**БЕЗОПАСНОСТЬ БАЗЫ ДАННЫХ И ИЗУЧЕНИЕ МЕТОДОВ ШИФРОВАНИЯ ДАННЫХ В ОБЛАЧНОМ ХРАНИЛИЩЕ.**

**к.т.н., проф. А.М. Филоненко, магистр Е.Н. Щербинина, магистр Б.В. Марценюк, НТУ «ХПИ», г. Харьков**

Безопасность данных - самая важная задача в современном мире. Правительства, компании и другие организации потеряли много денег, а многие другие закрылись из-за деятельности хакеров и злоумышленников. За прошедшие годы были разработаны различные схемы шифрования для защиты базы данных от различных атак злоумышленников. Поскольку данные - это жизненный канал каждой организации, существует потребность в удаленном и безопасном хранении данных. Облачное хранилище необходимо для удаленного хранения данных.

Для многих предприятий безопасность данных является одной из основных проблем при отправке файлов в облако. Они беспокоятся о том, что их файлы будут просмотрены или даже скомпрометированы.

Для защиты целостности хранимых данных необходимы методы шифрования данных. В прошлом многие компании чувствовали себя комфортно, позволяя поставщикам облачных услуг управлять всеми своими данными, полагая, что рисками безопасности можно управлять с помощью контрактов, средств контроля и аудита. Однако со временем стало очевидно, что поставщики облачных услуг не могут выполнять такие обязательства, отвечая на запросы правительства о предоставлении информации.

В этом документе обсуждается важность шифрования базы данных и дается подробный обзор различных методов шифрования.

**Ключевые слова:** шифрование, криптография, хеширование, безопасность, база данных, облачное хранилище, зашифрованный текст.

**I. Введение**

В век технологий всю нашу работу выполняют компьютеры. От общения с друзьями в социальных сетях до совершения онлайн-платежей через Интернет-банкинг - все делается онлайн через компьютеры. Поскольку эти средства эффективны и облегчают нашу работу, мы используем их так или иначе. Это означает, что при использовании этих онлайн-сервисов мы сохраняем все наши личные и конфиденциальные данные в базах данных этих веб-сайтов и приложений, что делает эти данные уязвимыми. Таким образом, защита этих важных пользовательских данных является одним из основных приоритетов во избежаниеи неправомерного использования данных [1].

Облачное хранилище - это система удаленного хранения, обслуживания, управления и резервного копирования данных. Услуга доступна пользователям через сеть, которой обычно является Интернет.

Важный способ защиты этих данных - это шифрование данных, хранящихся в базах данных этих веб-сайтов.

**II. Зачем нужно шифрование данных**

Необходимость шифрования данных перед их сохранением в базе данных заключается в том, что ограничение доступа посредством авторизации и аутентификации данных может помочь до определенного времени, но, что если злоумышленник каким-то образом попадет в базу данных. У него есть все данные базы данных, и он может использовать их по своему усмотрению, здесь вступает в игру шифрование данных перед их сохранением в базе данных. Если данные зашифрованы перед сохранением в базе данных, даже с доступом к базе данных, злоумышленник не может использовать эти данные.

Итак, цель статьи - обсудить важность шифрования базы данных и сделать подробный обзор различных методов шифрования.

**III. Шифрование базы данных**

Шифрование базы данных - это процесс шифрования данных в базе данных [2]. Это ключевая стратегия защиты содержимого данных в базе данных. Основная идея заключается в том, что злоумышленник каким-то образом сможет добраться до базы данных системы; из-за шифрования он не сможет злоупотреблять данными в базе данных.

На рис.1 показана основная работа процесса шифрования и дешифрования базы данных. Простой текст/данные, которые будут сохранены в базе данных, сначала преобразуются в зашифрованный текст с использованием соответствующего алгоритма и определенного ключа. Затем этот зашифрованный текст сохраняется в базе данных. Когда пользователь хочет извлечь данные из базы данных, зашифрованный текст преобразуется обратно в простой текст с использованием алгоритма дешифрования и того же ключа, который использовался при шифровании. Это вернет пользователю простой текст по запросу.



Рисунок 1 – Процес шифрования и дешифрования базы данных.

Шифрование базы данных можно выполнить двумя способами:

Шифрование: это процесс, в котором простой текст преобразуется в зашифрованный текст с помощью ключа, а затем, используя тот же ключ, мы можем расшифровать зашифрованный текст обратно в простой текст [3]. Шифрование выполняется с использованием различных алгоритмов, каждый из которых имеет свои преимущества и недостатки. Чаще всего используются алгоритмы шифрования DES, RC2, AES128, AES 256 и т. д. На рис.2 показано выполнение простого процесса шифрования.

Хеширование: это односторонний процесс, в котором простой текст преобразуется в хешированное значение (зашифрованная форма). После того, как данные хешируются с помощью функции хеширования, их нельзя изменить обратно на обычный текст [3]. Обычно этот подход используется для шифрования паролей, всякий раз, когда нам нужно войти в систему, введенный пароль шифруется с использованием хеш-функции, а затем сопоставляется с паролем, хранящимся в базе данных, которая уже находится в зашифрованной форме, если оба совпадают - пользователь получает доступ, иначе он получает сообщение о неверном имени пользователя/пароле. Наиболее часто используемые хеш-функции - это MD4, MD5, SHA, SHA-1 и т. д. На рис. 3 показана работа хеширования.

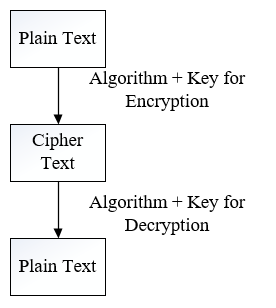


Рисунок 2 – Процесс шифрования



Рисунок 3 – Процесс хеширования

**IV. Шифрование данных облака**

Облачное шифрование - это услуга, предлагаемая поставщиками облачного хранилища, при которой данные или текст преобразуются с использованием алгоритмов шифрования и затем помещаются в облако хранилища. Облачное шифрование практически идентично внутреннему шифрованию с одним важным отличием: клиенту облачного сервиса необходимо время, чтобы узнать о политиках и процедурах провайдера для шифрования и управления ключами шифрования. Возможности облачного шифрования поставщика услуг должны соответствовать уровню конфиденциальности размещаемых данных. Поскольку шифрование потребляет больше ресурсов процессора, многие поставщики облачных услуг предлагают базовое шифрование только для нескольких полей базы данных, таких как пароли и номера учетных записей [4]. В этот момент использование провайдером шифрования всей базы данных клиента может стать настолько дорогостоящим, что может иметь смысл хранить данные внутри компании или зашифровать данные перед их отправкой в ​​облако. Чтобы снизить затраты, некоторые поставщики облачных услуг предлагают альтернативы шифрованию, которые не требуют такой вычислительной мощности. Эти методы включают в себя редактирование или обфускацию данных, которые должны оставаться конфиденциальными, или использование собственных алгоритмов шифрования, созданных поставщиком [5].

**V. Методы облачного шифрования данных**

- Блочные шифры и потоковые шифры.

Один из основных методов категоризации для широко используемых методов шифрования основан на форме входных данных, с которыми они работают. Это два типа: блочный шифр и потоковый шифр. В этом разделе обсуждаются основные функции двух типов, рабочий режим, и сравниваются их с точки зрения безопасности и производительности.

Блочный шифр.

В этом методе шифрования данные шифруются и дешифруются, если данные находятся в одном из блоков. В простейшем режиме вы делите простой текст на блоки, которые затем передаются в систему шифрования для создания блоков зашифрованного текста. ECB (режим электронной кодовой книги) - это основная форма блочного шифра, в которой блоки данных зашифровываются напрямую для генерации соответствующих зашифрованных блоков.

Потоковые шифры.

Потоковый шифр работает с потоком данных, работая с ним бит за битом. Потоковый шифр состоит из двух основных компонентов: генератора ключевого потока и функции смешивания. Функция смешивания обычно представляет собой просто функцию XOR, в то время как генератор потока ключей является основным блоком в технике шифрования потокового шифрования. Например, если генератор ключевого потока выдает серию нулей, выводимый зашифрованный поток будет идентичен исходному простому тексту.

- Симметричное и асимметричное шифрование

Процедуры шифрования данных в основном делятся на две категории в зависимости от типа ключей безопасности, используемых для зашифровывания/расшифровывания защищенных данные. Эти две категории: Асимметричные и симметричные методы шифрования.

Симметричное шифрование.

В этом типе шифрования отправитель и получатель согласовывают секретный (общий) ключ. Затем они используют этот секретный ключ для шифрования и дешифрования своих отправленных сообщений. На рис. 4 показан процесс симметричной криптографии. Узлы A и B сначала согласовывают метод шифрования, который будет использоваться при шифровании и дешифровании передаваемых данных. Затем они договариваются о секретном ключе, который они оба будут использовать в этой связи. После завершения настройки шифрования узел A начинает отправлять свои данные, зашифрованные с помощью общего ключа, на другой стороне узел B использует тот же ключ для дешифрования зашифрованных сообщений.

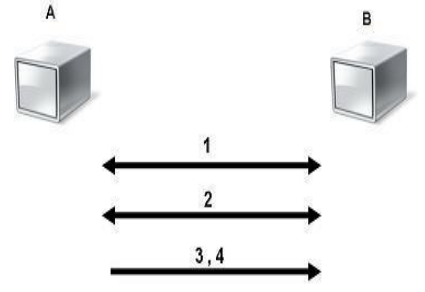


Рисунок 4 – Симметричное шифрование

Основная проблема симметричного шифрования заключается в том, как безопасно разделить секретный ключ между двумя узлами. Если ключ становится известным, по какой-либо причине, вся система рушится. Управление ключами для этого типа шифрования проблематично, особенно если уникальный секретный ключ используется для каждого однорангового соединения, тогда общее количество секретных ключей, которые будут сохранены и управляться для n узлов, будет n (n- 1) / 2.

Асимметричное шифрование.

Оно также известно как криптография с открытым ключом (PKC), потому что пользователи обычно используют два ключа: открытый ключ, который известен всем, и закрытый ключ, который известен только пользователю. На рис.5 ниже показано использование двух ключей между узлом A и узлом B. После согласования типа шифрования, который будет использоваться в соединении, узел B отправляет свой открытый ключ узлу A. Узел A использует полученный открытый ключ для шифрования его сообщения. Затем, когда приходят зашифрованные сообщения, узел B использует свой закрытый ключ для их расшифровки.

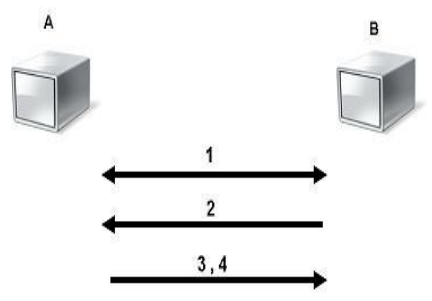


Рисунок 5 – Асимметричное шифрование.

Эта возможность решает проблему симметричного шифрования управления секретными ключами. Но с другой стороны, эта уникальная особенность шифрования с открытым ключом делает его математически более уязвимым для атак. Более того, методы асимметричного шифрования почти в 1000 раз медленнее, чем методы симметричного шифрования, поскольку они требуют большей вычислительной мощности. Чтобы получить преимущества обоих методов, обычно используется гибридная техника. В этом методе асимметричное шифрование используется для обмена секретным ключом, а затем используется симметричное шифрование для передачи данных между отправителем и получателем [6].

Например, в открытой системе, учитывая любые два принципала X и Y, X должен быть в состоянии зашифровать сообщение, которое может быть расшифровано только Y. Если существует некоторая привязка между основными идентификаторами и открытыми ключами, то эти операции могут легко быть исполненным. Наивная схема могла бы работать следующим образом:

1. принципал X ищет открытый ключ KY для принципала Y и использует его для вычисления шифрования для Y с помощью некоторой функции-лазейки ;
2. затем Y, получив это сообщение, вычисляет .

Но с этой схемой есть существенная проблема, учитывая наши определения безопасности для шифрования с общим ключом: она не удовлетворяет семантической безопасности, поскольку злоумышленник легко вычислит и и сравнит их с заданными шифротекстами в различных моделях атак. И снова мы видим, что без вероятностного шифрования не может быть семантической безопасности. Это особенно верно при настройке открытого ключа, поскольку каждый принципал по определению имеет доступ к функции шифрования для каждого другого принципала. Особенно, когда пространство возможных сообщений невелико, легко просто проверить все сообщения с помощью функции шифрования, чтобы выяснить, что было зашифровано.

**VI. Выводы**

С развитием технологий в настоящее время все делается с помощью компьютеров, поэтому безопасность этих данных в базе данных становится важной проблемой. Многие исследователи работали над этим и предлагали различные алгоритмы и архитектуры. У каждой схемы есть свои достоинства и недостатки. Но ни одна из них не является полностью безопасной и содержит определенные лазейки или недостатки, которые могут использоваться хакерами и злоумышленниками для получения доступа к базе данных. Так что в этой области есть возможности для улучшения.

Многие исследовательские проблемы еще предстоит определить. Криптографические методы используются для обеспечения безопасной связи между пользователем и облаком. Симметричное шифрование обладает скоростью и вычислительной эффективностью, чтобы обрабатывать шифрование больших объемов данных в облачном хранилище. В этой статье предложен алгоритм симметричного шифрования для безопасного хранения облачных пользовательских данных в облачном хранилище. Предлагаемый алгоритм шифрования описан подробно, а процесс дешифрования является обратным шифрованию. Этот алгоритм используется для шифрования данных пользователя в облаке. Поскольку после выхода из сеанса пользователь не может контролировать данные, ключ шифрования действует как первичная аутентификация для пользователя.

**Список литературы**

1. Бараани-Дастджерди, Ахмад, Йозеф Пепшик и Рейхане Сафави-Найни. «Безопасность в базах данных: исследование». Департамент компьютерных наук Университета Вуллонгонга (1996 г.).
2. Денни Черри и Томас Ларок, «2 - Шифрование базы данных при защите SQL-сервера», под редакцией Денни Черри, Томас Ларок, Syngress, Бостон, 2011 г., страницы 27-71, ISBN: 9781597496254.
3. Кесслер, Гэри К. «Обзор криптографии». (2003). (http://www.sciencedirect.com/science/article/pii/B97815974962541000 22).
4. Вамси Кришна, Ярлагадда и Шрирам Рамануджам, «Безопасность данных в облачных вычислениях», Журнал компьютерных и математических наук, том 2 (1), стр. 15–23, 2011 г.
5. Питер Мелл, Тим Гранс, «Эффективное и безопасное использование парадигмы облачных вычислений», NIST, Лаборатория информационных технологий, http://www.csrc.nist.gov/groups/SNS/cloud-computing / cloudcomputingv26.ppt. 2009 г.
6. Эман М. Мохамед, Хатем С. Абделькадер и Шериф Эль-Этриби, «Модель безопасности данных для облачных вычислений», Двенадцатая международная конференция по сетям, ISBN: 978-1-61208-245-5, стр 66-74, 2013.