

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К.И.Сатпаева

Институт информационных и телекоммуникационных технологий

Кафедра “Кибербезопасность, обработка и хранения информации”

Агыбай Алишер Багадатулы

Шифрование в реляционных серверах баз данных

ДИПЛОМНЫЙ ПРОЕКТ

Специальность 5В100200 – Системы информационной безопасности

Алматы 2019


КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ имени К.И. САТПАЕВА

СЭТБАЕВ
УНИВЕРСИТЕТИ



ИНСТИТУТ ИНФОРМАЦИОННЫХ И
ТЕЛЕКОММУНИКАЦИОННЫХ
ТЕХНОЛОГИЙ

КАФЕДРА КИБЕРБЕЗОПАСНОСТЬ,
ОБРАБОТКА И ХРАНЕНИЕ ИНФОРМАЦИИ

«Допущен к защите»
Заведующий кафедрой КБОиХИ
 Н.А.Сеилова

ДИПЛОМНЫЙ ПРОЕКТ

на тему: «ШИФРОВАНИЕ В РЕЛЯЦИОННЫХ СЕРВЕРАХ БАЗ ДАННЫХ»

по образовательной программе 5В100200 – «Системы информационной
безопасности»

Выполнил выпускник

Агыбай А.Б.

Научный руководитель

к.т.н., ассоц. проф. Айтхожаева Е.Ж.

Алматы 2019

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ
КАЗАХСТАН

Казахский национальный исследовательский технический университет
имени К.И.Сатпаева


Институт информационных и телекоммуникационных технологий

Кафедра "Кибербезопасность, обработка и хранение информации"

5В100200 - Системы информационной безопасности

УТВЕРЖДАЮ

Заведующий кафедрой КБОиХИ
канд. техн. наук, доцент

 Н.А.Сейлова
" 13 " 05 2019 г.

ЗАДАНИЕ

на выполнение дипломного проекта

Обучающемуся Азыбай Әлішеру Бағдатұлы

Тема: Шифрование в реляционных серверах баз данных

Утверждена приказом Ректора Университета № 1162-б от «16» 10 2018 г.

Срок сдачи законченной работы «__» __ 2019 г.

Исходные данные к дипломному проекту: изучение методов шифрования в серверах БД, прозрачное шифрование в Oracle и MS SQL Server, проектирование, реализация и шифрование БД в MS SQL Server.

Перечень подлежащих разработке в дипломном проекте вопросов:

а) криптография для защиты баз данных

б) прозрачное шифрование в Oracle

в) прозрачное шифрование в MS SQL Server

Перечень графического материала (с точным указанием обязательных чертежей):

методы шифрования в серверах баз данных – 1А3; прозрачное шифрование в Oracle – 1А3; диаграмма и таблицы базы данных - 1А3; шифрование в MS SQL Server - 1А3.

Рекомендуемая основная литература: из 13 наименований


ГРАФИК

подготовки дипломного проекта

| Наименования разделов, перечень разрабатываемых вопросов | Сроки представления научному руководителю | Примечание |
|--|---|------------|
| Криптография для защиты баз данных | 14.01.19-20.02.19 | восп |
| Прозрачное шифрование в Oracle | 20.02.19-15.03 | восп |
| Прозрачное шифрование в MS SQL Server | 15.03.19-30.04 | восп |

Подписи

консультантов и нормоконтролера на законченный дипломный проект с указанием относящихся к ним разделов проекта

| Наименования разделов | Консультанты, И.О.Ф. (уч. степень, звание) | Дата подписания | подпись |
|-----------------------|--|-----------------|--|
| Нормоконтролер | Зиро А.А. (магистр тех.наук, лектор) | 13.05.19 |  |

Научный руководитель



Е.Ж.Айтхожаева

Задание принял к исполнению обучающийся



А.Б. Агыбай

Дата

« 14 » 01 2019 г.

ОТЗЫВ

НАУЧНОГО РУКОВОДИТЕЛЯ

на дипломный проект

(наименование вида работы)

Ағыбай Ә.Б.

(Ф.И.О. обучающегося)

5В100200 - Системы информационной безопасности

(шифр и наименование специальности)

Тема:

Шифрование в реляционных серверах баз данных

Базы данных широко применяются во всех областях деятельности человека в качестве информационного ядра информационных систем, предназначенного для хранения данных общего пользования, и обеспечения ранжированного многопользовательского доступа к ним. Использование методов шифрования позволяет обеспечить надежную защиту БД, в том числе и от недобросовестного администратора. В отдельных случаях являются единственным средством обеспечения безопасности информации в базах данных.

В работе студента Ағыбай Ә.Б. «Шифрование в реляционных серверах баз данных» рассматриваются методы шифрования в современных серверах БД, использование прозрачного шифрования в Oracle, ставится и решается задача проектирования и защиты серверной БД с использованием прозрачного шифрования в широко распространенном сервере MS SQL Server 2012.

Ағыбай Ә.Б. проявил инициативу и самостоятельность, как в постановке, так и решении задачи, хорошие инженерные навыки в области анализа методов шифрования, проектирования и реализации защищенных баз данных с использованием методов шифрования, умение самостоятельно осваивать современный ИТ-инструментарий, умение работать с технической литературой.

Работа выполнена с использованием современных ИТ технологий: CASE-средства проектирования баз данных All Fusion Erwin Data Modeler, сервера БД Oracle и сервера БД MS SQL Server 2012.

Дипломный проект на тему «Шифрование в реляционных серверах баз данных» выполнен Ағыбай Ә.Б. на хорошем уровне и может быть допущен к защите.

Научный руководитель

ассоц.профессор, к.т.н.

(должность, уч. степень, звание)

Айтхожаева Е.Ж.

(подпись)

« 8 » 05 2019 г.

РЕЦЕНЗИЯ

на _____ дипломный проект
(наименование вида работы)

Ағыбай Ә.Б.
(Ф.И.О. обучающегося)

5В100200
(шифр и наименование специальности)

На тему: Шифрование в реляционных серверах баз данных

Выполнено:

- а) графическая часть на _____ 4 _____ листах
- б) пояснительная записка на 37 страницах

ЗАМЕЧАНИЯ К РАБОТЕ

В современном цифровом мире вопросы обеспечения информационной безопасности приобрели первостепенное значение. Критическая информация в основном хранится в базах данных (БД), взлом и хищение которых приобрели угрожающие размеры, что и определяет актуальность темы дипломного проекта.

Работа посвящена организации рассмотрению методов шифрования в современных серверах БД, оформлена аккуратно, хорошо структурирована: представленный материал разбит по главам, имеются оглавление, введение, заключение, список литературы, приложения.

В первой теоретической главе проводится рассмотрение и анализ применения методов шифрования для защиты баз данных, рассматриваются встроенные механизмы шифрования в серверах БД.

Вторая и третья главы посвящены решению практических задач. Рассматривается и решается задача применения прозрачного шифрования БД в сервере БД Oracle, а также проектирование и организация базы данных в сервере БД MS SQL Server 2012 и ее защита методом прозрачного шифрования. Используются современные информационные технологии БД: Oracle 11g, MS SQL Server 2012, Erwin Data Modeler 7.2.

Приложения содержат программный код SQL и графическую часть, отражают содержание дипломного проекта.

Пояснительная записка к дипломному проекту и графическая часть выполнены в соответствии с требованиями стандарта КазННТУ им. К.И.Сатпаева.

Оценка работы

Рецензируемый дипломный проект выполнен на актуальную тему и удовлетворяет требованиям, предъявляемым к дипломным проектам. Дипломный проект демонстрирует знание дипломником предметной области и умение решать задачи шифрования, проектирования и обеспечения безопасности баз данных с помощью методов шифрования.

Считаю, что дипломный проект Ағыбай Ә.Б. на тему: «Шифрование в реляционных серверах баз данных» заслуживает оценки отлично (А), а Ағыбай Ә.Б. - присвоения академической степени бакалавра по специальности 5В100200 – Системы информационной безопасности.

Рецензент

Канд. Тех. Наук Академик МАИН

Профессор кафедры СИБ АУиЭС

Д. Дымбаев С.Т.

« 09 »

2019





| | |
|---|-------------------------------|
| Университет: | Satbayev University |
| Название: | Шифрование в реляционных СУБД |
| Автор: | Агыбай Ә.Б. |
| Координатор: | Евгения Айтхожаева |
| Дата отчета: | 2019-05-06 08:27:48 |
| Коэффициент подобия № 1: | 24,2% |
| Коэффициент подобия № 2: | 1,7% |
| Длина фразы для коэффициента подобия № 2: | 25 |
| Количество слов: | 4 627 |
| Число знаков: | 34 748 |
| Адреса пропущенные при проверке: | |
| Количество завершенных проверок: | 46 |

! К вашему сведению, некоторые слова в этом документе содержат буквы из других алфавитов. Возможно - это попытка скрыть позаимствованный текст. Документ был проверен путем замещения этих букв латинским эквивалентом. Пожалуйста, уделите особое внимание этим частям отчета. Они выделены соответственно.

Количество выделенных слов 2

>> Самые длинные фрагменты, определенные, как подобные

| № | Название, имя автора или адрес гиперссылки (Название базы данных) | Автор | Количество одинаковых слов |
|----|---|------------------------------|----------------------------|
| 1 | URL_ http://www.citforum.ck.ua/database/oracle/tde/ | | 44 |
| 2 | URL_ http://www.itshop.ru/Shifrovanie-v-bazah-dannyh-SQL-Server/19/36233 | | 35 |
| 3 | Криптографические методы защиты информации в базах данных Azerbaijan Technical University (ATU) (Informasiya texnologiyalari və proqramlaşdırma) | Hüseynli Adəm Əliheydər oğlu | 17 |
| 4 | URL_ https://kbss.ru/blog/bd_mysql/304.html | | 17 |
| 5 | URL_ http://www.citforum.ck.ua/database/oracle/tde/ | | 17 |
| 6 | URL_ https://kbss.ru/blog/bd_mysql/304.html | | 16 |
| 7 | URL_ http://www.citforum.ck.ua/database/oracle/tde/ | | 15 |
| 8 | Криптографические методы защиты информации в базах данных Azerbaijan Technical University (ATU) (Informasiya texnologiyalari və proqramlaşdırma) | Hüseynli Adəm Əliheydər oğlu | 14 |
| 9 | URL_ https://kbss.ru/blog/bd_mysql/304.html | | 14 |
| 10 | URL_ http://www.citforum.ck.ua/database/oracle/tde/ | | 14 |

>> Документы, в которых найдено подобные фрагменты: из RefBooks



Не обнаружено каких-либо заимствований

>> Документы, содержащие подобные фрагменты: Из домашней базы данных

Не обнаружено каких-либо заимствований

>> Документы, содержащие подобные фрагменты: Из внешних баз данных

Документы, выделенные жирным шрифтом, содержат фрагменты потенциального плагиата, то есть превышающие лимит в длине коэффициента подобия № 2

| № | Название (Название базы данных) | Автор | Количество одинаковых слов (количество фрагментов) |
|---|---|------------------------------|---|
| 1 | Криптографические методы защиты информации в базах данных Azerbaijan Technical University (ATU) (Informasiya texnologiyalari və proqramlaşdırma) | Hüseynli Adəm Əliheydər oğlu | 188 (22) |
| 2 | Analiza metod zabezpieczenia danych w środowisku MS SQL Server 2016 na przykładzie bazy Adventure Works 2016 Uniwersytet Ekonomiczny w Katowicach (Wydział Informatyki i Komunikacji UE Katowice) | Magdalena Wiktoria Krochmal | 18 (2) |

>> Документы, содержащие подобные фрагменты: Из интернета

| № | Источник гиперссылки | Количество одинаковых слов (количество фрагментов) |
|---|--|--|
| 1 | URL_ http://www.itshop.ru/Shifrovanie-v-bazah-dannyh-SQL-Server/19i36233 | 272 (33) |
| 2 | URL_ http://www.citforum.ck.ua/database/oracle/tde/ | 264 (26) |
| 3 | URL_ https://wiki2.org/ru/%D0%A8%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1% | 151 (21) |
| 4 | URL_ https://kbss.ru/blog/bd_mysql/304.html | 89 (9) |
| 5 | URL_ https://docplayer.ru/39181467-Molodoy-uchyonyy-chast-ii.html | 86 (9) |
| 6 | URL_ http://support.infobase.com/index.php?videolearn360/Knowledgebase/Article/View/1700/0/search-basics---searching-for-non-video-content | 51 (8) |

Протокол анализа Отчета подобия

заведующего кафедрой / начальника структурного подразделения

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Ағыбай Э.Б.

Название: Шифрование в реляционных СУБД

Координатор: Евгения Айтхожаева

Коэффициент подобия 1:24,2

Коэффициент подобия 2:1,7

Тревога:2

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование:

Заимствования являются добросовестными

Дата *13.05.19*

Подпись заведующего кафедрой / *[подпись]*

начальника структурного подразделения

[подпись]

Окончательное решение в отношении допуска к защите, включая обоснование:

Допущен к защите

Дата 12.05.09

Подпись заведующего кафедрой /

начальника структурного подразделения


12.05.09

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Ағыбай Ә.Б.

Название: Шифрование в реляционных СУБД

Координатор: Евгения Айтхожаева

Коэффициент подобия 1: 24,2

Коэффициент подобия 2: 1,7

Тревога: 2

После анализа Отчета подобия констатирую следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

Обнаруженные зашифрованные являются добросовестными, так как обнаружены исползованием в работе общепринятых словосочетаний в области ИБ и исползованием названий программных продуктов и методов шифрования.

8.05.2019г.

.....*Айхон*.....

Дата

Подпись Научного руководителя

АНДАТПА

Бұл жұмыс ақпаратты қорғаудың криптографиялық әдістерін қолданумен дерекқорларды қорғауға арналған. Oracle және Microsoft компанияларының дерекқор серверлерінде салынған шифрлеудің механизмдері талданады. Кітапханаға арналған дерекқордың жобасы жасалып жүзеге асырылды. Пән аямағының талдауы жасалды. Бұл талдаудың негізінде сущностьтар, тәрсетілімдер және атрибуттары мен олардың арасындағы байланыстары анықталды, логикалық және физикалық деңгейде AllFusion Erwin Data Modeler 7.2 дерекқорды жобалаудың CASE құралында ER-диаграммалары құрылды. Дерекқорларды жүзеге асыру үшін T-SQL тілінде скрипттер өндірілді. MS SQL Server 12 ішіндегі жобаланған дерекқордың жүзеге асыру және оның шифрлеудің мөлдір әдісімен (TDE) қорғау жасалды. Oracle 11g дерекқор серверіндегі шифрлеудің мөлдір әдісін жүзеге асыру жасалды.

АННОТАЦИЯ

Данная работа посвящена реализации защиты базы данных с использованием криптографических методов защиты информации. Проанализированы встроенные механизмы шифрования в серверах баз данных компании Oracle и Microsoft, в том числе прозрачное шифрование.

Спроектирована и реализована база данных для библиотеки. Проведен анализ предметной области. На основе этого анализа определены сущности, представления, атрибуты и связи между ними, построены ER-диаграммы в CASE-средстве проектирования баз данных AllFusion Erwin Data Modeler 7.2 на логическом и физическом уровне. Сгенерированы скрипты на языке T-SQL для реализации базы данных. Выполнена реализация спроектированной базы данных в MS SQL Server 12 и ее защита прозрачным методом шифрования (TDE). Выполнена реализация прозрачного метода шифрования в сервере баз данных Oracle 11g.

ANNOTATION

This paper is devoted to the implementation of database protection using cryptographic methods of information protection. The types of encryption in database servers, including transparent data encryption, are considered. Analyzed the built-in encryption mechanisms in the database servers of Oracle and Microsoft.

Designed and implemented a database for the library. The analysis of the subject area. Based on this analysis, the entities, representations, attributes and relations between them are determined, ER-diagrams are built in CASE-tool design database AllFusion Erwin Data Modeler 7.2 at the logical and physical level. T-SQL scripts for database implementation were generated. The implemented database was implemented in MS SQL Server 12 and protected by the transparent encryption method (TDE). Implemented transparent encryption method in Oracle 11g database server.

СОДЕРЖАНИЕ

| | |
|--|----|
| Введение | 6 |
| 1 Криптография для защиты баз данных | 7 |
| 1.1 Типы шифрования в серверах баз данных | 7 |
| 1.2 Прозрачное шифрование данных | 10 |
| 2 Прозрачное шифрование в Oracle | 12 |
| 2.1 Принцип прозрачного шифрования в Oracle | 12 |
| 2.2 Создание бумажника и шифрование столбцов | 14 |
| 3 Прозрачное шифрование в MS SQL Server | 19 |
| 3.1 Алгоритм прозрачного шифрования | 19 |
| 3.2 Проектирование и реализация БД | 22 |
| 3.3 Шифрование БД | 27 |
| Заключение | 33 |
| Список использованной литературы | 34 |
| Приложение А | 35 |
| Приложение Б | 37 |

ВВЕДЕНИЕ

Современный мир информационных технологий невозможно представить без использования серверов баз данных, так как в настоящее время собрано большое количество информации, которые должны храниться в электронном виде.

Поэтому проблема обеспечения безопасности данных является одной из актуальных. Обеспечение надежности защиты данных стала одной из важнейших проблем в современном мире, так как утечка либо потеря данных, в особенности данные пользователей, может привести к потере авторитета бренда компании, снижению конкурентоспособности, штрафам – и даже к привлечению к ответственности.

По многим требованиям стандартов обеспечения информационной безопасности, где хранится конфиденциальная информация, должна быть защита данных [1].

Среди множества средств защиты БД можно выделить часто встречающиеся.

К основным средствам защиты данных относят следующие: защита с помощью пароля, защита полей и записей таблиц БД, прав доступа к объектам БД, шифрование данных и программ.

Организации или компании могут остановиться на криптографических методах защиты информации - шифрование или на уровне приложения, или на уровне базы данных, или на уровне места хранения. Использование криптографических средств засекречивания информации позволяет предотвратить несколько угроз. Для этой цели используется шифрование данных, т.е. передача и хранение конфиденциальных данных в зашифрованном виде.

Шифрование базы данных добавит дополнительный уровень защиты конфиденциальной информации, однако перед реализацией компания должна учесть, как шифрование повлияет на выполнение обычных операций. Наиболее важный вопрос: управление местонахождением и хранением ключа, который используется для засекречивания и рассекречивания данных. В программе должны быть план по восстановлению забытых ключей и план по восстановлению информации в случае потери информации или удаления ключей. Если в компании имеется инфраструктура открытых ключей (Public Key Infrastructure, PKI), то система PKI может помочь в защите и управлении ключами базы.

Данный проект посвящен изучению криптографических методов защиты информации в серверах баз данных и изучению встроенных механизмов шифрования в серверах БД. Целью дипломного проекта является реализация прозрачного метода шифрования данных при помощи встроенных механизмов шифрования в серверах БД компании Microsoft и Oracle.

1 Криптографическая защита баз данных

1.1 Типы шифрования в серверах баз данных

Средства шифрования в серверах Баз данных присутствовали очень давно. В настоящее время сервера БД поддерживают несколько встроенных механизмов шифрования: шифрование на уровне ячеек, шифрование на уровне файлов, специальные функции шифрования, симметричное шифрование, асимметричное шифрование, шифрование на транспортном уровне, прозрачное шифрование данных – TDE, функции необратимого шифрования [2-6].

Шифрование на уровне столбцов (англ. *Column-Level Encryption*) - позволяет шифровать отдельные столбцы с различными ключами, что может обеспечить дополнительную гибкость при защите. Ключи могут быть выданы пользователям и защищены паролем для предотвращения автоматической расшифровки, но это может усложнить администрирование БД. При использовании метода шифрования на уровне столбцов надо вносить несколько изменений в клиентские приложения. Помимо этого уменьшается производительность БД. Среди достоинств можно отметить, что этот метод позволяет зашифровать всего один столбец внутри таблицы. Кроме того, информация не расшифровывается, пока не придет время их использования, то есть информация из загруженной в память страницы зашифрованы. Можно выделить ключ пользователям и защитить его паролем, чтобы как то предотвратить автоматическую расшифровку.

Среди недостатков шифрования на уровне столбцов можно выделить необходимость внесения изменения в схемы, так как все засекреченные данные должны сохраниться с использованием типа данных «varbinary». Кроме того, наблюдается снижение общей производительности базы данных из-за повторной обработки при шифровании и расшифровке информации в столбцах. Требуется большого времени и просмотров таблицы, поскольку индексы для таблицы зашифрованы и не могут их использовать.

Microsoft использует технологию шифрования файловой системы (англ. *Encrypting File System, EFS*), в которой обеспечивается шифрование на уровне файлов. Каждый объект шифруется с помощью уникального ключа шифрования файлов (англ. *File Encryption Key*), который защищен сертификатом пользователя. Этот сертификат может быть и составным, что дает вероятность получения доступа к файлу больше чем одному из пользователей. Из-за увеличения сферы шифрования, использование шифрования файловой системы снижается производительность и усложняется администрирование, так как системному администратору требуется доступ к операционной системе для использования шифрования файловой системы.

В шифровании на уровне приложений процесс шифрования реализуется приложением, которое изменяет данные, то есть это происходит перед записью в базу данных. Этот подход более гибкий, так как приложение знает роли или права доступа пользователей, а также информация о том, какие данные являются секретными.

Главное преимущество шифрования, которое встроено в приложение, нет необходимости использовать дополнительное решение для защиты информации при передаче по каналам связи, так как, они отправляются уже зашифрованными. Еще один плюс такого метода — это то, что хищение секретной информации становится сложнее, так как злоумышленник не имеет доступа к приложению для того, чтобы расшифровать данные, хранящиеся в БД.

Недостатком шифрования на уровне приложения является то, что для реализации шифрования на уровне приложений необходимо внесение изменений в базу данных в целом. Также может возникнуть проблема с производительностью базы данных, у которой, к примеру, пропадает возможность индексирования и поиска. Еще одним минусом является система управления ключами. Так как не одно приложение, а несколько могут использовать БД, то ключи должны храниться во многих местах, поэтому неправильное управление ключами может привести к утечке информации или невозможно будет ее использования. В дополнение к этому, если возникнет необходимость изменения ключа, то для начала требуется расшифровать все данные со старыми ключами, и потом заново все зашифровать, используя новый ключ.

Шифрование на транспортном уровне. В SQL Server имеется два варианта шифрования данных на транспортном уровне, которые будут передаваться по сети между экземпляром SQL Server приложением клиента.

Ipsec. Реализовывается на уровне операционной системы и проверяет подлинности с использованием метода аутентификации Kerberos, сертификатов и общих ключей. Обеспечивает прозрачные для приложений службы шифрования с надежностью фильтрации для блокировки трафика по протоколам и портам передачи. IPsec возможно настроить с помощью локальной политики безопасности или групповой политики. Выбрав этот метод, надо убедиться, что операционные системы клиентов, сервера совместимы с этим протоколом.

SSL. Настраивается на SQL Serverе, широко применяется для поддержки веб-клиентов, но так же используется и для клиентов SQL Server. SSL обеспечивает проверку сервера, когда клиент запрашивает зашифрованное соединение. Если SQL Server работает на компьютере с сертификатом от публичного удостоверяющего центра, то компьютеры и экземпляр SQL Server гарантируют, что путь сертификатов ведет к корневому центру сертификации. Для обеспечения такой проверки со стороны сервера требуется, чтобы компьютер, на котором работает клиентское приложение, доверял корневому центру, которое используется

сервером. Возможно шифрование с использованием самозаверяющего сертификата, но защита самозаверяющего сертификата ненадежна.

Функции необратимого шифрования (хеширования). Функция MD5() — выполняет необратимое шифрование входящих и выходящих данных по алгоритму MD5 (Message-Digest Algorithm). Функция принимает на вход строку и выдает 128-битную контрольную сумму, которую вычисляет по алгоритму MD5. Значение, которое возвращает это 32-разрядное шестнадцатеричное число, уникальное для каждой строки. Если строки отличаются хотя бы одним символом, то сработает лавинный эффект, то есть результат функции MD5() для этих строк будет разным, но для двух одинаковых строк — результат всегда должен быть одинаков.

Функция PASSWORD() — производит необратимое шифрование данных, эта функция зашифрует пароли пользователей в MySQL, в столбце «password» таблицы привилегий пользователя системной базы данных mysql.

Функция SHA1 — вычисляет 160-битную контрольную сумму с использованием алгоритма SHA1 (Secure Hash Algorithm). Значение, которое возвращает представляет собой 40-разрядное шестнадцатеричное число, либо результат NULL (когда входной параметр равен NULL).

Пример - MySQL, в которой насчитывается около 14 соответствующих функций, для шифрования и дешифрования:

- AES_ENCRYPT(), шифрование с использованием алгоритма AES;
- AES_DECRYPT(), расшифровка с использованием AES;
- COMPRESS(), возвращает результат в бинарном виде;
- DES_ENCRYPT(), шифрование с использованием алгоритма DES;
- DES_DECRYPT(), дешифрование с использованием алгоритма DES;
- ENCODE(), шифрование строки поверхностным паролем (на выходе получается зашифрованное значение первоначальной длиной «plaintext»);
- DECODE(), расшифровка значения, зашифрованного функцией ENCODE();
- ENCRYPT(), шифрование с помощью Unix-ового системного скрипта «crypt»;
- MD5(), подсчет контрольной MD-5 суммы;
- SHA1(), SHA(), подсчет SHA-1 (160-бит).

Выше перечисленные способы шифрования применяют в основном на различных уровнях организации данных. Можно зашифровать отдельные таблицы (сущности), базу данных в целом или же применять шифрование к отдельным столбцам (атрибутам).

Для выбора подходящего варианта для защиты информации нужно следовать нескольким правилам:

- для получения более длинной или сложной цепочки бинарного кода шифрованного текста необходимо применить длинные ключи для шифрования;

- применение в асимметричном шифровании пары ключей (в сравнении с симметричным шифрованием) повышает сложность криптоанализа для злоумышленника, но уменьшает производительность;

- блочные шифры надежны по сравнению с поточными шифрами;

- зашифровать можно сжатые данные, но невозможно сжать зашифрованные данные;

- асимметричное шифрование в большинстве случаев замедляет работу системы, поэтому для шифрования большого количества данных лучше не использовать. Для такой цели хорошо подходит симметричное шифрование;

- следует пользоваться длинными и сложными паролями. Длинные сложные пароли надежнее, чем короткие пароли.

Применяя выше описанные указания, можно подобрать оптимальный вариант шифрования данных.

1.2 Прозрачное шифрование данных

Прозрачное шифрование базы данных (англ. *Transparent Database Encryption*, TDE) — технология, которая применяется в продуктах компании Microsoft и Oracle для шифрования и дешифрования секретных данных и ввода-вывода файлов БД [7-8]. Позволяет зашифровать секретные данные, такие как номера кредитных карт, хранящиеся в таблицах и табличных областях.

Данные шифруются перед записью в память и дешифруются во время чтения из памяти, что может решить проблему защиты «неактивных» данных, но не может обеспечить безопасность информации при передаче по каналам связи или во время использования. Шифрование и дешифрование реализуется на уровне SQL и полностью прозрачен для прикладных программ и пользователей. Резервные копии данных, записанные на диск или магнитную ленту, будут содержать эти данные только в зашифрованном виде. Для предотвращения расшифровки, TDE хранит секретные ключи шифрования в модуле защиты внешне по отношению к базе данных.

Реализация Microsoft. TDE применяется для файлов базы данных и журнала транзакций на уровне страниц. Шифрование страниц осуществляется с помощью специального симметричного ключа шифрования базы данных (англ. *Database Encryption Key*), который защищен сертификатом, хранящийся в системной базе данных «master» и шифруется с помощью главного ключа, или асимметричным ключом, защищенным модулем управления ключами (англ. *Extensible Key Manager*, EKM). Применение TDE не увеличивает размер зашифрованной информации, а влияние на производительность очень маленькое.

Реализация Oracle. TDE применяется для файлов базы данных на уровне столбцов. Для таблицы, в которой содержится выбранные к шифрованию столбцы, создается специальный симметричный ключ шифрования, защищенный мастер-ключом, который пределами БД, называемым бумажником, (англ. Wallet). Зашифрованные ключи таблиц хранятся в словаре данных (англ. Data Dictionary).

2 Прозрачное шифрование в Oracle

2.1 Принцип прозрачного шифрования в Oracle

Для защиты необходимых конфиденциальных данных в Oracle предусмотрено прозрачное шифрование. При TDE шифруются конфиденциальные данные, которые хранятся в файлах данных. Для предотвращения несанкционированной расшифровки, TDE хранит ключи шифрования в надежном месте, внешне по отношению к базе данных.

Прозрачное шифрование существует в Oracle Advanced Security с версии 10g. С версии 11g TDE позволяет шифровать информацию на уровне колонок таблиц или табличных пространств «прозрачно» для приложений и пользователей.

TDE реализуется для файлов БД на уровне столбцов. Для таблицы, в которой имеется столбцы к шифрованию, создается специальный для каждой таблицы симметричный ключ шифрования, который должен быть защищен главным ключом. Главный ключ хранится в безопасном месте за пределами БД. Место хранения мастер-ключ называется бумажником (англ. Wallet). Зашифрованные симметричные ключи таблиц содержатся в словаре данных (англ. Data Dictionary).

Перед началом шифрования необходимо определить столбец, который будет шифроваться, и сервер Oracle автоматически создаст криптографический стойкий ключ шифрования для таблицы, содержащей этот выбранный столбец, и зашифрует информацию обычного текста в этом столбце, используя указанный алгоритм шифрования или по умолчанию. Защита ключа таблицы имеет важное значение и сервер Oracle Database шифрует его, используя мастер-ключ, который хранится в бумажнике (wallet). Бумажник может быть файлом сервера базы данных. Секретные ключи таблиц хранятся в словаре данных. Когда пользователь выделяет данные столбца, определенного как зашифрованный, сервер Oracle Database извлекает из бумажника главный ключ, расшифровывает ключ шифрования таблицы, который находится в словаре данных, использует этот ключ для шифрования значения столбца. Сохраняет зашифрованные данные в базе данных, как показано на рисунке 2.1. Надписи на рисунке 2.1:

- *Data Dictionary*, словарь данных;
- *Encrypted Table Key*, зашифрованный ключ таблицы;
- *Master Key*, главный ключ;
- *Decrypted*, расшифрованный;
- *Wallet (outside the DB)*, бумажник (за пределами базы данных);
- *Decrypted Table Key*, расшифрованный ключ таблицы;
- *Column*, столбец;
- *Clear Text*, обычный текст;
- *Encrypted*, зашифрованный;
- *Table*, таблица;

- Database, база данных.

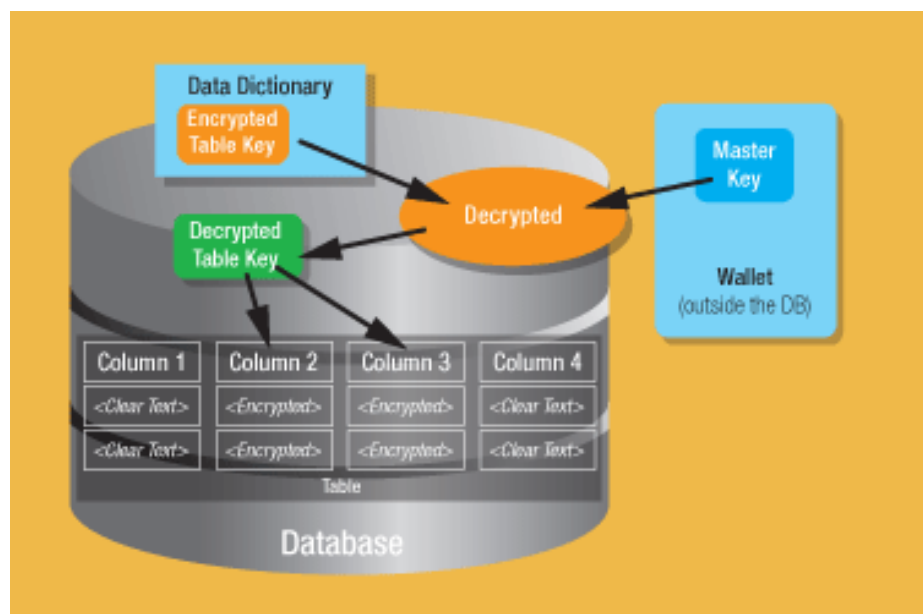


Рисунок 2.1 - Механизм прозрачного шифрования данных

Можно зашифровать любой или все столбцы таблицы. Если таблица имеет четыре столбца и столбцы второй и третий шифруются, сервер Oracle Database генерирует один зашифрованный ключ таблицы и использует его для шифрования этих столбцов. На диске информация столбцов 1 и 4 хранится в виде незашифрованного текста, а информация в двух других столбцах – в зашифрованном формате. Информация хранится зашифрованной, поэтому все значения нижнего уровня, такие, как резервные копии и архивные журнальные файлы, тоже имеют зашифрованный формат.

Когда пользователь выбирает зашифрованные столбцы для просмотра, сервер базы данных Oracle прозрачно извлекает из словаря данных зашифрованный ключ таблицы, а главный ключ извлекает из бумажника и расшифровывает ключ таблицы. После этого сервер базы данных расшифровывает зашифрованные данные на диске и возвращает пользователю обычный текст.

Благодаря такому шифрованию, в случае если данные будут украдены с диска, они не могут быть извлечены без главного ключа, который хранится в бумажнике, не входящим в украденные данные. Даже если украден и бумажник, главный ключ не может быть известен из него, так как не знает пароль от бумажника. Следовательно, злоумышленник не сможет расшифровать данные, даже если он украл диски или копии файлов данных. Это совпадает с требованиями соответствия многих нормативных и руководящих документов. И все это делается без внесения изменения в приложения или написания сложной системы шифрования и системы управления ключами.

Алгоритм реализации шифрования в Oracle database:

- создать бумажник (wallet) для хранения мастер ключа;

- становить местоположение бумажника;
- создать мастер ключ для шифрования словаря;
- открыть бумажник для шифрования данных;
- выделить необходимые для шифрования столбцы.

Для реализации прозрачного шифрования необходимых столбцов создадим бумажник для хранения мастер ключа.

2.2 Создание бумажника и шифрование столбцов

Существует два способа создания электронного бумажника: автоматическое создание бумажника приложением TDE, ручное создание бумажника - для этого имеется встроенный в базу данных инструмент Wallet Manager (рис. 2.2).



Рисунок 2.2 - Wallet Manager

Автоматическое создание бумажника с помощью SQL Developer или SQL Plus. Для этого необходимо сначала определить путь к местоположению бумажника. В нашем случае бумажник будет храниться в директории `D:\app\admin\product\11.2.0\dbhome_2\admin\ORACLE\wallet`.

Для создания бумажника вручную необходимо открыть, встроенное в базу данных, приложение Wallet Manager и нажать на кнопку создать новый бумажник (рис. 2.3). Далее необходимо ввести пароль для управления бумажником (рис. 2.4). Пароль обязательно должен удовлетворять требованиям защиты, то есть длина должна быть не менее 8 символов, должен содержать буквы верхнего и нижнего регистров, обязательно должны присутствовать символы и цифры.

После создания бумажника можно увидеть несколько сертификатов которые создаются автоматически приложением Wallet Manager (рис. 2.5).

Создать бумажник можно командой:

Alter system set encryption key authentication by пароль

В этой команде задаем пароль для нашего бумажника. После создания бумажника надо его открыть. Когда открываем Базу данных бумажник открывается автоматически. После принудительного закрытия бумажника необходимо его открыть. Открыть и закрыть бумажник можно командами (рис. 2.6):

Alter system set encryption wallet open authentication by пароль
Alter system set encryption wallet close.



Рисунок 2.3 - Создание нового бумажника

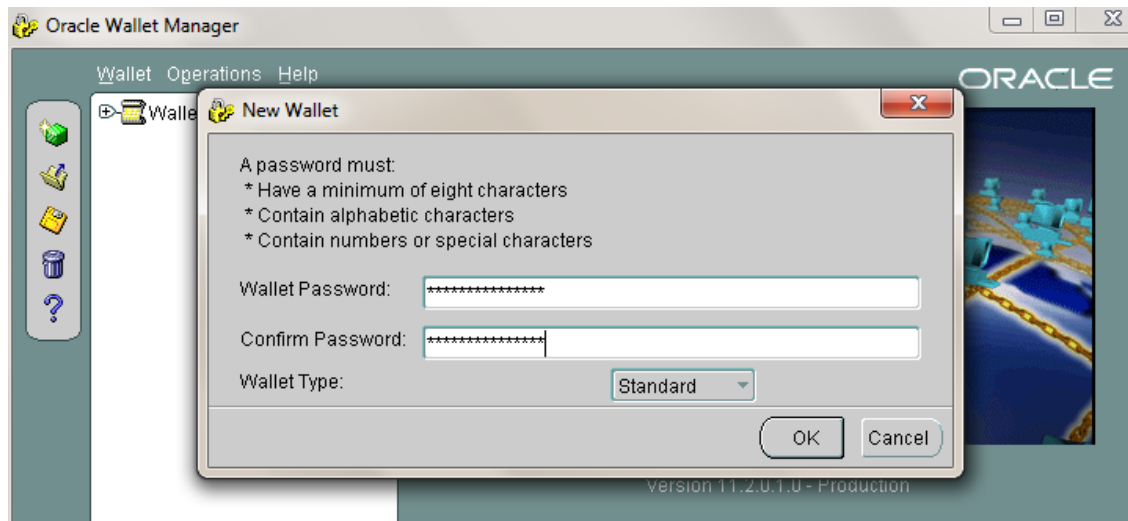


Рисунок 2.4 - Пароль для управления бумажником

Стоит отметить, что бумажник создается только один раз. После создания бумажника невозможно создать еще раз. При попытке создания нового бумажника система выведет, что бумажник уже существует.

После завершения работы с бумажником можно приступать к зашифрованию необходимых столбцов. Для этого следует, выбрать какие столбцы необходимо зашифровать и добавить ключевое слово ENCRYPT. Зашифровать таблицу можно с помощью команды:

```
ALTER TABLE "имя_таблицы" MODIFY ("имя_столбца" ENCRYPT);
```

Эта команда реализует две вещи: автоматически создает ключ для шифрования данных, этот ключ будет храниться в словаре данных, а словарь данных будет зашифрован с помощью мастер ключа, который хранится в бумажнике и зашифрует введенный нами столбец.

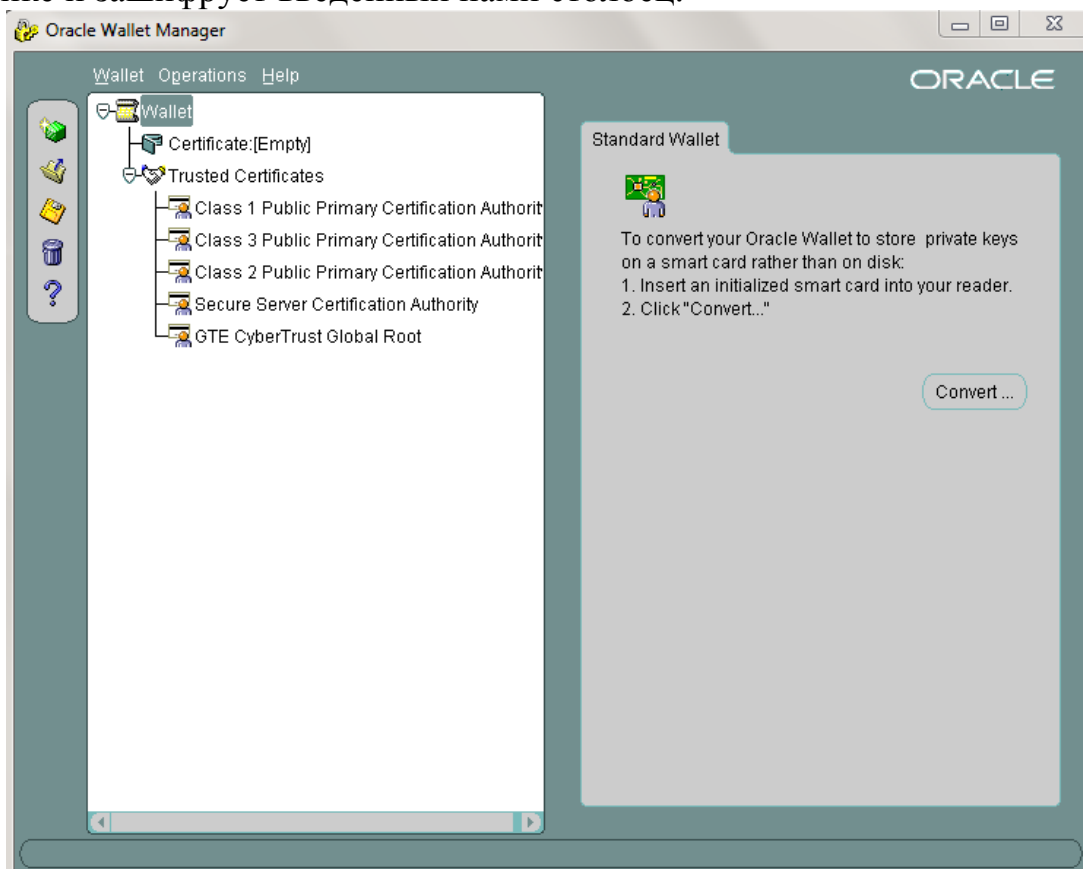


Рисунок 2.5 - Сертификаты

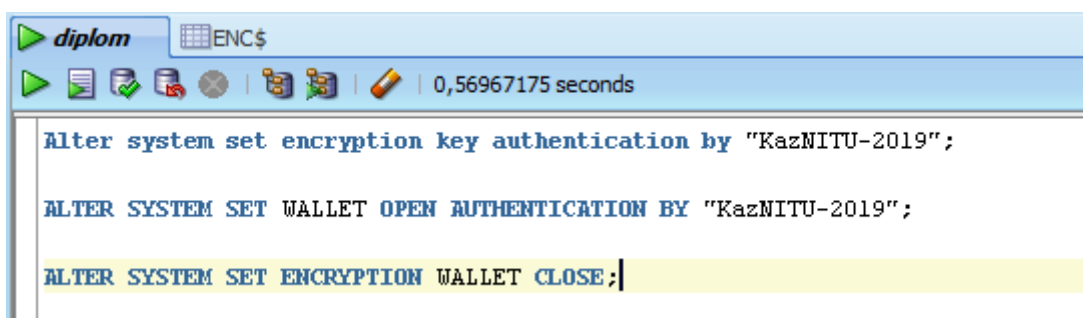


Рисунок 2.6 - Работа с бумажником

Была создана таблица SOTRUDNIKI в базе данных Oracle 11g и к ней применено прозрачное шифрование (рис. 2.7). Был зашифрован столбец № KARTY при помощи команды:

```
ALTER TABLE SOTRUDNIKI MODIFY (№ KARTY ENCRYPT);
```

Алгоритм шифрования выбирается по умолчанию AES-192, который хранится в пакете DBMS_CRYPTO. В ней еще хранится несколько алгоритмов шифрования. Алгоритмы шифрования приведены в таблице 1.

Для изменения алгоритма шифрования необходимо дополнить команду шифрования:

```
ALTER TABLE "ИМЯ_ТАБЛИЦЫ" MODIFY ("ИМЯ_СТОЛБЦА"
ENCRYPT USING "Алгоритм_шифрования")
```

```
ALTER TABLE SOTRUDNIKI MODIFY (№_карты ENCRYPT USING
'AES256');
```

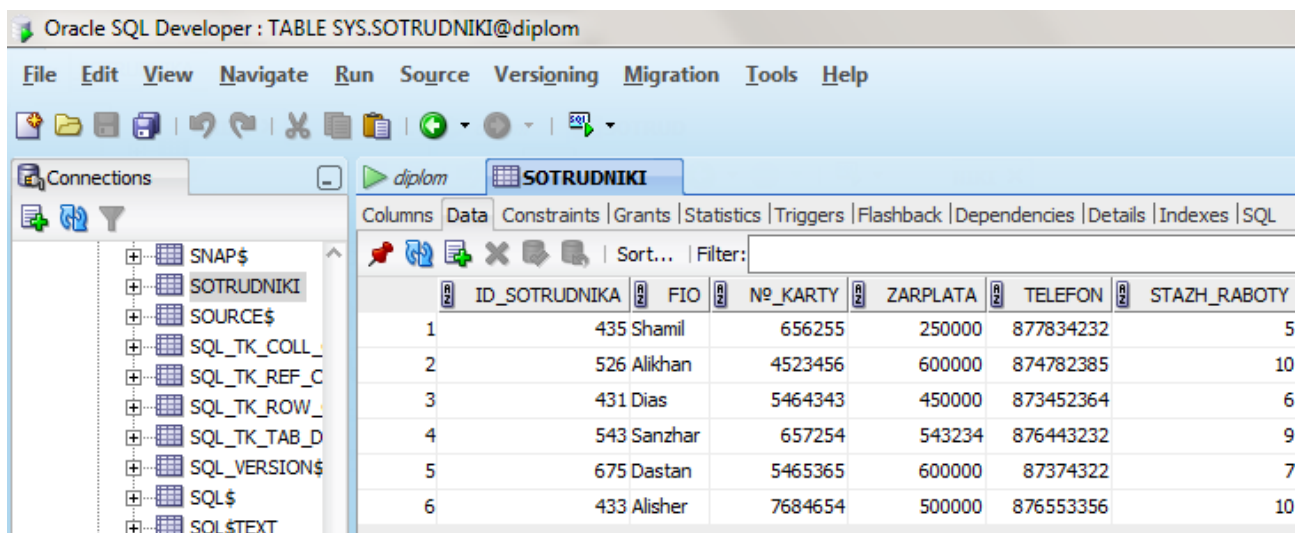


Рисунок 2.7 – Таблица SOTRUDNIKI

Для проверки зашифрована таблица или нет можно обратиться к системной таблице DBA_ENCRYPTED_COLUMNS или посмотреть описание таблицы, где есть специальная графа для зашифрованных данных. В ней прописаны данные: зашифрована таблица или нет и алгоритм шифрования.

Таблица 1 - Константы алгоритмов пакета DBMS_CRYPTO

| Константа | Фактическая длина ключа | Описание |
|-------------------|-------------------------|---|
| ENCRYPT_DES | 56 | Алгоритм симметричного шифрования DES |
| ENCRYPT_3DES_2KEY | 112 | Модифицированный DES3; блок шифруется в три прохода с двумя ключами |
| ENCRYPT_3DES | 156 | DES3; блок шифруется в три прохода |
| ENCRYPT_AES128 | 128 | Алгоритм симметричного шифрования AES |
| ENCRYPT_AES192 | 192 | Алгоритм симметричного шифрования AES |
| ENCRYPT_AES256 | 256 | Алгоритм симметричного шифрования AES |

При шифровании еще используются добавление «соли». При совпадении данных строки со строками других пользователей можно легко догадаться и расшифровать этот столбец. Для предотвращения таких не очень важных проблем необходимо добавить «соль». Она при шифровании добавляет к данным значения, после которого данные приобретают разные значения.

3 Прозрачное шифрование в MS SQL Server

3.1 Алгоритм прозрачного шифрования

С версии Microsoft SQL Server 2008 появился новый тип шифрования – прозрачное шифрование (TDE). Шифруется база данных целиком [9-12] При этом, когда страница выгружаются из ОЗУ, данные зашифровываются, а при загрузке информации в ОЗУ они автоматически расшифровываются. Вследствие этого шифрование происходит прозрачно для пользователя.

Для успешной защиты зашифрованных данных необходимо защитить ключ. Для этого используется иерархия ключей.

Недостатком прозрачного шифрования может быть то, что большие расходы по производительности при шифровании всей базы данных. Сервер вынужден каждый раз расшифровать данные при реализации запроса к информации. База данных сильно загружает память при повторном запросе.

Таблица 2 – Уровни шифрования ключей SQL Server

| Уровень SQL Server | Уровень ANSI X9.17 | Описание |
|---|------------------------|---|
| SMK | Главный ключ | SMK – ключ верхнего уровня, используемый для шифрования DMK. SMK шифруется с применением Windows DPAPI |
| DMK | Ключ шифрования ключей | DMK – симметричный ключ, используемый для шифрования симметричного ключа, асимметричного ключа и сертификата. Для каждой базы данных может быть определен только один DMK |
| Симметричные ключи, асимметричные ключи и сертификаты | Ключ данных | Симметричные ключи, асимметричные ключи и сертификаты используются для шифрования данных |

Модель шифрования SQL Server предоставляет функции управления

ключами шифрования, которые соответствуют стандарту ANSI X9.17. В стандарте определено несколько уровней ключей шифрования, использующихся для шифрования других ключей, которые применяются для шифрования данных. В таблице 2 перечислены уровни ключей шифрования SQL Server и ANSI X9.17.

Главный ключ службы Service master key(SMK) — ключ верхнего уровня и отец всех ключей в SQL Server. SMK — асимметричный ключ, который шифруется с использованием Windows Data Protection API (DPAPI). SMK создается автоматически, когда шифруется какой-нибудь объект, и привязан к учетной записи службы SQL Server. SMK используется для шифрования главного ключа базы данных Database master key (DMK).

Второй уровень иерархии ключей шифрования — DMK. С его помощью шифруются асимметричные ключи и симметричные ключи, сертификаты. Каждая база данных располагает лишь одним DMK.

Третий уровень содержит симметричные ключи, асимметричные ключи и сертификаты. Симметричные ключи — основное для шифрования в базе данных (см. таблицу 3).

Для изменения алгоритма шифрования необходимо дополнить команду шифрования:

```
ALTER TABLE "ИМЯ_ТАБЛИЦЫ" MODIFY ("ИМЯ_СТОЛБЦА"
ENCRYPT USING "Алгоритм_шифрования")
```

```
ALTER TABLE Student MODIFY (№_карты ENCRYPT USING
'AES128')
```

Для проверки, зашифрована таблица или нет, можно обратиться к системной таблице DBA_ENCRYPTED_COLUMNS или посмотреть описание таблицы, где есть специальная графа для зашифрованных данных. В ней прописаны данные: зашифрована таблица или нет и алгоритм шифрования.

Таблица 3 – Алгоритмы симметричного шифрования

| Наименование алгоритма шифрования | Длина используемого ключа | Длина шифруемого блока | Длина действительного ключа |
|-----------------------------------|---------------------------|------------------------|-----------------------------|
| DES | 64 бит | 64 бит | 56 бит |
| Triple_DES | 128 бит | 64 бит | 112 бит |
| Triple_DES_3KEY | 128 бит | 64 бит | 112 бит |
| DESX | 192 бит | 64 бит | 184 бит |
| RC2 | 128 бит | 64 бит | 112 бит |
| RC4 | 40 бит | - | 50 бит |
| RC4_128 | 128 бит | - | 112 бит |
| AES_128 (Rijndael) | 128 бит | 128 бит | 128 бит |
| AES_192 | 192 бит | 128 бит | 192 бит |
| AES_256 | 256 бит | 128 бит | 256 бит |

Иерархия ключей прозрачного шифрования (рис. 3.1):

- для каждой базы данных имеется свой специальный ключ - Database Encryption Key;
- Database Encryption Key также шифруется специальным сертификатом, для надежной защиты, которую создает база данных Master;
- сертификат, который создается в Базе данных Master, должен быть зашифрован главным ключом;
- главный ключ БД Master шифруется главным ключом Service Master – Key.

Для реализации шифрования необходимо выполнить некоторые действия:

- Создать мастер ключ (Master Key);
- создать и получить сертификат, который должен быть защищен главным ключом;
- создать ключ для шифрования данных и обеспечить ему защиту с помощью сертификата;
- зашифровать данные.

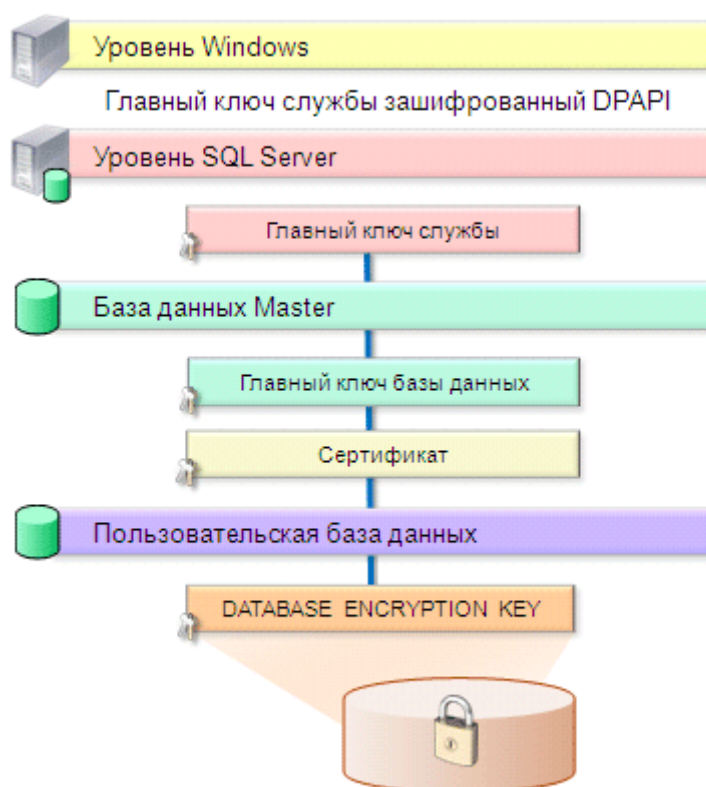


Рисунок 3.1 - Иерархия ключей шифрования

Мастер-ключ можно создать командой:

`CREATE MASTER KEY ENCRYPTION BY PASSWORD = пароль`

Далее необходимо получить сертификат. Он должен быть зашифрован мастер-ключом который был создан до этого. Сертификат можно создать с помощью команды:

```
CREATE CERTIFICATE "имя_сертификата" WITH SUBJECT  
"описание_сертификата"
```

После создания мастер-ключа и сертификата, зашифрованного мастер-ключом, нужно создать их резервные копии и сохранить в надежном месте, чтобы при потере ключа или сертификата можно было восстановить их. Резервные копии создаются командой:

```
BACKUP MASTER KEY TO FILE = место хранения копии
```

```
BACKUP CERTIFICATE имя_сертификата TO FILE = место хранения  
копии
```

После этого в базе данных, которую мы должны зашифровать, должны создать Database Encryption Key (DEK), который шифруется сертификатом. Database Encryption Key (DEK) можно создать командой:

```
CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM =  
AES_256 ENCRYPTION BY SERVER CERTIFICATE DEK
```

Проверить, что Database Encryption Key (DEK) действительно создан, можно в системном представлении sys.dm_database_encryption_keys.

3.2 Проектирование и реализация БД

Для проектирования базы данных сначала необходимо провести анализ предметной области создаваемой базы данных. Нашей предметной областью будет служить Библиотека. Они занимаются хранением, продажей, выдачей книг, журналов. База данных в основном предназначена для упрощения работы сотрудников этой библиотеки, которые выполняют полный контроль над большим объемом книг и журналов.

На основе проведенного анализа предметной области бы спроектирована структура нашей базы данных. Предметная область содержит информацию:

- об отделах, в которой хранятся книги и журналы, в зависимости от жанра и году издания;
- о книгах, журналах, учебниках которые имеются в данной библиотеке;
- о продаже и выдаче книг читателям;
- о читателях, которые зарегистрированы в библиотеке;
- о сотрудниках, которые работают в этой сфере.

При проектировании были определены атрибуты, сущности, связи между ними. Связи между сущностями были установлены через первичные и внешние ключи. Проектируемая модель базы данных имеет следующие сущности:

- knigi;

- chitately;
- otdely;
- sotrudniki;
- vidacha_knig.

Сущность «otdely» содержит информацию об отделах библиотек. Атрибуты - номер отдела, название отдела, телефон_отдела, адрес_отдела.

Сущность «sotrudniki» содержит информацию о сотрудниках библиотеки. Атрибуты – номер_сотрудника, ФИО, № карточки, зарплата, телефон, стаж работы.

Сущность «knigi» включает в себя данные о книгах, которые имеются в библиотеке. Атрибуты – ном_книги, автор_книги, название, год_издания, жанр.

Сущность «vidacha_knig» содержит информацию о книгах, которые уже выданы читателям.

Сущность «chitately» имеет информацию о читателях, которые зарегистрированы в этой библиотеке. Имеет атрибуты – номер_читателя, ФИО, дата_рождения, адрес.

Связи между сущностями представлены в таблице 4.

Таблица 4 – Связи между сущностями

| Главная сущность | Первичный ключ | Подчиняющаяся сущность | Тип связи |
|------------------|----------------|------------------------|-----------|
| otdel | nom_otdela | Kniga | 1:M |
| | | Sotrudnik | 1:M |
| kniga | id_kinigi | Vidacha_knig | M:1 |
| sotrudniki | ID_sotrud | Viadacha knig | 1:M |
| chitately | Id_chitately | Vidacha_knig | 1:M |
| | | Sotrudniki | 1:M |
| vidacha_knig | id | Kniga | 1:1 |
| | | Chitateli | 1:M |

В CASE-средстве AllFusion Erwin Data Modeler 7.2 была спроектирована логическая и физическая модели базы данных. С помощью первичных ключей и внешних ключей были установлены связи между таблицами. Полученная ER-диаграмма логического уровня представлена на рисунке 3.2. На рисунке 3.3 представлена ER-диаграмма физического уровня.

Спроектированная база данных была реализована в Microsoft SQL Server 2012.

При реализации базы данных в СУБД SQL-сервер можно выделить два метода создания базы:

- с помощью графических инструментов Microsoft SQL Server Management Studio;
- с помощью языка запросов T-SQL.

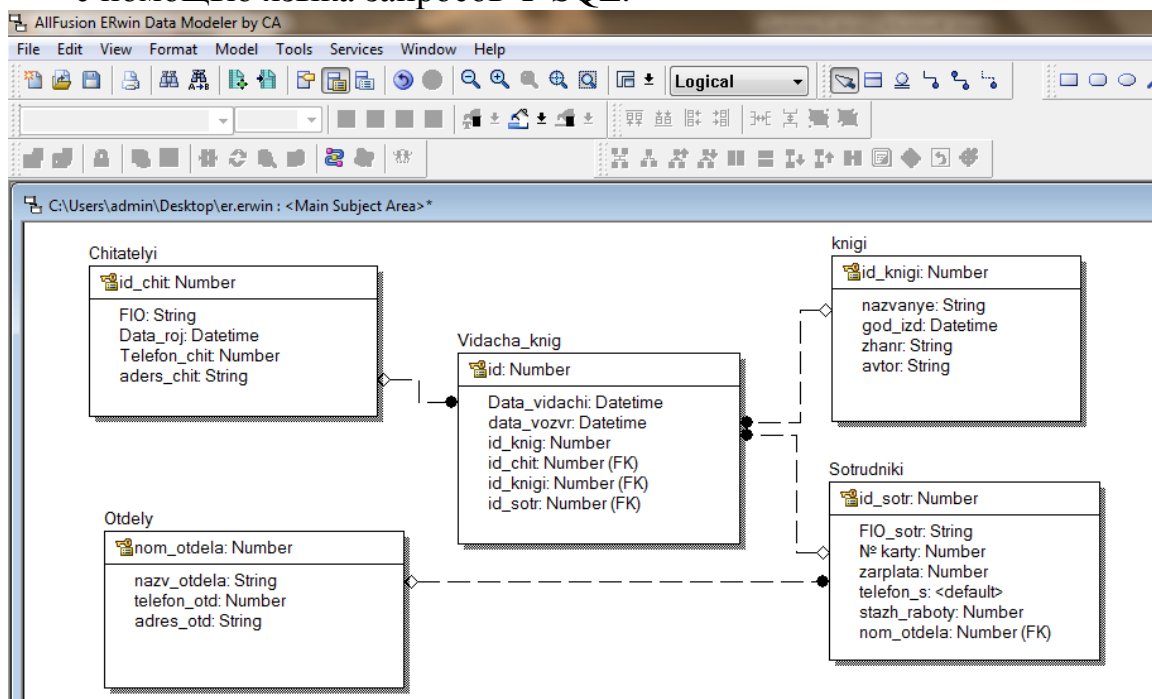


Рисунок 3.2 - ER- диаграмма на логическом уровне

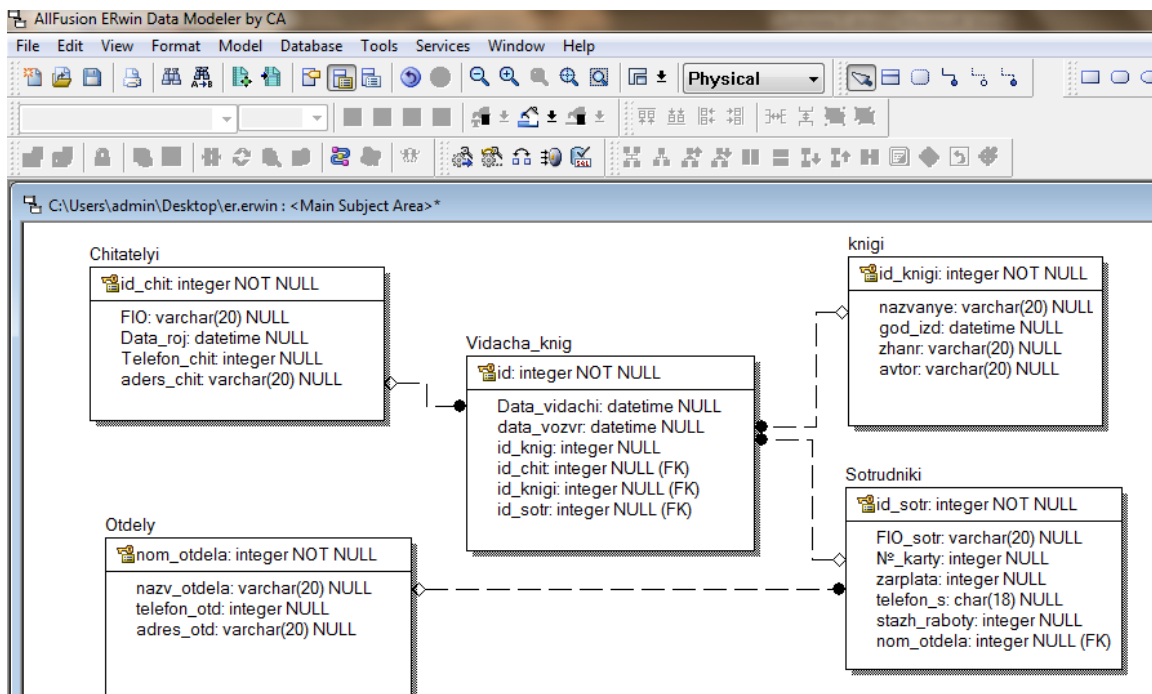


Рисунок 3.3 - ER- диаграмма на физическом уровне

Создание базы данных в системе SQL-сервер реализуется командой CREATE DATABASE «имя базы данных». Эта процедура требует наличия прав администратора SQL-сервера. В политике безопасности первым

уровнем защиты данных является аутентификация пользователя. Для этого необходимо выполнить настройки аутентификации на уровне ОС такие как «журнал паролей», «минимальный максимальный срок действия пароля», «минимальная длина пароля», «требования сложности пароля»

Для создания базы данных необходимо ввести команду CREATE DATABASE «имя базы данных». Далее требуется создать таблицы в этой базе данных. Создать ее можно с помощью команды CREATE TABLE «имя_таблицы» (сущности таблицы) или с помощью графических инструментов Microsoft Sql Server Managment Studio.

На рисунке 3.4 приведена диаграмма спроектированной базы данных.

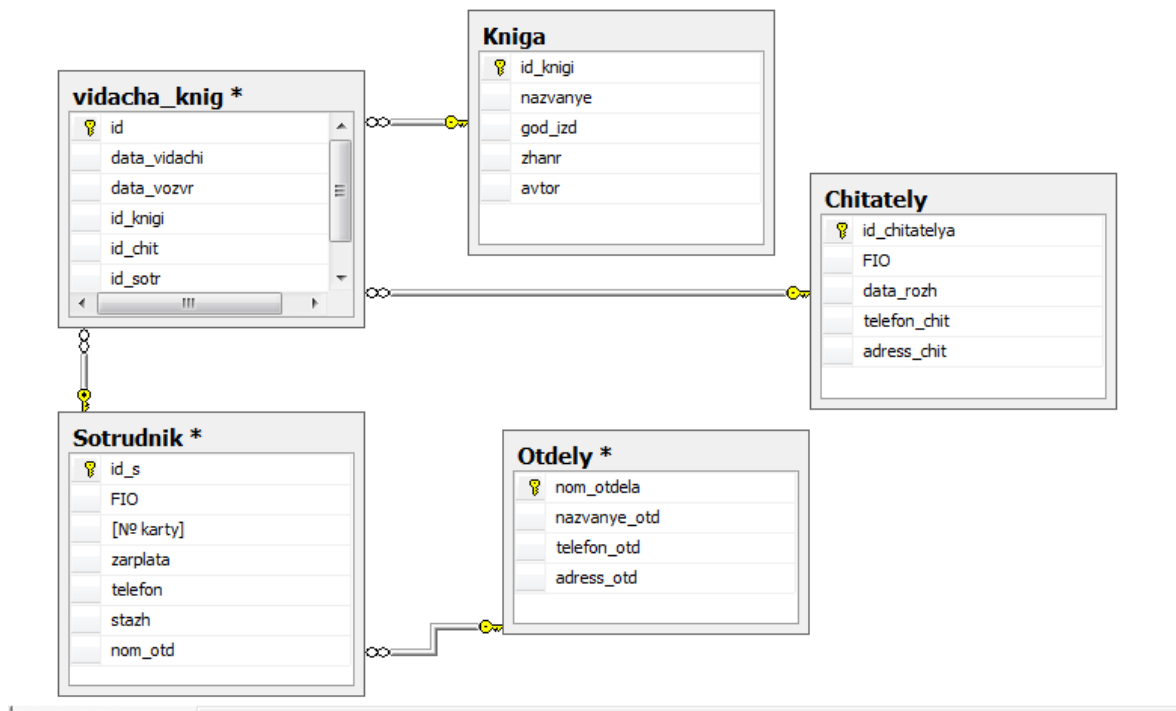


Рисунок 3.4 - Диаграмма проектируемой базы данных.

Ниже приведена часть скриптов создания базы данных с использованием языка T-SQL. Полностью скрипты создания базы данных приведены в Приложении А.

Create database Библиотека

-- создаем таблицы

```

Create table chitately (id_chitatelya int PRIMARY KEY,
                        FIO varchar(20),
                        Data_rojdeniya date,
                        Telefon_chatatelya int,
                        Addres chitatelya varchar(20)
    )
    
```

```

Create table sotrydniki ( id_sotrudnika int primary key,
                        FIO varchar(20),
                        № karty int,
    )
    
```


Zarplata int,
Telefon int,
Stash_raboty varchar (10)

);

Далее создаем сущности для таблицы.

Структура таблицы Читатели (рис 3.5).

| | Имя столбца | Тип данных | Разрешить ... |
|---|---------------|------------|-------------------------------------|
| ▶ | id_chitatelya | int | <input type="checkbox"/> |
| | FIO | nchar(10) | <input checked="" type="checkbox"/> |
| | data_rozh | date | <input checked="" type="checkbox"/> |
| | telefon_chit | int | <input checked="" type="checkbox"/> |
| | adress_chit | nchar(10) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.5 - Структура таблицы Читатели

На рисунке 3.6 показана Структура таблицы Книга.

| | Имя столбца | Тип данных | Разрешить ... |
|---|-------------|------------|-------------------------------------|
| ▶ | id_knigi | int | <input type="checkbox"/> |
| | nazvanye | nchar(10) | <input checked="" type="checkbox"/> |
| | god_izd | date | <input checked="" type="checkbox"/> |
| | zhanr | nchar(10) | <input checked="" type="checkbox"/> |
| | avtor | nchar(10) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.6 - Структура таблицы Книга

Структура таблицы Отделы рисунок 3.7

| | Имя столбца | Тип данных | Разрешить ... |
|---|--------------|------------|-------------------------------------|
| ▶ | nom_otdela | int | <input type="checkbox"/> |
| | nazvanye_otd | nchar(10) | <input checked="" type="checkbox"/> |
| | telefon_otd | int | <input checked="" type="checkbox"/> |
| | adress_otd | nchar(10) | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.7- Структура таблицы Отделы

На рисунке 3.8 показаны Структура таблицы Сотрудники.

| | Имя столбца | Тип данных | Разрешить ... |
|-----|-------------|----------------|-------------------------------------|
| ▶ 🔑 | id_s | int | <input type="checkbox"/> |
| | FIO | nchar(20) | <input checked="" type="checkbox"/> |
| | [№ karty] | bigint | <input checked="" type="checkbox"/> |
| | zarplata | int | <input checked="" type="checkbox"/> |
| | telefon | numeric(18, 0) | <input checked="" type="checkbox"/> |
| | stazh | int | <input checked="" type="checkbox"/> |
| | nom_otd | int | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.8 - Структура таблицы Сотрудники

Структура таблицы Выдача_книг (рис. 3.9)

| | Имя столбца | Тип данных | Разрешить ... |
|-----|--------------|------------|-------------------------------------|
| ▶ 🔑 | id | int | <input type="checkbox"/> |
| | data_vidachi | date | <input checked="" type="checkbox"/> |
| | data_vozvr | date | <input checked="" type="checkbox"/> |
| | id_knigi | int | <input checked="" type="checkbox"/> |
| | id_chit | int | <input checked="" type="checkbox"/> |
| | id_sotr | int | <input checked="" type="checkbox"/> |
| | | | <input type="checkbox"/> |

Рисунок 3.9 - Структура таблицы Выдача_книг.

При заполнении таблиц можно воспользоваться командой:
 INSERT INTO имя_таблицы VALUES («значения для заполнения по порядку»)

3.3 Шифрование БД

Как было упомянуто ранее, для реализации прозрачного шифрования необходимо выполнить несколько действий, которые связаны в основном с ключами. Для успешной работы шифрования надо создать иерархию ключей и их резервной копии, чтобы обратиться к ним в случае утраты ключа и сертификата.

Ниже представлена реализация шифрования для нашей базы данных, которую мы создали ранее.

Для начала необходимо создать главный ключ шифрования. Проверку созданного мастер ключа можно посмотреть в системной таблице sys.key_encryptions (рис. 3.10). Удалить мастер ключ можно командой : drop master key.

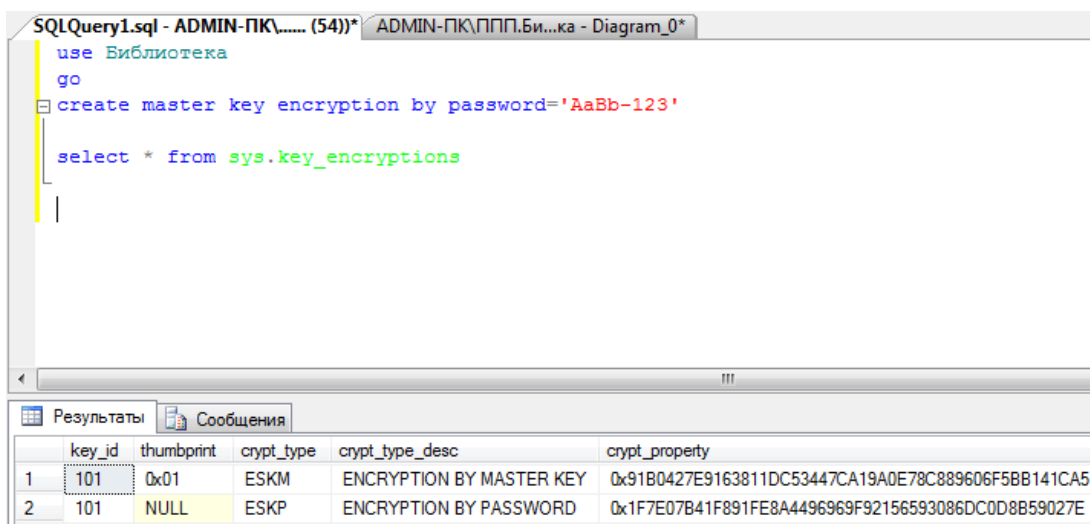


Рисунок 3.10 - Создание мастер-ключ и его проверка.

После создания мастер-ключ необходимо создать его резервную копию и сохранить его в надежное место (рис. 3.11). Результат можно посмотреть в директории, который указали (рис 3.12). Резервную копию можно создать командой:

```

BACKUP MASTER KEY TO FILE = 'место хранения копии'
ENCRYPTION BY PASSWORD = 'пароль'

```

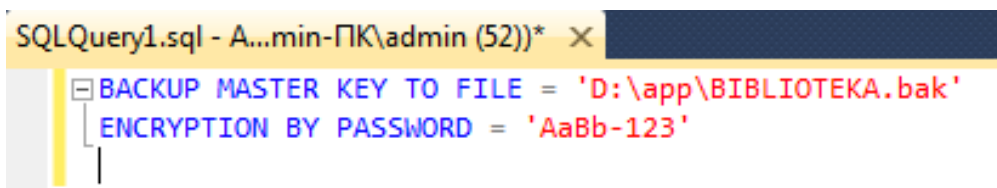


Рисунок 3.11 - Резервная копия мастер-ключ

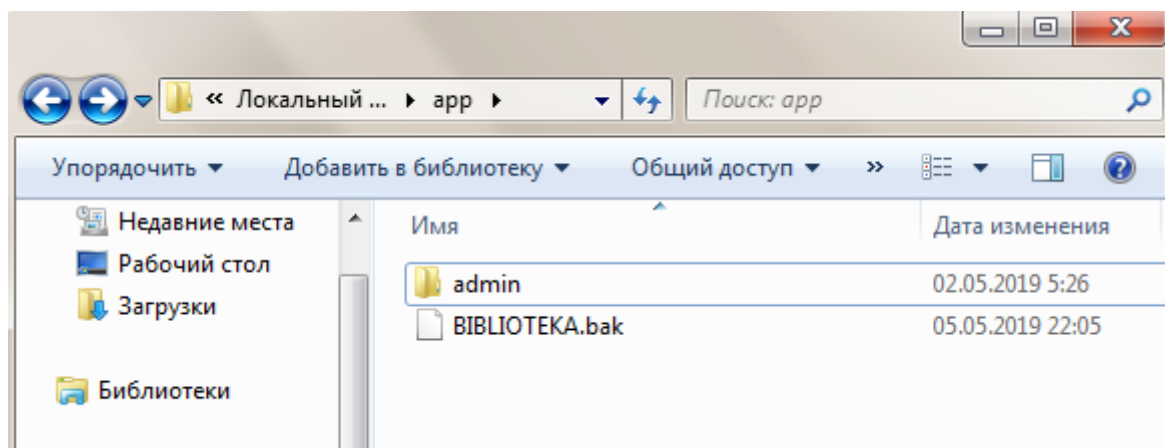


Рисунок 3.12 - Результат создания резервной копии

Далее создадим сертификат, который будет защищен мастер-ключом. Сертификат можно создать с помощью команды CREATE CERTIFICATE (рис 3.13). Созданный сертификат можно проверить в системной таблице sys.certificates (рис. 3.14).

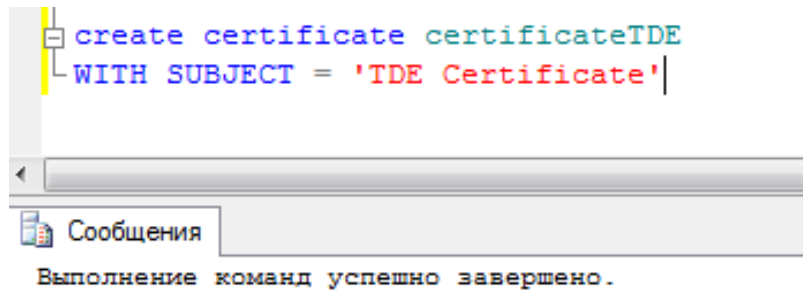


Рисунок 3.13 - Создание сертификата

В таблице 5 представлено назначение столбцов системной таблицы sys.certificates.

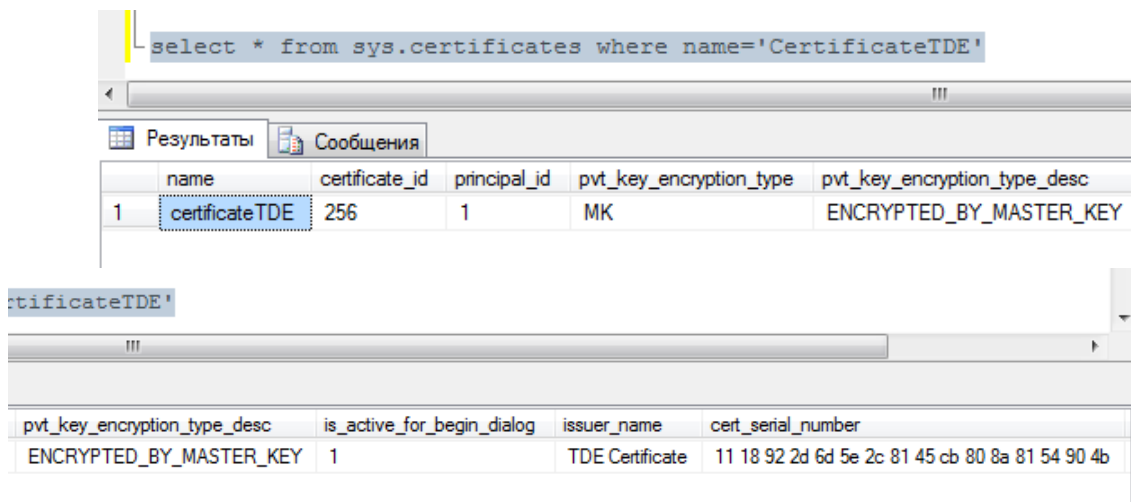


Рисунок 3.14 - Проверка созданного сертификата

Таблица 5 – Содержимое sys.certificates

| Имя столбца | Description |
|---------------------------|--|
| Name | Имя сертификата, который мы указали, уникален в нашей базе данных. |
| идентификатор_сертификата | Идентификатор сертификата, уникален в нашей базе данных. |
| principal_id | Идентификатор пользователя базы данных, владеющего сертификатом. |
| pvt_key_encryption_type | Способ шифрования закрытого ключа. NA = сертификат не имеет закрытого ключа. |

Продолжение таблицы 5

| | |
|------------------------------|--|
| pvt_key_encryption_type | МК = закрытый ключ зашифрован паролем пользователя. МК = закрытый ключ, который зашифрован главным ключом службы. |
| issuer_name | Имя поставщика сертификата. |
| is_active_for_begin_dialog | Если значение равно 1, то сертификат используется для инициализации скрытых диалоговых окон службы. |
| pvt_key_encryption_type_desc | Описание способа шифрования секретного ключа. NO_PRIVATE_KEY ENCRYPTED_BY_MASTER_KEY ENCRYPTED_BY_PASSWORD ENCRYPTED_BY_SERVICE_MASTER_KEY |
| cert_serial_number | Серийный номер сертификата. |
| ИД безопасности | Идентификатор SID имени входа для сертификата. |
| string_sid | Представление идентификатора SID |
| expiry_date | Срок действия сертификата. |
| start_date | Дата выхода сертификата. |
| attested_by | Только для системного использования. |
| pvt_key_last_backup_date | Дата и время последнего экспорта закрытого ключа сертификата. |

Теперь создаем ключ шифрования нашей базы данных (рис. 3.15) с использованием нашего сертификата:

```
USE Biblioteka
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM=AES_256
ENCRYPTION
BY SERVER CERTIFICATE CertifacteTDE
```

Далее запускаем процесс шифрования с помощью команды ALTER DATABASE SET ENCRYPTION ON (рис. 3.16). Если процесс шифрования запущен, то в этой же таблице можно увидеть, на сколько процентов он завершен. Данные таблицы можно просмотреть стандартным запросом:

```
select * from sys.dm_database_encryption_keys
```

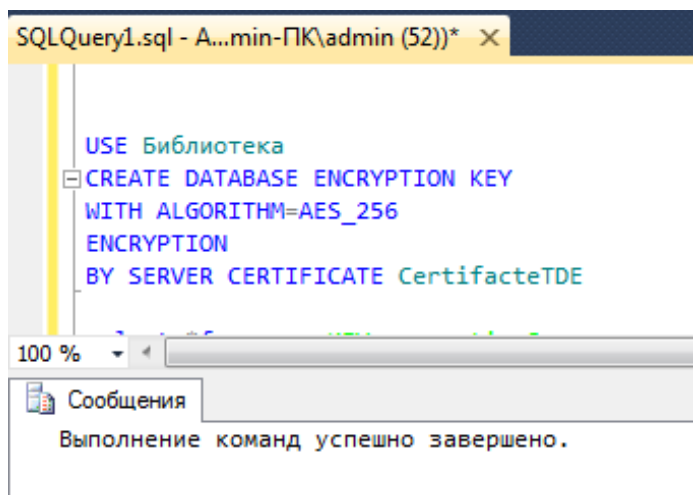


Рисунок 3.15 - Создание ключа шифрования

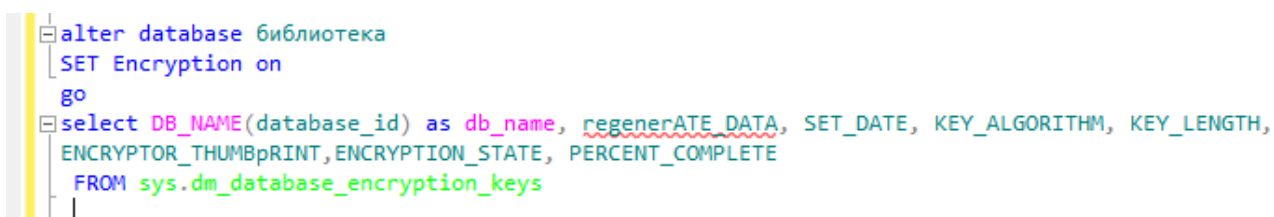


Рисунок 3.16 - Процесс шифрования

В таблице 6 представлено содержимое системной таблицы sys.dm_database_encryption_keys.

Таблица 6 - Содержимое sys.dm_database_encryption_keys

| Имя столбца | Description |
|------------------|--|
| database_id | Идентификатор базы данных. |
| encryption_state | Выводит, является ли база данных зашифрованной или незашифрованной. 0 - нет ключа шифрования базы данных (DEK), нет шифрования; 1 - Database Encryption Key (DEK) создан, но база данных не зашифрована; 2 - выполняется первоначальное шифрование; |

Продолжение таблицы 6

| | |
|----------------------|--|
| encryption_state | 2 - база данных зашифрована; 4 - выполняется изменение ключа; 5 - Выполняется расшифровка; 6 - Производится изменение защиты (изменился сертификат или асимметричный ключ, которым зашифрован ключ шифрования базы данных); |
| create_date | Выводит дату создания ключа шифрования. |
| regenerate_date | Выводит дату повторного создания ключа шифрования. |
| modify_date | Выводит дату изменения ключа шифрования. |
| set_date | Выводит дату применения ключа шифрования к базе данных. |
| opened_date | Выводит, когда ключ базы данных был открыт в последний раз. |
| key_algorithm | Выводит алгоритм, используемый для ключа. |
| key_length | Выводит длину ключа. |
| encryptor_thumbprint | выводит отпечаток шифратора. |
| encryptor_type | Область применения: SQL Server (с SQL Server 2012 до <u>текущей версии</u>). Описывает шифратор. |
| percent_complete | Процент выполнения шифрования базы данных. Значение 0, если изменения состояния не было. |

Для обеспечения надежности необходимо создавать резервные копии и зашифровать их. SQL-сервер шифрует резервную копию ключом, который сохраняется вместе с самой резервной копией. Поэтому этот ключ тоже нужно зашифровать. Это делается при помощи сертификата, который хранится отдельно от базы и не попадает в резервную копию [13].

Создать зашифрованные резервные копии можно с помощью команды:
 BACKUP DATABASE backup
 To disk=' backup2.bak'
 WITH ENCRYPTION:(
 ALGORITHM=AES_256
 Server certificate= cert)

ЗАКЛЮЧЕНИЕ

Проблемы защиты баз данных являются очень важными в настоящее время. Согласно результатам последних анализов, ущерб от нарушения одного из параметров «конфиденциальность-целостность-доступность» одной записи данных составляет очень большие денег. Эти деньги идут на восстановление потерянной информации, расследование, предотвращение ущерба репутации компании и т.д. Впоследствии проблема обеспечения безопасности баз данных становится крайне актуальной. Малейший сбой работы базы перебивает работу всей системы и работу компании в целом. Обеспечение безопасности данных становится одной из важнейших проблем в современных технологиях, так как потеря либо утечка данных, в особенности, если это данные пользователей, приводит к потере репутации бренда компании, снижению конкурентоспособности, большим штрафам – и даже к привлечению к ответственности.

Вопросы и проблемы безопасности данных необходимо решать с этапа проектирования и реализации базы данных.

В ходе выполнения работы были изучены и проанализированы методы шифрования в серверах базы данных. Рассмотрен прозрачный метод шифрования (TDE), реализован метод в MS SQL Server и Oracle.

Рассмотрено создание бумажника (wallet) в Oracle 11g и его использование для прозрачного шифрования столбцов базы данных.

Спроектирована база данных для предметной области «Библиотека» с использованием CASE – средства проектирования базы данных AllFusion Erwin Data Modeler 7.2.

Для обеспечения безопасности информации спроектированной базы данных были применены криптографические методы защиты информации. Данные были зашифрованы с использованием метода прозрачного шифрования (TDE). Для реализации TDE были созданы мастер-ключ, главный ключ базы данных, сертификат, ключ для шифрования БД. Был использован алгоритм шифрования AES_256.

TDE в ORACLE дает возможность шифровать отдельные столбцы, а в MS SQL Server шифруется вся база данных целиком, что отрицательно влияет на производительность системы.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

- 1 Шаньгин В.Ф. Информационная безопасность компьютерных систем и сетей. – М.: ИД ФОРУМ: ИНФРА – М, 2017. – 416 с.
- 2 Luc Bouganim, Yanli Guo. Database Encryption //Электронная версия на сайте <https://www-smis.inria.fr>
- 3 Айтхожаева Е.Ж. Криптография в серверах баз данных. Вестник КазНТУ, № 5 (105) 2014.– Алматы, НТИЦ КазНТУ, 2014.
- 4 Кабдыгалиев Ш.Ж., Айтхожаева Е.Ж. Анализ механизмов защиты серверов баз данных Oracle. Подготовка инженерных кадров в контексте глобальных вызовов XXI века: Труды Международной научно-практической конференции в рамках Сатпаевских чтений. - Алматы, КазНТУ, 2013 г. - Том 3.
- 5 Астанаева А.Ә., Айтхожаева Е.Ж. Шифрование баз данных средствами MS SQL Server. Журнал Поиск, № 2(3)/2014. - Алматы: Высшая школа Казахстана, 2014.
- 6 Айтхожаева Е.Ж., Жансейітова А. Т. Использование специальных функций шифрования в MS SQL Server. Научно-технический прогресс: актуальные и перспективные направления будущего: сборник материалов V Международной научно-практической конференции (7 апреля 2017 года), Том I - Кемерово: ЗапСибНЦ.
- 7 Абдулхассан Ф.Х. Прозрачное шифрование данных (TDE). Журнал Молодой ученый, №8 (67)/2014.
- 8 Нанда А. Прозрачное шифрование данных (TDE). Журнал «Oracle Magazine». - 2015.
- 9 Справочник по продуктам Microsoft // Электронная версия на сайте <http://msdn.microsoft.com/ru-ru/library>.
- 10 Справочник по Transact-SQL (компонент Database Engine) // Электронная версия на сайте <https://msdn.microsoft.com/ru-ru/library/ms189774.aspx>.
- 11 Справочник по Transact-SQL (компонент Database Engine) // Электронная версия на сайте <https://msdn.microsoft.com/ru-ru/library/bb677274.aspx>.
- 12 Самородов Ф. А. Шифруйте резервные копии баз данных в SQL-сервере // Электронная версия на сайте <http://www.specialist.ru/center/advice/118/shifrujte-rezervnie-kopii-baz-dannih-v-sqlserve>.
- 13 Документация по SQL Server. Прозрачное шифрование данных (TDE) // Электронная версия на сайте <https://docs.microsoft.com/ru-ru/sql/sql-server/sql-server-technical-documentation?toc=.%2Ftoc%2Ftoc.json&view=sql-server-2017>.

Приложение А

Скрипты для создания таблицы

```
-- создаем таблицы
Create table chitately (id_chitatelya int PRIMARY KEY,
                        FIO varchar(20),
                        Data_rojdenya date,
                        Telefon_chatatelya int,
                        Adres chitatelya varchar(20)
)
Create table sotrudniki ( id_sotrudnika int primary key,
                          FIO varchar(20),
                          № karty int,
                          Zarplata int,
                          Telefon int,
                          Stash_raboty varhcar (10)
);
Create table KNIGI (id_knigi int Primary key,
                   Navanye nvvarchar (20),
                   God_izd date,
                   Zhanr nvvarchar(15),
                   Avtor nvvarchar(30)
);
Create table Otdely ( nom_otdela int Primary key,
                    Nazvanye_otd nvvarchar (20),
                    Telefon_otd int,
                    Address_otd nvvarchar (20)
);
Create table Vidacha_knig ( id int Primary key,
                            Data_vidachi date,
                            Data_vozvr date
);
```

Скрипты для управления бумажником

Alter system set encryption key authentication by пароль – создает бумажник

Alter system set encryption wallet open authentication by пароль – открывает бумажник

Alter system set encryption wallet close – закрывает бумажник

Скрипты для зашифрования данных в базе данных Oracle

ALTER TABLE “имя_таблицы” MODIFY (“имя_столбца” ENCRYPT) –
Алгоритм по умолчанию AES192

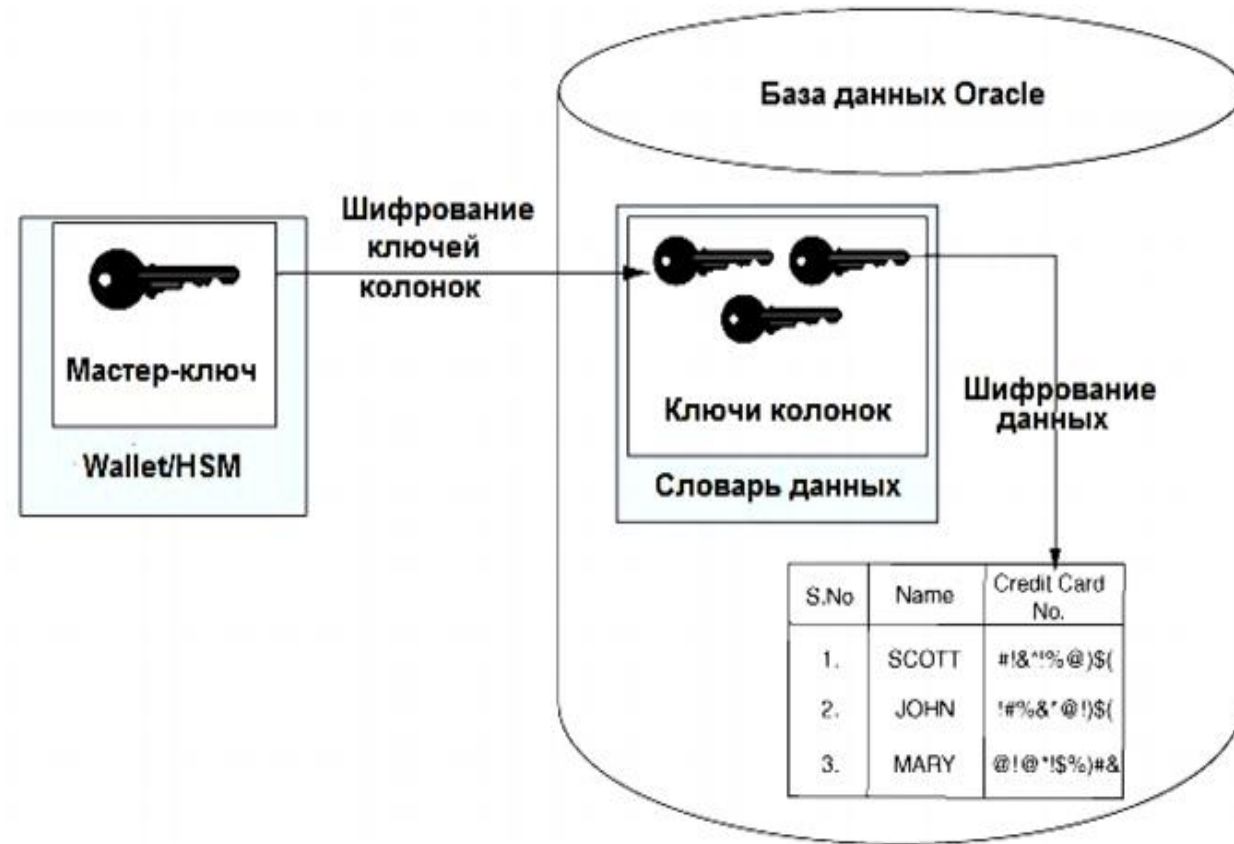
ALTER TABLE “ИМЯ_ТАБЛИЦЫ” MODIFY (“ИМЯ_СТОЛБЦА”
ENCRYPT USING “Алгоритм_шифрования”)

ALTER TABLE SOTRUDNIKI MODIFY (№_карты ENCRYPT USING
‘AES256’) – алгоритм AES256

Приложение Б

Приложение Б

Шифрование с помощью TDE на уровне колонок



Прозрачное шифрование данных



Иерархия ключей шифрования в SQL Server



| | | | | | Дипломный проект | | |
|--------------|--------------|-------|-------|------|--|----------|---------|
| | | | | | Лист | Масса | Масштаб |
| Изм. | Лист | Ф.И.О | Подп. | Дата | Методы шифрования в серверах баз данных | | |
| Разработал | Агыбай. | | | | | | |
| Нормоконтр | Зиро А. | | | | | | |
| Руководитель | Айтхожаева Е | | | | | | |
| Зав. каф. | Сейлова Н. | | | | Лист 1 | Листов 4 | |
| | | | | | Тема: Шифрование в реляционных серверах баз данных | | |
| | | | | | КазНИТУ ИИиТТ КБОУиХИ 5В100200 | | |

Приложение Б

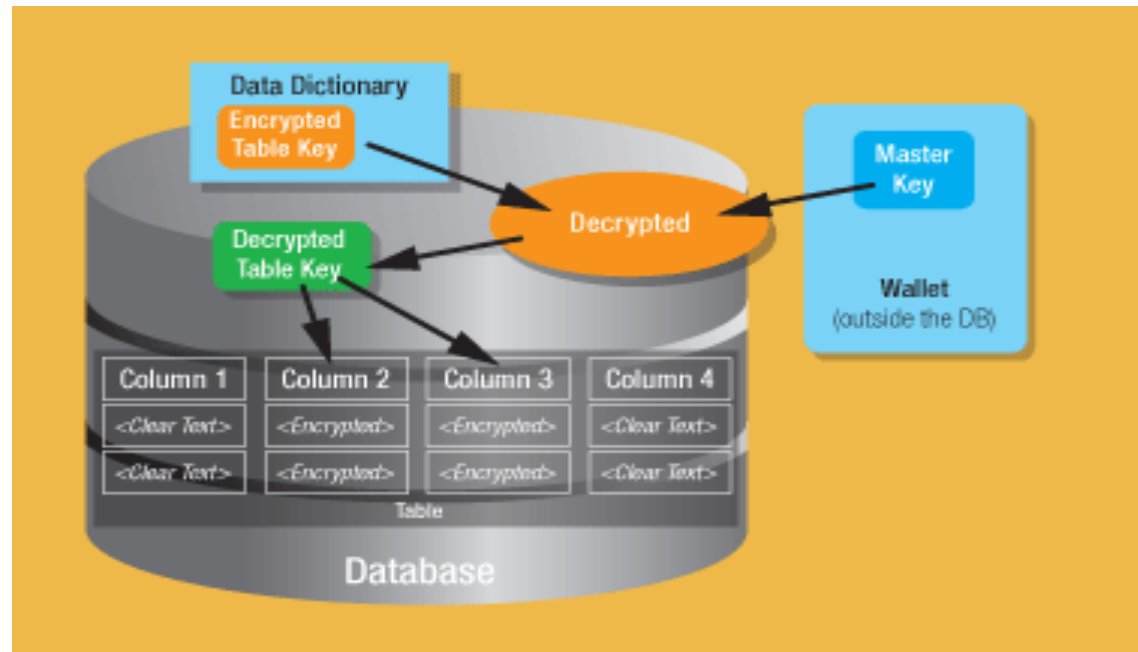
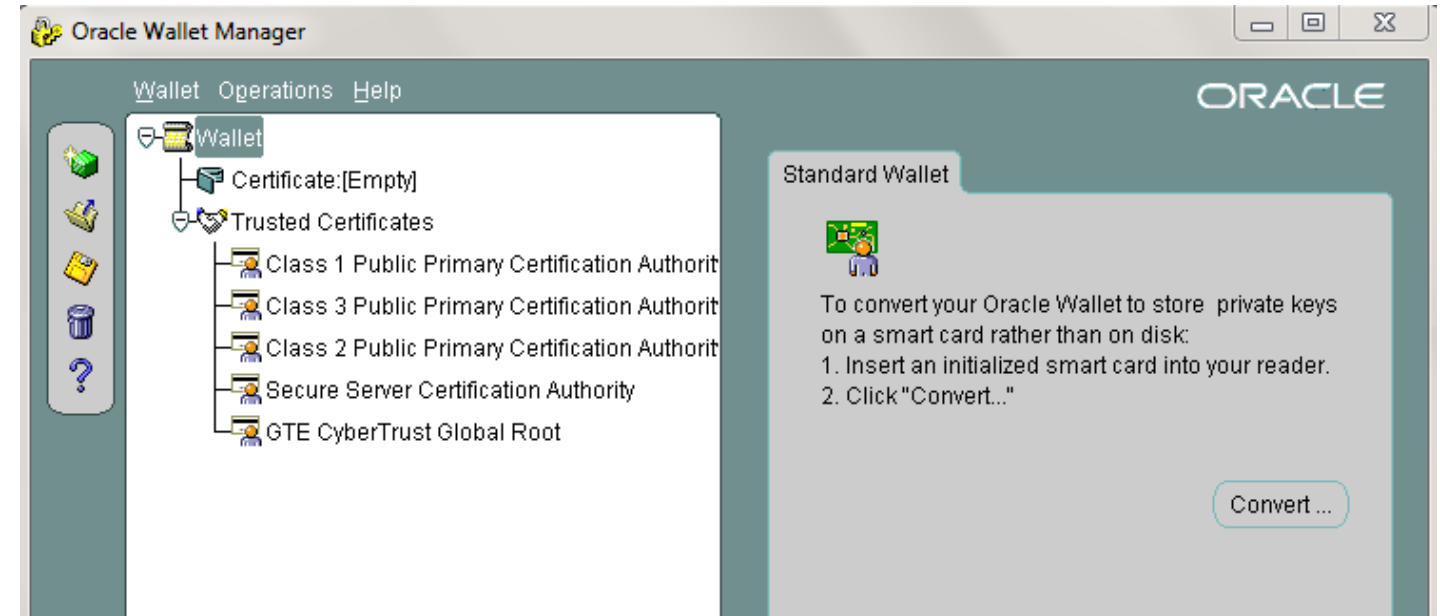
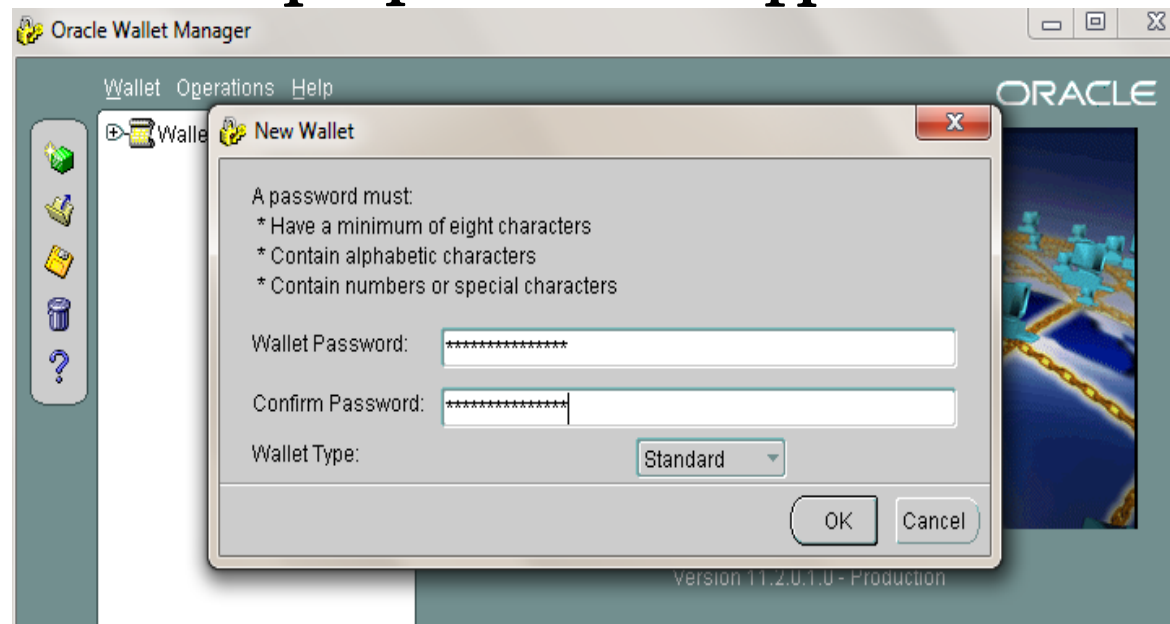


Схема прозрачного шифрования



Сертификаты бумажника



Пароль для доступа к бумажнику

| ID_SOTRUDNIKA | FIO | NO_KARTY | ZARPLATA | TELEFON | STAZH_RABOTY |
|---------------|-------------|----------|----------|-----------|--------------|
| 1 | 435 Shamil | 656255 | 250000 | 877834232 | 5 |
| 2 | 526 Alikhan | 4523456 | 600000 | 874782385 | 10 |
| 3 | 431 Dias | 5464343 | 450000 | 873452364 | 6 |
| 4 | 543 Sanzhar | 657254 | 543234 | 876443232 | 9 |
| 5 | 675 Dastan | 5465365 | 600000 | 87374322 | 7 |
| 6 | 433 Alisher | 7684654 | 500000 | 876553356 | 10 |

Таблица для шифрования

```

ALTER SYSTEM SET encryption key authentication by "KazNITU-2019";
ALTER SYSTEM SET WALLET OPEN AUTHENTICATION BY "KazNITU-2019";
ALTER SYSTEM SET ENCRYPTION WALLET CLOSE;
    
```

Команды для работы с бумажником

| Дипломный проект | | | | | |
|--|------|--------------|-------|------|--|
| Изм. | Лист | Ф.И.О | Подп. | Дата | |
| | | | | | Лист |
| | | | | | Масса |
| | | | | | Масштаб |
| Разработал | | Агыбай. | | | Прозрачное шифрование в Oracle |
| Нормоконтр | | Зиро А. | | | |
| Руководитель | | Айтхожаева Е | | | |
| Зав. каф. | | Сейлова Н. | | | |
| | | | | | Лист 2 |
| | | | | | Листов 4 |
| Тема: Шифрование в реляционных серверах баз данных | | | | | КазНИТУ ИИиТТ КБОУХИ 5В100200 |

Приложение Б

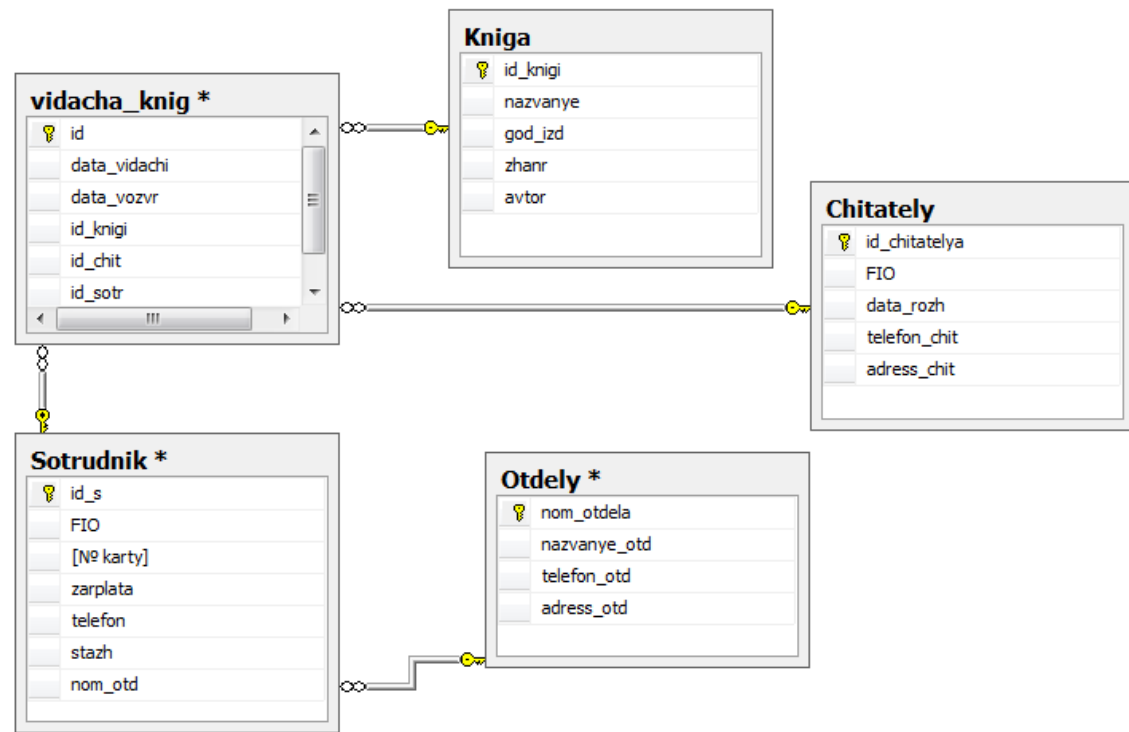


Диаграмма проектируемой базы данных

| Имя столбца | Тип данных | Разрешить ... |
|--------------|------------|-------------------------------------|
| nom_otdela | int | <input type="checkbox"/> |
| nazvanye_otd | nchar(10) | <input checked="" type="checkbox"/> |
| telefon_otd | int | <input checked="" type="checkbox"/> |
| adress_otd | nchar(10) | <input checked="" type="checkbox"/> |

Структура таблицы отдели

| Имя столбца | Тип данных | Разрешить ... |
|-------------|----------------|-------------------------------------|
| id_s | int | <input type="checkbox"/> |
| FIO | nchar(20) | <input checked="" type="checkbox"/> |
| [№ karty] | bigint | <input checked="" type="checkbox"/> |
| zarplata | int | <input checked="" type="checkbox"/> |
| telefon | numeric(18, 0) | <input checked="" type="checkbox"/> |
| stazh | int | <input checked="" type="checkbox"/> |
| nom_otd | int | <input checked="" type="checkbox"/> |

Структура таблицы Сотрудники

| Имя столбца | Тип данных | Разрешить ... |
|--------------|------------|-------------------------------------|
| id | int | <input type="checkbox"/> |
| data_vidachi | date | <input checked="" type="checkbox"/> |
| data_vozvr | date | <input checked="" type="checkbox"/> |
| id_knigi | int | <input checked="" type="checkbox"/> |
| id_chit | int | <input checked="" type="checkbox"/> |
| id_sotr | int | <input checked="" type="checkbox"/> |

Структура таблицы Выдача_книг

| Имя столбца | Тип данных | Разрешить ... |
|---------------|------------|-------------------------------------|
| id_chitatelya | int | <input type="checkbox"/> |
| FIO | nchar(10) | <input checked="" type="checkbox"/> |
| data_rozh | date | <input checked="" type="checkbox"/> |
| telefon_chit | int | <input checked="" type="checkbox"/> |
| adress_chit | nchar(10) | <input checked="" type="checkbox"/> |

Структура таблицы Читатели

| Имя столбца | Тип данных | Разрешить ... |
|-------------|------------|-------------------------------------|
| id_knigi | int | <input type="checkbox"/> |
| nazvanye | nchar(10) | <input checked="" type="checkbox"/> |
| god_izd | date | <input checked="" type="checkbox"/> |
| zhanr | nchar(10) | <input checked="" type="checkbox"/> |
| avtor | nchar(10) | <input checked="" type="checkbox"/> |

Структура таблицы Книга

| Дипломный проект | | | | | |
|------------------|------|--------------|-------|------|--|
| Изм. | Лист | Ф.И.О | Подп. | Дата | Тема: Шифрование в реляционных серверах баз данных |
| | | Агыбай. | | | Лист 3 Листов 4 КазНИТУ ИИиТТ КБОиХИ 5В100200 |
| | | Зиро А. | | | |
| | | Айтхожаева Е | | | |
| | | Сейлова Н. | | | |

Приложение Б

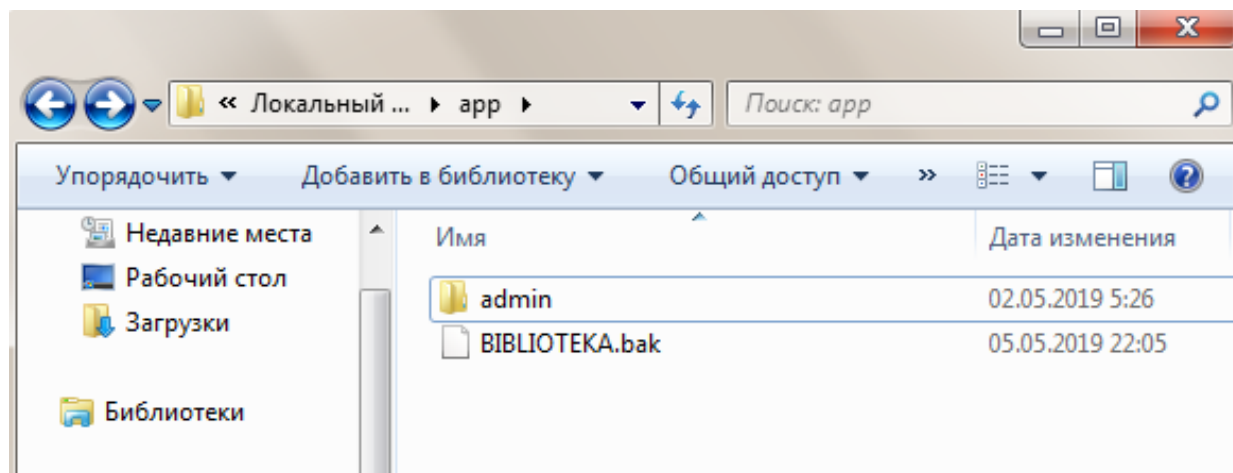
```
SQLQuery1.sql - ADMIN-ПК\..... (54)* ADMIN-ПК\ППП.Би...ка - Diagram_0*
use Библиотека
go
create master key encryption by password='AaBb-123'
select * from sys.key_encryptions
```

| key_id | thumbprint | crypt_type | crypt_type_desc | crypt_property |
|--------|------------|------------|--------------------------|--|
| 1 | 0x01 | ESKM | ENCRYPTION BY MASTER KEY | 0x91B0427E9163811DC53447CA19A0E78C889606F5BB141CA5 |
| 2 | NULL | ESKP | ENCRYPTION BY PASSWORD | 0x1F7E07B41F891FE8A4496969F92156593086DC0D8B59027E |

Создание мастер_ключ

```
SQLQuery1.sql - A...min-ПК\admin (52)*
BACKUP MASTER KEY TO FILE = 'D:\app\BIBLIOTEKA.bak'
ENCRYPTION BY PASSWORD = 'AaBb-123'
```

Создание резервной копии



Проверка резервной копии

```
create certificate certificateTDE
WITH SUBJECT = 'TDE Certificate'
```

Сообщения
Выполнение команд успешно завершено.

Создание сертификата

```
select * from sys.certificates where name='CertificateTDE'
```

| name | certificate_id | principal_id | pvt_key_encryption_type | pvt_key_encryption_type_desc |
|----------------|----------------|--------------|-------------------------|------------------------------|
| certificateTDE | 256 | 1 | MK | ENCRYPTED_BY_MASTER_KEY |

| pvt_key_encryption_type_desc | is_active_for_begin_dialog | issuer_name | cert_serial_number |
|------------------------------|----------------------------|-----------------|---|
| ENCRYPTED_BY_MASTER_KEY | 1 | TDE Certificate | 11 18 92 2d 6d 5e 2c 81 45 cb 80 8a 81 54 90 4b |

Проверка сертификата в системной таблице

```
SQLQuery1.sql - A...min-ПК\admin (52)*
USE Библиотека
CREATE DATABASE ENCRYPTION KEY
WITH ALGORITHM=AES_256
ENCRYPTION
BY SERVER CERTIFICATE CertifacteTDE
```

Сообщения
Выполнение команд успешно завершено.

Создание ключа для базы данных Библиотека

| | | | | | | Дипломный проект | | |
|--------------|------|--------------|-------|------|----------------------------|--|----------|---------|
| | | | | | | Лист | Масса | Масштаб |
| Изм. | Лист | Ф.И.О | Подп. | Дата | Шифрование в MS SQL Server | | | |
| Разработал | | Агыбай. | | | | | | |
| Нормоконтр | | Зиро А. | | | | | | |
| Руководитель | | Айтхожаева Е | | | | Лист 4 | Листов 4 | |
| Зав. каф. | | Сейлова Н. | | | | Тема: Шифрование в реляционных серверах баз данных | | |
| | | | | | | КазНИТУ ИИиТТ КБОиХИ 5В100200 | | |