КОМПЬЮТЕРНАЯ МАТЕМАТИКА

УДК 049.1

Н.Н. ГЛИБОВЕЦ, И.Е. МЕЛЬНИК, М.О. СИДОРЕНКО

СОЗДАНИЕ ДИНАМИЧЕСКОЙ СИСТЕМЫ РАСПРОСТРАНЕНИЯ КОНТЕНТА С ИСПОЛЬЗОВАНИЕМ ПРОТОКОЛА BITTORRENT

Рассмотрены возможности использования протокола одноранговых сетей BitTorrent при создании динамической сети распространения контента и алгоритм шифрования информации в таких сетях на всех уровнях взаимодействия. Предложено архитектуру системы и разработан ее прототип.

Введение. Для надежного распространения контента в Интернете используют географически распределенную сетевую инфраструктуру, состоящую из серверов с контентом, позволяющую оптимизировать доступ контента конечным пользователям. Такие сети носят название Сети Доставки Контента (CDN или Content Delivery Network) и имеют клиент серверную архитектуру.

В последнее время начали развиваться одноранговые сети [1]. Данные сети не имели развитой инфраструктуры серверов, а взамен использовали децентрализованную обработку данных непосредственно на компьютерах пользователей, и позволяли распространять информацию с использованием так называемых P2P (peer-to-peer) протоколов. Основное отличие такого подхода состоит в отсутствии необходимости установки дорогостоящего оборудования для каждого из узлов. Каждый пользователь сети является в определенном смысле сервером, распространяющим информацию, находящуюся на его компьютере.

Однако в одноранговых сетях тяжело реализовывать поддержку авторских прав распространения информации. Поэтому появились гибридные сети, в которых одноранговая сеть используется для передачи данных, а для хранения, индексирования и обработки информации в доступных файлах применяет-

ся трекер-сервер, распространяющий метаданные доступной в сети информации.

По состоянию на сентябрь 2011 года [2] общий объем данных, передающийся в Европе используя P2P протоколы, составлял значительную часть от общего трафика, иногда превышая 50 %.

Подобные сети распространены широко, но многих не устраивают при организации массовых трансляций, потоковой передаче больших файлов, видео, музыки из-за значительной стоимости и проблем масштабируемости. Клиент получает данные с серверов, находящихся к нему ближе территориально. В настоящее время наиболее распространенным протоколом в одноранговых сетях является открытый протокол обмена информацией BitTorrent.

Основная цель данной работы – разработка архитектуры и прототипа системы распространения контента, которая позволила бы использовать преимущества одноранговых сетей гибридного типа и при этом обеспечила бы достаточно высокий уровень защиты распространяемой информации. Система должна иметь и возможность интеграции с существующими сетями распространения контента. На начало 2012 года нам неизвестен сервис, обладающий указанными свойствами, реализованный на базе одноранговых сетей.

Наиболее близко находится Bittorent DNA, реализующий распространение файлов используя дополнительное программное обеспечение, которое пользователь должен предварительно получить для каждого загружаемого файла. Главный недостаток системы – невозможность обеспечить полноценную защиту распространяемой информации (файлы, распространяющиеся с обычных серверов должны быть публично доступны) [3].

Опишем решение указанной проблемы в виде разработанной нами системы динамического распространения контента с использованием протокола BitTorrent и вспомогательной защиты информации.

1. Протокол BitTorrent фактически признан стандартом большинства современных одноранговых файлообменных систем. Например, Twitter и Facebook используют его для распространения обновлений между серверами [4].

Типичный процесс загрузки пользователем информации через сеть BitTorrent состоит из следующих шагов [5]. Сначала устанавливается клиентское программное обеспечение, реализующее необходимые функции для работы в данной сети. Далее ищутся страницы, содержащие файл метаданных (.Torrent) загрузки файла метаданных и его открытия программой-клиентом. Используя алгоритмы поиска других пользователей клиентское программное обеспечение устанавливает P2P соединения и загружает файл(ы). Одновременно ведется распространение информации другим пользователям сетей.

Одна из ключевых особенностей протокола – эффективная и комфортная работа с файлами большого объема [6].

В случае использования децентрализованных хеш-таблиц, каждый из клиентов фактически становится трекер-сервером, распространяя информацию о других клиентах в ответ на DHT-запросы [7, 8]. Кроме того, даже после прекращения передачи клиенту данных некоторое время продолжают поступать DHT-запросы. Применяя механизм обмена пирами (PEX), каждый из участников обменивается информацией о наличии других узлов сети, которые также участвуют в распространении каждого конкретного файла.

Согласно стандарту BEP-3 версии 11031 от 28.02. 2008 [5] протокол использует стандарт кодирования метаданных «Bencoding».

Для более эффективного использования системы требуется также реализовать ее совместимость с классическими системами распространения контента. Протоколом BitTorrent предусмотрено распространение информации с использованием так называемых Web-сидов. В этом случае реальными пользователи системы заменяются серверами, работающими по протоколам http или ftp (описаны в стандартах BEP-17 и BEP-19 [9, 10]).

2. Шифрование данных. С точки зрения разграничения доступа, распространяемый контент следует разделить на три типа: предназначенный для общего доступа с акцентом на защиту в момент распространения; предоставлением пользователю полного доступа к дешифрованным файлам; с необходимостью защиты и после получения информации клиетом.

Шифрование необходимо в первую очередь для контента второго и третьего типов. Функционально алгоритмы могут быть очень похожими, однако третий тип требует также реализации дополнительных уровней защиты программного обеспечения клиента.

Учитывая возможность появления дополнительных требований к защите информации, которые не были рассмотрены в данной работе, может возникнуть необходимость во внесении изменений в эти алгоритмы. Поэтому, шифрования и дешифрования данных в нашей системе выделены в отдельный компонент системы, который со временем может быть заменен на более совершенный. Этот компонент реализован в виде специальной надстройки над указанным протоколом. Важным является и предоставление возможности доступа к файлу сразу после загрузки.

Учитывая важность фактора аппаратной реализации [10] и криптографической стойкости [11] мы пришли к решению об использовании метода шифрования AES Rijndael (AES), с возможностью использования различных режимов кодирования. В разработанном прототипе системы, компонент шифрования/дешифрования реализовано упрощенную версию указанного подхода.

Алгоритм шифрования. Для сохранения совместимости с протоколом BitTorrent, использования основных его преимуществ и одновременно надежной защиты информации разработан алгоритм шифрования, оперирующий с теми же базовыми понятиями, что и протокол BitTorrent.

Полная реализация предлагаемого алгоритма кодирования предполагает использование AES-контейнеров фиксированного размера, являющихся аналогом фрагментов файла в том значении, которое используется при работе с протоколом BitTorrent. Это отличие от типичных алгоритмов кодирования обусловлено в первую очередь требованием увеличения скорости работы с большими файлами, которые в подавляющем большинстве передаются в P2P сетях. Используем дополнительный слой между программным обеспечением, работающим с полученным файлом и закодированными данными. Слой должен предоставлять доступ к произвольному месту закодированного файла аналогично доступу к не закодированному файлу. Для повышения криптостойкости закодированного файла каждый из контейнеров кодируется отдельным ключом. Поэтому, количество операций, которые необходимо выполнить для подбора общего ключа в файл возрастает в сотни раз.

К недостатку этого механизма шифрования можно отнести некоторые ограничения изменения данных после их загрузки на сервер. Над загруженными данными можно выполнять только такие операции: изменение наполнения файла; удаление файлов; добавление новых файлов, с указанием их в конце файла метаданных.

Следует отметить, что в результате данных операций обновленный файл метаданных будет выглядеть для сервера и участников передачи данных новым файлом, никак не связанным с его предыдущей версией, поскольку файл метаданных будет иметь новый хеш.

Этот метод формирования контейнеров может оказаться неэффективным при передаче большого количества файлов небольшого размера. Поэтому желательно применить конкатенацию небольших файлов перед их разбивкой на контейнеры.

Процесс кодирования. На первом этапе делаются предварительные расчеты параметров: оптимальный размер контейнера, их количество.

Учитывая, что емкость каждого контейнера меньше, чем его фактический размер, объем конечного файла возрастет. Выбор размера контейнера чрезвычайно важен.

Если есть необходимость в применении цифровой подписи контента, она должна применяться непосредственно к кодированным данным.

На втором этапе выполняется формирование контейнеров и их заполнение данными. При подготовке данных к передаче, в отличие от типичной работы с протоколом BitTorrent, файлы предварительно размещаются в контейнерах, после чего происходит их распространение. Отдельно обрабатываются файлы небольшого размера (по сравнению с размером контейнера), сливающихся перед размещением в контейнеры.

Третьим этапом является генерация файла, который будет передаваться, а также файла метаданных и формирование ключа декодирования.

Файл, который будет распространяться с использованием протокола BitTorrent, является конкатенацией вышеупомянутых заполненных контейнеров.

Файл метаданных создается как и в основном протоколе, однако должен учитывать, что фактический размер каждого фрагмента является меньшим.

В нашем алгоритме, ключи хеш-таблицы info файла метаданных имеют такое значение: piece length – размер контейнера (не данных, которые в нем закодированы); pieces – содержит конкатенованные хеши контейнеров (не данных, которые в них закодированы); dencoded – ?маркер, показывающий, описывает ли данный файл метаданных файл или несколько файлов, которые зашифрованы вышеуказанным принципом. Допустимые значения 0 и 1.

Ключ декодирования является слиянием ключей декодирования каждого из контейнеров-фрагментов закодированного файла.

На последнем этапе полученные файлы загружаются на соответствующие серверы. Зашифрованный файл может распространяться на серверы с использованием протокола BitTorrent, а также других протоколов на выбор пользователя.

Авторизация пользователя, дешифрования данных. Регистрируясь, пользователь создает два ключа: один из них всегда доступен на сервере, другой – в клиенте.

Возможны ситуации, когда у одного пользователя установлено несколько клиентских программ на разных компьютерах или других устройствах. В этом случае используется сертификат, установленный в данном клиентском ПО.

Процесс загрузки контента начинается с загрузки файла метаданных или запроса к DHT сети. Получив нужную информацию, клиент выполняет загрузку файла.

Если клиент имеет право просмотра файла, а файл закодирован, сервер формирует ключ декодирования, который вместе с персональным ключом клиента образует ключ для декодирования файла.

Клиент декодирует загруженный файл с помощью собственного ключа и полученного ключа декодирования. Для открытия загруженного файла может быть дешифрован каждый фрагмент отдельно, а может и весь файл.

3. Архитектура системы. При разработке архитектуры системы использован компонентно-ориентированный подход. Это позволяет разбивать функционально сложные компоненты на более простые и использовать для их реализации готовые решения, что приведет к существенному упрощению модификации системы.

На рис. 1 показана упрощенная семантическая сеть, которая в обобщенном виде демонстрирует особенности работы системы с точки зрения кодирования/декодирования информации. Перечень операций, для которых использовано сокращенное обозначение: *S* – создание и хранение; *T* – передача данных; *C* – подключение; *A* – авторизация; *F* – формирования.

Для обеспечения доступа к системе пользователей с различными ролями предусмотрено два уровня прав доступа использования клиентского программного обеспечения: владелец контента и конечный пользователь.

Описание полного сценария работы системы с точки зрения кодирования/декодирования информации выглядит так.

1. Владелец контента с помощью встроенных функций программы готовит контент к распространению. Процесс подготовки данных состоит из нескольких этапов:

а) применение цифровой подписи владельца, используя его персональный сертификат;

б) генерация уникального ключа, который будет использоваться исключительно для данного контента;

в) шифрования подписанного контента с помощью созданного ключа с использованием алгоритма, описанного в разделе 2.3.1 данного документа;

г) создания файла метаданных для данного контента.

2. Подготовленные данные (зашифрованный файл и файл метаданных) загружаются на соответствующие серверы:

а) загрузка метаданных и ключа декодирования на сервер метаданных;

б) создание копий на первичных серверах распространения информации протоколом BitTorrent, используя данный протокол;

в) в случае использования владельцем контента классических серверов (вне данной системой) – распространение контента на данные серверы.

3. С момента появления файла метаданных на сервере информация становится доступной для загрузки пользователями, желающие ее распространять, просматривать и использовать.

4. После загрузки пользователь получает закодированный файл. Для его декодирования программное обеспечение пользователя инициирует использование следующих действий:

а) запрос персонального ключа декодирования на сервере, используя собственный ключ пользователя;

б) в ответ сервер формирует ключ декодирования исключительно для этого запроса;

в) получив ответ с сервера и используя собственный ключ на стороне пользователя получается полный ключ декодирования загруженных ранее файлов.

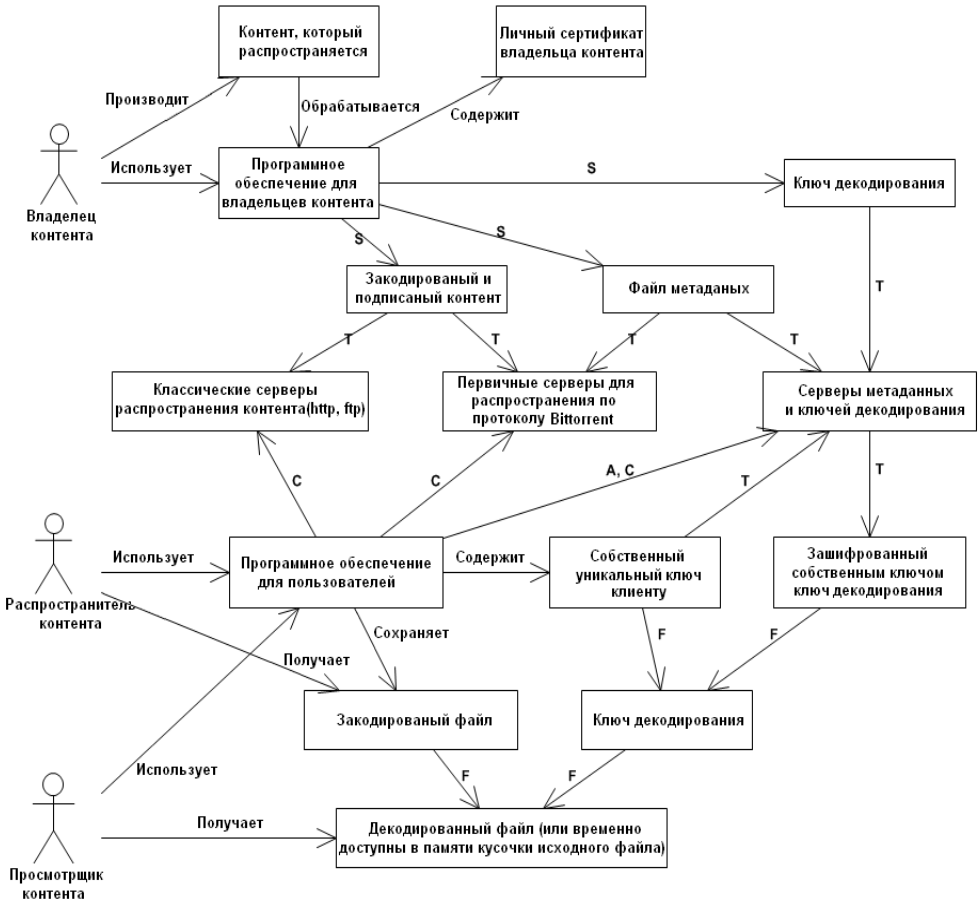

РИС. 1. Обобщенная архитектура системы с точки зрения кодирования/декодирования информации

Основные интерфейсы, использующие клиентское программное обеспечение, с одной стороны, предоставляют сервер, а с другой – показаны на рис. 2.

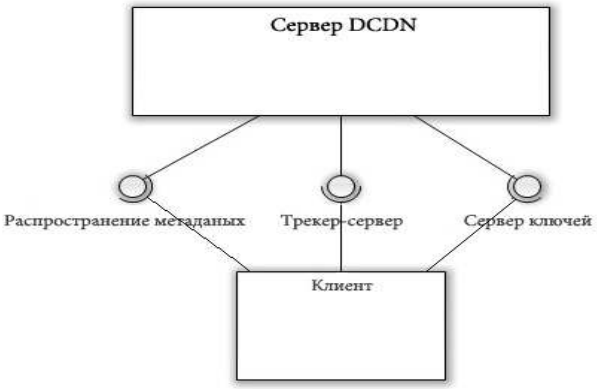

РИС. 2. Обобщенная компонентная структура и интерфейсы

Серверная часть динамической системы распространения контента реализует типичные функции торрент-трекера, дополнена специфическими функциями, обеспечивающими деятельность других компонентов. Кроме предоставления служебной информации о распространяемых файлах, система также обслуживает запросы по поставкам ключей дешифровки кодированных файлов. Для реализации данных функций сервера целесообразно использовать отдельный компонент, не зависящий от выбора программного обеспечения для трекер-сервера.

Для обеспечения выполнения указанных функций, при использовании готовых компонент для трекера и/или каталога доступных файлов, необходимо реализовать дополнительно функциональные возможности, специфичные для нужд системы. Поскольку функциональные возможности системы компонентно расширяющиеся, единственная модификация, которую требует программное обеспечение трекер-сервера – это интерфейс для проверки данных пользователей, необходимый для работы сервера ключей декодирования. Для обеспечения большей совместимости с готовым программным обеспечением, его реализовано, как отдельный компонент, использующий базу данных трекер-сервера.

Клиентская часть. Как видно из рис. 3, клиентское программное обеспечение использует такие интерфейсы: входные метаданные, трекер-сервер, сервер ключей доступа, файловая система пользователя.

Обратная совместимость с протоколом одноранговой сети BitTorent позволяет использовать в качестве транспортного компонента программное обеспечение, совместимое с данным протоколом. Большинство современных клиентов данной одноранговой сети также поддерживают возможность расширения с помощью встроенного стандартизированного API.

В качестве демонстрационного решения при работе прототипа использовано программное решение uTorrent, реализующее наиболее эффективные алгоритмы выбора других участников сети и установки соединений с ними.

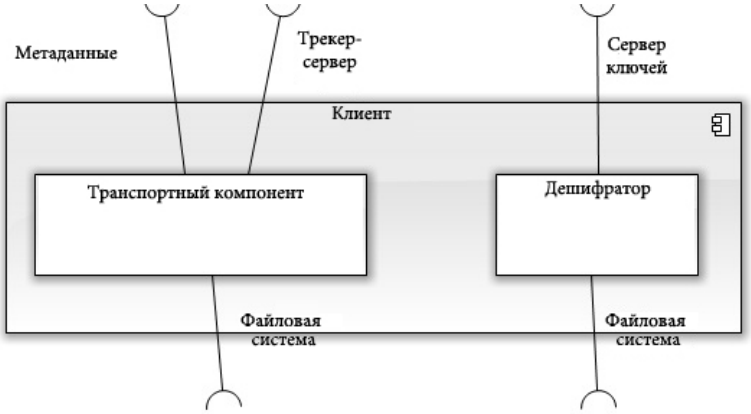

РИС. 3. Основные компоненты клиентского программного обеспечения

Поскольку в рамках данной работы описывается только прототип динамической системы распространения контента, мы реализовали только базовые функции защиты информации, которые можно будет заменить на более совершенные при дальнейшей разработке.

Реализация защиты контента вышеуказанных типов отличается дополнительными надстройками в виде программного обеспечения с повышенным уровнем защиты контента, который нуждается в защите и базируются на алгоритме из раздела 2.

Использование распространенных алгоритмов шифрования также позволяет использовать готовые решения при разработке системы шифрования. Таким образом, компонент дешифровки можно разбить на более мелкие компоненты.

Основное требование к подкомпонентам шифрования – поддержка работы с AES контейнерами в нужном режиме.

В работе использовано программное обеспечение с открытым кодом TrueCrypt, реализующее все необходимые возможности системы дешифровки.

Заключение. Рассмотрены особенности использования протокола P2P передачи данных BitTorrent в качестве основы для разработки сетей распространения контента гибридного типа и создан прототип динамической системы распространения контента.

Предложена система шифрования, в которой ключ передается отдельно от данных. Благодаря этому обеспечивается возможность доступа к информации в любые моменты времени, а защита используется для разграничения прав доступа.

Выявлены и описаны преимущества использования P2P сетей по сравнению с классической архитектурой распространения информации. Проведен анализ и описаны проблемы классических реализаций файлообменных сетей по указанному протоколу.

В рамках работы разработан и реализован алгоритм шифрования данных, адаптированный к использованию в сети, работающей по протоколу BitTorrent. Алгоритм базируется на создании AES (Rijndael) контейнеров с уникальными ключами, которые выступают как фрагменты файла и распространяется по протоколу BitTorrent. Использование такого метода позволяет утверждать о надежности шифрования передаваемых файлов. Кроме того, адаптация под передачу по протоколу BitTorrent позволяет использовать большинство его преимуществ в полной мере. Также в работе описано расширение протокола, обеспечивающее корректную работу с зашифрованными таким алгоритмом данными.

М.М. Глибовець, І.Е. Мельник, М.О. Сидоренко

СТВОРЕННЯ ДИНАМІЧНОЇ СИСТЕМИ ПОШИРЕННЯ КОНТЕНТУ
З ВИКОРИСТАННЯМ ПРОТОКОЛУ BITTORRENT

Розглянуто можливості використання протоколу однорангових мереж BitTorrent при створенні динамічної мережі поширення контенту та методи захисту інформації у таких мережах на всіх рівнях взаємодії. Запропоновано архітектуру системи та розроблено її прототип.

M.M. Glybovets, I.E. Melnyk, M.O. Sydorenko

CREATING A DYNAMIC CONTENT DISTRIBUTION SYSTEM USING
THE BITTORRENT

In the article it was considered the possibilities of using the protocol BitTorrent peer networks to create a dynamic content distribution network and encryption algorithms in these networks at all levels of interaction.Also was proposed system architecture and developed it's prototype.

1. *Глибовец Н.Н., Жигмановський А.А.* Распределенное управление знаниями на основе архитектуры Peer-to-Peer // Сборник научных трудов НАУ Инженерия программного обеспечения. – 2012. – № 1 (9). – С. 59 – 65.
2. *Internet* Observatory [Електронный ресурс] / Режим доступа http://www.internetobservatory.net/
3. *BitTorrentDNA* (Downloading with DNA Downloader) – 2012.
4. *Ernesto*. TorrentFreak (Facebook Uses BitTorrent, and They Love It) [Electronic resource] / Mode for access: http://torrentfreak.com/facebook-uses-bittorrent-and-they-love-it-100625/
5. *Cohen B.* BitTorrent.org (The BitTorrent Protocol Specification - BEP 3) / Bram Cohen <bram at bittorrent.com>.[Electronic resource] / Mode for access: http://www.bittorrent.org/beps/bep_0003.html
6. *Wikimedia* Commons (Animated gif that ilustrates the bitorrent protocol for sharing files) [Electronic resource] / Mode for access: http://en.wikipedia.org/wiki/File:Torrentcomp_small.gif
7. *Loewenstern A.* BitTorrent.org (DHT Protocol BEP 5) / Andrew Loewenstern <drue at bittorrent.com> [Electronic resource] / Mode for access: http://www.bittorrent.org/beps/bep_0005.html

8. *Hazel G., Norberg A.* BitTorrent.org (Extension for Peers to Send Metadata Files BEP 9) / Greg Hazel <greg at bittorrent.com>, Arvid Norberg <arvid at bittorrent.com> [Electronic resource] / Mode for access: http://bittorrent.org/beps/bep_0009.html
9. *Burford M.* BitTorrent.org (WebSeed – HTTP / FTP Seeding (GetRight style) BEP 19) / Michael Burford < michael at getright.com > [Electronic resource] / Mode for access: http://www.bittorrent.org/beps/bep_0019.html
10. *Лаборатория* Чеканова (Intel Core i5 (Clarkdale): анализ аппаратного ускорения шифрования AES (рус.). «Наиболее популярный стандарт симметричного Шифрования в мире ИТ») [Electronic resource] // THG. – 2010. – Mode for access: http://www.thg.ru/cpu/aes_clarkdale/index.html. iXBT.com
11. *Гусенко С.* Алгоритмы шифрования – финалисты конкурса AES. Часть 2. [Электронный ресурс] / Режим доступа: http://www.ixbt.com/soft/alg-encryption-aes-2.shtml

Получено 15.10.2012

Об авторах:

Глибовец Николай Николаевич,
доктор физико-математических наук, профессор, декан факультета информатики Национального университета «Киево-Могилянская академия»,
e-mail: glib@ukma.kiev.ua

Мельник Игорь Егорович,
магистр факультета информатики Национального университета «Киево-Могилянская академия»,
e-mail: melnik@ukma.kiev.ua

Сидоренко Марина Олеговна,
ассистент кафедры информатики факультета информатики Национального университета «Киево-Могилянская академия».
e-mail: tinuriel@gmail.com