УДК 004.056.55: 003.26
https://doi.org/10.47533/2020.1606-146X.102

А А.КУЛЬЖАНОВА*, Д. Р. РАХИМОВА, Т. И. БАКИБАЕВ

Казахский национальный университет имени аль-Фараби, Алматы, Казахстан
Алматинский Менеджмент Университет, Алматы, Казахстан
akbota.kulzhanova1594@gmail.com, di.diva@mail.ru,
@gmail.com, timurbakibayev@gmail.com

ПРОБЛЕМЫ ПРОЦЕССА БЛОЧНОГО ШИФРОВАНИЯ ДАННЫХ И ЕГО ПРИМЕНЕНИЕ ДЛЯ ПОВЫШЕНИЯ БЕЗОПАСНОСТИ ДАННЫХ

В данной статье рассматривается анализ существующих методов шифрования и механизмов данных во избежание возможности проблемы частичного обновления данных в больших файлах. Например, нам нужно идентифицировать идентичные блоки без возможности расшифровать данные. Мы сталкиваемся с проблемой, когда файлы зашифрованы разными ключами, то зашифрованные версии будут разными. Также стоит проанализировать и выбрать наилучший алгоритм шифрования для шифрования и дешифрования файлов. В данной статье мы обращаем внимание на безопасность системы. Наилучшим решением является разработка системы шифрования и дешифрования файлов, которое позволяет хранить файлы, как и на локальном компьютере, так и на различных носителях, что и является результатом данной статьи, чтобы оптимизировать эффективность шифрования, скорость передачи данных, скорость обработки изменений файлов, необходимых для хранения файлов в пространстве.

***Ключевые слова**: безопасность облачных вычислений, конфиденциальность, безопасность данных, дедупликация, шифрование.*

Введение. С развитием технологий развивается и информация. Миллиарды людей по всему миру имеют дело с терабайтами данных. Из-за роста информации растет и необходимость в ее защите и обработке. Помимо обработки и защиты возникают трудности с записью и размещением информации. Иногда использование мощных серверов приводит к увеличению затрат, но это не решает всей проблемы [1].

Сегодня существует множество алгоритмов для шифрования и дешифрования данных. Но и с ними возникает проблема избыточности данных, например, если два разных пользователя загрузили один и тот же файл, нет смысла хранить обе копии, так как избыточный файл занимает место, вследствие чего мы имеем просто ненужную копию. В данном случае необходимо определить идентичные блоки без возможности дешифрования данных. В этом случае проблема усугубляется тем, что если файлы были зашифрованы разными ключами, зашифрованные версии будут разными. Одно

* E-mail корреспондирующего автора: akbota.kulzhanova1594@gmail.com

из хороших решений - метод горизонтального масштабирования. В этом случае необходимо увеличить емкость хранилища, время отклика, пропускную способность [2, 3,5].

Теперь у нас увеличилась потребность в хранении данных по сравнению с предыдущими годами. Облачное хранилище помогает удовлетворить эту потребность, предоставляя пользователям расширенную емкость и доступ. Дедупликация данных –обычная практика для поставщиков облачных услуг. Дедупликация данных – это процесс, в котором должны быть идентифицированы идентичные копии пользовательских данных, а все остальные копии, кроме одной, должны быть удалены. Это сделано для уменьшения накладных расходов на хранилище. Но здесь одна проблема – конфиденциальность информации [4].

Перед тем как использовать те же облачные хранилища, мы не задумываемся о безопасности наших данных. Наша цель в будущем – обеспечить безопасность данных в импровизированном облаке, при этом избегая избыточности данных. Первоначальной задачей являлся выбор правильного алгоритма шифрования и разработки дешифратора, шифратора файлов для дальнейшей интеграции в облако.

С развитием цифровых устройств и спросом на них на рынке выросла потребность потребителей в доступе к своим данным из любого места и в любое время. По мере роста потребителей и объемов данных управление информацией становится дорогостоящим и сложным и возникает большой риск кражи информации. Стремительный рост использования Интернета, хранения и защиты данных требует новых способов управления данными, таких как размер, разнообразие и доступность. Это то, что делают облачные вычисления [6].

Облачные вычисления используются для предоставления ресурсов потребителям в качестве услуги, к которой потребитель может получить доступ через Интернет. Это упрощает работу, поскольку больше не нужно много оборудования для хранения всех данных. Человек, работающий с облачными вычислениями, больше не зависит от инфраструктуры, которая полностью контролируется поставщиком услуг [8, 9].

Что касается безопасности, то на данный момент это большая проблема для всех информационных технологий. Когда безопасность используется в неконтролируемой среде, такой как облачные вычисления, эта проблема усугубляется. Понимание того, что существуют риски безопасности, связанные с использованием облачных вычислений, дает нам понимание того, что инструменты безопасности должны быть улучшены. Риски в облачных вычислениях связаны с открытыми, общими и распределенными средами. Для более глубокого понимания природы рисков следует провести полный анализ этих рисков. Анализ рисков предполагает разделение существующих проблем и проблем, которые были получены в результате облачных вычислений [4, 5, 6].

Как уже упоминалось, традиционная инфраструктура отличается от инфраструктуры облачных вычислений. Мы заинтересованы в повышении безопасности облачных инфраструктур, а также в повышении безопасности данных. Отличительные характеристики облачных инфраструктур от традиционных инфраструктур позволяют обнаруживать все больше и больше проблем безопасности данных, связанных с характеристиками облака и жизненным циклом облачных данных. Но также эти данные могут принести много неудобств, которые вызваны ухудшением безопасности [7].

Рассмотрим, как вычисления происходят в облаке. Процесс выглядит следующим образом: пользователь отправляет свои данные в дата-центр. Затем данные доставляются на виртуальные машины, которые выполняют параллельные вычисления. После окончания обработки данных пользователи получают доступ к данным. В ходе этого процесса возможна утечка личных или конфиденциальных данных [7].

Исходя из этих фактов, существует три состояния данных в облаке. Первое состояние – это состояние покоя данных или данных, которые находятся в процессе сохранения или предварительной обработки данных. Второе состояние – данные в процессе передачи, это уже переходное состояние. Третье и последнее состояние – это данные, которые можно использовать, то есть имеются данные, к которым есть доступ и которые можно легко обработать. Безопасность данных, конфиденциальность данных и использование данных охватывают все три аспекта. Для обеспечения безопасности и целостности данных на всех этапах жизненного цикла данных могут быть реализованы различные механизмы и меры, которые могут быть предприняты заранее или во время обработки данных [10]. О мерах и технологиях обеспечения безопасности данных в облачных вычислениях мы поговорим в следующей части.

Существующие проблемы. Нет уверенности, что все вышеперечисленное полностью удовлетворительно и обеспечивает полную защиту, но у нас есть начальные этапы составления и обоснование, которые полностью отражают понимание облачных архитектур. Было сказано, что технологии не стоят на месте и развиваются, но, к сожалению, на данный момент у нас нет такой услуги, которая соответствовала бы нашим условиям. Почему это происходит? Потому что для крупных компаний это не совсем выгодно. Изучив материал, становится понятно, что описания этой проблемы практически нигде нет, что учит необходимости придумывать и совершенствовать технику и безопасность.

У нас есть несколько проблем с безопасностью, таких как:

1. Владельцы облака предоставляют свои решения для шифрования. Отсюда идет защита только от неавторизованных пользователей, но доступ к данным имеет сам владелец, что является одной из проблем.

2. Также есть проблемы с памятью. Чаще всего забивается архив или сам контейнер, который был зашифрован. При смене данных возникают некоторые неудобства, приходится скачивать громкий файл целиком.

3. Многие путают понятие защиты канала связи и облака, думая, что VPN защитит все их файлы и избавит от всех неприятностей.

4. Многие программы на рынке сами шифруют файлы, получая их из облака. Но в этом случае необходимо загрузить копию на свой локальный компьютер, что также может быть чрезвычайно неудобным из-за полной памяти локального компьютера. Поработав с данными (зашифрованными), эти данные возвращаются из программы в облако, что тоже крайне неудобно.

Методология исследования. В своей работе мы используем простые криптографические примитивы и методы защиты данных. Например, блочное шифрование. Блочное шифрование служит основным строительным блоком в нашей работе. Существует множество примеров использования блочных шифров для защиты данных [11, 12].

Предположим, что у нас уже есть некий файл в облаке. Но как насчет шифрования данных? Основная цель шифрования данных – защитить информацию от посторонних. Что мы и делаем в наших исследованиях. Для достижения этих целей

необходим грамматический метод работы с данными, которые содержат три этапа по текстовым блокам [13]. Этот метод работает с простыми операциями, основанными на генерации ключей. Мы решаем сразу три задачи: во-первых, мы можем создавать файлы, не опасаясь за свою безопасность; во-вторых, мы используем довольно простой алгоритм шифрования; и третье – самое важное преимущество состоит в том, что у нас есть фактор случайности, который помогает избежать кражи и нанесения ущерба нашим данным.

При традиционном подходе шифрование происходит следующим образом – информация шифруется вместе с ключом, и только после этого ее можно передать в облачное хранилище. Мы уже упоминали, что это одна из проблем, потому что мы просто заполняем память нашего хранилища, тем самым усложняя процесс быстрой обработки данных. Это дает еще одну проблему: если были загружены два одинаковых файла, то мы просто не можем это определить. Наш подход помогает избежать этого без ущерба для безопасности, что доказывает эффективность этого метода для безопасного хранения информации в облаке [14].

Когда пользователь использует дедупликацию данных, ее можно выполнить перед загрузкой в облако или после загрузки данных в хранилище, то есть на стороне сервера. Кроме того, дедупликация имеет различные уровни детализации, здесь мы говорим об уровнях файла или блока. У обоих методов есть свои плюсы и минусы. При более внимательном рассмотрении на уровне файлов сравниваются первые два файла – файл системы хранения для проверки и файл на уровне дедупликации, чтобы убедиться, что тот же файл еще не существует. Чаще всего используется механизм дедупликации на стороне клиента, поскольку он позволяет избежать проблем с загрузкой сети и пропускной способностью. Если мы говорим об отсутствии дедупликации на уровне файла, небольшое изменение в файле приводит к повторному восстановлению всего файла, что нарушает технику дедупликации. Тогда ситуацию можно спасти измененным блоком на уровне блока, а не всего файла. Здесь мы пытаемся добиться лучшей производительности сети, потому что индексы, сгенерированные для идентификации файла, ниже при дедупликации на уровне файла.

Важным фактором в нашем исследовании являются алгоритмы поиска изменений на примере git. Рассмотрите возможность внесения изменений в репозиторий. Допустим, у вас есть некоторые изменения данных в вашем репозитории Git, также есть копия этих данных. Процесс изменения и фиксации состояния в виде снимков этих изменений в репозитории происходит каждый раз, когда у нас нет состояния, которое мы хотим сохранить. Данные, хранящиеся в файле, могут быть двух типов: отслеживаемые и неотслеживаемые. Отслеживаемые файлы – это файлы, о которых знает Git. Остальные – это неотслеживаемые файлы. Во время первого клонирования репозитория файлы будут иметь статус отслеживаемых и немодифицированных, потому что Git только что их проверил, а мы ничего не редактировали. Только после редактирования данных Git будет считать их измененными. Цикл довольно простой, первые изменения файла индексируются, затем записываются все проиндексированные изменения. И этот цикл повторяется каждый раз, когда вы делаете коммит [15].

Перед нами стояла задача подобрать правильный алгоритм шифрования, который был бы достаточно устойчивым. Выбор пал на алгоритм AES-256. AES является симметричным алгоритмом блочного шифрования. По результатам тестирования данный алгоритм является широко используемым в данное время. Кроме того, является улучшенной версией DES.

Почему именно блочное шифрование? Мы используем его по причине того, что процедуры шифрования и дешифрования идентичны, но они будут отличаться лишь порядком действий. Данное свойство помогает во многом, например, при создании устройства шифрования, потому что используются одни и те же блоки для шифрования и дешифрования данных. Именно из-за гибкости блочного шифрования он является наилучшим решением для его использования в наших целях. Еще одним большим преимуществом блочного шифрования является его способность зашифровывать одним и тем же ключом один или несколько файлов. Также выбор симметричного алгоритма шифрования обосновывается его быстротой по сравнению с асимметричными алгоритмами. И симметричный алгоритм более надежен, так как получатель сообщения должен знать секретный ключ, который нужно вернуть уже по определенному зашифрованному каналу. Первым делом нужен непосредственно сам файл с данными (рисунок 1). Далее с помощью программы шифратора-дешифратора указываем путь к данному файлу, шифруем его (рисунок 2) и получаем зашифрованный файл (рисунок 3). Также можно и дешифровать ранее зашифрованный файл, используя хранилище ключей.

Рисунок 1 – Файл до шифровки данных.

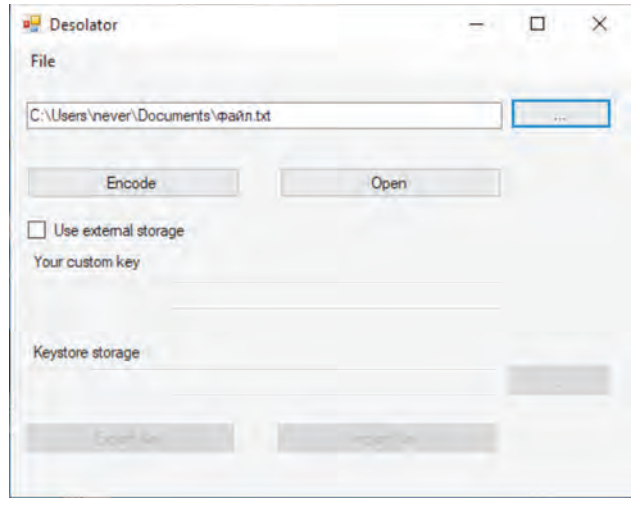

Рисунок 2 – Процесс шифрования данных.

Рисунок 3 – Файл, прошедший процесс шифровки

Дедупликация уменьшает объем пространства, необходимого для определенного набора файлов. В настоящее время в области облачных вычислений существует несколько способов смягчения атак и, таким образом, защиты данных. Простой пример кражи данных – опасность дублирования на стороне клиента. Мошенник может получить доступ к данным, просто угадав хеш-функцию. Чтобы всего этого избежать и снизить вероятность кражи, злоумышленнику предлагается способ подтверждения проверки. То есть клиент проверяет свои данные, что позволяет минимизировать кражу данных или взлом. И, наконец, когда дело доходит до беспокойства клиента о безопасности данных, хранящихся в базе данных поставщика, также не нужно опасаться, что поставщик сможет увидеть данные, поскольку файл будет зашифрован.

Заключение. В процессе написания статьи были изучены механизмы и методы шифрования файлов в облачных системах, существующие методы дедупликации данных в облачных системах, а также был написан шифратор и дешифратор файлов.

Дальнейшие ожидаемые действия:

Наша будущая работа включает разработку импровизированного облака, в котором будет позволено вносить изменения в исходный код без полного изменения зашифрованной версии файла. На сервер будут отправляться только зашифрованные изменения, экономя трафик и время. Будут учтены современные методы дедупликации данных в облаке.

ЛИТЕРАТУРА

1 Чжан Ю., Сюй С., Шэнь Х. С. Безопасность данных в облачном хранилище. – Спрингер, 2020. – С. 1-171.

2 Кача Л., Зитуни А. Обзор безопасности данных в облачных вычислениях //Труды по вычислительным методам в системах и программном обеспечении. – Спрингер, Чам, 2017. – С. 250-261.

3 Альбугми А. и др. Безопасность данных в облачных вычислениях //Пятая Международная конференция 2016 года по коммуникационным технологиям будущего поколения (FGCT). – IEEE, 2016. – С. 55-59.

4 Чжан Ю., Сюй С., Шэнь Х. С. Безопасность данных в облачном хранилище. – Спрингер, 2020. – С. 1-171.

5. Ахмед Х. А. С. и др. Обзор проблем и рисков безопасности облачных вычислений // Журнал телекоммуникаций, электронной и вычислительной техники (JTEC). – 2017. – Т. 9. – №. 1-2. – С. 87-91.

6 Латиф Р. и др. Оценка рисков облачных вычислений: систематический обзор литературы //Информационные технологии будущего. – Спрингер, Берлин, Гейдельберг, 2014. – С. 285-295.

7 Амато Ф. и др. Повышение безопасности в облаке путем формального моделирования ресурсов IaaS //Компьютерные системы будущего поколения. – 2018. – Т. 87. – С. 754-764.

8 Садику М. Н.О., Муса С. М., Момох О. Д. Облачные вычисления: возможности и проблемы //Потенциал IEEE. – 2014. – Т. 33. – №. 1. – С. 34-36.

9 Арора Р., Парашар А., Преобразование С. С. I. Защищенных пользовательских данных в облачных вычислениях с использованием алгоритмов шифрования //Международный журнал инженерных исследований и приложений. – 2013. – Т. 3. – №. 4. – С. 1922-1926

10 Ван К. и др. Публичный аудит с сохранением конфиденциальности для безопасного облачного хранилища //Письма по компьютерной архитектуре IEEE. – 2013. – Т. 62. – №. 02. – С. 362-375.

11Сухак М. и др. Удаленный аудит данных в средах облачных вычислений: обзор, таксономия и открытые проблемы //ACM Computing Surveys (CSUR). – 2015. – Т. 47. – №. 4. – С. 1-34.

12 Ворку С. Г. и др. Безопасная и эффективная схема публичного аудита с сохранением конфиденциальности для облачных хранилищ //Компьютеры и электротехника. – 2014. – Т. 40. – №. 5. – С. 1703-1713.

13 РОЙ, Чандрима; ПАНДЕЙ, Манджуша; СВАРУПРАУТАРАЙ, Сиддхартх. Предложение по оптимизации узла данных путем горизонтального масштабирования узла имени с использованием инструментов больших данных. В: 2018 3-я Международная конференция по конвергенции технологий (I2CT). IEEE, 2018. стр. 1-6.

14 ААКИБ, Сайед Мутахар. Эффективный Кластерный подход для оценки вертикальной и горизонтальной масштабируемости веб-серверов с использованием линейных и нелинейных рабочих нагрузок. В: 2019 3-я Международная конференция по тенденциям в области электроники и информатики (ICOEI). IEEE, 2019. стр. 287-291.

15 Чжан Ю., Сюй С., Шэнь Х. С. Безопасная дедупликация //Безопасность данных в облачных хранилищах. – Спрингер, Сингапур, 2020. – с. 55-86.

REFERENCES

1 Zhang Y., Xu C., Shen X. S. Data Security in Cloud Storage. – Springer, 2020. – C. 1-171.

2 Kacha L., Zitouni A. An overview of data security in cloud computing //Proceedings of the Computational Methods in Systems and Software. – Springer, Cham, 2017. – C. 250-261.

3 Albugmi A. et al. Data security in cloud computing //2016 Fifth International Conference on Future Generation Communication Technologies (FGCT). – IEEE, 2016. – C. 55-59.

4 Zhang Y., Xu C., Shen X. S. Data Security in Cloud Storage. – Springer, 2020. – C. 1-171.

5 Ahmed H. A. S. et al. A review of challenges and security risks of cloud computing //Journal of Telecommunication, Electronic and Computer Engineering (JTEC). – 2017. – T. 9. – №. 1-2. – C. 87-91.

6 Latif R. et al. Cloud computing risk assessment: a systematic literature review //Future information technology. – Springer, Berlin, Heidelberg, 2014. – C. 285-295.

7 Amato F. et al. Improving security in cloud by formal modeling of IaaS resources //Future Generation Computer Systems. – 2018. – Т. 87. – С. 754-764.

8 Sadiku M. N. O., Musa S. M., Momoh O. D. Cloud computing: opportunities and challenges // IEEE potentials. – 2014. – Т. 33. – №. 1. – С. 34-36.

9 Arora R., Parashar A., Transforming C. C. I. Secure user data in cloud computing using encryption algorithms //International journal of engineering research and applications. – 2013. – Т. 3. – №. 4. – С. 1922-1926

10 Wang C. et al. Privacy-preserving public auditing for secure cloud storage //IEEE Computer Architecture Letters. – 2013. – Т. 62. – №. 02. – С. 362-375.

11 Sookhak M. et al. Remote data auditing in cloud computing environments: a survey, taxonomy, and open issues //ACM Computing Surveys (CSUR). – 2015. – Т. 47. – №. 4. – С. 1-34.

12 Worku S. G. et al. Secure and efficient privacy-preserving public auditing scheme for cloud storage //Computers & Electrical Engineering. – 2014. – Т. 40. – №. 5. – С. 1703-1713.

13 ROY, Chandrima; PANDEY, Manjusha; SWARUPRAUTARAY, Siddharth. A proposal for optimization of data node by horizontal scaling of name node using big data tools. In: 2018 3rd International Conference for Convergence in Technology (I2CT). IEEE, 2018. p. 1-6.

14 AAQIB, Syed Mutahar. An Efficient Cluster-Based Approach for Evaluating Vertical and Horizontal Scalability of Web Servers using Linear and Non-Linear Workloads. In: 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI). IEEE, 2019. p. 287-291.

15 Zhang Y., Xu C., Shen X. S. Secure Deduplication //Data Security in Cloud Storage. – Springer, Singapore, 2020. – p. 55-86.

А. А. КУЛЬЖАНОВА, Д. Р. РАХИМОВА, Т. И. БАКИБАЕВ

Әл-Фараби атындағы Қазақ ұлттық университеті, Алматы., Қазақстан
Алматы Менеджмент Университеті, Алматы., Қазақстан

ДЕРЕКТЕРДІ БЛОКТЫҚ ШИФРЛАУ ПРОЦЕСІНІҢ ПРОБЛЕМАСЫ ЖӘНЕ ДЕРЕКТЕРДІҢ ҚАУІПСІЗДІГІН АРТТЫРУ ҮШІН ОНЫҢ ҚОЛДАНЫСЫ

Бұл мақалада үлкен көлемді файлдарда деректерді ішінара жаңарту мәселесін болдырмау мақсатында қолданыстағы шифрлау әдістері мен мәліметтердің тетіктерінің талдауы қарастырылады. Мысалы, біз деректерді шифрды ашуға мүмкіндігі жоқ бірдей блоктарды анықтауымыз керек. Бізде файлдар әр түрлі кілттермен шифрланған болса, онда шифрланған нұсқалар әр түрлі болады деген мәселеге тап болдық. Сондай-ақ, файлдарды шифрлау және дешифрлеу үшін шифрлаудың ең жақсы алгоритімін талдауға және таңдауға тұрарлық. Бұл мақалада біз жүйенің қауіпсіздігіне назар аударамыз. Ең жақсы шешім - шифрлаудың тиімділігін, деректерді беру жылдамдығын оңтайландыру үшін файлдарды компьютерде де, әр түрлі тасымалдағыштарда сақтауға мүмкіндік беретін файлдарды шифрлау және дешифрлеу жүйесін дамыту, ол осы мақаланың нәтижесі болып табылады.

Түйін сөздер*: бұлтты есептеу қауіпсіздігі, құпиялылық, деректердің қауіпсіздігі, қосарландыру, шифрлау.*

A. A. KULZHANOVA, D. R. RAKHIMOVA, T. I. BAKIBAYEV

Al-Farabi Kazakh National University, Almaty, Kazakhstan
Almaty Management University, Almaty, Kazakhstan

PROBLEMS OF THE BLOCKED DATA ENCRYPTION PROCESS AND ITS APPLICATION FOR INCREASING DATA SECURITY

This article examines the analysis of existing encryption methods and data mechanisms in order to avoid the possibility of the problem of partial data refresh in large files. For example, we need to identify identical blocks without the ability to decrypt the data. We are faced with the problem that if files are encrypted with different keys, then the encrypted versions will be different. It is also worth analyzing and choosing the best encryption algorithm for encrypting and decrypting files. In this article, we pay attention to the security of the system. The best solution is to develop a file encryption and decryption system that allows you to store files both on a local computer and on various media, which is the result of this article, in order to optimize the encryption efficiency, data transfer speed, and the speed of processing file changes necessary for storing files in space.

Keywords*: cloud computing security, privacy, data security, deduplication, encryption.*