# ИССЛЕДОВАНИЕ МЕХАНИЗМОВ ОБЕСПЕЧЕНИЯ ЗАЩИЩЕННОГО ДОСТУПА К ДАННЫМ, РАЗМЕЩЕННЫМ В ОБЛАЧНОЙ ИНФРАСТРУКТУРЕ

**Сахаров Дмитрий Владимирович,**
к.т.н., доцент кафедры защищенных систем связи Санкт-Петербургского государственного университета телекоммуникаций имени проф. М.А. Бонч-Бруевича, г. Санкт-Петербург, Россия, d.sakharov@rkn.gov.ru

**Левин Марк Вадимович,**
аспирант кафедры защищенных систем связи Санкт-Петербургского государственного университета телекоммуникаций имени проф. М.А. Бонч-Бруевича, г. Санкт-Петербург, Россия, m.va.levin@gmail.com

**Фостач Елена Сергеевна,**
студент Санкт-Петербургского государственного университета телекоммуникаций имени проф. М.А. Бонч-Бруевича, г. Санкт-Петербург, Россия, elena.fostach@gmail.com

**Виткова Лидия Андреевна,**
аспирант кафедры защищенных систем связи Санкт-Петербургского государственного университета телекоммуникаций имени проф. М.А. Бонч-Бруевича, г. Санкт-Петербург, Россия, iskinlidia@gmail.com

## АННОТАЦИЯ

С развитием облачных технологий возрастает роль распределенной инфраструктуры, функциональная совместимость и портативность которой являются неотъемлемой составляющей. Однако, обеспечение доступности услуг
и масштабируемости виртуальных ресурсов, безопасности и конфиденциальности пользовательских данных имеет первостепенное значение.
В рамках актуальной статьи авторами было проведено исследование, позволяющее более детально разобраться в вопросах безопасности, с которыми приходится сталкиваться при проектировании архитектуры облачных сред.
Первым этапом исследования было выбрано изучение мировых стандартизирующих документов в исследуемой области, на которые опираются результаты данной работы.
В основу проведенного исследования были положены принципы организации доступа к облачному пространству, критерии к шифрованию информации, передаваемой как между клиентами облачных услуг, так и хранимой на удаленном сервере. Особое внимание при проведении исследования было уделено вопросам аутентификации.
Показано, что для создания надежной, с точки зрения безопасности, облачной архитектуры необходимо использовать криптостойкие протоколы смешенного шифрования с проверкой подлинности сообщений, а так же внедрять механизмы аутентификации, которые позволят идентифицировать каждого пользователя, пытающегося получить доступ к конфиденциальным данным. Исследование содержит актуальные для поставщиков облачных услуг решения, которые позволят поддержать доступность, конфиденциальность и целостность личных данных в облачной среде.

**Ключевые слова:** защита персональных данных; облачная архитектура; безопасность облачных вычислений; конфиденциальность информации; механизмы аутентификации; угрозы информационной безопасности.

**Для цитирования:** *Сахаров Д.В., Левин М.В., Фостач Е.С., Виткова Л.А.* Исследование механизмов обеспечения защищенного доступа к данным, размещенным в облачной инфраструктуре // Наукоемкие технологии в космических исследованиях Земли. 2017. Т. 9. № 2. С. 40–46.

**Введение**

С развитием облачных технологий, происходит смена парадигм информационной безопасности от идеи локальной защиты ресурсов к облачной модели защиты приложений, данных и сервисов. В связи с этим, для создания безопасных виртуальных услуг, прежде всего, необходимо обеспечить меры защиты со стороны поставщика облачной инфраструктуры.

Мировые стандарты, такие как Security Recommendations for Cloud Computing Providers (BSI), Cloud Computing Information Assurance Framework (ENISA), The Cloud Security Alliance Consensus Assessments Initiative (Cloud Security Alliance) и Security Assessment Provider Requirements and Customer Responsibilities (NIST), диктуют свои требования к построению облачных решений.

Облачные вычисления представляют собой технологию распределенной обработки данных, где ресурсы и мощности предоставляются пользователю в качестве услуг. Технология облачных вычислений является результатом конвергенции более ранних технологий, таких как параллельные вычисления и распределенные вычисления.

Согласно документу с рекомендациями Национального Института Стандартизации (The NIST Definition of Cloud Computing) предоставленных Питером Меллом и Тимоти Грансем, облаком называется услуга или форма предоставления услуг, обладающая пятью характеристиками:

1. **Самообслуживание** (on demand self-service) — возможность подключения и отключения облачных услуг самим пользователем за счет предоставляемых ему механизмов.

2. **Доступность по сети** (broad network access) — получение доступа к облачным ресурсам через сеть Интернет независимо от месторасположения потребителя услуг.

3. **Наличие пула ресурсов** (resources pooling) — обеспечение избыточного объема ресурсов с целью возможного предоставления неограниченного количества услуг.

4. **Эластичность и масштабируемость** (scalability and elasticity) — возможность контроля количества и скорости потребления услуг.

5. **Измеримость** (measurable service) — контроль потребления услуг в пределах фиксированного временного отрезка [1].

Провайдеры облачных вычислений предлагают свои услуги на базе трех основных моделей сервисов:

- инфраструктура как услуга (IaaS);
- платформа как сервис (PaaS);
- программное обеспечение как услуга (SaaS) [1].

IaaS представляет собой фундаментальную часть, где вычислительная инфраструктура (серверы, хранилища данных, сетевые ресурсы, операционные системы) предоставляются в качестве подключаемой услуги. Данный подход позволяет потребителю облачных услуг уменьшить совокупную стоимость владения инфраструктурой, т.е. IaaS превращает стоимость капитальных расходов [CAPex] в операционные расходы [OPex]. Так же необходимо отметить еще одно важное преимущество данного сервиса — высокая скорость масштабирования, т.е. увеличение или уменьшение количества используемых инфраструктурных услуг, что позволяет потребителю оптимально задействовать ресурсы.

PaaS предоставляет платформу, включающую в себя средства разработки продуктов и среду исполнения программного кода, размещенную на предоставленной поставщиком услуг инфраструктуре. Данная услуга ориентирована преимущественно на отдельный стек технологий, среди которых можно отметить разнообразие языков программирования и вариативность подключаемых библиотек.

SaaS, в свою очередь, представляет собой набор приложений, которые предоставляются пользователю. Поставщик SaaS услуг осуществляет техническую поддержку приложений, отслеживает и производит их обновление.

Таким образом, исходя из рассмотренных видов предоставляемых провайдером облачных услуг и предъявляемых к ним требований со стороны стандартизирующих организаций, можно сделать вывод, о том, что независимо от того, какой сервис предоставляется потребителю (SaaS, PaaS, IaaS), необходимо обеспечить меры защиты, с точки зрения безопасности, на каждом уровне предоставляемых услуг.

**Решение ключевых проблем при построении облачной инфораструктуры**

Несмотря на то, что виртуализация сетевых функций и облачные вычисления дают возможность дистанционно разграничить ИТ-инфраструктуру и пользователей, необходимо решить возросшие вместе с этим риски эксплуатации уязвимостей информационной безопасности для того, чтобы в полной мере воспользоваться новыми возможностями вычислительной парадигмы.

Особое значение приобретает данная проблема для поставщиков SaaS услуг. Пользователь, который доверил свои данные для хранения в облаке, теряет контроль над их целостностью, конфиденциальностью и доступностью. Таким образом, одной из важных задач поставщика услуг является обеспечение трех базовых свойств информационной безопасности, включая задачи организации места хранения и способа представления пользовательских данных.

В Российской Федерации в соответствии с действующим ФЗ «Об информации, информационных технологиях и защите информации», «**конфиденциальность информации**» определяется как обязательное для выполнения лицом, получившим доступ к определенной информации, требование не передавать такую информацию третьим лицам без согласия её обладателя» [2]. **Целостность информации** гарантирует обеспечение одинакового поддержания данных во время любой проводимой над ними операции, например, хранения, передачи, извлечения. **Доступность информации** гарантирует беспрепятственный доступ к защищаемой информации для законных пользователей.

Таким образом, для поддержания трех базовых свойств информационной безопасности — конфиденциальности, целостности и доступности данных пользователей, обозначим ключевые проблемы, которые необходимо решить в первую очередь:

1. Исследование механизмов аутентификации пользователей и использования защищенного канала связи на пути между клиентом и сервером.

2. Исследование способов хранения информации в зашифрованном виде, ее обработки и поиска в облачном хранилище.

Решение данных проблем позволит повысить уровень конфиденциальности, целостности и доступности данных в облачных средах.

Актуальное исследование посвящено изучению первой проблемы — организации доступа к данным пользователей и создания канала безопасной передачи данных. В то время как проблема способа хранения, обработки и поиска запрашиваемой пользователем информации в инфраструктуре облака будет рассмотрена нами в следующей работе.

За основу была взята схема функциональной архитектуры облачной среды (рис. 1), на базе которой построено данное исследование. На схеме показан способ развертывания баз данных и приложений на ресурсах облачной инфраструктуры вместе с сетевой схемой взаимодействия. Наглядно показаны уровни коммутации (L2) и маршрутизации (L3) данных между объектами облачной инфраструктуры.

В соответствии с решаемой проблемой, обозначим ключевые аспекты информационной безопасности, которые должны лежать в основе каждого надежного облачного сервиса:

1. Определение способов **конфиденциальной** передачи данных.

2. Организация доступа **авторизованных** пользователей к данным.

**Определение способов конфиденциальной передачи данных**

Для решения поставленной задачи необходимо использовать криптографические механизмы, позволяющие обеспечить надежное шифрование данных. Как известно, существует два типа алгоритмов шифрования — симметричные и ассиметричные. Симметричное шифрование дает большое преимущество в скорости шифрования и снижении нагрузки на вычислительные ресурсы, однако уступает в надежности ассиметричному шифрованию в силу особенностей аппаратной реализации специфических математических преобразований.

Очевидно, использование смешанного шифрования даст существенное преимущество. В качестве реализации рассмотрим следующие алгоритмы шифрования AES[3], RSA[4]:

1. Стандарт AES (*Advanced Encryption Standard*), является симметричным алгоритмом блочного шифрования. Алгоритм основан на нескольких заменах, подстановках и линейных преобразованиях, каждое из которых выполняется блоками по 16 байт. Операции повторяются несколько раз, каждый из которых называется «раунд». В течение каждого раунда, на основе ключа шифрования вычисляется уникальный ключ раунда и встраивается в вычисления. Благодаря подобной блоковой структуре AES, изменение даже одного бита или в ключе, или в текстовом блоке приводит к полному изменению всего шифра — явное преимущество относительно традиционных потоковых шифров. Благодаря описанным преимуществам, шифр AES является криптостойким по результатам проведенного исследования Агентством национальной безопасности США.

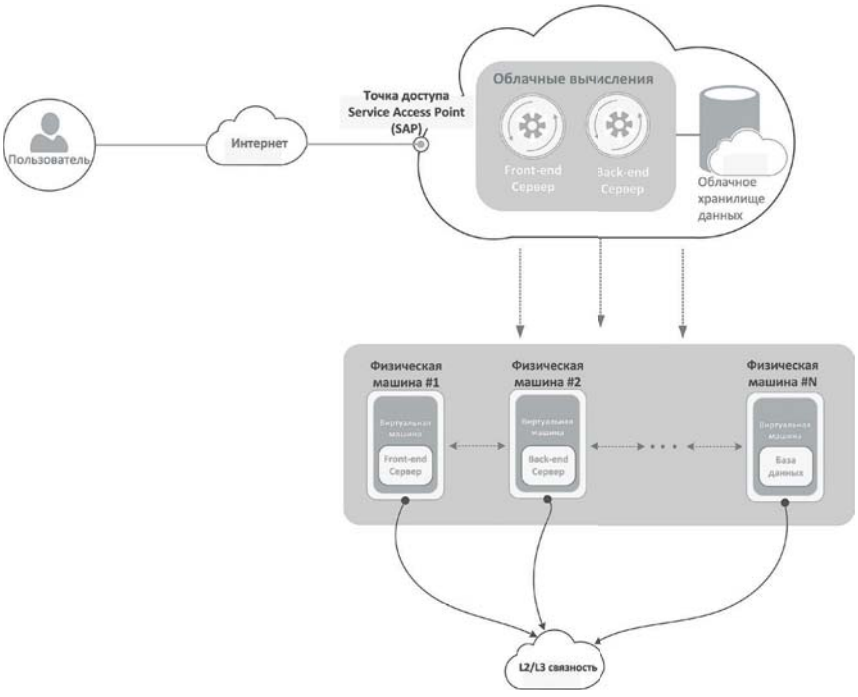


**Рис. 1.** Функциональная архитектура облачных сред

2. RSA — один из наиболее успешных асимметричных алгоритмов шифрования на сегодняшний день. В противоположность традиционным симметричным системам шифрования, RSA работает с двумя различными ключами: «открытым» и «закрытым» ключом. Оба работают совместно друг с другом, и сообщение, зашифрованное одним из них, может быть расшифровано только вторым. Так как закрытый ключ не может быть вычислен из открытого ключа, последний может храниться в открытом доступе. Безопасность RSA основана на математической проблеме факторизации целых чисел. Шифруемое сообщение рассматривается как одно большое число. Во время шифрования оно возводится в степень ключа и делится с остатком на произведение первых двух. Повторяя процесс с другим ключом, можно получить исходный текст. Лучший из известных методов взлома заключается в факторизации множителя, использованного при делении. На сегодняшний день невозможно произвести подобную факторизацию для чисел длиннее 768 бит. Поэтому современные системы шифрования используют минимальную длину ключа в 3072 бита.

Важно отметить, что с целью снижения вероятности перехвата в открытом виде передаваемого сообщения, шифрование данных должно происходить до того момента, как информация покинет браузер пользователя (т.е. до момента отправки сообщения на сервер).

Рассмотрим протокол защищенной передачи данных TLS v1.2, в котором реализованы алгоритмы шифрования информации на базе уже рассмотренных ранее алгоритмов AES, RSA, аутентификации пользователей и контроля целостности получаемых данных.

Работа TLS протокола начинается с согласования версии используемого протокола, способа шифрования данных между узлами соединения, а так же проверки достоверности полученных сертификатов, после чего будет установлен криптографически безопасный канал. Отметим, что шифрование с открытым ключом должно использоваться только в процедуре во время первоначальной настройки соединения (*TLS Handshake*), которая позволяет установить общий секретный ключ шифрования без предварительных знаний узлов соединения друг о друге. После настройки *TLS*-туннеля должна использоваться симметричная криптография, общение в пределах текущей сессии будет зашифровано именно установленными симметричными ключами. Это необходимо для увеличения быстродействия, так как криптография с открытым ключом требует значительно больше вычислительной мощности.

После того, как мы определили протокол, который обеспечит соединение на участке между клиентом и облаком, необходимо перейти к вопросу аутентификации.

Отметим одну из ключевых особенностей протокола TLS v1.2, которая заключается в возможности установления подлинности личности, клиента и сервера (*Chain of Trust*) за счет использования сертификатов подлинности, предоставляемыми центрами сертификации (*CA – certificate authorities*). Центры сертификации выдают подписанные сертификаты, доверие к которому неоспоримо. Таким образом, целый ряд выданных сертификатов образует цепочку доверия. Благодаря этому можно проверить подлинность каждого доверительного узла. Центры сертификации осуществляют проверку, выявляя тем самым, был ли скомпрометирован закрытый ключ сертификата, или была ли скомпрометирована вся процедура сертификации.

Передача каждого сообщения осуществляется с добавлением MAC-значения (*Message Authentication Code*), который представляет собой одностороннюю криптографическую функцию хэширования, ключи которой известны обоим участникам соединения. При отправке сообщения каждый раз генерируется его MAC-значение, по которому принимающая сторона может проверить полученную информацию на предмет подмены.

Таким образом, показано, что использование протокола TLS v1.2 позволяет создать канал конфиденциальной передачи данных. Однако, отметим, что механизмы работы данного протокола не обеспечивают контроль времени жизни каждой пользовательской сессии и повторную аутентификацию клиента для возобновления сессии в случае разрыва установленного соединения. Так же отметим, что протокол TLS v1.2 не позволяет аутентифицировать самого пользователя, в связи с этим, рассмотрим механизм аутентификации пользователей в рамках протокола OAuth2.0.

**Организация доступа авторизованных пользователей к ресурсам**

Глобальное развитие облачных сервисов приводит к тому, что каждый пользователь сети Интернет окружен в независимости от используемой платформы доступа, огромным количеством служб, позволяющих создавать и распространять медиа-контент или получать мгновенный доступ к электронным услугам.

Очевидно, что перед разработчиками сервисов возникает задача обеспечения безопасности. Необходимо решать задачи защиты данных от несанкционированного доступа пользователей, работающих в большом количестве приложений. Ситуация осложняется тем, что работа пользователя не должна затрудняться внутренними механизмами безопасности и перемещение между сервисами должно происходить максимально быстро и безопасно для услуг, предоставляемых пользователю.

Чтобы решить задачу, связанную с упрощением авторизации пользователя при работе с большим количеством приложений и онлайн сервисов был разработан протокол OAuth.

При использовании OAuth-авторизации к основным преимуществам принято относить отсутствие передачи логина и пароля в приложение, с которым работает пользователь. Таким образом, приложение может выполнить только то, что явно разрешил пользователь. Так же, отпадает необходимость решения вопроса обеспечения защищенного хранения пароля и логина приложением.

Актуальная версия стандарта OAuth 2.0, опубликована в 2012 году в документе IETF RFC6749. OAuth 2.0 позволяет сторонним приложениям получать доступ от своего

имени или ограниченный доступ к HTTP-службе от имени владельца ресурса, организовав процесс согласования взаимодействия между владельцем ресурса и HTTP-службой. Результатом авторизации является Access Token — ключ, предъявление которого является пропуском к защищенным ресурсам. Стандарт не определяет формат ключа, который получает приложение, поэтому ключ сам по себе не может быть использован для аутентификации пользователя [6].

Приведенный ниже (рис. 2) алгоритм демонстрирует ключевые особенности логики работы протокола, позволяющие решить задачу авторизации, т.е. предоставления права на использование ресурса. Чтобы определить присутствие прав, необходим токен (запись или значение, обеспечивающее уникальную идентификацию). Отметим, что один и тот же токен может быть повторно использован для различных пользователей, в то же время у одного и того же пользователя в процессе авторизации могут поменяться токены при наличии специфических временных требований для их обновления. Чтобы получить права на работу с ресурсом необходимо предъявить соответствующий токен.

Таким образом, снижение риска несанкционированного доступа к ресурсам, и, как следствие, обеспечение доступности информации, можно добиться за счет внедрения механизма аутентификации.

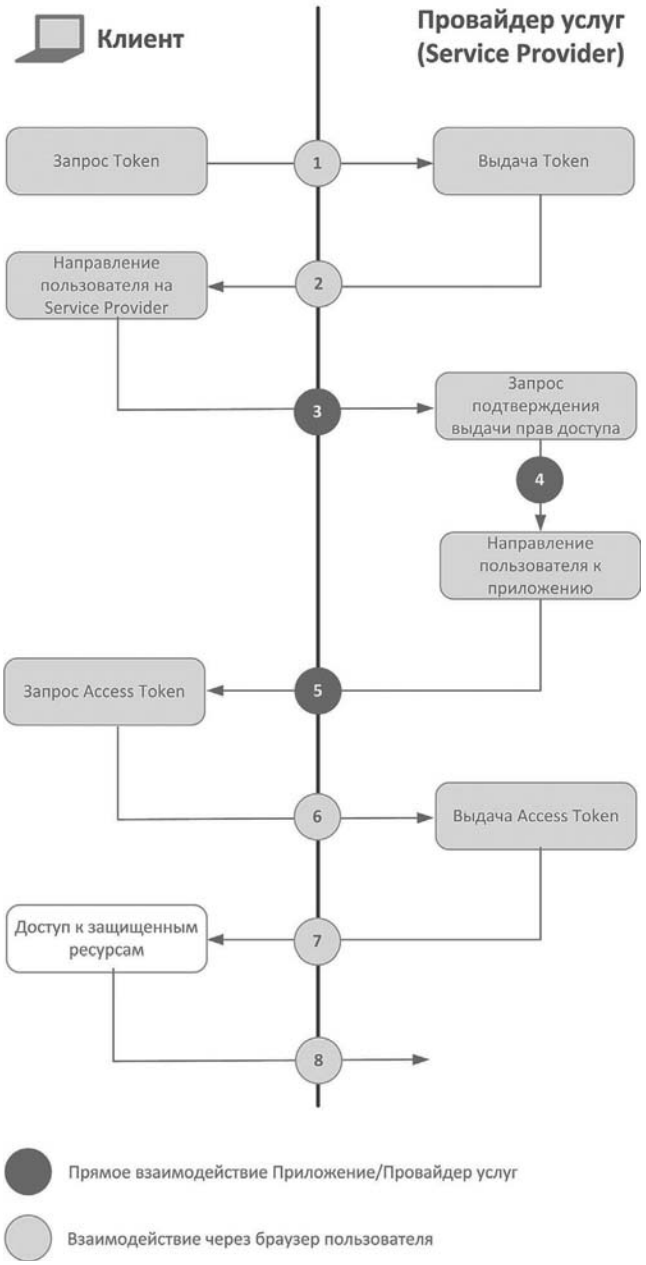


**Рис. 2.** Алгоритм работы протокола OAuth 2.0

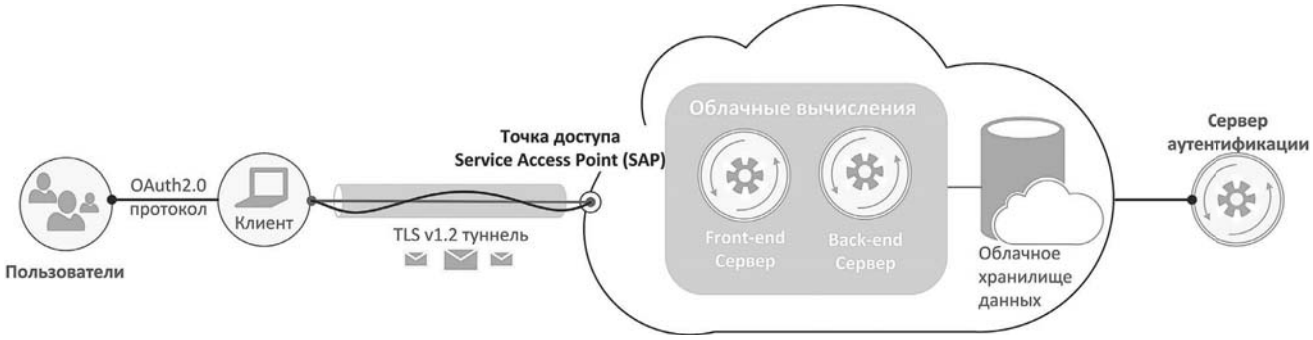


**Рис. 3.** Концептуальная схема построения защищенной облачной среды

Суммируя описанные ранее подходы, представим концептуальную схему (рис. 3), которая отражает ключевые элементы облачной архитектуры (клиентскую часть приложения, сервер аутентификации, сервер приложения (который включает в себя механизмы обработки информации), а так же хранилище данных). Дополнительно, на схеме отмечено, на каких сегментах сети применимы рассмотренные ранее протоколы OAuth 2.0, TLSv1.2 для обеспечения надежного соединения.

**Выводы**

В работе рассмотрен алгоритм установления защищенного TLS соединения, в основе которого лежит обмен открытыми ключами шифрования, алгоритм обмена сертификатами для проверки достоверности узлов, участвующих в обмене конфиденциальными данными, а так же алгоритм проверки целостности полученной информации на стороне принимающего узла (клиента, либо сервера), базирующийся на подсчете MAC-суммы каждого отправленного сообщения. Было выявлено, что для создания надежного TLS соединения важно иметь возможность аутентификации именно клиентской части приложения, а не самого пользователя. В работе отмечено, что в основе протокола TSL v1.2 отсутствуют механизмы контроля времени жизни пользовательской сессии и механизмы повторной аутентификации для возобновления сессии в случае разрыва соединения.

Работа содержит исследование возможности получения авторизованного доступа пользователей к ресурсам, где показано, что для этой цели необходимо внедрение средств аутентификации пользователей за счет протокола OAuth2.0. Для этого была описана диаграмма поэтапного обмена информацией между клиентом и провайдером услуг аутентификации.

В заключение исследования представлена генерализованная схема организации защищенного доступа к облачной середе, которая включает в себя рассмотренные ранее механизмы обеспечения целостности, конфиденциальности и доступности.

Последующими шагами исследования можно полагать рассмотрение способов хранения конфиденциальной информации в зашифрованном виде, а так же методов ее обработки и поиска в облачном хранилище.

**Литература**

1. *Wayne Jansen*, *Timothy Grance*. NIST SP 800–144 Guidelines on Security and Privacy in Public Cloud Computing, December 09, 2011. 80 p. URL: http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909494

2. Об информации, информационных технологиях и о защите информации. Федеральный закон от 27 июля 2006 г. № 149-ФЗ. URL: http://pravo.gov.ru/proxy/ips/?docbody=&nd=102108264 (дата обращения 25.12.2016).

3. FIPS197. Advanced Encryption Standard. Federal Information Processing Standard, NIST, U.S. Dept. of Commerce, November 26, 2001. URL: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf (дата обращения 01.02.2017).

4. U.S. Patent 4,405,829. Cryptographic Communications system and method. Ronald L. Rivest, Adi Shamir, Leonard M. Adleman. Declared 14.12.1977. Published 20.09.1983.

5. *Dierks T.*, *Rescorla E.* The Transport Layer Security (TLS) Protocol, Version 1.2 (RFC5246). URL: https://tools.ietf.org/html/rfc5246 (дата обращения 13.12.2016).

6. *Li W.*, *Mitchell C.J.* (2014) Security Issues in OAuth 2.0 SSO Implementations. In: Chow S.S.M., Camenisch J., Hui L.C.K., Yiu S.M. (eds) Information Security. ISC2014. Lecture Notes in Computer ScienceSpringer-Verlag, 2014. Vol. 8783. Pp. 529–541.

# RESEARCH OF MECHANISMS OF THE PROTECTED ACCESS PROBLEM TO CLOUD DATA STORAGE

**Dmitry V. Sakharov,**
St. Petersburg, Russia, d.sakharov@rkn.gov.ru

**Mark V. Levin,**
St. Petersburg, Russia, m.va.levin@gmail.com

**Elena S. Fostach,**
St. Petersburg, Russia, elena.fostach@gmail.com

**Lidiya A. Vitkova,**
St. Petersburg, Russia, iskinlidia@gmail.com

ABSTRACT
Cloud computing is a new computational paradigm that offers a distributed infrastructure. Despite the potential gains achieved from the cloud computing, the model security is still questionable which impacts the cloud model adoption. The security problem becomes more complicated under the cloud model as new dimensions have entered into the problem scope related to the model architecture, multi-tenancy and elasticity. Cloud computing security concerns, especially data security and privacy protection issues, remain the first problem of cloud computing services.
In the actual article we introduce a detailed analysis of the cloud security problem. We investigated the problem from the cloud architecture. Based on this analysis we offers a detailed specification of the cloud security problem and key features that should be covered by any proposed security solution.

**Keywords:** privacy protection; cloud architecture; cloud computing; cloud computing security; data segregation; data security.

**References**

1. Wayne Jansen, Timothy Grance. NIST SP 800–144 *Guidelines on Security and Privacy in Public Cloud Computing*, December 09, 2011. 80 p. URL: http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909494.
2. *Ob informatsii, informatsionnykh tekhnologiyakh i o zashchite informatsii* [On information, information technologies and protection of information] Federal'nyy zakon ot 27 iyulya 2006 g. № 149-FZ [Federal law of July 27, 2006 149-FZ] URL: http://pravo.gov.ru/proxy/ips/?docbody=&nd=102108264 (date of access 25.12.2016). (In Russian)
3. FIPS197. Advanced Encryption Standard. Federal Information Processing Standard, NIST, U.S. Dept. of Commerce, November 26, 2001. URL: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf (date of access 01.02.2017).
4. U.S. Patent 4,405,829. Cryptographic Communications system and method. Ronald L. Rivest, Adi Shamir, Leonard M. Adleman. Declared 14.12.1977. Published 20.09.1983.
5. Dierks T., Rescorla E. The Transport Layer Security (TLS) Protocol, Version 1.2 (RFC5246). URL: https://tools.ietf.org/html/rfc5246 (дата обращения 13.12.2016).
6. Li W., Mitchell C.J. (2014) Security Issues in OAuth 2.0 SSO Implementations. In: Chow S.S.M., Camenisch J., Hui L.C.K., Yiu S.M. (eds) Information Security. ISC2014. Lecture Notes in Computer ScienceSpringer-Verlag, 2014. Vol. 8783. Pp. 529–541.

**Information about authors:**

Sakharov D.V., PhD, associate professor at the Department of Protected communication systems of the St. Petersburg State University of Telecommunications prof. Bonch-Bruevich;
Levin M.V., postgraduate student of the St. Petersburg State University of Telecommunications prof. Bonch-Bruevich;
Fostach E.S., graduate student of the St. Petersburg State University of Telecommunications prof. Bonch-Bruevich;
Vitkova L.A., postgraduate student of the St. Petersburg State University of Telecommunications prof. Bonch-Bruevich.

**For citation:** Sakharov D.V., Levin M.V., Fostach E.S., Vitkova L.A. Research of mechanisms of the protected access problem to cloud data storage. *H&ES Research*. 2017. Vol. 9. No. 2. Pp. 40–46. (In Russian)