МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

УДК 004.056 DOI 10.17223/20710410/56/4

МЕТОДИКА ОЦЕНКИ БЕЗОПАСНОСТИ КРИПТОГРАФИЧЕСКИХ ПРОТОКОЛОВ[1]

А. Ю. Нестеренко, А. М. Семенов

Национальный исследовательский университет «Высшая школа экономики», Московский институт электроники и математики им. А. Н. Тихонова (МИЭМ НИУ ВШЭ), г. Москва, Россия

E-mail: anesterenko@hse.ru, amsemenov@hse.ru

Предлагается метод оценки безопасности криптографических протоколов, используемых для защиты информации как в информационно-телекоммуникационных сетях, так и в сетях «Интернета вещей». Описывается порядок оценки безопасности информационной системы, включающий в себя построение перечня угроз, модели угроз и детализации модели и возможностей нарушителя. Рассматривается понятие «свойство безопасности», приводится расширенный перечень указанных свойств, их классификация и формальная математическая модель. В рамках модели, для заданных свойств безопасности, предлагается метод получения численных значений показателей эффективности, зависящих от вероятности успеха и алгоритмической сложности решения ряда известных математических задач. Приводятся результаты применения предложенного метода к анализу стандартизируемых в Российской Федерации протоколов ESP и IKEv2 семейства IPSec.

Ключевые слова: *свойство безопасности, криптографический протокол, показатель эффективности защиты информации.*

METHODOLOGY FOR ASSESSING THE SECURITY OF CRYPTOGRAPHIC PROTOCOLS

A. Yu. Nesterenko, A. M. Semenov

National Research University "Higher school of economics", Tikhonov Moscow Institute of Electronics and Mathematics (MIEM NRU HSE), Moscow, Russia

This paper proposes a method for evaluating the security of cryptographic protocols used to protect information in telecommunication networks, as well as in networks of the "Internet of Things". The procedure for evaluation of information system security is described, including the construction of the list of threats, the threat model, and detailing of the model and the abilities of the intruder. The concept of security property is considered, the extended list of the specified properties, their classification and formal mathematical model are given. As part of the model, for given properties

[1]Исследование выполнено при финансовой поддержке РФФИ, проект № 19-37-90155.

of security, we propose a method for obtaining numerical values of performance parameters depending on the probability of success and algorithmic complexity of the solution of a number of known mathematical problems. In conclusion, the results of the application of the proposed method to the analysis of ESP and IKEv2 protocols of IPSec family standardized in the Russian Federation are presented.

Keywords: *security property, cryptographic protocol, information security performance indicator.*

Введение

В настоящее время не вызывает сомнений, что обеспечение безопасности информации, передаваемой в сетях связи (информационно-телекоммуникационных сетях, сетях «Интернета вещей» и т. п.) должно реализовываться при помощи криптографических механизмов — схем и протоколов. При этом высокий уровень защиты информации, обеспечиваемый криптографическими механизмами, является необходимым фактором для обоснования безопасности информационных (автоматизированных) систем, в состав которых входят сети связи.

Принятый в Российской Федерации порядок оценки безопасности информационных (автоматизированных) систем заключается в следующем:

1. Формируется модель угроз, создающих опасность нарушения безопасности передаваемой информации. Разработка модели угроз проводится для конкретной системы на основе базовых моделей, регламентируемых ФСТЭК и ФСБ России, а также государственными стандартами в области защиты информации.
   В настоящей работе будем основываться на базовой модели угроз, регламентируемой стандартом [1]. Данная модель включает в себя:
   - угрозу несанкционированного доступа к передаваемой информации (нарушение конфиденциальности);
   - угрозу несанкционированной передачи информации;
   - угрозу несанкционированного изменения информации (нарушение целостности);
   - угрозу отказа от факта отправки или приёма сообщения;
   - угрозу внесения вредоносного программного обеспечения;
   - угрозу отказа в обслуживании или предоставлении услуг (нарушение доступности).
2. Формируется модель нарушителя, содержащая совокупность возможностей, которые могут быть использованы при создании способов, подготовке и проведении атак, целью которых является реализация перечисленных угроз безопасности.
   Будем основываться на модели нарушителя, регламентируемой рекомендациями по стандартизации [2]. В данной модели инструментом реализации угроз безопасности являются проводимые нарушителем атаки на информационную систему и, в частности, на криптографические протоколы, обеспечивающие безопасность передачи информации. Каждая атака нарушителя может быть задана следующими характеристиками:
   а) объектом проведения атаки, безопасность которого должна обеспечиваться в течение определённого периода времени и/или определённого этапа жизненного цикла средства защиты информации;

б) возможностями, которые могут быть использованы при создании способов, подготовке и проведении атак; каждая возможность определяется сведениями, а также техническими средствами;
в) местом проведения атаки.

Применительно к анализу криптографических схем и протоколов понятие объекта атаки позволяет уточнить сформулированный выше перечень угроз. Это вызвано тем, что в качестве объектов атаки могут выступать параметры схем и протоколов, используемые для обеспечения криптографической защиты информации.

Возможности нарушителя в части воздействия на канал связи, по которому происходит обмен защищаемой информацией, принято описывать расширенной моделью Долева — Яо [3]. В рамках данной модели нарушитель обладает следующими возможностями:

— нарушителю известны форматы всех передаваемых сообщений;
— нарушитель может перехватить и получить содержимое любого сообщения от любого пользователя в сети связи;
— нарушитель может инициировать установление соединения с любым другим пользователем;
— нарушитель может изменять содержимое передаваемых пользователями сообщений и, в частности, посылать сообщения от имени другого пользователя;
— нарушитель может использовать все доступные ему комбинации сообщений или частей сообщений для формирования новых сообщений, в том числе расшифровывать и зашифровывать сообщения с помощью известных ему ключей шифрования, применяя любые доступные алгоритмы;
— нарушитель является полноценным пользователем сети, обладающим корректным собственным идентификатором и допустимым множеством ключевой информации;
— нарушитель может накапливать всю переданную в сеть связи информацию, проводить её анализ с применением специализированных технических средств и использовать результаты анализа для компрометации криптографических схем и протоколов;
— нарушитель может организовывать одновременное выполнение некоторого числа сессий одного и того же протокола защиты информации; сессии могут выполняться одновременно для различных участников протокола, при этом нарушитель может использовать информацию, передаваемую в ходе всех выполняемых сессий протокола.

Перечисленные методы реализации угроз безопасности принято называть «активными» атаками.

В модели Долева — Яо нарушитель может проводить также «пассивные» атаки на протокол, основанные на перлюстрации и последующем криптографическом анализе передаваемых в ходе выполнения протокола сообщений. При проведении пассивных атак предписанное заранее (регламентированное) выполнение прокола не меняется — нарушитель не изменяет передаваемые сообщения, не инициирует соединений и не вмешивается в логику взаимодействия пользователей сети.

Для усложнения рассматриваемой модели будем допускать, что нарушитель может компрометировать набор долговременных ключей любого потенциаль-

ного участника протокола, который не является участником атакуемой сессии выполнения протокола. Вариант нарушителя данного типа описан в [3, 4] и использован при построении атак на протокол Нидхема — Шредера [5].

3. Проводится исследование криптографических схем и протоколов, обеспечивающих безопасность сети связи, а также входящих в состав протокола криптографических преобразований. Целью исследований является определение численных значений одного или нескольких показателей эффективности [6], которые позволяют оценить уровень защищённости информационной системы. Система считается защищённой (безопасной), если значение показателя эффективности превышает величину, установленную нормативными, правовыми документами или требованиями по безопасности.

В результате оценки безопасности информационной (автоматизированной) системы мы должны получить ответы на следующие вопросы: что защищаем, от кого защищаем и как оценить уровень обеспечиваемой защиты?

Ответы на первые два вопроса могут быть получены с учётом области применения информационной (автоматизированной) системы и действующей в Российской Федерации нормативной базы. Однако единой методологии определения показателей эффективности и их значений применительно к криптографическим протоколам в настоящее время нет. В зарубежных публикациях принято использовать несколько подходов:

— базовую модель Белларе — Рогавея [7] и её модификации [8 – 11], в которых в качестве показателя эффективности может рассматриваться вероятность нарушения формального определения безопасного протокола; получение точных численных оценок показателя эффективности в данной модели не предполагается;
— модель Конетти — Кравчука [12] и её модификации [13 – 15], в которых в качестве показателя эффективности выступает величина отклонения от 1/2 вероятности различения двух моделей — практической модели протокола в рамках описанной выше модели нарушителя и «идеальной» модели протокола, реализующей обмен сообщениями по «идеальному» каналу связи без искажений и активного нарушителя.

Среди ранних работ по анализу криптографических протоколов стоит выделить [16, 17]. Более поздние обзоры зарубежных публикаций могут быть найдены в монографиях [18, 19].

В отечественных работах по анализу протоколов принято использовать два подхода:

— применение «практической стойкости», т. е. классического криптографического анализа для получения оценок стойкости используемых в протоколе криптографических примитивов [20, 21]; при данном подходе показателем эффективности служит минимальное из всех возможных значений трудоёмкости реализации известных атак на криптографические преобразования;
— применение теории «доказуемой стойкости», позволяющей исследовать безопасность протоколов в заданных вероятностных моделях поведения нарушителя с ограниченными вычислительными ресурсами; аналогично методу Канетти — Кравчука в качестве показателя эффективности в данном подходе выступает величина отклонения от 1/2 вероятности различения заданных параметров моделей от случайных равновероятно распределённых случайных величин [22, 23].

Настоящая работа использует первый подход к определению показателя эффективности защиты. Для расширения области его применения приводится способ построения формальной модели протокола — графа зависимостей между состояниями

субъектов взаимодействия. В п. 1 рассматривается понятие «свойства безопасности», позволяющего связать между собой угрозы безопасности и возможные атаки нарушителя (рис. 1). Приводится классификация свойств безопасности и их взаимосвязь между собой.

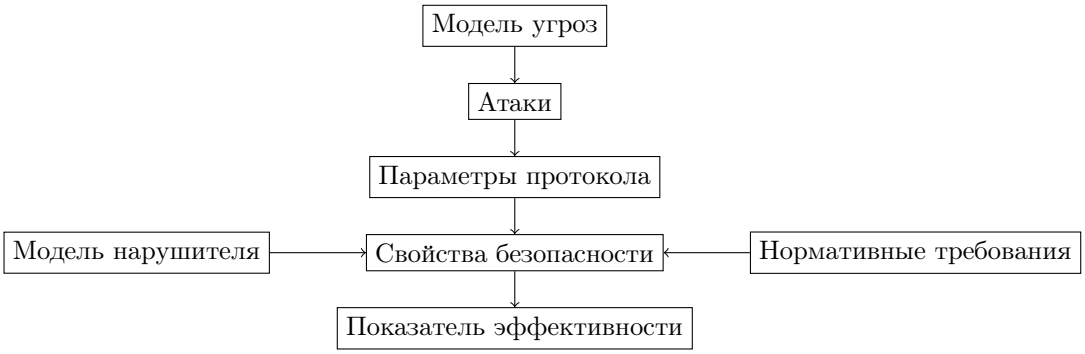

Рис. 1. Схема вычисления показателя эффективности

В п. 2 рассматривается формальная модель криптографического протокола и с её помощью моделируются базовые свойства безопасности. В п. 3 даётся формальное определение показателей эффективности защиты и предлагается способ определения их численных значений, основанный на трудоёмкости решения известных математических задач.

В п. 4 приводится методика получения численных значений показателя эффективности. Данная методика сложилась в ходе проведения исследований таких протоколов, как TLS 1.3 [24] и SP FIOT [25]; фрагменты методики рассматривались ранее в работах [26 – 28]. В п. 5 приводятся результаты применения предложенной методики к анализу стандартизированных в Российской Федерации протоколов ESP и IKEv2 семейства IPSec [29].

1. Свойства безопасности

Рассмотрим криптографические механизмы, получившие наибольшее распространение в современных информационных системах и обеспечивающие защищённый обмен информации с трёхсторонним участием — двух абонентов, обменивающихся информацией, и доверенного центра, обеспечивающего функции аутентификации участников взаимодействия. При этом участие доверенного центра в обмене информацией может быть косвенным, т. е. без отправки и получения сообщений.

Данные механизмы могут представлять собой совокупность, состоящую из нескольких схем и протоколов. В состав такой совокупности, как правило, входят:

— протокол односторонней, взаимной или многосторонней аутентификации участников информационного взаимодействия;
— протокол выработки общей для участников взаимодействия ключевой информации, действующей в рамках одной сессии информационного взаимодействия;
— транспортный протокол, предназначенный для передачи защищённой информации по каналам связи;
— процедуры выработки производной ключевой информации, контроля за временем и объёмом используемой ключевой информации;

— вспомогательные протоколы, предназначенные для передачи ошибок информационного взаимодействия, квитирования абонентов, инициализации процедуры выработки нового сессионного ключа и т. п.

В информационных системах «Интернета вещей» к этой совокупности могут добавляться протоколы взаимодействия различных сегментов общей сети связи. Проводимый анализ безопасности должен учитывать все элементы и выдавать единое значение показателя эффективности.

Инструментами, которые позволяют не только связать между собой угрозы безопасности и возможные атаки нарушителя, но и получать численные значения показателя эффективности, являются так называемые «свойства безопасности» (их называют также функциями-сервисами безопасности [30]).

Определение 1. Пусть π_0, $0 < \pi_0 \leqslant 1$, — заданное действительное число. Под свойством безопасности будем подразумевать свойство протокола прямо или косвенно обеспечивать невозможность реализации заданной угрозы с вероятностью, превышающей значение π_0.

Значение π_0 может быть равно 0,5, 0,1, 0,01, 0,001 и т. п. Оно определяет вероятность нарушения свойства безопасности и позволяет вывести из рассмотрения атаки с ничтожной вероятностью успеха, например случайное угадывание зашифрованного текста. Точное значение π_0 может определяться действующими требованиями по безопасности, моделью угроз безопасности передаваемой информации или рассчитываться с использованием риск-ориентированного подхода.

Свойства безопасности впервые вводятся в [7], а позднее расширяются в RFC 3552 [31] и в рамках проекта AVISPA [32]. Они рассматриваются также в ГОСТ Р ИСО/МЭК 27033-1:2011, разд. 7.3, и в работах [30, 28]. Расширим перечень из [32] и будем использовать следующие свойства безопасности:

С 1. *Свойство аутентификации субъекта* (*участника протокола*) *другим субъектом* (*участником протокола*) заключается в подтверждении одним субъектом подлинности другого субъекта, а также в получении гарантии того, что субъект, подлинность которого подтверждается, действительно принимает участие в выполнении текущей сессии протокола.
Свойство аутентификации субъекта может быть как односторонним, так и взаимным. В последнем случае свойство должно выполняться для всех участвующих во взаимодействии субъектов. Данное свойство содержится в [30, разд. 3; 31, п. 2.1.3; 32, свойство G1].

С 2. *Свойство аутентификации сообщения* заключается в подтверждении подлинности источника сообщения и целостности передаваемого сообщения.
Подлинность источника сообщения означает, что протокол должен обеспечивать гарантии того, что полученное сообщение или его часть были созданы участником взаимодействия в ходе выполнения текущей сессии протокола в некоторый момент времени, предшествующий получению сообщения. Фактически в рамках данного свойства сообщение однозначно связывается со своим источником (субъектом, отправившим сообщение), а выполнение свойства гарантирует, что сообщение не было искажено, в частности подделано нарушителем, при передаче по каналам связи. Данное свойство содержится в [30, разд. 3; 31, п. 2.1.2; 32, свойство G2].

С 3. *Свойство целостности сообщений* заключается в том, что получатель сообщения обладает возможностью проверить, что полученные им данные (или их

часть) не были модифицированы, уничтожены и являются теми же самыми данными, что послал отправитель. Данное свойство содержится в [30, разд. 2; 31, п. 2.1.2].

С 4. *Свойство защиты от повторов* заключается в том, что один раз корректно принятое участником протокола сообщение не должны быть принято повторно. В зависимости от протокола данное свойство может быть сформулировано в виде одного из следующих требований:
   — должна быть обеспечена гарантия того, что сообщение выработано в рамках текущей сессии протокола;
   — должна быть обеспечена гарантия того, что сообщение выработано в рамках заданного интервала времени;
   — сообщение не было принято ранее.

   В отечественной литературе данное свойство часто называют *свойством невозможности навязывания ложных сообщений*, подразумевая под этим защиту как от повторного принятия истинных сообщений, так и от подделанных нарушителем сообщений (свойство С 2). Данное свойство содержится в [30, разд. 3; 32, свойство G3].

С 5. *Свойство неявной аутентификации получателя* заключается в том, что протокол должен обладать средствами, гарантирующими, что отправленное сообщение может быть прочитано только теми участниками, для которых оно предназначено. Только законные авторизованные участники должны иметь доступ к данной информации, многоадресным сообщениям или групповому взаимодействию. Данное свойство содержится в [32, свойство G4].

С 6. *Свойство групповой аутентификации* заключается в том, что законные авторизованные члены заранее определённой группы пользователей могут аутентифицировать источник и содержание информации или группового сообщения. Сюда также входят протоколы, в которых участники группового взаимодействия не доверяют друг другу. Данное свойство содержится в [30, разд. 3; 32, свойство G5].

С 7. *Свойство аутентификации субъекта* (*участника протокола*) *доверенной третьей стороной*. В протоколах, явно реализующих взаимодействие участников с доверенной третьей стороной, данное свойство эквивалентно первому из перечисленных свойств.

   В случае использования инфраструктуры открытых ключей данное свойство может выполняться косвенно, путём заверения открытых ключей участников взаимодействия электронной подписью удостоверяющего (доверенного) центра; при этом привязка аутентификации субъекта к какой-либо сессии протокола не может быть обеспечена. Данное свойство содержится в [32, свойство G6].

С 8. *Свойство конфиденциальности ключа* предполагает, что в ходе информационного взаимодействия значение ключа не может стать известным нарушителю, а также легитимным пользователям информационной системы, для которых данный ключ не предназначен. Данное свойство может применяться как к исходной ключевой информации, так и к производным сессионным ключам.

С 9. *Свойство аутентификации ключа* предполагает, что один из участников взаимодействия получает подтверждение того, что никакой другой участник, кроме заранее определённого второго участника и, возможно, доверенного центра, не может обладать секретным ключом, выработанным в ходе выполнения протокола. Данное свойство содержится в [30, разд. 3; 32, свойство G7].

С 10. *Свойство подтверждения ключа* заключается в том, что один из участников взаимодействия получает подтверждение того, что второй участник (или группа участников) действительно обладает заданным секретным ключом и/или имеет доступ к информации, необходимой для выработки заданного секретного ключа. Данное свойство содержится в [30, разд. 3; 32, свойство G8].

С 11. *Свойство стойкости при компрометации производных ключей* состоит в том, что компрометация производных ключей, т. е. ключей, используемых непосредственно для шифрования и имитозащиты передаваемой информации, не приводит к нарушению других свойств безопасности как в рамках текущей, так и в других сессиях протокола, в частности к компрометации производных ключей, выработанных ранее или планируемых к выработке в дальнейшем.
В литературе данное свойство часто называют защитой от «чтения вперед/назад» или используют термин «perfect forward secrecy». Данное свойство содержится в [30, разд. 3; 32, свойство G9].

С 12. *Свойство стойкости при компрометации ключа аутентификации* состоит в том, что компрометация долговременного ключа аутентификации не приводит к нарушению конфиденциальности информации, переданной до момента компрометации ключа, а в случае пассивного нарушителя — и к нарушению конфиденциальности информации, передаваемой после завершения текущей сессии протокола. В литературе данное свойство иногда называют защитой от «чтения назад».

С 13. *Свойство формирования новых ключей* заключается в том, что протокол, обладающий данным свойством, позволяет формировать уникальные сессионные и/или производные ключи для каждой сессии протокола. Данное свойство содержится в [32, свойство G10].

С 14. *Свойство защиты от навязывания ключевых значений* гарантирует, что ни один из участников протокола не может навязать значение общего секретного, сессионного или производного ключа по своему выбору другому участнику протокола.

С 15. *Свойство защиты от навязывания параметров безопасности* гарантирует, что используемые в ходе выполнения протокола или согласуемые на этапе установления соединения параметры безопасности не могут быть навязаны нарушителем. В качестве параметров безопасности могут выступать наборы используемых криптографических преобразований, численные параметры алгоритмов и алгебраических структур, в которых выполняется протокол, случайные значения, вырабатываемые в ходе выполнения протокола и т. п. Данное свойство содержится в [32, свойство G11].

С 16. *Свойство конфиденциальности* заключается в том, что данные, передаваемые в ходе информационного взаимодействия, не могут стать известными нарушителю и/или легитимным участникам, для которых они не предназначены. Данное свойство содержится в [30, разд. 3; 31, п. 2.1.1; 32, свойство G12]. Легко видеть, что нарушение свойства конфиденциальности ключевой информации (С 8) приводит к нарушению конфиденциальности передаваемых данных.

С 17. *Свойство инвариантности отправителя* заключается в том, что на протяжении выполнения всего протокола получатель сообщений сохраняет уверенность в том, что источник сообщения остался тем же, что и источник, с которым было начато взаимодействие (сессия протокола). Данное свойство содержится в [32, свойство G16].

С 18. *Свойство анонимности субъекта* (*участника протокола*) состоит в том, что нарушитель, осуществляющий перехват сообщений, не должен иметь возможность связать сообщения одного из участников с самим участником или его идентификатором. Данное свойство содержится в [32, свойство G13].

С 19. *Свойство анонимности субъекта для других участников* заключается в том, что каждый участник взаимодействия не должен иметь возможность узнать реальную личность других участников, а должен взаимодействовать с их псевдонимом или случайным идентификатором. Данное свойство содержится в [32, свойство G14].

С 20. *Свойство защищённости от атак «отказ в обслуживании»* подразумевает, что реализующее протокол средство защиты информации обеспечивает алгоритмические, технические и организационно-штатные меры защиты от указанного типа атак. Данное свойство содержится в [32, свойство G15].

Теоретическое исследование протокола может лишь проверить наличие алгоритмических мер, обеспечивающих защиту от данного класса атак, а также наличие эксплуатационной документации, содержащей описание технических и организационно-штатных мер защиты. В рамках предлагаемой методики представляется возможным получить лишь тривиальное численное значение показателя эффективности для данного свойства.

С 21. *Свойство защищённости от утечек по скрытым* (*логическим*) *каналам* подразумевает, что протокол содержит реализацию алгоритмических мер защиты от атак, реализуемых нарушителем путём применения непредусмотренных коммуникационных каналов передачи информации. Отметим, что современные транспортные протоколы, такие, как ESP, IPSec или ADTP FIOT, содержат ряд мер, предназначенных для обеспечения данного свойства.

Классификация угроз безопасности, реализуемых с использованием скрытых каналов, модель нарушителя и перечень мер защиты информационной системы от атак с использованием скрытых каналов должны разрабатываться на основе стандартов [33, 34]. Получение численных оценок показателей эффективности мер защиты от скрытых логических каналов выходит за рамки настоящей работы. Отдельные результаты в данном направлении получены в работах [35 – 37].

С 22. *Свойство защищённости от KCI-атак.* Под KCI-атакой (атакой имперсонификации при компрометации долговременного секретного ключа) понимается атака, при выполнении которой нарушитель, получивший доступ к долговременному секретному ключу участника, может выдать себя перед ним за любого другого участника в рамках текущей или будущей сессии выполнения протокола. Свойство считается выполненным, если KCI-атака невыполнима. Данное свойство описано в [10].

С 23. *Свойство защищённости от UKS-атак.* Под UKS-атакой понимается последовательность действий нарушителя, в результате которой законные авторизованные участники в процессе информационного взаимодействия вырабатывают общий ключ, но один из участников считает, что он выработал общий ключ с третьим участником (навязанным нарушителем в ходе выполнения протокола). При этом компрометации общего ключа как таковой не происходит, но нарушается требование аутентификации участников. Свойство считается выполненным, если подобная ситуация невозможна. Данное свойство описано в [38, 39]

С 24. *Свойство невозможности отказа от совершённых действий* представляет собой возможность проследить за всеми действиями участника взаимодействия.

Согласно Р 1323565.1.012-2017, разд. 6.1.14, данное свойство должно обеспечиваться средством криптографической защиты информации, реализующим криптографический протокол. Данное свойство содержится в [32, свойство G17].

С 25. *Свойство доказательства происхождения* заключается в неоспоримом доказательстве отправки сообщения. Данное свойство содержится в [32, свойство G18].

С 26. *Свойство доказательства доставки* заключается в неоспоримом доказательстве получения сообщения. Данное свойство содержится в [32, свойство G19].

С 27. *Свойство целостности множества состояний* (*криптографическое связывание состояний*) заключается в том, что все участники информационного взаимодействия после выполнения протокола (или его части) в рамках одного сеанса связи имеют одинаковое представление обо всех участниках этого сеанса и выполняемых ими ролях, а также о состоянии выполнения протокола. Данное свойство описано в рекомендациях [40 – 42].

Можно провести классификацию свойств безопасности по объектам применения, влияющим на безопасность исследуемого криптографического протокола (табл. 1).

Таблица 1

Свойства безопасности по объектам применения

Объект применения	Свойства безопасности
Аутентификация	С 1, С 2, С 5, С 6, С 7, С 9
Целостность	С 3, С 27
Ключи	С 8, С 10, С 11, С 12, С 13, С 14
Субъект взаимодействия	С 17, С 18, С 19, С 24
Атаки нарушителя	С 4, С 15, С 20, С 21, С 22, С 23
Защищаемые данные	С 16, С 25, С 26

Отметим, что для большинства используемых на практике криптографических протоколов все перечисленные свойства безопасности не могут быть выполнены одновременно. Примерная классификация свойств безопасности, которые могут обеспечиваться протоколами с различными целевыми назначениями, приведена в табл. 2 (см. также [30]).

Таблица 2

Свойства безопасности по целевому назначению криптографических протоколов

Класс протоколов	Свойства безопасности
Протоколы обеспечения целостности сообщения	С 4, С 10, С 13, С 3, С 22, С 23
Протоколы на основе цифровой подписи	С 1, С 2, С 8, С 9, С 11, С 12, С 17, С 22, С 23
Протоколы на основе цифровой подписи вслепую	С 1, С 5, С 11, С 12, С 19, С 22, С 23
Протокол односторонней аутентификации	С 1, С 2, С 8, С 9, С 11, С 12, С 17, С 22, С 23
Протокол взаимной аутентификации	С 1, С 2, С 6, С 8, С 9, С 11, С 12, С 17, С 22, С 23
Протокол групповой аутентификации	С 1, С 2, С 6, С 9, С 11, С 12, С 17, С 22, С 23
Протоколы конфиденциальной передачи	С 13, С 15, С 16, С 3, С 22, С 23,
Протоколы распределения ключей	С 1, С 2, С 8, С 9, С 22, С 23
Протоколы выработки общего ключа	С 1, С 2, С 4, С 8, С 9, С 10, С 11, С 22, С 23

Следует отметить, что на практике сложно отнести криптографический протокол к тому или иному классу, поскольку в большинстве случаев протоколы обеспечивают выполнение свойств, характерных для нескольких целевых назначений.

Для построения формальной модели свойств безопасности полезно разбить сформулированные свойства на два больших класса:

— базовые свойства, выполнение которых зависит от сложности решения математических задач, используемых в криптографических преобразованиях;
— производные свойства, являющиеся комбинацией базовых и других производных свойств.

Зависимость между свойствами безопасности представлена в табл. 3.

Таблица 3

Зависимости между свойствами безопасности

Свойство	Зависимость
С 1 — аутентификации участника протокола другим участником	Базовое
С 2 — аутентификации сообщения	С 1, С 3
С 3 — целостности сообщений	Базовое
С 4 — защиты от повторов	Базовое
С 5 - неявной аутентификации получателя	С 1, С 9
С 6 — групповой аутентификации	С 1, С 9
С 7 — аутентификации субъекта доверенной третьей стороной	С 1
С 8 — конфиденциальности ключа	Базовое
С 9 — аутентификации ключа	С 1, С 2, С 10, С 15
С 10 — подтверждения ключа	Базовое
С 11 — стойкости при компрометации производных ключей	С 13
С 12 — стойкости при компрометации ключа аутентификации	С 13
С 13 — формирования новых ключей	С 15
С 14 — защиты от навязывания ключевых значений	С 1, С 3
С 15 — защиты от навязывания параметров безопасности	С 1, С 2, С 3
С 16 — конфиденциальности	С 3, С 9, С 10
С 17 — инвариантности отправителя	С 1, С 9
С 18 — анонимности субъекта	Базовое
С 19 — анонимности субъекта для других участников	Базовое
С 20 — защищённости от атак «отказ в обслуживании»	Базовое
С 21 — защищённости от утечек по скрытым (логическим) каналам	Базовое
С 22 — защищённости от KCI-атак	С 1, С 9, С 10, С 12, С 13, С 14
С 23 — защищённости от UKS-атак	С 1, С 9, С 10, С 15, С 27
С 24 — невозможности отказа от совершенных действий	С 25, С 26, С 27
С 25 — доказательства происхождения	С 1, С 2, С 9
С 26 — доказательства доставки	С 9, С 10
С 27 — целостности множества состояний	С 1, С 17, С 22, С 23

Указанные зависимости позволяют свести исследование большого числа свойств безопасности к малому числу базовых свойств.

Отметим также, что в криптографических механизмах, представляющих собой совокупность нескольких протоколов, свойства безопасности могут наследоваться. Например, транспортный протокол, реализующий только шифрование и имитозащиту передаваемой информации, сам по себе не обеспечивает свойство аутентификации субъектов взаимодействия, однако он может его наследовать в случае использования ключевой информации, предварительно полученной в ходе протокола выработки ключей с аутентификацией участников. Подобная ситуация характерна для многих современных криптографических механизмов, включая TLS, IPSec или SP FIOT.

2. Формализация модели безопасности и моделирование свойств безопасности

Данное выше описание свойств безопасности носит качественный характер и не позволяет предъявить какой-либо способ определения показателя эффективности за-

щиты. Формализуем модель криптографического протокола и с её помощью опишем свойства безопасности как составные части исследуемого протокола.

Предлагаемая модель отталкивается от атомарного подхода к описанию протоколов и является моделью дискретной динамической системы. Состояния системы определяются в некоторые моменты времени и, согласно спецификации протокола, зависят от функций перехода в следующее состояние и данных, поступающих из канала связи.

Определение 2. Обозначим символом $\mathbb{B} = \{\texttt{false}, \texttt{true}\}$ булево множество, элементы которого принимают значения «истина» или «ложь». Символом $\mathbb{V}_\infty^*$ обозначим множество двоичных последовательностей произвольной конечной длины, включая последовательность длины 0 (обозначим её $\varnothing$). Символом $\mathbb{V}_\infty$ обозначим множество последовательностей ненулевой длины.

Пусть $a = (\alpha, \sigma)$ — некоторая абстрактная ячейка памяти. Будем говорить, что ячейка характеризуется:

1) значением $\alpha \in \mathbb{V}_\infty^*$; неопределённому значению ячейки a соответствует символ $\varnothing$;
2) подтверждением $\sigma \in \mathbb{B}$: $\sigma = \texttt{true}$ соответствует подтверждённому значению, $\sigma = \texttt{false}$ — неподтверждённому значению ячейки a.

Понятие «подтверждения» ячейки вводится для того, чтобы формализовать уверенность субъекта, владеющего ячейкой $a = (\alpha, \sigma)$, в том, что значение α, содержащееся в подтверждённой ячейке, истинно, а не вычислено ошибочно или подделано и/или навязано нарушителем.

Определение 3. Пусть $\{t_k\}_0^\infty$ — монотонно возрастающая последовательность натуральных чисел, где $k = 0, 1, \ldots, k_{\max}$ для некоторого натурального значения $k_{\max}$, определяемого спецификацией протокола.

Для субъекта A будем называть его состоянием в момент времени t_k множество ячеек памяти

$$A(t_k) = \{a_1, \ldots, a_{n(A)} : \; a_i = \big(\alpha_i(t_k), \sigma_i(t_k)\big)\},$$

значения и подтверждения которых могут изменяться с изменением момента времени. Количество ячеек памяти $n(A)$ зависит от роли субъекта и определяется спецификацией протокола.

У различных субъектов точные значения временных меток t_k могут отличаться. Можно считать, что t_0 — это время начальной инициализации состояния субъекта, а $t_1, t_2, \ldots$ — времена отправки и получения сообщений из канала связи.

В ряде случае, например в транспортных протоколах, реализуется только процедура отправки (получения) сообщений, а значение величины $k_{\max}$ может быть, формально, не ограничено. Однако существующие в Российской Федерации требования по ограничению объёма зашифровываемой на одном ключе информации (см. Р 1323565.1.012-2017) накладывают ограничения на число передаваемых сообщений и, как следствие, на количество возможных состояний $k_{\max}$.

Определение 4. Будем говорить, что модель протокола определена, если для каждого субъекта:

— определено множество ячеек памяти, образующих изменяемое в ходе выполнения протокола состояние;
— определено количество возможных состояний,

а также в соответствии со спецификацией протокола определены функции перехода из одного состояния в другое, позволяющие однозначно определить значение и подтвер-

ждение каждой ячейки памяти, т. е. для всех $k = 0, 1, \ldots, k_{\max}$ и всех $i = 1, \ldots, n(A)$ определены:

1) целые неотрицательные числа l_k;
2) отображения

$$\texttt{var}_{i,k}(x_1, \ldots, x_{n(A)+l_k}) : (\mathbb{V}_\infty^*)^{n(A)+l_k} \to \mathbb{V}_\infty^*,$$
$$\texttt{conf}_{i,k}(x_1, \ldots, x_{n(A)+l_k}) : (\mathbb{V}_\infty^* \times \mathbb{B})^{n(A)} \times (\mathbb{V}_\infty^*)^{l_k} \to \mathbb{B},$$

такие, что

$$\alpha_i(t_{k+1}) = \texttt{var}_{i,k}(\alpha_1(t_k), \ldots, \alpha_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)),$$
$$\sigma_i(t_{k+1}) = \texttt{conf}_{i,k}(a_1(t_k), \ldots, a_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)),$$

где $i = 1, \ldots, n(A)$, а значения $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$ рассматриваются как реализации l_k случайных величин, принимающих значения из $\mathbb{V}_\infty^*$ в момент времени t_k.

Полагаем, что в подавляющем большинстве случаев введённые отображения $\texttt{var}_{i,k}$ и $\texttt{conf}_{i,k}$ задаются тривиальными соотношениями

$$\begin{array}{lcl} l_k & = & 0, \\ \alpha_i(t_{k+1}) & = & \texttt{var}_{i,k}(\alpha_1(t_k), \ldots, \alpha_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)) \quad = \quad \alpha_i(t_k), \\ \sigma_i(t_{k+1}) & = & \texttt{conf}_{i,k}(a_1(t_k), \ldots, a_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)) \quad = \quad \sigma_i(t_k) \end{array}$$

при $k \in \{1, \ldots, k_{\max}\}$. В данном случае аргументы функций $\texttt{var}_{i,k}$ и $\texttt{cont}_{i,k}$, отличные от $\alpha_i(t_k)$ и $\sigma_i(t_k)$ соответственно, являются несущественными, т. е. не влияют на возвращаемое значение.

Для остальных случаев поясним смысл, который вкладывается в случайные значения $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$:

1) Случай $l_k = 0$ описывает автономное изменение состояния, которое субъект выполняет без какого-либо влияния извне. Такое изменение может использоваться для детализации спецификации протокола, например для изменения или подтверждения состояний элементов ключевой системы.
2) В ряде протоколов для аутентификации субъектов взаимодействия или выработки общей ключевой информации требуется генерация случайных значений; именно эти значения выступают в качестве величин $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$, изменяющих состояние участника протокола (для подавляющего числа протоколов в этом случае можно считать, что $l_k = 1$).
3) Во всех протоколах субъект взаимодействия обрабатывает данные, поступающие из канала связи и рассматриваемые как случайные величины $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$, принимающие значения в своей области определения.

Введённые отображения $\texttt{var}_{i,k}$ формально должны быть представлены в виде двухэтапной процедуры. На первом этапе должны быть определены отображения

$$\texttt{validate}_{j,k}(\xi_j(t_k)) : \mathbb{V}_\infty^* \to \mathbb{B}, \quad j = 1, \ldots, l(k),$$

определяющие принадлежность значения $\xi_j(t_k)$ заданной области определения. На втором этапе отображения $\texttt{var}_{i,k}$ должны реализовывать определяемые спецификацией протокола функции

$$\texttt{evaluate}_{i,k}(x_1, \ldots, x_{n(A)+l_k}) : (\mathbb{V}_\infty^* \times \mathbb{B})^{n(A)} \times (\mathbb{V}_\infty^*)^{l_k} \to \mathbb{V}_\infty^*,$$

изменяющие значение переменой a_i.

В случае, если одна из функций $\texttt{validate}_{j,k}$ возвращает `false`, то переход в следующее состояние должен блокироваться:

— если это указано в спецификации протокола, сообщение, содержащее хотя бы одно из таких значений $\xi_j(t_k)$, должно отбрасываться — для транспортных протоколов, либо протокол должен прекращать свое выполнение — для протоколов выработки ключа;
— если в спецификации протокола область допустимых значений для $\xi_j(t_k)$ не определена, то считаем, что это ошибка синтеза протокола, приводящая к нарушению его безопасности с вероятностью 1.

Если спецификацией протокола определена область допустимых значений для $\xi_j(t_k)$, то функции $\texttt{validate}_{j,k}$ не влияют на оценку безопасности. Далее мы выводим их из рассмотрения, полагая функции $\texttt{var}_{i,k}$ и $\texttt{evaluate}_{i,k}$ эквивалентными. Вместе с тем при практической реализации протокола функции $\texttt{validate}_{j,k}$ играют существенную роль, поскольку их отсутствие приводит к появлению уязвимостей в программном или аппаратном обеспечении средства защиты информации.

Функции $\texttt{conf}_{i,k}$ предназначены для подтверждения того, что значение $\alpha_i(t_k)$ является истинным, а величины, использованные для определения или формирования значения $\alpha_i(t_k)$, не были искажены или навязаны нарушителем в процессе обмена информацией по каналам связи.

Примером функции подтверждения могут служить функции проверки имитовставки или электронной подписи, которые позволяют гарантировать истинность подтверждаемых значений при помощи криптографических преобразований. При этом допускается, что одна функция $\texttt{conf}_{i,k}$ может подтверждать истинность нескольких ячеек $\alpha_{i_1}(t_k), \alpha_{i_2}(t_k), \ldots$, если все они одновременно являются аргументами функции $\texttt{conf}_{i,k}$, например, при проверке имитовставки проверяется истинность как сообщения, так и используемого секретного ключа. Далее всегда предполагается, что одним из аргументов функции $\texttt{conf}_{i,k}$ является некоторая ключевая информация, определяемая перед началом протокола (исходная ключевая информация) или вырабатываемая в ходе его выполнения.

Определение 5. Зафиксируем индекс $i \in \{1, \ldots, n(A)\}$ и момент времени t_k, $k \in \{1, \ldots, k_{\max}\}$.

1) Будем говорить, что значение ячейки $a_i = (\alpha_i(t_k), \sigma_i(t_k))$ подтверждено в момент времени t_k, если $\sigma_i(t_k) = \texttt{true}$.
2) Будем говорить, что значение ячейки a_i подтверждено косвенно, если $\sigma_i(t_k) = \texttt{false}$ и значение $\alpha_i(t_k)$ определено равенством

$$\alpha_i(t_k) = \texttt{var}_{i,k-1}(\alpha_{i_1}(t_{k-1}), \ldots, \alpha_{i_{s_i}}(t_{k-1})), \quad i_1, \ldots, i_{s_i} \in \{1, \ldots, n(A)\},$$

т. е. зависит только от существенных значений $\alpha_{i_1}(t_{k-1}), \ldots, \alpha_{i_{s_i}}(t_{k-1})$, таких, что

$$\sigma_{i_1}(t_{k-1}) = \cdots = \sigma_{i_{s_i}}(t_{k-1}) = \texttt{true}.$$

Можно предположить, что криптографический протокол является безопасным для субъекта A в момент времени t_k, где $k \in \{1, \ldots, k_{\max}\}$, если значения всех ячеек памяти состояния $A(t_k)$ являются либо подтверждёнными с использованием криптографических преобразований или выработаны самим субъектом, либо подтверждены косвенным образом. Однако, как мы покажем позднее, это предположение является необходимым, но не достаточным условием безопасности протокола.

В отличие от большинства других подходов к моделированию криптографических протоколов, предложенная модель ориентирована не на поиск возможных действий нарушителя и построение атак на протокол, а на поиск и построение графа зависимостей между всеми ячейками состояния субъекта, позволяющими проследить состояние ячеек памяти и подтвердить их «истинность», начиная с некоторого шага выполнения протокола (момента времени t_k).

Если спецификация протокола допускает существование неподтверждаемых переменных, то поиск возможных атак должен производиться с использованием автоматических верификаторов, таких, как Avispa [32], Scyther [43] или Proverif [44].

В рамках сформулированной модели вопрос о том, какие именно ячейки памяти должны входить в состояние субъекта, не конкретизируется. Уточнение перечня используемых ячеек памяти, а также построение зависимостей между ними должны производиться при анализе протокола (см. далее п. 4) и учитывать предъявляемые к протоколу свойства безопасности. Для упрощения данной процедуры проведём формализацию ряда базовых свойств безопасности (см. табл. 3).

2.1. Свойство аутентификации субъекта

Напомним, что с 2020 г. вопросы идентификации и аутентификации субъектов взаимодействия в Российской Федерации должны решаться с учётом ГОСТ Р 58833-2020 [45]. Согласно данному стандарту, при взаимодействии сторон с целью доступа к информации должны быть выполнены следующие процедуры:

1) первичная идентификация, в ходе которой регистрирующей стороной (доверенным центром) субъекту доступа должен присваиваться уникальный идентификатор $ID \in \mathbb{V}_\infty$;
2) вторичная идентификация, целью которой является опознавание субъекта доступа, т. е. предъявление субъектом присвоенного ранее идентификатора ID при попытке доступа к информации; выполнение вторичной идентификации производится субъектом, предоставляющим доступ к информации, — в нашем случае другим участником взаимодействия;
3) аутентификация субъекта доступа, в которую должны входить действия по проверке подлинности субъекта доступа, а также принадлежности субъекту доступа предъявленного идентификатора и аутентификационной информации.

Аутентификация субъекта доступа, согласно [45], может осуществляться с использованием нескольких факторов:

— фактора знания определённой информации, например секретного ключа или пароля;
— фактора владения определённым предметом;
— биометрическим фактором, описывающим определённые характеристики аутентифицируемого субъекта.

Поскольку при разработке криптографических протоколов принято использовать только фактор знания ключевой информации, мы должны дополнить положения стандарта [45] и ввести в использование понятие секретного ключа аутентификации, однозначно связанного с уникальным идентификатором субъекта. Далее будем обозначать множество ключей аутентификации символом $\mathbb{K}_a$.

Для использования асимметричной ключевой системы введём в рассмотрение множество ключей проверки кода аутентификации, которое будем обозначать символом $\mathbb{K}_c$, а также однонаправленную функцию $h : \mathbb{K}_a \to \mathbb{K}_c$, такую, что для любого $K_a \in \mathbb{K}_a$ выполнено условие $h(K_a) \in \mathbb{K}_c$. Под однонаправленной функцией h будем

подразумевать эффективно вычислимую функцию, для которой неизвестен эффективный алгоритм обращения [46, с. 79]. Для симметричной ключевой системы будем полагать, что выполнено равенство $\mathbb{K}_a = \mathbb{K}_c$, а h есть тривиальное отображение, не изменяющее значение своего аргумента.

Механизм связывания идентификатора субъекта с его ключом аутентификации зависит от схемы выработки ключей аутентификации:

— для асимметричных ключевых схем связывание происходит путём включения идентификатора и другой аутентифицирующей информации в состав сертификата открытого ключа участника протокола;
— для симметричных схем уникальные идентификаторы $ID_{A_1}, ID_{A_2}, \ldots ID_{A_r}$, определяемые для некоторого целого $r \geqslant 2$, используются при выработке общей ключевой информации для группы субъектов $A_1, A_2, \ldots, A_r$; примером такой схемы является ключевая система, регламентированная Р 1323565.028-2019.

Определение 6. Будем говорить, что криптографический протокол обеспечивает свойство аутентификации субъекта B, выполняемой субъектом A (свойство С1), если:

1) с субъектом B связан идентификатор ID_B;
2) для субъекта B определены ключ аутентификации субъекта $K_a \in \mathbb{K}_a$ и ключ проверки кода аутентификации $K_c \in \mathbb{K}_c$, однозначно связанные с идентификатором ID_B;
3) для некоторого натурального m определены функции выработки кода аутентификации $\texttt{mac} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ и проверки кода аутентификации $\texttt{conf} : \mathbb{K}_c \times \mathbb{V}_\infty^* \times \mathbb{V}_m \to \mathbb{B}$, такие, что $\texttt{conf}\,(K_c, M, \texttt{mac}(K_a, M)) = \texttt{true}$ для любых значений $K_a, K_c, M \in \mathbb{V}_\infty^*$;
4) субъекту A известны идентификатор ID_B субъекта B и подтверждённое значение ключа проверки кода аутентификации K_c;
5) в состав протокола входит последовательность шагов, представленная на рис. 2; символом $\in_R$ обозначается выбор случайного элемента из заданного множества;

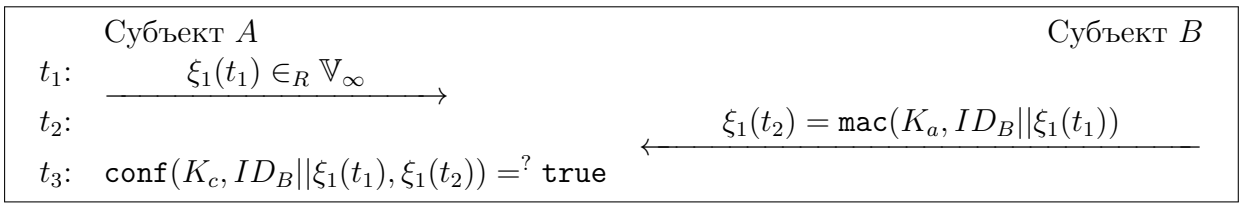

Рис. 2. Протокол аутентификации субъекта

6) истинно условие проверки на шаге t_3.

Данное определение согласуется как с симметричными, так и с асимметричными ключевыми системами — в качестве функции mac могут быть использованы как алгоритмы выработки имитовставки, так и алгоритмы выработки электронной подписи. Для общности изложения далее будем считать, что mac есть отображение с конечным числом аргументов, существенно зависящее от ключа аутентификации K_a, идентификатора субъекта B и случайного значения $\xi_1(t_1)$ и удовлетворяющее требованиям, предъявляемым к ключевым криптографическим функциям хеширования [46].

Начальное множество состояний субъекта A, выполняющего процесс аутентификации субъекта B, может быть определено следующим образом:

$$A(t_0) = \{a_1(t_0) = (K_c, \texttt{true}),\ a_2(t_0) = (ID_B, \texttt{false}),\ a_3 = (\varnothing, \texttt{false})\}.$$

Здесь ячейка a_1 соответствует ключу проверки кода аутентификации субъекта B, ячейка a_2 — идентификатору субъекта B, ячейка a_3 — вырабатываемому субъектом A случайному значению, используемому при проверке кода аутентификации. В следующие моменты времени состояние субъекта A имеет вид

$$\begin{array}{lll} A(t_1) = \{a_1(t_1) = (K_c, \texttt{true}), & a_2(t_1) = (ID_B, \texttt{false}), & a_3(t_1) = (\xi_1(t_1), \texttt{true})\}, \\ A(t_2) = \{a_1(t_2) = (K_c, \texttt{true}), & a_2(t_2) = (ID_B, \sigma), & a_3(t_2) = (\xi_1(t_1), \texttt{true})\}, \end{array}$$

где $\sigma = \texttt{conf}(K_c, ID_B||\xi_1(t_1), \xi_1(t_2))$, или может быть описано следующими нетривиальными функциями перехода:

$$\alpha_3(t_1) = \xi_1(t_1),\ \sigma_3(t_1) = \texttt{true},\ \sigma_2(t_2) = \texttt{conf}(\alpha_1(t_2), \alpha_2(t_2)||\alpha_3(t_2), \xi_1(t_2)).$$

Сделаем несколько замечаний к определению 6:

1) В рассмотренном протоколе (см. рис. 2) ключ проверки кода аутентификации K_c является исходной ключевой информацией для субъекта A. Поскольку именно эта информация обеспечивает аутентификацию субъекта B, её значение должно быть подтверждено до начала выполнения протокола, например, с помощью организационно-технических мер, удостоверяющего центра или в рамках другого протокола.
2) Из определения следует, что свойство аутентификации субъекта выполнено только для протоколов, включающих в себя взаимодействие субъектов (отправку и получение сообщений). Транспортные протоколы, предусматривающие только отправку сообщений от одного субъекта к другому, данному свойству не удовлетворяют. Вместе с тем использование в таких протоколах ключевой информации, владелец которой аутентифицирован ранее иным способом, позволяет говорить о наследовании свойства С 1.
3) Включение идентификатора ID_B в состав сообщения $\xi_1(t_2)$=$\texttt{mac}(K_a, ID_B||\xi_1(t_1))$ является принципиальным при использовании симметричной ключевой системы. В случае исключения идентификатора ID_B не представляется возможным предъявить алгоритмический способ различения того, кто из субъектов взаимодействия является автором пары сообщений $(\xi_1(t_1), \texttt{mac}(K_a, \xi_1(t_1)))$ (это следует из совпадения ключей аутентификации у обоих субъектов). Для асимметричной ключевой системы исключение идентификатора ID_B не является критичным, поскольку субъекты имеют различные ключи аутентификации.

Отметим, что на настоящий момент в Российской Федерации действует только морально устаревший стандарт ГОСТ Р ИСО/МЭК 9594-8-98 [47], регламентирующий процедуры аутентификации с использованием фактора знания секретного ключа. При этом определению 6 соответствует лишь часть процедур «строгой» аутентификации из [47], см. разд. 10. Современные стандарты серии ISO/IEC 9798 (части 1–6) в большинстве своём соответствуют определению 6. Протоколы из ISO/IEC 9798-5:2009 и ISO/IEC 9798-6:2010 могут формально не соответствовать этому определению и в случае необходимости их применения на территории Российской Федерации должны пройти дополнительное исследование на соответствие рассматриваемой модели.

2.2. Свойство целостности сообщений

В большинстве некриптографических протоколов для защиты от случайных искажений данные передаются вместе со своими кодами целостности, выработанными с помощью сжимающих отображений, таких, как Fletcher16 [48], CRC32 [49] и т. п., а также бесключевых криптографических функций хэширования, например функции «Стрибог» [50]. Используемая нами модель нарушителя делает применение таких функций бесполезным для защиты от преднамеренных искажений, а для обеспечения целостности, так же как и в п. 2.1, приходится использовать сжимающие преобразования, зависящие от секретного ключа.

Определение 7. Будем говорить, что криптографический протокол обеспечивает свойство целостности сообщения $M \in \mathbb{V}_\infty^*$, отправляемого субъектом B субъекту A (свойство С 3), если:

1) для субъекта B определены ключ аутентификации субъекта $K_a \in \mathbb{K}_a$ и ключ проверки кода аутентификации $K_c \in \mathbb{K}_c$;
2) субъекту A известно подтверждённое значение ключа проверки кода аутентификации субъекта K_c;
3) для некоторого натурального m определены функции выработки кода аутентификации $\mathtt{mac} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ и проверки кода аутентификации $\mathtt{conf} : \mathbb{K}_c \times \mathbb{V}_\infty^* \times \mathbb{V}_m \to \mathbb{B}$, такие, что $\mathtt{conf}\,(K_c, M, \mathtt{mac}(K_a, M)) = \mathtt{true}$ для любых значений $K_a, K_c \in \mathbb{K}$, $M \in \mathbb{V}_\infty^*$;
4) в состав протокола входит последовательность шагов, представленная на рис. 3;

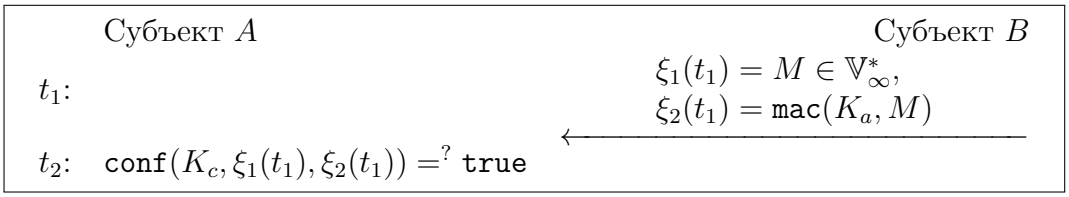

Рис. 3. Протокол подтверждения целостности сообщения

5) истинно условие проверки на шаге t_2.

Начальное множество состояний субъекта A, выполняющего процесс получения сообщений от субъекта B, может быть определено следующим образом:

$$A(t_0) = \{a_1(t_0) = (K_c, \mathtt{true}),\ a_2(t_0) = (\varnothing, \mathtt{false}),\ a_3 = (\varnothing, \mathtt{false})\}.$$

Здесь ячейка a_1 соответствует ключу проверки кода аутентификации субъекта B, ячейка a_2 — получаемому из канала связи сообщению, ячейка a_3 — коду целостности получаемого сообщения. В следующий момент времени состояние субъекта A имеет вид

$$A(t_1) = \{a_1(t_1) = (K_c, \mathtt{true}),\ a_2(t_1) = (\xi_1(t), \sigma),\ a_3(t_1) = (\xi_2(t_1), \sigma)\},$$

где $\sigma = \mathtt{conf}(K_c, \xi_1(t_1), \xi_2(t_1))$.

Отметим, что, как и в случае аутентификации субъекта, выполнение свойства целостности существенным образом зависит от того, подтверждена ли исходная ключевая информация — ключ проверки кода целостности K_c.

2.3. Свойство аутентификации сообщения

Свойство аутентификации сообщения (свойство С 2) является производным и следует из комбинации свойств аутентификации субъекта (свойство С 1, п. 2.1) и целостности передаваемых сообщений (свойство С 3, п. 2.2).

Определение 8. Криптографический протокол обеспечивает выполнение свойства аутентификации субъектом A сообщения $M \in \mathbb{V}_\infty^*$, отправленного субъектом B (свойство С 2), если в условиях определений 6 и 7 имеет место:

1) протокол содержит последовательность шагов, приведённую на рис. 4;

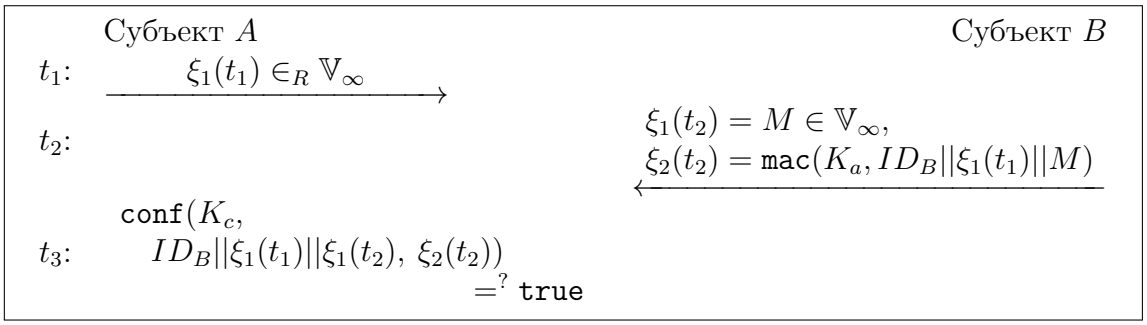

Рис. 4. Протокол аутентификации сообщения

2) истинно условие проверки на шаге t_3.

В данном определении мы совместили процедуры проверки кода аутентификации, выполняемые для проверки свойств аутентификации субъекта B и целостности сообщения M.

Следует отметить, что для выполнения свойства аутентификации нескольких отправляемых субъектом B сообщений (с целью снижения объёма передаваемой в канал связи информации) сообщение $\xi_1(t_1)$ может отправляться единожды, а значение $ID_B||\xi_1(t_1)$ заменяться на результат применения некоторого сжимающего отображения, известного субъектам A и B.

2.4. Свойство защиты от навязывания параметров безопасности

Свойство защиты от навязывания параметров безопасности (С 15) также является производным и основано на выполнении следующих свойств:

— аутентификации субъекта (свойство С 1),
— целостности сообщений (свойство С 3) и, как следствие,
— аутентификации сообщения (свойство С 2).

Достаточно часто в рамках одной сессии протокола между субъектами взаимодействия осуществляется согласование криптографических параметров, прямо или косвенно влияющих на безопасность передаваемых данных. В качестве таких параметров могут выступать алгоритмы блочного шифрования, функции хеширования, параметры циклической абелевой группы, в которых производится выработка производных ключей и т. п. Вмешательство нарушителя в процесс согласования параметров может привести к компрометации вырабатываемой в ходе выполнения протокола ключевой информации. Наиболее ярким примером являются атаки на понижение версии протокола TLS 1.2 [51].

Определение 9. Криптографический протокол обеспечивает выполнение свойства защиты от навязывания параметров безопасности $P \in \mathbb{V}_\infty$ субъекту A (свойство С 15), если сообщение, в котором передаются параметры безопасности для субъекта A, удовлетворяет свойству аутентификации сообщения (свойство С 2).

На практике согласование параметров безопасности выполняется в режиме «запрос — ответ», т. е. субъект A формирует перечень доступных для него параметров, а субъект B выбирает из них те, что будут использоваться в дальнейшем. Интепретируя параметры безопасности как строки из $\mathbb{V}_\infty$, получаем, что субъект A направляет субъекту B множество строк $\{S_1, \ldots, S_r\}$, а субъект возвращает одну строку S_i, $i \in \{1, \ldots, r\}$, из полученного множества. Тогда можем считать, что протокол, обеспечивающий свойство защиты от навязывания параметров безопасности, должен содержать последовательность шагов, представленную на рис. 5.

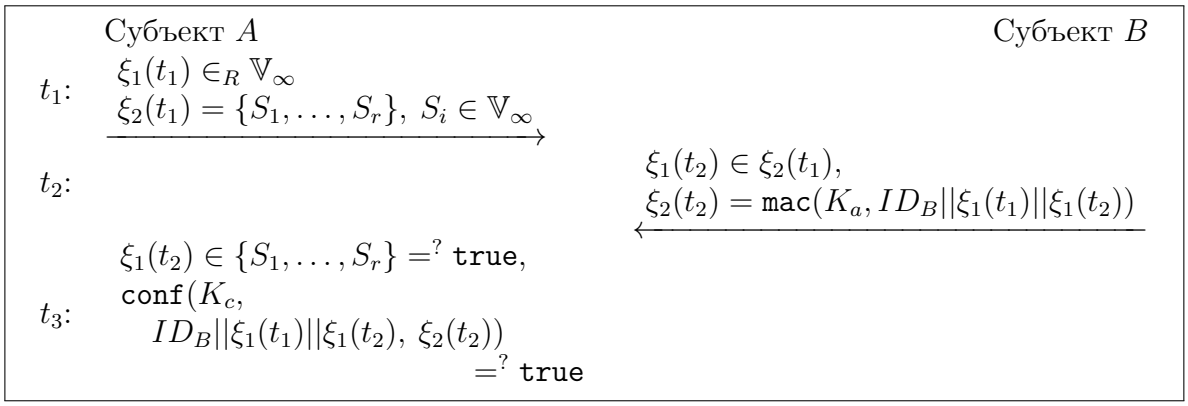

Рис. 5. Протокол защиты от навязывания параметров безопасности

В данном протоколе используется определяемая, как и ранее, исходная ключевая информация K_a, K_c. Субъект A сначала проверяет, что выбранный субъектом B параметр безопасности $\xi_1(t_2)$ принадлежит запрашиваемому множеству, а затем проверяет истинность кода аутентификации сообщения, содержащего $\xi_1(t_2)$. Отметим, что функции `mac` и `conf` должны быть зафиксированы до начала согласования параметров безопасности.

Примером описанного механизма может служить этап установления соединения в протоколе TLS 1.3, в котором все критически важные параметры передаются при помощи сообщения, подписанного электронной подписью сервера (субъекта B).

2.5. Свойства подтверждения и аутентификации ключа

Как мы видели ранее, для обеспечения свойств безопасности необходимо использование подтверждённой исходной ключевой информации K_c. В случае асимметричной ключевой системы подтверждение ключа K_c производится путём проверки электронной подписи удостоверяющего центра, выдавшего сертификат ключа K_c. В случае симметричной ключевой системы подтверждение обеспечивается организационно-штатными мерами доставки ключевой информации до субъектов взаимодействия.

Для производной ключевой информации, вырабатываемой в рамках криптографического протокола, требуется обеспечить подтверждение непосредственно в ходе взаимодействия субъектов.

Пусть $K_A \in \mathbb{K}_a$ — значение, выработанное субъектом A, а $K_B \in \mathbb{K}_a$ — значение, выработанное субъектом B. Теперь субъект A должен получить подтверждение того, что

значение K_A совпадает со значением K_B, т. е. проверить, что для некоторой функции f выполнено равенство

$$f(K_A, M) =^? f(K_B, M), \quad M \in \mathbb{V}_\infty,$$

в котором правая часть вычислена субъектом B, а левая часть — субъектом A. В качестве функции f может выступать, например, режим блочного шифрования или алгоритм выработки имитовставки.

Наиболее простой протокол, реализующий подтверждение субъектом A ключа K_A, изображён на рис. 6.

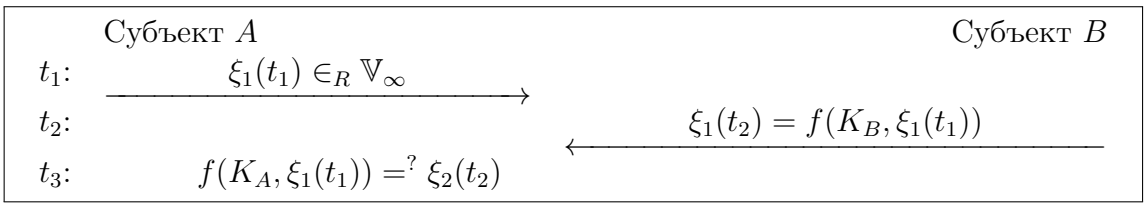

Рис. 6. Простой протокол подтверждения ключа

В данном протоколе субъект B с использованием подтверждаемого значения K_B преобразует случайное сообщение $\xi_1(t_1)$, а субъект A проверяет корректность результата преобразования с помощью подтверждаемого значения K_A. Сделаем ряд замечаний к протоколу рис. 6:

— Легко видеть, что сообщение $\xi_1(t_2)$ не удовлетворяет свойству аутентификации сообщений. Субъект A получает подтверждение, что он вычислил значение K_A правильно, но того, кто ему это подтверждение направил, субъект A идентифицировать не может. Тем самым у нарушителя появляется потенциальная возможность навязать субъекту A ложное значение ключа K_A.

— Другим недостатком предложенного протокола является возможность накопления нарушителем пар открытый/шифрованный текст $(\xi_1(t_1), f(K_B, \xi_1(t_1)))$ и их использование в дальнейшем для реализации алгоритмических методов определения секретного значения K_B либо для последующего навязывания ложных, но корректно зашифрованных значений. Это приводит к необходимости применять на этапе подтверждения ключа преобразование f, отличное от того, что будет использовано при взаимодействии субъектов.

Для учёта второго замечания можно модифицировать протокол рис. 6 и обмениваться только зашифрованными с помощью преобразования f сообщениями. Пример такой модификации приведён на рис. 7.

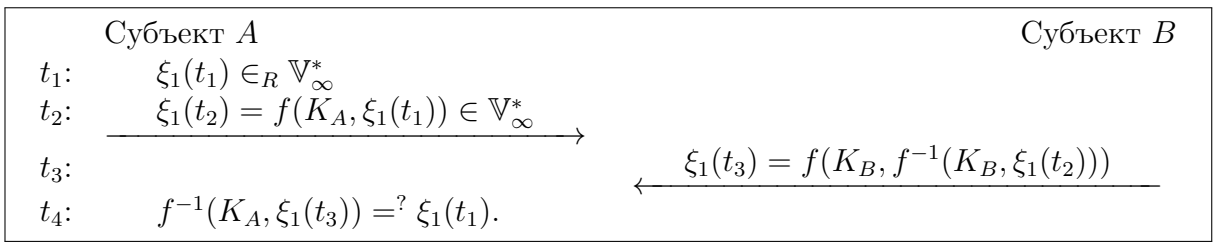

Рис. 7. Уязвимый протокол подтверждения ключа

В данном протоколе нарушитель не может накапливать пары открытый/шифрованный текст $(\xi_1(t_1), f(K_A, \xi_1(t_1)))$, поскольку значение $\xi_1(t_1)$ не передаётся субъектом A в канал связи. Предполагается также, что субъект B с помощью подтверждаемого ключа K_B сначала расшифровывает случайное сообщение $\xi_1(t_1)$, а потом повторно зашифровывает его (такую последовательность действий субъекта B можно назвать «расшифрование и зашифрование»).

Однако протокол уязвим, поскольку нарушитель может реализовать атаку «отражением сообщений» — перехватить сообщение $\xi_1(t_2) = f(K_A, \xi_1(t_1))$ и направить его обратно субъекту A вместо сообщения $\xi_1(t_3)$. После этого условие проверки на шаге t_4 всегда будет истинно, поскольку $f^{-1}(K_A, f(K_A, \xi_1(t_1))) = \xi_1(t_1)$.

Реализация атаки «отражением сообщений» становится возможной в силу следующих причин:

1) Построим формальную модель состояний субъекта A. Начальное множество состояний может быть определено следующим образом:

$$A(t_0) = \{a_1(t_0) = (K_A, \texttt{false}),\ a_2(t_0) = (\varnothing, \texttt{false})\}.$$

Здесь ячейка a_1 соответствует подтверждаемому ключу K_A, а ячейка a_2 — вырабатываемому случайному сообщению. Тогда последовательность состояний субъекта A описывается следующими нетривиальными функциями:

$$\alpha_2(t_1) = \xi_1(t_1), \quad \sigma_2(t_1) = \texttt{true}, \quad \sigma_1(t_4) = (\texttt{f}(\alpha_1(t_3), \xi_1(t_3)) =^{?} \alpha_2(t_3)).$$

Легко видеть, что содержимое ячейки a_1 подтверждается значением функции $f(\alpha_1(t_3), \xi_1(t_3)) =^{?} \alpha_2(t_3)$, не зависящим от какой-либо исходной ключевой информации.

2) С точки зрения субъекта A значение $\xi_1(t_3)$ рассматривается как реализация некоторой случайной величины. При этом ожидается, что вероятность угадывания нарушителем случайного значения $\xi_1(t_3)$, такого, что $\sigma_1(t_4)$ примет истинное значение, будет минимальной. Вместе с тем значение $\xi_1(t_3) = \xi_1(t_2)$ передаётся субъектом A в открытом виде. Это позволяет нарушителю перехватить его, отправить обратно субъекту A и с вероятностью 1 быть уверенным в том, что $\sigma_1(t_4) = \texttt{true}$.

Протокол рис. 7 иллюстрирует сделанное ранее высказывание (см. примечание к определению 5 на с. 46) о том, что подтверждение всех ячеек состояния субъекта является лишь необходимым условием безопасности протокола. Дополнительно должны рассматриваться вероятности подделки поступающих из канала связи значений, а также, в общем случае, и трудоёмкости алгоритмов подделки.

Защитой от атаки «отражением сообщений» является применение некоторого известного субъектам A и B преобразования h к неизвестному для нарушителя сообщению $\xi_1(t_1)$, т. е. вычисление равенства

$$\xi_1(t_3) = f(K_B, h(f^{-1}(K_B, \xi_1(t_2))))$$

(такую последовательность действий субъекта B можно назвать «расшифрование, преобразование и зашифрование»).

Если преобразование h отлично от преобразования f, является однонаправленным для нарушителя и не позволяет ему по значению $\xi_1(t_3) = h(x)$ определить значение аргумента x, то повторное применение преобразования f представляется излишним. В качестве однонаправленного преобразования h может выступать, например, бесключевая функция хеширования.

Определение 10. Будем говорить, что криптографический протокол обеспечивает для субъекта A свойство подтверждения факта обладания субъектом B ключом $K_B \in \mathbb{K}_a$ (свойство С 10), если:

1) субъект A обладает ключом $K_A \in \mathbb{K}_a$, для которого подтверждается выполнение равенства $K_A = K_B$;
2) определена зависящая от секретного ключа функция $f : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_\infty^*$, для которой определена обратная функция $f^{-1} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_\infty^*$, такая, что для любого сообщения $M \in \mathbb{V}_\infty$ равенство
$$f^{-1}(K_A, f(K_B, M)) = M$$
справедливо, когда $K_A, K_B \in \mathbb{K}_a$ и $K_A = K_B$;
3) для некоторого $m \in \mathbb{N}$ определена однонаправленная функция $h : \mathbb{V}_\infty \to \mathbb{V}_m$;
4) в состав протокола входит последовательность шагов, представленная на рис. 8;

	Субъект A		Субъект B
t_1:	$\xi_1(t_1) \in_R \mathbb{V}_\infty^*$		
t_2:	$\xi_1(t_2) = f(K_A, \xi_1(t_1)) \in \mathbb{V}_\infty^*$	$\xrightarrow{\qquad\qquad}$	
t_3:		$\xleftarrow{\qquad\qquad}$	$\xi_1(t_3) = h(f^{-1}(K_B, \xi_1(t_2)))$
t_4:	$th(\xi_1(t_1)) =^? \xi_1(t_3)$		

Рис. 8. Протокол подтверждения ключа

5) истинно условие проверки на шаге t_4.

Следует отметить, что условие существования (эффективно вычислимого субъектом B) обратного преобразования f^{-1} является необходимым, так как в противном случае вычисление сообщения $\xi_1(t_1) = f^{-1}(K, \xi_1(t_2))$ невозможно.

Используя в качестве преобразования h функцию вычисления кода аутентификации `mac`, можно добиться зависимости значения $\sigma_1(t_4)$ от исходной ключевой информации. В этом случае необходимо использовать равенство
$$\xi_1(t_3) = \texttt{mac}(K_a, f^{-1}(K_B, \xi_1(t_2)))).$$

Учитывая, что сообщение $\xi_1(t_3)$ должно удовлетворять свойству аутентификации сообщений (свойство С 2), дадим ещё одно определение.

Определение 11. Будем говорить, что криптографический протокол обеспечивает для субъекта A свойство аутентификации принадлежащего субъекту B ключа $K_B \in \mathbb{K}_a$ (свойство С 9), если:

1) с субъектом B связан идентификатор ID_B;
2) для субъекта B определены ключ аутентификации субъекта $K_a \in \mathbb{K}_a$ и ключ проверки кода аутентификации $K_c \in \mathbb{K}_c$, однозначно связанные с идентификатором ID_B;
3) субъекту A известны идентификатор ID_B субъекта B и подтверждённое значение ключа проверки кода аутентификации K_c;
4) субъект A обладает ключом $K_A \in \mathbb{K}_a$, для которого подтверждается выполнение равенства $K_A = K_B$;

5) для некоторого натурального m определены функции выработки кода аутентификации $\mathtt{mac} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ и проверки кода аутентификации $\mathtt{conf} : \mathbb{K}_c \times \mathbb{V}_\infty^* \times \mathbb{V}_m \to \mathbb{B}$, такие, что

$$\mathtt{conf}\,(K_c, M, \mathtt{mac}(K_a, M)) = \mathtt{true},$$

для любых значений $K_a, K_c, M \in \mathbb{V}_\infty^*$;

6) определена зависящая от секретного ключа функция $f : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_\infty^*$, для которой определена обратная функция $f^{-1} : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_\infty^*$, такая, что для любого сообщения $M \in \mathbb{V}_\infty$ равенство

$$f^{-1}(K_A, f(K_B, M)) = M$$

справедливо, когда $K_A, K_B \in \mathbb{K}_a$ и $K_A = K_B$;

7) в состав протокола входит последовательность шагов, представленная на рис. 9;

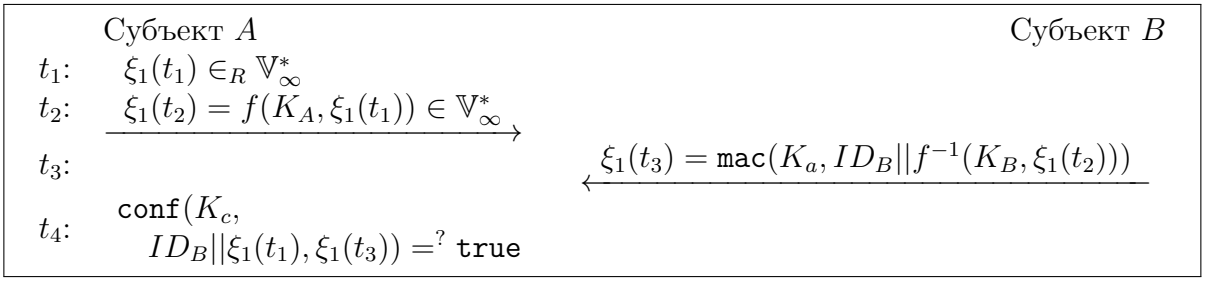

Рис. 9. Протокол подтверждения ключа

8) истинно условие проверки на шаге t_4.

Отметим, что формально следуя определению 8, можно было бы определить множество отправляемых субъектом B сообщений следующим образом:

$$\xleftarrow{\begin{array}{c}\xi_1(t_3) = \mathtt{mac}(K_B, \mathtt{f}^{-1}(K_B, \xi_1(t_2))),\\ \xi_2(t_3) = \mathtt{mac}(K_a, ID_B||\xi_1(t_2)||\xi_1(t_3)).\end{array}}$$

Предложенный на рис. 8 вариант достигает той же цели с меньшей длиной данных, передаваемых в канал связи.

Отметим, что если функции f, mac и conf согласуются в ходе выполнения протокола, то для протокола должно быть выполнено свойство защиты от навязывания параметров (свойство С 15).

2.6. Свойство конфиденциальности ключа

Свойство конфиденциальности ключа является базовым и применяется как к исходной ключевой информации, так и к сессионным (производным) ключам, вырабатываемым в ходе выполнения протокола. Формальное определение свойства конфиденциальности ключа тесно связано с показателями эффективности, рассматриваемыми далее в п. 3.2.

Определение 12. Пусть заданы $K_a \in \mathbb{K}_a$ — исходная ключевая информация, $h : \mathbb{K}_a \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ — некоторая однонаправленная функция и множество пар $(\xi_{i_k}(t_k), h(K_a, \xi_{i_k}(t_k)))$, перехваченных нарушителем в ходе выполнения одной или нескольких сессий протокола.

Будем говорить, что протокол обеспечивает конфиденциальность исходной ключевой информации (свойство С 8), если нарушитель не может определить значение K_a с вероятностью, большей чем π_0 (см. определение 1), и трудоёмкостью, не превосходящей некоторое значение Q_0.

Легко видеть, что данное определение может быть расширено на случай нескольких однонаправленых функций $h_1, h_2, \ldots$

Отличие производной ключевой информации от исходной заключается в том, что она вырабатывается в ходе выполнения протокола. В случае, когда выработка происходит с использованием значений, передаваемых между субъектами взаимодействия, протокол должен реализовывать механизмы защиты передаваемых значений от подделки и навязывания нарушителем.

Определение 13. Пусть заданы $K_a \in \mathbb{K}_a$ — исходная ключевая информация, отображение $\mathtt{var} : \mathbb{K}_a \times \mathbb{V}_\infty^* \times \cdots \times \mathbb{V}_\infty^* \to \mathbb{K}_a$, используемое для выработки производной ключевой информации, и $h : \mathbb{K} \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ — некоторая однонаправленная функция.

Будем говорить, что протокол обеспечивает конфиденциальность производной ключевой информации K_A (свойство С 8), если выполнены следующие условия:

1) величина K_A (в общем виде) определяется равенством

$$K_A = \mathtt{var}(K_a, \xi_{i_1}(t_{k_{i_1}}), \ldots, \xi_{i_l}(t_{k_{i_l}}),\ \beta_{j_1}, \ldots, \beta_{j_r}), \tag{1}$$

   где $\beta_{j_1}, \ldots, \beta_{j_r}$ определены равенствами $\xi_{j_s}(t_{k_{j_s}}) = h(\beta_{j_s})$, $s = 1, \ldots, r$, а величины $\xi_{i_1}(t_{k_{i_1}}), \ldots, \xi_{i_l}(t_{k_{i_l}})$ и $\xi_{j_1}(t_{k_{j_1}}), \ldots, \xi_{j_r}(t_{k_{j_r}})$ передаются между субъектами взаимодействия в ходе выполнения протокола (они могут перехватываться нарушителем);
2) в равенстве (1) либо переменные $K_a, \xi_{i_1}(t_{k_{i_1}}), \ldots, \xi_{i_l}(t_{k_{i_l}})$, либо переменные $\beta_{j_1}, \ldots, \beta_{j_r}$ могут являться несущественными;
3) если переменные $\xi_{i_1}(t_{k_{i_1}}), \ldots, \xi_{i_l}(t_{k_{i_l}})$ являются существенными, то они должны передаваться в составе сообщений, для которых выполнено свойство аутентификации сообщений (свойство С 2, п. 2.3);
4) если переменные $\beta_{j_1}, \ldots, \beta_{j_r}$ являются существенными, то они не могут быть определены нарушителем с вероятностью, большей чем величина π_0, и трудоёмкостью, не превосходящей некоторое значение Q_0;
5) после выработки производной ключевой информации K_A она должна использоваться таким образом, чтобы удовлетворять определению 12.

2.7. Свойство конфиденциальности

Согласно ГОСТ Р ИСО/МЭК 27033-1:2011, угроза нарушения конфиденциальности передаваемой информации является одной из основных угроз при обеспечении безопасности сетей связи. Однако свойство конфиденциальности (свойство С 16) не является базовым и выполняется только при выполнении совокупности рассмотренных ранее свойств.

Во-первых, используемый для шифрования информации ключ, являющийся, как правило, производной ключевой информацией, должен быть неизвестен нарушителю, т. е. удовлетворять свойству конфиденциальности ключа (свойство С 8).

Во-вторых, этот ключ должен удовлетворять свойству аутентификации ключа (свойство С 9). Это позволит получающему сообщения субъекту быть уверенным в том, что он не только получает сообщения, зашифрованные на том самом ключе, который

используется для их расшифрования, но и в том, что отправителем этих сообщений является аутентифицированный субъект взаимодействия.

В-третьих, для зашифрования информации, передаваемой в информационных системах, попадающих под действие нормативного регулирования, допускается применять только алгоритмы, входящие в национальную систему стандартизации Российской Федерации. Перечень допустимых алгоритмов шифрования определяется согласно ГОСТ Р 34.12-2015, ГОСТ Р 34.13-2015, а также ряда рекомендаций по стандартизации.

Схема зависимостей свойства конфиденциальности приведена на рис. 10.

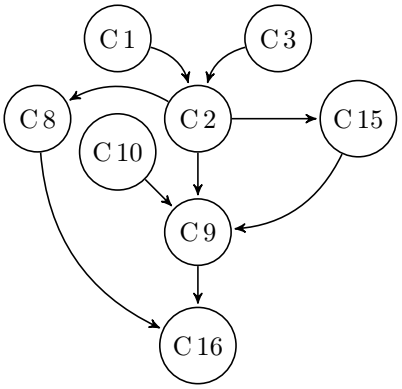

Рис. 10. Схема зависимостей свойства конфиденциальности

Далее рассмотрим ряд свойств, нарушение которых может привести к появлению негативных эффектов, не предполагаемых спецификацией криптографического протокола.

2.8. Свойство целостности множества состояний

С целью защиты от атак, использующих для компрометации одной сессии протокола данные, перехваченные в ходе выполнения другой сессии, рассмотрим свойство целостности множества состояний (свойство С 27). Пусть

$$A(t_0) = \{a_1, \ldots, a_{n(A)} : \; a_i = \big(\alpha_i(t_0), \sigma_i(t_k)\big)\}$$

— начальное состояние субъекта A. Будем считать, что для некоторого натурального $n_1(A)$, такого, что $1 \leqslant n_1(A) < n(A)$, ячейки $a_1, \ldots, a_{n_1(A)}$ содержат исходную ключевую информацию, а также любые другие значения, подтверждённые до начала выполнения протокола, т. е.

$$\sigma_i(t_0) = \texttt{true}, \quad i = 1, \ldots, n_1(A).$$

Также будем считать, что для некоторого натурального $n_2(A)$, такого, что $n_1(A) < n_2(A) < n(A)$, ячейки $a_{n_1(A)+1}, \ldots, a_{n_2(A)}$ содержат случайные значения, вырабатываемые субъектом A с использованием генератора случайных чисел (ГСЧ), т. е. найдутся такие временные метки t_{k_i}, что

$$\sigma_i(t_{k_i}) = \texttt{true}, \quad i = n_1(A) + 1, \ldots, n_2(A).$$

Определение 14. Будем говорить, что криптографический протокол удовлетворяет свойству целостности множества состояний (свойство С 27) для субъекта A, если найдется такая временная метка t_{k_0}, что для всех $t_k \geqslant t_{k_0}$ выполнено

$$\sigma_i(t_k) = \mathtt{conf}(a_1(t_{k-1}), \ldots, a_{n_2(A)}(t_{k-1}), \ldots), \quad i = n_2(A) + 1, \ldots, n(A), \tag{2}$$

и зависимость от $a_1(t_{k-1})$, …, $a_{n_2(A)}(t_{k-1})$ является существенной.

Будем говорить, что криптографический протокол удовлетворяет строгому свойству целостности множества состояний для субъекта A, если одновременно с условием (2) выполнено

$$\alpha_i(t_k) = \mathtt{var}(\ldots, \alpha_{n_1(A)+1}(t_{k-1}), \ldots, \alpha_{n_2(A)}(t_{k-1}), \ldots), \quad i = n_2(A) + 1, \ldots, n(A), \tag{3}$$

и зависимость от $\alpha_{n_1(A)+1}(t_{k-1})$, …, $\alpha_{n_2(A)}(t_{k-1})$ является существенной.

Если криптографический протокол удовлетворяет определению 14, то субъект A может удостовериться в том, что каждая из ячеек памяти его состояния, вырабатываемая в ходе выполнения протокола, подтверждается с использованием значений, которые не могут быть навязаны нарушителем.

Существенная зависимость от значений $\alpha_{n_1(A)+1}, \ldots, \alpha_{n_2(A)}$, вырабатываемых с использованием ГСЧ, позволяет говорить о том, что данные значения выработаны непосредственно в ходе выполнения протокола, т. е. в реальном времени, и не могут быть продублированы в ходе выполнения другой сессии протокола. Невозможность дублирования случайных значений должна обеспечиваться используемым ГСЧ.

2.9. Свойство защищённости от KCI-атак

Рассмотрим свойство защищённости от KCI-атак — Key Compromise Impersonation attack (свойство С 22). Данные атаки реализуются в случае компрометации исходной ключевой информации (долговременного ключа) одного из легальных субъектов, или в случае определения нарушителем исходной ключевой информации (нарушения свойства С 8, п. 2.6).

В качестве примера такой атаки рассмотрим протокол MTI(C0) [19] и построим KCI-атаку для него.

Пусть q — нечётное простое число, $G = \langle g \rangle$ — циклическая абелева группа, порождаемая элементом g порядка q. Будем считать, что в группе G решение задачи дискретного логарифмирования имеет высокую трудоёмкость.

Субъекты A и B обладают парами асимметричных ключей (закрытый и открытый): $a \in \mathbb{F}_q^*$, $K_{cA} = g^a \in G$ и $b \in \mathbb{F}_q^*$, $K_{cB} = g^b \in G$. Будем считать, что открытые ключи K_{cA}, K_{cB} известны обоим субъектам взаимодействия, а их значения подтверждены до начала выполнения протокола. Схема работы протокола MTI(C0) представлена на рис. 11.

Рассмотрим реализацию KCI-атаки на протокол MTI(C0) в рамках предположения, что нарушитель C знает закрытый ключ a участника A и открытые ключи K_{cA} и K_{cB}. Нарушитель C пытается выдать себя за субъекта B перед субъектом A (рис. 12).

При реализации атаки нарушитель C может сформировать сообщение $\xi_1(t_4)$ от лица субъекта B (при условии знания ключа a) таким образом, что субъект A ничего не заподозрит и будет думать, что ключ выработан с субъектом B, а на самом деле он будет выработан с нарушителем C.

В качестве основных методов противодействия KCI-атакам можно выделить следующие:

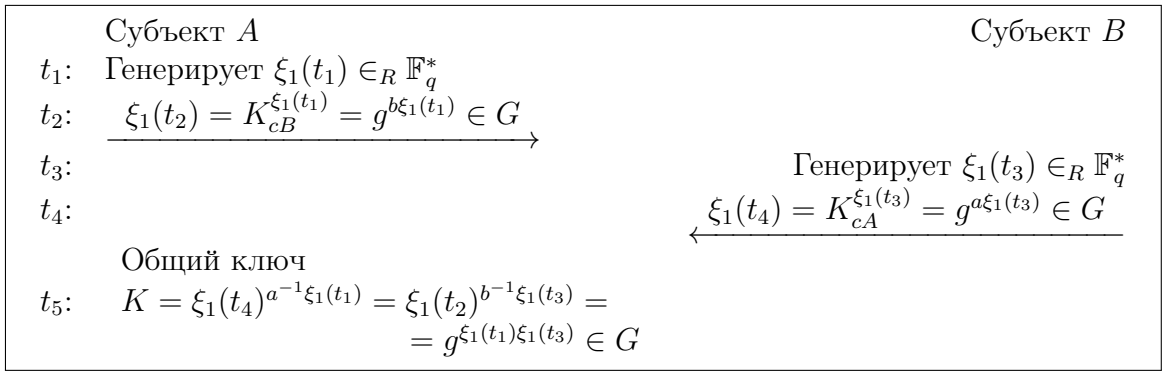

Рис. 11. Протокол MTI(C0)

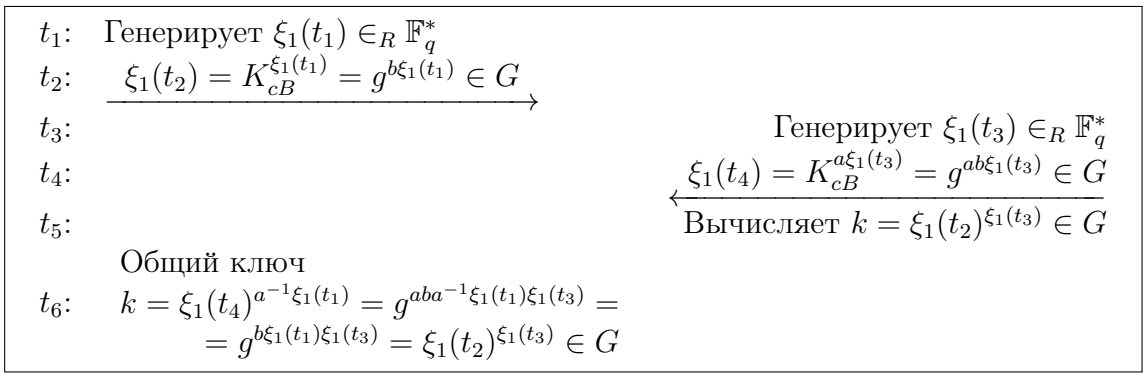

Рис. 12. KCI-атака на протокол MTI(C0)

— уникальность сессионных ключей — ключи для каждой сессии протокола должны генерироваться независимо;
— запрет на использование ключей аутентификации при формировании сессионных ключей (см., например, равенство (3));
— использование производных ключей, сформированных субъектами в процессе выполнения протокола независимо друг от друга, при выработке сессионного ключа;
— обязательная аутентификация субъекта, с которым производится взаимодействие, а также аутентификация выработанного сессионного ключа.

Поскольку KCI-атаки нацелены на эксплуатацию недостатков ключевой системы протокола и механизмов аутентификации, то можно говорить о защищённости протокола от данного класса атак только в случае выполнения следующих свойств: C 1 (аутентификация субъекта), C 9 (аутентификация ключа), C 10 (подтверждение ключа), C 12 (стойкость при компрометации ключа аутентификации), C 13 (формирование новых ключей) и C 14 (защита от навязывания ключевых значений).

3. Определение показателей эффективности защиты информации

В настоящее время в Российской Федерации принято оценивать стойкость средств криптографической защиты информации с точки зрения «практической стойкости», т. е. путём оценки трудоёмкости известных аналитику методов решения математических задач, решение которых приводит к компрометации используемых криптографических преобразований и алгоритмов. Изложение принятой методологии оценки стойкости с разной долей детализации может быть найдено в работах [20, 21, 52, 53].

При этом минимальная трудоёмкость и вероятность успешного решения рассматриваемых задач выступают в качестве показателей эффективности защиты. Представляется естественным распространить эти же показатели и на анализ криптографических протоколов.

В соответствии с определённой выше моделью нарушителя оценка возможности компрометации криптографического протокола может осуществляться при помощи следующих подходов:

1) При помощи «пассивных» атак, т. е. перехвата, накопления и последующего анализа перехваченной информации. В рамках применяемой в работе модели нарушителя криптографического протокола такие атаки сводятся к обращению однонаправленных функций, т. е. к решению сложных математических задач. Для каждой из таких задач рассматривается некоторое множество алгоритмов, находящих решение задачи с вероятностью π и трудоёмкостью Q. Отбрасывая алгоритмы с ничтожной вероятностью успеха, меньшей чем некоторое заранее фиксированное значение π_0, мы можем выбрать алгоритм с наименьшей трудоёмкостью. Именно такой алгоритм и считается наилучшим алгоритмом компрометации криптографического протокола при проведении «пассивных» атак.
2) При помощи «активных» атак, сводящихся к навязыванию одному или нескольким субъектам ложных значений, поступающих из канала связи; цель такого навязывания состоит в том, чтобы заставить легитимного субъекта сделать ложный вывод о том, что значения одной или нескольких ячеек его памяти являются истинными (подтверждёнными).

   Навязываемые значения могут вычисляться нарушителем как случайным образом, так и с использованием методов, применяемых при реализации «пассивных» атак. В первом случае считаем, что трудоёмкость выработки навязываемых значений ничтожна и основную роль при анализе играет вероятность π принять ложное значение за истинное, при этом число попыток навязывания ограничено только временем действия исходной ключевой информации и спецификацией протокола (если спецификация содержит подобные ограничения). Если полученное после исследования значение вероятности π меньше, чем заранее фиксированное значение π_0, то отбрасываем такой способ компрометации протокола как маловероятный.

   Во втором случае, когда нарушитель вырабатывает навязываемые значения путём решения сложных математических задач, в качестве вероятности успеха π естественно принять величину вероятности успеха алгоритма, имеющего наименьшую трудоёмкость реализации Q.

Описанные подходы позволяют получать единообразные численные значения показателей эффективности защиты, в качестве которых будем использовать минимально допустимую вероятность успеха алгоритма компрометации криптографического протокола π_0 и минимальную трудоёмкость Q_0 алгоритма компрометации, имеющего вероятность успеха, большую или равную π_0.

Дадим более формальное описание сказанного. Обозначим $k = k_{\max} - 1$ и рассмотрим $\mathbb{V}_{n_j}(t_k)$, $j = 1, \ldots, l_k$, — конечные множества, задающие область определения случайных величин, принимающих на указанных множествах в момент времени t_k соответственно значения $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$. Тогда величина, определяемая равенством

$$\sigma_i(t_{k_{\max}}) = \mathtt{conf}_{i,k}(a_1(t_k), \ldots, a_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)) \in \mathbb{B},$$

в котором $a_j(t_k) = \{\alpha_j(t_k), \sigma_j(t_k)\}$, $j = 1, \ldots, n(A)$ — некоторые фиксированные значения из $\mathbb{V}^*_\infty \times \mathbb{B}$, такие, что для каждой существенной ячейки памяти выполнено условие $\sigma_j(t_k) = \mathtt{true}$, может рассматриваться как реализация случайной величины, принимающей два значения — истина или ложь.

Определим символом

$$\pi_{i,k_{\max}} = \begin{cases} 0, & \text{если } l_k = 0, \\ \mathsf{P}[\sigma_i(t_{k_{\max}}) = \mathtt{true}] & \text{иначе} \end{cases} \tag{4}$$

вероятность принять случайный вектор $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$ в качестве значения, подтверждающего $\alpha_i(t_{k_{\max}})$ в момент времени $t_{k_{\max}}$. Будем говорить, что величина $\pi_{i,k_{\max}}$ определяет вероятность принять ложное значение $\alpha_i(t_{k_{\max}})$ за истинное.

Поскольку значение каждой из существенных переменных $a_i(t_k)$ должно быть подтверждено, для них также могут быть определены вероятности $\pi_{i,k}$ принять ложное значение $\alpha_i(t_k)$ за истинное. После этого мы аналогично должны определить вероятности $\pi_{i,k-1}$, $\pi_{i,k-2}$ и так далее.

Представляя множество зависимостей между переменными в качестве графа, мы можем расположить в его узлах значения $\pi_{i,k}$, $k = 1, \ldots, k_{\max}$, и задать величину

$$\pi = \max_{L_{i,j}} \left(1 - \prod_{\pi_{i,k} \in L_{i,j}} (1 - \pi_{i,k}) \right), \quad j \in \mathbb{N}, \tag{5}$$

где $L_{i,1}, L_{i,2}, \ldots$ — пути в графе, приводящие к значению ячейки $a_i(t_{k_{\max}})$.

Определение 15. Величину π, задаваемую равенством (5), будем называть вероятностью успешной компрометации криптографического протокола.

Выбор значений $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$, определяющих вероятности $\pi_{i,k}$, может производиться нарушителем двумя способами: случайным образом или с использованием предварительных вычислений. Рассмотрим оба способа подробнее.

3.1. Случайное угадывание

Мы можем рассматривать принимаемые субъектом A величины $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$ как случайные значения, для которых задано распределение вероятностей

$$\mathsf{P}[\xi_j(t_k) = v] = \frac{1}{|\mathbb{V}_{n_j}(t_k)|}, \quad j = 1, \ldots, l_k, \quad v \in \mathbb{V}_{n_j}(t_k).$$

Такая ситуация возникает, когда $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$ вырабатываются субъектом B с использованием секретного ключа, не известного нарушителю. В этом случае нарушитель просто угадывает значения, выбирая их случайным образом из области определения. В этом случае вероятность

$$\mathsf{P}[\mathtt{conf}_{i,k}(a_1(t_k), \ldots, a_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)) = \mathtt{true}] = \pi_{i,k}$$

определяет вероятность однократного навязывания значения $\alpha_i(t_k)$.

В качестве примера рассмотрим изображённый на рис. 2 протокол аутентификации. Для ложной аутентификации нарушителю нужно предъявить значение $\xi_1(t_2)$, такое, чтобы у субъекта A выполнялось равенство

$$\mathtt{conf}(K_c, \alpha_1(t_1)||\alpha_2(t_1), \xi_1(t_2)) = \mathtt{true}$$

при $\alpha_1(t_1) = ID_B$, $\alpha_2(t_1) = \xi_1(t_1)$ (значение $\xi_1(t_1)$ отправляется субъектом A в канал связи и доступно нарушителю).

Поскольку нарушителю неизвестен секретный ключ K_a, он выбирает значение кода аутентификации случайным образом. Если в качестве функции `mac` используется алгоритм выработки электронной подписи, регламентированный ГОСТ Р 34.10-2012, и используется эллиптическая кривая с порядком группы точек q бит, где q — нечётное простое число, то вероятность однократного нарушения свойства аутентификации может быть оценена величиной $\pi_{1,3} = (q-1)/2$ (множитель $1/2$ возникает из-за того, что в алгоритме выработки электронной подписи используется только x-координата точки эллиптической кривой и точки (x, y) и $(x, -y)$ дают одинаковое значение подписи).

Рассмотрим другой пример, возникающий при исследовании транспортных криптографических протоколов. Пусть субъект A принимает от субъекта B аутентифицируемые сообщения M_1, M_2, ... (свойство С 2, п. 2.3) в соответствии со схемой, изображённой на рис. 13.

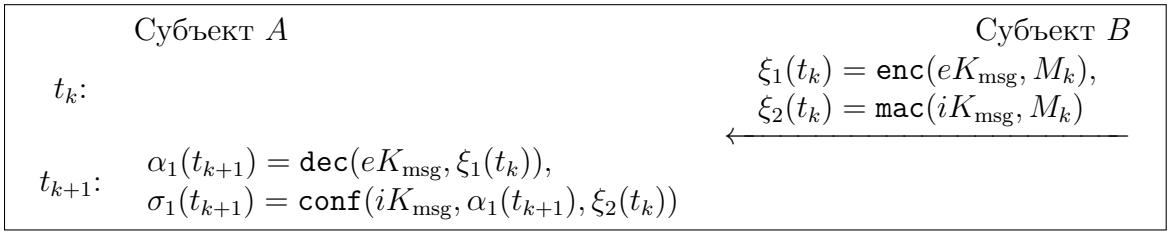

Рис. 13. Транспортный протокол приема сообщений

При этом мы предполагаем, что субъект B и производные ключи eK_{msg}, iK_{msg} были ранее аутентифицированы субъектом A в процессе выполнения протокола выработки общих ключей. Такая ситуация реализуется в большинстве современных протоколов, например TLS 1.3, IPSec, SP FIOT и т. п. Также будем считать, что для выработки кода аутентификации (значения $\xi_2(t_k)$) длиной n бит используется алгоритм, рекомендуемый отечественными стандартами или рекомендациями по стандартизации.

Применяя предложенный выше подход, мы можем оценить вероятность навязывания нарушителем субъекту одного ложного сообщения в течение заданного интервала времени. Обозначим символом V пропускную способность канала связи, по которому передаются сообщения (на практике могут использоваться значения 100 Мбайт/с, 1Гбит/с, 100 Гбит/с и т. п.). За время T (измеряемое в секундах) может быть передано не более VT бит информации, или не более $m = \lceil VT/l \rceil$ сообщений, где l — минимально возможная длина (в битах) сообщений $M_1, M_2, \ldots$ Тогда, согласно равенству (5), получаем, что для $m \leqslant n$ вероятность навязывания ложного сообщения равна

$$\pi = 1 - \left(1 - \frac{1}{2^n}\right)^m =$$

$$= \frac{m}{2^n} - \frac{m(m-1)}{2^{2n}} + \frac{m(m-1)(m-2)}{2^{3n}} + \ldots + \frac{1}{2^{mn}} = \sum_{i=1}^{m} \frac{(-1)^{i-1} m!}{i!(m-i)!2^{in}},$$

где 2^{-n} — вероятность случайного угадывания значения кода аутентификации. При $m \geqslant n$ считаем, что $\pi = 1$.

3.2. Применение вычислительных алгоритмов

Пусть, как и ранее, $k = k_{\max} - 1$ и

$$\sigma_i(t_{k_{\max}}) = \mathtt{conf}_{i,k}(a_1(t_k), \ldots, a_{n(A)}(t_k),\ \xi_1(t_k), \ldots, \xi_{l_k}(t_k)) \in \mathbb{B}.$$

Для подделки значений $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$, получаемых субъектом A из канала связи, нарушитель может использовать подход, отличный от случайного выбора значений.

Обозначим символом $\Xi_{i,k_{\max}}$ множество передаваемых по каналу связи значений

$$\Xi_{i,k_{\max}} = \{\xi_1(t_1), \ldots, \xi_{l_1}(t_1),\ \xi_2(t_1), \ldots, \xi_{l_2}(t_2), \ldots, \xi_1(t_k), \ldots, \xi_{i-1}(t_k)\} \tag{6}$$

и будем считать, что для некоторого индекса $j \in [1, \ldots, l_k]$ значение $\xi_j(t_k)$ определяется субъектом B равенствами

$$\xi_j(t_k) = f(\Xi^{(1)}_{j,k}, B_{j,k}, \Gamma_{j,k}), \tag{7}$$

в которых:

— $\Xi^{(1)}_{j,k} \subseteq \Xi_{j,k}$ — множество переданных ранее по каналу связи значений, элементы которых известны нарушителю;

— $B_{j,k} = \{\beta_1, \ldots, \beta_{r_{j,k}}\}$, где $r_{j,k} \in \mathbb{N}$, — множество значений, удовлетворяющих равенствам

$$\xi_u(t_v) = h_j(\beta_j) \in \Xi_{i,k}$$

для некоторых $u, v \in \mathbb{N}$, $v \leqslant k$, и однонаправленных отображений $h_1, \ldots, h_{r_{j,k}}$ (в общем случае используемые при определении величин $\beta_1, \ldots, \beta_{r_{j,k}}$ значения $\xi_u(t_v)$ могут не принадлежать множеству $\Xi^{(1)}_{j,k}$);

— $\Gamma = \{\gamma_1, \ldots, \gamma_{s_k}\}$ — множество произвольно формируемых субъектом B значений, где s_k — целое неотрицательное число.

Отметим, что преобразование f может быть составным и включать выработку производной ключевой информации, используемой не только при получении значения $\xi_j(t_k)$, но и других значений, вычисляемых позднее.

В качестве примера рассмотрим изображённый на рис. 14 вариант протокола Диффи — Хеллмана, реализуемый в циклической абелевой группе $G = \langle g \rangle$, порождаемой элементом g порядка q, где q — нечётное простое число. Как и ранее, будем считать, что в группе G решение задачи дискретного логарифмирования имеет высокую трудоёмкость.

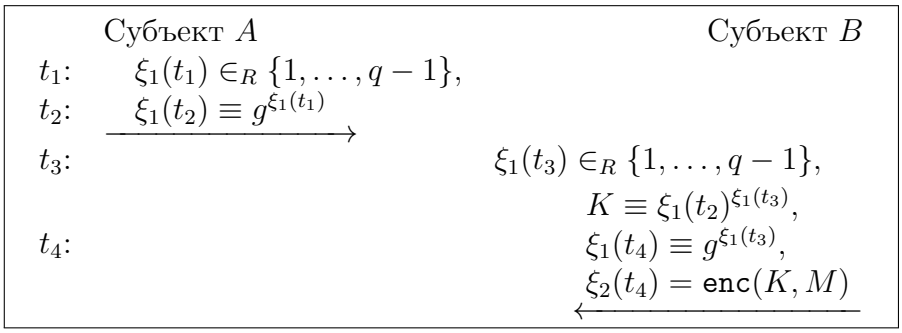

Рис. 14. Протокол Диффи — Хеллмана с передачей зашифрованного сообщения

В примере субъект A принимает от субъекта B сообщение M, зашифрованное на общем ключе K. Тогда значение $\xi_2(t_4)$, в соответствии с (7), может быть представлено

в виде

$$\xi_2(t_4) = f\left(\Xi_{2,4}^{(1)} = \{\xi_1(t_2)\},\ B_{2,4} = \{\xi_1(t_3)\},\ \Gamma_{2,4} = \{M\}\right),$$

где отображение f представляет собой процедуру зашифрования сообщения M, а однонаправленное отображение h_1 определяется сравнением

$$\xi_1(t_4) \equiv g^{\xi_1(t_3)} \pmod{p}$$

и представляет собой операцию возведения в степень в группе G (отметим, что протокол рис. 14 является иллюстрацией равенства (7) и не безопасен, поскольку общий ключ K может быть навязан субъекту A с помощью «атаки посередине», нарушено также свойство целостности передаваемого сообщения M, п. 2.2).

Вернёмся к (7). Нарушитель, которому известны отображения $f, h_1, \ldots, h_{r_k}$, может определить значения $\beta_j = h_j^{-1}(\xi_j(t_{l_j}))$ и навязать субъекту A ложные значения $\gamma_1, \ldots, \gamma_{s_k}$. Однонаправленные отображения $h_1, \ldots, h_{r_k}$, как правило, выбираются при синтезе протокола таким образом, чтобы вычисление обратного отображения являлось сложной математической задачей. К таким задачам могут быть отнесены задача определения секретного ключа алгоритма блочного шифрования, задача дискретного логарифмирования в группе точек эллиптической кривой, задача вычисления первого прообраза для функции хеширования и т. п.

Для каждой из перечисленных задач могут быть рассмотрены методы её решения, характеризуемые трудоёмкостью $Q_{j,k}$ и вероятностью успеха $p_{j,k}$, для которой выполнено неравенство $p_{j,k} \geqslant \pi_0$ (значение π_0 позволяет вывести из рассмотрения алгоритмы с ничтожной вероятностью успеха). Тогда вероятность навязывания ложных значений $\gamma_1, \ldots, \gamma_{s_k}$, или, другими словами, компрометации ячейки памяти $a_i(t_{k_{\max}})$, может быть определена равенством

$$\pi_{i,k_{\max}} = \prod_{j=1}^{r_k} p_{j,k}, \quad k = k_{\max} - 1$$

и трудоёмкостью

$$Q_{i,k_{\max}} = \sum_{j=1}^{r_k} Q_{j,k}.$$

Вероятность π успешной компрометации протокола в целом определяется с использованием равенства (5). При этом общая трудоёмкость компрометации протокола задаётся величиной

$$Q = \min_{L_{i,j}} \sum_{\pi_{i,k} \in L_{i,j}} Q_{i,k},$$

где $L_{i,1}, L_{i,2}, \ldots$ — пути в графе, приводящие к значению ячейки $a_i(t_{k_{\max}})$. Сделаем ряд замечаний:

1) Определённое равенством (6) множество $\Xi_{j,k}$ состоит из значений, передаваемых в ходе выполнения одной сессии выполнения протокола. Однако, согласно принятой модели нарушителя, для навязывания значений $\gamma_1, \ldots, \gamma_{s_k}$ нарушитель может использовать значения, передаваемые во всех сессиях, выполнявшихся до момента проведения атаки $t_{k_{\max}}$. Это необходимо учитывать при определении вероятности и трудоёмкости обращения значений однонаправленных функций.
2) Трудоёмкость компрометации протокола при случайном выборе нарушителем значений $\xi_1(t_k), \ldots, \xi_1(t_{l_k})$, см. п. 3.1, определяется длиной пути в графе, соответствующей максимальной вероятности компрометации протокола.

Теперь мы можем дать определение безопасного криптографического протокола. Будем считать, что допустимые значения показателей эффективности защиты π_0 и Q_0 заданы и, согласно рекомендациям Р 1323565.1.012-2017, определены действующими требованиями по безопасности для каждого класса средств защиты информации.

Определение 16. Будем говорить, что протокол обладает подтверждённым состоянием субъекта A в момент времени t_{k_A}, где $k_A \in \{1, \ldots, k_{\max}\}$, если значение всех ячеек памяти состояния $A(t_{k_A})$ является подтверждённым явно или косвенно, т. е.

$$\sigma_i(t_{k_A}) = \texttt{true}, \quad i = 1, \ldots, n(A).$$

Будем говорить, что протокол является безопасным для субъекта A начиная с некоторого момента времени t_{k_A}, где $k_A \in \{1, \ldots, k_{\max}\}$, если:

1) для всех $k \geqslant k_A$ протокол обладает подтверждённым состоянием $A(t_k)$ субъекта A;
2) выполнено одно из двух утверждений:
   — вероятность π успешной компрометации протокола удовлетворяет неравенству $\pi < \pi_0$;
   — трудоёмкость Q компрометации протокола удовлетворяет неравенству $Q > Q_0$.

Будем называть протокол безопасным начиная с некоторого момента времени t_k, если он является безопасным для каждого участвующего в информационном взаимодействии субъекта $A, B, \ldots$ начиная с момента времени t_{K_A}, t_{K_B}, $\ldots$ соответственно и $t_k \geqslant \max\{t_{K_A}, t_{K_B}, \ldots\}$.

4. Методика оценки безопасности

Теперь мы можем сформулировать методику оценки безопасности созданного ранее или вновь разрабатываемого криптографического протокола.

Область применения методики: криптографические протоколы выработки общего ключа, а также транспортные криптографические протоколы, предназначенные для передачи конфиденциальной информации между субъектами взаимодействия.

Исходными данными для проведения исследования являются:

— класс средств защиты, в которых предполагается использование исследуемого протокола (в случае, если класс средств не определён, анализ должен проводиться для максимального класса средства защиты (см. рекомендации Р 1323565.012-2017));
— модель угроз и модель нарушителя, относительно которых оценивается эффективность защиты, обеспечиваемой исследуемым протоколом (в случае, если модель угроз не определена, должна использоваться модель согласно [1]; если не определена модель нарушителя, то должна использоваться модель, приведённая в рекомендациях Р 1323565.012-2017 для выбранного класса средств защиты информации);
— допустимые значения показателей эффективности защиты π_0 и Q_0, определённые действующими требованиями по безопасности для выбранного класса средств защиты информации (если такие требования существуют);
— спецификация протокола, в соответствии с которой предполагается его практическая реализация;
— условия практической эксплуатации протокола — к таким данным могут относиться сведения о пропускной способности канала, объёме и допустимом времени передачи конфиденциальной информации, сроках смены ключевой информации, допустимом числе ложных попыток аутентификации и т. п.

В результате проведения исследования будут получены:

— перечень свойств безопасности, обеспечиваемых исследуемым протоколом (допускается ситуация, при которой данный перечень свойств может оказаться пустым);
— численные значения показателей эффективности защиты π и Q;
— если заданы допустимые значения показателей эффективности защиты π_0 и Q_0, то заключение о безопасности или небезопасности исследуемого протокола.

Последовательность исследований должна состоять из следующих шагов:

1. Необходимо построить формальную модель протокола, описывающую состояния каждого субъекта взаимодействия и преобразования, применяемые к ячейкам памяти и поступающим из канала связи данным, т. е.

— определить используемые в протоколе криптографические преобразования;
— определить ключевую и криптографически опасную информацию, в частности определить процедуры выработки сессионной и производной ключевой информации;
— определить множество ячеек памяти, образующих состояния участвующих во взаимодействии субъектов; в качестве значений, помещаемых в ячейки памяти, должны выступать
  — исходная ключевая информация;
  — случайные значения, вырабатываемые субъектами в ходе взаимодействия;
  — производная ключевая информация;
  — а также значения, отличные от указанных выше и используемые для выработки производной ключевой информации;

  дополнительно, в состояние субъекта могут включаться значения, определяемые спецификацией протокола, которые, по мнению исследователя, могут влиять на значения определяемых показателей эффективности защиты;
— определить число возможных состояний каждого субъекта (с учётом действующих требований по безопасности, накладывающих ограничения на использование ключевой информации);
— определить области допустимых значений для каждой из случайных величин $\xi_1(t_k), \ldots, \xi_{l_k}(t_k)$, получаемых субъектом из канала связи или генератора случайных величин (для всех возможных значений индекса k);
— определить отображения, задающие переход субъекта из одного состояния в другое, включая функции изменения значений ячеек памяти и функции, подтверждающие эти значения.

После построения модели протокола должно быть показано, что все ячейки состояний каждого из субъектов взаимодействия должны быть подтверждены явно или косвенно (см. определение 5). При синтезе нового протокола это свойство должно выполняться в обязательном порядке.

Если для разработанного ранее протокола это свойство не выполнено, то появляется возможность построения атаки на протокол, направленной на навязывание неподтверждённого значения. Для поиска таких атак должны быть применены средства автоматизированной верификации протоколов, такие, как Scyther [43], Proverif [44], Avispa [32] или им подобные.

2. Необходимо рассмотреть приведённый в п. 1 перечень свойств безопасности и удалить из него свойства, неприменимые к исследуемому протоколу в связи с выбранной моделью угроз и условиями практического применения протокола. Для оставшихся в перечне свойств должна быть проведена проверка их выполнимости в соответствии

с формальными определениями, сформулированными в п. 2, а также с учётом зависимостей, указанных в табл. 3.

Если для проверки выполнимости какого-либо свойства безопасности должны быть определены значения трудоёмкости Q и вероятности успеха π обращения одной или нескольких однонаправленных функций, то такие значения должны определяться с учётом условий практической эксплуатации протокола, содержащихся в исходных данных для проведения исследования.

3. В обязательном порядке должен быть проведен анализ используемой ключевой информации, включающий в себя рассмотрение следующих вопросов:

— Перед началом выполнения протокола субъектам должна быть доступна исходная ключевая информация, используемая для аутентификации сторон взаимодействия (ключи аутентификации). Если используются симметричные ключи, то они должны быть предварительно распределены с использованием организационно-технических мер защиты. Если используются асимметричные пары ключей, то открытые ключи должны быть подтверждены электронной подписью удостоверяющего (доверенного) центра, а открытые ключи удостоверяющего центра должны быть доставлены субъектам с использованием организационно-технических мер защиты. Отсутствие подтверждённой исходной ключевой информации приводит к нарушению свойства аутентификации субъекта (свойство С 1) и, как следствие, к нарушению большинства из рассмотренных свойств безопасности.

— Необходимо, чтобы ключи аутентификации не использовались непосредственно для шифрования и имитозащиты передаваемой информации. В противном случае возможно как исчерпание ресурса ключа, так и реализация нарушителем атак, использующих конфиденциальную информацию для нарушения свойства аутентификации.

— Основное требование к производной ключевой информации, применяемой для шифрования и имитозащиты передаваемых сообщений, заключается в невозможности её определения нарушителем с трудоёмкостью, меньшей чем тотальное опробование всех возможных значений. Каждый ключ, как правило, представляется в виде двоичного вектора длины m бит, таким образом, нарушителю необходимо опробовать 2^m ключей для компрометации сообщений. Отсюда следует, что необходимо проверить выполнимость следующих условий:
  — множество значений, которые может принимать производный ключ, совпадает с множеством $\mathbb{V}_m$;
  — принимаемые производным ключом значения непредсказуемы, т. е. последовательность нескольких выработанных в различных сессиях протокола производных ключей $K_1, K_2, \ldots$ должна быть статистически неотличима от последовательности случайных равновероятно распределённых на множестве $\mathbb{V}_m$ величин.

— При практическом применении средств защиты информации могут нарушаться правила эксплуатации средств, превышаться заданные ограничения на объём обрабатываемой информации или возникать уязвимости в программном обеспечении, все вместе или по отдельности приводящие к возможности практического определения нарушителем производных ключей или исходной ключевой информации. Это приводит к необходимости встраивания в криптографические протоколы мер, минимизирующих объём скомпрометированной информации. В качестве таких мер могут выступать:

  - использование односторонних функций, не позволяющих вычислять значения ключей аутентификации по значениям производных ключей;
  - использование в каждой сессии протокола уникальных случайных значений для выработки производных ключей (свойства С 12, С 13);
  - использование «древовидных» структур выработки производных ключей, не позволяющих нарушителю по известному производному ключу K_n вычислить значения ключей K_{n-1} и K_{n+1} (свойство С 11).
- Дополнительно в рамках математических исследований должны быть проверены следующие гипотезы:
  - о ничтожной вероятности совпадения различных производных ключей, вырабатываемых в рамках одной сессии протокола;
  - о статистической независимости последовательности производных ключей $K_1, K_2, \ldots$, вырабатываемых в различных сессиях протокола.

4. Необходимо проверить, возможно ли применение известных ранее атак для компрометации исследуемого протокола. Для этого необходимо, во-первых, подготовить базу известных атак на криптографические протоколы из рассматриваемого класса, а во-вторых, провести классификацию известных атак, проведя систематизацию по следующим принципам:

- по методам реализации атаки; к таким методам могут быть отнесены повтор или отражение передаваемых сообщений, использование задержек и перемешивание передаваемых сообщений, изменение формата передаваемых сообщений, использование сообщений из других сессий и т. п.;
- по объектам проведения атаки; в качестве объектов атаки могут выступать передаваемые данные, секретные ключи, случайные значения, вырабатываемые в ходе протокола, и т. п.;
- по свойствам безопасности, поскольку каждая успешно применимая атака приводит к нарушению одного или нескольких свойств безопасности;
- техническим возможностям, необходимым для проведения атаки, например возможностям по перехвату передаваемых данных;
- месту проведения атаки: может ли данная атака проводиться внешним нарушителем или внутренним.

Использование подобной классификации позволяет сузить перечень атак, которые могут быть применены для компрометации исследуемого протокола. Действительно, если протокол содержит явно прописанные в спецификации меры защиты от атак повтором, то такой класс атак может оказаться неприменимым. Аналогично, атака на идентификатор субъекта взаимодействия может оказаться неприменимой, если моделью угроз определено, что данный идентификатор представляет собой общедоступную информацию. Удостоверившись в выполнении определённого ранее перечня свойств безопасности, также можно отсеять часть атак на исследуемый протокол. Для оставшихся атак должна быть показана невозможность их применения либо предложен способ компрометации исследуемого протокола.

5. С использованием описанного в п. 3 метода должны быть определены численные значения показателей эффективности защиты информации — вероятности успешной компрометации протокола π и трудоёмкости успешной компрометации Q. Данные величины должны быть получены для всех способов компрометации исследуемого протокола, предложенных в ходе четвёртого шага исследования. После чего, согласно определению 16, должен быть сделан вывод о безопасности протокола. Отметим, что

полнота проводимого исследования может быть достигнута только в случае выполнения всех перечисленных шагов.

Предложенная методика обладает следующими достоинствами:

— методика позволяет определить конкретные численные значения показателей эффективности защиты; следует добавить, что на настоящий момент времени авторам не известен какой-либо другой поход к определению значений рассматриваемых показателей защиты;
— результаты исследования могут быть использованы при сертификации средств криптографической защиты информации, реализующих исследуемый протокол.

Также можно указать ряд недостатков предложенной методики исследования:

— поскольку при проведении анализа рассматриваются только известные атаки на криптографические алгоритмы и протоколы, то всегда существует вероятность, что найдётся атака, имеющая сложность меньше чем у атак, рассмотренных в ходе исследования; таким образом, полученные значения показателей эффективности должны рассматриваться не как точные значения, а как оценки сверху;
— несогласованность с положениями теории «доказуемой стойкости», принятой в зарубежных изданиях.

5. Анализ протоколов семейства IPSec

В 2021 г. авторами настоящей работы было успешно завершено исследование семейства протоколов, составляющих архитектуру безопасности сети Интернет (Internet Protocol security architecture, IPsec [54, 55]). Данное семейство предназначено для обеспечения защищённого обмена IP-пакетами и включает в себя два основных протокола:

— протокол IKEv2, предназначенный для выработки общей ключевой информации, взаимной аутентификации субъектов взаимодействия и создания/удаления защищённых соединений;
— протокол ESP, обеспечивающий непосредственную защиту передаваемых IP-пакетов в рамках одной сессии защищённого взаимодействия.

Для иллюстрации сказанного в предыдущих пунктах построим формальную модель входящих в IPSec криптографических протоколов.

5.1. Описание протоколов семейства IPSec

В основе протокола выработки общей ключевой информации IKEv2 лежит схема выработки общего ключа «Сигма» [56], реализуемая в группе точек эллиптической кривой $\mathcal{E}$, порождённой точкой P простого порядка q. Взаимодействие между субъектами реализуется путём отправки и получения пары сообщений (по схеме «запрос — ответ»). В качестве канала связи используется протокол UDP.

Для аутентификации сторон взаимодействия используются ключи аутентификации и проверки кода аутентификации субъектов (K_{aA}, K_{cA} для субъекта A и K_{aB},K_{cB} для субъекта B), а также K_u — ключ проверки кода аутентификации удостоверяющего центра. Помимо перечисленных ключей аутентификации, в протоколе IKEv2 дополнительно вырабатывается и применяется производная ключевая информация, приведённая в табл. 4.

В ходе первого сообщения субъект A, инициирующий выполнение протокола, направляет субъекту B следующую информацию: $\xi_1(t_1)$ — уникальный идентификатор соединения $A \to B$, выступающий в качестве заголовка сообщений, отправляемых от субъекта A к субъекту B; $\xi_2(t_1)$ — уникальный двоичный вектор и $\xi_3(t_2)$ — точка эллиптической кривой, используемые для выработки производной ключевой информации;

Таблица 4

Ключевая информация протокола IKEv2

Информация	Назначение	Вычисляется из	Время жизни
$\xi_3(t_1)$, $\xi_3(t_3)$	Случайные значения, вырабатываемые независимо A и B	ДСЧ	Удаляются сразу после использования
Q_{AB}	Общая точка эллиптической кривой $\mathcal{E}$	$\xi_3(t_1)$, $\xi_3(t_3)$	Удаляется сразу после использования
$SKEYSEED$	Общая ключевая информация	$\xi_2(t_1)$, $\xi_2(t_3)$, Q_{AB}	Удаляется после выработки производных ключей
SK_d	Вычисление производных ключей	$SKEYSEED$ и $\xi_2(t_1)$, $\xi_2(t_3)$, $\xi_1(t_1)$, $\xi_1(t_3)$	Удаляется после закрытия защищённого соединения
iSK_A	Имитозащита сообщений $A \to B$	См. выше	См. выше
iSK_B	Имитозащита сообщений $B \to A$	См. выше	См. выше
eSK_A	Шифрование сообщений $A \to B$	См. выше	См. выше
eSK_B	Шифрование сообщений $B \to A$	См. выше	См. выше
aSK_A	Аутентификация субъекта A	См. выше	См. выше
aSK_B	Аутентификация субъекта B	См. выше	См. выше
K_{encr}	Исходная ключевая информация для ESP	См. выше	Время жизни протокола ESP

$\xi_4(t_2)$ — перечень параметров безопасности, используемых для аутентификации субъектов взаимодействия.

При получении сообщения субъект B выбирает приемлемый для себя набор параметров безопасности $\xi_4(t_4)$, вырабатывает $\xi_1(t_3)$ — уникальный идентификатор соединения $B \to A$, $\xi_2(t_3)$ — уникальный двоичный вектор, а также свою точку эллиптической кривой $\xi_3(t_4)$, используемую для выработки общей ключевой информации. Первая пара сообщений, которую принято называть этапом инициализации защищённого соединения, представлена на рис. 15.

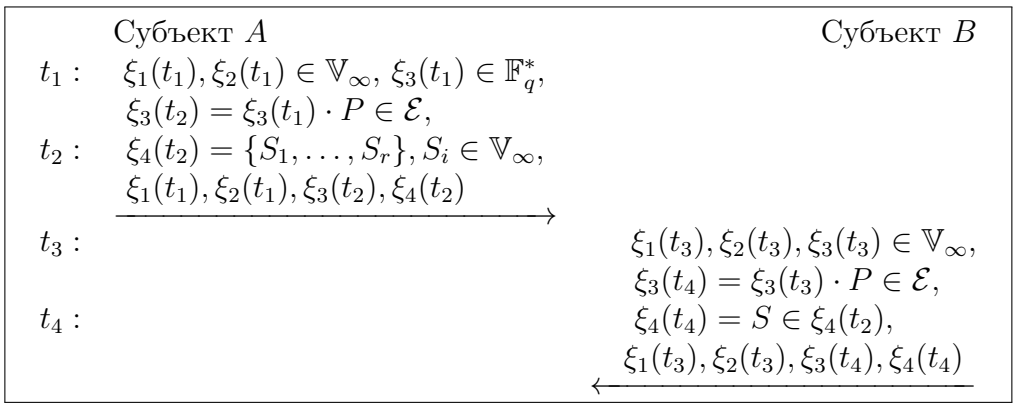

Рис. 15. Схема этапа инициализации протокола IKEv2

После этапа инициализации субъекты приступают к выработке ключевой информации. Вычисляется общая точка эллиптической кривой

$$Q_{AB} = \xi_3(t_1)\xi_3(t_3) \cdot P,$$

далее вычисляется общая ключевая информация $SKEYSEED$=$\texttt{prf}(\xi_2(t_1)||\xi_2(t_3), Q_{AB})$ и производные ключи

$$\begin{gathered} SK_d||iSK_A||iSK_B||eSK_A||eSK_{eB}||aSK_{pA}||aSK_B = \\ = \texttt{prf+}(SKEYSEED, \xi_2(t_1)||\xi_2(t_3)||\xi_1(t_1)||\xi_1(t_3)), \end{gathered} \tag{8}$$

где $\texttt{prf}(K, S) : \mathbb{K} \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ есть функция hmac [57] выработки имитовставки длиной m бит, а prf+ представляет собой итеративное отображение, вырабатывающее псевдослучайную последовательность с использованием функции prf:

$$\texttt{prf+}(K, S) = T_1||T_2||T_3 \ldots,$$
$$\text{где } \; T_1 = \texttt{prf}(K, S||\texttt{0x01}), \;\; T_2 = \texttt{prf}(K, T_1||S||\texttt{0x02}), \;\; T_3 = \texttt{prf}(K, T_2||S||\texttt{0x03}), \;\; \ldots$$

Исходная ключевая информация K_{encr} для протокола ESP определяется равенством

$$K_{\text{encr}} = \texttt{prf+}(SK_d, Q_{AB}||\xi_2(t_1)||\xi_2(t_3)). \tag{9}$$

Отметим, что все случайные значения, используемые для генерации ключевой информации, вырабатываются субъектами A и B независимо друг от друга. Схема выработки ключевой информации представлена на рис. 16.

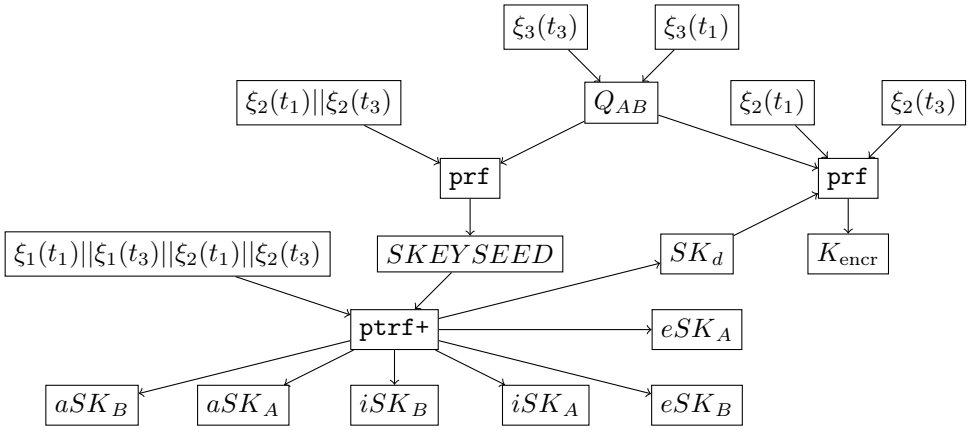

Рис. 16. Иерархия ключевой информации протокола IKEv2

После выработки ключевой информации субъекты переходят к этапу аутентификации. Субъект A направляет субъекту B следующую информацию: $\xi_1(t_1)$ — уникальный идентификатор соединения $A \to B$, выступающий в качестве заголовка сообщений, отправляемых от субъекта A к субъекту B; $\xi_1(t_5)$ — зашифрованное сообщение M_A, содержащее идентификатор ID_A субъекта A; $\texttt{Cert}(K_{aA})$ — сертификат ключа проверки кода аутентификации субъекта A (в состав сертификата должны входить значение ключа K_{aA} и электронная подпись под значением ключа, выработанная удостоверяющим центром) и уникальную метку $AUTH_A$, используемую для аутентификации

субъекта A; $\xi_2(t_5)$ — код аутентификации отправляемого сообщения. Метка $AUTH_A$ определяется равенствами

$$\begin{aligned} BLOB_A &= \underbrace{\xi_1(t_1)||\xi_2(t_1)||\xi_3(t_2)||\xi_4(t_2)}_{\text{см. рис.15}}||\xi_2(t_3)||\mathtt{prf}(aSK_A, ID_A), \\ AUTH_A &= \mathtt{sign}(K_{aA}, BLOB_A), \end{aligned} \tag{10}$$

где $\mathtt{sign}$ — функция выработки электронной подписи, а подписываемое сообщение $BLOB_A$ зависит не только от отправленного ранее субъектом A сообщения $\xi_1(t_1)$, $\xi_2(t_1)$, $\xi_3(t_2)$, $\xi_4(t_2)$ и выработанного субъектом B случайного значения $\xi_2(t_3)$, но и от производного ключа aSK_A.

Далее субъект B выполняет следующую последовательность шагов:

— расшифровывает полученное сообщение и проверяет его корректность, т. е. выполнение условия $\mathtt{conf}(iSK_A, \mathtt{dec}(eSK_A, \xi_1(t_5)), \xi_2(t_5)) =^? \mathtt{true}$; если условие не выполнено, протокол прерывается с уведомлением о неудаче;

— выделяет из расшифрованного сообщения K_{cA} — ключ проверки кода аутентификации субъекта A — и подтверждает его с помощью ключа доверенного центра K_u, т. е. проверяет выполнение условия $\mathtt{verify}(K_u, \mathtt{Cert}(K_{cA})) =^? \mathtt{true}$; если условие не выполнено, протокол прерывается с уведомлением о неудаче;

— выделяет из расшифрованного сообщения идентификатор ID_A, формирует в соответствии с (10) сообщение $BLOB_A$ и проверяет его корректность, т. е. выполнение условия $\mathtt{verify}(K_{cA}, BLOB_A, AUTH_A) =^? \mathtt{true}$; если условие не выполнено, протокол прерывается с уведомлением о неудаче;

— в случае успешного завершения всех проверок аутентифицирует субъекта A.

После этого для собственной аутентификации субъект B направляет аналогичное сообщение субъекту A. Схема этапа аутентификации представлена на рис. 17.

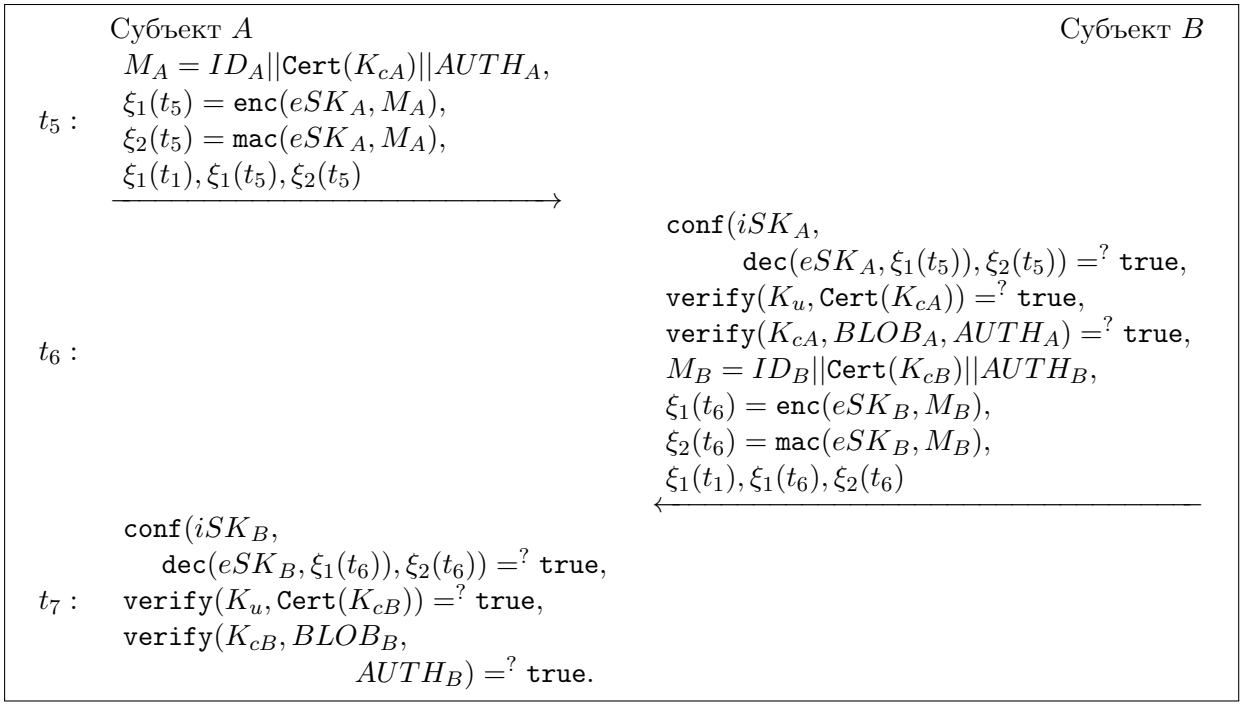

Рис. 17. Схема этапа аутентификации протокола IKEv2

После того как субъект A завершает аутентификацию субъекта B, начинается процесс обмена зашифрованными IP-пакетами по каждому из созданных однонаправленных соединений — в первом соединении сообщения направляются от субъекта A к субъекту B, во втором соединении — в обратную сторону. Механизм шифрования и имитозащиты передаваемых пакетов определяется протоколом ESP.

Согласно Р 1323565.1.030-2021, протокол ESP использует выработанную в ходе выполнения протокола IKEv2 ключевую информацию K_{encr} (см. табл. 4) для генерации производных ключей шифрования передаваемых IP-пакетов.

Пусть $K_{\text{encr}} = K_1||K_s$, где двоичная длина K_s зависит от длины блока используемого блочного шифра. Производные ключи вырабатываются согласно равенству

$$K_{\text{msg}} = \texttt{tree}(K_1, i_1, i_2, i_3) = \texttt{kdf}(\texttt{kdf}(\texttt{kdf}(K_1, l_1, \texttt{0x00}||i_1), l_2, i_2),$$

где функция $\texttt{kdf}(K, L, S) : \mathbb{K} \times \mathbb{V}_\infty^* \times \mathbb{V}_\infty^* \to \mathbb{V}_m$ построена с использованием функции `hmac` (см. Р.50.1.113-2016):

$$\texttt{kdf}(K, L, S) = \texttt{hmac}(K, \texttt{0x01}||L||\texttt{0x00}||S||\texttt{0x00}||\texttt{0x01});$$

l_1, l_2, l_3 — фиксированные константы из $\mathbb{V}_\infty$; $i_1 \in \mathbb{Z}_{2^8}$; $i_2, i_3 \in \mathbb{Z}_{2^{16}}$.

На каждом производном ключе K_{msg} может быть зашифровано несколько пакетов, которые нумеруются с помощью счётчика N_p, $N_p \in \{0, \ldots, 2^{24}\}$; в протоколе ESP с помощью счётчика N_s нумеруются также все передаваемые в рамках одного соединения пакеты. Счетчик N_s используется для защиты от атак навязывания повторных пакетов, а счётчик N_p — для контроля объёма информации, зашифрованной на одном ключе K_{msg}. Шифрование сообщения $M \in \mathbb{V}_\infty$ (IP-пакета) осуществляется согласно схеме рис. 18.

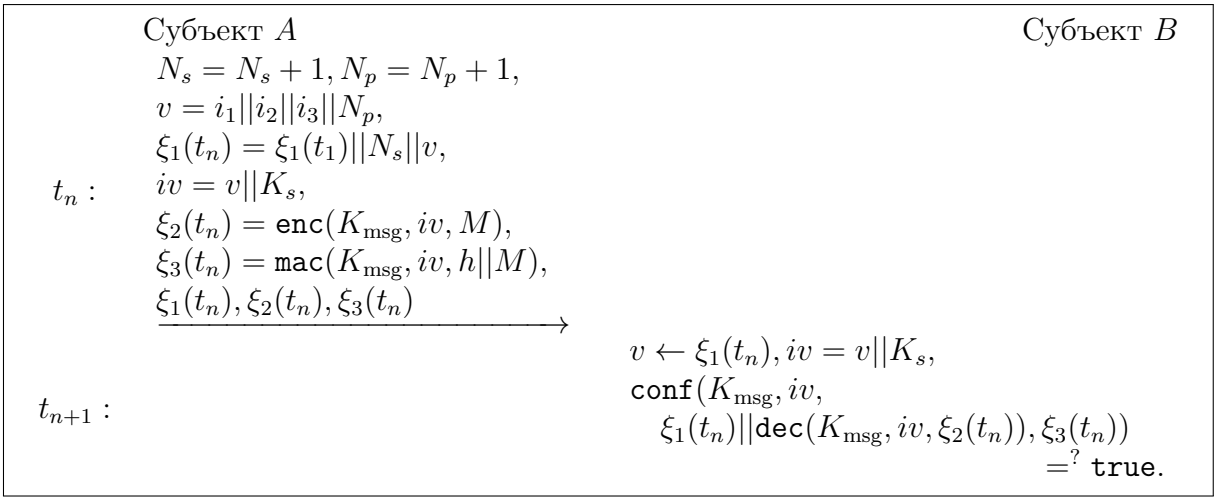

Рис. 18. Схема протокола ESP

5.2. Построение формальной модели

Построим модель состояний субъекта A, инициализирующего выполнение протокола IKEv2. Из рис. 17 видно, что количество возможных состояний субъекта A в процессе выполнения протокола равно 7, последующие состояния будем относить к протоколу ESP. В табл. 5 перечислено множество ячеек памяти, образующих $A(t_0)$ — состояние

субъекта A в момент времени t_0. Ячейки $\alpha_1, \ldots, \alpha_{22}$ относятся к протоколу IKEv2, ячейки $\alpha_{23}, \ldots, \alpha_{25}$ — к протоколу ESP.

Таблица 5

Множество состояний субъекта A

Ячейка	Значение	Ячейка	Значение
$\alpha_1(ID_A, \texttt{true})$	Идентификатор субъекта A	$\alpha_2(\varnothing, \texttt{false})$	Идентификатор субъекта B
$\alpha_3(K_{aA}, \texttt{true})$	Ключ аутентификации субъекта A	$\alpha_4(\varnothing, \texttt{false})$	Ключ проверки кода аутентификации субъекта B
$\alpha_5(K_u, \texttt{true})$	Ключ проверки кода аутентификации удостоверяющего центра	$\alpha_6(\varnothing, \texttt{false})$, $\alpha_7(\varnothing, \texttt{false})$	Идентификатор соединения $\xi_1(t_1)$ и случайное значение $\xi_2(t_1)$
$\alpha_8(\varnothing, \texttt{false})$	Случайное значение $\xi_3(t_1)$	$\alpha_9(\varnothing, \texttt{false})$, $\alpha_{10}(\varnothing, \texttt{false})$	Идентификатор соединения $\xi_1(t_3)$ и случайное значение $\xi_2(t_3)$ субъекта B
$\alpha_{11}(\varnothing, \texttt{false})$	Точка кривой $\xi_3(t_4)$, выработанная субъектом B	$\alpha_{12}(\varnothing, \texttt{false})$	Множество параметров безопасности $\xi_4(t_4)$
$\alpha_{13}(\varnothing, \texttt{false})$	Общая точка Q_{AB}	$\alpha_{14}(\varnothing, \texttt{false})$	Общая ключевая информация $SKEYSEED$
$\alpha_{15}(\varnothing, \texttt{false})$	Ключ SK_d	$\alpha_{16}(\varnothing, \texttt{false})$	Ключ iSK_A
$\alpha_{17}(\varnothing, \texttt{false})$	Ключ iSK_B	$\alpha_{18}(\varnothing, \texttt{false})$	Ключ eSK_A
$\alpha_{19}(\varnothing, \texttt{false})$	Ключ eSK_B	$\alpha_{20}(\varnothing, \texttt{false})$	Ключ aSK_A
$\alpha_{21}(\varnothing, \texttt{false})$	Ключ aSK_B	$\alpha_{22}(\varnothing, \texttt{false})$	Ключ K_{encr}
$\alpha_{23}(\varnothing, \texttt{false})$	Ключ K_{msg}	$\alpha_{24}(\varnothing, \texttt{false})$	Синхропосылка
$\alpha_{25}(\varnothing, \texttt{false})$	Принимаемое сообщение		

Последовательность последующих состояний субъекта A описывается следующими нетривиальными функциями:

$$
\begin{aligned}
A(t_1)\colon\quad & \alpha_6 \leftarrow (\xi_1(t_1), \texttt{true}),\ \alpha_7 \leftarrow (\xi_2(t_1), \texttt{true}),\ \alpha_8 \leftarrow (\xi_3(t_1), \texttt{true});\\
A(t_5)\colon\quad & \alpha_9 \leftarrow (\xi_1(t_3), \texttt{false}),\ \alpha_{10} \leftarrow (\xi_2(t_3), \texttt{false}), \alpha_{11} \leftarrow (\xi_3(t_4), \texttt{true}),\\
& \alpha_{12} \leftarrow (\xi_4(t_4), \texttt{false}), \alpha_2 \leftarrow (ID_B, \sigma),\\
& \alpha_4 \leftarrow (K_{cB},\ \texttt{verify}(a_5, \texttt{Cert}(K_{cB}) \leftarrow \xi_1(t_6))),\\
& \alpha_{13} \leftarrow (Q_{AB}, \sigma),\ \alpha_{14} \leftarrow (SKEYSEED, \sigma),\ \alpha_{15} \leftarrow (SK_d, \texttt{false});\\
A(t_7)\colon\quad & \alpha_{16} \leftarrow (iSK_A, \texttt{false}),\ \alpha_{17} \leftarrow (iSK_B, \texttt{false}),\\
& \alpha_{18} \leftarrow (eSK_A, \texttt{false}),\ \alpha_{19} \leftarrow (eSK_B, \texttt{false}),\ \alpha_{20} \leftarrow (aSK_A, \texttt{false}),\\
& \alpha_{21} \leftarrow (aSK_B, \sigma),\ \alpha_{22} = (K_{\text{encr}}, \texttt{false}),\ \alpha_{23} = (K_{\text{msg}}, \texttt{false}),\\
& \{\sigma_9, \sigma_{10}, \sigma_{12}\} \leftarrow \sigma,
\end{aligned}
$$

где $\sigma = \texttt{verify}(a_4, \underbrace{a_9||a_{10}||a_{11}||a_{12}||a_7||\texttt{prf}(aSK_B, ID_B \leftarrow \xi_1(t_6))}_{BLOB_B}, AUTH_B \leftarrow \xi_1(t_6))$.

Легко видеть, что по завершении протокола IKEv2 значения ячеек $\alpha_{15}, \ldots, \alpha_{20}$, α_{22}, α_{23} не являются подтверждёнными. С другой стороны, поскольку они однозначно выражаются через подтверждённые значения (см. (8) и (9)), будем говорить, что значения, содержащиеся в ячейках $\alpha_{15}, \ldots, \alpha_{20}, \alpha_{22}, \alpha_{23}$, подтверждены косвенно. Это же можно сказать и о производном ключе K_{msg}, используемом в протоколе ESP.

Изменение состояний субъекта A, получающего сообщения от субъекта B в ходе выполнения протокола ESP, для всех $k \geqslant 8$ описывается следующим образом:

$$
\begin{aligned}
&\alpha_{24} = v \leftarrow \xi_1(t_k)\ ||\ K_s \leftarrow a_{22},\\
&\alpha_{25} = \texttt{dec}(\alpha_{23}, \alpha_{24}, \xi_2(t_k)),\\
&\sigma_{24} = \sigma_{25} = \texttt{conf}(\alpha_{23}, \alpha_{24}, \xi_1(t_n)||\alpha_{25}, \xi_3(t_k)).
\end{aligned}
$$

Построенная модель позволяет говорить, что для протокола IKEv2 выполнены следующие свойства безопасности: С 1, С 2, С 9, С 10, С 15, С 16, С 18 (в части идентификатора инициатора протокола), С 3, С 27. Исследуя процедуры выработки производной ключевой информации, можно показать, что для протокола IKEv2 выполнены свойства С 11, С 12, С 13, С 14. Протокол ESP наследует указанные свойства, дополнительно обеспечивает свойства С 4, С 17 и содержит механизмы, необходимые для выполнения свойств С 20 и С 21.

Трудоёмкость компрометации протокола IKEv2 существенно зависит от используемых криптографических преобразований и следует из трудоёмкости решения задачи дискретного логарифмирования в группе точек эллиптической кривой $\mathcal{E}$ простого порядка q (см. рис. 15), задачи определения секретного ключа eSK_A или eSK_B по известным нарушителю значениям $\xi_1(t_5), \xi_2(t_5)$ или $\xi_1(t_6), \xi_2(t_6)$, см. рис. 17, задачи подделки кодов аутентификации $\xi_2(t_5)$ или $\xi_2(t_6)$ при неизвестном значении аутентифицируемого сообщения M_A или M_B (см. рис. 17), задачи подделки значения электронной подписи под сообщениями $BLOB_A$ или $BLOB_B$ (см. равенства (10)).

Для компрометации протокола ESP нарушителю достаточно уметь решать задачи определения ключа шифрования K_{msg} и навязывания кода аутентификации сообщения $\xi_3(t_n)$, см. рис. 18.

Авторы выражают благодарность В. А. Смыслову, С. В. Матвееву и В. Н. Цыпышеву за содержательные замечания и помощь при проведении анализа протоколов семейства IPSec.

ЛИТЕРАТУРА

1. ГОСТ Р ИСО/МЭК 27033-1:2011. Информационная технология. Методы и средства обеспечения безопасности. Безопасность сетей. Ч. 1. Обзор и концепции. М.: Стандартинформ, 2012. 73 с.
2. Р 1323565.1.012-2017. Информационная технология. Криптографическая защита информации. Принципы разработки и модернизации шифровальных (криптографических) средств защиты информации. М.: Стандартинформ, 2017. 28 с.
3. *Dolev D. and Yao A.* On the security of public key protocols // IEEE Trans. Inform. Theory. 1983. V. 29. No. 2. P. 198–208.
4. *Basin D. and Cremers C.* Modeling and analyzing security in the presence of compromising adversaries // LNCS. 2010. V. 6345. P. 340–356.
5. *Lowe G.* Breaking and fixing the Needham — Schroeder Public-Key Protocol using FDR // LNCS. 1996. V. 1055. P. 1–20.
6. ГОСТ Р 50922-2006. Защита информации. Основные термины и определения. `https://docs.cntd.ru/document/1200058320`. 2008.
7. *Bellare M. and Rogaway P.* Entity authentication and key distribution // LNCS. 1993. V. 773. P. 232–249.
8. *Bellare M., Pointcheval D., and Rogaway P.* Authenticated key exchange secure against dictionary attacks // LNCS. 2000. V. 1807. P. 139–155.
9. *Bellare M. and Rogaway P.* Provably secure session key distribution — the three party case // 27th ACM Symp. Theory Computing. ACM Press, 1995. P. 57–66.
10. *Blake-Wilson S., Johnson D., and Menezes A.* Key agreement protocols and their security analysis // LNCS. 1997. V. 1355. P. 30–45.
11. *Blake-Wilson S. and Menezes A.* Entity authentication and authenticated key transport protocols employing asymmetric techniques // LNCS. 1998. V. 1361. P. 137–158.

12. *Canetti R. and Krawczyk H.* Analysis of key-exchange protocols and their use for building secure channels // LNCS. 2001. V. 2045. P. 453–474.
13. *LaMacchia B., Lauter K., and Mityagin A.* Stronger security of authenticated key exchange // LNCS. 2007. V. 4784. P. 1–16.
14. *Krawczyk H.* HMQV: A high-performance secure Diffie — Hellman protocol // LNCS. 2005. V. 3621. P. 546–566.
15. *Menezes A. and Ustaoglu B.* On the importance of public-key validation in the MQV and HMQV key agreement protocols // LNCS. 2006. V. 4329. P. 133–147.
16. *Rabin M.* Digitized Signatures and Public Key Functions as Intractable as Factorization. Technical Report: MIT/LCS/TR-212. MIT Laboratory for Computer Science, Cambridge, 1979.
17. *Goldwasser S. and Micali S.* Probabilistic encryption // J. Computer System Sci. 1984. V. 28. P. 270–299.
18. *Mao W.* Modern Cryptography: Theory and Practice. Prentice Hall, New Jersey, 2003. 707 p.
19. *Boyd C., Mathuria A., and Stebila D.* Protocols for Authentication and Key Establishment. Second Ed. Berlin; Heidelberg: Springer Verlag, 2020. 521 p.
20. *Алферов А. П., Зубов А. Ю., Кузьмин А. С., Черемушкин А. В.* Основы криптографии. М.: Гелиос АРВ, 2002. 480 с.
21. *Бабаш А. В., Шанкин Г. П.* Криптография. М.: Солон-Пресс, 2007. 512 с.
22. *Алексеев Е. К., Ахметзянова Л. Р., Ошкин И. Б., Смышляев С. В.* Обзор уязвимостей некоторых протоколов выработки общего ключа с аутентификацией на основе пароля и принципы построения протокола SESPAKE // Математические вопросы криптографии. 2016. Т. 7. № 4. С. 7–28.
23. *Ahmetzyanova L. R., Alekseev E. K., Sedov G. K., et al.* Practical significance of security bounds for standardized internally re-keyed block cipher modes // Математические вопросы криптографии. 2019. Т. 10. № 2. С. 31–46.
24. Р 1323565.1.030-2020. Информационная технология. Криптографическая защита информации. Использование криптографических алгоритмов в протоколе безопасности транспортного уровня (TLS 1.3). М.: Стандартинформ, 2020. 73 с.
25. Р 1323565.1.028-2018. Информационная технология. Криптографическая защита информации. Криптографические механизмы защищенного взаимодействия контрольных и измерительных устройств. М.: Стандартинформ, 2019. 66 с.
26. *Нестеренко А. Ю.* Об одном подходе к построению защищенных соединений // Математические вопросы криптографии. 2013. Т. 4. №. 2. С. 101–111.
27. *Нестеренко А. Ю., Лебедев П. А., Семенов А. М.* Краткий анализ криптографических механизмов защищенного взаимодействия контрольных и измерительных устройств. Технический комитет по стандартизации «Криптографическая защита информации». «Криптографические исследования». 2019. Сер. б/н. `https://tc26.ru/standarts/kriptograficheskie-issledovaniya/`.
28. *Semenov A. M.* Analysis of Russian key-agreement protocols using automated verification tools // Математические вопросы криптографии. 2017. Т. 8. № 2. С. 131–142.
29. Р 1323565.1.035–2021. Информационная технология. Криптографическая защита информации. Использование российских криптографических алгоритмов в протоколе защиты информации ESP. М.: Стандартинформ, 2021. 52 с.
30. *Черемушкин А. В.* Криптографические протоколы: основные свойства и уязвимости // Прикладная дискретная математика. Приложение. 2009. № 2. С. 115–150.
31. IETF. RFC 3552. Guidelines for Writing RFC Text on Security Considerations. 2003. `https://tools.ietf.org/html/rfc3552`.

32. The AVISPA Project. Properties (Goals). 2021. http://www.avispa-project.org/delivs/6.1/d6-1/node3.html.
33. ГОСТ Р 53113.1-2008 Информационная технология. Защита информационных технологий и автоматизированных систем от угроз информационной безопасности, реализуемых с использованием скрытых каналов. Ч. 1. Общие положения. М.: Стандартинформ, 2008. 12 с.
34. ГОСТ Р 53113.2-2009 Информационная технология. Защита информационных технологий и автоматизированных систем от угроз информационной безопасности, реализуемых с использованием скрытых каналов. Ч. 2. Рекомендации по организации защиты информации, информационных технологий и автоматизированных систем от атак с использованием скрытых каналов. М.: Стандартинформ, 2009. 12 с.
35. *Видякин В. В.* О связи скрытых информационных каналов и субпротоколов // Обозрение прикл. и промышл. матем. 2006. Т. 13. Вып. 1. С. 87–88.
36. *Князев А. В., Ронжин А. Ф.* Инструментальный анализ мутных протоколов // Обозрение прикл. и промышл. матем. 2007. Т. 14. Вып. 4. С. 577–646.
37. *Матвеев С. В.* Некоторые подходы к оценке пропускной способности скрытых каналов в IP-сетях // Системы высокой доступности. 2012. Т. 8. Вып. 2. С. 68–71.
38. *Blake-Wilson S. and Menezes A.* Unknown key-share attacks on the Station-to-Station (STS) protocol // LNCS. 1999. V. 1560. P. 154–170.
39. *Diffie W., van Oorschot P., and Wiener M.* Authentication and authenticated key exchanges // Des. Codes Crypt. 1992. V. 2. P. 107–125.
40. IETF. RFC 8654. Extended Message Support for BGP. 2019. https://tools.ietf.org/html/rfc8654.
41. IETF. RFC 3748. Extensible Authentication Protocol (EAP). 2004. https://tools.ietf.org/html/rfc3748.
42. IETF. RFC 7029. Extensible Authentication Protocol (EAP) Mutual Cryptographic Binding. 2013. https://tools.ietf.org/html/rfc7029.
43. *Cremers C.* Scyther — Semantics and Verification of Security Protocols. Ph.D. Thesis. Eindhoven Univ. Technology, 2006. 205 p.
44. Proverif: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial. 2020. 150 p. http://prosecco.gforge.inria.fr/personal/bblanche/proverif/manual.pdf.
45. ГОСТ Р 58833-2020. Защита информации. Идентификация и аутентификация. Общие положения. М.: Стандартинформ, 2020. 28 с.
46. Словарь криптографических терминов / под ред. Б. А. Погорелова и В. Н. Сачкова. М.: МЦМНО. 2006. 94 с.
47. ГОСТ Р ИСО/МЭК 9594-8-98. Информационная технология. Взаимосвязь открытых систем. Справочник. Ч. 8. Основы аутентификации. М.: Стандартинформ, 2001. 29 с.
48. *Fletcher J. G.* An arithmetic checksum for serial transmissions // IEEE Trans. Communications. 1982. V. 30. No. 1. P. 247–252.
49. *Peterson W. W. and Brown D. T.* Cyclic codes for error detection // Proc. IRE. 1961. V. 49. No. 1. P. 228–235. doi:10.1109/JRPROC.1961.287814.
50. ГОСТ Р 34.11-2012. Информационная технология. Криптографическая защита информации. Функция хэширования. М.: Стандартинформ, 2012. 25 с.
51. *Alashwali E. and Rasmussen K.* What's in a Downgrade? A Taxonomy of Downgrade Attacks in the TLS Protocol and Application Protocols Using TLS. Cryptology ePrint Archive. 2019. Report 2019/1083. https://eprint.iacr.org/2019/1083.
52. *Качалин И. Ф., Кузьмин А. С., Суслов Е. А. и др.* Об основных концепциях криптографической стойкости // Тезисы XII Всерос. школы-коллоквиума по стохастическим ме-

тодам и VI Всерос. симпозиума по прикладной и промышленной математике. Сочи-Дагомыс, 1–7 октября 2005 г. С. 982–983.

53. *Лось А. Б., Нестеренко А. Ю., Рожков М. И.* Криптографические методы защиты информации. М.: Изд-во Юрайт, 2016. 473 с.
54. IETF. RFC 4303. IP Encapsulating Security Payload (ESP). 2005. `https://datatracker.ietf.org/doc/html/rfc4303`.
55. IETF. RFC 7296. Internet Key Exchange Protocol Version 2 (IKEv2). 2014. `https://datatracker.ietf.org/doc/html/rfc7296`.
56. *Krawczyk H.* SIGMA: The 'SIGn-and-MAc' approach to authenticated Diffie — Hellman and its use in the IKE protocols // LNCS. 2003. V. 2729. P. 400–425.
57. Р 50.1.113-2016. Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования. М.: Стандартинформ, 2016. 28 с.

REFERENCES

1. GOST R ISO/MEK 27033-1:2011. Informatsionnaya tekhnologiya. Metody i sredstva obespecheniya bezopasnosti. Bezopasnost' setey. Ch. 1. Obzor i kontseptsii. [Information Technology. Security Techniques. Network Security. P. 1. Overview and Concepts.] Moscow, Standartinform, 2012. 73 p. (in Russian)
2. R 1323565.1.012-2017. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Printsipy razrabotki i modernizatsii shifroval'nykh (kriptograficheskikh) sredstv zashchity informatsii [Information Technology. Cryptographic Data Security. Principles of Creation and Modernization for Cryptographic Modules]. Moscow, Standartinform, 2017. 28 p. (in Russian)
3. *Dolev D. and Yao A.* On the security of public key protocols. IEEE Trans. Inform. Theory, 1983, vol. 29, no. 2, pp. 198–208.
4. *Basin D. and Cremers C.* Modeling and analyzing security in the presence of compromising adversaries. LNCS, 2010, vol. 6345, pp. 340–356.
5. *Lowe G.* Breaking and fixing the Needham — Schroeder Public-Key Protocol using FDR. LNCS, 1996, vol. 1055, pp. 1–20.
6. GOST R 50922-2006. Zashchita informatsii. Osnovnye terminy i opredeleniya [Data Protection. Basic Terms and Definitions]. `https://docs.cntd.ru/document/1200058320`. 2008. (in Russian)
7. *Bellare M. and Rogaway P.* Entity authentication and key distribution. LNCS, 1993, vol. 773, pp. 232–249.
8. *Bellare M., Pointcheval D., and Rogaway P.* Authenticated key exchange secure against dictionary attacks. LNCS, 2000, vol. 1807, pp. 139–155.
9. *Bellare M. and Rogaway P.* Provably secure session key distribution — the three party case. 27th ACM Symp. Theory Computing, ACM Press, 1995, pp. 57–66.
10. *Blake-Wilson S., Johnson D., and Menezes A.* Key agreement protocols and their security analysis. LNCS, 1997, vol. 1355, pp. 30–45.
11. *Blake-Wilson S. and Menezes A.* Entity authentication and authenticated key transport protocols employing asymmetric techniques. LNCS, 1998, vol. 1361, pp. 137–158.
12. *Canetti R. and Krawczyk H.* Analysis of key-exchange protocols and their use for building secure channels. LNCS, 2001, vol. 2045, pp. 453–474.
13. *LaMacchia B., Lauter K., and Mityagin A.* Stronger security of authenticated key exchange. LNCS, 2007, vol. 4784, pp. 1–16.

14. *Krawczyk H.* HMQV: A high-performance secure Diffie — Hellman protocol. LNCS, 2005, vol. 3621, pp. 546–566.
15. *Menezes A. and Ustaoglu B.* On the importance of public-key validation in the MQV and HMQV key agreement protocols. LNCS, 2006, vol. 4329, pp. 133–147.
16. *Rabin M.* Digitized Signatures and Public Key Functions as Intractable as Factorization. Technical Report: MIT/LCS/TR-212, MIT Laboratory for Computer Science, Cambridge, 1979.
17. *Goldwasser S. and Micali S.* Probabilistic encryption. J. Computer System Sci., 1984, vol. 28, pp. 270–299.
18. *Mao W.* Modern Cryptography: Theory and Practice. Prentice Hall, New Jersey, 2003. 707 p.
19. *Boyd C., Mathuria A., and Stebila D.* Protocols for Authentication and Key Establishment. Second Ed. Berlin; Heidelberg, Springer Verlag, 2020. 521 p.
20. *Alferov A. P., Zubov A. Yu., Kuz'min A. S., and Cheremushkin A. V.* Osnovy kriptografii [Fundamentals of Cryptography]. Moscow, Gelios ARV, 2002. 480 p. (in Russian)
21. *Babash A. V. and Shankin G. P.* Kriptografiya [Cryptography]. Moscow, Solon-Press, 2007. 512 p. (in Russian)
22. *Alekseev E. K., Akhmetzyanova L. R., Oshkin I. B., and Smyshlyaev S. V.* Obzor uyazvimostey nekotorykh protokolov vyrabotki obshchego klyucha s autentifikatsiey na osnove parolya i printsipy postroeniya protokola SESPAKE [A review of the password authenticated key exchange protocols vulnerabilities and principles of the SESPAKE protocol construction]. Matematicheskie Voprosy Kriptografii, 2016, vol. 7, no. 4, pp. 7–28. (in Russian)
23. *Ahmetzyanova L. R., Alekseev E. K., Sedov G. K., et al.* Practical significance of security bounds for standardized internally re-keyed block cipher modes. Matematicheskie Voprosy Kriptografii, 2019, vol. 10, no. 2, pp. 31–46.
24. R 1323565.1.030-2020. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Ispol'zovanie kriptograficheskikh algoritmov v protokole bezopasnosti transportnogo urovnya (TLS 1.3) [Information Technology. Cryptographic Data Security. The Use of the Russian Cryptographic Algorithms in the Transport Layer Security Protocol (TLS 1.3)]. Moscow, Standartinform, 2020. 73 p. (in Russian)
25. R 1323565.1.028-2018. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Kriptograficheskie mekhanizmy zashchishchennogo vzaimodeystviya kontrol'nykh i izmeritel'nykh ustroystv [Information Technology. Cryptographic Data Security. Cryptographic Mechanisms of Secure Interactions of Control and Measuring Devices]. Moscow, Standartinform, 2019. 66 p. (in Russian)
26. *Nesterenko A. Yu.* Ob odnom podkhode k postroeniyu zashchishchennykh soedineniy [On an approach to the construction of secure connections]. Matematicheskie Voprosy Kriptografii, 2013, vol. 4, no. 2, pp. 101–111. (in Russian)
27. *Nesterenko A. Yu., Lebedev P. A., and Semenov A. M.* Kratkiy analiz kriptograficheskikh mekhanizmov zashchishchennogo vzaimodeystviya kontrol'nykh i izmeritel'nykh ustroystv [Brief Analysis of Cryptographic Mechanisms of Secure Interaction of Control and Measuring Devices]. Technical Committee on Standardization "Cryptographic Protection of Information". Ser. n/n "Cryptographic research", 2019. `https://tc26.ru/standarts/kriptograficheskie-issledovaniya/`. (in Russian)
28. *Semenov A. M.* Analysis of Russian key-agreement protocols using automated verification tools. Matematicheskie Voprosy Kriptografii, 2017, vol. 8, no. 2, pp. 131–142.
29. R 1323565.1.035–2021. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Ispol'zovanie rossiyskikh kriptograficheskikh algoritmov v protokole zashchity informatsii ESP. [Information Technology. Cryptographic Protection of Information. Using

Russian Cryptographic Algorithms in the ESP Information Protection Protocol]. Moscow, Standartinform, 2021. 52 p. (in Russian)

30. *Cheremushkin A. V.* Kriptograficheskie protokoly: osnovnye svoystva i uyazvimosti [Cryptographic protocols: main properties and vulnerabilities]. Prikladnaya Diskretnaya Matematika. Prilozhenie, 2009, no. 2, pp. 115–150. (in Russian)
31. IETF. RFC 3552. Guidelines for Writing RFC Text on Security Considerations, 2003. `https://tools.ietf.org/html/rfc3552`.
32. The AVISPA Project. Properties (Goals), 2021. `http://www.avispa-project.org/delivs/6.1/d6-1/node3.html`.
33. GOST R 53113.1-2008 Informatsionnaya tekhnologiya. Zashchita informatsionnykh tekhnologiy i avtomatizirovannykh sistem ot ugroz informatsionnoy bezopasnosti, realizuemykh s ispol'zovaniem skrytykh kanalov. Ch. 1. Obshchie polozheniya [Information Technology. Protection of Information Technology and Automated Systems against Security Threats Posed by Use of Covert Channels. P. 1. General Principles]. Moscow, Standartinform, 2008. 12 p. (in Russian)
34. GOST R 53113.2-2009 Informatsionnaya tekhnologiya. Zashchita informatsionnykh tekhnologiy i avtomatizirovannykh sistem ot ugroz informatsionnoy bezopasnosti, realizuemykh s ispol'zovaniem skrytykh kanalov. Ch. 2. Rekomendatsii po organizatsii zashchity informatsii, informatsionnykh tekhnologiy i avtomatizirovannykh sistem ot atak s ispol'zovaniem skrytykh kanalov [Information Technology. Protection of Information Technology and Automated Systems against Security Threats Posed by Use of Covert Channels. P. 2. Recommendations on Protecting Information, Information Technology and Automated Systems against Covert Channel Attacks]. Moscow, Standartinform, 2009. 12 p. (in Russian)
35. *Vidyakin V. V.* O svyazi skrytykh informatsionnykh kanalov i subprotokolov [On the connection of hidden information channels and subprotocols]. Obozrenie Prikl. i Promyshl. Matem., 2006, vol. 13, iss. 1, pp. 87–88. (in Russian)
36. *Knyazev A. V. and Ronzhin A. F.* Instrumental'nyy analiz mutnykh protokolov [Instrumental analysis of turbid protocols]. Obozrenie Prikl. i Promyshl. Matem., 2007, vol. 14, iss. 4, pp. 577–646. (in Russian)
37. *Matveev S. V.* Nekotorye podkhody k otsenke propusknoy sposobnosti skrytykh kanalov v IP-setyakh [Some approaches to estimating the bandwidth of hidden channels in IP-networks]. Sistemy Vysokoy Dostupnosti, 2012, vol. 8, iss. 2, pp. 68–71. (in Russian)
38. *Blake-Wilson S. and Menezes A.* Unknown key-share attacks on the Station-to-Station (STS) protocol. LNCS, 1999, vol. 1560, pp. 154–170.
39. *Diffie W., van Oorschot P., and Wiener M.* Authentication and authenticated key exchanges. Des. Codes Crypt., 1992, vol. 2, pp. 107–125.
40. IETF. RFC 8654. Extended Message Support for BGP. 2019. `https://tools.ietf.org/html/rfc8654`.
41. IETF. RFC 3748. Extensible Authentication Protocol (EAP). 2004. `https://tools.ietf.org/html/rfc3748`.
42. IETF. RFC 7029. Extensible Authentication Protocol (EAP) Mutual Cryptographic Binding. 2013. `https://tools.ietf.org/html/rfc7029`.
43. *Cremers C.* Scyther — Semantics and Verification of Security Protocols. Ph.D. Thesis, Eindhoven Univ. Technology, 2006. 205 p.
44. Proverif: Automatic Cryptographic Protocol Verifier, User Manual and Tutorial. 2020. 150 p. `http://prosecco.gforge.inria.fr/personal/bblanche/proverif/manual.pdf`.

45. GOST R 58833-2020. Zashchita informatsii. Identifikatsiya i autentifikatsiya. Obshchie polozheniya [Information Protection. Identification and Authentication. General]. Moscow, Standartinform, 2020. 28 p. (in Russian)
46. Slovar' kriptograficheskikh terminov [Dictionary of Cryptographic Terms]. B. A. Pogorelov and V. N. Sachkov (eds.). Moscow, MCCME publ., 2006. 94 p. (in Russian)
47. GOST R ISO/MEK 9594-8-98. Informatsionnaya tekhnologiya. Vzaimosvyaz' otkrytykh sistem. Spravochnik. Ch. 8. Osnovy autentifikatsii [Information Technology. Open Systems Interconnection. The Directory. P. 8. Authentication Framework]. Moscow, Standartinform, 2001. 29 p. (in Russian)
48. *Fletcher J. G.* An arithmetic checksum for serial transmissions. IEEE Trans. Communications, 1982, vol. 30, no. 1, pp. 247–252.
49. *Peterson W. W. and Brown D. T.* Cyclic codes for error detection. Proc. IRE, 1961, vol. 49, no. 1, pp. 228–235. doi:10.1109/JRPROC.1961.287814.
50. GOST R 34.11-2012. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Funktsiya kheshirovaniya [Information Technology. Cryptographic Data Security. Hash Function]. Moscow, Standartinform, 2012. 25 p. (in Russian)
51. *Alashwali E. and Rasmussen K.* What's in a Downgrade? A Taxonomy of Downgrade Attacks in the TLS Protocol and Application Protocols Using TLS. Cryptology ePrint Archive, 2019, Report 2019/1083. https://eprint.iacr.org/2019/1083.
52. *Kachalin I. F., Kuz'min A. S., Suslov E. A., et al.* Ob osnovnykh kontseptsiyakh kriptograficheskoy stoykosti [About main conceptions of cryptographic security]. Proc. XII All-Russian School-Colloquium on Stochastic Methods and VI All-Russian Symp. Appl. Industr. Math., Sochi, Dagomys, October 1–7, 2005, pp. 982–983. (in Russian)
53. *Los' A. B., Nesterenko A. Yu., and Rozhkov M. I.* Kriptograficheskie metody zashchity informatsii [Cryptographic methods of data security]. Moscow, Yurayt Publ., 2016. 473 p. (in Russian)
54. IETF. RFC 4303. IP Encapsulating Security Payload (ESP), 2005. https://datatracker.ietf.org/doc/html/rfc4303.
55. IETF. RFC 7296. Internet Key Exchange Protocol Version 2 (IKEv2), 2014. https://datatracker.ietf.org/doc/html/rfc7296.
56. *Krawczyk H.* SIGMA: The 'SIGn-and-MAc' approach to authenticated Diffie — Hellman and its use in the IKE protocols. LNCS, 2003, vol. 2729, pp. 400–425.
57. R 50.1.113-2016. Informatsionnaya tekhnologiya. Kriptograficheskaya zashchita informatsii. Kriptograficheskie algoritmy, soputstvuyushchie primeneniyu algoritmov elektronnoy tsifrovoy podpisi i funktsii kheshirovaniya [Information Technology. Cryptographic Data Security. Cryptographic Algorithms to Accompany the Usage of Digital Signature and Hash Function Algorithms]. Moscow, Standartinform, 2016. 28 p. (in Russian)