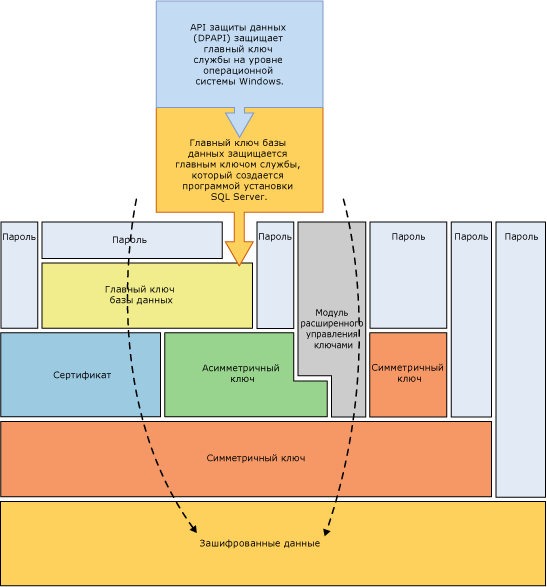
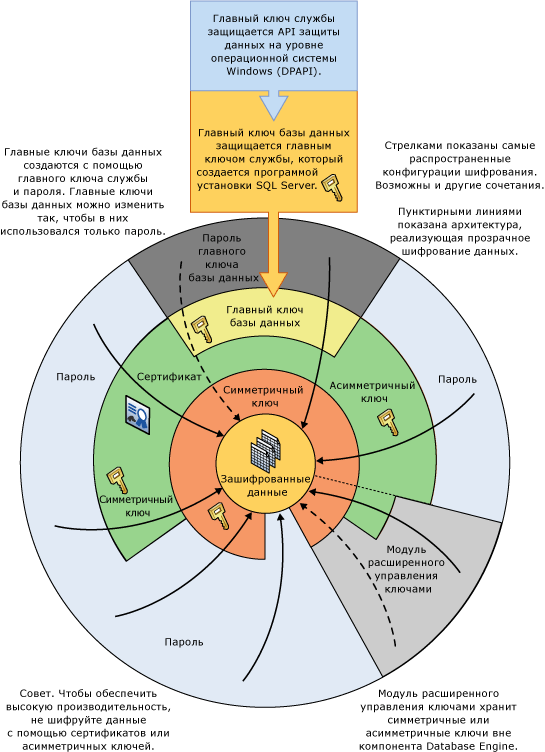
**Рекомендации по выполнению лабораторной работы**

MS SQL SERVER - реляционная СУБД, разработанная компанией Microsoft. Основной язык запросов - Transact-SQL – представляет собой расширение стандартного языка SQL. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия.

Одна из основных проблем безопасного хранения данных на сервере БД – это хранение закрытых данных в открытом виде, в результате чего злоумышленник может получить доступ к данным путем простого запроса на выборку. Рассмотрим иерархию средств шифрования, применяемых в MS SQL.

SQL Server шифрует данные, используя иерархическую структуру средств шифрования и управления ключами. На каждом уровне данные низшего уровня шифруются на основе комбинации сертификатов, асимметричных ключей и симметричных ключей. Асимметричные и симметричные ключи можно хранить вне модуля расширенного управления ключами SQL Server.

Имеются следующие особенности процедуры шифрования, которые необходимо учитывать при обеспечении защиты данных:

* симметричное шифрование работает быстрее, чем ассиметричное, либо шифрование с помощью сертификатов;
* главные ключи базы данных защищены главным ключом службы. Главный ключ службы создается при установке SQL Server и шифруется API-интерфейсом защиты данных Windows (DPAPI);
* иерархия шифрования может быть дополнена другими уровнями;
* модуль расширенного управления ключами хранит симметричные или асимметричные ключи вне SQL Server;
* прозрачное шифрование данных (TDE) должно использовать симметричный ключ, который называется ключом шифрования базы данных, защищенный сертификатом, который, в свою очередь защищается главным ключом базы данных master или асимметричным ключом, хранящимся в модуле расширенного управления ключами;
* главный ключ службы и все главные ключи базы данных являются симметричными ключами.

Шифрование в SQL сервер реализуется следующими механизмами:

1. Функции Transact-SQL

Шифрование данных с помощью парольной фразы выполняется процедурой EncryptByPassPhrase (используется модификация алгоритма DES и 128-битный ключ), а для их дешифрования используется – DecryptByPassPhrase.

1. Сертификаты

Сертификаты представляют собой подписанные цифровой подписью *инструкцию*, которая связывает значение *открытого ключа* с *идентификатором* пользователя, устройства или службы, имеющей соответствующий *закрытый ключ*. Сертификаты поставляются и подписываются центром сертификации (CA), сущность получающая сертификат является *субъектом этого сертификата*. Сертификат содержит: идентификационные данные субъекта, открытый ключ субъекта, срок действия, идентификационные данные поставщика сертификата, цифровую подпись поставщика, которая подтверждает действительность связи между открытым ключом и идентификационными данными субъекта.

Главное преимущество сертификатов в том, что они позволяют не хранить на узлах совокупность паролей отдельных субъектов. Вместо этого узел просто устанавливает доверительные отношения с поставщиком сертификата; после этого поставщик может подписать неограниченное количество сертификатов. Самостоятельно подписанные сертификаты, созданные SQL Server, соответствуют стандарту X.509.

1. Асимметричные ключи

Асимметричный ключ состоит из закрытого ключа и соответствующего открытого ключа. Каждый из этих ключей позволяет дешифровать данные, зашифрованные другим ключом. На выполнение асимметричных операций шифрования и дешифрования требуется сравнительно много ресурсов, но они обеспечивают более надежную защиту, чем симметричное шифрование. Асимметричный ключ можно использовать для шифрования симметричного ключа перед его сохранением в базе данных.

1. Симметричные ключи

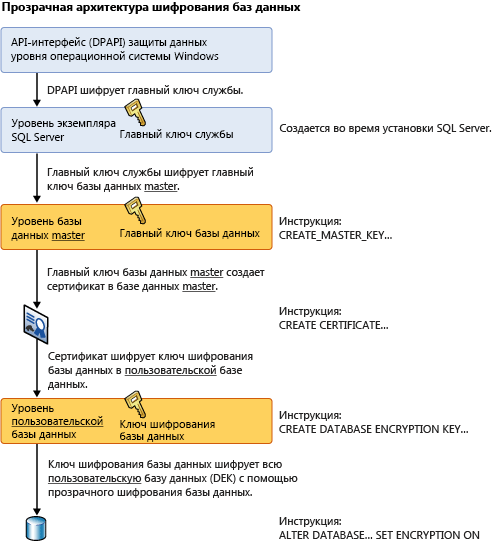
Симметричный ключ — это ключ, используемый и для шифрования, и для дешифрования данных. Данные при использовании симметричного ключа шифруются и дешифруются быстро, и он вполне подходит для повседневной защиты конфиденциальных данных, хранящихся в базе данных.

1. Прозрачное шифрование данных (TDE)

Прозрачное шифрование данных (TDE) является особым случаем шифрования с использованием симметричного ключа. TDE шифрует всю базу данных, используя симметричный ключ, который называется ключом шифрования базы данных. Ключ шифрования базы данных защищен другими ключами или сертификатами, которые, в свою очередь, защищаются главным ключом базы данных или асимметричным ключом, хранящимся в модуле расширенного управления ключами.

Рассмотрим более подробно технологию прозрачного шифрования данных. Функция прозрачного шифрования данных (TDE) выполняет в реальном времени шифрование и дешифрование файлов данных и журналов в операциях ввода-вывода. При шифровании используется ключ шифрования базы данных (DEK), который хранится в загрузочной записи базы данных для доступности при восстановлении. Ключ шифрования базы данных является симметричным ключом, защищенным сертификатом, который хранится в базе данных master на сервере, или асимметричным ключом, защищенным модулем расширенного управления ключами. Функция прозрачного шифрования данных защищает "неактивные" данные, то есть файлы данных и журналов.

Шифрование файла базы данных проводится на уровне страниц. Страницы в зашифрованной базе данных шифруются до записи на диск и дешифруются при чтении в память. Прозрачное шифрование данных не увеличивает размер зашифрованной базы данных. На следующем рисунке показана архитектура прозрачного шифрования данных.



Порядок включения шифрования**:**

1. Создать главный ключ
2. Создать или получить сертификат, защищенный главным ключом
3. Создать ключ шифрования базы данных и защитить его с помощью сертификата
4. Задать ведение шифрование базы данных

Источники:

1. Криптографические функции <https://msdn.microsoft.com/ru-ru/library/ms173744.aspx>
2. Encription Hierarhy <https://msdn.microsoft.com/en-us/library/ms189586.aspx>
3. TDE <http://www.interface.ru/home.asp?artId=36844>
4. Прозрачное шифрование данных <https://msdn.microsoft.com/ru-ru/library/bb934049.aspx>
5. Обеспечение безопасности среды SQL Server [http://www.osp.ru/win2000/2010/ 01/13001638/](http://www.osp.ru/win2000/2010/%2001/13001638/)
6. Включение шифрования соединений в ядре СУБД <https://msdn.microsoft.com/ru-ru/library/ms191192.aspx>

**Задание на самостоятельную работу**

1. Включить шифрование соединений в ядре СУБД SQL Server и описать алгоритм
2. Включить прозрачное шифрование данных и описать каждый шаг алгоритма
3. Создать таблицу в БД и реализовать процедуру (или процедуры) на языке T-SQL, которые предоставляют возможности внесения и получения данных, хранящихся в зашифрованном виде (SELECT вернет зашифрованные данные) с использованием EncryptByPassPhrase/DecryptByPassPhrase.
4. Продемонстрировать процесс шифрования данных с использованием симметричного ключа, зашифрованного с помощью ассиметричного ключа (написать скрипт):
   1. Создать ассиметричный ключ и зашифровать его парольной фразой
   2. Создать симметричный ключ и зашифровать его с помощью ассиметричного ключа
   3. Открыть симметричный ключ и расшифровать его с помощью ассиметричного ключа
   4. Убедиться, что ключ открыт
   5. Внести шифрованные данные в таблицу