Московский государственный университет имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Кафедра автоматизации систем вычислительных комплексов

Нгуен Кхань Кхуен

Доказательство с нулевым разглашением для взаимной аутентификации пользователей группового чата

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Научный руководитель:

к.ф.-м.н., с.н.с. профессор

Д. Ю. Гамаюнов

Москва, 2018

Оглавление

Аннотация 4

1 Введение 5

1.1 Цель работы 9

1.2 Постановка задачи 9

1.3 Определения 10

1.4 Протокол mpOTR - фазы и свойства 10

1.5 Обзор протоколов с нулевым разглашением 12

1.5.1 Критерии сравнения протоколов аутентификации 12

1.5.2 Протокол Файге – Фиата – Шамира 12

1.5.3 Протокол Шнорра 13

1.5.4 Протокол Миллионеров-Социалистов 13

1.5.5 Вывод 13

2 Исследование и построение решения задачи 14

2.1 Протокол Миллионеров-Социалистов 14

2.1.2 Описание протокола Миллионеров-социалистов 15

2.1.3 Безопасность протокола SMP 17

2.1.4 Реализация протокола SMP 18

2.1.5 Вывод 19

2.2 Круговая аутентификация 20

2.2.1 Описание алгоритма круговой аутентификации 21

2.2.2 Обработка сообщений о результате аутентификации 22

2.2.3 Выбор цели для инициирования SMP 24

2.2.4 Безопасность. Возможные атаки. 26

3 Описание практической части 27

3.1 Реализация 27

3.2 Тестирование 28

4 Заключение 30

5 Список литературы 31

Аннотация

Данная работа посвящена исследованию и разработке системы взаимной аутентификации пользователей для приватных IM[1]-конференций в сети Интернет. В работе предложен групповой алгоритм взаимной аутентификации на основе известного алгоритма с нулевым разглашением – протокола Миллионеров Социалистов (SMP – Socialist Millionaire Protocol). В работе также сделана реализация предложенного протокола и проведено экспериментальное исследование его характеристик, которое демонстрирует применимость алгоритма для систем группового обмена сообщениями.

[1] IM, Instant Messaging – система обмена мгновенными сообщениями, а также сам процесс обмена мгновенными сообщениями между участниками некоторой коммуникации.

1 Введение

Интернет коммуникации путем обмена мгновенными сообщениями на сегодняшний день является самым распространённым способом общения, формируя новую сферу информационного взаимодействия. Общение в сети имеет свои особенности в отличие от традиционного прямого общения в реальной жизни. Главной особенностью интернет общения является физическое отсутствие участников коммуникации, что приводит к проблеме обеспечения безопасности данных коммуникаций.

Любое общение подразумевает обмен некой информации, а так как данная информация перед тем как попасть к адресату путешествует по сети Интернет, велика вероятность утечки данных. Проектировщики первых протоколов общения в сети Интернет нечасто задумывались о том, что передаваемые по сети данные необходимо защищать от злоумышленников, в итоге большая часть путешествовавшей по сети информации передавалась в формате открытого текста. Но чем больше пользователей подключались к сети Интернет, чем больше компаний начинали использовать сеть Интернет в своем бизнесе, тем более остро вставали вопросы обеспечения конфиденциальности передаваемой по сети информации. Также в последние несколько лет люди почувствовали угрозу скрытного прослушивания коммуникаций, что вызвано серией недавних публикаций о системах слежения уровня государства, в том числе публикации документов АНБ США Эдвардом Сноуденом [1]. Это подтолкнуло многих исследователей к построению протоколов защищенных коммуникаций, которые, во-первых, обеспечат конфиденциальность информации, переданной во время коммуникации, даже после того, как коммуникация завершена, во-вторых, обеспечат надежную защиту коммуникаций от прослушивания либо защиту пользователей-участников коммуникации от идентификации.

Если описать указанные цели более формально, то мы получим определения двух свойств безопасности коммуникаций, реализация которых в протоколах защищенных коммуникаций наиболее актуальна на сегодняшний день – это свойства совершенной секретности в будущем и отказуемости. Совершенная секретность в будущем – это свойство сохранения конфиденциальности информации, переданной во время коммуникации, даже в случае компрометации долговременных ключей шифрования участников коммуникации после

завершения этой коммуникации. Отказуемостью называется свойство коммуникаций, которое означает, что невозможно математически однозначно доказать ни участие конкретного пользователя в коммуникации, ни авторство сообщений, переданных в рамках данной коммуникации. То есть участник коммуникации может отрицать свое в ней участие или то, что данный участник был автором каких-либо сообщений, переданных в рамках данной коммуникации.

В работе Коростелевой М. В. «Исследование протокола криптографически защищенных групповых коммуникаций с функцией отказуемости» [2] были разработаны и исследованы основные фазы протокола защищенных групповых коммуникаций на основе протокола, целью которого обеспечить следующие свойства безопасности – конфиденциальность, аутентичность участников, аутентичность сообщений, целостность передачи информации, согласованность состояний транскрипта, совершенная секретность в будущем и отказуемость.

Разработка полного варианта протокола защищенных групповых коммуникаций является целью команды разработчиков и исследователей на факультете ВМК Московского государственного университета имени М. В. Ломоносова научного семинара по прикладной криптографии[2]. В рамках данной работы наше внимание будет сосредоточено на разработке и исследовании системы аутентификации.

Описание протокола приведено позднее, а пока отметим лишь то, что протокол mpOTR построен на основе криптосистемы с открытым ключом, где каждый участник обладает парой открытых и закрытых ключей. Открытый ключ – известен всем пользователям системы и необходим для проверки электронной подписи, сделанную закрытым ключом, которым обладает только владелец ключа. Основной уязвимостью данной системы является возможность подмены открытого ключа одного из участников на другой открытый ключ с известной противнику секретной половиной этого ключа (атака человека по середине[3]) при передаче пользователями ключей друг другу. Так в 2007 году Оливер Гоффарт опубликовал плагин mod_otr [3] для сервиса Jabber, автоматически проводящую данную атаку на пользователей, не проверяющих открытые ключи своих собеседников. Плагин позволяет атакующему, выдавая

[2] Научный семинар по прикладной криптографии проводится Лабораторией безопасности информационных систем совместно с кафедрой Информационной безопасности факультета ВМК МГУ имени М. В. Ломоносова.
[3] англ. Man In The Middle, MITM.

себя за друга человека, узнавать содержание зашифрованных сообщений, отправляемых данному участнику, и отправлять сообщения от него. Таким образом нарушается основное свойство конфиденциальности коммуникации.

Для защиты от атак типа человека по середине необходимо проводить аутентификацию открытых ключей. В данном случае аутентификация – это доказательство проверяющему того, что доказывающий обладает открытым ключом проверяющего (то есть ключ не был подменен при ее передаче). Однако если доказывающий будет передавать ключи в открытом виде проверяющему для проверки, совпадает ли имеющийся у доказывающего ключ проверяющего с реальным ключом проверяющего, возможна подмена переданного доказывающим ключа, а проверка не обнаружит данной подмены. Пользователи, аутентифицирующие открытые ключи собеседников, должны быть уверены в том, что эти ключи принадлежат ожидаемым им людям.

Публичные ключи созданы таким образом, что имеют сами по себе защиту от подмены. У каждого ключа есть свой уникальный *отпечаток*, поэтому получив ключ из ненадежного источника, можно запросить у отправителя отпечаток, используя альтернативный способ связи, и сравнить его с отпечатком полученного ранее ключа. Тогда достоверность ключа будет равна достоверности выбранного способа связи, а безопасность – точности проверки ключей.

В данной работе мы рассмотрим методы аутентификации, основанные на Доказательстве с нулевым разглашением, не требующего сравнения пользователями ключей. *Доказательством с нулевым разглашением* (ДНР) называется интерактивный вероятностный протокол, который позволяет убедиться одной из сторон (Проверяющему) в достоверности некого утверждения, без предоставления второй стороны (Доказывающей) доказательств о самом утверждении, и обладающий свойствам полноты, корректности и нулевого разглашения. Свойство *полноты* означает, что если утверждение действительно верно, то Доказывающий убедит в этом Проверяющего с любой наперед заданной точностью; *корректность*: если утверждение неверно, то с пренебрежимо малой вероятностью Доказывающий сможет убедить Проверяющего; *нулевое разглашение*: если утверждение верно, то любой, даже «нечестный», Проверяющий не узнает ничего кроме самого факта, что утверждение верно.

Задачу аутентификации можно сформулировать следующим образом: показать Проверяющему, что Доказывающий обладает верным ключом (ключом Проверяющего). Использование ДНР позволяет Доказывающему доказать, что тот обладает ключом Проверяющего, не раскрывая его, тем самым защищая от подмены переданного ключа.

Работа состоит из нескольких разделов. В разделе 2 приводится подробное описание предложенного протокола взаимной групповой аутентификации на основе протокола Миллионера-Социалиста, который был выбран в результате исследования (глава 1.5). Раздел 3 посвящен описанию практической части – реализации и экспериментальному исследованию. С заключением и основными результатами проведенных исследований можно познакомится в заключительном разделе 4.

1.1 Цель работы

Целью данной работы является разработка алгоритма взаимной аутентификации пользователей группового криптографически защищенного чата на основе группового варианта доказательства с нулевым разглашением.

1.2 Постановка задачи

1. Провести исследование и сравнительный анализ протоколов доказательства с нулевым разглашением и оценить их применимость для взаимной аутентификации пользователей группового криптографического чата.

2. Предложить алгоритм групповой взаимной аутентификации на основе доказательства с нулевым разглашением и протокол на его основе.

3. Реализовать предложенный алгоритм для модельной системы на основе mpOTR.

4. Провести экспериментальное исследование предложенного алгоритма.

1.3 Определения

Коммуникации. Коммуникациями между двумя или несколькими людьми (пользователями, участниками коммуникации) будем называть общение между этими людьми в сети Интернет по одному из протоколов мгновенного обмена сообщениями.

Аутентификация пользователей - процесс проверки нескомпроментированности собеседников конференции и их публичных ключей.

Модель нарушителя. В системах обеспечения безопасности коммуникаций с помощью обмена мгновенными сообщениями нарушитель является глобальным неограниченным в ресурсах человеком, который может перехватывать трафик и контролировать физические устройства: один или несколько узлов, участвующих в процессе коммуникации, могут находится под контролем нарушителя во время коммуникации (нарушитель может выдавать себя за другого человека); любой узел в некоторый момент времени после завершения коммуникации может находится под контролем нарушителя.

Отпечаток публичного ключа (англ. fingerprint) - это последовательность байтов идентифицирующая более длинный публичный ключ, созданная применением криптографически стойкой хеш-функции к нему и используемая для упрощения управления ключами.

1.4 Протокол mpOTR - фазы и свойства

Разрабатываемый протокол групповых коммуникаций должен удовлетворять следующим требованиям:

- *Конфиденциальности* - защита сообщений от третьих лиц (невозможность их прочтения)
- *Совершенной секретности в будущем* - невозможность доказать авторство сообщений после коммуникации
- *Аутентичности участников беседы* - подтверждение личностей собеседников
- *Аутентичности сообщений* - невозможность отказа от авторства сообщений внутри чата

- *Отказуемости* - невозможность доказать участие пользователей и содержание беседы после коммуникации ни для нарушителя ни для пользователей
- *Целостности передачи информации* - обеспечение получения пользователями сообщений в первоначальном виде, котором было передано адресатом.
- *Согласованности состояний транскрипта* у всех участников данной IM-конференции - обеспечение получения правильной последовательности сообщений.

Основные фазы протокола защищенных групповых коммуникаций с функцией отказуемости:

1. Фаза установки канала

На этой фазе устанавливается небезопасное соединение между пользователями. Протокол использует децентрализованную архитектуру, где все участники беседы равнозначны, поэтому атака на один из узлов не приведет к нарушениям требуемых свойств протокола.

2. Фаза аутентификации и обмена ключами

На этом этапе происходит установка шифрованного канала. Пользователи идентифицируют друг друга по долговременным ключевым парам, генерируют пару приватных и публичных ключей для подписи будущих сообщений, устанавливают общий групповой ключ для шифрования, идентификатор сессии, аутентифицируют временные ключи с помощью долговременных и обмениваются публичными долговременными и временными ключами.

3. Фаза коммуникации

Для *отправки* сообщения пользователи набирают текст сообщения, который потом шифруется групповым ключом. К шифротексту добавляется идентификатор сессии и подпись, сгенерированная приватным временным ключом. Результат отправляется всем участникам.

Для *получения* сообщения производится проверка подписи полученного сообщения по временному публичному ключу отправителя, который хранится у каждого пользователя, для проверки авторства сообщения и неизменности его содержания. Далее проверяется идентификатор сессии, если он принадлежит данной сессии, то сообщение расшифровывается групповым ключом и пользователь получает текст исходного сообщения.

На данном этапе также обеспечивается целостность транскрипта коммуникации, путем повторной отправки потерянных сообщений.

4. Фаза завершения

На фазе завершения происходит удаление из оперативной памяти группового ключа и публикация всех временных публичных ключей для поддержания свойства отказуемости.

1.5 Обзор протоколов с нулевым разглашением

Доказательство с нулевым разглашением (ДНР) – это протокол, позволяющий убедить одного субъекта в том, что первый субъект обладает определенной информацией, не раскрывая её. Рассмотрим несколько решений, существующих и используемых сегодня, чтобы получить представление о разнообразии протоколов ДНР [13]. Один из описанных ниже протоколов – протокол Миллионеров-Социалистов, послужил основой для разработок протокола аутентификации в рамках данной работы.

1.5.1 Критерии сравнения протоколов аутентификации

1. Вычислительная эффективность – число модульных умножений необходимых для выполнения протокола для обеих сторон;
2. Коммуникационная эффективность – данное свойство отражает количество сообщений, необходимую для осуществления аутентификации;
3. Отсутствие третьей стороны – примером третьей стороны может служить доверенный сервер распределения симметричных ключей или сервер, реализующий дерево сертификатов открытых ключей;
4. Количество участников – число аутентифицируемых пользователей за один проход протокола.

1.5.2 Протокол Файге – Фиата – Шамира

Протокол стал первым практическим протоколом идентификации и считается лучшим доказательством подлинности с нулевым разглашением. Безопасность протокола основывается на сложности извлечения квадратного корня из числа по модулю большого числа с неизвестным разложением на простые множители. На этапе предвычислений происходит выбор доверенным центром параметров безопасности – k, t, далее генерация секретов участниками и выполнения t итераций доказательства, где каждое состоит из 4

последовательных шагов. Требует передачи *3t* сообщений *и 3t модульных умножений.* Для достижения хорошей безопасности достаточно брать *t = 4, k = 5*. Протокол позволяет аутентифицировать одного пользователя за один прогон.

Существуют модификации протокола, позволяющее обойтись без доверенного центра [8], однако требования *t* итераций и большого числа чередующихся пересылок сообщений прежде чем проверяющая сторона сможет убедиться в идентичности доказывающей стороны с достаточной степенью вероятности является недостатком данного протокола. [9]

1.5.3 Протокол Шнорра

Один из самых используемых протоколов аутентификации, являющийся основой для стандартов Белоруссии и Южной Кореи. Протокол позволяет доказать Проверяющему знание секрета за 3 этапов: выбор параметров протокола, выбор параметров каждого пользователя и выполнение доказательства (4 шага и 3 сообщения) и требует наличия доверенного центра для выработки параметров. Протокол использует порядка 30 модульных умножения. Безопасность протокола основана на сложности дискретного логарифмирования и считается обладающим свойством нулевого разглашения, однако доказательства данного свойства до сих пор нет. [10]

1.5.4 Протокол Миллионеров-Социалистов

Протокол взаимной аутентификации, используемый в криптографическом протоколе секретных коммуникаций для 2 участников – OTR [4], не требует наличия доверенного центра и предвычислений. Состоит из одной итерации в 5 шагов и требует передачи 4 сообщений, 7 модульных умножений. Безопасность протокола основана на сложности вычисления дискретного логарифма. [5]

1.5.5 Вывод

Протокол Миллионеров-Социалистов является наиболее эффективным по числу операций и по нагрузке среды передачи сообщений, при этом не требует наличия доверенного центра для своей работы. Однако допускает аутентификацию только двух пользователей. В результате исследования не было найдено протокола групповой взаимной аутентификации со свойством нулевого разглашения, что подтверждает актуальность проводимой нами работы.

2 Исследование и построение решения задачи

2.1 Протокол Миллионеров-Социалистов

Протокол Миллионеров-Социалистов[4] (SMP) основывается на решении следующей задачи: два подпольных миллионера хотят выяснить, кто из них богаче, но они не намериваются раскрывать точную сумму своего благосостояния. С математической точки зрения, проблема заключается в сравнении двух чисел, не раскрывая их значений друг другу. В 1996 году в работе Маркуса Якобссона и Моти Юнга частный случай, в котором Алиса и Боб хотят выяснить, равны ли их состояния, был назван задачей социалистов-миллионеров [5].

Очевидным решением является вмешательство третей стороны, которая бы получила два значения и сравнила их. Но в данном подходе третья сторона и является основной уязвимостью, из-за возможности ее компрометации, а секретные значения и их сравнение будут доверены ей.

Эффективное решение задачи социалистов-миллионеров, не подразумевающие наличие третей стороны, было предложено в 2001 году в работе Фабрис Будо, Берри Шонмейкерса и Жака Траоре [6]. Протокол вместо доходов сравнивает общий секрет[5] пользователей, который содержит отпечатки публичных ключей и “пароль” - общую приватную информацию, которой обладают собеседники. О пароле можно договориться заранее вне данной коммуникации или же в процессе коммуникации установить пароль как ответ на вопрос, который знают участники.

Пользователи на фазе коммуникации, получив публичные ключи друг друга, могут запросить проверку подлинности полученных ключей (то есть принадлежность полученного ключа ожидаемому человеку), и после ввода паролей в работу вступает протокол SMP. Каждый участник формирует секрет, который и будет сравниваться. Секреты будут равны только в том случае, если каждый обладает достоверными открытыми ключами других.

Структура общего секрета выглядит следующим образом:

$$sha256(0x01,\ pk1,\ pk2,\ \ sid,\ passphrase).$$

0x01 – версия протокола, *pk1* – это отпечаток долговременного открытого ключа пользователя, инициирующего протокол; *pk2* - это отпечаток долговременного

[4] Англ. Socialist Millionaire Protocol (SMP)
[5] подробнее SMP раздел 4.2 Общий секрет

открытого ключа пользователя, принимающего SMP; *sid* - идентификатор сессии, выработанный каждым участником значение, подтверждающие актуальность сессии; *passphrase – пароль* - строковая константа, о которой участники заранее договорились.

Ключевыми атрибутами секрета являются отпечатки ключей и пароль. Без пароля протокол бы был уязвим атаке человека по середине. Допустим такую ситуацию, когда в секрете нет пароля, тогда нарушитель имеет возможность в процессе его передачи для аутентификации подменить секреты на свои, которые бы включали его открытый ключ, добиваясь равенства секретов и, таким образом, обходя аутентификацию. Если же в секрете будут присутствовать пароль, нарушитель не сможет, перехватив секрет, подобрать такой, чтобы секреты были равны, потому что не знает пароль, а в силу свойств протокола – нулевого разглашения - из перехваченного секрета нельзя узнать пароль, и даже сами пользователи не знают секреты друг друга, если они не равны. Так же проверив равенство паролей, пользователь удостоверяется в том, что имеют дело со своим напарником, так как только он знает секрет.

2.1.2 Описание протокола Миллионеров-социалистов

Пусть существует 2 стороны *A* и *B*, обладающие сформулированными секретами $secretA$ и $secretB$ соответственно.

Вычисления проводятся в мультипликативной группе Z_p^*, где p – большое простое число, g – образующая группы. Подробное описание работы протокола представлено на Таблице 1.

Alice: (1 шаг) ● Генерация x_2 и x_3 , где $x_2 != 0;\ x_3 != 0$ ● Вычисления $g_{2a} = g_1^{x_2};\ g_{3a} = g_1^{x3}$, где $g_{2a} != 1;\ g_{3a} != 1$ ● Доказательство ZK: $proof(g_{2a})$ и $proof(g_{3a})$ ● Отправка g_{2a}, g_{3a}
Bob: (2 шаг)

2а

- Получение g_{2a}, g_{3a}
- Проверка ZK: $check(g_{2a})$ и $check(g_{3a})$
- Генерация y_2, y_3, где $y_2 != 0;\ y_3 != 0$
- $g_2 = g_{2a}^{y_2} = g_1^{y_2 * x_2}$

  $g_3 = g_{3a}^{y_3} = g_1^{y_3 * x_3}$
- $P_b = g_3^r$; $Q_b = g_1^r * g_2^{secretB}$
- Доказательство ZK: P_b, Q_b
- Отправка g_{2b}, g_{3b}, P_b, Q_b

2б

- $g_{2b} = g_1^{y_2}$,

  $g_{3b} = g_1^{y_3}$;
- Доказательство ZK: g_{2b} и g_{3b}
- Генерация r из Z

<u>Alice:</u> 3 шаг.

- Проверка ZK: g_{2b} и g_{3b}
- $g_2 = g_{2b}^{x_2} = g_1^{x_2 * y_2}$;

  $g_3 = g_{3b}^{x_3} = g_1^{x_3 * y_3}$
- Проверка ZK: $check(P_b)$, $check(Q_b)$
- $P_a = g_3^s$,

  $Q_a = g_1^s g_2^{secretA}$

  $R_a = (Q_a / Q_b)^{x_3}$

• Доказательство ZK: $proof(P_a), proof(Q_a), proof(R_a)$ • Отправка P_a, Q_a, R_a
Bob: 4 шаг. • Проверка ZK: $check(P_a), check(Q_a), check(R_a)$ • $P_{ab} = P_a / P_b$; $Q_{ab} = Q_a / Q_b$ $R_b = (Q_a / Q_b)^{y_3}$ • Доказательство ZK: $proof(R_b)$ • $R_{ab} = R_a^{y_3}$ • Проверка совпадения секретов: $R_{ab} == P_{ab}$ • Отправка R_b
Alice: 5 шаг • Проверка ZK: $check(R_b)$ • $P_{ab} = P_a / P_b$; • $R_{ab} = R_b^{x_3} = (Q_a / Q_b)^{x_3 * y_3}$ • Проверка совпадения секретов: $R_{ab} == P_{ab}$

Таблица 1 Алгоритм Миллионера-Социалиста

Все передаваемые значения проходят проверку на то, что они были сгенерированы соответственно протоколу с помощью доказательства Шнорра, которое обеспечивает проверку правильности вычислений отправляемых переменных. Участник вместе со сгенерированными значениями отправляет доказательства их валидности - *proof*(x), а принимающий проверяет эти доказательства – *check*(x). Если проверка прошла не успешна, протокол завершается.

Доказательство валидности - proof (x)

1. Генерация r из Z
2. Вычисление $W = g_1^r$; $c = hash(W)$; $D = r - x * c$

3. Отправка - $(g_1^x,\ c,\ D)$

Проверка валидности - check(x)

1. Получение $(g_1^x,\ c,\ D)$
2. Проверка равенства c $==\ h(g_1^D * g_1^{x*c})$

Покажем, что сравнение R_{ab}, P_{ab} равносильно сравнению $secretA,\ secretB$

$R_{ab} = R_a^{y_3} = (Q_a/Q_b)^{x_3*y_3} = (g_1^{s-r} * g_2^{secretA-secretB})^{x_3y_3} = g_3^{s-r} * g_2^{x_3y_3(secretA-secretB)} = P_a/P_b * g_2^{x_3y}$

$= P_{ab} * g_2^{x_3y_3(secretA-secretB)}$

⇨ $R_{ab} ==\ P_{ab} \Leftrightarrow secretA == secretB$

2.1.3 Безопасность протокола SMP

Безопасность протокола Миллионеров-Социалистов доказана в статье[3] и основывается на следующих криптографических предположениях – дискретного логарифмирования, Деффи-Хелмана и DDH (англ. decisional Diffie-Hellman). Предположение дискретного логарифмирования состоит в том, что для группы Z_p, где $g, y \in Z_p$, $g \neq 1$ невозможно вычислить log_py за разумное время; Деффи-Хелмана – для группы Z_p и любых a, b невозможность вычислить п g_{ab}, зная g_a, g_b; DDH - для группы Z_p и любых a, b, c невозможно определить $g^c = g^{ab}$, зная $g,\ g^a,\ g^b,\ g^c$.

2.1.4 Реализация протокола SMP

Протокол реализован на языке Javascript как объект *SMcontext*, использующий доступную в реализации чата библиотеки Cryptico [14] и BigInteger [15].

Аутентичность всех собеседников пользователя указывается в их статусах – Secure или *not Secure*, которая находится справа от имен собеседников. Пользователи, чьи ключи аутентифицированы, имеют статус – Secure (рис. 1).

Рисунок 1 Статус аутентификации

При нажатии на статус участника запускается SMP, которое отправляет сообщение – START_SMP, уведомляющее выбранного пользователя о начале проверки, и пользователям предлагается ввести пароль (Рисунок 2). Далее вызывается функция smpInit, где происходит формирование общего секрета и вызов первого шага протокола SMP – sm_step1 или sm_step2b в зависимости от инициирующего бита, указывающий был ли участник инициатором данной аутентификации.

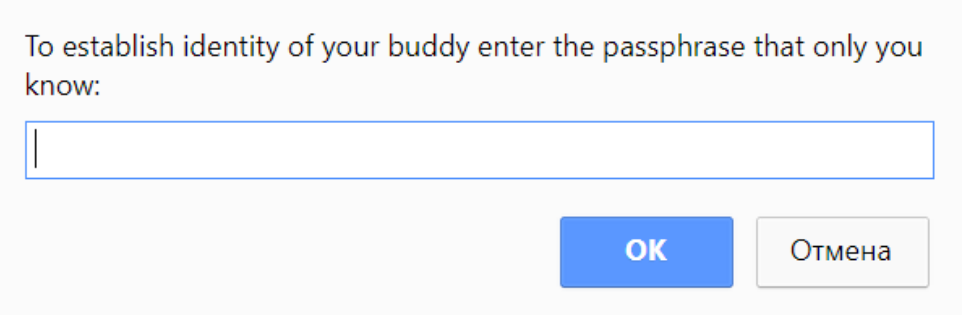


Рисунок 2 Всплывающее окно - Аутентификация участника

Так как аутентификация попарна, то сообщения отправляются не широковещательно, а только имеющему отношение к проверке пользователю. Сообщения, отправляемые протоколом, имеют один из следующих типов – SMP_STEP1, SMP_STEP2, SMP_STEP3, SMP_STEP4. При получении таких сообщений генерируется событие “RECV_SMP”, которое вызывает обработчик сообщения и следующий шага протокола - sm_auth и изменяет поле «следующий ожидаемый шаг» next_expected. По завершению работы протокола пользователям выводится сообщение о результате аутентификации.

Если секреты и/или ключи не совпали, то пользователю доступны возможности остановить чат или запросить повторно ввод секрета. Если аутентификация прошла успешно ключ добавляется в список доверенных ключей пользователя - whitelist для того, чтобы в последующем не производить аутентификацию снова.

2.1.5 Вывод

Протокол SMP позволяет надежно передать и сравнить ключи, при условии, что собеседники владеют общим секретом. Протокол включает в себя 5 шагов (инициатор выполняет нечетные шаги, принимающий – четные), 2 раунда (один раунд – это один прием и передача сообщений) и 4 передаваемых сообщения (на 1-4 шагах). На устройстве с 8ГБ оперативной памяти и процессором Intel Core-i7 3.50 GHz работа протокола в среднем занимает 1.2 секунды, которое включает в себя все требуемые вычисления и передачу сообщений.

Однако если каждый участник будет аутентифицировать всех своих собеседников, то для n участников количество запусков SMP будет пропорционально квадрату от числа участников - $\frac{n^2-n}{2}$ (если рассматривать участников как вершины графа и ребра – как факт запуска протокола, то система эквивалентна полному графу). В последующих главах будет предложен алгоритм, позволяющий уменьшить число SMP, и проведено исследование предложенного алгоритма.

2.2 Круговая аутентификация

Для того чтобы участнику аутентифицировать всех своих собеседников не обязательно попарно проверять каждого участника. Рассмотрим следующий пример: пусть имеется 3 участника: *A, B, C*, где участник *A* аутентифицировал *B*. Тогда если участник *B* аутентифицирует *C*, то результат данной аутентификации будет равен аутентификации *A* и *C*, если допустить, что все участники установили один общий пароль, который будет идентифицировать сразу группу людей. При большом количестве участников не удобно хранить и устанавливать с каждым участником отдельный секрет, поэтому данное предположение будет допустимо.

Основная идея алгоритма заключается в том, что все участники организуют направленный круг и аутентифицируют одного соседа по этому кругу. Задача каждого участника успешно инициировать аутентификацию[6] с помощью протокола Миллионеров-Социалистов. В процессе алгоритма принимать и обрабатывать сообщения о результатах аутентификации остальных участников чата. Если аутентификация соседа неуспешна, участник выбирает следующего участника для инициализации аутентификации. Когда получены результаты об аутентификации о всех пользователях конференции, алгоритм завершается.

Соглашение о сетевой модели

Рассматривается сеть, которая

- Имеет фиксированную топологию
- Является связной
- Является асинхронной

Параметры алгоритма

- Централизация. Алгоритм является децентрализованным, другими словами может быть запущен спонтанно некоторым произвольным подмножеством процессов, неинициаторы которого вовлекаются в алгоритм, когда по ходу вычисления поступает сообщение, запускающее его выполнение.
- Топология. Алгоритм спроектирован в расчете на топологию каждый с каждым.
- Первоначальные сведения. Каждый процесс знает собственное уникальное имя, публичный и приватный ключ, имена соседей и их публичные ключи и

[6] Назовем успешной аутентификацией ту, при которой SMP возвращает успех, то есть два участника обладают общим секретом и верными ключами

секретной информацией, которой обладают только пользователи, договорившиеся заранее о данном сеансе связи.

- Число решений. SMP является протоколом взаимной аутентификации, в котором каждый из участников является одновременно и доказывающим, и проверяющим, что позволяет каждому участнику за один сеанс выполнения протокола доказать свою аутентичность другому участнику.
- Модель нарушителя. Участник, чей ключ был подменен злоумышленником (нарушитель может выдавать себя за данного участника и/или подслушивать сообщения) и человек, нелегальным образом обладающий закрытым ключом участника, не знающий секрета и пытающийся выдать себя за участника.

2.2.1 Описание алгоритма круговой аутентификации

Предварительные этап. Участникам необходимо организовать ориентированный круг. Отсортировав всех участников по их уникальным именам, строится список участников, следующих по кругу после данного участника, для дальнейшей работы алгоритма.

Рабочий этап. Происходит выбор цели для инициализации аутентификации. Если участник нашел ближайшего соседа, ответившего успешно на SMP, инициировать аутентификацию ему не требуется, участник переходит в фазу ожидания результатов аутентификации от других участников, и алгоритм выбора цели вернет неуспех; в противном случае, будет выбран участник для инициализации проверки. При успешном выборе цели участник начинает с выбранной целью SMP, результат которой (имя собеседника) добавляет либо в список доверенных пользователей, либо в список нарушителей в зависимости от результата проверки.

Если проверка для пары участников прошла успешна, то они формируют группу доверенных и рассылают всем результаты проверки. При неудачной проверке участник выбирает следующего по кругу участника для аутентификации. Получив сообщения от доверенных пользователей, участник получает информацию о сформировавшихся группах, принимая его результаты и добавляя их в свой список доверенных[7].

[7] Подробнее в следующей главе

В конечном итоге образуется группа доверенных пользователей, и каждый участник будет располагать информацией о всех участниках, при этом не аутентифицируя всех пользователей.

В таблице 2 указаны основные шаги алгоритма, где status – массив аутентичности участников, имеет значения {GOOD, BAD, UNKNOWN, BAD_NOT_SURE}; aim – выбранный участник для инициализации SMP.

```
Sort(peers); // сортировка участников
CSMP()
  if Choose_aim():  // выбор цели для инициирования SMP
        Smp(aim); // инициировать SMP в случае необходимости
        if smp.status[aim] == GOOD: // если aim прошел аутентификации
              Send_results(); // разослать информацию об образованной группе
        else     CSMP() // aim не прошел аутентификацию, выбор след цели для SMP
  endif
end function
```

Таблица 2 Основные шаги Кругового алгоритма

2.2.2 Обработка сообщений о результате аутентификации

Когда участник сформировал группу с каким-либо из участников или добавил нового участника, он рассылает всем сообщение со списком доверенных и списком нарушителей вместе с идентификатором сессии и подписью, таким образом сообщая участникам о формировании новый группы или о ее составе.

При получении сообщения с информацией об аутентификации участник проверяет подпись и идентификатор сессии, далее обрабатывает данное сообщение в зависимости от того, какими данными об отправителе он обладает. Если аутентичность отправителя неизвестна или в процессе проверки, то сообщение сохраняется (хранится только последнее пришедшее сообщение от каждого участника). Если отправитель считается доверенным, то получатель принимает его информацию об аутентификации, то есть участники, числящиеся в списке его доверенных, добавляются в список доверенных получателя, а участники, числящиеся в списке его нарушителей, добавляются в список нарушителей.

Рассмотрим для наглядности пример, где каждый участник прошел SMP с двумя своими соседями и где *A* успешно прошел аутентификацию с *B*, а *B* с *C*

(Рисунок 3. Формирование группы доверенных), но *F* не прошел с *A*, а *C* с *D.* На этом этапе у *A* есть информация про *B* и *F*, у *B* про *A* и *C*, у *C* про *B* и *D*. После рассылки сообщений о результатах своих проверок, *A* и *C* будут знать друг о друге (Рассмотрим пример со стороны *A*: *A* после успешной аутентификации с *B* принимает от *B* список проверенных им пользователей, то есть о *C* и потом принимает и список проверенных от *C (*то есть принимает и *D* в группу*)*, таким образом A узнает за время одной передачи о *C* и *D*). *A*, *B* и *C* сформировали группу доверенных, где каждый обладает равной информацией об *A*,*B*,*C*,*D*,*F*, но в группе неизвестно об аутентичности *E*. На следующем этапе *C* инициирует SMP с *E.*

Рисунок 3. Формирование группы доверенных

Если отправитель сообщения числится нарушителем, то участники, числящиеся в списке его доверенных, сохраняются у получателя как нарушители из одной группы после того, как они подтвердят, что они из одной группы.

В таблице 3 указаны основные шаги алгоритма, где good, bad – списки доверенных и нарушителей отправителя sender; mail – хранит полученные и

необработанные сообщения; list_to_check – список доверенных участников для нарушителя.

```
handleMessage(sender, good, bad ) {
    switch (status[sender]){
        case  UNKNOWN:
      case  IN_PROCESS:
          mail[sender][good] := good; mail[sender][bad] := bad;
          break;
      case GOOD:
          forall peer ∈  good:
              if status[peer] == UNKNOWN:
                 status [peer] := GOOD;
                 handleMessage( peer, mail[ peer][good], mail[ peer][bad]);
              end if
          forall peer ∈  bad:
              if status[peer] ==UNKNOWN:
                 status[peer] := BAD;
                handleMessage( peer, mail[ peer][good], mail[ peer][bad]);
              end if
          break;

       case BAD_NOT_SURE:
       case BAD:
          forall peer ∈  good:
              // Подтверждение утверждений об отправителе
              if list_to_check[sender] && list_to_check[sender].has(peer):
                 list_to_check[sender].delete(peer);
                 status[sender] := BAD;
              end if
                 // Сохранение участников для будущего подтверждения
              if status[peer] ==  UNKNOWN:
                 status[peer] :=  BAD_NOT_SURE;
                 if list_to_check[peer] == undefined:
                    list_to_check[peer] := [sender];
                 else list_to_check[peer].add(sender);
                 handleMessage( peer, mail[peer][good], mail[peer][bad]);
              end if
          }
          break;
    }
```

Таблица 3 Алгоритм обработки сообщений о результате аутентификации

2.2.3 Выбор цели для инициирования SMP

Если участник успешно инициировал одну аутентификацию или обладает информацией об аутентичности всех участников, то алгоритм выбора цели вернет неуспех. Инициировать SMP требуется только тем участникам, которые не нашли ближайшего доверенного соседа. В этом случае, участник *A* выбирает пользователя *B*, если он удовлетворяет следующим условиям:

- Пользователь *B* в круге стоит после того, кто не прошел аутентификацию с *A*
- Участник A не обладает информации об аутентичности B

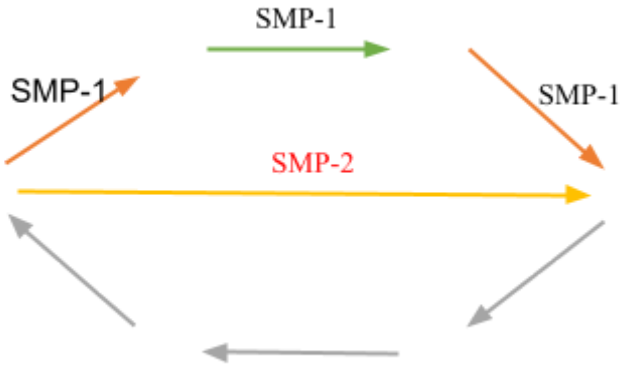
В таблице 4 представлены основные шаги алгоритма выбора цели.

```
choose_aim(){
 if amount_unknown == 0: // если имеется инф-ция о все участниках
            status := DONE; // изменить статус CSMP и вернуть 0
            return 0;
end if
forall peer ∈circle  do: // поиск первого неизвестного участника
          if result[peer] == unknown:
                    aim = peer;
                    return 1;
          end if

          if result[peer] == GOOD: // при нахождении успешно аутентиф. участника
                    status := done; // вернуть 0
                    return 0;
          end if
end function
```

Таблица 4 Алгоритм выбора цели для инициализации SMP

Полученная информация о сформулированных группах избавляет участника от личной проверки всех участников. На Рисунок 4. Выбор цели *C* не будет целью для *A*, так как B и *C* – группа и для *A* – *B* нарушитель, поэтому результат проверки *C* будет заведомо известен для *A*: *C* так же является нарушителем.



A

B

C

F

D

E

Рисунок 4. Выбор цели для участника А.[8]

2.2.4 Безопасность. Возможные атаки.

Безопасность данного алгоритма основывается на безопасности схемы цифровой подписи (RSA, то есть на сложности разложения числа на множители) и на безопасности алгоритма для попарной аутентификации – SMP (подробнее в разделе 3.2, однако круговой алгоритм работает и для любого другого протокола взаимной попарной аутентификации). Поэтому после того как участник удостоверился в аутентичности ключа собеседника, он может доверять сообщениям подписанные проверенным ключом, так как нарушитель, не обладая закрытым ключом проверенного собеседника, не сможет отправить сообщение от его имени.

Рассмотрим наиболее широко известных атак на криптографические протоколы [7] для предложенного алгоритма Круговой аутентификации.

[8] Зеленым обозначена успешная аутентификация, оранжевым – неуспешная, желтым – инициирование СМ, серым – не рассматривается.

- *повтор* - атака, при которой при доказательстве утверждения используются сообщения из одного из предыдущих сеансов; Данная атака не пройдет успешно, так как каждое сообщение отправляется вместе с идентификатором сессии, которое показывает актуальность передаваемого сообщения.
- *отражение* - атака, в ходе которой участнику протокола отправляются сообщения, ранее полученные от него в рамках текущего активного взаимодействия; Противодействие к атаке состоит в обеспечении целостности сеансов протоколов, которое достигается обновлением идентификатора сессии в каждый фиксированный промежуток времени.
- *вынужденная задержка* - атака, когда нарушитель перехватывает сообщение и передаёт его с некоторой задержкой; Противодействие атаке обеспечивается ограничением временного промежутка для ответа.
- *атака с выбранным текстом* - атака, когда нарушитель пытается получить информацию о секрете путём передачи специальных значений параметров проверки. Нулевого разглашение секрета гарантируется протоколом SMP, а круговой алгоритм в свою очередь взаимодействует с секретным значением только через SMP.
- *атака на основе подобранного шифротекста* – атака, при которой злоумышленник может доказать владение секретом, не обладая им на самом деле, или, другими словами, может имитировать то лицо, которому на самом деле принадлежит секрет. Предложенный протокол будет уязвим к атаке, однако возможным решением является шифрование сообщений публичным ключом.

Подробнее рассмотрим атаку Сивиллы, при которой злоумышленник наполняет сеть большим числом подконтрольными ему узлами, а жертва подключается только к узлам, контролируемыми злоумышленником, тем самым отсоединившись от общей сети. Нарушитель, контролируя узлы, имеет возможность перехватывать сообщения и подменять передаваемые участнику ключи для того, чтобы либо выдать себя за реального участника, либо подслушивать беседу. Данный участник в нашей модели будет считаться нарушителем, так как в чате для остальных пользователей фактически будет не сам участник, а злоумышленник, а для жертвы беседа может быть поддельной злоумышленником. Беседа с данным

участником уже небезопасна как для жертвы, так и для остальных участников, поэтому аутентификация должна выявить присутствие нарушителя.

Рассмотрим несколько возможных вариантов развития события. Первое – при аутентификации нарушитель перехватывает все сообщения, тем самым имитировав для жертвы процесс групповой аутентификации. Тогда нарушитель имеет возможность отправить сообщения, в которых бы утверждалось, что все участники (поддельные для жертвы) доверенные. Однако жертва не примет ни одного сообщение, если не пройдет аутентификацию с его отправителем, а так как нарушитель не обладает паролем, то сообщения и не будут приняты, и в конечном итоге по завершению круговой аутентификации, жертва поймет, что беседа не безопасна. Остальные участники так же узнают о нарушителе, скрывающийся под именем жертвы, по причине незнания нарушителем пароля.

Второй вариант развития событий - нарушитель не перехватывает сообщения об аутентификации. В этом случае жертва аутентифицируется с 2 своими соседями, и SMP вернет неуспех для участников, так как ключи в секретах будут не совпадать, хоть пароли и совпадают. Таким образом, в известных нам ситуациях круговая аутентификация сможет противостоять атаке Сивиллы.

3 Описание практической части

3.1 Реализация

Предложенный алгоритм Круговой аутентификации был реализован на языке Javascript в модуле CSMP (288 строк кода) [6]. Для реализации криптографических вычислений была использована библиотека Cryptico [14] и для работы с длинной арифметикой – BigInteger [15]. Описание реализации протокола SMP в главе 2.1.4. Программная реализация Кругового алгоритма представлена на рис. 5.

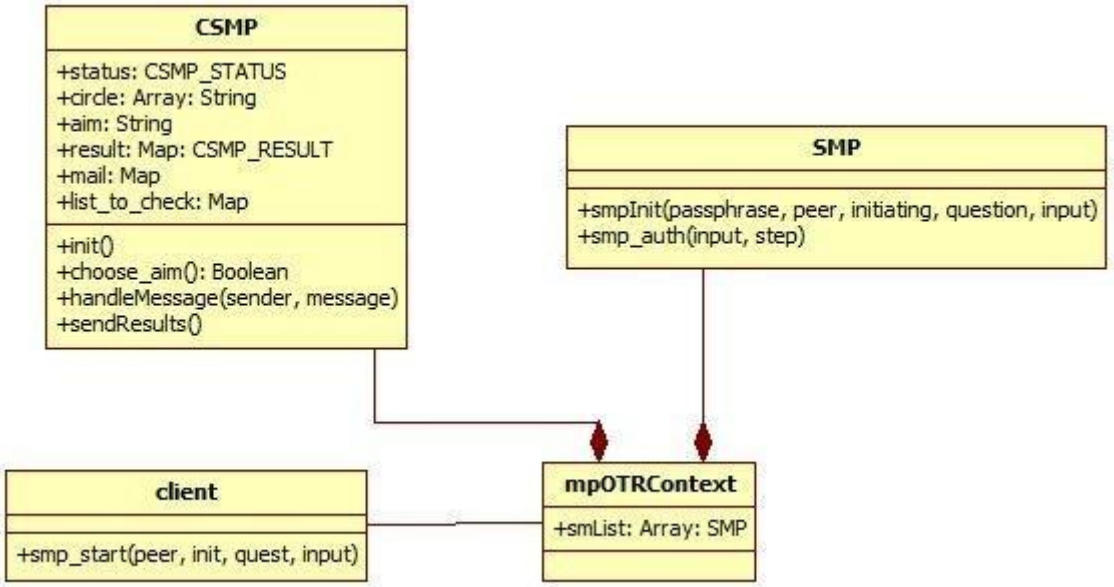


Рисунок 5. UML диаграмма программы

Протокольные сообщения имеют один из 2 типов CSMP_INIT (для инициализации круговой аутентификации) и CSMP_RESULT (для передачи сообщений).

На фазе коммуникаций протокола mpOTR пользователям доступна Круговая аутентификация (попарная так же доступна), которая запрашивает ввод пароля для своей работы (Рис. 2). В ходе аутентификации участникам подсвечивается зеленым собеседники, успешно прошедшие аутентификацию и красным – неуспешно прошедшие аутентификацию (Рис. 1). В результате работы протокола пользователь будет обладать информацией о потенциально небезопасных участниках. Далее пользователи могут остановить чат либо аутентифицировать участника/участников снова.

3.2 Тестирование

Будем называть аутентификацию, где каждый участник проходит проверку SMP со всеми участниками (случай полного графа) – полной, а аутентификацию с использованием кругового протокола - круговым.

Цель тестирования: сравнить полную аутентификацию и круговую по времени работы, вычислительной сложности и количеству передаваемых сообщений в зависимости от количества участников.

Тесты были проведены в браузере Google Chrome 65.0.3325.181. Одновременно запускается несколько экземпляров программы – по требуемому количеству участников, и эти процессы воспроизводят аутентификацию участников по всем этапам предложенного алгоритма – от начала аутентификации до его завершения.

Количество участников	Полная аутентификация - Время (мс)	Круговая аутентификация - Время (мс)
3	2646	2379
4	4120	3217
5	6736	3982
7	14088	5598
9	26886	8734

Таблица 5. Время работы алгоритмов в зависимости от числа участников

	Полная аутентификация	Круговая аутентификация
Кол-во раундов	*n-1*	$2 + 2p$
Вычислительная сложность	$O(n^2)$	$O(n)$
Количество сообщений	$2(n^2 - n)$	$n^2 + (3 + 4p)n - p$

Таблица 6. Сравнение полной и круговой аутентификации по вычислительной сложности (кол-во запусков SMP) и кол-ву передаваемых сообщений

Тестирование показало эффективность использования предложенного алгоритма Круговой аутентификации в сравнении с Полной. Уменьшение вычислительной сложности от квадратичной до линейной (Таблица 6), где *n* – число участников чата, *p* – максимальное число подряд идущих нарушителей, не образующих группу), позволило добиться уменьшения времени для аутентификации всех участников (Таблица 5).

4 Заключение

В рамках данной работы были достигнуты следующие результаты:

1. Проведено исследование и сравнительный анализ протоколов доказательства с нулевым разглашением, по результатам которого был сделан вывод об отсутствии протокола взаимной групповой аутентификации со свойством нулевого разглашения и выбран протокол Миллионеров-Социалистов как основа для работы.

2. Был предложен протокол для групповой взаимной аутентификации на основе выбранного протокола Миллионеров-Социалистов, особенностью которого является то, что он разрабатывалась без жёсткой привязки к определённому протоколу аутентификации, поэтому поддерживает многие схемы.

3. Был реализован предложенный протокол для модельной системы на основе mpOTR и алгоритм Полной аутентификации для экспериментального исследования и их сравнения.

4. Проведено экспериментальное исследование предложенного протокола Круговой аутентификации и его сравнение с алгоритмом Полной аутентификации.

По результатам анализа можно сделать вывод о применимости протокола Круговой аутентификации для систем группового обмена сообщениями. При этом есть возможности для совершенствования алгоритмов, используемых в предложенном варианте протокола, повышения эффективности их работы.

В дальнейшем планируется работа над увеличением эффективности предложенного алгоритма, например, путем хранения аутентифицированных долговременных открытых ключей для избегания повторной проверки ключей при последующем соединении и рассмотрении ряда проблем, возникающих при их использовании.

5 Список литературы

[1] The Guardian. Edward Snowden [HTML]

(http://www.theguardian.com/world/edward-snowden).

[2] В. Коростелева М. Исследование протокола криптографически защищенных групповых коммуникаций с функцией отказуемости // Проблемы информационной безопасности. Компьютерные системы - 2015. - с. 79.

[3] mod_otr - Man in the Middle module for Off-The-Record [HTML]
(https://www.ejabberd.im/mod_otr).

[4] Off-the-Record Messaging Protocol version 3 [HTML]
(https://otr.cypherpunks.ca/Protocol-v3-4.1.1.html).

[4] Markus Jakobsson, Moti Yung Proving without knowing: On oblivious, agnostic and blindfolded provers. // Advances in Cryptology—CRYPTO'96. — 1996. — С. 189.

[5] Fabrice Boudot, Berry Schoenmakers, Jacques Traore A Fair and Efficient Solution to the Socialist Millionaires' Problem // Discrete Applied Mathematics. — 2001. — С. 3.

[6] Шейдаев В., Гамаюнов Д., Нгуен К. К., Рыбникова В., Шурыгин А. Исходный код реализации протокола круговой аутентификации для группового чата защищенных коммуникаций [HTML]
(https://bitbucket.org/Enr1g/p2p_mpotr.js/commits/branch/Authentification).

[7] А. В. Черемушкин Криптографические протоколы: Основные свойства и уязвимости // Прикладная дискретная математика– 2009. – С. 124-125.

[8] Мао Венбо Соверменная криптография: теория и практика // Пер. с англ. – М.: Изд-во Вильямс, 2005. – с. 685-688.

[9] Гашков С.Б., Э. А. Применко, М. А. Черепнев Криптографические методы защиты информации // М.: Изд-во «Академия» - 2010. - с. 137.

[10] Ященко В.В. Введение в криптографию. Под общей ред. В. В. Ященко — СПб.: Питер, 2001. - с. 46.

[11] А.М. Иванцов, С.М. Рацеев О применении эллиптических кривых в некоторых протоколах аутентификации и распределения ключей // Информационные системы. Автоматизация процессов управления. № 2 (48), 2017. – с. 40-41

[12] Andrew Chi-Chih Yao Protocols for Secure Computations // Foundations of Computer Science. — 1982., c. 1-5

[13] Subhasish Paramanik Comparison of Zero Knowledge Authentication Protocols [PDF]

(http://ethesis.nitrkl.ac.in/5755/1/110CS0371-2.pdf)

[14] Cryptico. An easy-to-use encryption system utilizing RSA and AES for javascript. [HTML]

(https://github.com/wwwtyro/cryptico)

[15] BigInteger. arbitrary-length integer library for Javascript [HTML]

(https://www.npmjs.com/package/big-integer)