На правах рукописи

ТЮЛЬКИН Михаил Валерьевич

АЛГОРИТМЫ ПОВЫШЕНИЯ УРОВНЯ КОНФИДЕНЦИАЛЬНОСТИ ДАННЫХ ПРИ ПЕРЕДАЧЕ В COMET-ПРИЛОЖЕНИЯХ

Специальность 05.13.19 –

Методы и системы защиты информации, информационная безопасность

АВТОРЕФЕРАТ

диссертации на соискание ученой степени

кандидата технических наук

Уфа – 2013

Работа выполнена на кафедре прикладной физики
ФГБОУ ВПО «Пермский национальный исследовательский политехнический университет»

Научный руководитель	д-р ф.-м. наук, доцент, **Кротов Лев Николаевич,**
Официальные оппоненты	д-р.техн. наук, профессор, **Фрид Аркадий Исаакович,** ФГБОУ ВПО «Уфимский государственный авиационный технический университет»
	канд.ф.-м.наук, доцент **Ермаков Дмитрий Германович,** ФГБУН Институт математики и механики им. Н.Н. Красовского УрО РАН, старший научный сотрудник
Ведущее предприятие	**ФГБОУ ВПО «Пермский государственный национальный исследовательский университет»**

Защита состоится 01 ноября 2013 г. в 10:00 часов
на заседании диссертационного совета Д 212.288.07
при Уфимском государственном авиационном техническом университете
в актовом зале 1-го корпуса по адресу: 450000, г. Уфа, ул. К. Маркса, 12

С диссертацией можно ознакомиться в библиотеке
Уфимского государственного авиационного технического университета.

Автореферат разослан «___» _____ 2013 г.

Ученый секретарь
диссертационного совета
д-р.техн. наук, доцент **И.Л. Виноградова**

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы исследования.Впервые термин *Comet* в 2006 ввел *AlexRussell*, инженер компании *Google*, для обозначения нового класса веб-приложений,позволяющих передачу информации от сервера к клиенту по инициативе первого. Клиентом в *Comet*-приложении, как и в веб-приложении, является браузер конечного пользователя, который передает веб-серверу информацию по *HTTP* протоколу. Однако, сервер в *Comet*-приложении, в отличие от классическоговеб-приложения, может передавать данные клиенту как по своему синхронному каналу передачи информации через *HTTP* протокол, так и, через третий элемент *Comet*-приложения – *Comet*-сервер, по асинхронному каналу связи через протокол *WebSocket*. В *Comet*-приложении клиент устанавливает связь с *Comet*-сервером после получения первого ответа от веб-сервера, в котором содержатся данные для подключения.

Стандарт протокола *WebSocket*, предложенного *I. Fette* и *A. Melnikov*, был принят относительно недавно, в декабре 2011 года, однако, уже в течение года данный стандарт был реализован во всех наиболее популярных браузерах (*Firefox, Chrome, Safari, Opera, InternetExplorer*), которые охватывают около 95% аудитории всех пользователей *Internet*. Компания *Microsoft* реализовала данный протокол в десятой версии браузера *InternetExplorer*, в феврале 2013 г.

Данный факт подтверждает, что развитие *Comet*-приложений является актуальной темой востребованной в мировом веб-сообществе.

Степень разработанности темы.Различные решения асинхронного взаимодействия в веб-приложениях, с предложением новых технологий были предложены в работах таких ученых *D. Crane, P. McCarthy, A. Russell, I. Fette, A. Melnikov, G. Wilkins, D. Davis, M. Nesbitt.*

Однако, несмотря на стремительность внедрения и развития,*Comet*-приложения обладают рядом существенных недостатков с точки зрения защиты информации. А именно, протокол *WebSocket*не предусматривает какой-либо системы разграничения доступа к пересылаемым через *Comet*-сервер данным, т.е. все данные, пересылаемые через *Comet*-сервер, получают все подключенные клиенты. Дополнительно протокол *WebSocket*регламентирует и защищает передачу данных в канале связи только между клиентом и *Comet*-сервером, но не предусматривает защиты идентификационных данных клиента передаваемых от веб-сервера к *Comet*-серверу через клиента, т.к. *Comet*-сервер не может доподлинно определить какой подключившийся к нему клиент является клиентом веб-сервера. Отсутствие защиты идентификационных данных позволяет подменить их злонамеренному клиенту или враждебному программному обеспечению (далее ПО) на компьютере пользователя, и тем самым получить доступ к информации,передаваемой через *Comet*-сервер ипредназначенной для другого клиента. Данные аспекты не позволяют передавать веб-серверу через *Comet*-сервер конфиденциальную информацию, т.к. всегда существует вероятность несанкционированного ознакомления с ней.

Другая уязвимость информационного обмена в *Comet*-приложении заключается в том, что при компрометации *Comet*-сервера будет скомпрометирован весь информационный поток, идущий через этот элемент *Comet*-

приложения, что, в свою очередь, опять же снижает такие показатели безопасности информации, как конфиденциальность и целостность, и не позволяет клиенту доверять пересылаемым через *Comet*-сервер данным.

Таким образом, учитывая динамику внедрения *Comet*-приложений в веб-среду, повышение уровня конфиденциальности передаваемых данных в *Comet*-приложении является первоочередной задачей требующей решения.

Объектом исследований является совокупность процессов в *Comet*-приложении.

Предметом исследованийявляютсяспособы обеспечения защиты информации в *Comet*-приложении.

Цельюработы являетсяразработка алгоритмов повышения уровня конфиденциальности данных при передаче в *Comet*-приложении.

Задачи исследования

1. Адаптация алгоритма избирательного разграничения доступа применительно к *Comet*-приложениям.

2. Разработка алгоритма защиты от несанкционированной модификации идентификационных данных клиента в *Comet*-приложении с применением криптографических преобразований;

3. Создание алгоритма повышения конфиденциальности и целостности передаваемых через *Comet*-сервер данныхна основе криптографических методов защиты информации;

4. Разработка архитектуры *Comet*-сервера с применением теории многопоточного программирования повышающей защищенность *Comet*-приложения перед классом атак «отказ в обслуживании», а также с целью реализации предложенных алгоритмов.

Научная новизна работы состоит в следующем:

1. Новизна адаптированного алгоритма избирательного разграничения доступа, позволяющего в рамках протокола *WebSocket* впервые реализовать разграничение доступа клиентов к данным пересылаемым в *Comet*-приложении через *Comet*-сервер, алгоритм отличается тем, что реализует внутри одного канала связи по протоколу *WebSocket* несколько каналов для различных категорий данных, вводя новый формат сообщений для обмена защищаемыми данными в *Comet*-приложении, предусматривающий указание категории пересылаемых данных $D_{K,Ch}$, списка идентификаторов клиентов $D_{K,ID}$, которым эти данные предназначены, и метки времени сообщения $D_{K,T}$, для оценки его актуальности.

2. Впервые предложен алгоритм защиты идентификационных данных клиента, пересылаемых *Comet*-серверу, основывающийся на создании защищенного пакета данных, с помощью уже известных способов криптографических преобразований для предотвращения, модификации или повреждения враждебным программным окружением клиента или самим злонамеренным клиентом.

3. Новый алгоритм повышения конфиденциальности и целостности данных, передаваемых через *Comet*-сервер, отличается тем, что защищает конфиденциальные данные на всем пути следования от веб-сервера к клиенту через

Comet-сервер, не влияя на работу *Comet*-сервера, и, тем самым, позволяя ему обрабатывать поступающие сообщения в штатном режиме, но без доступа к их исходному тексту. Данный алгоритм повышает конфиденциальность передаваемых данных путем криптографического преобразования их исходного текста, с последующим транспортным кодированием полученного шифротекста, для исключения несанкционированной модификации или ознакомления с текстом передаваемого сообщения.

4. Разработан и описан ряд новых математических моделей архитектуры *Comet*-серверов под различные классы задач, решаемые *Comet*-сервером, для реализации предложенных алгоритмов. Данные модели отличаются тем, что реализуют выполнение задач *Comet*-сервера в различных потоках исполнения, что позволяет*Comet*-серверу, при реализации атаки типа «отказ в обслуживании», потерять лишь часть функционала, реализуемого атакованным потоком исполнения, но не выйти полностью из строя, и, тем самым, сохранить работоспособность *Comet*-приложения в целом.

Теоретическая и практическая ценностьполученных результатов состоит в возможности комплексно защитить передаваемые данные в *Comet*-приложении, что, в свою очередь, позволяет применять *Comet*-приложения для обработки и передачи конфиденциальных данных. Предложенный алгоритм разграничения доступа позволяет поддерживать принятую в организации политику разграничения доступа пользователей на уровне *Comet*-приложения и *Comet*-сервера, без модификации последнего. Предложенный алгоритм защиты идентификационных данных клиента *Comet*-приложения исключает возможность выдачи злоумышленником себя за легального пользователя, а также, исключает возможность модификации или перехвата идентификационных данных клиента враждебным программным окружением, если клиент *Comet*-приложения выполняется не в защищенной программно-аппаратной среде. Предложенный алгоритм защиты данных передаваемых через *Comet*-сервер позволяет сохранить конфиденциальность данных при компрометации *Comet*-сервера. Предложенные архитектуры *Comet*-серверов позволяют снизить ущерб от реализации атак типа «отказ в обслуживании».

Методология и методы исследования.Результаты работы получены с использованием фундаментальных положений теории информации, методов аналитического и имитационного моделирования, методов системного анализа. При решении конкретных практических задач использовались методы объектно-ориентированного программирования, основные положения и методы теории потокового программирования, теории планирования параллельных вычислительных процессов. В качестве инструментария для решения практических задач использовались такие языки программирования как *C++, PHP, JavaScript*.

Положения, выносимые на защиту:

1. Адаптированный к *Comet*-приложениям алгоритм избирательного доступа к данным, пересылаемым в *Comet*-приложении через *Comet*-сервер.

2. Алгоритм защиты от несанкционированной модификации идентификационных данных клиента *Comet*-приложения пересылаемых *Comet*-серверу.

3. Алгоритмы повышения конфиденциальности и целостности передаваемых данных через *Comet*-сервер.

4. Ряд архитектурComet-серверов, спроектированных под различные типы *Comet*-приложений и основанных на предложенных алгоритмах.

Достоверность полученных результатов основана на использовании в теоретических построениях законов и подходов, справедливость которых общепризнана, а также, известного и корректного математического аппарата; вводимые допущения мотивированы фактами, известными из практики. Достоверность и обоснованность научных положений подтверждена соответствием результатов теоретических и экспериментальных исследований.

Апробация результатов. Основные результаты докладывались и обсуждались на VII международной конференции «*Dnyvědy. Moderní informační technologie*», Чехия, 2012 г.; IX международной научно-практической конференции «Технические науки – от теории к практике», г. Новосибирск, 2012 г.; I международной научно-практической конференции «Технические науки – основа современной инновационной системы», г. Иошкар-Ола, 2012 г.; краевой научно-технической конференции «Автоматизированные системы управления и информационные технологии», г. Пермь, 2012 г.

Результаты работы, в частности, программные реализации *Comet*-серверов были успешно применены ООО «Нью Солюшнс» в ряде интернет-порталов провайдеров Дом.ru и U-tel.

Публикации.Основные результаты диссертации опубликованы в 12 научных изданиях, в том числе 8 работ – в рецензируемых журналах из списка ВАК, получено 2 свидетельсва о государственной регистрации программ для ЭВМ.

Структура и объем диссертации. Диссертация работа состоит из введения, четырех глав, заключения, библиографического списка и приложений. Содержит 177 стр. машинописного текста, из которых основной текст составляет 125 стр.,22 рисунка, библиографический список из 87 наименований, приложения на 57 стр.

СОДЕРЖАНИЕ ДИССЕРТАЦИИ

Во введении обоснована актуальность работы, сформулированы цель и основные задачи исследования, показаны научная новизна и практическая ценность работы, изложены основные положения, выносимые на защиту.

В первой главе содержится краткий обзор истории развития веб-приложений в отношении изменения их математических моделей, обсуждаются их достоинства и недостатки с точки зрения обеспечения конфиденциальности и доступности информации в приложении, делается акцент на разделение математических моделей веб-приложений по синхронности передачи данныхна классические и неклассические. Приводятся примеры применения *Comet*-технологии в веб-приложениях, позволяющих построить новый тип веб-приложений – *Comet*-приложения.

Приведены ключевые возможности *Comet*-приложения в обеспечении доступности информации, а также рассмотрены его принципиальнаяструктура

и основные элементы, такие как клиент, сервер и *Comet*-сервер, а также принципы взаимодействия между ними, см. рисунок 1.

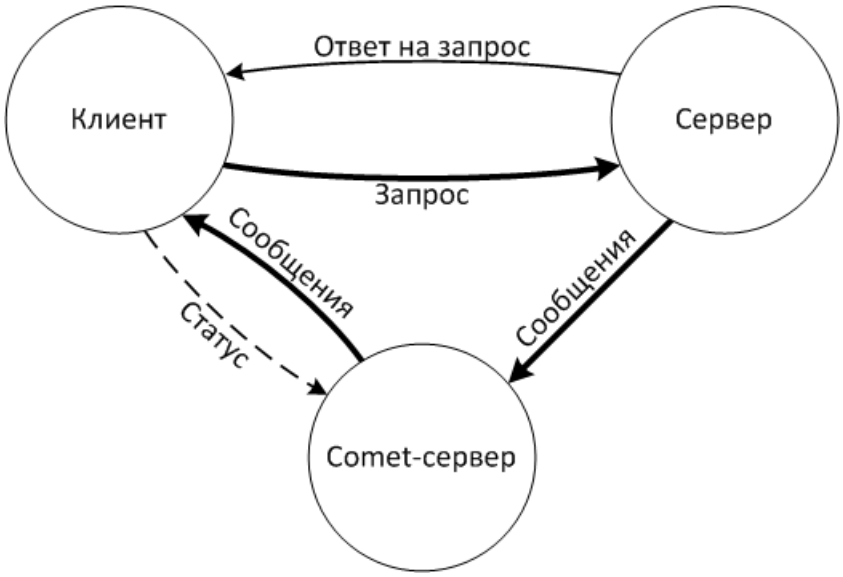

Рисунок 1 – Устройство *Comet*-приложения

Рассмотрены проблемы *Comet*-приложения, связанные с обеспечением конфиденциальности информации при передаче, а именно, проблемы протокола *WebSocket*, регламентирующего передачу информации в *Comet*-приложении, заключающиеся в том, что данный протокол не предусматривает какого-либо механизма разграничения доступа к пересылаемым данным, поскольку не имеет механизмов защиты идентификационных данных клиента, а также, не реализует защиты пересылаемых данных через *Comet*-сервер. Эти два фактора в сумме не позволяют пересылать веб-серверу конфиденциальную информацию через *Comet*-сервер, поскольку доступ к такой информации будут иметь все клиенты приложения. А последний фактор дополнительно не позволяет клиенту доверять пересылаемой информации, поскольку если будет скомпрометирован *Comet*-сервер, то конфиденциальность всех, пересылаемых через него, данных будет нарушена.

На основе этих положений сформулирован перечень задач, имеющих необходимость в решении.

Во второй главепредложена адаптация известного метода избирательного разграничения доступа применительно к *Comet*-приложениям.Приведено математическое описание модели взаимодействия элементов *Comet*-приложения согласно данной адаптации,предложен новый алгоритм, реализующий данный метод разграничения доступа клиентов*Comet*-приложения к данным на технологическом уровне *Comet*-сервера.

Предлагаемый алгоритм охватывает трехсторонний обмен информацией в *Comet*-приложении и предполагает наличие двух логических каналов связи у *Comet*-сервера: серверного и клиентского. Пересылаемые данные разделяются на категории доступа, которые называются каналами, с целью исключения передачи клиенту данных той категории, к которой он не имеет доступа. Работа приложения рассмотренапо стадиям.На стадии подготовки к работемежду сер-

вером и *Comet*-сервером происходит синхронизация времени.На стадии подключения клиента, в ответ на первый запрос, сервер включает в ответ идентификационные данные клиента, а именно:

– $D_{K,ID}$ - уникальный идентификатор клиента;

– $D_{K,Ch}$- множество категорий данных;

– $D_{K,T}$ - метка времени подключения клиента к серверу.

На этапе инициализации клиент подключается к *Comet*-серверу и пересылает своих идентификационные данные в сообщениях M_{ID}, M_{Ch}, M_T, которые составляются по формуле:

$$M_i = Si_i + D_i, \quad (1)$$

где Si_i - сигнатура типа передаваемого сообщения, следовательно:

+ - операция конкатенации;

D_i - передаваемые данные.

Таким образом,

$$M_{ID} = Si_{ID} + D_{ID} \quad (2)$$

$$M_{Ch} = Si_{Ch} + D_{Ch} \quad (3)$$

$$M_T = Si_T + D_T \quad (4)$$

По готовности клиент передает, $M_R = Si_R$ - сообщение готовности. Пока *Comet*-сервер не получит данное сообщение впервые, никакие данные не будут отправлены клиенту.Когда веб-сервер передает информацию для клиента, генерируется сообщение M_S, по следующей формуле:

$$M_S = Si_S + D_{S,ID} + D_{S,Ch} + D_S, \quad (5)$$

где Si_S - сигнатура серверного сообщения;

$D_{S,ID}$ - множество идентификаторов клиентов, для которых предназначено сообщение;

$D_{S,Ch}$ - множество категорий данных, к которым относится это сообщение;

D_S - текст сообщения.

Comet-сервер при получении сообщения M_Sпреобразует его в индивидуальное для каждого клиента сообщение M_C по формуле:

$$M_C = Si_C + D_{\mathrm{C},Ch} + D_S, \quad (6)$$

где Si_C - сигнатура сообщения от *Comet*-сервера;

$D_{\mathrm{C},Ch}$ - множество категорий отдельного клиента, к которым он имеет доступ.

Таким образом,*Comet*-сервер берет на себя роль ретранслятора сообщений и является наилучшим местом информационного обмена в *Comet*-сообщении для внедрения системы разграничения доступа. Предлагаемый алгоритм разграничения доступа основан на адаптации метода избирательного разграничения доступа,реализуется на стороне *Comet*-сервера и состоит из следующей последовательности действий, выполняемых*Comet*-сервером, при генерации сообщения M_C:

1. Проверка условия согласно формуле (7), которое означает, что идентификатор клиента был получен, и он принадлежит множеству идентификато-

ров, полученных от веб-сервера, или, сообщение от веб-сервера имеет сигнатуру широковещательного сообщения Si_{all}:

$$((D_{K,ID} \in D_{S,ID}) \wedge (D_{K,ID} \neq 0)) \vee (D_{S,ID} = Si_{all}) \quad (7)$$

2. Вычисление множества категорий доступа, к которым допущен данный клиент и для которых предназначено полученное сообщение, при этом проверяется условие, что полученное множество не является пустым:

$$D_{C,Ch} = D_{K,Ch} \cap D_{S,Ch} \neq \emptyset \quad (8)$$

3. Проверка условия, что метка времени подключения клиента к приложению старше метки времени получения сообщения от сервераT_S:

$$T_K < T_S \quad (9)$$

4. Проверка условия, что метка готовности была получена:

$$\exists M_R \quad (10)$$

Проанализированы возможные модели поведения злоумышленников по отношению к *Comet*-приложению, возможные реализуемые ими информационныеугрозы, как активные, так и пассивные,а также,их последствия для работы *Comet*-приложения. По результатам анализа наиболее уязвимым местом *Comet*-приложения является линия связи между *Comet*-сервером и веб-сервером, поскольку её контроль или блокировка могут отразиться наиболее пагубно на работе приложения.

Предложен алгоритм защиты идентификационных данных клиента, используемых в вышеописанном алгоритме, реализующем разграничение доступа в *Comet*-приложении, при их передаче от веб-сервера *Comet*-серверу через клиента. Учитывая тот факт, что клиент *Comet*-приложения может выполняться под контролем злоумышленника или работать во враждебном программном окружении, алгоритм защиты идентификационных данных должен исключать реализацию пассивных атак, направленных на несанкционированное ознакомление с передаваемыми данными через клиента, а также, должен позволять обнаруживать активные атаки, направленные на модификацию или порчу передаваемых данных. Для этого работа *Comet*-приложения должна быть перестроена по следующему алгоритму:

1. До начала сеанса работы веб-сервер генерирует ключ K и передает *Comet*-серверу;

2. Веб-сервер, при подключении клиента, генерирует сообщение M_{Status} с данными идентифицирующими клиента, по формуле:

$$M_{Status} = D_{K,ID} + D_{K,Ch} + D_{K,T} \quad (11)$$

Затем подвергает полученное сообщение криптопреобразованию, получая зашифрованное сообщение$M_{E,Status}$, по формуле:

$$M_{E,Status} = E(M_{Status}, K) \quad (12)$$

где $E(x, y)$ - функция шифрования пакета данных x с ключом y;

3. Веб-сервер передает клиенту $M_{E,Status}$ в составе ответа;

4. Клиент передает $M_{E,Status}$*Comet*-серверу при подключении;

5. *Comet*-сервер восстанавливает M_{Status} и идентифицирует клиента.

Таким образом, скрытие оригинально текста сообщения содержащего идентификационные данные не позволяет ознакомиться с ним без знания клю-

ча, который не передается по каналам связи доступным злоумышленнику. В случае, если же линия связи между *Comet*-сервером и веб-сервером доступна для злоумышленника, то для защиты информационного обмена между этими элементами *Comet*-приложения рекомендуется применять проверенные средства защиты информации, например, защищенные сертифицированные протоколы стека *IPSec*.

Предложен алгоритм повышения уровня конфиденциальных данных при передаче через *Comet*-сервер, который позволяет сохранить неизвестность передаваемых данных при компрометации *Comet*-сервера злоумышленником. Для достижения этой цели данный алгоритм предполагает выполнение следующих шагов:

1. В ответ сервера включается сеансовый ключ K;

2. Клиент передает *Comet*-серверу свои идентифицирующие данные в сообщении $M_{E,Status}$;

3. Веб-сервер генерирует сообщение $M_{S,E}$ по формуле:

$$M_{S,E} = Si_S + D_{S,ID} + D_{S,Ch} + D_{S,E} \quad (13)$$

где $D_{S,E}$ - преобразованный текст сообщения по формуле:

$$M_{D,E} = f(D_S, K) \quad (14)$$

где f – функция преобразования.

При этом для открытых данных, требующих только обеспечения целостности, функция преобразования имеет вид:

$$f(D_S, K) = D_S + H(D_S, K) \quad (15)$$

где H – функция хеширования.

Для конфиденциальных данных, дополнительно требующих сокрытие оригинально текста сообщения, функция преобразования принимает вид:

$$f(D_S, K) = E(D_S, K) \quad (16)$$

где E – функция шифрования.

4. Веб-сервер передает полученное сообщение $M_{S,E}$ в составе сообщения M_S:

$$M_S = Si_S + D_{S,ID} + D_{S,Ch} + M_{S,E} \quad (17)$$

5. *Comet*-сервер обрабатывает полученное сообщение M_S и генерирует сообщение M_C, для каждого клиента, при этом шифротекст$M_{S,E}$ не затрагивается при обработке:

$$M_C = Si_C + D_{C,Ch} + M_{S,E} \quad (18)$$

Таким образом, в канале передачи данных от веб-сервера к клиенту через *Comet*-сервер создается туннель для передачи шифрованных сообщений, поскольку шифротекст не затрагивается при обработке сообщений Comet-сервером, это позволяет сохранить конфиденциальность текста передаваемых сообщений при компрометации *Comet*-сервера.

Разработан ряд архитектур *Comet*-серверов под различные типы задач для реализации предложенных алгоритмов. Детально проанализированы задачи *Comet*-сервера, на основе анализа определены основные элементы *Comet*-сервера (вычислительные потоки, области памяти, сокеты). Разработанные модели архитектуры *Comet*-сервера, получили следующие названия:

1. Однопоточная модель;
2. Многопоточная модель;
3. Многопоточная модель с выделением потока под одного клиента;
4. Многопоточная модель с выделением потока под группу клиентов;
5. Многопоточная модель с выделением потока под сообщение;
6. Многопоточная модель с выделением потока под сообщение и учетом хронологии поступления сообщений.

Отличительной чертой данных моделей *Comet*-серверов является то, что, благодаря многопоточности, выполнено разнесение задач *Comet*-сервера по различным потокам исполнения, таким образом, позволяя снизить нагрузку и увеличить сопротивляемость *Comet*-сервера перед атаками типа «отказ в обслуживании». Поскольку высокая нагрузка, генерируемая злоумышленником, реализующем такую атаку на точке входа данных в *Comet*-сервер, обслуживается отдельнымпотоком исполнения, это отразится только на данном потоке,не влияя на прочие потоки исполнения. В результате *Comet*-сервер вместо полного отказа в обслуживании допускает только отказ части своего функционала.

Для каждого типа архитектуры описаны и проиллюстрированы способы и порядок взаимодействия между элементами, см. рисунок 2 на примере многопоточной модели.

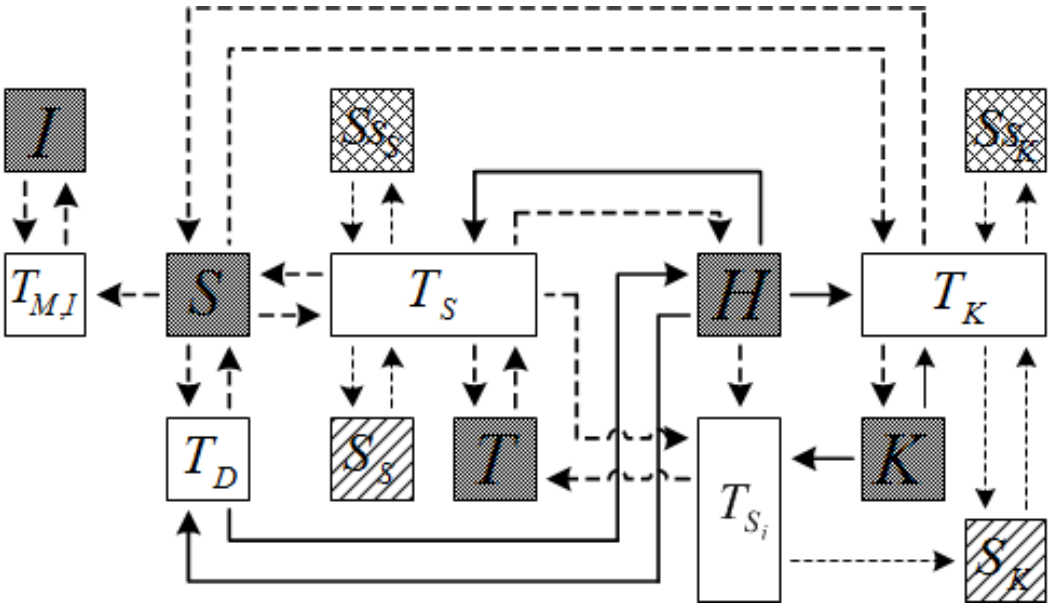

Рисунок 2 – Многопоточная модель с выделением потока под сообщение.

Примечание:

Ss_K - слушающий сокет для клиентских подключений;

Ss_S - слушающий сокет для серверных подключений;

S_K - сокет подключенного клиента;

S_S - сокет подключенного сервера;

Структуры в памяти:

K – структура в памяти, хранящая записи обо всех подключенных клиентах;

H – структура в памяти, хранящая записи об передаваемых сообщениях;

Ch – структура в памяти, хранящая записи об передаваемых сообщениях

в хронологическом режиме;

S– структура в памяти, хранящая информацию сервисного характера;

I – управляющий интерфейс;

T_i – поток исполнения,

где i – может принимать следующие значения:

M – для главного потока;

I – для потока обслуживающего интерфейс управления;

K – для потока обслуживающего клиентские подключения;

S – для потока обслуживающего серверные подключения;

D – для сервисного потока обслуживающего Comet-сервер;

Ch – для рассылающего сообщения в хронологическом режиме;

Правила доступа к данным для потоков:

сплошная тонкая линия - длительный доступ к данным в блокирующем режиме;

пунктирная тонкая линия - недлительный доступ к данным в блокирующем режиме;

сплошная полужирная линия - длительный доступ к данным в не блокирующем режиме;

пунктирная полужирная линия - длительный доступ к данным в не блокирующем режиме.

Произведен поиск таких параметров *Comet*-сервера, которые наиболее полно отражают производительность и эффективность его работы.

В третьей главе описана реализация протокола*Clive*, отражающего предложенные алгоритмы по повышению уровня конфиденциальности данных, и пример *Comet*-приложения работающего по данному протоколу. Отмечено, что реализация протокола *Clive* может основываться как на протоколе прикладного уровня *WebSocket*, так и на протоколе сетевого уровня *TCP*, что позволяет строить полноценные*Comet*-приложения.

Предложены реализации клиента и сервера *Comet*-приложения, выполненных на скриптовых языках программирования *JavaScript* и *PHP*, соответственно, с использованием методологии объектно-ориентированного программирования. Показаны примеры построения *Comet*-приложения. Помимо основного кода клиентской части приложения был разработан код специального *flash*-элемента, реализующего передачу информации из среды выполнения клиента в сеть, а также, в обратном направлении, посредством *TCP* протокола и механизма, так называемых *flash*-сокетов, а основной код клиента дополнен соответствующими механизмами осуществляющими управление передачей информации.

Приведены практические рекомендации по выбору защитных алгоритмов для защиты информации, передаваемой в *Comet*-приложении, для тех разработчиков, которые не имеют заранее установленных требований к безопасности и руководствуются скорее скоростью работы защитных алгоритмов при легкости внедрения и заданных показателях надежности.При выработке реко-

мендаций сделан акцент на возможность применения таких алгоритмов в клиенте приложения, выполняемых в среде *JavaScript*. При этом с целью определения наиболее быстрого алгоритма при заданных равных параметрах надежности было проведено имитационное моделирование непредсказуемой передачи данных через *Comet*-сервер. Таким образом,из алгоритмов хеширования наиболее производительным оказался алгоритм *MD5*, превзошедший по скорости на 26% своего лучшего конкурента *SHA-1* и на 245% худшего *RIPEMD-160*. Из алгоритмов симметричного шифрования был выбран алгоритм *Rabbit*, который при заданной длине ключа в 128 бит, показал лучшую производительность, более чем в 8 раз у алгоритма *AES* и более чем в 23 раза у алгоритма *3DES*. Полученные результаты представлены на рисунке 3 и рисунке 4.

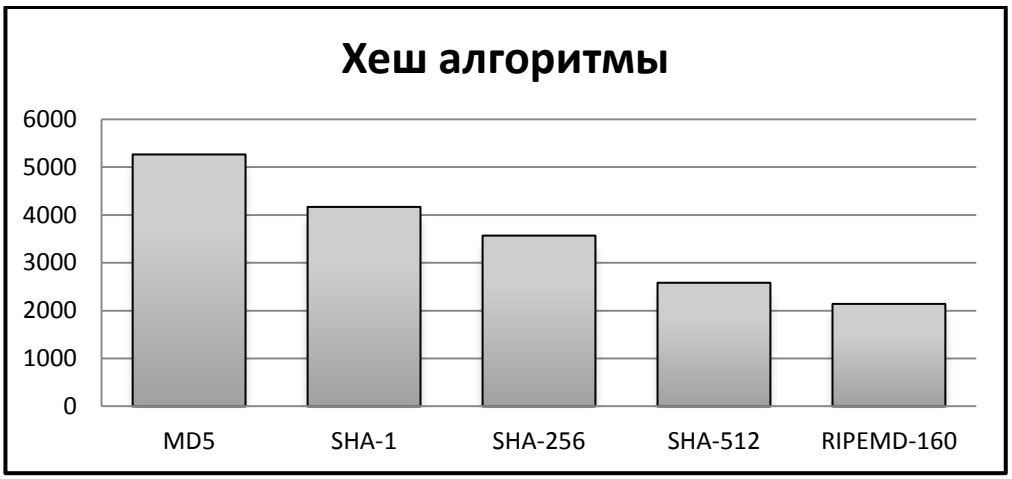

Рисунок 3 – Сравнение хеш-алгоритмов

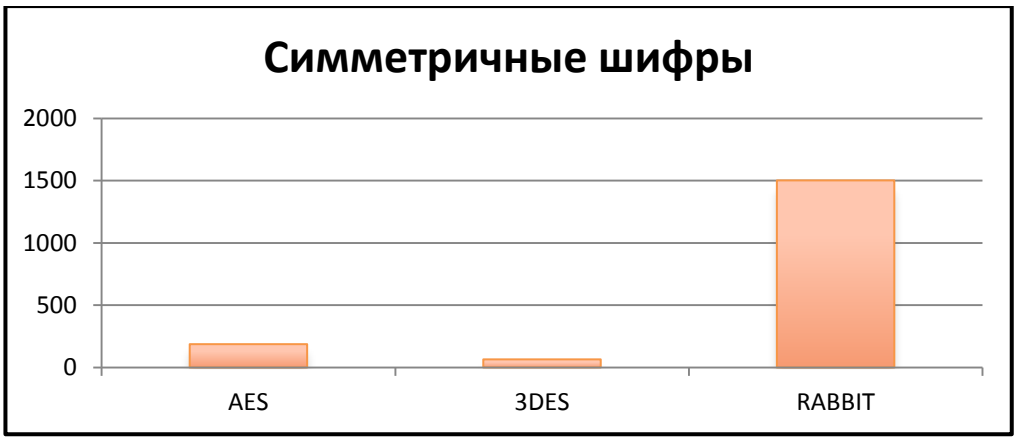

Рисунок 4 – Сравнение симметричных шифров

Обоснован выбор моделей архитектуры *Comet*-сервера (1 и 6, для приложений с малой или дозированной нагрузкой и для приложений с большой или не дозированной нагрузкой, соответственно) для реализации, а также языка программирования и семейства операционных систем. Реализация выполнялась под два типа операционных систем *Windows* и *UNIX*, под их 32-х и 64-х битные версии, при этом в качестве языка программирования использовался компилируемый язык программирования*C++*. Описаны общие аспекты организации программ и алгоритм инициализации. Приведено описание используемых при-

емов организации структур данных в памяти, организации взаимодействия вычислительных потоков и использования механизмов их синхронизации, обоснован выбор данных приемов и механизмов, даны практические замечания по их применению.

Проведена оценка, согласно разработанной методике оценки, при которой выполнялась имитация работы, выбранных моделей архитектуры *Comet*-сервера, реализованных в виде исполняемых приложений, с дозированной и «стрессовой» нагрузкой. Для моделирования такой нагрузки было разработано программно-инструментальное средство тестирования, которое позволяет имитировать большое число клиентских подключений и большой объем траффика на серверном канале связи. При этом, тестирование реализаций выполнялось в два этапа, для сообщений малого и большого объема соответственно, причем, на каждом этапе имитировалась как широковещательная рассылка сообщений, так и узковещательная.

При проведении эксперимента замерялись такие показатели доступности информации, как время, затрачиваемое на рассылку сообщения, а также процент потерь сообщений при отправке. Данные показатели замерялись для каждого реализованного типа архитектуры, а затем вычислялось отношение показателей одного типа, чтобы определить уровень превосходства одной модели над другой по каждому показателю. Отношениеi-го показателяδ_i вычислялось по формуле:

$$\delta_i = \frac{A_i - B_i}{B_i} * 100\%, \tag{19}$$

где $A_i - i$-ый показатель однопоточной модели;

$B_i - i$-ый показатель многопоточной модели.

Из данной формулы заметно, что положительное значение δ соответствует более низкому значению выбранного показателя многопоточной модели по отношению к однопоточной, а отрицательное наоборот. Для однозначного принятия решения о том, какая модель обеспечивает более высокую доступность информации, был введен суммарный показатель доступности информации для конкретной модели$\delta_{\text{сумм.}}$:

$$\delta_{\text{сумм.}} = \frac{\sum_n^{i=1}\left(\frac{A_i - B_i}{B_i} * 100\%\right)}{n}, \tag{20}$$

Результаты испытаний и значения всех показателей доступности для каждой модели приведены в таблице 1.

Таблица 1 – Показатели доступности информации для реализованных моделей

Замеряемый показатель	Тестируемая модель		Отношение показателей, %
	Однопоточная (A)	Многопоточная (B)	
Широковещательная рассылка сообщений длиной 100 байт			
Потери отправленных сообщений, шт	392	18	2078
Время рассылки одного сообщения, мс	9,293	10,468	-11
Минимальное время получения клиентом сообщения, мс	2,920	2,163	35
Максимальное время получения клиентом сообщения, мс	245,392	250,756	-2
Узковещательная рассылка сообщений длиной 100 байт			
Потери отправленных сообщений, шт	0	0	-
Время рассылки одного сообщения, мс	0,486	0,591	-18
Минимальное время получения клиентом сообщения, мс	2,871	2,008	43
Максимальное время получения клиентом сообщения, мс	5,544	4,620	20
Широковещательная рассылка сообщений длиной 100000 байт			
Потери отправленных сообщений, шт	60398	0	-
Время рассылки одного сообщения, мс	12,892	10,391	24
Минимальное время получения клиентом сообщения, мс	728,467	5,423	13333
Максимальное время получения клиентом сообщения, мс	41618,540	572,626	7168
Узковещательная рассылка сообщений длиной 100000 байт			
Потери отправленных сообщений, шт	0	0	-
Время рассылки одного сообщения, мс	1,547	0,755	105
Минимальное время получения клиентом сообщения, мс	15,107	11,824	28
Максимальное время получения клиентом сообщения, мс	534,742	315,754	69
Средний суммарный показатель доступности			**1759**

Полученные в ходе тестирования результаты отражают правильность

выбора моделей архитектуры *Comet*-серверов для подставленных задач. Обе модели в целом успешно справляются со своей задачей, показывая малый процент потерь сообщений (менее 0,0004% для модели 1, и менее 0,0001% для модели 6). При широковещательной рассылке сообщений малого объема модель 1 показывает на 10% лучшую производительность, однако эффективность моделей при этом остается примерно равной. При узковещательной рассылке сообщений малого объема модель 6 начинает демонстрировать лучшие показатели производительности и эффективности, превосходящие аналогичные показатели модели 1 на 20% и 32% соответственно.

Подготовленность модели 6 становится наиболее заметна в условиях высокой нагрузки при увеличении объема сообщений в 1000 раз.При широковещательной рассылке за отведенное время эксперимента модель 1 произвела доставку только 40%, что соответственно составило потери равные 60%. Модель 6 произвела доставку всех сообщений за отведенное время без потерь. При этом такие параметры производительности, как минимальное и максимальное время между отправкой сообщения сервером и доставкой сообщения клиенту у модели 6 меньше в 134 и 72 раза соответственно, чем у модели 1.Дополнительно, разность между данными параметрами модели 6 меньше в 80 раз, чем у модели 1. При узковещательной рассылке сообщений большого объема разрыв в производительности и эффективности между моделями сокращается, но при этоммодель 6 превосходит модель 1 по всем параметрам почти в 2 раза.

В заключении сформулированы основные результаты работы.

В приложениях приведена техническая спецификация разработанного протокола *Clive*, а также, программный код примеров клиента и сервера *Comet*-приложения на языках *JavaScript* и *PHP*, кроме того,приведен программный код программно-инструментального средства для тестирования *Comet*-сервера. Приведены документы, подтверждающие внедрение.

ОСНОВНЫЕ РЕЗУЛЬТАТЫ И ВЫВОДЫ РАБОТЫ

1. Адаптирован известный метод избирательного разграничения доступа применительно к данным, пересылаемым в *Comet*-приложении через *Comet*-сервер, для существующих технологий передачи данных по инициативе вебсервера, таких как протокол *WebSocket*, *flash*-сокеты или *java*-сокеты, что позволяет реализовать механизм разграничения доступа на уровне *Comet*-сервера и, тем самым, реализовать принятую политику разграничения доступа на технологическом уровне, чтобы исключить получение клиентом той категории данных, к которой он не имеет доступа. Показано, что предлагаемая адаптация совместима со стандартизированным протоколом *WebSocket*, и позволяет внутри одного канала связи по данному протоколу реализовать несколько каналов для различных категорий информационных сообщений, доступ к которым требует контроля.

2. Разработан новый алгоритм защиты идентификационных данных клиента в *Comet*-приложении на основе метода криптографического преобразования передаваемой информации, а именно, применения симметричных шифров,

что позволяет исключить несанкционированную модификацию или подмену вышеуказанных данных клиента, скомпрометированного злоумышленником.

3. Предложен новый алгоритм повышения конфиденциальности данных передаваемых через *Comet*-сервер в *Comet*-приложении, совместимый с предлагаемым алгоритмом разграничения доступа, на основе симметричного шифрования и цифровых подписей с использованием хеш-функций, который позволяет исключить несанкционированную модификацию и/или несанкционированное ознакомление с передаваемыми данными через скомпрометированный *Comet*-сервер. Проведена оценка производительности различных шифров (*3DES, AES, RABBIT*) при равных показателях защищенности, а также, проведена оценка производительности различных хеш-функций (*MD5,SHA-1,SHA-256,SHA-512,RIPEMD-160*), реализация которых совместима с данным алгоритмом и может быть использована в предлагаемом алгоритме. По результатам эксперимента, основанного на имитации непредсказуемой передачи сообщений через Comet-сервер, наиболее высокую производительность показал шифр *RABBIT* и хеш-функция *MD5,* обрабатывающие в среднем 1501и 5264 байт за одну миллисекунду в среде исполнения клиента *Comet*-приложения *JavaScript*.

4. Разработан ряд архитектур *Comet*-серверов, на основе положений теории многопоточного программирования, для различных видов задач решаемых *Comet*-сервером, что позволяет повысить значения показателей доступности передаваемых данных и, тем самым, обеспечить большую устойчивость *Comet*-сервера перед атаками вида «отказ в обслуживании». Наиболее востребованные модели, а именно, однопоточная и многопоточная, были реализованы в виде программ на языке*C++*, была произведенаоценка показателей доступности (временные задержки различного типа при доставке сообщений и потери сообщений) информации для каждой из них в ряде экспериментов с имитацией широковещательной и узковещательной рассылки сообщений различной длины. По результатам эксперимента многопоточная модель в 17,5 раз превзошла однопоточную по интегральному показателю доступности информации.

Перспективы дальнейшей разработки темы.Перспективными направлениями для дальнейших исследований являются разработка методов и алгоритмов защиты потоковой передачи данных через *Comet*-сервер с применением потоковых шифров на клиентской и серверной стороне *Comet*-приложения, а также, разработка алгоритмов обеспечения целостности передаваемых данных для потоковой передачи через *Comet*-сервер.

ОСНОВНЫЕ ПУБЛИКАЦИИ ПО ТЕМЕ ДИССЕРТАЦИИ
В рецензируемых журналах из списка ВАК

1. Управление преобразованием информации и разграничением доступа для устройств обмена систем управления на примере модели Comet / М.В. Тюлькин, Е.Л. Кротова, Л.Н. Кротов, И.В. Капгер // Глобальный научный потенциал. 2012. № 4(13). С. 83-88.

2. Разработка методов управления преобразованием информации и разграничением доступа для устройств обмена систем управления на примере модели Comet. Часть 1 – протокол Clive / М.В. Тюлькин, Е.Л. Кротова,

Л.Н. Кротов, И.В. Капгер // Перспективы науки. 2012. № 4(31). С. 68-71.

3. Разработка методов управления преобразованием информации и разграничением доступа для устройств обмена систем управления на примере модели Comet. Часть 2 – Реализация / М.В. Тюлькин, Е.Л. Кротова, Л.Н. Кротов, И.В. Капгер // Естественные и технические науки. 2012. № 2(58). С. 348-351.

4. Разработка архитектуры и организация информационных потоков в Comet-серверах для Web-приложений модели Comet со схемой взаимодействия WebSocket. Описание Comet-сервера / М.В. Тюлькин, И.В. Капгер, Е.Л. Кротова, Л.Н. Кротов // Вестник Ижевского государственного технического университета. 2012. № 2. С. 150-153.

5. Разработка методов анализа уязвимостей, выбор и реализация криптографической защиты трехстороннего информационного обмена в Web-приложениях модели Comet / М.В. Тюлькин, И.В. Капгер, Е.Л. Кротова, Л.Н. Кротов // вопросы защиты информации. 2012. № 2. С. 6-12.

6. Разработка архитектуры и организация информационных потоков в Comet-серверах для web-приложений модели Comet со схемой взаимодействия WebSocket. Модели архитектуры Comet-сервера / М.В. Тюлькин, И.В. Капгер, Е.Л. Кротова, Л.Н. Кротов // Вестник Ижевского государственного технического университета. 2012. № 4. С. 118-120.

7. Разработка и реализация метода санкционированного запуска и функционирования защищаемого программного продукта в потенциально враждебном программно-аппаратном окружении / М.В. Тюлькин, И.В. Капгер, Е.Л. Кротова, Л.Н. Кротов // вопросы защиты информации. 2012. № 4. С. 16-20.

8. Аспекты применения симметричных шифров в клиенте Web-приложения с использованием технологии Comet / М.В. Тюлькин, И.В. Капгер, Е.Л. Кротова, Л.Н. Кротов // вопросы защиты информации. 2013. № 2. С. 10-14.

В других изданиях

9. Аспекты проверки подлинности сообщений клиентом Web-приложения с использованием технологии Comet / М.В. Тюлькин, И.В. Капгер, Л.Н. Кротов, Е.Л. Кротова // Технические науки – от теории к практике: материалы IX междунар. заочн. науч.-практ. конф., 17 апр. 2012 г. - Новосибирск. 2012. С. 30-36.

10. Проблема масштабирования Web-приложений, использующих Comet-сервера, её особенности и попытка решения / М.В. Тюлькин, И.В. Капгер, Е.Л. Кротова, Л.Н. Кротов // Технические науки – основа современной инновационной системы: материалы Iмеждунар. науч.-практ. конф., 25 апр. 2012 г. - Иошкар-Ола. 2012. С. 123-125.

11. ОднопотоковыйComet-сервер «Vortex» / М.В. Тюлькин // Свидетельство о государственной регистрации программы для ЭВМ №2012616041, опубликовано 02.07.2012.

12. МногопотоковыйComet-сервер «Vortex-Pro» / М.В. Тюлькин // Свидетельство о государственной регистрации программы для ЭВМ №2012616044, опубликовано 02.07.2012.

Диссертант М.В. Тюлькин