**УДК 004.735**

# АНАЛИЗ И ВЫБОР МЕХАНИЗМА МЕЖСАЙТОВОЙ АВТОРИЗАЦИИ

**Аксёнов А.Н., Аксёнова М.В.**

*Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования «Московский государственный технический университет имени Н.Э. Баумана» (МГТУ им. Н.Э.Баумана), Москва, Россия (105005, 2-я Бауманская ул., д. 5, стр. 1), e-mail: ax.andrey@gmail.com*

**В данной статье авторы рассматривают механизмы реализации межсайтовой авторизации. Задача подразумевает реализацию возможности авторизации на веб-ресурсе, используя учетную запись с другого ресурса, при этом сохраняя существующие права (привилегированные, обычные или ограниченные) и обеспечивая защиту данных пользователя. Авторы рассматривают три возможных варианта реализации задачи: вход с проверкой логина и пароля на головном сайте, вход по зашифрованной ссылке, а также используя существующее решение на основе технологии OpenID. Авторы подробно анализируют способы их реализации, достоинства и недостатки с точки зрения простоты реализации, удобства для пользователя и безопасности. Сделан вывод о том, что оптимальным вариантом является метод, использующий зашифрованную строку. Его достаточная простота реализации, высокая безопасность и удобство для пользователя перевешивают небольшие недостатки, в данном случае не критичные: работа с задержкой и невозможность зайти на вспомогательный ресурс с закладки.**

Ключевые слова: веб-сайт, авторизация, шифрование, защита данных, межсайтовая авторизация, OpenID, Rivest Cipher.

# ANALYSIS AND OPTION OF CROSS-SITE AUTHENTICATION MECHANISM

**Aksenov A.N., Aksenova M.V.**

*Federal budget-funded institution Bauman Moscow State Technical University (BMSTU), Moscow, Russia (105005, Vtoraya Baumanskaya St., 5, Bld. 1), e-mail: mumc@bmstu.ru*

**In this article, the authors review the mechanisms for the implementation of cross-site authorization. The task involves the implementation of authorization capabilities on a web resource using an account from another resource, while maintaining existing rights (preference shares, ordinary or limited) and protecting user data. The authors consider three possible variants of realization of the problem: input validation of username and password on our main site, login via an encrypted link, and using an existing solution based on OpenID technology. The authors analyzed the ways of their implementation, advantages and disadvantages from the point of view of simplicity of implementation, user friendliness and security. It is concluded that the best solution is the method of using the encrypted string. Its sufficient ease of implementation, high security and user convenience outweigh the minor flaws, which in this case is not critical: working with delay and inability to get to the secondary resource from the bookmark.**

Keywords: website, authorization, encryption, data protection, cross-site authorization, OpenID, Rivest Cipher.

## Постановка задачи

Необходимо обеспечить взаимодействие двух сайтов, расположенных на разных серверах и имеющих различные доменные имена, так, чтобы пользователи, имеющие учетную запись на головном сайте, могли пользоваться услугами вспомогательного сайта. Пользователи, не имеющие учетной записи на основном сайте, не должны иметь права доступа к услугам вспомогательного сайта. В случае если на головном сайте пользователь имеет ограниченные права (например, если его права были ограничены администрацией головного сайта из-за нарушения правил пользования), то такой пользователь также не имеет права авторизовываться на дополнительном сайте. Пользователи же, имеющие повышенные

права (например, администрация), также должны иметь повышенные права на вспомогательном сайте [3].

При этом данный механизм должен обеспечивать защиту личных данных пользователей, недопустимость раскрытия их персональной информации, а также защитить вспомогательный сайт от возможных посещений со стороны и, в особенности, попыток получить привилегированные права на вспомогательном ресурсе.

Данная задача актуальна в случаях, когда услуги одного ресурса разнесены на два сервера, имеющие разные доменные имена для распределения нагрузки на аппаратное обеспечение серверов и загрузки интернет каналов. Также задача может быть актуальна для различных дополнительных сервисов, обеспечивающих вспомогательными услугами пользователей крупного ресурса [7].

Стандартом авторизации в сети интернет является проверка логина и пароля. Далее, в процессе работы с ресурсом проверка авторизации пользователя осуществляется с помощью механизма кук (cookie) или с помощью сессий. Однако для обеспечения безопасности архитектура веб-сайтов в сети интернет организована таким образом, что к кукам или к сессиям, установленным одним сайтом, невозможно получить доступ с другого сайта [4]. Поэтому на каждом сайте (основном и вспомогательном) должны быть открыты отдельные изолированные сессии.

**Поиск и анализ вариантов решения задачи**

*Проверка логина и пароля на стороне основного сайта и выдача одобрения или отказа*

Механизм должен работать следующим образом: на основном сайте, на котором, собственно, и хранятся данные об учетных записях пользователей, с помощью простого скрипта осуществляется проверка соответствия логина и пароля. Логин и хэш пароля передаются запросом к данному скрипту со стороны вспомогательного сайта. В случае если логин и хэш пароля совпадают с данными учетной записи пользователя, то скрипт выдает флаг одобрения. В противном случае – флаг отказа. В случае если пользователь имеет привилегированные права на основном сайте, скрипт выдает флаг о привилегированных правах.

Достоинствами данного способа являются простота реализации и режим работы в реальном времени. Явными недостатками являются, во-первых, открытая передача между сайтами логина и хэша пароля пользователя, которые возможно перехватить в случае атаки. Этот минус можно скомпенсировать, передавая логин и хэш пароля в зашифрованном виде, однако это увеличит сложность реализации механизма межсайтовой авторизации.

Вторым недостатком является то, что доступ к данному скрипту открыт со стороны внешних ресурсов. Таким образом, злоумышленник может, узнав адрес этого скрипта, проверять перебором соответствия логинов и хэшей паролей. Этот недостаток можно скомпенсировать, проверяя, от кого приходит запрос на проверку данных пользователя, и, в случае если этот запрос исходит не от разрешенного ресурса, скрипт не должен выдавать результатов. Это возможно сделать, используя, например, параметр referer, приходящий с любым интернет-запросом. Однако этот параметр легко подделать при формировании фальшивого запроса, поэтому фильтрацию необходимо осуществлять с помощью IP-адреса ресурса.

*Авторизация пользователя на вспомогательном ресурсе, используя зашифрованную строку, передаваемую вместе с http запросом*

Этот механизм работает таким образом: на стороне основного сайта генерируется ссылка на вспомогательный сайт. В ссылке в зашифрованном виде содержится имя пользователя и параметр, удостоверяющий его привилегированные права (в случае их наличия). Для каждого пользователя такой параметр будет уникальным. Таким образом, снимается необходимость передачи хэша пароля между сайтами, что обеспечивает необходимую безопасность сохранности данных. Эта ссылка с параметром должна генерироваться только в том случае, когда пользователь уже авторизован на основном сайте и не имеет ограниченных прав. Таким образом, мы обеспечим безопасность от входов на вспомогательный сайт со стороны.

При реализации данного механизма встает вопрос выбора способа шифрования данных. Для поставленной задачи вполне достаточно механизмов симметричного шифрования, то есть криптографической системы с открытым ключом. Рассмотрим несколько способов шифрования [5].

*Анализ методов шифрования*

Шифр Плейфера или квадрат Плейфера – ручная симметричная техника шифрования, в которой впервые использована замена биграмм. Изобретена в 1854 году Чарльзом Уитстоном, но названа именем Лорда Лайона Плейфера, который внедрил данный шифр в государственные службы Великобритании. Шифр предусматривает шифрование пар символов (биграмм) вместо одиночных символов, как в шифре подстановки и в более сложных системах шифрования Виженера [2]. Таким образом, шифр Плейфера более устойчив к взлому по сравнению с шифром простой замены, так как затрудняется частотный анализ. Он может быть проведен, но не для 26 возможных символов (латинский алфавит), а для 26x26=676 возможных биграмм. Анализ частоты биграмм возможен, но является значительно более трудным и требует намного большего объема зашифрованного текста.

Достоинством данного шифра является его простая реализация и отсутствие ключа для расшифровки. Однако, как и большинство шифров формальной криптографии, шифр Плейфера также может быть легко взломан, если имеется достаточный объем текста. Получение ключа является относительно простым, если известны шифрованный и обычный текст. Когда известен только зашифрованный текст, криптоаналитики анализируют соответствие между частотой появления биграмм в шифрованном тексте и известной частоте появления биграмм в языке, на котором написано сообщение.

RC4 (Rivest Cipher 4) – потоковый шифр, широко применяющийся в различных системах защиты информации в компьютерных сетях (например, в протоколах SSL и TLS, алгоритме безопасности беспроводных сетей WEP, для шифрования паролей в Windows NT). Алгоритм RC4 строится, как и любой потоковый шифр, на основе параметризованного ключом генератора псевдослучайных битов с равномерным распределением. Длина ключа может составлять от 40 до 256 бит [1].

Основные преимущества шифра – высокая скорость работы и переменный размер ключа. RC4 довольно уязвим, если используются не случайные или связанные ключи, один ключевой поток используется дважды. Эти факторы, а также способ использования могут сделать криптосистему небезопасной (например, WEP).

Работа многих поточных шифров основана на линейных регистрах сдвига с обратной связью (LFSR). Это позволяет достичь высокой эффективности реализаций шифра в виде ИС, но затрудняет программную реализацию таких шифров. Поскольку шифр RC4 не использует LFSR и основан на байтовых операциях, его удобно реализовывать программно. Типичная реализация выполняет от 8 до 16 машинных команд на каждый байт текста, поэтому программная реализация шифра должна работать очень быстро.

Советский и российский стандарт симметричного шифрования, введенный в 1990 году, также является стандартом СНГ. Полное название – «ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования». Блочный шифроалгоритм. При использовании метода шифрования с гаммированием может выполнять функции поточного шифроалгоритма.

Достоинства данного метода шифрования: бесперспективность силовой атаки (XSL-атаки в учет не берутся, так как их эффективность на данный момент полностью не доказана); эффективность реализации и соответственно высокое быстродействие на современных компьютерах; наличие защиты от навязывания ложных данных (выработка имитов ставки) и одинаковый цикл шифрования во всех четырех алгоритмах ГОСТа.

Основные проблемы ГОСТа связаны с неполнотой стандарта в части генерации ключей и таблиц замен. Тривиально доказывается, что у ГОСТа существуют «слабые» ключи

и таблицы замен, но в стандарте не описываются критерии выбора и отсева «слабых». В мае 2011 года известный криптоаналитик Николя Куртуа заявил об обнаружении серьезных уязвимостей в данном шифре.

После анализа достоинств и недостатков нескольких методов шифрования был выбран метод шифрования RC4 (Rivest Cipher 4).

*Анализ достоинств и недостатков метода с передачей зашифрованной строки*

Достоинствами данного метода являются достаточная простота реализации, высокая безопасность, а также удобство для пользователя, поскольку ему не приходится вводить на стороне дополнительного сайта никаких данных – авторизация будет производиться автоматически.

Недостатками данного механизма является, во-первых, то, что пользователь, воспользовавшись ссылкой, сгенерированной для другого пользователя, сможет войти на вспомогательный сайт под чужим именем. Получить данную ссылку он может либо получив прямой доступ к компьютеру другого пользователя и скопировав ссылку, либо получив ее от другого пользователя напрямую, например, просто попросив ее. Это, естественно, маловероятно, но возможно. Недостаток легко компенсируется, если передавать зашифрованную строку не открытым параметром, а с помощью POST запроса. Однако это добавит неудобства пользователям, поскольку в этом случае пользователь может получить доступ к вспомогательному сайту только по прямой ссылке с основного. Он не сможет войти авторизованным пользователем на вспомогательный сайт, просто набрав его адрес, или выбрав закладку в браузере. Этот недостаток неустраним, хотя и не сильно существенен.

Второй, более важный недостаток состоит в том, что пользователь, в случае, если он будет в какой-то момент ограничен в правах на основном сайте (например, в случае нарушения прав пользования ресурсом) и при этом оставит (сохранит) страницу с ссылкой на дополнительный ресурс заранее, то, пройдя по этой ссылке с сохраненной страницей, он получит доступ к услугам на вспомогательном сайте.

Этот недостаток можно скомпенсировать, передавая время генерирования ссылки вместе с зашифрованной строкой, содержащей имя пользователя и флаг привилегированных прав, естественно, также в зашифрованном виде. На вспомогательном сайте же необходимо осуществлять дополнительную проверку промежутка времени между временем генерирования ссылки и временем входа на вспомогательный сайт. В случае если этот промежуток не превышает какого-то разумного времени (допустим, даже часа), то пользователь может авторизоваться на вспомогательном сайте. Некоторый разумный промежуток времени необходим, поскольку пользователь может открыть страницу основного сайта, сгенерировав таким образом ссылку, но пройти по ней не сразу, а спустя

какое-то время. Однако при данной реализации пользователь также будет лишен возможности заходить на вспомогательный сайт, просто набрав его имя в адресной строке и выбрав закладку.

*Авторизация с помощью механизма OpenID*

На данный момент существует распространенный признанный механизм межсайтовой авторизации, называющийся OpenID и разработанный создателем сервиса Живой Журнал (livejournal.com) Брэдом Фицпатриком.

OpenID – это спецификация, описывающая, как пользователь может доказать, что является владельцем того или иного идентификатора. Для простоты можно считать, что идентификатор – это строка, которая уникальным образом идентифицирует пользователя. Этот стандарт разработан для того, чтобы пользователь, имея учетную запись на одном сервисе, мог использовать ее для входа на другие сервисы и сайты. По последним данным с официального сайта OpenID он поддерживается более чем 50000 сайтами, среди которых Facebook, Yahoo!, Google, Twitter, Яндекс, Вконтакте и Живой Журнал.

Центральное место в OpenID занимает аутентификация, которая включает в себя три ключевых понятия [6]:

- идентификатор OpenID: строка, которая уникально идентифицирует пользователя;
- клиент OpenID (OpenID Relying Party или RP): online-ресурс (чаще всего Web-сайт, но им также может быть файл, изображение или любой ресурс, к которому необходимо контролировать доступ), который использует OpenID для идентификации обращающихся к нему пользователей;
- провайдер OpenID (OP): сайт, на котором пользователи могут завести OpenID и который может в будущем авторизовывать и аутентифицировать пользователей, обращающихся к RP.

Некий пользователь, пытаясь получить доступ к ресурсу, который является частью Web-сайта RP, поддерживающего OpenID, должен предъявить свой идентификатор OpenID в той форме, которая позволяет распознать в нем протокол OpenID. В частности, идентификатор OpenID указывает на местоположение OP. Таким образом, RP извлекает идентификатор и перенаправляет пользователя на сайт OP, где он должен будет подтвердить, что является владельцем данного идентификатора. Этот идентификатор следует нормализовать, т.е. преобразовать к виду заявленного идентификатора. Заявленный идентификатор может затем использоваться для нахождения OP (этот процесс называется «обнаружением провайдера»), который должен будет аутентифицировать пользователя. Клиенты OpenID (RP) получают предъявленные пользователем идентификаторы и преобразуют их в заявленные идентификаторы. После этого они перенаправляют браузер

пользователя (пользовательский агент) к OP, где можно ввести параметры учетной записи для аутентификации.

RP ничего не знает о деталях аутентификации по заявленным идентификаторам. Все, что его интересует – это результат аутентификации (положительный или отрицательный). Если аутентификация прошла успешно, то пользователь возвращается к защищенному ресурсу, к которому он изначально пытался получить доступ. В противном случае RP отказывает в доступе к ресурсу. Провайдеры OpenID отвечают за выделение идентификаторов и выполнение аутентификации. Они также предоставляют Web-интерфейс для управления выданными идентификаторами. При запросе на аутентификацию по заявленному идентификатору OP перенаправляет пользователя на страницу авторизации, которая запрашивает его имя и пароль. В этот момент процесс аутентификации полностью переходит к OP. Если пользователь успешно аутентифицировался, OP перенаправляет его браузер по адресу, указанному RP («return-to» URL). В противном случае пользователь должен получить сообщение, что попытка аутентификации окончилась неудачно [6].

Достоинствами такого метода являются: высокая безопасность личных данных пользователя, широкая распространенность (а значит, надежность кода) и документированность, а также возможность работы в режиме реального времени.

Однако, реализуя поставленную задачу, стандартный функционал OpenID необходимо дополнить и исправить, во-первых, разрешив авторизовываться на вспомогательном сайте только пользователям с основного сайта, а во-вторых, необходимо осуществить передачу флага о привилегированных правах. В случае если основной сайт, который должен являться OpenID провайдером, основан на одной из распространенных систем управления контентом, то, возможно, с помощью дополнительных модулей можно обеспечить его необходимым функционалом. В противном случае этот функционал необходимо реализовывать самостоятельно, что в несколько раз увеличивает время и сложность разработки данного метода по сравнению с первыми двумя методами. Кроме того, на стороне вспомогательного сайта тоже необходимо реализовывать возможность авторизации с помощью механизма OpenID. Также будет необходимо внести соответствующие исправления в стандартный механизм, как на стороне основного сайта, так и на стороне вспомогательного сайта.

**Выводы**

Изучив и проанализировав возможные варианты решения задачи, были сделаны следующие выводы: использование проверки логина и пароля на стороне сервера при своей простоте не обеспечивает критический уровень безопасности. Использование же OpenID было признано нецелесообразным, поскольку для данной задачи сложность интеграции этого механизма неоправданна. Оптимальным вариантом был признан метод, использующий

зашифрованную строку. Его достаточная простота реализации, высокая безопасность и удобство для пользователя перевешивают небольшие недостатки, в данном случае не критичные: работа с задержкой и невозможность зайти на вспомогательный ресурс с закладки.

## Список литературы


1. Бабаш А.В., Кудияров Д.С. Период функционирования генератора псевдослучайных чисел RC4 // Системы высокой доступности. – 2012. – Т. 8. – № 2. – С. 7-11.
2. Бабаш А.В. Определение периода гаммы в шифре Виженера по заданному шифртексту // Проблемы информационной безопасности. Компьютерные системы. – 2014. – № 4. – С. 66-75.
3. Басараб М.А., Иванов И.П., Колесников А.В. Анализ моделей прогнозирования процессов сервера корпоративной сети // Нелинейный мир. – 2015. – Т. 13. – № 3. – С. 18-31.
4. Кузовлев В.И., Орлов А.О. Методика выбора параметров и интерпретации результатов анализа выбросов в данных систем поддержки принятия решений // Инженерный журнал: наука и инновации: электронное научно-техническое издание. – 2013. – № 11 (23). – URL http://engjournal.ru/catalog/it/hidden/1045.html (дата обращения 29.03.2015).
5. Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф. Защита информации в компьютерных системах и сетях. – М: Радио и связь, 1999. – 328 с.
6. Стивен Перри Дж.. OpenID в Web-приложениях на Java // IBM developerWorks: Ресурс IBM для разработчиков и IT профессионалов. – 2010. – URL: http://www.ibm.com/developerworks/ru/library/j-openid/ (дата обращения: 26.06.2015).
7. Цибизова Т.Ю., Слепцова К.А. Автоматизированная система учета данных внутрикорпоративной сети управления информацией // Современные проблемы науки и образования. – 2015. – № 1; URL: www.science-education.ru/121-19593.



**Рецензенты:**

Пролетарский А.В., д.т.н., профессор, декан факультета «Информатика и системы управления», зав. кафедрой «Компьютерные системы и сети», МГТУ им. Н.Э.Баумана, г. Москва;

Неусыпин К.А., д.т.н., профессор, профессор кафедры «Системы автоматического управления», директор Научно-образовательного центра «Интеллектуальные системы», МГТУ им. Н.Э.Баумана», г. Москва.