REPORTS OF NATIONAL ACADEMY OF SCIENCES OF THE REPUBLIC OF KAZAKHSTAN
ISSN 2224-5227
Volume 5, Number 5 (2014), 141 – 153

UDC 004.021; 004.031.2; 004.056

THE INNOVATIVE WAY OF PROTECTING DATA BASES BY THE DISTRIBUTED DATA STORAGE METHOD

I. Syrgabekov, E. Zadauly, E. Kurmanbayev

Kazakh Humanities and Law Innovative University, Semey

Key words: distributed storage, data, security, cloud, data base, internal network.

Abstract. The common paradigm of IT-resources holds a chronic problem, it is the weak protection of data against the external invasion. The article presents the results of a study of a fundamentally new way of storing data using algorithms splitting / reconstruction, which are resistant to partial loss of storage data. It is an innovative approach in the sphere of the cloud distribution of the data. It is based on the concept of the distributed data storage, which is not in specialized stores, but is in the form of corporate cloud. The particularities of the developed algorithms in the study are a new paradigm in the sphere of computer security, it is the opportunity to realize the internally noncontradictory security updated model of the stored and processed data with a higher degree of protection from external invasion than in the public cloud systems.

УДК 004.021; 004.031.2; 004.056

ИННОВАЦИОННЫЙ СПОСОБ ЗАЩИТЫ ИНФОРМАЦИОННЫХ БАЗ ПО МЕТОДУ РАСПРЕДЕЛЕННОГО ХРАНЕНИЯ[1] [2]

И. Сыргабеков, Е. Задаулы, Е. Курманбаев

Казахский гуманитарно-юридический инновационный университет, г. Семей

Ключевые слова: распределенное хранение, информация, безопасность, облако, база данных, внутренняя сеть.

Аннотация. Общепринятая парадигма IT-ресурсов таит хроническую проблему – слабую защиту данных от внешнего вторжения. В статье приводятся результаты исследования принципиально нового способа хранения данных с применением алгоритмов расщепления/реконструкции, устойчивых к частичным потерям мест хранения. Это инновационный подход в сфере облачного распределения информации. В его основе лежит концепция распределенного хранения данных не в специализированных хранилищах, а в виде корпоративного облака. Особенности разработанного в исследовании алгоритма являют новую парадигму в сфере компьютерной безопасности – возможность реализовать внутренне не противоречивую актуальную модель безопасности хранимых и обрабатываемых данных с более высокой степенью защиты от внешнего вторжения, нежели в открытых облачных системах.

Защита данных – задача любой компании или индивидуального пользователя. С развитием компьютерных технологий и повышением общей емкости хранилищ проблема надежности хранения информации значительно обостряется. Например, безопасность стала актуальной для мобильных одноранговых сетей, в которых коллекция мобильных узлов с сетевыми интерфейсами может образовывать беспроводную временную сеть без фиксированной инфраструктуры. Схема управления и контроля доступа основана на дереве мобильных агентов управления правами на

[2] Статья подготовлена по гранту Республики Казахстан, номер госрегистрации 0112РК02909.

доступ к собственным ресурсам в посещаемых мобильных узлах. К настоящему времени изучены некоторые практические вопросы динамического управления ключами [1]. Современными исследованиями установлено также, что активные элементы, кодирующие информацию WM (Windows Mobile), нестабильны и разнообразны. Это подвергает сомнению каноническую модель WM и демонстрирует динамичность нейронного кода информации WM, а значит и преимущество распределенного и динамического хранения рабочей памяти [2].

Интеллектуальный уровень и быстродействие технологий повышаются, но аналогичный процесс происходит и с угрозами, которые несут технологии. Между тем все более интенсивно будут востребованы услуги открытых и частных облаков по мере их распространения. Кроме того, облако хранения является сетевой моделью онлайн-хранилища для поддержки асинхронных ресурсов различных платформ, таких как настольные компьютеры, смартфоны, ноутбуки. Одной из подобных систем является система C3ware, которая поддерживает работу группы на основе абстрактных облачных сервисов хранения и определяет совместные услуги, необходимые для групповых работ [3].

Поэтому на рынке появляются все новые и новые инструменты управления сложной облачной средой хранения данных. Это помогает достичь максимальных преимуществ от использования облака в IT. Но вот защитить базы данных в облачных системах становится труднее.

Состояние проблемы

Существующая структура организации IT-ресурсов опирается на исторически сложившую парадигму локальных вычислений, сфокусированных на локальных ресурсах [4]. Традиционный подход при этом подразумевает централизацию ресурсов в мощные специализированные кластеры для повышения качества IT-решений [5]. Организационно это выглядит в виде специализированных серверных площадок на предприятии, отдельно стоящих дата-центров [6] или в виде специализированных услуг сторонних организаций, обслуживающих потребности в IT-услугах. Однако локализация и централизация ресурсов имеет и «врожденные» недостатки, которые начинают влиять на весь дальнейший ход развития IT-индустрии.

Главный недостаток общепринятой платформы – низкий уровень надежности хранения информации. Причем это относится ко всем аспектам информационной безопасности: надежности хранения, недопущения утечки и устойчивости к искажению информации [7]. Очевидно, что если не принять вовремя мер, снижение информационной надежности может привести к катастрофическим последствиям. И связано это с общей тенденцией к переводу информационных ресурсов на компьютерные носители, с глубоким проникновением компьютерных технологий во все сферы деятельности человека. Более того, сегодня хранение большого объема информации осуществляется во внешних облачных системах – крупных сетевых хранилищах, доступ к которым осуществляется через Интернет. Частная информация в облачных системах защищена от вторжения, однако уже неоднократно внешние облачные системы подвергались несанкционированным взломам. Компании с повышенными требованиями к защите информации (правительственные, банковские, коммерческие структуры, госкомпании и т.д.) вообще предпочитают хранить информацию в собственных дорогостоящих (в том числе по обслуживанию) хранилищах, по сути, специализированных дата-центрах.

На фоне вышеизложенных тенденций, с повышением спроса на устройства хранения, отмечается общее снижение уровня надежности самих аппаратных платформ. Наблюдается: а) резкое увеличение плотности записи с одновременным снижением надежности. Единичный сбой жесткого диска может спровоцировать утерю катастрофических объемов информации; б) сближение параметров дорогих жестких дисков профессионального уровня и недорогих дисков для бытового массового применения; и как следствие, в) применение дешевых дисков и дисковых подсистем (массивов RAID, устойчивых к отказу одного диска) в серверах среднего и начального уровней.

Конечно, разработчики программ для облачных систем предпринимают все более оригинальные способы защиты. Прежде всего, совершенствуются системы защиты от взлома. Однако противостояние может длиться бесконечно, и это вынуждает вернуться к началу и искать новые способы хранения больших массивов. А такие способы есть. Причем не обязательно отказываться от облачных технологий. Достаточно изменить способ хранения информации.

Разработанная авторами настоящей статьи система распределенного хранения информации

предназначена именно для повышения безопасности информационных баз. В ней реализуется принцип распределенного хранения с применением алгоритмов расщепления/реконструкции, устойчивых к частичным утерям мест хранения. В основе описываемой системы лежит концепция распределенного хранения данных внутри местной локальной сети, выступающей в форме корпоративного облака. То есть исследование направлено на создание программ по организации хранения не в специализированных хранилищах, а на компьютерах внутренней сети компании. Это, по сути, то же облако, но – корпоративное облако с расширением хранилища на сетевые компьютеры с использованием патентованных программ.

Это абсолютно новый подход к проблеме безопасности информации, пока не имеющий аналогов в мире информационных технологий. Принципиальное отличие от существующих вариантов – именно в распределенном хранении в корпоративном облаке во внутренней компьютерной сети, допускающем отключение части компьютеров как мест хранения.

Новая система распределенного хранения апробирована авторами в Казахском гуманитарно-юридическом инновационном университете (КазГЮИУ), г. Семей. Система может быть рекомендована к установке в организациях, предъявляющих повышенные требования к безопасности хранения информации.

Методы решения задач

Современные потоки информации ориентированы на хранение больших массивов данных в основном в распределенных кластерных системах. Условно их можно разделить на два класса: распределенные файловые системы (Google File System, Hadoop Distributed System и др.) и распределенные хранилища структурированных данных (Google BigTable, HBase и др.). Эти системы имеют принципиальные отличия от традиционных файловых систем и реляционных баз данных. Например, распределенная файловая система Google File System является закрытой разработкой компании Google, используемой для хранения больших массивов данных. Внутри Google функционирует более 200 GFS-кластеров, крупнейшие из которых насчитывают более 5 тыс. машин, хранящих около 5 петабайт данных и обслуживающих порядка 10 тыс. клиентов [8]. Как и любая распределенная файловая система, GFS ориентирована на обеспечение высокой производительности, масштабируемости, надежности и доступности. Но проблемы безопасности остаются.

В отличие от столь сложных систем, разрабатываемая нами инновационная система распределенного хранения основана на расщеплении данных. Этот метод хранения имеет несколько аспектов, которые обусловили интерес к нему. Согласно первому аспекту, системы распределенного хранения с расщеплением данных позволяют содержать данные, исключая несанкционированный доступ к информации. Отдельные расщепленные данные сами по себе не несут осмысленной информации. Второй аспект связан с тем, что конфигурацию расщепления/восстановления можно составить таким образом, чтобы восстановление данных могло быть выполнено с применением только части расщепленных данных, то есть можно обеспечить устойчивость к потере данных.

Основным научным подходом в системах повышенной надежности и безопасности является применение сложных математических алгоритмов, реализующих процедуру разделения секрета [9], [10]. В криптографии под разделением секрета понимается любой метод распределения секрета среди группы участников, каждому из которых достается доля секрета. Воссоздать секрет может только коалиция участников. Для достижения устойчивости к утере информации применяют пороговую схему, когда количество долей, необходимых для восстановления секрета, меньше количества долей, на которые секрет был поделен [11].

Наиболее распространены алгоритмы на базе схем Ади Шамира [12], в частности известный алгоритм Рида-Соломона (применяется, например, в платформе Wuala [13]). Главный недостаток таких алгоритмов – в повышенной требовательности к вычислительным ресурсам, так как реализация алгоритма требует сложных математических расчетов и не поддается масштабированию [14].

Существуют альтернативные схемы разделения секрета, в основе которых лежат еще более сложные математические модели. Например, схема Блэкли [15] оперирует n-мерными гиперплоскостями, а схема Карнина-Грини-Хеллмана [16] использует математическую теорему о невозможности решения системы из n уравнений имея m неизвестных. По сумме недостатков на

практике системы, основанные на схемах разделения секрета, находят ограниченное применение, и, возможно, вышеизложенные проблемы реализации служат главным фактором, обуславливающим слабое развитие данных систем [17].

Особый интерес представляет метод доступа к данным вычислительного устройства в распределенной вычислительной системе с диспергированной сетью хранения, который обеспечивает повышение эффективности на уровне системы путем хранения метаданных и данных в едином комплексе частями в модуле DSTN (дисперсного хранения в сети) [18].

В то же время наибольшее распространение в алгоритмах коррекции нашли простые и быстродействующие алгоритмы четности. В частности, их применяют в распространенных системах отказоустойчивых дисковых массивов для серверов (RAID). С другой стороны, быстродействующие и простые алгоритмы четности не позволяют восстановить множественные утери данных. К примеру, в RAID-системах возможен отказ только одного диска из массива.

В разработанной нами системе распределенного хранения применяются патентованные алгоритмы четности с устойчивостью к множественным отказам. Для реализации системы используется ряд патентов Великобритании, Европейского союза и ноу-хау авторов статьи. Целевая платформа, на которой функционирует система, – MS Windows XP/Vista/7/8 с .NET Framework v.4/4.5.

Принцип действия системы

Сама модель распределенных вычислений не является новинкой в IT-индустрии. В бытность первых компьютеров эта модель считалась перспективным направлением. Но она не смогла выдержать конкуренции с бурным развитием технологий в области полупроводниковой техники и резким взлетом локальных мощностей компьютеров. Однако на новом витке развития компьютерных технологий, с ростом скоростей передачи данных, связанных с повсеместным внедрением широкополосных беспроводных и оптических каналов, появлением новых алгоритмов и идей в области распределенных вычислений, модель распределения ресурсов позволяет перейти на новый качественный уровень обработки информации.

Исследуемая система распределенного хранения данных позволяет хранить и обрабатывать клиентские данные на распределенных узлах. Данные разделяются на части при помощи алгоритма и распределяются по узлам системы.

Система состоит из двух функциональных компонентов – «Клиента» и распределенного массива в виде сети взаимодействующих друг с другом узлов. «Клиент» (далее без кавычек) – это компьютерное устройство в виде персонального компьютера, сервера или любого другого интеллектуального устройства, на котором установлено программное обеспечение «Клиент». Клиент служит шлюзом для входа в систему. Узел – это также любое компьютерное устройство с установленным программным обеспечением «Узел».

Клиент работает с системой, как с абстрактной облачной подсистемой. Взаимодействие Клиента с массивом узлов осуществляется по протоколу one-to-many (один ко многим). Любой акт записи информации в систему проходит предварительную обработку специальным алгоритмом, который: а) расщепляет информацию на нечитаемые составные части; б) добавляет динамически сгенерированные избыточные данные для повышения устойчивости к частичным утерям расщепленных частей; в) генерирует служебный метафайл, описывающий созданный массив. Акт записи набора сформированных данных реализуется путем распределения их по узлам системы. Любой акт чтения информации Клиентом из системы возможен только посредством обработки массива данных полученных от узлов алгоритмом, который может его восстановить, только зная его метаданные.

Узлы системы, составляющие распределенный массив, знают только ограниченное количество своих соседей. Никакой узел не может знать всю систему и составляющие его узлы. Узлы могут динамически подключаться и отключаться в системе, что не сказывается на работоспособности всей системы. Узлы могут обмениваться данными и автоматически обновлять пропавшие части хранимой информации по командам Клиента.

Клиент может авторизоваться на любом узле системы для работы со всей системой. Все информационное взаимодействие между компонентами системы осуществляется посредством каналов в виде виртуальных туннелей. Система распределенного хранения данных позволяет хранить и обрабатывать клиентские данные на распределенных узлах. Данные разделяются на

части при помощи алгоритма и распределяются по узлам системы.

Система способна выдержать массовые отключения и повреждения узлов, вплоть до 50% и более, в зависимости от размера системы и конфигурационных параметров алгоритма. На отдельных узлах хранится разделенная информация, не несущая никакого функционального смысла.

Система не требует сертификации на предмет использования крипто-алгоритмов, так как не использует шифрования для защиты хранимых данных. Прочитать данные можно, только восстановив их из «размазанных» в системе частей на Клиенте. Восстановить данные может только владелец (создатель) или тот, кому были делегированы права. Делегирование прав не означает передачи прав на владение. Владелец всегда имеет полный контроль над любыми изменениями в его файлах.

Таким образом, особенностями системы являются:

1) Анонимность. Клиентские данные разделяются на блоки и записываются на разные узлы системы, что гарантирует их анонимность и позволяет без использования методов шифрования защитить данные клиента от несанкционированного доступа. С разделенными подобным образом данными, хранящимися на разных узлах системы, может работать только их хозяин.

2) Устойчивость к потере данных. Алгоритм разделения позволяет восстанавливать данные при 50-80% потери узлов системы.

3) Минимальные требования к узлам системы. Все основные по работе с данными операции ложатся на клиента, что дает возможность снизить требования к узлам, хранящим данные.

4) Расширенный функционал. Система не только хранит данные, но и позволяет подключать следующие модули: почтовый сервис; IP телефонию; единый доступ к одному ресурсу для нескольких пользователей.

Архитектура системы

Архитектура приложения – это сервер-клиент. Система Distributed Cloud System разработана на Visual Studio 2010 на языке программирования C# с применением технологии .Net.

Система состоит из узлов, соединенных между собой доверительным каналом связи, и клиентов, работающих с этими узлами (рис. 1). Каждый узел знает только соседние узлы и не имеет представления обо всей системе в целом. Узел хранит данные клиентов в файле данных (Data file). Обработка поступающей информации происходит в порядке очереди.

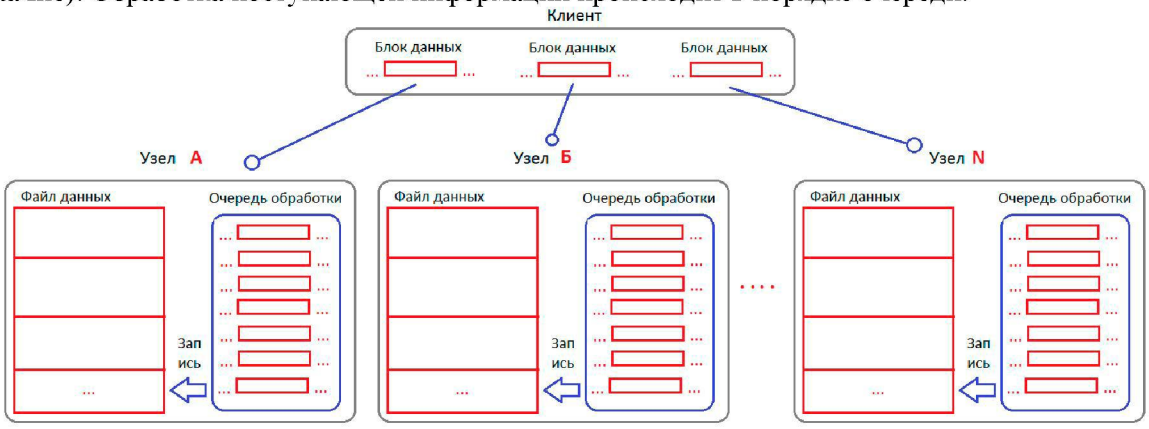

Рисунок 1 – Схема архитектуры системы распределенного хранения информации

Клиент получает доступ к узлам системы только при регистрации внутри системы. Клиент получает уникальный ключ, который используется при распределении данных. Клиент не знает обо всех узлах в системе.

Клиентское приложение ответственно за разбивку файла на блоки данных и распределение блоков данных между узлами. Для хранения файловой системы и распределенных блоков клиентское приложение использует метафайл (meta file или mf) (рис. 2). Благодаря древовидной системе метафайл описывает файловую структуру и расположение блоков данных.

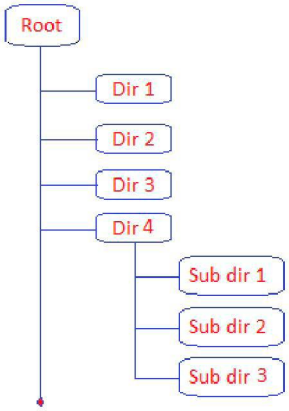

Рисунок 2 – Схема структуры метафайла

Сам метафайл при окончании работы разделяется на блоки и так же распределяется между узлами. Только уникальный ключ клиента CID в связке с данными авторизации (учетная запись, пароль) могут собрать метафайл.

Пользователь устанавливает клиентскую часть приложения и авторизуется в системе Distributed Cloud System, получая при этом уникальный ключ CID, который служит для формирования метафайла. Выбрав нужный ему файл, клиент запускает функцию «Положить в облако». При этом данный файл блокируется и в фоновом режиме разделяется на блоки и распределяется по узлам системы.

При необходимости получения своего файла клиент запускает функцию «Получить из облака» и ему из облака поступают блоки данных его файла и при помощи алгоритма компонуются в файл. Клиент также может дать доступ на свою директорию другим пользователям.

Описание приложения

Серверная часть приложения служит для хранения и обработки поступающей от клиента информации. Хранение осуществляется в файлах данных, которые разбиты на кластеры. Кластер в свою очередь разбит на страницы, которые ранжированы по 2-кратной системе. Размер кластера конфигурируется в настройках системы, сам кластер содержит кратное 2 количество страниц. Например, если кластер состоит из 32 мегабайт, то он содержит 256 страниц размером по 128 килобайт. По мере поступления новой информации система создает новые кластеры для хранения данных с различной размерностью страниц.

Все поступающие данные записываются в очередь, которая следит за всеми процессами записи данных. После успешного завершения получения клиентского блока данных система записывает этот блок в файл данных и отправляет клиенту уведомление об успешном получении блока данных в виде UID.

Узлы связаны между собой доверительным каналом связи, по которому они обмениваются данными. Например, в случае заведения нового пользователя узел системы оповещает соседние узлы о создании нового пользователя.

Серверная часть приложения выполняет следующие функции: а) авторизация клиента (входящие параметры: данные по авторизации; выходящие параметры: SSKey, Root_ID); б) сохранение файла клиента (входящие параметры: блок данных без UID; выходящие параметры: UID); в) передача блока данных клиента (входящие параметры: UID; выходящие параметры: блок данных); г) эхо запрос (входящие параметры: эхо запрос; выходящие параметры: IP адрес); д) опрос соседей (входящие параметры: запрос соседей; выходящие параметры: Node ID); е) Обмен данными между узлами.

При регистрации нового пользователя, при удалении пользователя или при модификации данных о пользователе узлы должны обмениваться этой информацией между собой.

Клиентская часть – это программное обеспечение, установленное на компьютере клиента, позволяющее клиенту регистрироваться в системе Distributed Cloud System и осуществляющее менеджмент блоков данных, файлов. Клиентская часть должна выполнять следующие функции: а)

разделение файла (входящие параметры: файл данных; выходящие параметры: блоки данных); б) сборка файла (входящие параметры: блок данных; выходящие параметры: файл или сообщение об ошибке); в) отправка данных (входящие параметры: блок данных; выходящие параметры: UID); г) получение данных (входящие параметры: UID; выходящие параметры: блок данных); д) авторизация (входящие параметры: UID; выходящие параметры: блок данных).

Элементы взаимодействия между клиентской и серверной частями: а) авторизация (входящие параметры: UID; выходящие параметры: блок данных); б) получение данных (входящие параметры: UID; выходящие параметры: блок данных); в) отправка данных (входящие параметры: блок данных; выходящие параметры: UID).

Программирование системы

Основные результаты программирования системы распределенного хранения с расщеплением данных: 1) спецификации на необходимые модули; 2) результаты проектирования и отработки карты взаимодействия компонентов; 3) протоколы взаимодействия компонентов; 4) программы модулей системы; 5) программы составных частей системы и стыковки модулей; 6) программы серверной компоненты; 7) программы узлов хранения; 8) программы дизайна пользовательского интерфейса; 9) программная система в стадии пре-альфа; 10) результаты тестирования и доработки функционала; 11) результаты отработки взаимодействия с аппаратной составляющей и рекомендации по использованию; 12) альфа-версия в окончательной сборке; 13) система с сервисами, установленная в КазГЮИУ; 14) результаты предварительного тестирования системы.

Система распределенного хранения информации на базе технологии расщепления данных использует модульную архитектуру. Поэтому программа в общем обозрении представляет собой набор компонентов (модулей) и набор встроенных программных интерфейсов к компонентам, что позволяет создать гибкую архитектуру программы и при этом обеспечить простоту разработки. На наш взгляд, создание именно модульной системы позволило придать необходимые качества корпоративному облаку. Ведь независимо от будущих изменений к требованиям гибкость, которую предоставляют отдельные и независимые компоненты системы, позволит быстро и без задержки в работе изменять/добавлять/удалять «на ходу» новый функционал и, соответственно, создавать новые возможности. В принципе работу любого модуля можно представить в виде схемы, изображенной на рис. 3.

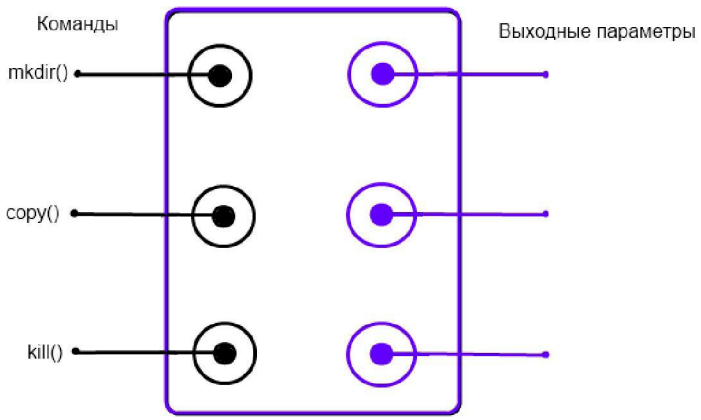

Рисунок 3 – Схема работы модуля

Во всех разработанных модулях в качестве входных параметров (слева на схеме) подаются различные команды: а) от других модулей; б) от пользователей, например нажатием кнопки на экране; в) сетевые команды и т.д. Все команды описаны программистами в интерфейсе модуля. А в качестве выходных параметров (справа на схеме) – потоки данных или, в ряде других случаев, команды на запуск других модулей. При помощи программных интерфейсов модули взаимодействуют с внешним миром и друг с другом. Модули могут быть вложенными друг в друга.

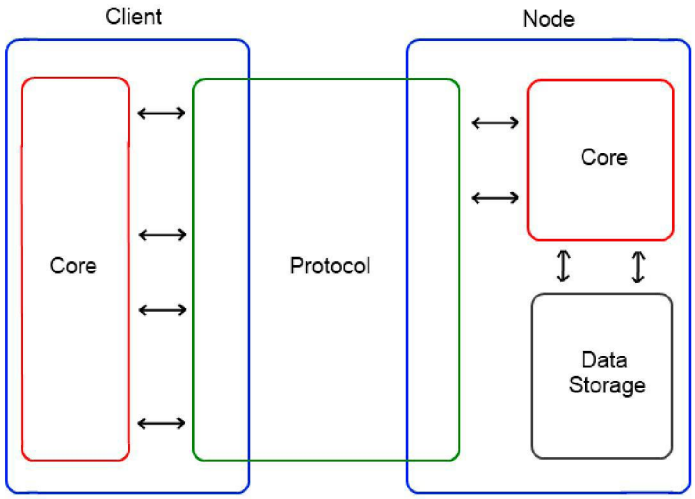

Рисунок 4 – Схема взаимодействия модулей

Схематично общая работа всех компонентов представлена на рис. 4. Здесь Core – это модуль, реализующий алгоритм расщепления данных на клиентском приложении и восстановления данных после расщепления на узловом приложении. Node – модуль, реализующий алгоритм хранения и распределение данных на узлах системы. Client – модуль, реализующий взаимодействие с системой распределенного хранения; предоставляет возможность конечному пользователю работать в удобном и привычном для него интерфейсе Windows. Предусматривается несколько режимов работы: постоянно работающий в фоновом режиме сервис (демон, в терминологии Linux), интерактивная программа. Protocol – модуль, реализующий функциональность взаимодействия Клиент-Узел. Data Storage – модуль, реализующий хранение данных на узлах облака. Как видно из схемы, архитектура системы распределенного хранения информации на базе технологии расщепления данных проста и надежна.

Взаимодействие между узлом (Node) и клиентом, осуществляется посредством программного модуля Protocol (рис. 5).

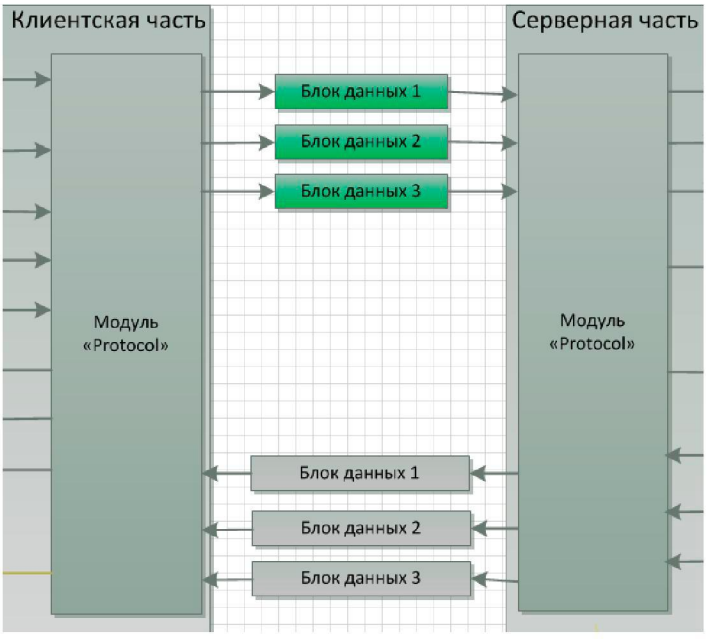

Рисунок 5 – Схема работы модуля Protocol.

Программный интерфейс модуля Protocol дает возможность другим модулям подключаться к себе и, используя набор стандартных типизированных команд (API), обмениваться информацией как между собой, так и с «внешним миром». Для системы распределенного хранения данных этот модуль является ключевым, он присутствует во всех компонентах системы.

Одним из важных компонентов модуля Node является внутренний модуль хранения данных Fst. Он состоит из следующих компонентов: а) модуль записи блоков данных; б) модуль чтения блоков данных; в) модуль управления потоками данных; г) модуль хранения данных.

Модуль Node Net – сетевой модуль, обеспечивает безопасную транзакционную модель передачи данных в системе наиболее надежного хранения данных.

Модуль Manager отвечает за компоновку блоков данных, за генерацию уникальных ключей и распределение блоков по местам хранения. Модуль использует уникальные свойства алгоритма для повышения надежности хранения – способность образовывать устойчивые кластеры и кластеры кластеров. Модуль Node Net использует принципы распределенной сети Kademlia для равномерного распределения блоков данных в сети. Алгоритм реализован в виде модуля Core.

Модуль Admin – модуль администратора, служит в системе ключевым звеном для управления пользователями. Также модуль «Admin» позволяет вручную распределять потоки данных в сети для уменьшения нагрузки на наиболее активные места хранения.

Модуль Storage Service. У этого модуля одна важная задача – «физическое» хранение блоков данных на узлах сети.

Компоновка модулей выполнена таким образом, что для более устойчивой и безопасной работы системы распределенного хранения информации на базе технологии расщепления данных все модули системы стыкуются между собой при помощи программного интерфейса API (Application Program Interface), то есть у каждого модуля есть свой набор входных и выходных параметров. Выходными параметрами, как правило, являются наборы данных – результат работы модуля.

Программным пакетом обеспечивается взаимодействие модулей. На рис. 6 схематично представлена взаимосвязь всех имеющихся модулей системы распределенного хранения информации на базе технологии расщепления данных.

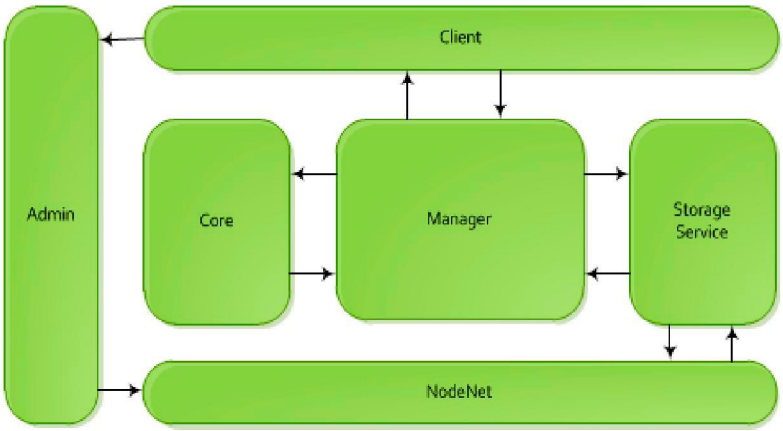

Рисунок 6 – Схема взаимодействия модулей

Как видно из рисунка, некоторые модули изолированы, так как имеют очень узкую специализацию.

Система требует для работы пользователей наличие графического интерфейса, который, по сути, является «оберткой» для большинства модулей системы. Основная задача графического интерфейса – максимально упростить работу пользователя и повысить защиту от необдуманных действий со стороны пользователей. Сам графический интерфейс GUI (Graphical user interface), присутствует в модуле Client и представляет собой привычную среду наподобие проводника. Связка GUI плюс системный сервис, позволяет быстро отображать важную информацию пользователю, что избавляет последнего от необходимости вручную запускать большое

количество модулей.

Система распределенного хранения информации инсталлируется на компьютеры с операционной системой семейства Windows (XP/7/8/8.1). Кроме того, обязательным требованием эксплуатации системы распределенного хранения информации является наличие программной платформы .NET Framework, последняя версия которой 4.5 выпущена в 2013 году компанией Microsoft.

Тестирование системы

Система распределенного хранения информации на базе технологии расщепления данных была протестирована в КазГЮИУ (г. Семей) на 16 единицах вычислительной техники. Сначала система была апробирована при помощи нагрузочного тестирования, и показала стабильный результат при работе с данными до 2 Гб. Далее было выполнено апробирование облачной системы распределенного хранения информации во внутренней сети университета. В формате теста замерялось соотношение скорости обработки данных (которая является плавающей величиной и полностью зависит от параметров вычислительной техники) с объемом обрабатываемых данных. Тестировались объемы данных в 1 Мб, 100 Мб, 300 Мб, 600 Мб. В качестве примера на рис. 7 показаны результаты обработки данных объемом 600 Мб.

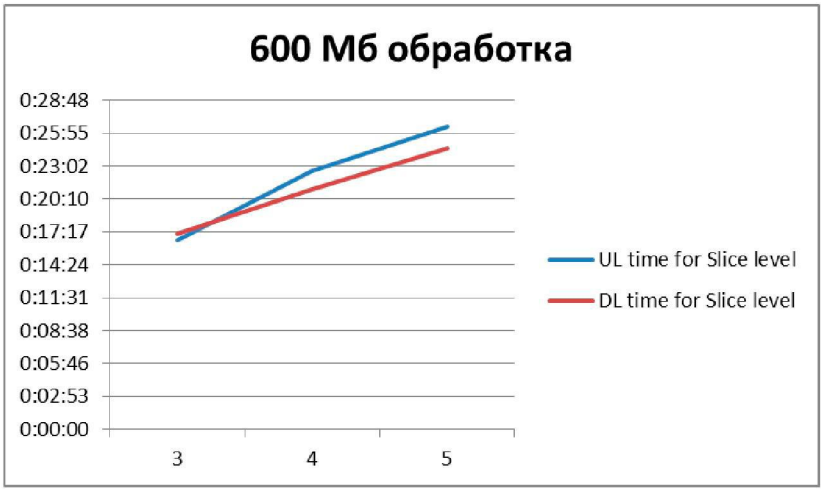

Рисунок 7 – Обработка данных размером в 600 Мб

Анализ результатов исследования показал, что тесты выявляют стабильную работу системы при реальных нагрузках до 2 Гб. Параметры стабилизации можно оценить по характерных графикам обработки данных, например, для объема 1,5 Гб (рис. 8) (UL dirty означает скорость загрузки данных в облако, UL raw – среднюю скорость загрузки данных в облако, DL dirty – скорость скачивания данных из облака, DL raw – среднюю скорость скачивания данных из облака).

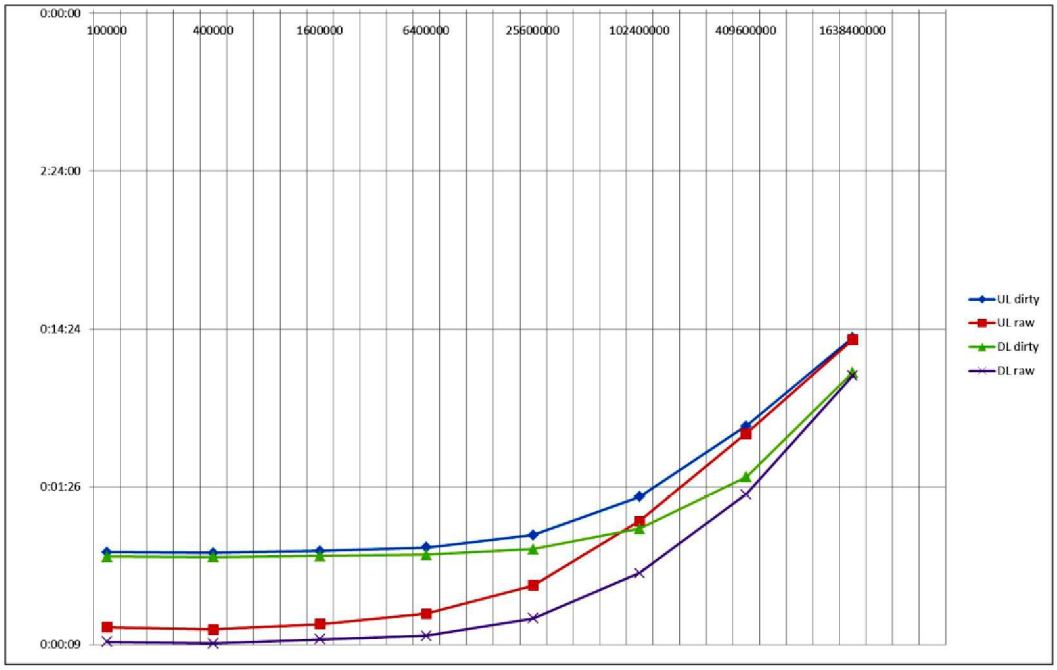

Рисунок 8 – Обработка данных размером до 1,5 Гб

Как видно из рис. 8, обработка данных объемом 1,5 Гб занимает около 14 минут. Пороговые величины скачивания и загрузки данных выравниваются, и принимают примерно одинаковое значение. Аналогичные параметры наблюдаются и при измерении скорости обработки данных в облаке.

Проведенные тестовые исследования позволяют с уверенностью заключить, что система распределенного хранения информации на базе технологии расщепления данных успешно стабильно работает во внутренней сети учреждения. Дальнейшее тестирование и расширение системы надежного хранения информации на базе технологии распределенного хранения данных должно определить направление развития прототипа.

Сравнительный аспект

Существует всего несколько примеров реализаций идеи распределенного хранения – это платформа Clever Safe (распределенное хранение на серверах фирмы с использованием аппаратно-программного комплекса на базе iSCSI и собственного патентованного алгоритма) [19], недавно появившийся Symform (технология Symform Cooperative Storage Cloud, использующая клиентские компьютеры как часть системы, а также использующая RAID-96 – собственный вариант RAID) [20] и Wuala (похожая на Symform технология с использованием алгоритма Соломона-Рида) [13] Все эти системы предлагают услуги распределенного хранения на своих серверах и применяют различные алгоритмы расщепления/восстановления, предполагая загрузку через Интернет данных на серверы фирм. Для защиты от несанкционированного доступа к информации применяются методы шифрования. Сами услуги либо платные, с установкой специализированных аппаратных устройств (Clever Safe), либо востребуют встречные услуги в виде предоставления свободного места на диске своего компьютера, которое система использует для своих целей (диск как составная часть системы распределенного хранения), и/или обеспечения постоянного доступа к компьютеру через Интернет извне.

Примером стандартной системы обеспечения надежности и безопасности хранения, например, может служить портал aws.amazon.com: сервис облачного хранения S3 (Simple Storage Service) и EBS (Elastic Block Store) [21], где используются технологии кластеризации и репликации.

Исследуемая нами система, в отличие от существующих систем распределенного хранения информации, впервые предлагает систему распределенного хранения как часть корпоративного облака. Имеющиеся аналоги распределенного хранения (Clever Safe, Wuala, Symform)

предполагают загрузку данных на серверы фирм через Интернет. Разрабатываемая же нами система хранения данных на базе технологии распределенного хранения предполагает хранение данных непосредственно в компьютерах внутренней сети.

Характеристика результатов исследования

Описанные параметры системы, вытекающие из особенностей алгоритма обработки информации, реализуют новую парадигму в области компьютерной безопасности. В общих чертах ее можно охарактеризовать как модель, в которой разделены близкие, но не тождественные характеристики хранимой информации. Речь идет о том, что под данными мы обычно подразумеваем две разные сущности: а) физические данные – биты, файлы, жесткие диски с файлами; б) саму информацию (семантическое наполнение) – смысл текста документа Word, лицо человека на цифровой фотографии, характеристики устройства, отображенные на чертеже и т.д. То есть понятие информационной безопасности на самом деле содержит в себе два совершенно разнородных контекста. В предлагаемой системе распределенного хранения информации эти два контекста существуют раздельно и поддаются раздельному регулированию.

Из описанных выше свойств системы вытекают уникальные особенности обеспечения безопасности, заключающиеся в реализации двух контекстов. Во-первых, за безопасность данных отвечают системный администратор, инженер компьютерной техники, соответствующий квалифицированный технический персонал. Однако, имея полный доступ к системе, они, тем не менее, не имеют доступа к данным, которые находятся не в их компетенции (к данным в смысле информационного наполнения). Во-вторых, за информационную безопасность отвечает тот, кто имеет соответствующую компетенцию именно в этой области. В частности, создатель/владелец документа, естественно, должен в первую очередь распоряжаться единолично параметрами безопасности своего же документа, не задумываясь о физической сохранности информации, находящейся в компетенции технической службы.

Высокая достоверность результатов исследования, а значит и степени безопасности, подтверждается высокими адаптивными, функциональными и динамическими качествами, проявленными системой надежного хранения информации на базе технологии распределенного хранения данных при внедрении ее во внутреннюю компьютерную сеть и последующем тестировании. В условиях повышенного риска утери информации роль систем и технологий усиления надежности хранения информации значительно повышается. И поскольку степень защиты от внешнего вторжения в исследованной корпоративной системе значительно выше, нежели в открытых облачных системах, то существенна и потенциальна значимость этой системы.

ЛИТЕРАТУРА

[1] Hsu Chien-Lung, Lin Yu-Li. Improved migration for mobile computing in distributed networks // Computer Standards & Interfaces. 2014. V. 36. Issue 3. P. 577-584.

[2] Sreenivasan K., Vytlacil J., D'Esposito M. Distributed and Dynamic Storage of Working Memory Stimulus Information in Extrastriate Cortex // Journal of Cognitive Neuroscience. 2014. V. 26. №5. P. 1141-1153.

[3] Lee H.C., Park J.E., Lee M.J. A Middleware Supporting Collaborative Services over Cloud Storage // Computer Journal. 2014. V. 57. Issue 2. P. 217-224.

[4] Новиков Ю.В., Кондратенко С.В. Основы локальных сетей. М.: Интернет-университет информационных технологий, 2005. 360 с.

[5] Dean J. Handling Large Datasets at Google: Current Systems and Future Directions. Data-Intensive Computing Symposium, 2008. – [Электронный ресурс] / Режим доступа: http://research.yahoo.com/files/6DeanGoogle.pdf.

[6] Харатишвили Д. Дата-центры в цифрах и фактах. – [Электронный ресурс] / Режим доступа: http://www.sapr.ru/article.aspx?id=20687&iid=942.

[7] Партыка Т.Л., Попов И.И. Информационная безопасность. М.: Форум-Инфра-М, 2004. 432 с.

[8] Мишечкин А. Кластер кластеру рознь // Windows IT Pro. 2008. №5. – [Электронный ресурс] / Режим доступа: http://www.osp.ru/win2000/2008/05/5529265.

[9] Шнайер Б. Алгоритмы разделения секрета / Сб: Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.: Триумф, 2002. С. 588–591.

[10] Шнайер Б. Разделение секрета / Сб: Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.: Триумф, 2002. С. 93–96.

[11] Mignotte M. How to Share a Secret // Lecture Notes in Computer Science. 1983. V. 149. P. 371–375.

[12] Adi Shamir. How to share a secret // Communications of the ACM. New York, ACM. 1979. B. 11. V. 22. P. 612–613.

[13] Comparison of encryption schemes. – [Электронный ресурс] / Режим доступа: http://www.wuala.com/en/learn/technology.

[14] Asmuth C., Bloom J. A modular approach to key safeguarding // Information Theory, IEEE Transactions on. 1983. B. 2. V. 29.

[15] Blakley G.R. Safeguarding cryptographic keys / Proceedings of the 1979 AFIPS National Computer Conference. NJ: AFIPS Press, 1979. P. 313–317.
[16] Karnin E., Greene J., Hellman M. On secret sharing systems // Information Theory, IEEE Transactions on. 1983. B. 1. V. 29. P. 35–41.
[17] Simmons C.J. An introduction to shared secret and/or shared control schemes and their application / Contemporary Cryptology. IEEE Press, 1991. P. 441–497.
[18] Resch J.K., Leggette W. Method for accessing e.g. data of computing device in dispersed storage network, involves generating access specific key based on content specific information, and executing access request regarding data object utilizing access specific key. Patent Number: US2014068259-A1.
[19] How Cleversafe Works. – [Электронный ресурс] / Режим доступа: http://www.cleversafe.com/overview/how-cleversafe-works.
[20] The Smartest Cloud Security. – [Электронный ресурс] / Режим доступа: http://www.symform.com/how-it-works/security.
[21] Security Resources. – [Электронный ресурс] / Режим доступа: http://aws.amazon.com/security/security-resources.

REFERENCES

[1] Hsu Chien-Lung, Lin Yu-Li. *Computer Standards & Interfaces*, **2014**, 36, *3*, 577-584.
[2] Sreenivasan K., Vytlacil J., D'Esposito M. *Journal of Cognitive Neuroscience*, **2014**, 26, *5*, 1141-1153.
[3] Lee H.C., Park J.E., Lee M.J. *Computer Journal*, **2014**, 57, *2*, 217-224.
[4] Novikov Ju.V., Kondratenko S.V. *Osnovy lokal'nyh setej*. M.: Internet-universitet informacionnyh tehnologij, **2005**, 360 (in Russ.).
[5] Dean J. http://research.yahoo.com/files/6DeanGoogle.pdf.
[6] Haratishvili D. http://www.sapr.ru/article.aspx?id=20687&iid=942 (in Russ.).
[7] Partyka T.L., Popov I.I. *Informaciomnaja bezopasnost'*. M.: Forum-Infra-M, **2004,** 432 (in Russ.).
[8] Mishechkin A. http://www.osp.ru/win2000/2008/05/5529265 (in Russ.).
[9] Shnajer B. *Prikladnaja kriptografija*. M.: Triumf, **2002**, 588–591 (in Russ.).
[10] Shnajer B. *Prikladnaja kriptografija*. M.: Triumf, **2002**, 93–96 (in Russ.).
[11] Mignotte M. *Lecture Notes in Computer Science*, **1983**, *149*, 371–375.
[12] Adi Shamir. *Communications of the ACM*. New York, ACM, **1979**, 11, *22*, 612–613.
[13] *Comparison of encryption schemes*. http://www.wuala.com/en/learn/technology.
[14] Asmuth C., Bloom J. *Information Theory*, **1983**, *2*, 29.
[15] Blakley G.R. *Safeguarding cryptographic keys*. NJ: AFIPS Press, **1979**, 313–317.
[16] Karnin E., Greene J., Hellman M. *Information Theory*, **1983**, 1, *29*, 35–41.
[17] Simmons C.J. *Contemporary Cryptology*. IEEE Press, **1991**, 441–497.
[18] Resch J.K., Leggette W. *Patent Number:* US2014068259-A1.
[19] *How Cleversafe Works*. http://www.cleversafe.com/overview/how-cleversafe-works.
[20] *The Smartest Cloud Security*. http://www.symform.com/how-it-works/security.
[21] *Security Resources*. http://aws.amazon.com/security/security-resources.

БӨЛІП САҚТАУ ӘДІСІ БОЙЫНША АҚПАРАТТЫҚ БАЗАЛАРДЫ ҚОРҒАУДЫҢ ИННОВАЦИЯЛЫҚ ӘДІСІ

И. Сырғабеков, Е. Задаұлы, Е. Құрманбаев

Қазақ инновациялық гуманитарлық-заң университеті, Семей қ.

Тірек сөздер: бөліп сақтау, ақпарат, қауіпсіздік, бұлт, деректер базасы, ішкі желі.

Аңдатпа. IT-ресурстарының жалпылай қабылданған парадигмасының созылмалы мәселесі бар – деректердің сырттан бұзып кіруден әлсіз қорғалуы. Мақалада сақтау орындарының ішінара жоғалуына тұрақты бөлшектеу/қайта құру алгоритмдерін қолдана отырып, деректерді сақтаудың түбелейлі жаңа әдісін зерттеу нәтижелері келтіріледі Бұл ақпаратты бұлттарға бөлу саласындағы инновациялық ұстаным. Оның негізінде деректерді мамандандырылған қоймаларда емес, корпоративтік бұлт түрінде бөліп сақтау концепциясы жатыр.

Зерттеу барысында әзірленген алгоритмнің ерекшеліктері компьютерлік қауіпсіздік саласындағы жаңа парадигманы - ашық бұлт жүйелеріне қарағанда сыртқы бұзып кіруден барынша жоғары қорғаныс деңгейі бар сақталатын және өңделетін деректердің ішкі жағынан қайшы келмейтін өзекті қауіпсіздік үлгісін жүзеге асыру мүмкіндігін алып келеді.

Информация об авторах

Сыргабеков Искендер Наримanович, генеральный директор АО «Рауан Медиа Групп», председатель совета директоров компании SKY Technologies.
E-mail: syrkonst@mail.ru.
Задаулы Еркин, Web-редактор журнала «Центр Азии», генеральный директор компании SKY Technologies.
E-mail: erkin.zadauly@gmail.com.
Курманбаев Ербол Асылханович, директор научно-инновационного центра Казахского гуманитарно-юридического инновационного университета, г. Семей.
Адрес: Республика Казахстан, г. Семей, ул. Абая, 94.