоторое время (по-умолчанию, 5 сек). Максимальное число попыток – 3. Если не удалось заблокировать таблицу после трёх попыток, следует выдать сообщение о занятости оригинальной таблицы и выполнить процедуру отката.

Конструкция **/\*+ Append \*/** задаёт подсказку оптимизатору для выполнения direct-path insert. Должна быть приведена в операторе именно таким образом.

* 1. Исходные данные для расшифрования

Исходные данные для операции расшифрования приведены в Табл. Е- 2 (Приложение Е).

Привилегии, указанные ниже, должны быть предоставлены администратору безопасности перед операциями расшифрования таблиц. Список команд для предоставления привилегий вносится в скрипт, формируемый при установке ПО Крипто БД на АРМ АДМБ.

Привилегии, необходимые для операции расшифрования:

- **alter any table**

**- drop any table**

**- drop any view**

**- update any table**

**- create any index**

**- select any table**

**- create any table**

**- lock any table**

**- unlimited tablespace** (или соответствующего размера квота на табличном пространстве).

* 1. Шаги по преобразованию целевой таблицы. Расшифрование.

Приведенные в таблице 3 шаги по преобразованию целевой таблицы выполняются для каждой из зашифрованных колонок.

Табл. 3 – Преобразование целевой таблицы (расшифрование)

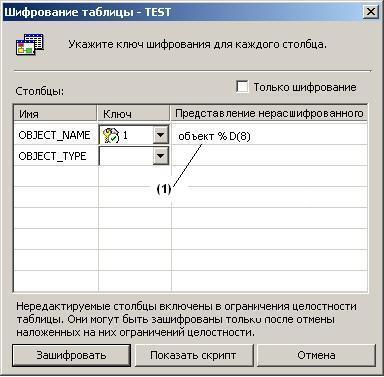
|  |  |  |
| --- | --- | --- |
| Шаг | Реализация | Дополнительно |
| Добавление суффикса (в случае зашифрования таблицы без построения промежуточных представлений и триггеров) | alter table %SCHEMA%.%ORIGTABLE% rename to %SCHEMA%.%ORIGTABLE%\_%SUFFIX% | Возможную ошибку игнорировать и продолжить обработку. |
| Проверка возможности преобразования таблицы | Аналогично п.п. 0. | При возникновении ошибки:  Прекратить дальнейшую обработку, выдать сообщение с соответствующей диагностикой. |
| Удаление промежуточного представления  \*\*) | drop view %SCHEMA%.%ORIGTABLE%\_%SUFFIX%\_IM | Возможную ошибку игнорировать и продолжить обработку. |
| Удаление основного представления  \*\*) | drop view %SCHEMA%.%ORIGTABLE% | Возможную ошибку игнорировать и продолжить обработку. |
| Создание промежуточной таблицы (модели) | create table %SCHEMA%.%ORIGTABLE%  (  [pk\_%ORIGTABLE% number NULL,]  …  %COLNAME% %DST\_COLTYPE% (%FINAL\_LENGTH%) %NULL\_NOTNULL%,  …  ) | При возникновении ошибки прекратить дальнейшую обработку, выдать сообщение с диагностикой сервера. |
| Установка размера колонки | alter table %SCHEMA%.%ORIGTABLE%  modify (  %COLNAME% %COLTYPE%(%COL\_PREC%[,%COL\_SCALE%])  ) | При возникновении ошибки :  Прекратить дальнейшую обработку, выдать сообщение с диагностикой сервера |
| Преобразование таблицы | См. П.2.2.4.1 | См. П.2.2.4.1 |
| Удаление информации об оригинальных колонках | delete from %OWNER%.ENCOLUMNS  where id = %ECOLUMN\_ID% | При возникновении ошибки: Прекратить дальнейшую обработку, выдать сообщение с диагностикой сервера. |
| Сохранение изменений | COMMIT | См. выше |
| Восстановление ограничений целостности | NCRYPTOUTL.RESTORE\_CONSTRAINTS  (  %SCHEMA%,  %ORIGTABLE%,  %SCHEMA%,  %ORIGTABLE%\_%SUFFIX%,  1  ) | См. выше. |
| Удаление целевой таблицы | для версии сервера БД 9i:  drop table %SCHEMA%.%ORIGTABLE%\_%SUFFIX% cascade constraint  для остальных версий:  drop table %SCHEMA%.%ORIGTABLE%\_%SUFFIX% cascade constraint PURGE | При возникновении ошибки выполнить:  Прекратить дальнейшую обработку, выдать сообщение с диагностикой сервера. |

## Маскирование

* 1. Описание решения

Маскирующие значения подставляются в результат запроса (только **select**) в случае отсутствия ключа шифрования или каких-либо иных ошибочных ситуациях. Шаблон маскирующих значений для колонки задаётся в процессе зашифрования колонки в консоли администратора безопасности и сохраняется в таблице зашифрованных колонок (ENCOLUMNS) в поле VALUEMASK (см. Приложение Г). Окно для задания маски приведено на Рис. 6.

Рис. 6. Интерфейс задания маски.



Правила задания маскирующих значений следующие.

Значение шаблона маскирующего значения может вводиться для каждой шифруемой колонки.

Значение по умолчанию – NULL.

После задания параметров шифрования и шаблона следует выполнить проверку валидности шаблона. Проверка проводится путём выполнения запроса соответствующей функции с введенным шаблоном. При возникновении исключения – выдается соответствующее сообщение, поле ввода шаблона очищается и появляется предложение повторить ввод.

* 1. Интерфейс маскирующих функций.

Функции маскировки значений зашифрованных колонок входят в состав пакета NCRYPTO как общедоступные (public) функции. Все функции вызывают исключение –20011 (Invalid mask) в случае задания некорректного значения маски.

**Функция генерации маскирующих значений для типов CHAR, VARCHAR2, CLOB, BLOB, RAW.**

Спецификация функции:

function maskval\_char(

maskchar IN varchar2,

maxlen IN binary\_integer default 16)

return varchar2;

Входящие параметры:

maskchar - Шаблон маски (см.табл.2.1).

maxlen **–** максимальное значение длины результирующего значения. В процедурах пакета NCRYPTO следует указывать актуальный размер колонки (Для char/varchar2/raw) или 256 (для blob/clob).

Возвращаемые параметры:

нет

Возвращаемое значение (**VARCHAR2**):

Маскирующая строка.

* 1. Функция генерации маскирующих значений для типов NUMBER.

function maskval\_numb(

numinterval IN varchar2 default null,

precision IN number default 38,

scale IN number default 0)

return number;

Входящие параметры:

numinterval - Шаблон маски (см. Приложение Г).

precision, scale – значения точности и шкалы результирующего значения. В процедурах пакета NCRYPTO следует указывать актуальные значения для колонки.

Возвращаемые параметры:

нет

Возвращаемое значение (NUMBER):

Маскирующее число.

* 1. Функция генерации маскирующих значений для типов DATE.

function maskval\_data(

dateinterval IN varchar2 default null)

return date;

Входящие параметры:

dateinterval - Шаблон маски (см. Приложение Г).

Возвращаемые параметры:

нет

Возвращаемое значение (DATE):

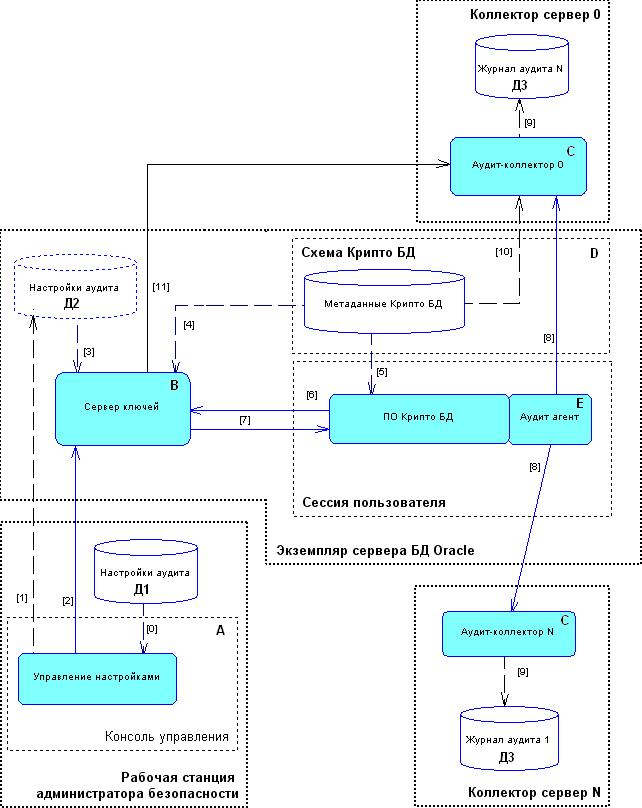
Маскирующая дата.

* 1. Аудит

**Общие сведения**

Регистрация событий доступа к колонкам таблиц с помощью подсистемы аудита возможна только для колонок, зашифрованных штатными средствами Крипто БД. Общая схема подсистемы аудита представлена на Рис. 7. Аудит ведётся на уровне колонок таблиц (с разбиением по типу доступа) для всех пользователей базы данных, с возможностью выборочной отмены аудита для некоторых пользователей.

* + Управление настройками аудита производится из консоли администратора безопасности.
  + Установка параметров аудита на уровне сессии пользователя осуществляется процессом сервера Крипто БД в соответствие с настройками системы аудита (п.2.1.3).
  + По умолчанию, зашифрованная колонка аудиту не подлежит.
  + При расшифровании таблицы настройки аудита для её колонок удаляются.

Рис. 7. Общая схема и потоки данных подсистемы аудита

Компоненты подсистемы аудита, изображенные на Рис. 7, приведены в Табл. 4. Хранилища данных подсистемы аудита приведены в Табл. 5, потоки данных подсистемы аудита –в Табл. 6.

Табл. 4 – Компоненты подсистемы аудита

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Реализация | Расположение | Функции |
| Консоль управления | Исполняемый модуль  (Windows приложение) | АРМ администратора безопасности | Первичное заполнение и управление настройками аудита.  Формирование установок аудита для сервера ключей.  Оперативное управление аудитом. |
| Сервер ключей | PL/SQL процедура (NCRYPTOKEY) | Сервер экземпляра БД | Установка параметров аудита для сессии пользователя.  Контроль доступности коллекторов аудита.  Управление установленными параметрами.  Формирование первичных записей событий аудита (служебные события – доступ к ключу, установка ключа, см. п. 2.1.6) |
| Аудит – коллектор | Java класс. | Отдельный физический хост, сервер экземпляра БД | Приём первичных записей событий аудита (все события). \*)  Формирование записей аудита и сохранение в файл. |
| Схема Крипто БД | PL/SQL триггер на таблицах репозитория:  ENCOLUMNS,  USERKEYS,  KEYSET | Сервер экземпляра БД | Формирование первичных записей событий аудита (служебные события – создание/удаление ключа шифрования, создание/удаление ключа шифрования для пользователя, зашифрование/расшифрование колонки, см. п. 2.1.6) |
| Аудит – агент | PL/SQL функция в составе пакета NCRYPTO | Сервер экземпляра БД | Запрос установок аудита для текущей сессии пользователя.  Формирование первичных записей событий аудита (события доступа в соответствие с установками, см. п. 2.1.6). |

Табл. 5 – Хранилища данных подсистемы аудита.

| Обозначение | Тип/расположение | Данные | Временное хранение |
| --- | --- | --- | --- |
| Д1 | Реестр OC Windows /  АРМ администратора безопасности | Настройки подсистемы аудита (п.2.1.3.1). |  |
| Д2 | Файл скрипта запуска сервера ключей /  Внешний носитель, файловая система АРМ администратора безопасности | Установки аудита (п.п. 2.1.4, 2.1.5). | Внутренние структуры данных пакета NCRYPTOKEY;  Внутренние структуры данных пакета NCRYPTO. |
| Д3 | Текстовый файл /  Файловая система хост-машины коллекторов аудита. | Сформированные события аудита (табл. 2.1). |  |

Табл. 6 – Потоки данных подсистемы аудита

| № | Источник | Назначение | Возникновение/  завершение | Данные / способ передачи |
| --- | --- | --- | --- | --- |
| [0] | Консоль администратора безопасности | Реестр ОС Windows | Управление настройками аудита/  Сохранение изменений | См. п. 2.1.3. |
| [1] | Консоль администратора безопасности | Скрипт запуска сервера ключей | Управление настройками аудита/  Сохранение изменений | Последовательность вызовов функций пакета NCRYPTOKEY с соответствующими параметрами (см. п.п. 2.1.5, 2.1.6). |
| [2] | Консоль администратора безопасности | Процесс сервера ключей | Команда установки/ снятия аудита | Вызов функций пакета NCRYPTOKEY с соответствующими параметрами (см. п.п. 2.1.5, 2.1.6). |
| [3] | Скрипт запуска сервера ключей | Процесс сервера ключей | Инициализация сервера ключей /  Завершение инициализации | Выполнение скрипта запуска / Вызов функций пакета NCRYPTOKEY с соответствующими параметрами (см. п.п. 2.1.5, 2.1.6). |
| [4] | Процесс сервера ключей | Метаданные Крипто БД | Инициализация сервера ключей /  Завершение инициализации | Данные таблиц:  ENCOLUMNS  USERKEYS  KEYSET |
| [5] | Метаданные Крипто БД | Сессия пользователя (ПО крипто БД) | Инициализация сессии/  Завершение инициализации | Данные таблиц:  ENCOLUMNS  USERKEYS  KEYSET |
| [6] | Сессия пользователя (ПО крипто БД) | Процесс сервера ключей | Запрос ключа шифрования/ | Данные сессии:  Идентификатор пользователя |
| [7] | Процесс сервера ключей | Сессия пользователя (ПО крипто БД) | Установка ключа шифрования/ | Данные установки аудита (п. 2.1.3.3) |
| [8] | Аудит – агент | Аудит - коллектор | Доступ к данным/ | Записи первичных событий (п. 2.1.6). |
| [9] | Аудит - коллектор | Журнал аудита | Обработка записей первичных событий/ | Сформированные события аудита (табл. 2.1). |
| [10] | Метаданные Крипто БД | Аудит - коллектор | Инициализация процесса аудит - коллектора/  Завершение инициализации | Данные таблиц:  ENCOLUMNS  USERKEYS  KEYSET |
| [11] | Процесс сервера ключей | Аудит - коллектор | Обработка служебных событий/  Завершение обработки | Записи первичных событий (п.п 2.1.6). |

Подсистема аудита регистрирует два вида событий:

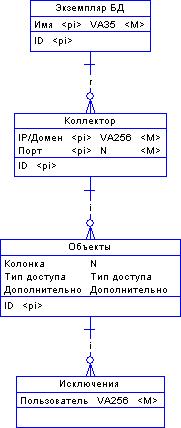
* + служебные (регистрируются всегда при задании глобального флага аудита);
  + доступ к данным (регистрируются в соответствие с настройками).
  + К служебным событиям относятся:
  + создание (генерация) ключа шифрования;
  + удаление ключа шифрования;
  + создание копии ключа шифрования для пользователя;
  + удаление копии ключа шифрования пользователя;
  + зашифрование колонки;
  + расшифрование колонки;
  + доступ к ключу шифрования (получение зашифрованного значения);
  + установка ключа шифрования (расшифрования ключа).
  + К событиям доступа к данным относятся:
  + чтение значения зашифрованной колонки (select);
  + вставка строк в таблицу, содержащую зашифрованные колонки (insert);
  + обновление значения зашифрованной колонки (update);
  + удаление строк из таблицы, содержащей зашифрованные колонки (delete).

Регистрация служебных событий осуществляется в случае установки глобального флага аудита (п.2.1.5). Регистрация событий доступа к данным возможна при установке глобального флага аудита (п.2.1.5) и задании соответствующих настроек (п.п. 2.1.4-2.1.6).

### Настройки аудита.

Все настройки, относящиеся к аудиту, хранятся в реестре Windows рабочей станции администратора безопасности. Структура хранения приведена на Рис. 8.

Общие настройки содержат информацию о нахождении аудит – коллекторов (доменное имя/IP адрес и порт). Аудит – коллекторы идентифицируются номером, уникальным в рамках экземпляра базы данных. Нумерация коллекторов начинается с 0. Коллектор с идентификатором 0 используется по умолчанию для сбора данных о служебных событиях.

Рис. 8. Структура хранения настроек аудита

### Аудит событий доступа

Все записи о настройках, относящихся к аудиту событий доступа, относятся к конкретному коллектору аудита. Запись о колонке, доступ к которой подлежит аудиту, содержит код колонки и флаги типа доступа (Табл. 7).

Табл. 7 – Настройки аудита

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Объект аудита | Тип доступа, подлежащий регистрации. | | | |
|  | select | insert | update | delete |
| <Код колонки> (ENCOLUMNS.ID) | <flag>:<info> | <flag>:<info> | <flag>:<info> | <flag>:<info> |
|  | <flag> - флаг аудита:  ‘Y’ – требуется аудит данного типа доступа;  ‘N’ – не требуется.  <info> - флаг дополнительной информации:  ‘Y’ – требуется указание дополнительной информации;  ‘N’ – не требуется. | | | |

Настройки аудита передаются процессу сервера ключей в скрипте запуска или из консоли администратора безопасности.

### Задание/снятие глобального флага аудита.

Осуществляется вызовом процедур пакета NCRYPTOKEY:

procedure enable\_audit(audit\_id IN number, audit\_host IN varchar2, audit\_port IN number);

procedure disable\_audit(audit\_id\_enc IN varchar2);

audit\_id - идентификатор коллектора аудита;

audit\_host – доменное имя или IP-адрес хоста коллектора аудита;

audit\_port – номер порта;

audit\_id\_enc - идентификатор коллектора аудита. Значение зашифровано открытым ключом сервера ключей. Base64 кодировка.

### Задание/снятие флага аудита доступа.

Осуществляется вызовом процедур пакета NCRYPTOKEY:

procedure set\_audit(audit\_id IN number, column\_id in NUMBER, operation IN varchar2, flags IN varchar2);

procedure remove\_audit(column\_id\_enc in varchar2);

audit\_id - идентификатор коллектора аудита;

column\_id – код зашифрованной колонки (ENCOLUMN.ID);

operation – строка флагов типа доступа (Табл. 7);

flags – строка флагов дополнительной информации;

column\_id\_enc - идентификатор колонки. Значение зашифровано открытым ключом сервера ключей. Base64 кодировка.

**Записи первичных событий.**

Записи первичных событий формируются аудит-агентами (события доступа к данным) или триггерами репозитория (служебные события) и передаются для обработки коллекторам аудита. Записи первичных событий содержат следующую информацию:

* + Категория события;
  + Тип события;
  + Код сессии;
  + Код субъекта;
  + Код объекта;
  + Флаг завершения операции;
  + Дополнительные данные.

Структура записи имеет следующий формат:

<категория события>:<тип события>:<код сессии>:<код объекта>:<код субъекта>:<флаг завершения>:<дополнительные данные>

Представление информации приведено в Табл. 8. Пример первичных записей аудита приведен в Приложении З.

Табл. 8 – Представление первичных событий

|  | Служебные события | События доступа к данным |
| --- | --- | --- |
| Категория события | ‘M1’ – служебное (для записи в журнал)  ‘M2’ – служебное (обновление внутренних структур данных)  ‘C’ – служебное (назначение/изменение или снятие аудита). | ‘A’ – доступ |
| Тип события | 'CA' – добавлен/изменен коллектор аудита;  'CR' – удалён коллектор аудита;  'AC' – добавлен/изменён аудит колонки;  'AR' – удалён аудит колонки;  'EA' – добавлено исключение аудита колонки;  'ER' – удалено исключение аудита колонки;  'LO' – инициализация сессии пользователя; \*\*\*)  'KM' - смена ключа шифрования;  'KC' - создание (генерация) ключа шифрования;  ‘KR’ - удаление ключа шифрования;  ‘UC’ - создание копии ключа шифрования для пользователя;  ‘UM’ - смена ключа шифрования для пользователя;  ‘UR’ - удаление копии ключа шифрования пользователя;  ‘EC’ - зашифрование колонки;  ‘DC’ - расшифрование колонки;  ‘GK’ – доступ к ключу;  ‘PK’ – установка ключа. | ‘1’ – чтение (select);  ‘2’- вставка (insert);  ‘3’ – обновление (update);  ‘4’ – удаление (delete). |
| Код субъекта | ‘< USER\_KEY\_ID >’ – для типов событий 'UC', 'UR', GK', 'PK', 'EA', 'ER'.;  ‘<DN-ADM>’ – для остальных типов событий.  < USER\_KEY\_ID > – Код ключа пользователя (значение колонки USERKEYS.ID).  <DN-ADM> – отличительное имя сертификата пользователя – администратора безопасности. Без лидирующих и завершающих пробелов в компонентах имени, переведено в верхний регистр. | ‘<USER\_KEY\_ID>’  < USER\_KEY\_ID > – Код ключа пользователя (значение колонки USERKEYS.ID). |
| Код сессии | <’SID’>  <SID> := sys\_context(‘userenv’,’sessionid’) | |
| Код объекта | ‘<key\_ID>’ – для типов событий 'KC', ‘KR’, ‘UC’, ‘UR’, ‘GK’, ‘PK’;  ‘<column\_ID>’ – для типов событий 'EC', 'DC', 'AC', 'AR', 'EA', 'ER'.  <key\_ID> - идентификатор ключа шифрования (значение колонки KEYSET.KEYID);  <column\_ID> - код колонки (значение колонки ENCOLUMNS. ID). | ‘<column\_ID>’  <column\_ID> - код колонки (значение колонки ENCOLUMNS. ID). |
| Флаг завершения операции | 'S' – операция успешна;  ‘F’ – операция неуспешна. | 'S' – доступ разрешён;  'F' – доступ запрещён \*) |
| Дополнительные данные | <os\_user>:  <ora\_user>:  <ora\_ext\_user>:  <ora\_proxy\_user>:  <client\_IP>:  <client\_term>  <os\_user> := sys\_context(‘userenv’,’OS\_USER’)  <ora\_user> := sys\_context(‘userenv’,’SESSION\_USER’)  <ora\_ext\_user> := sys\_context(‘userenv’,’EXTERNAL\_NAME’)  <ora\_proxy\_user> :=  sys\_context(‘userenv’,’PROXY\_USER’)  <client\_IP> := sys\_context(‘userenv’,’IP\_ADDRESS’)  Если <client\_IP> is NULL, то принимается <client\_IP> := ‘Console’  <client\_term> := sys\_context(‘userenv’,’TERMINAL’)  Дополнительно:  Для событий ‘CA’, ‘RA’  ‘HOST’:’PORT’  Для событий 'AC', 'AR', 'EA', 'ER'  <Coll-ID>:<Audit-Mask>:<Ext-Flag>  Для событий 'EA', 'ER'  <Excl-Name> | **<row\_ID>**  \*\*) |
| \*) 'S' – доступ с валидным ключом шифрования для данного пользователя;  'F' – доступ без ключа шифрования для данного пользователя.  \*\*) Опционально  \*\*\*) Событие генерируется из секции инициализации пакета NCRYPTO. Результат – всегда успешный – ‘S’. | | |

### Записи событий аудита.

Записи событий осмысленный набор информации, предназначенный для хранения и последующей обработки. Записи аудита формируются коллекторами аудита из записей первичных событий, сохраняются в файл и содержат следующую информацию:

* + Категория события;
  + Тип события;
  + Штамп времени;
  + Субъект события;
  + Объект события;
  + Флаг завершения операции;
  + Дополнительная информация.

Формат записи файла:

<штамп времени><delim><категория события><delim><тип события><delim><код сессии><delim><кодобъекта><delim><код субъекта><delim><флаг завершения><delim><дополнительные данные>

Представление информации приведено в Табл. 9. Пример записей событий аудита приведен в Приложении З.

Табл. 9 – Представление событий коллекторами аудита

|  | Служебные события | События доступа к данным |
| --- | --- | --- |
| <delim> | <HT> = 0x09 | |
| Тип события | 'CA' – добавлен/изменен коллектор аудита;  'CR' – удалён коллектор аудита;  'AC' – добавлен/изменён аудит колонки;  'AR' – удалён аудит колонки;  'EA' – добавлено исключение аудита колонки;  'ER' – удалено исключение аудита колонки;  'LO' – инициализация сессии пользователя;  'KC' - создание (генерация) ключа шифрования;  ‘KR’ - удаление ключа шифрования;  ‘UC’ - создание копии ключа шифрования для пользователя;  ‘UR’ - удаление копии ключа шифрования пользователя;  ‘EC’ - зашифрование колонки;  ‘DC’ - расшифрование колонки;  ‘GK’ – доступ к ключу;  ‘PK’ – установка ключа. | ‘S’ – чтение (select);  ‘I’- вставка (insert);  ‘U’ – обновление (update);  ‘D’ – удаление (delete). |
| Штамп времени | 'YYYY-MM-DD HH24:MI:SS' | |
| Код сессии | '<SID>'  Значение <код сессии> полученное из первичной записи аудита | |
| Субъект события | ‘<DN>’ – для типов событий 'UC', 'UR';  ‘<DN-ADM>’ – для остальных типов событий.  <DN> – отличительное имя сертификата пользователя, соответствующего закрытому ключу, с помощью которого был получен ключ шифрования для данного объекта доступа. Без лидирующих и завершающих пробелов в компонентах имени, переведено в верхний регистр.  <DN-ADM> – отличительное имя сертификата пользователя – администратора безопасности. | ‘<DN>’  <DN> – отличительное имя сертификата пользователя, соответствующего закрытому ключу, с помощью которого был получен ключ шифрования для данного объекта доступа. Без лидирующих и завершающих пробелов в компонентах имени, переведено в верхний регистр. |
| Объект события | <audit-collector> - для типов событий ‘CA’, ‘RA’;  <user\_session> – для типа события ‘LO’;  ‘<key\_fname>’ – для типов событий 'KC', ‘KR’, ‘UC’, ‘UR’, ‘GK’, ‘PK’;  ‘<schema>’: ‘<table>’: ‘<column>’ – для типов событий 'EC', 'DC', 'AC', 'AR'.  <audit-collector> - <ID-Coll>;  <user\_session> - строка ‘SESSION’;  <key\_fname> - «дружественное имя» ключа;  <schema> - имя схемы;  <table> - имя таблицы;  <column> - имя колонки. | ‘<schema>’: ‘<table>’: ‘<column>’  <schema> - имя схемы;  <table> - имя таблицы;  <column> - имя колонки. |
| Флаг завершения операции | 'S' – операция успешна;  ‘F’ – операция неуспешна. | 'G' – доступ разрешён;  'D' – доступ запрещён;  'P' – доступ назначен;  ‘X’ – доступ отозван.\*) |
| \*)  'G' – доступ с валидным ключом шифрования для данного пользователя;  'D' – доступ без ключа шифрования для данного пользователя;  'P' – ключ шифрования для пользователя создан, но текущие сессии его ещё не имеют;  'X' – ключ шифрования для пользователя удалён, но текущие сессии продолжают его использовать. | | |

### Коллекторы аудита.

**Общие сведения.**

1. Коллекторы аудита – автономные консольные java-приложения.
2. Соединения с БД производится только в момент инициализации приложения для чтения данных репозитория.

**Настройки коллектора аудита.**

Настройки коллектора аудита задаются в файле параметров, содержащий пары «параметр» = «значение». Данный файл содержит следующие параметры:

DB\_URL=<строка соединения для тонкого клиента>

USER\_NAME=<имя пользователя-администратора безопасности>

AUDIT\_PORT=<номер порта>

AUDIT\_FILE=<имя файла журнала (постоянная часть) с расширением (опционально)>

<строка соединения для тонкого клиента> ::= jdbc:oracle:thin:@<host>:<port>:<service name>

**Формирование имени файла журнала**

Имя файла журнала формируется по следующему правилу:

1. Из параметра AUDIT\_FILE выделяется постоянная часть имени файла.
2. К полученному имени добавляется ‘\_YYYYMMDD’. Здесь YYYYMMDD – дата на момент запуска коллектора.
3. К полученному имени добавляется расширение файла из параметра AUDIT\_FILE, если было задано.
4. Взаимодействие сервера Крипто БД и клиентского ПО

Сервер Крипто БД взаимодействует с клиентским ПО, установленным на АРМ пользователей, с использованием сетевого протокола взаимодействия Net\*8.

При взаимодействии выполняются следующие шаги:

ПО eToken SecurLogon для Oracle, установленное на АРМ пользователя, передает запрос прикладного ПО на аутентификацию и установление SSL-соединения на сервер БД.

Инициализация сессии. При этом проверяется наличие лицензии на продукт.

Чтение и установка ключей шифрования.   
При этом выполняются следующие действия.

* + Чтение зашифрованного ключа шифрования.
  + Расшифрование полученного значения с помощью закрытого ключа, соответствующего сертификату, предъявленному пользователем после успешной аутентификации в БД Oracle.
  + Зашифрование ключа шифрования на открытом ключе сервера.

Расшифрование ключа шифрования с помощью закрытого ключа из памяти ключа eToken, соответствующего открытому ключу сертификата, предъявленного пользователем.

Открытый ключ сервера расшифровывается аналогично.

Передача ключа шифрования на АРМ пользователя.

Чтение/изменение расшифрованных данных.

* 1. Вызов и загрузка

Включение сервера Крипто БД производится вызовом процедуры пакета **NCRYPTOKEY.SERVER\_STARTUP**. Вызов осуществляется из консоли администратора безопасности, запущенной под учётной записью АДМБ.

Файл, содержащий сценарий включения сервера ключей, формируется на этапе установки ПО Крипто БД и располагается в папке установки ПО Крипто БД в подпапке *"C:\Program Files\Aladdin\eToken Crypto DB\scripts"*. Текст сценария приложен также к документу [1].

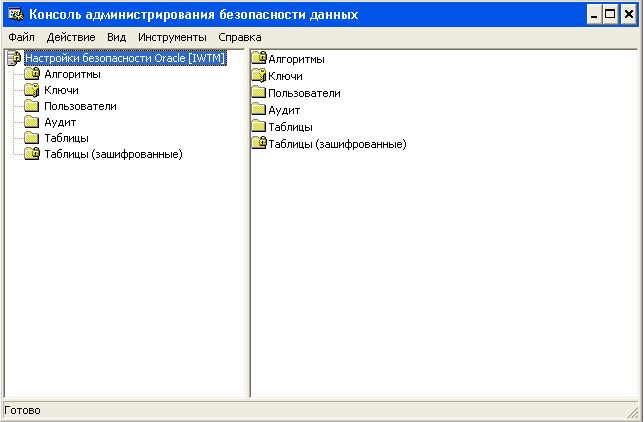
Включение сервера ключей производится непосредственно на сервере базы данных, а запуск — с АРМ администратора безопасности.

При запуске сервера ключей выполняются следующие действия:

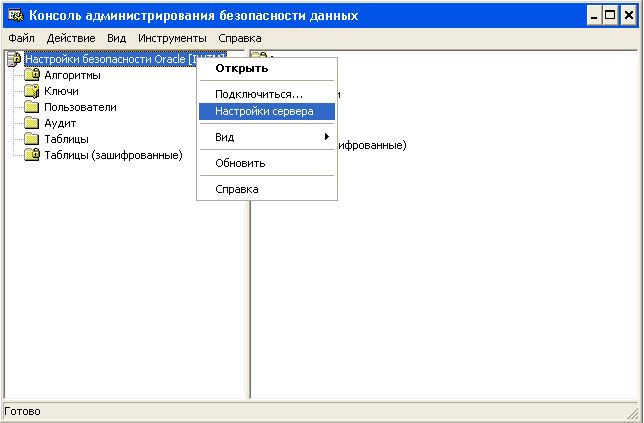
* + проверка целостности объектов Крипто БД;
  + установка настроек аудита;
  + установка закрытого ключа сервера ключей.

Для того чтобы запустить включённый сервер ключей, на АРМ администратора безопасности выполните следующее.

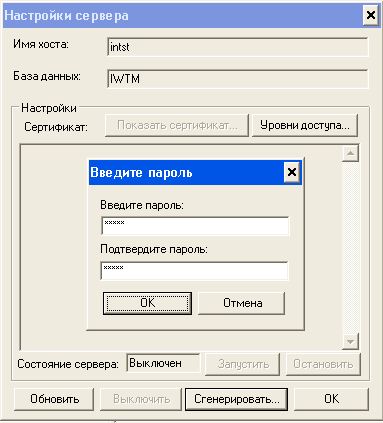
1. Убедитесь в том, что eToken, содержащий закрытый ключ и сертификат администратора безопасности, подключён к АРМ. Если это не так, подключите его.
2. В дереве консоли администратора безопасности, соединённой с целевой базой данных (см. Рис. 9), выберите корневой элемент ***Настройки безопасности Oracle [<tnsname>]***», где ***<tnsname>*** — сетевой псевдоним экземпляра базы данных Oracle.

 Рис. 9. Консоль АДМБ. Главное окно

1. Из контекстного меню (щёлкнув правой кнопки мыши) или из меню ***Действие*** выберите ***Настройки сервера*** (Рис. 10).

Рис. 10. Консоль АДМБ. Обращение к интерфейсу настроек сервера

1. В окне **Настройки сервера** убедитесь в том, что поле **Состояние сервера** принимает значение **«*Остановлен*»**, и щёлкните **Запустить**.
2. При необходимости в открывшемся окне запроса пароля введите пароль ключа eToken администратора безопасности и щёлкните **ОК**.
3. В окне **Ввод пароля** (Рис. 11) введите пароль закрытого ключа сервера БД в поле **Введите пароль** и щёлкните **ОК**.

 Рис. 11. Консоль АДМБ. Ввод пароля для защиты закрытого ключа сервера БД

1. В случае успешной активации поле **Состояние сервера** примет значение «*Запущен…*».
2. Щёлкните **ОК**.
   1. Входные данные

Крипто БД получает входные данные следующих типов:

* + параметры, устанавливаемые по умолчанию при установке и конфигурировании Крипто БД;
  + файлы сценариев, шаблоны конфигурации, исполняемые библиотеки, предназначенные для администратора базы данных и администратора безопасности и записанные при установке ПО Крипто БД в папку установки продукта;
  + параметры, вводимые администратором безопасности вручную при установке и конфигурировании Крипто БД;
  + параметры, вводимые администратором безопасности средствами консоли АДМБ;
  + параметры использования продукта, содержащиеся в серверной лицензии;
  + ключи шифрования, передаваемые с АРМ пользователя;
  + ключевая пара сервера экземпляра базы данных;
  + открытый ключ из ключа eToken;
  + команды администратора безопасности;
  + файлы сценариев;
  + эталонные значения хэш-функции объектов, хранящиеся на АРМ АДМБ;
  + вычисленные хэш-функции объектов – для сравнения их с эталонными значениями при выполнении контроля целостности;
  + метки объекта и субъекта для дополнительного ограничения доступа;
  + данные пользователей в колонках таблиц БД – для зашифрования;
  + данные пользователей, зашифрованные ранее;
  + зашифрованные копии ключей шифрования;
  + пароль защиты резервной копии ключа шифрования.
  1. Выходные данные

Крипто БД формирует выходные данные следующих типов:

* + ключевая пара сервера – для записи в реестр Windows на АРМ АДМБ;
  + ключей шифрования пользователей – в репозиторий;
  + зашифрованный ключ шифрования – на АРМ пользователя;
  + данные о событиях, отображаемые в окне журнала служебных событий в консоли АДМБ или сохраненные в файле;
  + зашифрованные (для хранения) данные колонок таблиц БД;
  + расшифрованные (для работы пользователей) данные колонок таблиц БД;
  + данные аудита;
  + сведения о результатах проведения КЦ;
  + файлы резервной копии ключа шифрования;
  + информационные сообщения, сообщения об ошибках.

Приложение А. Сервер Крипто БД

Табл.А- 1 – Переходы между состояниями сервера

|  |  |  |
| --- | --- | --- |
| Переход  (согл. Рис. 2) | Описание | Действия |
| 1 | Включение | Запуск процедура PL/SQL пакета;  Инициализация значения закрытого ключа сервера Крипто БД пустым значением;  Инициализация значения сертификата АДМБ пустым значением;  Инициализация параметров аудита пустыми значениями;  Ожидание управляющих событий. |
| 2 | Запуск | Установка сертификата АДМБ. |
| 3 | Инициализация | Аутентификация сеанса АДМБ;  Приём события «Запустить сервер»;  Расшифрование с помощью пароля закрытого ключа сервера Крипто БД ;  Проверка валидности закрытого ключа;  Установка параметров аудита (коллекторы, колонки, исключения);  Ожидание управляющих событий и событий пользовательских сессий. |
| 4 | Сброс состояния | Аутентификация сеанса а АДМБ;  Приём события «Остановить сервер»;  Уничтожение значения закрытого ключа СК;  Ожидание управляющих событий. |
| 5 | Выключение | Уничтожение значения закрытого ключа СК;  Завершение процесса СК. |

Табл.А- 2 – События, обрабатываемые сервером

| Команда | Описание | Действия/результат |
| --- | --- | --- |
| shutdown | Выключение сервера | Аутентификация сеанса АДМБ;  Завершение цикла ожидание управляющих событий;  Уничтожение канала связи СК;  Возвращение управления sql-консоли. |
| status | Запрос состояния | Возврат статуса сервера Крипто БД (см. табл. 1). |
| init | Инициализация | Установка сертификата АДМБ во внутренних структурах пакета. |
| syncronize | Получение значения для аутентификации сеанса | Генерация случайного числа (32 байта);  Возврат указанного числа, зашифрованного на открытом ключе сертификата АДМБ. |
| done | Завершение сеанса | Аутентификация сеанса АДМБ;  Обнуление значения сертификата АДМБ;  Генерация случайного числа (32 байта);  Возврат указанного числа, зашифрованного на открытом ключе сертификата АДМБ. |
| stop | Останов СК | Аутентификация сеанса АДМБ;  Обнуление значения закрытого ключа СК;  Обнуление значения сертификата АДМБ;  Обнуление структур установок аудита. |
| start | Запуск СК | Аутентификация сеанса администратора безопасности;  Проверка установки закрытого ключа сервера Крипто БД по его хэш-значению;  Получение (расшифрование по паролю) и проверка закрытого ключа сервера Крипто БД по хэш-значению;  Выполнение отложенных команд. |
| collector\_set | Установка настроек аудита | Аутентификация сеанса АДМБ;  Заполнение структур настроек аудита (параметры коллектора аудита). |
| collector\_get | Запрос настроек аудита | Аутентификация сеанса;  Возврат настроек аудита (параметры коллектора аудита). |
| column\_set | Установка настроек аудита | Аутентификация сеанса АДМБ;  Заполнение структур настроек аудита (параметры колонок для аудита). |
| column\_get | Запрос настроек аудита | Аутентификация сеанса;  Возврат настроек аудита (параметры колонок для аудита). |
| excpt\_add | Установка настроек аудита | Аутентификация сеанса АДМБ;  Заполнение структур настроек аудита (параметры исключений аудита). |
| excpt\_remove | Удаление настроек аудита | Аутентификация сеанса АДМБ;  Возврат настроек аудита (параметры исключений аудита). |
| decrypt | Расшифровать КШД | Аутентификация сеанса;  Расшифрование КШД с помощью закрытого ключа СК;  Возврат КШД. |

Табл.А- 3 Внутренние структуры сервера Крипто БД

|  |  |  |
| --- | --- | --- |
| Обозначение | Описание | Примечание |
| privateKey | Содержит значение закрытого ключа сервера Крипто БД. |  |
| certAdmin | Содержит значение сертификата АДМБ. |  |
| random | Случайное значение (32 байта). | Изменяется для выполнения каждого обращения к серверу Крипто БД. |
| status | Текущее состояние сервера Крипто БД. |  |

Приложение Б. Типы данных, определяемые сервером Крипто БД

-- описание пользователя

type USERSTRUCT is record (

name varchar2(1024), -- отображаемое имя пользователя

dn raw(4096) -- отличимое имя пользователя

);

-- список пользователей

type USERLIST is table of USERSTRUCT index by binary\_integer;

-- описание аудита столбца

type COLSTRUCT is record (

audid binary\_integer, -- идентификатор коллектора

access varchar2(4), -- флаги мониторинга доступа

ext varchar2(4), -- флаги дополнительной информации

excpt USERLIST -- исключенные пользователи

);

-- описание коллектора

type AUDITSTRUCT is record (

host varchar2(256), -- имя хоста для аудита

port number -- номер порта для аудита

);

Приложение В. Программный интерфейс сервера Крипто БД

-- получить состояние сервера

function server\_status return binary\_integer;

-- сравнение пользователей

function equals\_users

(

info in USERSTRUCT,

name in varchar2,

dn in raw default null) return boolean;

-- создание и завершение цикла обработки сообщений

procedure server\_startup;

procedure server\_shutdown;

-- перевод сервера в состояние установки параметров

procedure server\_init(password in varchar2, encryptedCert in raw);

-- синхронизация и остановка сервера

function server\_syncronize return raw;

procedure server\_done(random in raw);

-- перевести сервер в рабочее состояние

procedure server\_start(

random in raw,

password in varchar2,

encryptedKey in raw);

procedure server\_stop (random in raw);

-- установить настройки аудита коллектора

procedure collector\_set(

random in raw,

audid in binary\_integer,

host in varchar2,

port in binary\_integer);

-- установить настройки аудита столбца

procedure column\_set(

random in raw,

colid in binary\_integer,

audid in binary\_integer,

access in varchar2,

ext in varchar2);

-- добавить/удалить исключение

procedure excpt\_add(

random in raw,

colid in binary\_integer,

name in varchar2,

dn in raw default null);

procedure excpt\_remove(

random in raw,

colid in binary\_integer,

name in varchar2,

dn in raw default null);

-- получить настройки аудита

procedure collector\_get(audid in binary\_integer, info out AUDITSTRUCT);

procedure column\_get (colid in binary\_integer, info out COLSTRUCT );

-- расшифровать ключ шифрования

function decrypt(

keyid in binary\_integer,

alg in varchar2,

value in raw,

encrid in raw) return raw

Приложение Г. Описание объектов БД

Табл. Г- 1 – Алгоритмы (ALGSET).

|  |  |  |
| --- | --- | --- |
| Колонка | Тип, свойства | Описание колонки |
| ALGNAME | NVARCHAR2(128)  NOT NULL  Primary Key | Описание алгоритма. Не может быть изменено пользователем. Набор предопределенных значений. |
| ALGDESC | NVARCHAR2(1024)  NULL | Описание алгоритма. Может быть изменено пользователем. Допустимы символы Unicode.  Используется для подробного описания алгоритма. |
| KEYLEN | NUMBER(10)  NULL | Длина ключа для данного алгоритма (если применимо) в битах, иначе 0. Не может быть изменено пользователем.  Набор предопределенных значений.  Используется процедурами генерации ключа шифрования. |

Табл. Г- 2 – Ключи шифрования (KEYSET).

| Колонка | Тип, свойства | Описание |
| --- | --- | --- |
| KEYID | NUMBER(12)  NOT NULL  Primary Key | Идентификатор ключа.  Положительное целое число. |
| ALGNAME | NVARCHAR2(128)  NOT NULL  Foreign Key (ALGSET) | Идентификатор алгоритма, для которого создан ключ шифрования. |
| KEYFN | NVARCHAR2(64)  NOT NULL | Наименование ключа шифрования. Не может быть изменено пользователем. Допустимы символы Unicode.  Используется для идентификации ключа шифрования в визуальных интерфейсах пользователя. |
| KEYDESC | NVARCHAR2(256)  NULL | Описание ключа шифрования. Допустимы символы Unicode.  Используется для подробного описания ключа шифрования. |
| ENCRID | RAW(2000)  NOT NULL | Значение идентификатора ключа шифрования, зашифрованное с использованием этого же ключа шифрования по алгоритму, для которого создан ключ. В случае алгоритма, требующего начальную синхропосылку, выбирается нулевая.  Используется для проверки целостности ключа. |

Табл. Г- 3 – Криптографические копии ключей шифрования пользователей (USERKEYS).

|  |  |  |
| --- | --- | --- |
| Колонка | Тип, свойства | Описание |
| ID | NUMBER(12)  NOT NULL  Primary Key | Идентификатор ключа пользователя.  Положительное целое число. |
| KEYID | NUMBER(12)  NOT NULL  Foreign Key (KEYSET) | Идентификатор ключа.  Положительное целое число. |
| USERID | NUMBER(12)  NOT NULL  Foreign Key (USERCERTS) | Идентификатор сертификата пользователя, открытый ключ которого использован для данной копии. |
| KEYVALUE | VARCHAR2(4000)  NOT NULL | Ключ шифрования, зашифрованный на открытом ключе сертификата пользователя (криптографическая копия ключа шифрования пользователя).  Формат CMS (RFC-3852, RFC-4490). Закодировано в Base-64.  Используется процедурами вычисления ключа шифрования на стороне клиента. |
| ATTRVALUE | RAW(1)  NOT NULL | Атрибут метки безопасности ключа. |
| CHECKVALUE | RAW(64)  NOT NULL | Контрольная сумма строки. Хэш (SHA-1 или GOST34.11-94) от конкатенации значений:  Ключ шифрования  KEYID  ATTRVALUE |

Табл. Г- 4 – Зашифрованные колонки (ENCOLUMNS).

| Колонка | Тип, свойства | Описание |
| --- | --- | --- |
| ID | NUMBER(24)  NOT NULL  Primary Key | Идентификатор зашифрованной колонки.  Положительное целое число. |
| KEYID | NUMBER(12)  NOT NULL  Foreign Key (KEYSET) | Идентификатор ключа.  Положительное целое число. |
| SCHEMANAME | VARCHAR2(64)  NOT NULL | Имя схемы, в которой создана таблица. |
| TABLENAME | VARCHAR2(128)  NOT NULL | Имя таблицы, содержащей зашифрованную колонку. |
| COLNAME | VARCHAR2(128)  NOT NULL | Имя зашифрованной колонки. |
| COLTYPE | VARCHAR2(128)  NOT NULL | Оригинальный (до зашифрования) тип колонки. |
| COLLENGTH | NUMBER(6)  NOT NULL | Оригинальный (до зашифрования) размер колонки. |
| NULLABLE | NUMBER(1)  NOT NULL | Оригинальное (до зашифрования) ограничение целостности NULL/NOT NULL для колонки. |
| DEFVALUE | VARCHAR2(4000)  NULL | Оригинальное (до зашифрования) default-значение колонки. |
| ENCRVIEW | CLOB  NULL | Шаблон маскирующего значения. |
| GROUPNAME | VARCHAR2(64)  NULL | Имя группы пользователей. Зарезервировано.  Заполняется значением 'USERS'. |
| RLEVEL | NUMBER(4)  NOT NULL | Метка данных (чтение). |
| WLEVEL | NUMBER(4)  NOT NULL | Метка данных (запись). |
| CHECKVALUE | RAW(64)  NOT NULL | Контрольная сумма данной строки. Значение хэш-функции (SHA-1/GOST34.11-94), от конкатенации значений:  Ключ шифрования  SCHEMANAME  TABLENAME  COLNAME  RLEVEL  WLEVEL. |

Табл. Г- 5 – Журнал служебных событий (LOGTABLE).

| Колонка | Тип, свойства | Описание |
| --- | --- | --- |
| ID | NUMBER(12)  NOT NULL | Идентификатор записи о событии.  Положительное целое число. |
| TS | VARCHAR2(24)  NOT NULL | Штамп времени. |
| SID | VARCHAR2(12)  NOT NULL | Идентификатор сессии пользователя. |
| INFO | VARCHAR2(64)  NOT NULL | Дополнительная информация о завершении события. |
| CALLER | VARCHAR2(64)  NOT NULL | Процедура, вызвавшая событие. |
| MSG | VARCHAR2(4000)  NOT NULL | Полная информация о событии (описание). |

Табл. Г- 6 – Сообщения (MESSAGES).

|  |  |  |
| --- | --- | --- |
| Колонка | Тип, свойства | Описание |
| ID | VARCHAR2(256)  NOT NULL  PRIMARY KEY | Идентификатор сообщения |
| LANG | VARCHAR2(256)  NOT NULL | Идентификатор языка |
| MSG | VARCHAR2(256)  NOT NULL | Сообщение |

Табл. Г- 7 – Сертификаты пользователей (USERCERTS).

| Колонка | Тип, свойства | Описание |
| --- | --- | --- |
| ID | NUMBER(12)  NOT NULL  Primary Key | Идентификатор сертификата пользователя.  Положительное целое число. |
| CERTVALUE | VARCHAR2(4000)  NULL | Сертификат пользователя.  Формат X.509. Закодировано в Base-64.  Служит для быстрой перегенерации ключей пользователей. |
| REALDN | RAW(2000)  NOT NULL | Идентификатор пользователя, соответствующий значению Subject сертификата, открытый ключ из которого использован для создания копии ключа шифрования. |
| NAME | VARCHAR2(4000)  NOT NULL | Идентификатор пользователя, по которому пользователь может быть идентифицирован в БД. |
| SVRVALUE | VARCHAR2(4000)  NOT NULL | Открытый ключ сервера БД, зашифрованный открытым ключом пользователя.  Формат CMS (RFC-3852, RFC-4490). Закодировано в Base-64.  Используется для безопасной передачи ключа шифрования и проверки подлинности открытого ключа сервера БД. |

Табл. Г- 8 – Последовательности (SEQUENCES).

| Имя | Свойства | Назначение |
| --- | --- | --- |
| S\_PK\_COUNTER | start with 0  increment by 1  maxvalue 4294967295  cycle  cache 256 | Генерация UKM (user keying material) для модификации ключа шифрования в процессе зашифровании колонок (если применимо).  Генерирует значения для каждой зашифровываемой строки. |
| S\_PK\_ENCOLUMNS | start with 1  increment by 1  nomaxvalue  cache 10 | Генерация значений первичного ключа таблицы зашифрованных колонок (ENCOLUMNS). |
| S\_PK\_KEYSET | increment by 1  start with 1  nomaxvalue  nocache | Генерация значений первичного ключа таблицы ключей шифрования (KEYSET). |
| S\_PK\_USERKEYS | increment by 1  start with 1  nomaxvalue  cache 10 | Генерация значений первичного ключа таблицы ключей шифрования пользователей (USERKEYS). |
| S\_PK\_USERCERTS | increment by 1  start with 1  nomaxvalue  cache 10 | Генерация значений первичного ключа таблицы сертификатов пользователей (USERCERTS). |
| S\_PK\_LOGTABLE | start with 1  increment by 1  maxvalue 4294967295  nocycle  cache 10 | Генерация значений первичного ключа таблицы журнала служебных событий (LOGTABLE). |

Приложение Д. Удаление объектов

|  |  |  |
| --- | --- | --- |
| Описание действия | | SQL |
| Удаление существующих объектов (в случае пересоздания) | | drop table %OWNER%.ENCOLUMNS cascade constraints  drop table %OWNER%.USERKEYS cascade constraints  drop table %OWNER%.USERCERTS cascade constraints  drop table %OWNER%.KEYSET cascade constraints  drop table %OWNER%.ALGSET cascade constraints  drop table %OWNER%.LOGTABLE cascade constraints  drop sequence %OWNER%.S\_PK\_ENCOLUMNS  drop sequence %OWNER%.S\_PK\_KEYSET  drop sequence %OWNER%.S\_PK\_COUNTER  drop sequence %OWNER%.S\_PK\_USERKEYS  drop sequence %OWNER%.S\_PK\_USERCERTS  drop sequence %OWNER%.S\_PK\_LOGTABLE  drop procedure %OWNER%.logger  drop package %OWNER%.ncryptoredef  drop package %OWNER%.ncryptoutl  drop package %OWNER%.ncrypto  drop package %OWNER%.ncryptoadmin  drop package %OWNER%.ncryptokey  drop package %OWNER%.ncryptoserver  drop package %OWNER%.ncryptostub  для версии сервера БД 10g/11g дополнительно выполнить:  purge recyclebin |
| Имя | Тип | SQL |
| S\_PK\_ENCOLUMNS | sequence | create sequence S\_PK\_ENCOLUMNS  increment by 1  start with 1  nomaxvalue  cache 10 |
| S\_PK\_KEYSET | sequence | create sequence S\_PK\_KEYSET  increment by 1  start with 1  nomaxvalue  nocache |
| S\_PK\_COUNTER | sequence | create sequence S\_PK\_COUNTER  increment by 1  start with 0  maxvalue 4294967295  cycle  cache 256 |
| Имя | Тип | SQL |
| S\_PK\_USERKEYS | sequence | create sequence S\_PK\_USERKEYS  increment by 1  start with 1  nomaxvalue  cache 10 |
| S\_PK\_USERCERTS | sequence | create sequence S\_PK\_USERCERTS  increment by 1  start with 1  nomaxvalue  cache 10 |
| S\_PK\_LOGTABLE | sequence | create sequence S\_PK\_LOGTABLE  increment by 1  start with 1  minvalue 1  maxvalue 4294967295  nocycle  cache 10 |
| ALGSET | table | create table %OWNER%.ALGSET (  ALGNAME NVARCHAR2(128) not null,  ALGDESC NVARCHAR2(1024) null,  KEYLEN NUMBER(10) not null,  constraint PK\_ALGSET primary key (ALGNAME)  using index  tablespace %TABLESPACE%  )  tablespace %TABLESPACE% |
| KEYSET | table | create table %OWNER%.KEYSET (  KEYID NUMBER(12) not null,  ALGNAME VARCHAR(128) not null,  KEYFN NVARCHAR2(64) not null,  KEYDESC NVARCHAR2(256) null,  ENCRID RAW(64) null,  constraint PK\_KEYSET primary key (KEYID)  using index  tablespace %TABLESPACE%,  constraint FK\_KEYSET\_ALGSET foreign key (ALGNAME) references %OWNER%.ALGSET (ALGNAME)  )  tablespace %TABLESPACE% |
| IDX\_KEYSET\_ALGNAME | index | create index IDX\_KEYSET\_ALGNAME on %OWNER%.KEYSET ( ALGNAME ASC)  tablespace %TABLESPACE% |
| UQ\_KEYSET\_KEYFN | index | create unique index UQ\_KEYSET\_KEYFN on %OWNER%.KEYSET (KEYFN ASC)  tablespace %TABLESPACE% |
| Имя | Тип | SQL |
| LOGTABLE | table | create table %OWNER%.LOGTABLE  (  ID number(12) not null,  ts varchar2(24) not null,  sid number(12) not null,  status varchar2(64) null,  caller varchar2(64) null,  msg varchar2(4000) null  )  tablespace %TABLESPACE% |
| USERKEYS | table | create table %OWNER%.USERKEYS (  ID NUMBER(12) not null,  KEYID NUMBER(12) not null,  USERID NUMBER(12) not null,  KEYVALUE VARCHAR2(4000) not null,  constraint PK\_USERKEYS primary key (ID)  using index tablespace %TABLESPACE%,  constraint FK\_USERKEYS\_KEYSET foreign key (KEYID)  references KEYSET (KEYID),  constraint FK\_USERKEYS\_USERCERTS foreign key (USERID)  references USERCERTS (ID)  )  tablespace %TABLESPACE% |
| IDX\_USERKEYS\_USERKEY | index | create unique index IDX\_USERKEYS\_USERKEY on %OWNER%.USERKEYS (  KEYID ASC,  USEID ASC)  tablespace %TABLESPACE% |
| USERCERTS | table | create table %OWNER%.USERCERTS (  ID NUMBER(12) not null,  CERTVALUE VARCHAR2(4000) null,  REALDN RAW(2000) not null,  NAME VARCHAR2(4000) not null,  SVRVALUE VARCHAR2(4000) not null,  constraint PK\_USERCERTS primary key (ID)  using index tablespace %TABLESPACE%  )  tablespace %TABLESPACE% |
| Имя | Тип | SQL |
| IDX\_USERCERTS\_USECERT | index | create unique index IDX\_USERCERTS\_USERCERT on %OWNER%.USERKEYS (REALDN ASC)  tablespace %TABLESPACE% |
| ENCOLUMNS | table | create table %OWNER%.ENCOLUMNS (  ID NUMBER(24) not null,  KEYID NUMBER(12) not null,  SCHEMANAME VARCHAR(64) not null,  TABLENAME VARCHAR2(128) not null,  COLNAME VARCHAR2(128) not null,  COLTYPE VARCHAR(12) not null,  COLLENGTH NUMBER(6) not null,  COLPREC NUMBER(6) not null,  COLSCALE NUMBER(4) null,  NULLABLE NUMBER(1) not null,  DEFVALUE VARCHAR2(4000) null,  ENCRVALUE CLOB null,  GROUPNAME VARCHAR2(64) null,  constraint PK\_ENCOLUMNS primary key (ID)  using index  tablespace %TABLESPACE%,  constraint FK\_ECOLUMNS\_KEYSET foreign key (KEYID) references %OWNER%.KEYSET (KEYID)  )  tablespace %TABLESPACE% |
| IDX\_ENCOLUMNS\_KEYID | index | create unique index IDX\_ENCOLUMNS\_KEYID on %OWNER%.ENCOLUMNS (  SCHEMANAME ASC,  TABLENAME ASC,  COLNAME ASC,  KEYID ASC,  GROUPNAME ASC)  tablespace %TABLESPACE% |
| TD\_ENCOLUMNS\_AD,  TD\_USERKEYS\_AD,  TI\_ENCOLUMNS\_AI,  TI\_USERKEYS,  TI\_USERKEYS\_AI | trigger | См.п.п. 8.4 8). |
| udpclient | java source | Перед установкой пакеты специальным образом подготавливаются. См. п. 8.4. |
| ncryptoserver | Package |
| ncryptoadmin | Package,  Package body |
| ncryptoutl | Package,  Package body |
| Имя | Тип | SQL |
| ncryptostub | Package,  Package body |  |
| ncrypto | Package,  Package body |
| ncryptokey | Package,  Package body |
| ncryptoredef | Package,  Package body |

Приложение Е. Исходные данные

Табл. Е- 1 Исходные данные для операции зашифрования

| Параметр | Наименование | Значение/Источник/Метод получения параметра |
| --- | --- | --- |
| %OWNER% | Схема владельца | select sys\_context(‘userenv’,’session\_user’)  from dual |
| %SCHEMA% | Целевая схема | Выбирается пользователем (администратором безопасности) из списка доступных схем:  select username  from all\_users  where username not in (‘SYS’,’SYSTEM’) |
| %ORIGTABLE% | Целевая таблица | Выбирается пользователем (администратором безопасности) из списка доступных таблиц целевой схемы:  select table\_name  from all\_tables  where owner = %SCHEMA%  and table\_name not like ‘%%SUFFIX%’ |
| %SUFFIX% | Суффикс для переименования таблицы | Задается в настройках утилит администратора для всех схем экземпляра БД. |
| %COLNAME% | Колонка целевой таблицы (для зашифрования) | Выбирается пользователем (администратором безопасности) из списка доступных колонок целевой таблицы:  select column\_name, data\_type, data\_length, data\_precision, data\_scale, nullable  from all\_tab\_columns  where owner = %SCHEMA%  and table\_name = %ORIGTABLE%  and column\_type in  (‘NUMBER’,’CHAR’,’NCHAR’,’VARCHAR2’,’NVARCHAR2’,’RAW’, ‘BLOB’, ‘CLOB’, ‘DATE')  order by COLUMN\_ID |
| %COLTYPE% | Тип колонки целевой таблицы (для зашифрования) | Соответствует значению data\_type. Cм. выше |
| %DST\_COLTYPE% | Тип данных зашифровываемой колонки | == BLOB для %COLNAME% in (BLOB,CLOB)  == RAW для всех остальных типов |
| %COL\_PREC% | Точность колонки (только для NUMBER) | Соответствует значению data\_precision. Cм. выше |
| %COL\_SCALE% | Шкала колонки (только для NUMBER) | Соответствует значению data\_scale. Cм. выше |
| %COL\_LENGHT% | Размер колонки | Соответствует значению data\_length. Cм. выше |
| %NULL\_NOTNULL% | Параметр ограничения целостности целевой колонки (NULL/NOT NULL) | = TRUE, если nullable == ‘N’  = FALSE – в ином случае  См. выше |
| %ALGNAME% | Имя-идентификатор алгоритма шифрования | Выбирается пользователем (администратором безопасности) из списка доступных алгоритмов:  select algname  from algset |
| %FINAL\_LENGTH% | Размер зашифрованной колонки | Вычисляется функцией пакета  NCRYPTOUTL.get\_final\_size  (alg => ‘%ALGNAME%’,  datatype => ‘%COLTYPE%’,  origlen => %COL\_LENGTH%  ) |
| %ECOLUMN\_ID% | Идентификатор защищаемой колонки | Вычисляется для каждой защищаемой колонки как значение последовательности  s\_pk\_encolumns.nextval |
| %KEYID% | Идентификатор ключа шифрования | Выбирается пользователем (администратором безопасности) из списка доступных ключей для выбранного алгоритма:  select keyid  from keyset  where algname = %ALGNAME% |
| %IDX% | Признак наличия индекса на колонке целевой таблицы | TRUE - нужен поисковый индекс, FALSE - не нужен  Выбирается пользователем (администратором безопасности) |
| %type% | Суффикс-модифика-тор имен функций пакета | 'char' - для CHAR/VARCHAR/VARCHAR2,  'numb' - для NUMBER,  'raw' - для RAW  'date' - для DATE  'lob' - для BLOB/CLOB  типов колонки целевой таблицы |
| %COL\_LIST% | Результирующий список колонок модельной таблицы | Формируется как список имен всех колонок целевой таблицы, разделенных запятыми. Имена оригинальных колонок, которые не зашифровываются, остаются неизменными, имена зашифрованных колонок меняются на:  Cast(  %OWNER%.NCRYPTOSTUB.ENCRYPT  (  ‘%ALGNAME%’,F(%COLNAME%),%KEYID%  as %DST\_COLTYPE%[(%FINAL\_LENGTH%)]  )  %COLNAME%,  где F – функция преобразования данных в тип RAW:  utl\_raw.cast\_to\_raw - для CHAR/VARCHAR/VARCHAR2  utl\_raw.cast\_from\_number – для NUMBER  пусто – для RAW/BLOB/CLOB  Для длинных (4000 > %FINAL\_LENGTH% > 2000) символьных (CHAR/VARCHAR/VARCHAR2/RAW) формируется дополнительная колонка:  Cast(  %OWNER%.NCRYPTOADMIN.EXTRACT\_FROM\_BLOB(  %COLNAME%$$BLOB(1,2000) as RAW(2000)) %COLNAME%,  Cast(  %OWNER%.NCRYPTOADMIN.EXTRACT\_FROM\_BLOB(  %COLNAME%$$BLOB(2001,%FINAL\_LENGTH%-2000)  as RAW(%FINAL\_LENGTH%-2000)) %COLNAME%$$1,  Для длинных (%FINAL\_LENGTH% > 4000) символьных (CHAR/VARCHAR/VARCHAR2/RAW) формируются дополнительные колонки:  Cast(  %OWNER%.NCRYPTOADMIN.EXTRACT\_FROM\_BLOB(  %COLNAME%$$BLOB(1,2000) as RAW(2000)) %COLNAME%,  Cast(  %OWNER%.NCRYPTOADMIN.EXTRACT\_FROM\_BLOB(  %COLNAME%$$BLOB(2001, 2000)  as RAW(2000)) %COLNAME%$$1,  Cast(  %OWNER%.NCRYPTOADMIN.EXTRACT\_FROM\_BLOB(  %COLNAME%$$BLOB(2001,%FINAL\_LENGTH%-4000)  as RAW(%FINAL\_LENGTH%-4000)) %COLNAME%$$2 |
| %ENCRYPT\_ONLY% | Флаг создания промежуточных представле-ний и триггеров | Задается в консоли администратора безопасности, как пункт выбора интерфейса шифрования таблиц.  = FALSE - создать промежуточные представления и триггеры  = TRUE – не создавать |
| %MASK\_FMT% | Формат маскирую-щего значения | Задается в консоли администратора безопасности, как строка. |
| Для %DST\_COLTYPE% in (BLOB,CLOB) подстрока (%FINAL\_LENGTH%) должна быть опущена. | | |

Табл. Е- 2 Исходные данные для операции расшифрования

| Параметр | Наименование | Значение/Источник/Метод получения |
| --- | --- | --- |
|  |  |  |
| %OWNER% | схема владельца | **select sys\_context(‘userenv’,’session\_user’)**  **from dual** |
| %SCHEMA% | целевая схема | Выбирается пользователем (администратором безопасности) из списка доступных схем:  **select schemaname**  **from %OWNER%.encolumns** |
| %ORIGTABLE% | целевая таблица | Выбирается пользователем (администратором безопасности) из списка зашифрованных таблиц целевой схемы:  **select tablename**  **from %OWNER%.encolumns**  **where schemaname = %SCHEMA%** |
| %COLNAME% | Колонки (зашифрованные) целевой таблицы | Выбираются пользователем (администратором безопасности) из списка доступных колонок целевой таблицы:  **select**  **colame,**  **coltype,**  **colprec,**  **collength,**  **colscale,**  **keyid,**  **nullable,**  **id**  **from %OWNER%.encolumns**  **where schemaname = %SCHEMA%**  **and tablename = %ORIGTABLE%** |
| %COLTYPE% | тип колонок целевой таблицы (в оригинальной таблице) |
| %COL\_PREC% | точность колонок (в оригинальной таблице – для NUMBER) |
| %COL\_SCALE% | шкала колонок (в оригинальной таблице – для NUMBER) |
| %COL\_LENGHT% | размер колонок (в оригинальной таблице) |
| %KEYID% | идентификаторы ключей шифрования |
| %ID% | Идентификаторы зашифрованной колонки |
| %NULL\_NOTNULL% | Ограничение null/not null |
| %ECOLUMN\_ID% | идентификатор защищаемой колонки |
| %ALGNAME% | имя-идентификатор алгоритма шифрования | Выбирается пользователем (администратором безопасности) из списка доступных алгоритмов:  **select algname**  **from keyset**  **where keyid = %KEYID%** |
| %SUFFIX% | суффикс для переименования таблицы | Задается в настройках утилит администратора для всех схем экземпляра БД. |
| %type% | суффикс-модификатор имен функций пакета | '**char**' - для CHAR/VARCHAR/VARCHAR2,  '**numb**' - для NUMBER,  '**lob**' - для BLOB/CLOB  '**raw**' - для RAW типов колонки целевой таблицы) |
| %COL\_LIST% | Результирующий список колонок модельной таблицы | Формируется как список имен всех колонок целевой таблицы, разделенных запятыми. Имена оригинальных колонок, которые не зашифровываются, остаются неизменными, имена зашифрованных колонок меняются на:  **Cast(**  **%OWNER%.NCRYPTOSTUB.DECRYPT**  **(**  **‘%ALGNAME%’,F(%COLNAME%),%KEYID%**  **as %DST\_COLTYPE%[(%FINAL\_LENGTH%)]**  **)**  **%COLNAME%,**  где F – функция преобразования данных в тип RAW:  **utl\_raw.cast\_to\_raw** - для CHAR/VARCHAR/VARCHAR2  **utl\_raw.cast\_from\_number** – для NUMBER  пусто – для RAW/BLOB/CLOB |

Приложение Ж. Шаблоны маскирующих значений.

|  |  |  |
| --- | --- | --- |
| Тип зашифрованной колонки | Шаблон | Результат маскирования |
| CHAR/VARCHAR/  CLOB/BLOB/RAW | ‘string’ | Указанная строка символов.  Может содержать произвольное количество макроподстановок вида:  %N – номер записи в текущей выборке;  %D(N) – подставляется произвольное число, не превышающее N;  %S(N) – подставляется произвольная строка символов, длиной не более N;  Общая длина результирующего значения усекается до размера колонки. |
| NULL | NULL |
| NUMBER | NULL | NULL |
| ‘%константа’ | Значение указанной константы.  Точность усекается до значения шкалы колонки.  Неверное задание значения шаблона вызывает исключение -20011 (Invalid mask). |
| ‘X-Y’ | Случайное значение в интервале X…Y включительно.  X,Y – вещественные числа.  Точность усекается до значения шкалы колонки.  Неверное задание значения шаблона (X > Y, либо величины X или Y превышают разрядность колонки) вызывает исключение -20011 (Invalid mask). |
| DATE | NULL | NULL |
| ‘%SD’ | Значение системной даты (SYSDATE). |
| 'YYYYMMDDHH24MISS-YYYYMMDDHH24MISS' | Случайная дата в заданном интервале дат.  YYYY – четыре цифры года  MM – две цифры месяца (01..12)  DD – две цифры дня (01..31)  HH24 – две цифры часов (00..23)  MI - две цифры минут (00..59)  SS - две цифры секунд (00..59).  Неверное задание значения шаблона вызывает исключение -20011 (Invalid mask). |

Приложение З. Примеры записей аудита

**Первичные записи аудита**

**A S 128 12 133 S**

**A S 128 12 133 S**

**A U 131 12 134 S AAATpAAAEAAAADdAAA**

**…**

**M GK 1 133 S NULL petrov:hr:::10.1.1.168:edu**

**Записи файла аудита**

**2009-09-21 19:11:34 A S 128 TEST:TABLE\_A1:COL2 CN=USER1,OU=FINANCE G NULL**

**2009-09-21 19:11:34 A S 128 TEST:TABLE\_A1:COL2 CN=USER1,OU=FINANCE G NULL**

**2009-09-21 19:11:38 A U 131 TEST:TABLE\_A1:COL2 CN=MANAGER,OU=FINANCE G NULL**

**…**

**2009-09-21 22:12:09 M UC 145 KEY\_FIN\_DEP1 CN=USER1,OU=FINANCE S NULL**

**2009-09-21 22:18:39 M UR 145 KEY\_FIN\_DEP1 CN=MANAGER,OU=FINANCE S NULL**

**…**

**Структура записи аудита (события доступа):**

A<код-операции>:

<код-зашифрованной-колонки>:

<код-копии-ключа-пользователя>:

<валидность-операции>

Где возможны следующие значения:

<код-операции> = 1-select | 2-insert | 3-update | 4-delete

<валидность-операции> = S – успешно, | F – неуспешно (нет ключа)

Перечень терминов и сокращений

|  |  |  |
| --- | --- | --- |
| АДМБ | – | Администратор безопасности. |
| АРМ | – | Автоматизированное рабочее место. |
| БД | – | База данных. |
| ИУ | – | Идентифицирующее устройство. |
| Ключ eToken  (USB или SC) | – | Персональное программно-аппаратное средство (устройство) аутентификации и хранения данных. |
| Ключ шифрования данных (КШД) | – | Случайное значение, используемое в симметричных алгоритмах шифрования при зашифровании/расшифровании информации в колонках таблиц, с помощью системы. |
| КС | – | Контрольная сумма. |
| КЦ | – | Контроль целостности. |
| НСД | – | Несанкционированный доступ. |
| ОС | – | Операционная система. |
| ПК | – | Программный комплекс. |

Перечень ссылочных документов

1. СКЗИ «Крипто БД». Руководство администратора базы данных.
2. ГОСТ 28147‑89 «Системы обработки информации. Защита криптографическая».
3. ГОСТ Р 34.10-94. Государственный стандарт Российской Федерации. Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма.
4. ГОСТ Р 34.10-2001. Государственный стандарт Российской Федерации. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи.
5. ГОСТ Р 34.11-94. Государственный стандарт Российской Федерации. Информационная технология. Криптографическая защита информации. Функция хэширования.
6. RFC 4357 «Additional Cryptographic Algorithms for Use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms».

RFC 4490 «Using the GOST 28147-89, GOST R 34.11-94, GOST R 34.10-94, and GOST R 34.10-2001 Algorithms with Cryptographic Message Synta

1. Два SP указаны для разных вариантов ОС. [↑](#footnote-ref-1)