Федеральное государственное автономное образовательное учреждение
высшего образования «Санкт-Петербургский государственный
электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)

На правах рукописи

Березин Андрей Николаевич

МЕТОДЫ ПОВЫШЕНИЯ УРОВНЯ БЕЗОПАСНОСТИ ЗАЩИТНЫХ ПРЕОБРАЗОВАНИЙ ИНФОРМАЦИИ

Специальность 05.13.19 —
«Методы и системы защиты информации, информационная безопасность»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
доктор технических наук, профессор
Молдовян Николай Андреевич

Санкт-Петербург — 2016

Оглавление

Стр.

Обозначения

ЭЦП	электронная цифровая подпись
ЗФ	задача факторизации
ЗДЛ	задача дискретного логарифмирования
НОД	наибольший общий делитель
НОК	наименьшее общее кратное
ОК	открытый ключ
СК	секретный ключ
УЦ	удостоверяющий центр
КШ	коммутативное шифрование
УГЭЦП	утверждаемая групповая электронная цифровая подпись

Сокращения

$\|n\|$	битовая длина числа n
$\varphi(n)$	функция Эйлера от натурального числа n
$n \in S$	элемент n принадлежит множеству S
$p \| n$	число p делит n
$p \nmid n$	число p не делит n
$p \| \| n$	конкатенация битовых строк p и n
НОД(n,p)	наибольший общий делитель чисел n и p
НОК(n,p)	наименьшее общее кратное чисел n и p
$L(n)$	обобщённая функция Эйлера от натурального числа n
$\mathbb{Z}$	множество целых чисел
$\mathbb{N}$	множество натуральных чисел {1,2,...}

$\mathbb{Z}_n$	кольцо вычетов по модулю натурального числа n
$\mathbb{Z}_n^*$	мультипликативная группа кольца вычетов по модулю целого числа n
$a = b \bmod n$	число a сравнимо с числом b по модулю n, т.е. $a = b + Qn$, где $Q \in \mathbb{Z}$
$a \leftarrow_{\$} \mathbb{Z}_n$	сгенерировать случайное число a из множества $\mathbb{Z}_n$
$a \leftarrow_{\$} \{0,1\}^{256}$	сгенерировать случайное натуральное число a длиной 256 бит
$a \leftarrow_{\$} (\mathbb{Z}, a < B)$	сгенерировать случайное число a из множества $\mathbb{Z}$, такое, что выполняется условие $a < B$
$prime$	простое число
$\alpha, n, \gamma \leftarrow Gen()$	сгенерировать набор параметров (α, n, γ)
$y, x \leftarrow GenK(a, n)$	сгенерировать пару ключей (y, x) с использованием значений a, n и уравнения генерации ОК
ОК: (y, α, n)	ОК состоящий из набора значений (y, α, n)

Определения

Информация: сведения (сообщения, данные) независимо от формы их представления [1].

Информационные технологии: процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов [1].

Информационная система: совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств [1].

Информационно-телекоммуникационная сеть: технологическая система, предназначенная для передачи по линиям связи информации, доступ к которой осуществляется с использованием средств вычислительной техники [1].

Безопасность информационно-телекоммуникационных технологий (безопасность ИТТ): все аспекты, связанные с определением, достижением и поддержанием конфиденциальности, целостности, доступности, неотказуемости, подотчетности, аутентичности и достоверности информационно-телекоммуникационных технологий [2].

Информационная безопасность: все аспекты, связанные с определением, достижением и поддержанием конфиденциальности, целостности, доступности, неотказуемости, подотчетности, аутентичности и достоверности информации или средств ее обработки [2].

Безопасность информации (при применении информационных технологий): состояние защищенности информационной технологии, обеспечивающее безопасность информации, для обработки которой она применяется, и информационную безопасность автоматизированной информационной системы, в которой она реализована [3].

Конфиденциальность (информации [ресурсов автоматизированной информационной системы]): состояние информации [ресурсов автоматизированной информационной системы], при котором доступ к ней [к ним] осуществляют только субъекты, имеющие на него право [3].

Целостность (информации [ресурсов автоматизированной информационной системы]): состояние информации [ресурсов автоматизированной информационной системы], при котором ее [их] изменение осуществляется только преднамеренно субъектами, имеющими на него право [3].

Доступность (информации [ресурсов автоматизированной информационной системы]): состояние информации [ресурсов автоматизированной информационной системы], при котором субъекты, имеющие право доступа, могут реализовать их беспрепятственно [3].

Аутентичность: свойство, гарантирующее, что субъект или ресурс идентичны заявленным [2].

Неотказуемость: способность удостоверять имевшее место действие или событие так, чтобы эти события или действия не могли быть позже отвергнуты [2].

Подотчетность: состояние ресурсов автоматизированной информационной системы, при котором обеспечиваются их идентификация и регистрация [3].

Аутентификация: проверка принадлежности субъекту доступа предъявленного им идентификатора; подтверждение подлинности [4].

Электронная цифровая подпись: дополнительные данные или криптографическое преобразование какого-либо блока данных, позволяющие получателю блока данных убедиться в подлинности отправителя и целостности блока данных и защитить его от искажения с помощью, например, средств получателя [5].

Протокол: совокупность действий (инструкций, команд, вычислений, алгоритмов), выполняемых в заданной последовательности двумя или более субъектами с целью достижения определенного результата [6].

Введение

Актуальность темы исследования. Для защиты информации в информационно-телекоммуникационных технологиях широко используются алгоритмические средства, стойкость которых базируется на вычислительной сложности некоторых задач, в частности задач факторизации и дискретного логарифмирования по простому модулю. Безопасность таких алгоритмов определяется с точки зрения теории сложности и оценивается количеством операций (стойкостью), необходимых для решения вычислительно трудной задачи. При определении стойкости дополнительно учитывают стоимость оборудования, которое может быть использовано на текущий момент времени, и научные достижения в области новой вычислительной техники. Однако при таком подходе к оцениванию безопасности алгоритмов остаётся неучтённой вероятность появления прорывных методов решения используемой вычислительно трудной задачи.

В последнее время предлагается понятие интегрального показателя безопасности (Молдовян Н.А., 2013). Данный показатель учитывает стойкость используемых вычислительно трудных задач и вероятность того, что будут найдены прорывные алгоритмы решения в области вычислительно сложных задач, используемых в алгоритмических средствах защиты информации. Существенное повышение указанного показателя может быть достигнуто путём разработки алгоритмов защиты информации, для взлома которых потребуется решать две независимые вычислительно сложные задачи.

Вопросы безопасности приобретают ключевое значение в современных информационно-телекоммуникационных технологиях, в связи с этим тема диссертационной работы «Методы повышения уровня безопасности защитных преобразований информации» представляется своевременной и актуальной.

Степень разработанности темы. Исследования посвящённые вопросам разработки алгоритмов защиты информации, основанных на двух трудных задачах приведены в работах следующих авторов: Aboud S.J., Ahmad R.R., Al-Fayoumi M., Bao H., Binh D.V., Brickell E.F., Cao Z., Chang C.C., Chen C.Y., Chen T.H., Chen Y., Chiou S.Y., Ciss A.A., Desai C.G., Gao Y., Giang N.T., Girault M., Harn L., He J., He W., He Y., Hijazi M.S., Horng G., Huang S., Hwang M.S., Hwang T., Ismail E.S., Jeng A.B., Kiesler T., Kuo W.C., Laih C.S., Lee N.Y., Lee W.B., Li C.,

Li J., Li L., Lin C.H., Lin H.F., Liu J., Lu E., Madhur K., McCurley K.S., Minh N.H., Nimbalkar A.B., Pointcheval D., Pon S., Qian H., Shao Z., Sharma B., Shatnawi S.M., Shmuely Z., Shrivastava V., Tahat N.M., Tiersma H.J., Trika P., Tzeng S.F., Verma S., Vijay A., Vishnoi S., Wang C.T., Wei S., Xiao G., Yang C.C., Yang C.Y., Youssef A., Yu T., Zheng J., Васильев И.Н., Головачёв Д.А., Гортинская Л.В., Дернова Е.С., Костина А.А., Латышев Д.М., Молдовян А.А., Молдовян Д.Н., Молдовян Н.А., Нгуен Л.М., Рыжков А.В., Сухов Д.К., Щербаков В.А. В подавляющем большинстве работ указанных авторов для повышения уровня безопасности, оцениваемому по интегральному критерию, предложены только единичные алгоритмы аутентификации, электронной цифровой подписи, обмена секретом, в основном использующие модуль $p = 2n + 1$, где n - трудно факторизуемое число, для взлома которых требуется решить две вычислительно сложные задачи, однако известные методы их построения не позволяют реализовать алгоритмы защиты информации других типов, что сдерживает их применение на практике.

Решаемая научно-техническая задача. Решается задача разработки методов и алгоритмов защиты информации в процессе её сбора, хранения, обработки, передачи и распространения.

Цель и задачи исследования. Целью диссертационного исследования является повышение уровня информационной безопасности информационно-телекоммуникационных технологий. Решение сформулированной научно-технической задачи предусматривало:

1. Разработку метода построения алгоритмов и протоколов для применения в средствах защиты информации, обладающих повышенным уровнем безопасности, который позволит расширить виды алгоритмов и протоколов указанного типа.
2. Разработку протоколов локальной и удалённой аутентификации пользователей, субъектов и объектов информационных процессов, обладающих повышенным уровнем безопасности.
3. Разработку протоколов обеспечения конфиденциальности информации, передаваемой по открытым каналам связи, обладающих повышенным уровнем безопасности.
4. Разработку протоколов обеспечения анонимности в открытых компьютерных сетях, обладающих повышенным уровнем безопасности.

Научная новизна

1. Впервые предложен метод построения алгоритмов и протоколов, нарушение безопасности которых требует одновременного решения двух вычислительно сложных задач, отличающийся использованием задачи дискретного логарифмирования (ЗДЛ) по трудно факторизуемому модулю n, размер множителей которого выбирается таким образом, что, по крайней мере, решение ЗДЛ по модулю одного из делителей модуля имеет вычислительную сложность не ниже заданного уровня стойкости.
2. На основе предложенного метода разработаны новые протоколы аутентификации объектов и субъектов информационных процессов, обладающие повышенным уровнем безопасности: протокол электронной цифровой подписи (ЭЦП), отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры, благодаря чему достигнуто снижение размера, вычислительной сложности процедур генерации и проверки ЭЦП; протокол коллективной ЭЦП, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры, благодаря чему достигнуто снижение размера, вычислительной сложности процедур генерации и проверки ЭЦП; протокол утверждаемой групповой ЭЦП, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры и механизма маскирования ключей, благодаря чему руководитель и только он может доказывать стороннему проверяющему список лиц, которые подписывали документ, без разглашения секретных ключей подчинённых и своего собственного; протокол интерактивной аутентификации субъекта, отличающийся использованием ЗДЛ по трудно факторизуемому модулю n специальной структуры; протокол двухшаговой аутентификации субъекта, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры и элемента выделенной подгруппы мультипликативной группы кольца вычетов по модулю n в качестве запроса, благодаря чему достигнута возможность безопасной аутентификации субъекта за два шага.
3. На основе предложенного метода разработаны новые протоколы защиты информации, обладающие повышенным уровнем безопасности: прото-

колы обмена ключами, отличающиеся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры и рандомизирующего параметра, благодаря чему обеспечивается случайность значения ключа, формируемого в ходе протокола; протокол защитного преобразования информации, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры; протокол коммутативного защитного преобразования информации, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры и механизма расщепления сообщений, благодаря чему обеспечивается возможность выполнения защитных преобразований для произвольных сообщений; протокол стойкого защитного преобразования информации с использованием ключа малого размера, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры и процедуры бесключевого защитного преобразования совместно с аутентификацией по коротким ключам, благодаря чему возможно задать стойкость протокола, для малых длин ключа.

4. На основе предложенного метода разработаны новые протоколы обеспечения анонимности, обладающие повышенным уровнем безопасности: протокол слепой ЭЦП, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры, благодаря чему достигнуто снижение размера, вычислительной сложности процедур генерации и проверки ЭЦП; протокол слепой коллективной ЭЦП, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры, благодаря чему достигнуто снижение размера, вычислительной сложности процедур генерации и проверки ЭЦП.

Теоретическая и практическая значимость работы определяется тем, что предложен метод построения алгоритмов и протоколов, имеющих повышенный уровень безопасности, свободный от недостатков существующих аналогов, а разработанные протоколы аутентификации, обеспечения конфиденциальности и анонимности, имеют широкое применение в информационно-телекоммуникационных технологиях.

Методология и методы исследования. При выполнении диссертационного исследования были использованы аппарат и методы алгебры, теории вероят-

ности, дискретной математики, теории чисел, теории сложности и информационной безопасности.

Положения, выносимые на защиту:

1. Метод построения алгоритмов и протоколов, повышенный уровень безопасности которых обеспечивается тем, что для их взлома требуется одновременно решить задачи дискретного логарифмирования и факторизации.
2. Протоколы локальной и удалённой аутентификации пользователей, объектов и субъектов информационных процессов, обладающие повышенным уровнем безопасности.
3. Протоколы обеспечения конфиденциальности информации, передаваемой по открытым каналам связи, обладающие повышенным уровнем безопасности.
4. Протоколы обеспечения анонимности в открытых компьютерных сетях, обладающие повышенным уровнем безопасности.

Достоверность и апробация результатов. Достоверность подтверждается строгими математическими доказательствами, обеспечивается анализом состояния исследований в этой области на сегодняшний день и апробацией основных результатов на конференциях различного уровня.

Основные положения работы докладывались на: 69–й научно–технической конференции профессорско–преподавательского состава университета СПбГЭТУ «ЛЭТИ» (Санкт-Петербург, 2016), IX Санкт–Петербургской межрегиональной конференции «Информационная безопасность регионов России (ИБРР–2015)» (Санкт-Петербург, 2015), 20 Санкт-Петербургской ассамблеи молодых учёных и специалистов (Санкт-Петербург, 2015), 68–й научно–технической конференции профессорско–преподавательского состава университета СПбГЭТУ «ЛЭТИ» (Санкт-Петербург, 2015), 19 Санкт-Петербургской ассамблеи молодых учёных и специалистов (Санкт-Петербург, 2014), XIV Санкт–Петербургской международной конференции «Региональная информатика – 2014» (Санкт-Петербург, 2014), VIII Санкт–Петербургской межрегиональной конференции «Информационная безопасность регионов России (ИБРР–2013)» (Санкт-Петербург, 2013), XIX международной научно–методической конференции «Современное образование: содержание, тех-

нологии, качество» (Санкт-Петербург, 2013), 18 Санкт-Петербургской ассамблеи молодых учёных и специалистов (Санкт-Петербург, 2013), всеармейской научно–практической конференции «Инновационная деятельность в Вооружённых силах Российской Федерации» (Санкт-Петербург, 2013), XIII Санкт–Петербургской международной конференции «Региональная информатика – 2012» (Санкт-Петербург, 2012), XVIII международной научно–методической конференции «Современное образование: содержание, технологии, качество» (Санкт-Петербург, 2012), 65–й научно–технической конференции профессорско–преподавательского состава университета СПбГЭТУ «ЛЭТИ» (Санкт-Петербург, 2012), VII Санкт–Петербургской межрегиональной конференции «Информационная безопасность регионов России (ИБРР–2011)» (Санкт-Петербург, 2011).

Полученные результаты диссертационного исследования были использованы при выполнении научно-исследовательских работ по грантам Правительства Санкт-Петербурга, дипломы № ПСП 15 331, № ПСП 14 044, № ПСП 13038. Результаты диссертационного исследования внедрены при выполнении работ по гранту РФФИ (№14-07-00061 А) на базе СПИИРАН, в учебный процесс Санкт-Петербургского государственного электротехнического университета «ЛЭТИ» им. В.И. Ульянова (Ленина) и государственного университета морского и речного флота имени адмирала С.О. Макарова.

Личный вклад. Все научные положения, выносимые на защиту, были получены и сформулированы лично автором.

Публикации. Основные результаты по теме диссертации изложены в 25 печатных изданиях, 6 из которых изданы в журналах, 5 из которых опубликованы в ведущих рецензируемых журналах, входящих в перечень ВАК.

Структура и объем работы. Диссертационная работа изложена на 134 машинописных страницах, включает введение, 5 глав, заключение, список литературы (143 наименования), 35 рисунков и 13 таблиц.

В первой главе рассматриваются современные алгоритмические средства защиты информации в информационно-телекоммуникационных системах. Показано, что в основе таких средств, в основном, лежат вычислительно сложные задачи, такие как задачи факторизации и дискретного логарифмирования. Повысить безопасность использования таких средств можно путём разработки

алгоритмов и протоколов, взлом которых потребует одновременного решения двух независимых вычислительно трудных задач. Рассмотрены все существующие алгоритмы и протоколы такого типа. Показаны их недостатки. Произведена постановка задачи на диссертационное исследование.

Во второй главе рассматривается задача дискретного логарифмирования по трудно факторизуемому модулю в качестве нового примитива для построения алгоритмов и протоколов, взлом которых потребует одновременного решения двух независимых вычислительно трудных задач факторизации и дискретного логарифмирования по простому модулю. Рассмотрены необходимые требования к выбору параметров для указанной задачи и её особенности. Приведены алгоритмы для генерации необходимых параметров.

В третьей главе рассмотрены алгоритмы генерации ключей. Предложены три варианта генерации ключей в зависимости от используемых параметров задачи дискретного логарифмирования по трудно факторизуемому модулю и использования доверительного центра. Разработаны три протокола обмена ключами. Предложен протокол защитного преобразования (шифрования) информации, передаваемой по открытому каналу связи.

В четвёртой главе рассматриваются протоколы электронных цифровых подписей. Предложены протоколы простой, слепой, коллективной, коллективной слепой и утверждаемой групповой электронных цифровых подписей, взлом которых требует одновременного решения двух независимых вычислительно трудных задач факторизации и дискретного логарифмирования по простому модулю.

В пятой главе разрабатываются специальные протоколы защиты информации в информационно-телекоммуникационных системах. С использованием вычислительной сложности решения задачи дискретного логарифмирования по трудно факторизуемому модулю разработаны протоколы удалённой аутентификации, включающие интерактивный и двухшаговые протоколы. Предложен протокол коммутативного шифрования. На его основе построено шифрование по ключу малого размера, взлом которого требует одновременного решения двух независимых вычислительно трудных задач факторизации и дискретного логарифмирования по простому модулю.

Глава 1. Алгоритмические средства защиты информации в информационно-телекоммуникационных системах

Безопасность информационно-телекоммуникационных технологий определяется всеми аспектами, связанными с определением, достижением и поддержанием конфиденциальности, целостности, доступности, неотказуемости, подотчетности, аутентичности и достоверности информационно-телекоммуникационных технологий [2]. Информация, обрабатываемая с использованием информационно-телекоммуникационных технологий, не является жёстко привязанной к физическому носителю, поэтому мощными и универсальными средствами обеспечения конфиденциальности, неотказуемости, аутентичности, достоверности и анонимности являются алгоритмические средства обеспечения безопасности.

1.1 Современные алгоритмы и протоколы аутентификации и защиты информации

Современные алгоритмы и протоколы защиты информации в информационно-телекоммуникационных системах используются для обеспечения конфиденциальности, целостности, неотказуемости, подотчетности, аутентичности, достоверности и анонимности.

Алгоритмы и протоколы аутентификации и защиты информации в информационно-телекоммуникационных системах можно разделить на две группы, в зависимости от используемого типа ключей:

1) обе стороны, участвующие в протоколе, обладают общим ключом, который не знает никто, кроме них самих;
2) у одной стороны есть пара личный ключ, который не знает никто, кроме неё, но с его помощью она может выполнять некоторые алгебраические преобразования над информацией, и открытый ключ, по которому вторая сторона может с достаточно высокой долей вероятности определить кто выполнял эти преобразования.

Ярким примером алгоритма аутентификации с общим ключом является имитовставка, позволяющая проверяющему с определённой вероятностью (например, 2^{-32}) утверждать, что сообщение не было изменено в процессе переда-

чи [7; 8]. Но методы с обладанием общим ключом сталкивается с трудностями передачи и хранения этого ключа от одной стороны к другой, дополнительно, такие методы не могут обеспечить выполнения всех требуемых свойств, например, неотказуемость субъекта от созданного цифрового объекта. Использование пары открытого и секретного ключей позволяет строить такие алгоритмы и протоколы защиты информации, которые используются для обеспечения конфиденциальности, целостности, неотказуемости, подотчетности, аутентичности, достоверности и т.д.

В современном мире ярким примером этому служат системы электронных цифровых подписей, являющихся мощным средством для обеспечения неотказуемости субъекта от созданного им цифрового объекта, аутентичности, целостности и достоверности цифрового объекта, созданного субъектом [9; 10]. Но построение таких алгоритмов и протоколов невозможно без использования односторонних функций с секретом.

1.2 Используемые вычислительно сложные задачи и достигаемый уровень безопасности алгоритмических средств

Вычислительно сложные задачи базируются на предположении существования односторонних функций с секретом. Данный термин был введён в 1976 году Уитфилдом Диффи и Мартином Хеллманом в работе [11]. Они показали, что односторонние функции с секретом могут использоваться для обмена секретными ключами с использованием открытых каналов связи.

Формально односторонняя функция определяется следующим образом [11–13]. Функция f, сопоставляющая набору значений из X набор значений из Y называется односторонней, если легко вычислить f для всех значений $x \in X$, но для заданного значения $y \in Y$ вычислительно трудно найти x такой, что $f(x) = y$. Односторонняя функция с секретом - это односторонная функция $f : X \Rightarrow Y$ с дополнительным свойством, дающим некоторую информацию, с помощью которой становится возможно найти для заданного y, такой $x \in X$, что $f(x) = y$. Они предложили задачу дискретного логарифмирования в качестве односторонней функции с секретом и предположили существование других вычислительно сложных задач пригодных для использования в качестве односторонней функции с секретом.

Однако, как оказалось, найти такие вычислительно сложные задачи довольно нетривиальная задача. На текущий момент существует сравнительно небольшая группа вычислительно трудных задач, пригодных для построения алгоритмических средств защиты информации с открытым ключом [8;9;12;14–16]. Некоторые из предполагаемых вычислительно трудных задач оказались нестойкими, например, задача о рюкзаке. Она заключается в следующем: зная грузы, уложенные в рюкзак, легко подсчитать их общий вес, но, зная вес, вычислительно трудно определить грузы. На её основе Ральфом Мерклем и Мартином Хеллманом в 1978 году в работе [17] была предложена схема шифрования. Однако, спустя 4 года Ади Шамир предложил алгоритм решения данной задачи имеющий полиномиальную вычислительную сложность [18].

Другим примером является скрытая задача поиска сопряжённого элемента [9; 19–21]. Предполагалось, что протоколы, построенные на её основе, имеют стойкость к атакам на основе квантовых вычислений. Но позднее было показано, что данная задача оказалась не сложнее задачи дискретного логарифмирования в мультипликативной группе конечного поля [22; 23].

Таким образом вычислительная сложность таких задач определяется наилучшим алгоритмом её решения. На сегодняшний момент хорошо изученными задачами являются задача дискретного логарифмирования и задача факторизации [8; 9; 12; 14–16].

1.2.1 Задача факторизации

Безопасность многих алгоритмических средств защиты информации основана на решении задачи разложения чисел - факторизации. Например, схема шифрования с открытым ключом RSA, ЭЦП RSA [24], схема шифрования с открытым ключом Рабина [25].

Задача факторизации формулируется следующим образом.

Для заданного положительного целого числа n, нужно найти все его простые делители, $n = p_1^{e_1} p_2^{e_2} ... p_k^{e_k}$, где p_i попарно различные простые делители и $e_i \geq 1$ [12].

Рассмотрим применение ЗФ в схеме шифрования с открытым ключом Рабина.

Пользователь А выбирает два больших простых числа p, q, причём $p \equiv 3 \bmod 4$, $q \equiv 3 \bmod 4$ вычисляет $n = pq$. ОК является n, СК является p, q.

Пусть пользователь В хочет отправить сообщение M пользователю А. Для этого он зашифровывает сообщение $M < n$ путём его возведения в квадрат по модулю ОК пользователя А: $C = M^2 \bmod n$. После чего отправляет криптограмму C пользователю А.

Для расшифрования пользователь А должен извлечь квадратный корень по составному модулю, что можно сделать только зная разложение числа n на простые множители.

Пользователь А вычисляет корни из C по модулям p, q:

$$m_{p_1} = C^{p+1/4} \bmod p,\ m_{q_1} = C^{q+1/4} \bmod q,$$
$$m_{p_2} = p - m_{p_1},\ m_{q_2} = q - m_{q_1}$$

Затем пользователь А восстанавливает сообщение M:

$$M_1 = (m_{p_1}a + mq_1b) \bmod n, M_2 = (m_{p_1}a + mq_2b) \bmod n$$
$$M_3 = (m_{p_2}a + mq_1b) \bmod n, M_4 = (m_{p_2}a + mq_2b) \bmod n,$$

где $a = q(q^{-1} \bmod p)$,$b = p(p^{-1} \bmod q)$. Одним из минусов данной схемы является неоднозначность расшифрования, поэтому к исходному сообщению нужно добавлять некоторую фиксированную информацию, чтобы на приёмной стороне можно было однозначно восстановить сообщение. Без решения ЗФ модуля n извлечь квадратный корень по составному модулю, за исключением переборных алгоритмов, невозможно.

Рассмотрим наиболее известные основные методы решения задачи факторизации. Начнём с методов, обладающих экспоненциальной вычислительной сложностью:

1) перебор возможных делителей $\mathcal{O}\left(\sqrt{n}\log^2 n\right)$ [26];
2) метод факторизации Ферма $\mathcal{O}(n)$ [27];
3) ро-метод Полларда $\mathcal{O}\left(n^{1/4}\right)$ [28];
4) алгоритм Ленстры $\mathcal{O}\left(n^{1/3}\log^2 n\right)$ [26];
5) алгоритм Полларда — Штрассена $\mathcal{O}\left(n^{1/4}\log^4 n\right)$ [29];
6) $p-1$ метод Полларда $\mathcal{O}\left(n^{1/2}\log^c n\right)$, где c - некоторая положительная константа [30];

7) метод Лемана $\mathcal{O}\left(n^{1/3}\right)$ [31].

Однако данные алгоритмы являются сравнительно медленными и не применяются на практике, за исключением $p-1$ метода Полларда, который позволяет быстро находить небольшие простые делители n, не превышающих некоторого определённого значения B.

Для практического применения на классических компьютерах были разработаны субэкспоненциальные алгоритмы, вычислительная сложность которых оценивается L-нотацией [12].

L-нотация - асимптотическая нотация для оценки вычислительной сложности субэкспоненциальных алгоритмов, аналогична $\mathcal{O}$- нотации, записывается $L_n[\alpha, c]$ для некоторого числа n и определяется формулой

$$L_n[\alpha, c] = e^{(c+o(1))(\ln n)^{\alpha}(\ln \ln n)^{1-\alpha}},$$

где c и α некоторые положительные константы, причём $0 \leq \alpha \leq 1$.

Основными алгоритмами факторизации с субэкспоненциальной вычислительной сложностью являются:

1) факторизация методом непрерывных дробей $L_n(1/2, \sqrt{2})$ [32];
2) метод квадратичного решета $L_n(1/2, 1)$ [26];
3) факторизация Ленстры с помощью эллиптических кривых $L_p(1/2, \sqrt{2})$, где p наименьший простой делитель n [33];
4) алгоритм Диксона $L_n(1/2, 2\sqrt{2})$ [34];
5) специальный метод решета числового поля $L_n(1/3, (32/9)^{1/3})$ [26; 35];
6) общий метод решета числового поля $L_n(1/3, (64/9)^{1/3})$ [26; 35].

Самыми быстрыми методами факторизации из всех существующих являются методы решета числового поля [26; 35; 36]. Специальный метод решета числового поля применим только для чисел специального вида $r^e \pm s$, где $r \in \mathbb{N}$, $s \in \mathbb{Z}$ (например, числа Мерсенна). Общий метод решета числового поля является самым быстрым алгоритмом факторизации целых чисел длиной более 110 десятичных знаков. С его помощью удалось успешно факторизовать 768 битный RSA модуль [37], что на сегодняшний день является рекордом среди факторизации чисел.

1.2.2 Задача дискретного логарифмирования

Второй вычислительно сложной задачей, применяемой в алгоритмических средствах защиты информации, например, обмене ключами Диффи-Хеллмана [11], ЭЦП Эль-Гамаля [38], является задача дискретного логарифмирования [12; 13; 15; 16]. Она формулируется следующим образом.

Пусть G - конечная мультипликативная группа порядка γ, элемент α является генератором группы G, и элемент $\beta \in G$. Дискретный логарифм элемента β по основанию элемента α, обозначается как $\log_\alpha \beta$. Задача дискретного логарифмирования состоит в нахождении целого числа x, $1 \leq x \leq \gamma$, где γ порядок группы G, такого, что $\beta = \alpha^x$. Число x является дискретным логарифмом β по основанию α, или другими словами является индексом элемента β в группе G, порождённой элементом α: $x = ind_\alpha \beta$ [9; 12; 13; 39].

Рассмотрим использование ЗДЛ в схеме ЭЦП Эль-Гамаля, ставшую основой для современных стандартов ЭЦП [9; 15; 40].

Пользователь А генерирует большое простое число p, выбирает целое число α, являющееся первообразным корнем по модулю p. Затем генерирует случайное число $1 < x < p$ и вычисляет $y = \alpha^x \bmod p$. ОК является тройка чисел (y, p, α), СК - x.

Пусть пользователь А хочет подписать документ $M < p$. Для этого он выполняет следующие шаги.

1. Сгенерировать k такое, что $0 < k < p - 1$ и НОД$(k,p-1) = 1$.
2. Вычислить $r = \alpha^k \bmod p$.
3. Вычислить $s = (m - xr)k^{-1} \bmod (p - 1)$.

Подписью является пара чисел (r,s).

Пользователь В для проверки ЭЦП использует ОК пользователя А и проверяет выполнимость следующего соотношения $\alpha^M \equiv y^r r^s \bmod p$. Для подделки подписи или нахождения СК пользователя А потребуется решить ЗДЛ.

Вычислительная сложность алгоритмов решения ЗДЛ зависит от строения группы [9; 12; 14]. Алгоритмы, которые не зависят от конкретного вида групповой операции и элементов группы, имеют экспоненциальную вычислительную сложность. Для групп специального вида были разработаны субэкспоненциальные алгоритмы [41; 42].

Рассмотрим основные алгоритмы решения ЗДЛ:

1) переборный метод $\mathcal{O}(\gamma)$, где γ порядок группы G [12];
2) алгоритм Шенкса ("больших и малых шагов") $\mathcal{O}\left(\sqrt{\gamma}\right)$ [9; 12; 43];
3) ро-метод Полларда $\mathcal{O}\left(\sqrt{\gamma}\right)$ [12; 44];
4) алгоритм Полига-Хеллмана $\mathcal{O}\left(\sum_{i=1}^{r} e_i(\ln\gamma + \sqrt{p_i})\right)$, где $\gamma = p_1^{e_1}p_2^{e_2}...p_k^{e_k}$ и $\mathcal{O}\left(\sqrt{\gamma}\right)$ в худшем случае [12; 45];
5) метод вычисления индексов $L_n(1/2,c)$, где c - некоторая константа [12];
6) метод решета числового поля $L_n(1/3,(64/9)^{1/3})$ [12; 14; 46; 47];
7) метод решета функционального поля $L_n(1/3,(32/9)^{1/3})$ [14; 48].

Самым вычислительно эффективным и практически реализуемым алгоритмом для решения ЗДЛ по простому модулю p является алгоритм с использованием метода решета числового поля с оценкой $L_n(1/3,(64/9)^{1/3})$ [49]. Отдельно отметим существование алгоритма факторизации и дискретного логарифмирования Шора для квантового компьютера [50–52], имеющего полиномиальную вычислительную сложность. Но, по мнению большинства исследователей, в ближайшее время квантовый компьютер, с достаточным количеством кубитов, не будет создан [49]. Причём, по последним оценкам [42; 53], представленных в таблице 3, для решения задачи дискретного логарифмирования на эллиптических кривых потребуется значительно меньше кубитов.

Таблица 3 — Количество кубит квантового компьютера для решения задачи факторизации RSA модуля и дискретного логарифмирования на эллиптических кривых для сопоставимых уровней вычислительной сложности

RSA		ЭК	
Размер модуля (бит)	Количество кубит (штук)	Размер группы (бит)	Количество кубит (штук)
1024	2048	163	1000
2048	4096	224	1300
3072	6144	256	1500
4096	8192	383	2300
15360	30720	512	2800

1.3 Понятие интегрального уровня безопасности алгоритмических средств защиты информации

Трудоёмкость вычислительно сложной задачи, используемой в качестве односторонней функции с секретом, определяется количеством операций, необходимых самому лучшему алгоритму для её решения, и измеряется в битах. Например, 80 битовая вычислительная сложность ЗДЛ означает, что для решения это задачи потребуется выполнить 2^{80} операций модульного умножения. Нижний уровень вычислительной сложности определяется экспертами с учётом развития уровня техники и алгоритмов решения вычислительно сложных задач.

На текущий момент известны следующие методики и рекомендации по выбору необходимых длин параметров вычислительно сложных задач:

1) Ленстра/Верхойл 2001 год [54];
2) Ленстра 2004 год [55];
3) RFC3766 2004 год [56];
4) ECRYPT II 2012 год [49];
5) NIST 2012 год [57];
6) ANSSII 2014 год [58];
7) BSI 2015 год [59].

До 2012 года достаточной считалась 80 битовая вычислительная сложность, но с учётом развития техники, снижения её стоимости, оптимизации алгоритмов, на текущий момент по оценкам указанных методик для обеспечения безопасности использования вычислительно сложных задач в алгоритмических средствах защиты информации требуется 128 битовая вычислительная сложность (таблица 4).

Рассмотрим подробнее методику расчёта ECRYPT II [49]. Трудоёмкость самого быстрого алгоритма факторизации и дискретного логарифмирования GNFS (общее решето числового поля) асимтотически оценивается по формуле

$$L_n = e^{(c+o(1))(\ln n)^{1/3}(\ln\ln n)^{2/3}},$$

с константой $c = (64/9)^{1/3}$.

Учитывая уровень техники и теоретические разработки в области специальных микросхем TWIRL [60], SHARK [61] и т.д, ускоряющих определённые

Таблица 4 — Размеры параметров задач факторизации и дискретного логарифмирования для 128 битовой вычислительной сложности решений ЗФ и ЗДЛ

Методика/ рекомендации	Дата (год)	Выч-ная сложность (бит)	Модуль ЗФ (бит)	Размер подгруппы ЗДЛ (бит)	Размер группы ЗДЛ (бит)
Ленстра/Верхойл [54]	2076	115	6790	230	6790
Ленстра [55]	2090	128	4440	256	4440
ECRYPT II [49]	2031-2040	128	3248	256	3248
NIST [57]	>2030	128	3072	256	3072
ANSSII [58]	2021-2030	100	2048	200	2048
RFC3766 [56]	-	128	3253	256	3253
BSI [59]	-	112	2048	224	2048

шаги алгоритма GNFS, результирующая формула оценки необходимых длин параметров для обеспечения вычислительной сложности решения ЗДЛ и ЗФ приобрела следующий вид:

$$L_n = e^{((64/9)^{1/3}+o(1))(\ln n)^{1/3}(\ln\ln n)^{2/3}-14}.$$

Расчёты длин параметров для обеспечения необходимой вычислительной сложности решения ЗФ и ЗДЛ представлены в таблице 5. В таблице 6 приведены оценки вычислительной сложности для используемых на практике размеров модулей ЗФ и ЗДЛ.

Таблица 5 — Соответствие размеров параметров и вычислительной сложности решений ЗФ и ЗДЛ по методике ECRYPT II

Выч-ная сложность (бит)	Размер фактор. числа (бит)	Размер группы в ЗДЛ (бит)	Размер под-группы в ЗДЛ (бит)	Порядок группы точек ЭК (бит)
64	816	816	128	128
80	1248	1248	160	160
112	2432	2432	224	224
128	3248	3248	256	256
160	5312	5312	320	320
192	7936	7936	384	384
256	15424	15424	512	512

Таблица 6 — Вычислительная сложность решения ЗДЛ и ЗФ для используемых на практике размеров модулей в ЗФ и ЗДЛ по методике ECRYPT II

Размер (бит)	Вычислительная сложность (бит)
512	50
768	62
1024	73
1536	89
2048	103

Отметим, что все существующие подходы к оценки безопасности использования алгоритмических средств защиты информации опираются на следующие два факта:

1) решение используемой вычислительно сложной задачи является вычислительно нереализуемым за приемлемое время при использовании наилучшего известного алгоритма её решения, т.е. её вычислительная сложность W достаточна высока;
2) вероятность появления P в ближайшем будущем вычислительно эффективного алгоритма решения указанной задачи пренебрежимо мала.

Но безопасность использования алгоритмических средств защиты информации, в основе которых лежат вычислительно сложные задачи, так же должна учитывать и вероятность появления новых прорывных методов решения вычислительно сложных задач. Поэтому Молдовяном Н.А. был введён интегральный показатель безопасности $l = W/P$ [62]. Следовательно, увеличение вычислительной сложности решения вычислительно сложной задачи или уменьшение вероятности P ведёт к увеличению l.

Поиски вычислительно сложных задач с более высокими оценками вычислительной сложности их решений продолжаются, но на текущий момент, лучшими кандидатами для использования в алгоритмических средствах защиты информации являются ЗФ и ЗДЛ [49]. Существенного увеличения интегрального показателя безопасности l можно достичь путём разработки протоколов, взлом которых требует одновременного решения двух независимых вычислительно сложных задач.

Пусть $W_1 = 2^{128}$, а $P_1 = 2^{-32}$, тогда $l = 2^{128}/2^{-32} = 2^{160}$.

В случае использования двух независимых вычислительно трудных задач (пусть вероятность появления вычислительно эффективного алгоритма решения у обоих используемых задач будет одинакова и равна 10^{-32}) $W_1 = 2^{128}$, $P_1 = 2^{-32}$, $W_2 = 2^{128}$, $P_2 = 2^{-32}$, получим

$l = (W_1 + W_2)/(P_1 P_2) = (2^{128} + 2^{128})/(2^{-32} \cdot 2^{-32}) = 2 \cdot 2^{128}/2^{-64} = 2 \cdot 2^{192}$.

Т.е. в десятичной системе примерно получим оценку 10^{48} и 10^{58} соответственно, интегральный параметр безопасности возрос примерно в 10^{10} раз.

1.4 Алгоритмические средства обеспечения информационной безопасности информационно-телекоммуникационных систем, взлом которых требует решения нескольких вычислительно сложных задач

Первое упоминание о протоколе, взлом которого требует решения двух независимых вычислительно трудных задач, встречается в техническом отчёте Захавы Шмуеля из Израильского технического института [63]. Он предложил использовать большой составной модуль $n = pq$, где p, q некоторые большие простые числа, в схеме обмена ключами Диффи-Хеллмана [11], следующим образом.

Пусть составной модуль $n = pq$, где p, q большие секретные простые числа. Генератор группы α, где $1 < \alpha < n$, используется для генерации общего ключа следующим образом. Каждый пользователь A_i выбирает СК $0 < x_i \leq n$ и использует его для генерации своего ОК $k_i = \alpha^{x_i} (\mathrm{mod}\ n)$. Пусть два пользователя A_i, A_j сгенерировали свои ОК $K_i = \alpha^{x_i} (\mathrm{mod}\ n), K_j = \alpha^{x_j} (\mathrm{mod}\ n)$. Тогда они могут сгенерировать общий СК следующим образом $K_{ij} = \alpha^{x_i x_j} = K_j^{x_i} = K_i^{x_j} (\mathrm{mod}\ n)$. В случае использования составного модуля n оба делителя p, q держатся в секрете обоими пользователями. Нахождение общего СК K_{ij}, по известным параметрам модуля n, K_i, K_j трудно настолько, насколько трудна задача факторизации модуля n. Так же показано, что алгоритм, который сможет взламывать систему Диффи-Хеллмана по составному модулю пригоден для взлома RSA. Взлом такой системы, при знании n, α, K_i или n, α, K_j, потребует решения факторизации n, и нахождения дискретного логарифма по обоим модулям p и q. Таким образом стойкость системы зависит от вычислительной сложности двух вычислительно сложных задач. Такая схема генерации ОК пользователей была встроена в протокол ментального покера, предложенного Шамиром, Ривестом и Адельманом в 1981 году [64].

В 1988 году Кевин Маккёрлей предложил схему распределения ключей [65] с использованием аналогичного трудно факторизуемого модуля $n = pq$, где p, q большие секретные простые числа. Основным отличием стали дополнительные ограничения на множители p, q и генератор группы α. Для вычисления модуля n дополнительно генерировались два случайных числа r и s такие, что $2r+1$ имеет большой простой делитель, $4r + 1$ является простым числом, $8r + 3$ является

простым числом, s содержит большой простой делитель, $4s-1$ является простым числом, $8s-1$ является простым числом. Тогда $p = 8r + 3$, $q = 8s - 1$ и $n = pq$. В качестве генератора группы он предложил использовать $\alpha = 16$.

Таким образом схема обмена ключами выполняется следующим образом.

1. Пусть два пользователя A_i, A_j сгенерировали свои открытые ключи $K_i = 16^{x_i} (\mathrm{mod}\ n), K_j = 16^{x_j} (\mathrm{mod}\ n)$.
2. Тогда они могут сгенерировать общий секретный ключ следующим образом $K_{ij} = 16^{x_i x_j} = K_j^{x_i} = K_i^{x_j} (\mathrm{mod}\ n)$.

Также была предложена схема ЭЦП, аналогичная схеме ЭЦП Эль-Гамаля, для вышеописанной схемы распределения ключей [38]. Пусть владелец ОК $y = 16^x \bmod n$, где x - СК, хочет подписать сообщение $M \leq n$. Для этого ему необходимо выполнить следующие шаги.

1. Сгенерировать случайное число k.
2. Вычислить $u = 16^k \bmod n$.
3. Вычислить $t = My^k \bmod n$.

Подписью являются пара чисел (u, t). Для проверки ЭЦП используется следующее соотношение $M \equiv t(u^x)^{-1} \bmod n$.

В 1990 году на конференции EUROCRYPT Марк Жиральт [66] предложил использовать составной модуль $n = pq = (2fp' + 1)(2fq' + 1)$, где f, p', q' - простые числа, в схеме аутентификации Шнорра [67], следующим образом.

Пусть b целое число порядка f по обоим множителям p и q, т.е. $b^f = 1 \bmod n$. Значения n, f, b - несекретные, p и q держатся в секрете. Пользователь А вычисляет свой ОК по формуле $P = b^{-s} \bmod n$, где s - СК. Тогда схема аутентификации пользователя А выглядит следующим образом.

1. Пользователь А выбирает случайное натуральное число $r < f - 1$, вычисляет $x = b^r \bmod n$ и посылает его пользователю Б вместе со своим ОК P.
2. Пользователь В выбирает случайное число c из интервала $[0,2^t - 1]$ и посылает его пользователю А.
3. Пользователь А вычисляет $y = r + sc \bmod f$ и посылает его пользователю Б.
4. Пользователь В проверяет $b^y P^c = x \bmod n$.

Также предлагается использовать параметры следующей длины $|f| = 200$ бит, $|p'| = |q'| = 300$ бит, $|n| = 1000$ бит, для обеспечение 70 битовой стойкости.

На этой же конференции Кевин Маккёрлей и Эрнест Бриккель предложили свой вариант схемы аутентификации, основанной на ЗФ и ЗДЛ [68]. Основным отличием от оригинальной схемы Шнора стало вычисление $y = r+sc \bmod (p-1)$ по модулю $p-1$. Параметры схемы генерируются со следующими ограничениями: $qw|p-1, q^2 \not| p-1, q, > 512, b^q = 1 \bmod p$.

Это позволило построить схему аутентификации таким образом, что для её взлома потребуется решить ЗФ модуля $p-1$, содержащего два больших простых делителя $q,\ w$ и ЗДЛ по модулю p.

В 1994 году Л. Харн предложил свой вариант обмена ключей и ЭЦП [69]. Каждый пользователь выбирает большие простые числа $p = 2p'q' + 1$, где $p' = 2p''+1$, $q' = 2q''+1$ и p', q', p'', q'' - случайные большие простые числа, и случайно выбирает примитивный элемент α и $x \in [1, p-1]$. Затем вычисляет d, такое что $3d \bmod \phi(\phi(p)) = 1$ и $y = \alpha^x \bmod p$. Значения $p, \alpha, y, 3$ - ОК, (p', q', x', d) - СК. Пусть пользователь А хочет вычислить общий СК с пользователем В. Для этого необходимо выполнить следующие шаги.

1. Пользователь А берёт ОК пользователя В (p_B, α_B, y_B). Генерирует СК $k \in [1,p_B - 1]$. Затем, вычисляет $K_{AB} = y_B^k \bmod p_B$, $z_A = \alpha_B^k \bmod p_B$, $C = z_A^3 \bmod (p_B - 1)$ и отправляет пользователю В значение C.
2. Пользователь В $z_A = C^{d_B} \bmod (p_B - 1)$, $K_{AB} = z_A^{x_B} \bmod p_B$.

Для взлома данной схемы потребуется решить ЗФ числа $p_B - 1$ и ЗДЛ по модулю p_B. Так же в работе предложены схемы открытого шифрования и ЭЦП. В основу ЭЦП положена схема ЭЦП Эль-Гамаля, первый параметр которой вычисляется по модулю p, а второй - q, и дополнительно изменены некоторые параметры генерации и проверки подписи. Однако, в 1996 году Ли показал, что с высокой вероятностью можно выполнить подделку подписи, если решить ЗДЛ, и предложил улучшенный вариант ЭЦП, лишённый указанных недостатков [70].

В 1994 году Хи и Кейслер [71] предложили улучшенный вариант схемы ЭЦП Эль-Гамаля, использующий похожие идеи Л.Харна по использованию разных модулей для первого и второго параметров подписи, и модуля $p = 2rp'q'+1$, где r - некоторое случайное число. Ещё одним отличием стало использование

в процессе вычислений сложность задачи нахождения квадратного корня по составному модулю $p-1$.

Пусть p большое простое число, такое что $p = 2rp'q' + 1$, где r - некоторое случайное число, а p', q' большие простые числа, $n = p'q'$. Пусть СК пользователя - x, $1 < x < n$, ОК - $y = \alpha^x \bmod p$ и $z = y^x = \alpha^{x^2} (\mathrm{mod} p)$. Тогда для подписи сообщения M пользователь выполняет следующие шаги.

1. Генерирует случайное натуральное число $t < n$, НОД(t,$p-1$) $= 1$.
2. Вычисляет $k = t^2 \bmod (p-1)$.
3. Вычисляет $r = \alpha^k \bmod (p-1)$.
4. Вычисляет $s = (m - xr)t^{-1} \bmod (p-1)$.

Подписью к сообщению M является пара чисел (r,s).

Для проверки подписи используется соотношение $\alpha^{m^2} z^{r^2} = y^{2mr} r^{s^2} \bmod p$.

Однако, Харн [72], Ли [73] и Тиерсма [74] показали, что для взлома схемы Хи и Кейслера достаточно решить ЗДЛ по модулю p и решение ЗФ $p-1$ не потребуется. В 2003 году Ли [75] предложил свой вариант схемы ЭЦП Хи-Кейслера, но Шимин Вей [76] показал, что подделать подпись можно без решения каких либо вычислительно трудных задач.

В 1997 году Лаих и Куо [77] предложили две схемы ЭЦП, взлом которых потребует решения ЗФ и ЗДЛ. В них использовался модуль $p = 2rp'q' + 1$, где r - некоторое случайное число. Основным отличием стало использование $2t + 1$ пар ОК, СК. Позже, Ли [78] предложил модифицированную версию одного из алгоритмов ЭЦП, требующую только две пары ОК, СК, но в 2005 году Куан [79] показал, что модифицированная схема ЭЦП нестойкая, и подделать подпись можно без решения каких либо вычислительно сложных задач.

В 1998 году Зухуа Шао предложил две схемы ЭЦП [80], взлом которых требует одновременного решения ЗФ и ЗДЛ. В предложенных схемах использовался модуль $p = 4p_1q_1 + 1$, где $p_1 = 2p_2 + 1$, $q_1 = 2q_2 + 1$ и p_1, p_2, q_1, q_2 - большие простые числа, и вычислительную сложность нахождения квадратных корней по составному модулю. Указанные параметры модуля p генерировались доверительным центром и никогда не использовались пользователями системы. Доверительный центр выбирает элемент α порядка p_1q_1. Пользователь генерирует свой СК x, $1 < x < p_1q_1/2$ и вычисляет свой открытый ключ $y = \alpha^{x^2 + x^{-2}} \bmod p$. Чтобы подписать сообщение M пользователь выполняет следующие шаги.

1. Генерирует случайное t, $1 < t < p_1 q_1/2$.
2. Вычисляет $r = \alpha^{t^2+t^{-2}}$.
3. Решает систему сравнений для нечётного k
$$\begin{cases} xs - kt^{-1} \equiv Mt - x^{-1}r \bmod (p_1 q_1) \\ x^{-1}s - kt \equiv Mt^{-1} - xr \bmod (p_1 q_1) \end{cases}$$
и находит s, k.

Подписью к сообщению M является тройка чисел (k, r, s). Для проверки подписи к сообщению M используется следующие соотношение $y^{s^2+r^2} = r^{m^2+k^2}\alpha^{4(mk-sr)}$. Вторая схема ЭЦП идентична первой, за исключением системы сравнений и проверочного соотношения.

Однако, Лай [81] и Ли [82] показали, что стойкость схем ЭЦП Шао меньше предполагаемой, в частности, для её взлома достаточно решить только ЗФ. Позднее, Хи [83], Хванг [84], Ванг [85], Тзенг [86], предложили различные модификации схемы Шао. Но Шао [87], Тзункхер [88], показали, что безопасность этих схем основана только на задаче дискретного логарифмирования. Так же в работе Пона [89] были предприняты попытки создать сразу восемь безопасных модификаций схем ЭЦП Шао, однако Шао [90] и Куан [91] показали, что для взлома всех восьми модификаций его схем ЭЦП достаточно решить ЗДЛ. Кроме этого, атакующий может легко подделать подпись большинства модифицированных схем, без решения ЗДЛ и ЗФ.

В 2000 году Давид Поитчеваль [92] предложил схему аутентификации и ЭЦП, взлом которых потребует решения ЗДЛ и ЗФ. Обе схемы использовали составной модуль $n = pq$, где p,q - большие простые числа. Схема аутентификации выглядела следующим образом.

1. Доказывающий генерировал СК s, вычислял ОК $v = \alpha^{-s} \bmod n$. Генерировал случайное натуральное значение r, вычислял $x = \alpha^r \bmod n$, вычислял хэш $h = H(x)$, и отправлял значение h проверяющему.
2. Проверяющий выбирал случайный запрос e и высылал его доказывающему.
3. Доказывающий вычислял $y = r + es$ и отправлял проверяющему значение y.
4. Проверяющий проверял выполнимость соотношения $h \equiv H(\alpha^y v^e \bmod n)$.

Схема ЭЦП была получена выводом из вышеописанной схемы аутентификации. В работе указано, что нижняя граница стойкость предложенных схем определяется стойкостью ЗФ.

В 2005 году Гортицкая и Молдовян [93–95] предложили новый механизм генерации параметров подписи r, s в ЭЦП Эль-Гамаля. Параметры r, s представлялись в одинаковом виде $r = \alpha^k \bmod p$ и $s = \alpha^g \bmod p$, где значения k, g вычислялись из некоторой системы сравнений. Идея заключалась в том, чтобы сделать невозможным вычисление одного параметра, по известному второму. Для этого параметры r, s были аргументами некоторых заданных функций $F_1(r, s)$ и $F_2(r, s)$. При этом значение функции F_1 изменялось, а значение функции $F_2 = Z = const$ оставалось неизменным. В качестве функций F_1, F_2 предлагалось использовать такие функции как $rs \bmod p, r/s \bmod p$ и т.д. Условие постоянства записывалось, например, в виде $k - g \equiv U \mod \gamma$. Соотношение для проверки подписи имело, например, следующий вид $r/s = y^{(rs \bmod p)^H} \alpha^{rs \bmod p} \bmod p$. Такой метод, в дополнение к существующим, позволил сформировать достаточно большое количество протоколов с использованием вычислительной трудности ЗФ, ЗДЛ, так и протоколов, взлом которых требует одновременного решения ЗФ и ЗДЛ, с использованием как составного модуля $n = pq$, где p,q - большие простые числа, так и модуля вида $n = rpq - 1$, где p,q - большие простые числа, а r - некоторое чётное число [96–104].

В 2007 году Вей [105] предложил две схемы ЭЦП, взлом которых потребует решения ЗДЛ и ЗФ. Обе схемы использовали модуль $p = 4p_1q_1 + 1$, и вычислительную сложность нахождения квадратных корней по составному модулю, как в схеме ЭЦП Шао. Годом позже в работе Зенга [106] было показано, что возможна подделка ЭЦП без знания секретного ключа.

Попытки улучшения схемы Шао продолжились [107–113].

В 2010 году Аль-Файоми [114] предложил свой вариант ЭЦП, взлом которой потребует решения ЗДЛ и ЗФ. Использовался составной модуль $n = pq$, где p, q - большие простые числа. Генерация ключей происходила следующим образом.

1. Вычислить составной модуль $n = pq$, где p, q - большие простые числа.
2. Вычислить свой СК $\omega = (p - 1)/2$.
3. Сгенерировать случайное натуральное число $b < n$ порядка $\phi(n)$.

4. Вычислить $y = b^{\phi(n)/\omega} \bmod n$.

ОК пользователя являлась пара чисел (n, y), СК - ω.

Для подписи сообщения M использовался следующий алгоритм.

1. Вычислить $x = m^{-1} \bmod \omega$.
2. Вычислить подпись $s = y^x \bmod n$.

Для проверки подписи к сообщению M проверялось выполнимость соотношения $s^M \bmod n = y$.

Но из-за неправильного выбора y - порождающего элемента группы, данная схема оказалась основана только на вычислительной сложности ЗДЛ. Для решения ЗФ достаточно вычислить НОД от модуля n и $y^2 - 1$, значение которого равно наименьшему делителю модуля n.

В 2011 году Молдовян предложил использование ЗДЛ по двойному основанию с использованием составного значения модуля. На основе данного метода были предложены схемы ЭЦП [115], слепой ЭЦП [116], коллективной ЭЦП [117]. Составной модуль n использует следующую структуру $n = pq$, где $p = N_p r^2 + 1$, $q = N_q r^2 + 1$, а N_p, N_q - некоторые большие чётные числа. Открытый ключ генерируется по уравнению $y = \alpha^x \beta^w \bmod n$, а секретным ключом является четвёрка (p, q, x, w). Процедура генерации ЭЦП к документу M состоит из следующих шагов.

1. Сгенерировать случайные значения $k < r$ и $t < r$ и вычислить $R = \alpha^k \beta^t \bmod n$.
2. Вычислить хэш от документа M и разбить его на два значения H_1 и H_2.
3. Первый элемент подписи $E = \mathsf{F}(M||R)$, где F - некоторая стойкая хеш функция размера r.
4. Второй элемент подписи $S = \frac{k+xE}{H_1} \bmod r$.
5. Третий элемент подписи $U = \frac{t+wE}{H_2} \bmod r$.

Подписью являлась тройка чисел (E, S, U). Для проверки подписи к сообщению M использовался следующий алгоритм.

1. Вычислить $\tilde{E} = \mathsf{F}(M||R)$.
2. Вычислить $\tilde{R} = y^{-E} \alpha^{SH_1} \beta^{UH_2} \bmod n$.
3. Если $\tilde{E} = E$, то подпись верна, иначе подпись отвергается.

В 2012 году Вишхой и Шривастава [118] предложили свой вариант ЭЦП, взлом которой потребует решения ЗДЛ и ЗФ. Он использовал сразу два модуля

p, где p простое число, и $n = p'q'$, где p', q' - большие простые числа. Параметры схемы вычислялись по разным модулям. Годом позже Чиой и Хи [119] показали, что для взлома схемы Вишхоя и Шриваставы не потребуется решать какую либо вычислительно трудную задачу.

В 2012 году Вижай [120; 121] предложил варианты схемы ЭЦП RSA с использованием дополнительной вычислительно сложной задачи извлечения корней. Похожие идеи использовали в своей схеме шифрования с открытым ключом Цисс и Юссеф [122].

Исследования схем, взлом которых потребует решения двух независимых вычислительно трудных задач, продолжаются [123–125]. Но на текущий момент не известен универсальный метод, позволяющий строить протоколы, взлом которых требует одновременного решения двух независимых вычислительно сложных задач, различных типов.

1.5 Постановка задачи

Анализ литературных источников показал, что для построения алгоритмических средств защиты информации в основном используются две вычислительно сложные задачи: факторизации и дискретного логарифмирования. Все подходы к оценке стойкости базируются на оценке трудности решения вычислительно трудной задачи самым эффективным алгоритмом. Однако, при оценке стойкости необходимо учитывать вероятность появления прорывного алгоритма решения используемой вычислительно трудной задачи. Для этого вводится интегральный параметр безопасности, который учитывает:

1) вычислительную сложность решения вычислительно сложной задачи;
2) вероятность появления в ближайшем будущем вычислительно эффективного алгоритма решения указанной задачи.

Его повышение может быть достигнуто использованием вычислительно сложных задач с более высокими оценками вычислительной сложности, но на текущий момент наилучшими кандидатами для использования в алгоритмических средствах защиты информации являются ЗФ и ЗДЛ. Существенного повышения интегрального показателя безопасности можно достичь путём разработки алгоритмических средств защиты информации, взлом которых требует решения двух независимых вычислительно трудных задач.

Существующие методы к построению алгоритмических средств защиты информации, взлом которых требует решения двух независимых вычислительно трудных задач, можно разделить на две основных группы:

1) использующие трудно факторизуемый модуль;
2) использующие простой модуль специальной структуры, функция Эйлера от которого содержит составное число трудное для факторизации.

Первая группа методов характеризуется малым количеством работ, однако может быть использована для построения всех типов протоколов и является перспективной для дальнейшей проработки.

Вторая группа методов характеризуется большим количеством работ, но не является универсальной, т.е. не очевидно как применить их для построения основных типов протоколов без значительных модификаций. В основном применяются для построения схем ЭЦП, однако уравнения для генерации и проверки ЭЦП являются громоздкими, и как показала практика, они сравнительно часто содержат ошибки из-за затруднительного анализа стойкости таких схем.

В связи с этим представляется актуальным разработка нового универсального метода к построению алгоритмических средств защиты информации, взлом которых требует решения двух независимых вычислительно трудных задач, являющимся свободным от указанных недостатков. Для этого были сформулированы и решены следующие частные задачи:

1. Разработка метода построения алгоритмов и протоколов для применения в средствах защиты информации, обладающих повышенным уровнем безопасности, который позволит расширить виды алгоритмов и протоколов указанного типа.
2. Разработка протоколов локальной и удалённой аутентификации пользователей, субъектов и объектов информационных процессов, обладающих повышенным уровнем безопасности.
3. Разработка протоколов обеспечения конфиденциальности информации, передаваемой по открытым каналам связи, обладающих повышенным уровнем безопасности.
4. Разработка протоколов обеспечения анонимности в открытых компьютерных сетях, обладающих повышенным уровнем безопасности.

1.6 Выводы к первой главе

1. Анализ литературных источников показал, что для построения алгоритмических средств защиты информации в основном используются две вычислительно сложные задачи: факторизации и дискретного логарифмирования.
2. Существующие подходы к оценке стойкости алгоритмических средств защиты информации базируются на учёте трудности решения вычислительно трудной задачи самым эффективным алгоритмом.
3. Существующие методы к построению алгоритмических средств защиты информации, взлом которых требует решения двух независимых вычислительно трудных задач факторизации и дискретного логарифмирования, обладают существенными недостатками, такими как отсутствие универсальности и позволяют строить в основном протоколы ЭЦП.
4. Анализ стойкости большинства существующих алгоритмов и протоколов, взлом которых требует решения двух независимых вычислительно трудных задач факторизации и дискретного логарифмирования, затруднителен из-за громоздкости уравнений проверки и генерации подписи, из-за чего около половины из них оказались нестойкими.

Глава 2. Новый метод построения алгоритмических средств защиты информации в информационно-телекоммуникационных системах, обладающих повышенным уровнем безопасности

2.1 Обоснование нового метода

Рассмотрим ЗДЛ по трудно факторизуемому модулю $n = pq$, являющимся произведением двух больших простых чисел p, q. ЗДЛ по трудно факторизуемому модулю определяется относительно некоторых известных чисел n, $\alpha < n$ (НОД(α,n) $= 1$), y, x и состоит в нахождении такого x, что выполняется следующее выражение

$$y = \alpha^x \bmod n.$$

Предлагаемый метод основан на следующих фактах [12].

1. Решение ЗДЛ в $\mathbb{Z}_n^*$ может быть сведено к комбинации решения двух задач: ЗФ числа n и ЗДЛ по модулю каждого простого делителя p, q.

$$y = \alpha^x \bmod n \Rightarrow \begin{cases} y = \alpha^x \bmod p \\ y = \alpha^x \bmod q \end{cases}$$

2. Вычислительная сложность решения ЗДЛ в $\mathbb{Z}_n^*$ не ниже чем вычислительная сложность решения ЗФ числа n.

Вычислительная сложность решения ЗФ числа $n = pq$, определяется размером меньшего делителя, например $p > q$. Вычислительная сложность решения ЗДЛ по простому модулю p примерно равна вычислительной сложности решения ЗФ числа n, если $|p| = 2|q|$, то есть вычислительная сложность решения ЗДЛ по трудно факторизуемому модулю n, вычислительная сложность решения ЗДЛ по простому модулю p и вычислительная сложность решения ЗФ n являются значениями одного порядка. Если разрабатывать протоколы таким образом, чтобы вычислительная сложность решения обеих задач будет не ниже некоторого заданного уровня вычислительной сложности, то в случае появления прорывного алгоритма решения для одной из рассматриваемых задач, стойкость таких протоколов не изменится (рисунок 2.1).

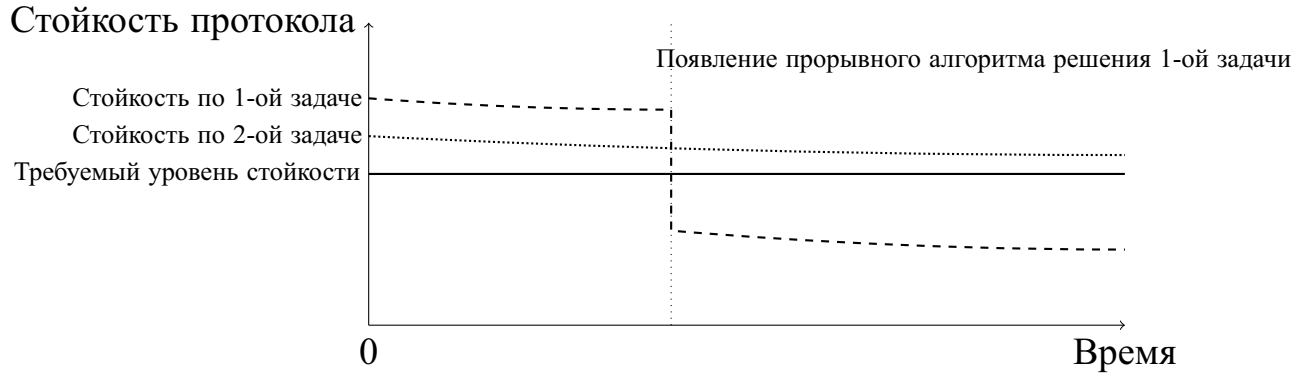

Рисунок 2.1 — Стойкость протоколов, взлом которых требует одновременного решения двух вычислительно сложных задач, в случае появления прорывного алгоритма решения первой задачи

Поскольку вероятность появления такого алгоритма достаточна мала, то итоговая вероятность P в интегральном параметре безопасности l значительно снижается.

Например, при оценке вероятности появления прорывного алгоритма решения в 10^{-6}, и уровне стойкости в 10^{38} (соответствует 128 битовой стойкости) получим.

1. Для протоколов, взлом которого требует решения одной вычислительно трудной задачи, $l = 10^{38}/10^{-6} = 10^{44}$.
2. Для протоколов, взлом которого требует одновременного решения двух независимых вычислительно трудных задач (пусть вероятность появления вычислительно эффективного алгоритма решения у обоих используемых задач будет одинакова и равна 10^{-6}), $l = (10^{38} + 10^{38})/(10^{-6} \cdot 10^{-6}) = 2 \cdot 10^{38}/10^{-12} = 2 \cdot 10^{50}$.

Таким образом достигается увеличение величины интегрального параметра безопасности в 10^6 раз.

Следовательно, построение протоколов, основанных на ЗДЛ по трудно факторизуемому модулю n, по аналогии с известными протоколами, основанными на ЗДЛ по простому модулю, и разработка новых - представляет собой интересный и достаточно общий метод для создания протоколов, взлом которых потребует одновременное решение ЗФ и ЗДЛ по простому модулю. Применяя этот метод можно расширить количество и типы указанных алгоритмов и протоколов.

2.2 Ограничения на выбор параметров, используемых в новом методе

В случае использования составного модуля $n = pq$, являющимся произведением двух больших простых чисел p, q, обратим внимание на порядок генератора мультипликативной группы α.

Порядком числа α *называется наименьшее из чисел* γ*, для которого выполняется сравнение* $\alpha^{\gamma} = 1 \bmod n$ [9].

Как показано в работах [94; 96] не любые значения порядка группы подходят для использования в ЗДЛ по трудно факторизуемому модулю. Рассмотрим возможные варианты значений порядка γ:

1) простой порядок γ такой, что $\gamma \nmid (p-1)$, $\gamma \mid (q-1)$;
2) простой порядок γ такой, что $\gamma \mid (p-1)$, $\gamma \nmid (q-1)$;
3) простой порядок γ такой, что $\gamma \mid (p-1)$, $\gamma \mid (q-1)$, $\gamma^2 \nmid (p-1)$ и $\gamma^2 \nmid (q-1)$;
4) составной порядок $\gamma = \gamma'\gamma''$ такой, что $\gamma' \mid (p-1)$, $\gamma'' \mid (q-1)$, $\gamma' \nmid (q-1)$ и $\gamma'' \nmid (p-1)$.

Пусть p, q - большие простые числа, $n = pq$, γ - простой порядок порождающего элемента группы такой, что $\gamma \nmid (p-1)$ и $\gamma \mid (q-1)$, α - порождающий элемент группы, т.е. $\alpha^{\gamma} = 1 \bmod n$. То возможна быстрая факторизация модуля n с использованием алгоритма Евклида НОД$(\alpha - 1, n) = p$ (см. теорему 2.1).

Теорема 2.1. *Пусть* p*,* q *- большие простые числа,* $n = pq$*,* γ *- простой порядок порождающего элемента группы такой, что* $\gamma \nmid (p-1)$ *и* $\gamma \mid (q-1)$*,* α *- порождающий элемент мультипликативной группы порядка* γ*, т.е.* $\alpha^{\gamma} = 1 \bmod n$*. Тогда НОД*$(\alpha - 1, n) = p$*.*

Доказательство. По теореме Эйлера

$\forall\, \alpha \in \mathbb{Z}_n^*$: НОД$(\alpha, n) = 1 \;\; \exists\, \gamma : \alpha^{\gamma} = 1 \bmod n, \gamma \mid \varphi(n)$,

где $\varphi(n)$ - функция Эйлера от числа n, равная $\varphi(n) = (p-1)(q-1)$.

$\forall\, \beta$:НОД$(\beta, n) = 1 \Rightarrow \beta^{\varphi(n)} = 1 \bmod n$

$\beta^{\varphi(n)} = \alpha^{\gamma} = 1 \bmod n \Rightarrow$

$\forall\, \beta\ \exists\, \alpha' : \ \beta^{\frac{\varphi(n)}{\gamma}} = \alpha' \neq 1 \bmod n$

$\forall \alpha\ \exists\, i \in \mathbb{Z}_n : \ \alpha = (\alpha')^i = (\beta^i)^{\frac{\varphi(n)}{\gamma}} \neq 1 \Rightarrow$

$\Rightarrow \alpha = \beta^{\frac{\varphi(n)}{\gamma}} \bmod n = \beta^{\frac{(p-1)(q-1)}{\gamma}} \bmod n \Rightarrow$

$$\Rightarrow \left| \begin{array}{c} \gamma \nmid (p-1) \\ \gamma \mid (q-1) \\ \beta^{\frac{(p-1)(q-1)}{\gamma}} \bmod n \end{array} \right| \Rightarrow (\beta^{(p-1)})^{\frac{(q-1)}{\gamma}} \bmod n \Rightarrow$$

$$\Rightarrow \left| \begin{array}{c} \alpha = (\beta^{(p-1)})^{\frac{(q-1)}{\gamma}} \bmod p \\ \beta^{(p-1)} = 1 \bmod p \end{array} \right| \Rightarrow \alpha = 1^{\frac{(q-1)}{\gamma}} \bmod p \Rightarrow$$

$\Rightarrow \alpha = 1 \bmod p \Rightarrow$

$\Rightarrow \alpha - 1 = 0 \bmod p \Rightarrow p|(\alpha - 1) \Rightarrow$

НОД$(\alpha - 1, n) = p$. □

Аналогичное доказательство для случая когда $\gamma \mid (p-1)$ и $\gamma \nmid (q-1)$.

Пусть p, q - большие простые числа, $n = pq$, γ - простой порядок порождающего элемента группы такой, что $\gamma \mid (p-1)$, $\gamma \mid (q-1)$, $\gamma^2 \nmid (p-1)$ и $\gamma^2 \nmid (q-1)$, α - порождающий элемент группы, т.е. $\alpha^\gamma = 1 \bmod n$. Тогда $\alpha \neq 1 \bmod p$ и $\alpha \neq 1 \bmod q \Rightarrow$ НОД$(\alpha - 1, n) = 1$ (см. теорему 2.2).

Теорема 2.2. *Пусть p, q - большие простые числа, $n = pq$, γ - простой порядок порождающего элемента группы такой, что $\gamma \mid (p-1)$, $\gamma \mid (q-1)$, $\gamma^2 \nmid (p-1)$ и $\gamma^2 \nmid (q-1)$, α - порождающий элемент мультипликативной группы порядка γ, т.е. $\alpha^\gamma = 1 \bmod n$. Тогда НОД$(\alpha - 1, n) = 1$.*

Доказательство. По теореме Эйлера

$\forall\, \alpha \in \mathbb{Z}_n^* :$ НОД$(\alpha, n) = 1 \;\; \exists\, \gamma : \alpha^\gamma = 1 \bmod n, \gamma \mid \varphi(n)$,

где $\varphi(n)$ - функция Эйлера от числа n, равная $\varphi(n) = (p-1)(q-1)$.

$$\left| \begin{array}{c} \gamma \mid (p-1) \\ \gamma \mid (q-1) \\ \gamma \mid \varphi(n) \end{array} \right| \Rightarrow \forall\, \beta :\text{НОД}(\beta, n) = 1 \Rightarrow \beta^{L(n)} = \beta^\gamma = 1 \bmod n$$

где $L(n)$ - обобщённая функция Эйлера от числа n, равная

$L(n) =$НОК$(p-1)(q-1)$.

$$\left| \begin{array}{c} \gamma \mid (p-1) \\ \gamma \mid (q-1) \end{array} \right| \Rightarrow \gamma^2 \mid \varphi(n)$$

$\beta^{\varphi(n)} = \beta^{L(n)} = \beta^\gamma = \alpha^\gamma = 1 \bmod n$

$\exists\, \alpha', \beta' :\;\; \alpha' = \beta'^{\frac{(p-1)(q-1)}{\gamma^2}} \bmod n \neq 1,\;\; \alpha'^\gamma = 1 \bmod n$

$\forall \alpha\ \exists\ i \in \mathbb{Z}_n :\ \alpha = (\alpha')^i \bmod n \Rightarrow$
$\Rightarrow \alpha = (\beta')^{i\frac{(p-1)(q-1)}{\gamma^2}} \bmod n = \beta^{\frac{(p-1)(q-1)}{\gamma^2}} \bmod n \Rightarrow$
$\Rightarrow \left| \begin{array}{l} \gamma \not| \frac{(p-1)}{\gamma}\frac{(q-1)}{\gamma} \\ \gamma^2 \not| (p-1) \\ \gamma^2 \not| (q-1) \end{array} \right| \Rightarrow \alpha = \beta^{\frac{(p-1)(q-1)}{\gamma^2}} = \beta^{\frac{L(n)}{\gamma^2}} \neq 1 \bmod n \Rightarrow$
$\Rightarrow \left| \begin{array}{l} \alpha \neq 1 \bmod p \\ \alpha \neq 1 \bmod q \end{array} \right| \Rightarrow$
$\Rightarrow$ НОД$(\alpha - 1,n) = 1.$ □

Пусть p, q - большие простые числа, $n = pq$, $\gamma = \gamma'\gamma''$ - составной порядок порождающего элемента группы такой (γ' и γ'' простые числа), что $\gamma' \mid (p-1)$, $\gamma'' \mid (q-1)$, $\gamma' \not| (q-1)$ и $\gamma'' \not| (p-1)$, α - порождающий элемент группы, т.е. $\alpha^\gamma = 1 \bmod n$. Тогда $\alpha \neq 1 \bmod p$ и $\alpha \neq 1 \bmod q \Rightarrow$ НОД$(\alpha - 1,n) = 1$ (см. теорему 2.3).

Теорема 2.3. *Пусть p, q - большие простые числа, $n = pq$, $\gamma = \gamma'\gamma''$ - составной порядок порождающего элемента группы (γ' и γ'' простые числа) такой, что $\gamma' \mid (p-1)$, $\gamma'' \mid (q-1)$, $\gamma' \not| (q-1)$ и $\gamma'' \not| (p-1)$, α - порождающий элемент мультипликативной группы порядка γ, т.е. $\alpha^\gamma = 1 \bmod n$.*
Тогда НОД$(\alpha - 1,n) = 1$.

Доказательство. По теореме Эйлера
$\forall\ \alpha \in \mathbb{Z}_n^* :$ НОД$(\alpha,n) = 1\ \ \exists\ \gamma : \alpha^\gamma = 1 \bmod n, \gamma \mid \varphi(n)$,
где $\varphi(n)$ - функция Эйлера от числа n, равная $\varphi(n) = (p-1)(q-1)$.
$\left| \begin{array}{c} \gamma' \mid (p-1) \\ \gamma'' \mid (q-1) \\ \gamma = \gamma'\gamma'' \mid \varphi(n) \end{array} \right| \Rightarrow \forall\ \beta :$НОД$(\beta,n) = 1 \Rightarrow \beta^{L(n)} = \beta^\gamma = 1 \bmod n$
где $L(n)$ - обобщённая функция Эйлера от числа n, равная
$L(n) =$НОК$(p-1)(q-1)$.
$\beta^{\varphi(n)} = \beta^{L(n)} = \beta^\gamma = \alpha^\gamma = 1 \bmod n$
$\exists\ \alpha', \beta' :\ \alpha' = \beta'^{\frac{(p-1)(q-1)}{\gamma}} \bmod n \neq 1,\ \alpha'^\gamma = 1 \bmod n$
$\forall \alpha\ \exists\ i \in \mathbb{Z}_n :\ \alpha = (\alpha')^i \bmod n \Rightarrow$
$\Rightarrow \alpha = (\beta')^{i\frac{(p-1)(q-1)}{\gamma}} \bmod n = \beta^{\frac{(p-1)(q-1)}{\gamma}} \bmod n \Rightarrow$

$$\Rightarrow \left| \begin{array}{l} \gamma \not| \frac{(p-1)}{\gamma'} \frac{(q-1)}{\gamma''} \\ \gamma'' \not| (p-1) \\ \gamma' \not| (q-1) \end{array} \right| \Rightarrow \alpha = \beta^{\frac{(p-1)(q-1)}{\gamma}} = \beta^{\frac{L(n)}{\gamma}} \neq 1 \bmod n \Rightarrow$$

$$\Rightarrow \left| \begin{array}{l} \alpha \neq 1 \bmod p \\ \alpha \neq 1 \bmod q \end{array} \right| \Rightarrow$$

$$\Rightarrow \text{НОД}(\alpha - 1, n) = 1. \qquad \square$$

Таким образом в ЗДЛ по трудно факторизуемому модулю в качестве порядка порождающего элемента группы, стойкого к быстрой факторизации с использованием алгоритма Евклида, возможно применение:

1) простого порядка γ такого, что $\gamma \mid (p-1)$, $\gamma \mid (q-1)$, $\gamma^2 \not| (p-1)$ и $\gamma^2 \not| (q-1)$;
2) составного порядка $\gamma = \gamma'\gamma''$ такого, что $\gamma' \mid (p-1)$, $\gamma'' \mid (q-1)$, $\gamma' \not| (q-1)$ и $\gamma'' \not| (p-1)$.

Однако в случае использования простого порядка необходимо учитывать следующую особенность. Значение $n-1$, как правило, является легко факторизуемым, $n-1 = (u\gamma+1)(v\gamma+1)-1 = uv\gamma^2 + u\gamma + v\gamma = \gamma(uv\gamma + v + u)$, где u и v некоторые множители.

Из данного соотношения можно легко вычислить значение γ. Следовательно, при применении в ЗДЛ по трудно факторизуемому модулю n значение γ должно быть несекретным.

Для второго случая данная особенность является неактуальной, т.к. $n-1 = (u\gamma'+1)(v\gamma''+1)-1 = u\gamma'v\gamma'' + v\gamma'' + u\gamma'$. Причём значение γ должно быть секретным, так как $\text{НОД}(\alpha^{\gamma'}-1, n) = p$.

2.3 Особенности задачи дискретного логарифмирования по трудно факторизуемому модулю

Рассмотрим возможные пути решения ЗДЛ по трудно факторизуемому модулю. Как было показано ранее в разделе 2.1, решение ЗДЛ в $\mathbb{Z}_n^*$ может быть сведено к комбинации решения двух задач: факторизации n и нахождения дискретного логарифма по модулю каждого простого делителя p, q составного модуля n.

$$y = \alpha^x \bmod n \Rightarrow \begin{cases} y = \alpha^x \bmod p \\ y = \alpha^x \bmod q \end{cases}$$

Обратим внимание, что при решении ЗДЛ по модулям p, q, может возникнуть ситуация, в которой полученные решения не равны, а сравнимы между собой, то есть

$$\begin{cases} y = \alpha^{x_1} \bmod p \\ y = \alpha^{x_2} \bmod q \end{cases} \cdot$$
$$x_1 \neq x_2$$

В случае когда α имеет простой порядок γ такой, что $\gamma \mid (p-1)$, $\gamma \mid (q-1)$,

$$\alpha^\gamma = 1 \bmod n \Rightarrow \begin{cases} \alpha^\gamma = 1 \bmod p \\ \alpha^\gamma = 1 \bmod q \end{cases} \Rightarrow$$
$$y = \alpha^x \bmod n \Rightarrow \begin{cases} y = \alpha^{x_1} \bmod p \\ y = \alpha^{x_2} \bmod q \end{cases},$$

значения x_1 и x_2, получаемые при решении ЗДЛ по простым модулям p и q соответственно, будут одинаковы и равны значению $x = x_1 = x_2$. Из этого следует, что для решения ЗДЛ по трудно факторизуемому модулю $n = pq$, необходимо и достаточно решить:

1) ЗФ модуля n;
2) ЗДЛ по модулю наименьшего простого делителя n (имеющую наименьшую трудоёмкость из двух ЗДЛ).

Другими словами, решение ЗДЛ по обоим делителям p и q не потребуется. Учитывая соотношение вычислительной сложность решения ЗФ и ЗДЛ по простому модулю, вычислительная сложность решения ЗДЛ по модулю наименьшего простого делителя будет меньше вычислительной сложности решения ЗФ и меньше требуемого уровня стойкости в случае использования соотношения размеров простых делителей $|p| = 2|q|$. Поэтому, в случае использования простого порядка γ, использование соотношения размеров простых делителей $|p| = 2|q|$ невозможно. Для обеспечения требуемого уровня вычислительной сложности

решения обоих задач необходимо выбирать размер делителей n в соотношении $2|p| = 2|q|$, где 2 означает, что размер делителя должен превышать требования к размеру делителей ЗФ в 2 раза для выбранного уровня стойкости. В таком случае вычислительная сложность решения ЗДЛ по модулю меньшего простого делителя n и вычислительная сложность решения ЗФ числа n будут не ниже требуемого уровня стойкости (рисунок 2.2). Причём вычислительная сложность решения ЗФ числа n будет выше вычислительной сложности решения ЗДЛ по модулям p, q.

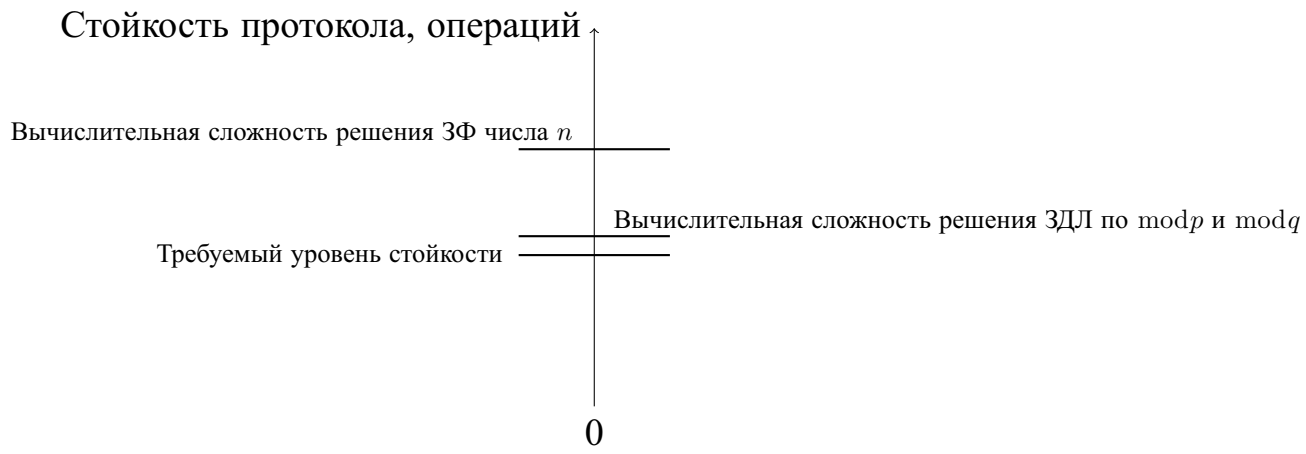

Рисунок 2.2 — Стойкость протоколов, взлом которых требует одновременного решения двух вычислительно трудных задач, в случае использования простого значения порядка γ

В случае использования составного порядка $\gamma = \gamma'\gamma''$ такого, что $\gamma' \mid (p-1)$, $\gamma'' \mid (q-1)$, $\gamma' \nmid (q-1)$ и $\gamma'' \nmid (p-1)$,

$$\alpha^{\gamma} = 1 \bmod n \Rightarrow \begin{cases} \alpha^{\gamma'} = 1 \bmod p \\ \alpha^{\gamma''} = 1 \bmod q \end{cases} \Rightarrow$$

$$y = \alpha^{x} \bmod n \Rightarrow \begin{cases} y = \alpha^{x_1} \bmod p \\ y = \alpha^{x_2} \bmod q \end{cases},$$

значения x_1 и x_2, получаемые при решении ЗДЛ по простым модулям p и q соответственно, будут различны $x_1 \neq x_2 \neq x$, но сравнимы по модулям делителей порядка γ

$$\begin{cases} x \equiv x_1 \bmod \gamma' \\ x \equiv x_2 \bmod \gamma'' \end{cases}.$$

Для восстановления исходного значения x потребуется решение данной системы сравнений с использованием китайкой теоремы об остатках [9].

$$c_1 = \gamma'\gamma^{-1} \bmod \gamma'$$
$$c_2 = \gamma''\gamma^{-1} \bmod \gamma''$$
$$x = x_1 c_1 \gamma\gamma'^{-1} + x_2 c_2 \gamma\gamma''^{-1} \bmod \gamma$$

Из этого следует, что для решения ЗДЛ по трудно факторизуемому модулю $n = pq$, необходимо и достаточно решить:

1) ЗФ числа n;
2) ЗДЛ по модулю p;
3) ЗДЛ по модулю q.

В этом случае можно использовать соотношения размеров простых делителей модуля $|p| = 2|q|$. Для обеспечения требуемого уровня вычислительной сложности решения обоих задач необходимо выбирать размер делителей модуля n следующим образом:

1) размер $|q|$, меньшего делителя модуля n, должен обеспечивать вычислительную сложность решения ЗФ числа n не ниже требуемого уровня;
2) $|p| = 2|q|$, такое соотношение обеспечивает примерно равный уровень вычислительной сложности решения ЗДЛ по модулю p с вычислительной сложностью решения ЗФ числа n.

В таком случае вычислительная сложность решения ЗДЛ по модулю n и вычислительная сложность решения ЗФ числа n будут не ниже требуемого уровня стойкости (рисунок 2.3). Причём вычислительная сложность решения ЗФ числа n будет примерно равна вычислительной сложности решения ЗДЛ по модулю p, а вычислительная сложность решения ЗДЛ по модулю q будет меньше.

Покажем, что в случае использования составного порядка $\gamma = \gamma'\gamma''$, гипотетический алгоритм решения ЗДЛ по трудно факторизуемому модулю n, может быть использован для решения ЗФ числа n и ЗДЛ по модулю p.

1. Выберем случайное число $X \leftarrow_{\$} (\mathbb{N}, X > \gamma)$ и вычислим $y = \alpha^X \bmod n$.
2. Используя гипотетический алгоритм решения ЗДЛ по трудно факторизуемому модулю вычислим значение x такое, что $y = \alpha^x \bmod n$.
3. Разложим на множители значение $X - x$ и найдём значения γ' и γ''. Это достаточно лёгкая вычислительная процедура, поскольку наибольшими множителями этого разложения являются γ' и γ'', размер которых

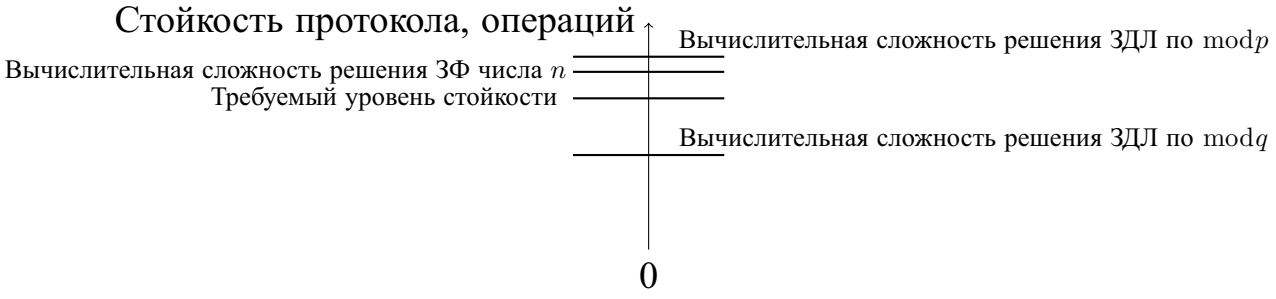

Рисунок 2.3 — Стойкость протокола, взлом которых требует одновременного решения двух вычислительно сложных задач, в случае использования составного значения порядка $\gamma = \gamma'\gamma''$

для обеспечения 128 битовой вычислительной сложности решения ЗДЛ равен 128 битам.

4. Вычислим значение $D = \alpha^{\gamma'} \bmod n$ (отметим, что значение p делит значение $D - 1$, поскольку имеет место соотношение $D \equiv \alpha^{\gamma'} \equiv 1 \ (\mathrm{mod} p)$.
5. Используя алгоритм Евклида найдём наибольший общий делитель НОД$(n, D - 1) = p$.

Таким образом, разложение на множители числа n завершено.

Теперь покажем как гипотетический алгоритм решения ЗДЛ по трудно факторизуемому модулю n, может быть использован для решения ЗДЛ по модулю p. Допустим, нужно найти значение u из заданных значений β и z, где $z = \beta^u \bmod p$ и β - число порядка γ' по модулю p.

1. Сгенерируем значение $\lambda \leftarrow_{\$} (\mathbb{N}, \lambda^{\gamma''} \equiv 1 \bmod q)$, имеющее простой порядок γ'' по модулю q.
2. Вычислим целое значение $\alpha' < n$ из следующей системы сравнений

$$\begin{cases} \alpha' \equiv \beta \bmod p \\ \alpha' \equiv \lambda \bmod q \end{cases}.$$

3. Вычислим целое значение $y < n$ из следующей системы сравнений

$$\begin{cases} y \equiv z \bmod p \\ y \equiv \lambda \bmod q \end{cases}.$$

4. Применим гипотетический алгоритм для вычисления неизвестного x из выражения $y = \alpha'^x \bmod n$.

5. Получим u из уравнения $u = x \bmod \gamma'$.

Значения α' и y, вычисленные на шагах 2 и 3, соответственно, имеют порядок $\gamma'\gamma''$. Из уравнения $y = \alpha'^x \bmod n$ можно получить

$$y \equiv \alpha'^x \bmod n \Rightarrow (y \bmod p) \equiv (\alpha' \bmod p)^x \bmod p \Rightarrow$$
$$\Rightarrow z = \beta^x \bmod p \Rightarrow u \equiv x \bmod \gamma'.$$

Следовательно, предложенный алгоритм вычисляет корректное значение u.

Учитывая данные выкладки и то, что вычислительная сложность вспомогательных вычислений много меньше вычислительной сложности решений ЗФ числа n и ЗДЛ по модулю p, приходим к выводу, что сложность решения ЗДЛ по модулю n имеет один порядок с суммарной сложностью решения ЗФ числа n и ЗДЛ по модулю p.

2.4 Алгоритмы для генерации необходимых параметров

Для генерации подходящих больших чисел можно использовать алгоритмы генерации простых чисел, описанные в [9; 14; 126]. Генерация p,q и $n = pq$ выполняется следующим образом.

1. Сначала необходимо сгенерировать простое значение порядка порождающего элемента группы $\gamma \leftarrow_{\$} (\mathbb{N}, \gamma - prime)$ необходимой длины, например, с использованием алгоритма на основе подбора разложения функции Эйлера [126].

2. Затем необходимо сгенерировать числа $p \leftarrow_{\$} (\mathbb{N}, p = \gamma N + 1,$ НОД$(N, \gamma) = 1, p - prime)$, $q \leftarrow_{\$} (\mathbb{N}, q = \gamma N' + 1,$ НОД$(N', \gamma) = 1, N \neq N', q - prime)$ требуемой длины, где N, N' - случайные чётные числа требуемой длины, условия НОД(N,γ) $= 1$, НОД(N',γ) $= 1$ необходимы для выполнения условий $\gamma \mid (p-1)$, $\gamma \mid (q-1)$, $\gamma^2 \nmid (p-1)$ и $\gamma^2 \nmid (q-1)$, например, с использованием детерминистических алгоритмов генерации простых чисел, описанных в работах [126; 127]. Необходимая длина случайных чётных чисел N и N' высчитывается в зависимости от требуемой длины p и q по формулам $|N| = |p| - |\gamma|$, $|N'| = |q| - |\gamma|$ соответственно.

Для составного порядка $\gamma = \gamma'\gamma''$ выполняются аналогичные операции, но сначала генерируются γ' и γ'' нужного размера. И на втором шаге необходимо сгенерировать числа $p = \gamma' N + 1$, $q = \gamma'' N' + 1$ с условиями НОД(N',γ') $= 1$ и НОД(N,γ'') $= 1$, $\gamma' \neq \gamma''$ (для выполнения условий $\gamma' \mid (p-1)$, $\gamma'' \mid (q-1)$,

$\gamma' \nmid (q-1)$ и $\gamma'' \nmid (p-1)$). Необходимая длина случайных чётных чисел N и N' высчитывается по формулам $|N| = |p| - |\gamma'|$, $|N'| = |q| - |\gamma''|$ соответственно.

После выполнения описанных этапов мы имеем значения: $n, p, q, \gamma(\gamma', \gamma'')$. Осталось сгенерировать порождающий элемент группы α. Способ случайного выбора α и проверки выполнимости соотношения $\alpha^\gamma = 1 \bmod p$ является вычислительно неэффективным. Для этого используем два следующих алгоритма нахождения чисел, относящихся к заданному показателю, аналогичные алгоритмы описаны в книгах [9; 126], но для простого модуля. Предложенные алгоритмы используют критерии описанные в работах [94; 96] и в разделе 2.2 и позволяют быстро сгенерировать порождающий элемент группы α, удовлетворяющий $\alpha \not\equiv 1 \bmod p$, $\alpha \not\equiv 1 \bmod q$.

В случае выбора простого значения γ, найти подходящий α можно следующим вычислительно эффективным способом.

1. Сгенерировать случайное число $\beta \leftarrow_\$ (\mathbb{N}, \beta < n,$ НОД$(\beta, n) = 1)$.
2. Вычислить $\gamma^* = \varphi(n)/\gamma^2$ и $z = \beta^{\gamma^*} \bmod n$.
3. Если $z = 1$, то перейти на шаг 1, иначе $\alpha = \beta^{\gamma *} \bmod n$.

Блок схема предложенного алгоритма представлена на рисунке 2.4.

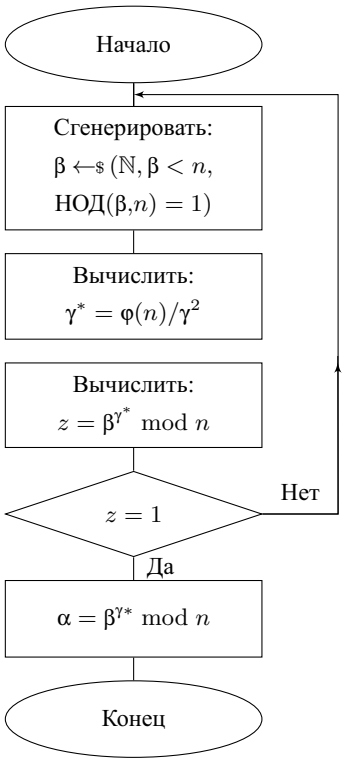

Рисунок 2.4 — Блок схема поиска порождающего элемента группы в случае использования простого порядка

В случае выбора составного значения $\gamma = \gamma'\gamma''$.

1. Сгенерировать случайное число $\beta \leftarrow_{\$} (\mathbb{N}, \beta < n,$ НОД$(\beta, n) = 1)$.
2. Вычислить $\gamma^* = \varphi(n)/\gamma$ и $z = \beta^{\gamma^*} \bmod n$.
3. Если $z = 1$, то перейти на шаг 1, иначе $\alpha = \beta^{\gamma *} \bmod n$.

Блок схем алгоритма поиска порождающего элемента группы в случае использования составного значения порядка представлена на рисунке 2.5.

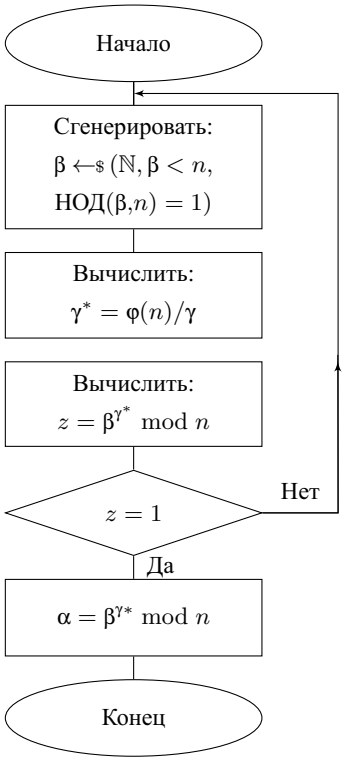

Рисунок 2.5 — Блок схема поиска порождающего элемента группы в случае использования составного порядка

После генерации порождающего элемента группы, мы имеем все параметры для использования в протоколах, в основе которых лежит вычислительная сложность решения ЗДЛ по трудно факторизуемому модулю: $n, p, q, \alpha, \gamma(\gamma', \gamma'')$.

2.5 Обоснование необходимых длин параметров

Для обоснования необходимых длин параметров используется методика ECRYPT II [49].

Рассмотрим ЗДЛ по трудно факторизуемому модулю $n = pq$, являющимся произведением двух больших простых чисел p, q. ЗДЛ по трудно факторизуемому модулю определяется относительно некоторых известных чисел $\alpha < n$

(НОД(α,n) $= 1$), y, и состоит в нахождении такого x, что выполняется следующее выражение

$$y = \alpha^x \bmod n.$$

Как было показано в разделе 2.3, ЗДЛ по трудно факторизуемому модулю эквивалентна решению следующих вычислительно сложных задач в случае использования:

1) простого порядка γ - ЗФ n, ЗДЛ q;

2) составного порядка $\gamma = \gamma'\gamma''$ - ЗФ n, ЗДЛ q, ЗДЛ p.

На сегодняшний день рекомендуемый уровень стойкости протоколов составляет 2^{128} операций модульного умножения для решения вычислительно сложной задачи, лежащей в её основе (для построения протоколов с 128 битовой стойкостью), для долгосрочных перспектив рекомендуется ориентироваться на 2^{160} операций модульного умножения (160 битовая стойкость) [49;57], однако, на практике используется модуль RSA длиной 2048, что эквивалентно 100 битовой стойкости при оценке по методике ECRYPT II.

Для обеспечения вычислительной сложности решения ЗФ требуется чтобы меньший делитель q имел следующие размеры:

1) для 100 битовой стойкости - 1024 бит;

2) для 128 битовой стойкости - 1624 бит;

3) для 160 битовой стойкости - 2656 бит.

Размер второго делителя p может изменяться, но он не должен быть меньше размера q.

Для обеспечения вычислительной сложности решения ЗДЛ для простого порядка γ требуется чтобы оба делителя модуля n были не ниже следующих размеров:

1) для 100 битовой стойкости - 2048 бит;

2) для 128 битовой стойкости - 3248 бит;

3) для 160 битовой стойкости - 5112 бит.

Для обеспечения вычислительной сложности решения по ЗДЛ для составного порядка $\gamma = \gamma'\gamma''$ требуется, чтобы больший делитель p модуля n был не ниже следующих размеров:

1) для 100 битовой стойкости - 2048 бит;

2) для 128 битовой стойкости - 3248 бит;

3) для 160 битовой стойкости - 5112 бит.

А второй делитель q, в данном случае, должен быть не меньше рекомендуемого размера для обеспечения вычислительной сложности решения ЗФ n, и иметь размер не менее:

1) для 100 битовой стойкости - 1024 бит;

2) для 128 битовой стойкости - 1624 бит;

3) для 160 битовой стойкости - 2656 бит.

Размер порядка γ для обеспечения невозможности применения метода "больших и малых шагов" [9; 12; 43] должен иметь следующие размеры:

1) для 100 битовой стойкости - 200 бит;

2) для 128 битовой стойкости - 256 бит;

3) для 160 битовой стойкости - 320 бит.

Результирующие рекомендуемые размеры сведены в таблицу 7.

Таблица 7 — Рекомендуемые размеры параметров протоколов для обеспечения необходимого уровня стойкости

	Стойкость (бит)	Размеры параметров (бит)					
		$\|n\|$	$\|p\|$	$\|q\|$	$\|\gamma\|$	$\|\gamma'\|$	$\|\gamma''\|$
Простой порядок	100	4096	2048	2048	200	-	-
	128	6496	3248	3248	256	-	-
	160	10224	5112	5112	320	-	-
Составной порядок	100	3072	2048	1024	200	100	100
	128	4872	3248	1624	256	128	128
	160	7768	5112	2656	320	160	160

2.6 Выводы ко второй главе

1. Обоснован новый метод построения протоколов, взлом которых требует одновременного решения двух вычислительно трудных задач: факторизации и дискретного логарифмирования, отличающийся возможностью использования соотношений делителей составного модуля 1 к 2 и универсальностью для построения основных типов протоколов, что позволяет повысить производительность, уменьшить размеры ЭЦП и позволяет построить все основные типы протоколов.
2. Предложены ограничения на выбор параметров, используемых в новом методе, предотвращающих быструю факторизацию трудно факторизуемого модуля с использованием алгоритма Евклида.
3. Показаны особенности ЗДЛ по трудно факторизуемому модулю, в зависимости от выбранной структуры порядка группы.
4. Показано, что ЗДЛ по трудно факторизуемому модулю эквивалентна решению следующих вычислительно сложных задач в случае использования:
    а) простого порядка - ЗФ составного модуля, ЗДЛ меньшего делителя составного модуля;
    б) составного порядка - ЗФ составного модуля, ЗДЛ первого множителя составного модуля, ЗДЛ второго множителя составного модуля.
5. Показана связь ЗФ и ЗДЛ, а именно, гипотетический алгоритм решения ЗДЛ по трудно факторизуемому модулю, может быть использован для факторизации составного модуля и решению ЗДЛ по множителям составного модуля.
6. Предложены вычислительно эффективные алгоритмы генерации необходимых параметров ЗДЛ по трудно факторизуемому модулю, для использования при разработке новых протоколов.
7. Произведён расчёт необходимых длин параметров для обеспечения требуемых уровней стойкости протоколов, построенных с использованием ЗДЛ по трудно факторизуемому модулю.

Глава 3. Открытое распределение ключей и шифрование с открытым ключом в информационно-телекоммуникационных системах

Для создания алгоритмов и протоколов, взлом которых требует решения двух независимых вычислительно сложных ЗФ и ЗДЛ, необходимо разработать алгоритмы генерации пар ключей и распределения их по открытым каналам связи. Существующие алгоритмы не пригодны для использования в ЗДЛ по трудно факторизуемому модулю.

3.1 Децентрализованная генерация ключей

Пусть p, q - большие простые числа, $n = pq$ - составной трудно факторизуемый модуль, α - порождающий элемент, γ - порядок элемента α, т.е. $\alpha^{\gamma} = 1 \bmod n$. Возможны два случая генерации ОК с использованием:

1) простого порядка γ такого, что $\gamma \mid (p-1)$, $\gamma \mid (q-1)$, $\gamma^2 \nmid (p-1)$, $\gamma^2 \nmid (q-1)$. Причём γ будет несекретным.
2) составного порядка $\gamma = \gamma'\gamma''$ такого, что $\gamma' \mid (p-1)$, $\gamma'' \mid (q-1)$, $\gamma' \nmid (q-1)$ и $\gamma'' \nmid (p-1)$. В таком случае значение γ нужно хранить в секрете.

Рассмотрим возможные сценарии генерации ОК. Генерация параметров $n, p, q, \alpha, \gamma(\gamma', \gamma'')$ производится с использованием алгоритмов, описанных в разделе 2.4). Далее приводится описание трёх алгоритмов генерации ключей, обеспечивающих невозможность быстрого решения ЗФ модуля n с использованием алгоритма Евклида, такая возможность описана в разделе 2.2.

3.1.1 Децентрализованная генерация ключей с использованием простого порядка группы

Первый случай, децентрализованная генерация ОК при использовании простого порядка группы, когда каждый пользователь генерирует все параметры для ОК самостоятельно.

1. Каждый i-тый пользователь ($i = 1, 2, 3, ...$) генерирует случайное простое значение порядка группы $\gamma_i \leftarrow_{\$} (\mathbb{N}, \gamma_i - prime)$.
2. Затем, генерирует простые числа $p_i \leftarrow_{\$} (\mathbb{N}, \gamma_i \mid (p_i - 1), \gamma_i^2 \nmid (p_i - 1), p_i - prime)$, $q_i \leftarrow_{\$} (\mathbb{N}, \gamma_i \mid (q_i - 1), \gamma_i^2 \nmid (q_i - 1), q_i \neq p_i, q_i - prime)$ требуемой длины.

3. Затем вычисляет составной трудно факторизуемый модуль $n_i = p_i q_i$.
4. Находит $\alpha_i \leftarrow_{\$} (\mathbb{N}, \alpha_i^{\gamma_i} = 1 \bmod n_i)$, и берёт его в качестве порождающего элемента группы.
5. Генерирует элемент СК $x_i \leftarrow_{\$} (\mathbb{N}, x_i < \gamma_i, |x_i| \approx |\gamma_i|)$.
6. Затем вычисляет элемент ОК $y_i = \alpha_i^{x_i} \bmod n_i$.

ОК является четвёрка значений (y_i, α_i, γ_i, n_i), СК являются (x_i, q_i, p_i). Причём после проверки условий для порядка γ_i рекомендуется уничтожить делители p_i, q_i, так как для дальнейших вычислений в протоколах эти значения не потребуются. Блок схема такой генерации ОК представлена на рисунке 3.1.

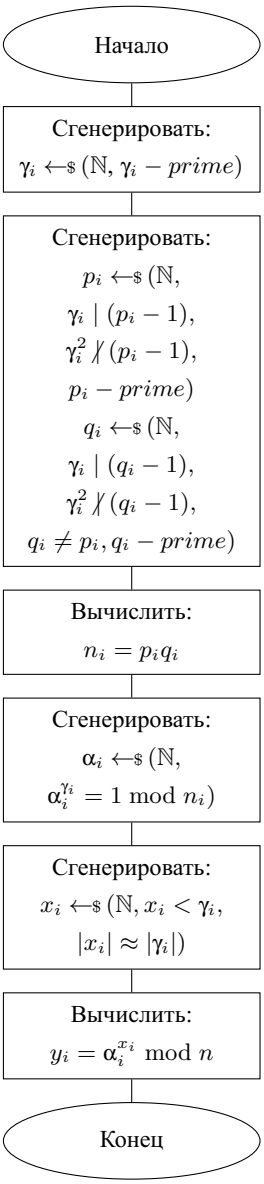

Рисунок 3.1 — Децентрализованная генерация ключей с использованием простого порядка группы

3.1.2 Децентрализованная генерация ключей с использованием УЦ при использовании простого порядка группы

Так как в случае использования простого порядка группы γ, он должен быть несекретным, то возможна генерация параметров p, q, n, γ, α в централизованном режиме с использованием УЦ. В таком случае УЦ выполняет следующие действия:

1. УЦ генерирует случайное простое значение порядка группы $\gamma \leftarrow_{\$} (\mathbb{N}, \gamma - prime)$.
2. Затем, генерирует простые числа $p \leftarrow_{\$} (\mathbb{N}, \gamma \mid (p-1), \gamma^2 \nmid (p-1), p - prime)$, $q \leftarrow_{\$} (\mathbb{N}, \gamma \mid (q-1), \gamma^2 \nmid (q-1), q \neq p, q - prime)$ требуемой длины.
3. Затем вычисляет составной трудно факторизуемый модуль $n = pq$ и уничтожает p, q.
4. Находит $\alpha \leftarrow_{\$} (\mathbb{N}, \alpha^{\gamma} = 1 \bmod n)$, и берёт его в качестве порождающего элемента группы.
5. Предоставляет пользователям числа (n, γ, α), являющиеся системными параметрами.

Пользователи запрашивают у УЦ системные параметры (n, γ, α) и выполняют следующие действия для генерации пары своих СК и ОК:

1. Каждый i-тый пользователь ($i = 1, 2, 3, ...$) генерирует свой личный СК $x_i \leftarrow_{\$} (\mathbb{N}, x_i < \gamma, |x_i| \approx |\gamma|)$.
2. Затем вычисляет ОК $y_i = \alpha_i^{x_i} \bmod n$.

Такая схема генерации может использоваться только в случае, если все участники доверяют УЦ, в противном случае стойкость их ОК будет основана только на ЗДЛ по простому модулю.

Блок схема генерации ключей при использовании УЦ и простого порядка группы представлена на рисунке 3.2.

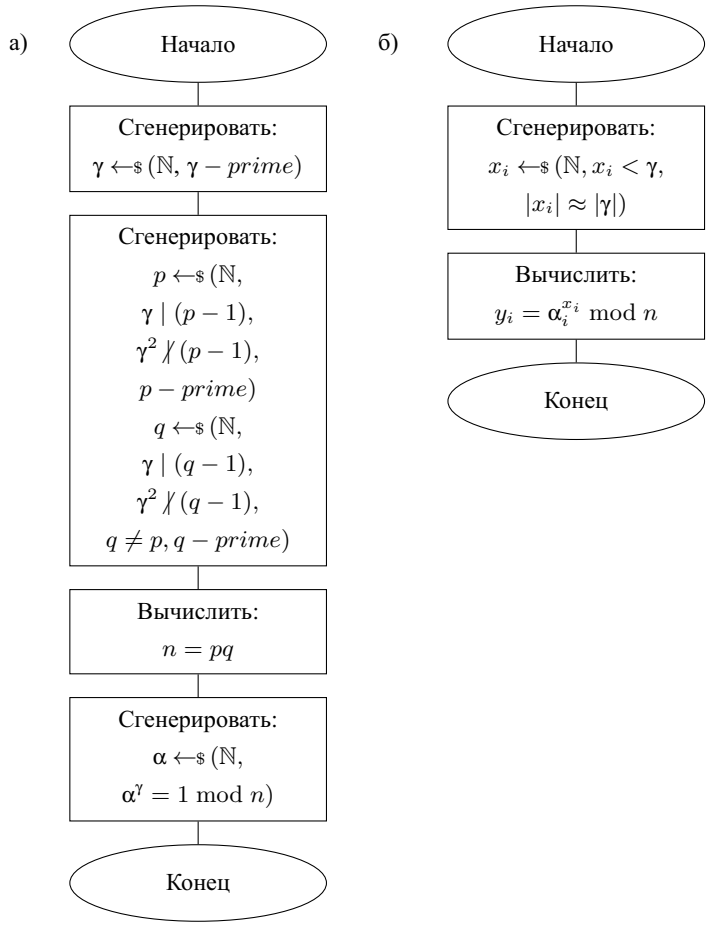

Рисунок 3.2 — а) Генерация системных параметров с использованием УЦ; б) децентрализованная генерация ключей

3.1.3 Децентрализованная генерация ключей с использованием составного порядка группы

В случае использования составного порядка группы $\gamma = \gamma'\gamma''$ значение γ необходимо держать в секрете. Следовательно, значения γ не могут быть выработаны централизованно в УЦ. Поэтому, в таком случае возможна только децентрализованная генерация ОК, состоящая из следующих шагов.

1. Каждый i-тый пользователь ($i = 1, 2, 3,...$) генерирует случайные простые значения $\gamma_i' \leftarrow_\$ (\mathbb{N}, \gamma_i' - prime)$, $\gamma_i'' \leftarrow_\$ (\mathbb{N}, \gamma_i'' \neq \gamma', \gamma_i'' - prime)$.
2. Генерирует простые числа $p_i \leftarrow_\$ (\mathbb{N}, \gamma_i' \mid (p_i - 1), \gamma_i'' \nmid (p_i - 1), p_i - prime)$, $q_i \leftarrow_\$ (\mathbb{N}, \gamma_i' \nmid (q_i - 1), \gamma_i'' \mid (q_i - 1), q_i \neq p_i, q_i - prime)$ требуемой длины.
3. В качестве порядка группы берёт $\gamma_i = \gamma_i'\gamma_i''$.
4. Затем вычисляет составной трудно факторизуемый модуль $n_i = p_i q_i$ и уничтожает делители p_i, q_i.

5. Находит $\alpha_i \leftarrow_\$ (\mathbb{N}, \alpha_i^{\gamma_i} = 1 \bmod n_i)$, и берёт его в качестве порождающего элемента группы.
6. Генерирует личный СК $x_i \leftarrow_\$ (\mathbb{N}, x_i < \gamma_i, |x_i| \approx |\gamma_i|)$.
7. Затем вычисляет элемент ОК $y_i = \alpha_i^{x_i} \bmod n$.

В таком случае ОК является тройка значений (α_i, y_i, n_i), СК является пара значений (x_i, γ_i). Блок схема генерации ключей при использовании составного порядка группы представлена на рисунке 3.3.

Рисунок 3.3 — Децентрализованная генерация ключей с использованием составного порядка группы

3.2 Протоколы обмена ключами

3.2.1 Протокол обмена ключами с генерацией системных параметров для ОК в УЦ

Пусть УЦ генерирует системные параметры (α, n, γ) по алгоритму, описанному в разделе 3.1.2.

По окончанию процедуры УЦ предоставляет всем пользователям значения (α, n, γ) для генерации их секретных ключей. Пользователи генерируют ОК следующим образом.

1. Каждый i-ый пользователь ($i = 1, 2, 3, ...$) генерирует случайное число $x_i \leftarrow_{\$} (\mathbb{N}, x_i < \gamma, |x_i| \approx |\gamma|)$, которое будет его СК.
2. Затем каждый пользователь вычисляет свой открытый ключ $y_i = \alpha^{x_i} \bmod n$.
3. Значение y_i будет являться ОК пользователя.

После генерации ОК, каждый пользователь отправляет свои ОК в УЦ. Два пользователя i и j могут вычислить общий секретный ключ следующим образом, аналогично схеме Диффи-Хеллмана [11] и Захавы Шмуеля [63].

1. Пользователь i генерирует случайное число $u_i \leftarrow_{\$} (\mathbb{N}, u_i < \gamma, |u_i| \approx |\gamma|)$.
2. Затем вычисляет значение $R_i = y_j^{u_i} \bmod n$ и оправляет R_i пользователю j.
3. Пользователь j генерирует случайное число $u_j \leftarrow_{\$} (\mathbb{N}, u_j < \gamma, |u_j| \approx |\gamma|)$.
4. Затем вычисляет значение $R_j = y_i^{u_j} \bmod n$ и оправляет R_j пользователю i.
5. Пользователь i вычисляет $Z_{ij} = y_j^{u_i} R_j^{x_i} \bmod n$.
6. Пользователь j вычисляет $Z_{ji} = y_i^{u_j} R_i^{x_j} \bmod n$.

Доказательство корректности протокола..

$$\left.\begin{array}{l} Z_{ij} = y_j^{u_i} R_j^{x_i} \bmod n = \alpha^{x_j u_i} \alpha^{u_j x_i} \bmod n \\ Z_{ji} = y_i^{u_j} R_i^{x_j} \bmod n = \alpha^{x_i u_j} \alpha^{u_i x_j} \bmod n \end{array}\right\} \Rightarrow$$
$$\Rightarrow Z_{ji} = Z_{ij} = Z$$

Таким образом пользователи сформировали общий секретный ключ Z.

Основным отличием от схем Диффи-Хеллмана [11], Захавы Шмуеля [63], Бихама [128] является не только применение трудно факторизуемого модуля и

простого порядка группы, но и использование рандомизирующих параметров u_i и u_j. Эти шаги увеличивают трудоёмкость вычислений (с 2-ух операций возведения в степень, до 6ти), но благодаря этому ключи в разных сеансах связи двух абонентов будут различны. Такая схема применена в двух последующих протоколах обмена ключами, описываемых в разделах 3.2.2 и 3.2.3. Для генерации ключей и системных параметров используется алгоритм, описанный в разделе 3.1.1.

Общая схема протокола представлена на рисунке 3.4.

УЦ	**Пользователь** i	**Пользователь** j
	Генерация ключей	
$\gamma \leftarrow_\$ (\mathbb{N}, \gamma - prime)$		
$p \leftarrow_\$ (\mathbb{N}, \gamma \mid (p-1),$		
$\gamma^2 \nmid (p-1), p - prime)$		
$q \leftarrow_\$ (\mathbb{N}, \gamma \mid (q-1),$		
$\gamma^2 \nmid (q-1), q \neq p,$		
$q - prime)$		
$n = pq$		
$\alpha \leftarrow_\$ (\mathbb{N}, \alpha^\gamma = 1 \bmod n)$		
$\xrightarrow{(n, \alpha, \gamma)}$		
	$\xrightarrow{(n, \alpha, \gamma)}$	
	$x_i \leftarrow_\$ (\mathbb{N}, x_i < \gamma,$	$x_j \leftarrow_\$ (\mathbb{N}, x_j < \gamma,$
	$\lvert x_i \rvert \approx \lvert \gamma \rvert)$	$\lvert x_j \rvert \approx \lvert \gamma \rvert)$
	$y_i = \alpha^{x_i} \bmod n$	$y_j = \alpha^{x_j} \bmod n$
	$\xrightarrow{OK_i : (y_i)}$	
	$\xleftarrow{OK_j : (y_j)}$	
	Обмен случайными значениями	
	$u_i \leftarrow_\$ (\mathbb{N}, u_i < \gamma,$	
	$\lvert u_i \rvert \approx \lvert \gamma \rvert)$	
	$R_i = \alpha^{u_i} \bmod n$	
	$\xrightarrow{R_i}$	
		$u_j \leftarrow_\$ (\mathbb{N}, u_j < \gamma,$
		$\lvert u_j \rvert \approx \lvert \gamma \rvert)$
		$R_j = \alpha^{u_j} \bmod n$
	$\xleftarrow{R_j}$	
	Вычисление общего секрета	
	$Z_{ij} = y_j^{u_i} R_j^{x_i} \bmod n$	$Z_{ji} = y_i^{u_j} R_i^{x_j} \bmod n$
	$Z_{ij} = Z_{ji} = Z$	

Рисунок 3.4 — Протокол обмена ключами с генерацией параметров для ОК в УЦ и использованием простого порядка группы

3.2.2 Протокол обмена ключами без генерации системных параметров для ОК в УЦ с использованием простого порядка группы

Каждый i-ый пользователь ($i = 1, 2, 3, ...$) генерирует параметры y_i, α_i, γ_i, n_i, x_i, q_i, p_i с использованием алгоритмов, описанных в разделе 3.1.1. После чего обменивается своим открытым ключом (y_i, α_i, γ_i, n_i) с другими пользователями.

Два пользователя i и j могут вычислить общий секретный ключ следующим образом.

1. Пользователь i генерирует случайное число $u_i \leftarrow_{\$} (\mathbb{N}, u_i < \gamma_j, |u_i| \approx |\gamma_j|)$.
2. Затем вычисляет значение $R_i = \alpha_j^{u_i} \bmod n_j$ и оправляет R_i пользователю j.
3. Пользователь j генерирует случайное число $u_j \leftarrow_{\$} (\mathbb{N}, u_j < \gamma_i, |u_j| \approx |\gamma_i|)$.
4. Затем вычисляет значение $R_j = \alpha_i^{u_j} \bmod n_i$ и оправляет R_j пользователю i.
5. Пользователь i вычисляет $Z'_{ij} = y_j^{u_i} \bmod n_j$, $Z''_{ij} = R_j^{x_i} \bmod n_i$, $Z_{ij} = Z'_{ij} Z''_{ij}$.
6. Пользователь j вычисляет $Z'_{ji} = y_i^{u_j} \bmod n_i$, $Z''_{ji} = R_i^{x_j} \bmod n_j$, $Z_{ji} = Z'_{ji} Z''_{ji}$.

На последнем шаге значения $Z_{ij} = Z_{ji} = Z$, то есть пользователи сформировали общий секретный ключ Z.

Доказательство корректности протокола.

$$\left\{ \begin{array}{l} Z'_{ij} = y_j^{u_i} \bmod n_j = \alpha_j^{x_j u_i} \bmod n_j \\ Z''_{ji} = R_i^{x_j} \bmod n_j = \alpha_j^{u_i x_j} \bmod n_j \end{array} \right\} \Rightarrow Z''_{ji} = Z'_{ij}$$

$$\left\{ \begin{array}{l} Z'_{ji} = y_i^{u_j} \bmod n_i = \alpha_i^{x_i u_j} \bmod n_i \\ Z''_{ij} = R_j^{x_i} \bmod n_i = \alpha_i^{u_j x_i} \bmod n_i \end{array} \right\} \Rightarrow Z''_{ij} = Z'_{ji}$$

$$\left\{ \begin{array}{l} Z''_{ji} = Z'_{ij} \\ Z''_{ij} = Z'_{ji} \\ Z_{ij} = Z'_{ij} Z''_{ij} \\ Z_{ji} = Z'_{ji} Z''_{ji} \end{array} \right\} \Rightarrow Z_{ij} = Z_{ji} = Z$$

Таким образом пользователи сформировали общий секретный ключ Z.

Основное отличие от протокола, описанного в разделе 3.2 заключается в использовании разных модулей для ключей разных пользователей и разных генераторов группы. Дополнительно данный протокол является более предпочтительным с точки зрения безопасности, так как в данном случае стойкость обоих ЗФ и ЗДЛ зависит только от пользователей, генерирующих свои ключи и системные параметры самостоятельно.

Общая схема протокола представлена на рисунке 3.5.

Пользователь i	**Пользователь** j
..... Генерация открытых ключей	
$\gamma_i \leftarrow_\$ (\mathbb{N}, \gamma_i - prime)$	$\gamma_j \leftarrow_\$ (\mathbb{N}, \gamma_j - prime)$
$p_i \leftarrow_\$ (\mathbb{N}, \gamma_i \mid (p_i - 1),$	$p_j \leftarrow_\$ (\mathbb{N}, \gamma_j \mid (p_j - 1),$
$\gamma_i^2 \nmid (p_i - 1), p_i - prime)$	$\gamma_j^2 \nmid (p_j - 1), p_j - prjme)$
$q_i \leftarrow_\$ (\mathbb{N}, \gamma_i \mid (q_i - 1),$	$q_j \leftarrow_\$ (\mathbb{N}, \gamma_j \mid (q_j - 1),$
$\gamma_i^2 \nmid (q_i - 1), q_i \neq p_i, q_i - prime)$	$\gamma_j^2 \nmid (q_j - 1), q_j \neq p_j, q_j - prjme)$
$n_i = p_i q_i$	$n_j = p_j q_j$
$\alpha_i \leftarrow_\$ (\mathbb{N}, \alpha_i^{\gamma_i} = 1 \bmod n_i)$	$\alpha_j \leftarrow_\$ (\mathbb{N}, \alpha_j^{\gamma_j} = 1 \bmod n_j)$
$x_i \leftarrow_\$ (\mathbb{N}, x_i < \gamma_i, \lvert x_i \rvert \approx \lvert \gamma_i \rvert)$	$x_j \leftarrow_\$ (\mathbb{N}, x_j < \gamma_j, \lvert x_j \rvert \approx \lvert \gamma_j \rvert)$
$y_i = \alpha_i^{x_i} \bmod n$	$y_j = \alpha_j^{x_j} \bmod n$
$\xrightarrow{OK_i : (y_i, \alpha_i, \gamma_i, n_i)}$	
$\xleftarrow{OK_j : (y_j, \alpha_j, \gamma_j, n_j)}$	
..... Обмен случайными значениями	
$u_i \leftarrow_\$ (\mathbb{N}, u_i < \gamma_j,$	
$\lvert u_i \rvert \approx \lvert \gamma_j \rvert)$	
$R_i = \alpha_j^{u_i} \bmod n_j$	
$\xrightarrow{R_i}$	
	$u_j \leftarrow_\$ (\mathbb{N}, u_j < \gamma_i,$
	$\lvert u_j \rvert \approx \lvert \gamma_i \rvert)$
	$R_j = \alpha_i^{u_j} \bmod n_i$
$\xleftarrow{R_j}$	
..... Вычисление общего секрета	
$Z'_{ij} = y_j^{u_i} \bmod n_j$	$Z'_{ji} = y_i^{u_j} \bmod n_i$
$Z''_{ij} = R_j^{x_i} \bmod n_i$	$Z''_{ji} = R_i^{x_j} \bmod n_j$
$Z_{ij} = Z'_{ij} Z''_{ij}$	$Z_{ji} = Z'_{ji} Z''_{ji}$
..... $Z_{ij} = Z_{ji} = Z$	

Рисунок 3.5 — Протокол обмена ключами без генерации системных параметров для ОК в УЦ с использованием простого порядка группы

3.2.3 Протокол обмена ключами без генерации системных параметров для ОК в УЦ с использованием составного порядка группы

Алгоритм обмена ключами с использованием составного порядка группы γ, такого что $\gamma' \mid (p-1)$, $\gamma'' \mid (q-1)$, $\gamma' \nmid (q-1)$ и $\gamma'' \nmid (p-1)$ отличается от описанного в разделе 3.2.2 только процедурой генерации ключей и условиями генерации случайных значений $u_i \leftarrow_{\$} (\mathbb{N}, u_i < \gamma_i, |u_i| \approx |\gamma_i|)$ и $u_j \leftarrow_{\$} (\mathbb{N}, u_i < \gamma_i, |u_i| \approx |\gamma_i|)$. Ограничения при генерации случайных значений u_i и u_j служат для привязки размера генерируемого значения к размеру γ, который в свою очередь привязан к вычислительной сложности решения ЗДЛ. Это позволяет обеспечивать вычислительную сложность решения ЗДЛ $R_i = \alpha_j^{u_i} \bmod n_i$ и ЗДЛ $R_i = \alpha_j^{u_i} \bmod n_j$ в предлагаемом протоколе на одном уровне с вычислительной сложностью решения ЗДЛ для уравнений генерации открытых ключей $y_i = \alpha_i^{x_i} \bmod n$ и $y_j = \alpha_j^{x_j} \bmod n$.

Значение γ всех участников протокола хранится в секрете и не используется в протоколе обмена ключами. Для генерации ключей два пользователя могут использовать алгоритм децентрализованной генерации с использованием составного порядка группы, описанный ранее в разделе 3.1.3.

Общая схема протокола представлена на рисунке 3.6.

Пользователь i	**Пользователь** j
..... Генерация открытых ключей	
$\gamma'_i \leftarrow_\$ (\mathbb{N}, \gamma'_i - prime)$	$\gamma'_j \leftarrow_\$ (\mathbb{N}, \gamma'_j - prime)$
$\gamma''_i \leftarrow_\$ (\mathbb{N},$	$\gamma''_j \leftarrow_\$ (\mathbb{N},$
$\gamma''_i \neq \gamma', \gamma''_i - prime)$	$\gamma''_j \neq \gamma', \gamma''_j - prime)$
$p_i \leftarrow_\$ (\mathbb{N}, \gamma'_i \mid (p_i - 1),$	$p_j \leftarrow_\$ (\mathbb{N}, \gamma'_j \mid (p_j - 1),$
$\gamma''_i \nmid (p_i - 1), p_i - prime)$	$\gamma''_j \nmid (p_j - 1), p_j - prime)$
$q_i \leftarrow_\$ (\mathbb{N}, \gamma'_i \nmid (q_i - 1),$	$q_j \leftarrow_\$ (\mathbb{N}, \gamma'_j \nmid (q_j - 1),$
$\gamma''_i \mid (q_i - 1), q_i \neq p_i, q_i - prime)$	$\gamma''_j \mid (q_j - 1), q_j \neq p_j, q_j - prime)$
$\gamma_i = \gamma'_i \gamma''_i$	$\gamma_j = \gamma'_j \gamma''_j$
$n_i = p_i q_i$	$n_j = p_j q_j$
$\alpha_i \leftarrow_\$ (\mathbb{N}, \alpha_i^{\gamma_i} = 1 \bmod n_i)$	$\alpha_j \leftarrow_\$ (\mathbb{N}, \alpha_j^{\gamma_j} = 1 \bmod n_j)$
$\xrightarrow{\textbf{OK}_i\textbf{:}\ (y_i, \alpha_i, n_i)}$	
$\xleftarrow{\textbf{OK}_j\textbf{:}\ (y_j, \alpha_j, n_j)}$	
..... Обмен случайными значениями	
$u_i \leftarrow_\$ (\mathbb{N}, u_i < \gamma_i,$	
$\lvert u_i \rvert \approx \lvert \gamma_i \rvert)$	
$R_i = \alpha_j^{u_i} \bmod n_j$	
$\xrightarrow{R_i}$	
	$u_j \leftarrow_\$ (\mathbb{N}, u_j < \gamma_j,$
	$\lvert u_j \rvert \approx \lvert \gamma_j \rvert)$
	$R_j = \alpha_i^{u_j} \bmod n_i$
$\xleftarrow{R_j}$	
..... Вычисление общего секрета	
$Z'_{ij} = y_j^{u_i} \bmod n_j$	$Z'_{ji} = y_i^{u_j} \bmod n_i$
$Z''_{ij} = R_j^{x_i} \bmod n_i$	$Z''_{ji} = R_i^{x_j} \bmod n_j$
$Z_{ij} = Z'_{ij} Z''_{ij}$	$Z_{ji} = Z'_{ji} Z''_{ji}$
..... $Z_{ij} = Z_{ji} = Z$	

Рисунок 3.6 — Протокол обмена ключами без генерации системных параметров для ОК в УЦ с использованием составного порядка группы

3.3 Протокол шифрования с открытым ключом

В отличии от протоколов обмена ключами, алгоритм шифрования (защитного преобразования) работает при любом выборе структуры порядка группы γ, так как значение порядка в протоколе не участвует.

Для генерации системных параметров и пары ОК, СК пользователи могут использовать протоколы децентрализованной генерации ключей, описанные в разделе 3.1.

Для отправки зашифрованного сообщения отправителю потребуется от получателя его ОК состоящий минимум из тройки значений (y, α, n). Обозначим алгоритм генерации ОК, СК как $y, \alpha, n, x \leftarrow GenK()$.

Чтобы отправить некоторое сообщение $M < n$ владельцу ОК, с использованием вычислений по трудно факторизуемому модулю, можно воспользоваться следующим алгоритмом открытого шифрования по аналогии с шифрованием по открытому ключу Эль-Гамаля [38]. Процедура шифрования сообщения $M < n$ состоит из следующих шагов.

1. Отправитель берёт ОК получателя (y, α, n).
2. Генерирует случайное число $u \leftarrow_{\$} (\mathbb{N})$, с длиной зависящей от желаемого уровня стойкости протокола (см. раздел 2.5).
3. Вычисляет $R = \alpha^u \bmod n$ и $Z = y^u \bmod n$.
4. Затем шифрует сообщение M по формуле: $C = MZ \bmod n$.
5. Отправляет криптограмму (R, C) получателю.

Получатель выполняет следующие шаги, для расшифрования сообщения $M < n$.

1. Вычисляет $W = R^{-x} \bmod n$, используя свой личный СК x.
2. Восстанавливает сообщение M: $M = CW \bmod n$.

Основное отличие от оригинальной схемы шифрования по открытому ключу Эль-Гамаля [38], является использование ЗДЛ по трудно факторизуемому модулю. Общая схема протокола представлена на рисунке 3.7, блок схемы алгоритмов шифрования и расшифрования - на рисунке 3.8.

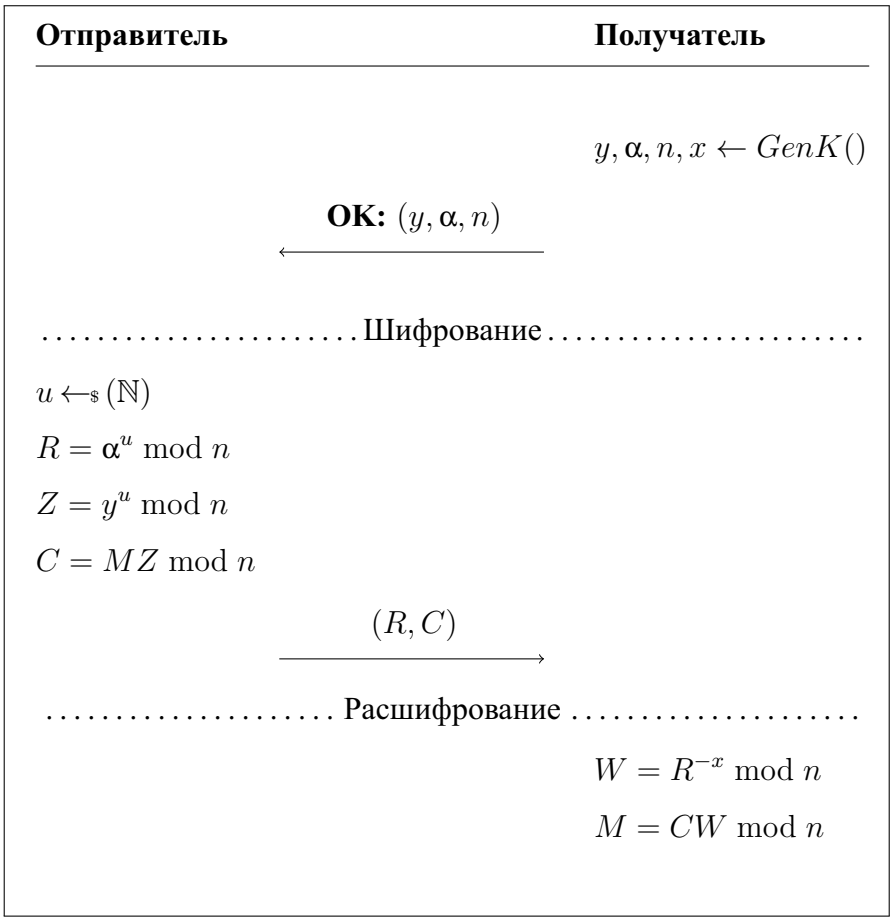

Рисунок 3.7 — Протокол шифрования с открытым ключом

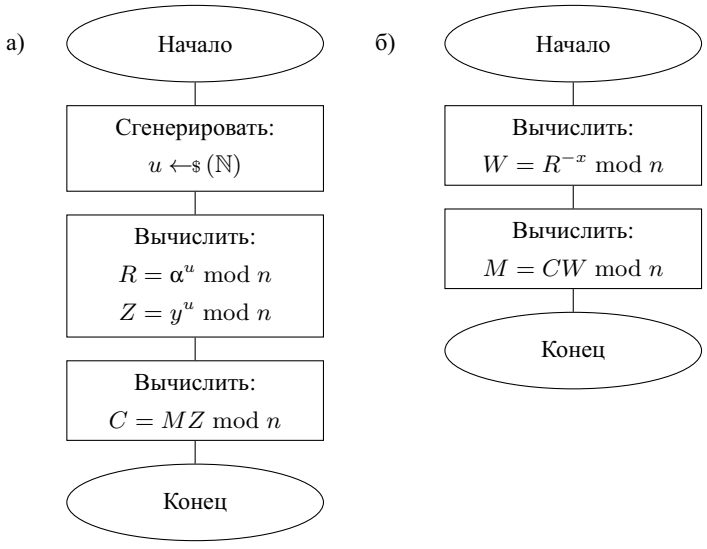

Рисунок 3.8 — Протокол шифрования с открытым ключом: а) шифрование сообщения M; б) расшифрование криптограммы C

Доказательство корректности расшифрования.

$\{CW = MZR^{-x} = My^u(\alpha^u)^{-x} = M\alpha^{xu}\alpha^{-ux} = M(\mathrm{mod} n)\} \Rightarrow$

$\Rightarrow\ CW \bmod n = M.$

Сравнение разработанного протокола шифрования с открытым ключом с аналогами представлено в таблице 8. Мак-Кёрлей [65] предлагал схожие идеи по использованию составного модуля в протоколе шифрования с открытым ключом, однако, основным отличием являются дополнительные ограничения на множители p, q и генератор группы α. Для вычисления модуля n дополнительно генерировались два случайных числа r и s такие, что $2r + 1$ имеет большой простой делитель, $4r + 1$ является простым числом, $8r + 3$ является простым числом, s содержит большой простой делитель, $4s - 1$ является простым числом, $8s - 1$ является простым числом. Тогда $p = 8r + 3$, $q = 8s - 1$ и $n = pq$. В качестве генератора группы, он предложил использовать $\alpha = 16$. А нижнюю границу стойкости протокола устанавливал по ЗФ модуля n, из за чего стойкость ЗДЛ была много ниже требуемого уровня стойкости, это означает, что стойкость протокола фактически зависит только от стойкости ЗФ.

Таблица 8 — Сравнение характеристик предложенного протокола шифрования по открытому ключу с аналогами

Протокол	Коэффициент увеличения криптограммы	Трудоёмкость шифрования (возв. в ст. по мод.)	Трудоёмкость расшифрования (возв. в ст. по мод.)
предложенный	2	2	1
Цисс 2013 [122]	2	3	2
Измаил 2011 [112]	2	3	2
Харн 1994 [69]	1	5	3
Мак-Кёрлей 1988 [65]	2	2	1

3.4 Выводы к третьей главе

1. Разработаны алгоритмы децентрализованной генерации открытых ключей, для использования в протоколах, основанных на ЗДЛ по составному модулю:
    а) алгоритм генерации открытых ключей с использованием простого порядка группы;
    б) алгоритм генерации открытых ключей с использованием простого порядка группы с генерацией системных параметров в удостоверяющем центре;
    в) алгоритм генерации открытых ключей с использованием составного порядка группы.
2. Разработаны протоколы обмена ключами:
    а) с генерацией системных параметров для открытого ключа в удостоверяющем центре;
    б) без генерации системных параметров для ОК в УЦ с использованием простого порядка группы;
    в) без генерации системных параметров для ОК в УЦ с использованием составного порядка группы.
3. Разработан алгоритм шифрования информации с открытым ключом.
4. Разработанные протоколы отличаются использованием ЗДЛ по трудно факторизуемому модулю, благодаря чему и их взлом потребует одновременного решения ЗФ и ЗДЛ.
5. Разработанные протоколы обмена ключами, отличаются использованием дополнительных случайных параметров, благодаря чему ключи в разных сеансах связи одних и тех же абонентов будут различны.

Глава 4. Протоколы электронной цифровой подписи для аутентификации подписывающего и проверки целостности информации в информационно-телекоммуникационных системах

Существующие протоколы ЭЦП, взлом которых требует одновременного решения ЗФ и ЗДЛ, отличаются большими размерами подписи, трудоёмкостью алгоритмов генерации и проверки подписи, громоздкостью уравнений проверки и генерации подписи, затрудняющей анализ протокола.

4.1 Протоколы индивидуальных электронных цифровых подписей

4.1.1 Простая электронная цифровая подпись

Протокол индивидуальной ЭЦП позволяет владельцу секретного ключа подписывать документ, и любой другой субъект сможет проверить целостность документа, идентифицировать и аутентифицировать подписавшего его субъекта, при наличии ОК подписывающего субъекта.

Для генерации системных параметров, ОК и СК пользователя протокола применяются алгоритмы генерации с использованием любого порядка группы ($\gamma - prime$ или $\gamma = \gamma'\gamma''$, где $\gamma' - prime,\ \gamma'' - prime$), описанные в разделе 3.1. Это возможно благодаря тому, что значение γ, которое является секретным в случае использования составного значения порядка, не используется в процедуре проверки подписи.

Пусть пользователь зарегистрировал в УЦ свой ОК, состоящий из тройки параметров (n, α, y). Значение y вычислялось как $y = \alpha^x \bmod n$, где x - секретный ключ пользователя, а $n = pq$. Тогда можно выполнить подписание документа M, аналогично алгоритму ЭЦП Шнорра [129].

1. Сгенерировать случайное число $k \leftarrow_{\$} (\mathbb{N}, |k| = |\gamma|, k < \gamma)$ и вычислить $R = \alpha^k \bmod n$.
2. Используя некоторую специальную хэш-функцию F_H ($|\mathsf{F}_H| = |\gamma|$) вычислить первый элемент подписи: $E = \mathsf{F}_H(R||M)$.
3. Вычислить второй элемент подписи $S = k + xE \bmod \gamma$.
4. Подписью является пара чисел (E, S).

Для проверки необходимо выполнить следующие шаги.

1. Вычислить $\tilde{R} = y^{-E}\alpha^S \bmod n$.

2. Вычислить хэш $\tilde{E} = \mathsf{F}_H(\tilde{R}||M)$.
3. Сравнить значения E и $\tilde{E}$. Если $E = \tilde{E}$, то подпись к сообщению M является подлинной, иначе подпись отвергается.

Основное отличие от оригинальной схемы ЭЦП Шнорра [129] является использование ЗДЛ по трудно факторизуемому модулю. Общая схема протокола представлена на рисунке 4.1, блок схема алгоритмов генерации и проверки ЭЦП - на рисунке 4.2.

Доказательство корректности протокола ЭЦП.

$$\tilde{R} = y^{-E}\alpha^S = \alpha^{-xE}\alpha^{k+xE} = \alpha^k = R \ (\mathrm{mod} n) \Rightarrow$$

$$\left.\begin{array}{l} \tilde{R} = R \\ \Rightarrow \ \tilde{E} = \mathsf{F}_H(\tilde{R}||M) \\ E = \mathsf{F}_H(R||M) \end{array}\right| \Rightarrow \tilde{E} = E.$$

Подписывающий		**Проверяющий**
$y, \alpha, n, x \leftarrow GenK()$	**OK:** (y, α, n) $\longrightarrow$	
..... Подписание		
$k \leftarrow_{\$} (\mathbb{N}, \lvert k\rvert = \lvert\gamma\rvert,$		
$k < \gamma)$		
$R = \alpha^k \bmod n$		
$E = \mathsf{F}_H(R\|\|M)$		
	$M, (E, S)$ $\longrightarrow$	
..... Проверка		
		$\tilde{R} = y^{-E}\alpha^S \bmod n$
		$\tilde{E} = \mathsf{F}_H(\tilde{R}\|\|M)$
		if $E = \tilde{E}$
		then подпись верна
		else подпись отвергнута

Рисунок 4.1 — Протокол индивидуальной ЭЦП, взлом которого требует решения двух независимых вычислительно трудных задач

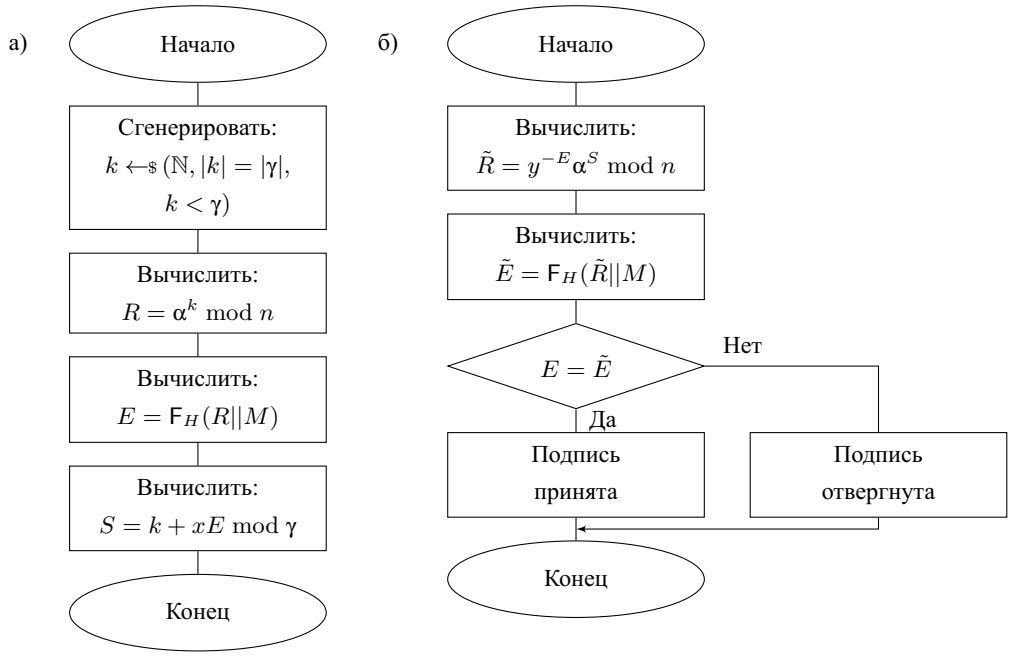

Рисунок 4.2 — Алгоритмы индивидуальной ЭЦП: а) подписание сообщения M; б) проверка подписи

В таблице 9 приведено сравнение предложенной схемы индивидуальной ЭЦП с другими схемами, взлом которых требует одновременного решения ЗФ и ЗДЛ. Отметим, что основной вклад в трудоёмкость вычислений, по сравнению с другими операциями, вносит операция возведения в степень по модулю, поэтому пренебрежём учётом других операций. В некоторых схемах применяется вероятностный алгоритм генерации ЭЦП, этот факт отмечен в таблице 9, и количество операций зависит от генерации параметров, для сравнения был выбран самый быстрый вариант с генерацией параметров ЭЦП с первого раза. ЭЦП Аль-Файоми [114] не стойкая по ЗФ. ЭЦП Верма [113] не стойкая по обеим задачам. Для сравнения были выбраны алгоримы ЭЦП, предложенные за последние года.

Таблица 9 — Сравнение характеристик предложенного протокола индивидуальной ЭЦП с аналогами, взлом которых требует одновременного решения ЗФ и ЗДЛ, для 100 битовой стойкости

Протокол ЭЦП	Размер подписи (бит)	Трудоёмкость генерации ЭЦП (возв. в ст. по мод.)	Трудоёмкость проверки ЭЦП (возв. в ст. по мод.)
предложенный	400	1	2
Чиной 2016 [125]	4096+4096	13 (вер. алг.)	3
Нимбакар 2014 [124]	2048+2048+2048	2	4
Шао 2014 [123]	200+2048	2	5
Мадхур 2012 [121]	2048+2048+2048 +2048+2048	4 (вер. алг.)	5
Молдовян 1 2013 [103]	200+2048	1 (вер. алг.)	3
Молдовян 2 2013 [103]	200+2048	2	3
Молдовян 3 2013 [103]	200+200+200	2	3
Виджей 2012 [120]	2048+2048	2	4
Молдовян 2011 [115]	200+200+200	2	3
Измаил 2011 [110]	2048+2048	3	4
Верма 2011 [113] (не стойк.)	2048+2048+2048	7	5
Аль-Файоми 2010 [114] (не стойк.)	2048	1	3

4.1.2 Слепая электронная цифровая подпись

Впервые слепая ЭЦП была предложена Чаумом в 1983 году [130]. Основное отличие слепой ЭЦП от индивидуальной ЭЦП в том, что подписывающий не знает содержимое подписываемого документа M, подготовленного субъектом, запрашивающим подпись к документу M, из-за этого подписывающий не может установить связь между подписанным документом и его автором (запрашивающим), после подписания документа M. Это достигается за счёт использования в процессе подписания документа специальных маскирующих параметров, которые позволяют скрыть от подписывающего содержимое подписываемого сообщения.

В схеме слепой ЭЦП присутствуют три действующих лица:

1) запрашивающий, который хочет получить подпись к какому либо-документу, чтобы подписывающий не знал его содержание;
2) подписывающий - субъект осуществляющий подписание сообщения запрашивающего;
3) проверяющий - субъект осуществляющий проверку подписи.

Слепая ЭЦП обеспечивает:

1) нулевое разглашение - в процессе подписания подписывающий не может узнать содержание подписываемого сообщения M;
2) неотслеживаемость - после подписания подписывающий не может отследить пару (сообщение M, ЭЦП);
3) неподложность - только подписывающий может сгенерировать ЭЦП.

В основу такой схемы может быть положена схема ЭЦП, описанная в разделе 4.1.1. Для генерации системных параметров, ОК и СК пользователя протокола применяются алгоритмы генерации с использованием простого порядка группы ($\gamma - prime$), описанные в разделах 3.1.1 и 3.1.2. Применение составного значения порядка в данной схеме невозможно из-за вычислений, которые производит запрашивающий при демаскировании подписи.

Для генерации слепой ЭЦП необходимо выполнить следующие шаги.

1. Подписывающий генерирует случайное число $k \leftarrow_{\$} (\mathbb{N}, |k| = |\gamma|, k < \gamma))$ и вычисляет $\bar{R} = \alpha^k \bmod n$.
2. После этого отправляет $\bar{R}$ запрашивающему.

3. Запрашивающий генерирует два случайных числа $\varepsilon \leftarrow_{\$} (\mathbb{N}, |\varepsilon| = |\gamma|, \varepsilon < \gamma)$ и $\mu \leftarrow_{\$} (\mathbb{N}, |\mu| = |\gamma|, \mu \neq \varepsilon, \mu < \gamma)$ (маскирующие экспоненты) и вычисляет $R = \bar{R} y^{\mu} \alpha^{\varepsilon} \bmod n$.
4. Затем используя специальную хэш-функцию F_H ($|\mathsf{F}_H| = |\gamma|$), вычисляет $E = \mathsf{F}_H(R||M)$, $\bar{E} = E + \mu \bmod \gamma$. E - первый элемент подписи к сообщению M, а $\bar{E}$ - первый элемент слепой подписи, который запрашивающий отправляет подписывающему.
5. Подписывающий вычисляет второй элемент слепой подписи $\bar{s} = k + x\bar{E} \bmod \gamma$.
6. Затем отправляет $\bar{s}$ запрашивающему.
7. Запрашивающий демаскирует параметр $\bar{s}$, путём вычисления второго элемента подписи к сообщению M: $S = \bar{s} + \varepsilon \bmod \gamma$.
8. Подписью является пара чисел (E, S).

Для проверки подписи используется алгоритм проверки индивидуальной подписи, описанный в разделе 4.1.1. Общая схема протокола представлена на рисунке 4.3, блок схемы алгоритмов - на рисунке 4.4.

Доказательство корректности протокола слепой ЭЦП.

$$\tilde{R} = y^{-E}\alpha^{S} = \alpha^{-xE}\alpha^{\bar{S}+\varepsilon} = \alpha^{-x(\bar{E}-\mu)}\alpha^{k+x\bar{E}+\varepsilon} =$$
$$= (\alpha^{x})^{\mu}\alpha^{k}\alpha^{\varepsilon} = \tilde{R}y^{\mu}\alpha^{\varepsilon} = R \ (\mathrm{mod} n) \Rightarrow$$
$$\Rightarrow \left.\begin{array}{c} \tilde{R} = R \\ \tilde{E} = \mathsf{F}_H(\tilde{R}||M) \\ E = \mathsf{F}_H(R||M) \end{array}\right| \Rightarrow \tilde{E} = E.$$

Запрашивающий	**Подписывающий**	**Проверяющий**
	OK: (y, α, n) $\longleftarrow$	
	$y, \alpha, n, x \leftarrow GenK()$	
	$\longrightarrow$ **OK:** (y, α, n)	
	Подписывание	
	$k \leftarrow_{\$} (\mathbb{N}, \lvert k\rvert = \lvert\gamma\rvert,$	
	$k < \gamma))$	
	$\bar{R} = \alpha^k \bmod n$	
	$\bar{R}$ $\longleftarrow$	
$\varepsilon \leftarrow_{\$} (\mathbb{N}, \lvert\varepsilon\rvert = \lvert\gamma\rvert,$		
$\varepsilon < \gamma)$		
$\mu \leftarrow_{\$} (\mathbb{N}, \lvert\mu\rvert = \lvert\gamma\rvert,$		
$\mu \neq \varepsilon, \mu < \gamma)$		
$R = \bar{R} y^{\mu} \alpha^{\varepsilon} \bmod n$		
$E = \mathsf{F}_H(R \lvert\rvert M)$		
$\bar{E} = E + \mu \bmod \gamma$		
$\bar{E}$ $\longrightarrow$		
	$\bar{s} = k + x\bar{E} \bmod \gamma$	
$\bar{s}$ $\longleftarrow$		
$S = \bar{s} + \varepsilon \bmod \gamma$		
	Проверка	
	$M, (E, S)$ $\longrightarrow$	
		$\tilde{R} = y^{-E} \alpha^S \bmod n$
		$\tilde{E} = \mathsf{F}_H(\tilde{R} \lvert\rvert M)$
		if $E = \tilde{E}$
		then подпись верна
		else подпись отвергнута

Рисунок 4.3 — Протокол слепой ЭЦП, взлом которого требует решения двух независимых вычислительно трудных задач

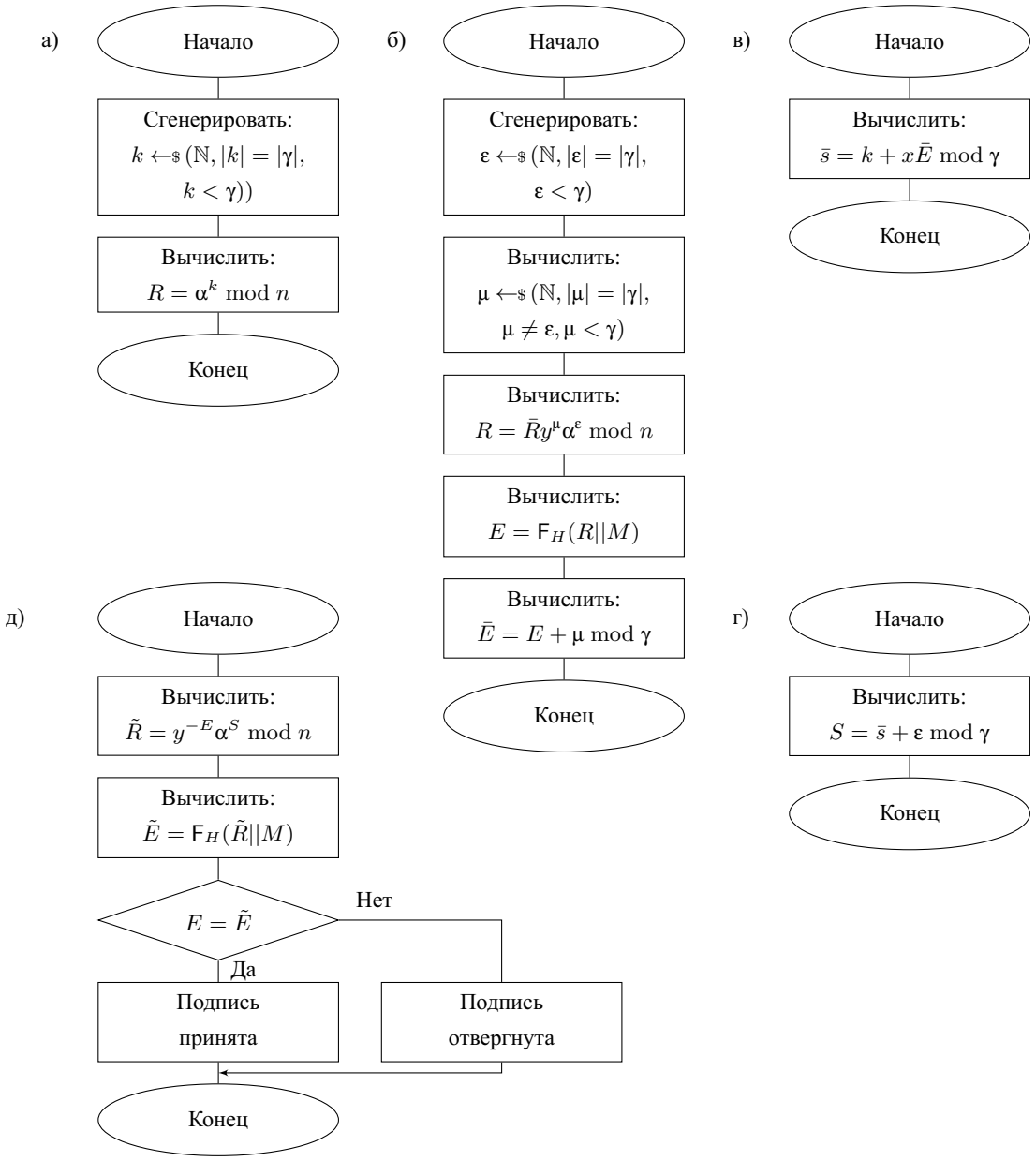

Рисунок 4.4 — Алгоритмы слепой ЭЦП: а) инициализация протокола; б) генерация ослепляющих параметров и вычисление первого элемента подписи; в) подписание ослеплённого параметра; г) - вычисление второго элемента подписи; д) проверка подписи

В таблице 10 приведено сравнение предложенной схемы слепой ЭЦП с другими схемами, взлом которых требует одновременного решения ЗФ и ЗДЛ. Сравнение производится по аналогии со сравнением в разделе 4.1.1. Одна из схем является частично слепой.

Таблица 10 — Сравнение характеристик предложенного протокола слепой ЭЦП с аналогами, взлом которых требует одновременного решения ЗФ и ЗДЛ, для 100 битовой стойкости

Протокол ЭЦП	Размер подписи (бит)	Трудоёмкость генерации ЭЦП (возв. в ст. по мод.)	Трудоёмкость проверки ЭЦП (возв. в ст. по мод.)
предложенный	400	3	2
Молдовян 2012 [102]	200+2048	5	3
Молдовян 2012 [116]	600	5	3
Тахат 2009 [109]	2048+2048	9	4
Тахат 2008 [131] (частично слепая)	2048+2048+2048	5	4

4.2 Протоколы совместных электронных цифровых подписей

4.2.1 Простая коллективная электронная цифровая подпись

Протокол коллективной ЭЦП может быть построен с использованием метода, основанного на совместном формировании общего рандомизирующего параметра подписи, и ранее рассмотренного в работах [132; 133]. В данном методе все вычисления выполняются по одному и тому же модулю [134], поэтому он должен быть общесистемным параметром. Чтобы обеспечить необходимость одновременного решения ЗДЛ и ЗФ для взлома протокола, требуется участие доверительного центра, который генерирует значение системных параметров n, α, γ. Такой алгоритм генерации описан в разделе 3.1.2.

Пусть коллектив подписывающих включает m пользователей, владеющих открытыми ключами $y_i = \alpha^{x_i} \bmod n$, где x_i – секретный ключ i-го пользователя (i = 1, 2, …, m). Данному коллективу подписывающих сопоставляется коллективный открытый ключ y вычисляемый как произведение по модулю n всех

открытых ключей y_i по формуле $y = \prod_{i=1}^{m} y_i \bmod n$. Коллективная подпись к сообщению M формируется следующим образом.

1. Каждый i-ый пользователь (i = 1, 2, ..., m) генерирует случайное число $k_i \leftarrow_{\$} (\mathbb{N}, |k_i| = |\gamma|, k_i < \gamma))$, вычисляет значение $R_i = \alpha^{k_i} \bmod n$ и рассылает R_i всем подписывающим.
2. Пользователи вычисляют значение коллективного рандомизирующего параметра $R = \prod_{i=1}^{m} R_i \bmod n$.
3. Вычисляется значение коллективного открытого ключа $y = \prod_{i=1}^{m} y_i \bmod n$ и первый элемент E коллективной ЭЦП: $E = \mathsf{F}_H(R||M||y)$, где F_H ($|\mathsf{F}_H| = |\gamma|$) некоторая специфицированная хэш-функция.
4. Каждый i-ый подписывающий, используя свой личный секретный ключ и значения k_i и R вычисляет свою долю подписи в виде значения S_i: $S_i = k_i + x_i E \bmod \gamma$.
5. Каждый i-ый подписывающий отправляет остальным свою долю подписи S_i.
6. Вычисляется второй элемент коллективной ЭЦП: $S = \sum_{i=1}^{m} S_i \bmod \gamma = \sum_{i=1}^{m} k_i + E \sum_{i=1}^{m} x_i \bmod \gamma$.

Коллективной подписью является пара чисел (E, S). Процедура проверки коллективной ЭЦП к документу M осуществляется следующим образом.

1. Вычисляется значение коллективного открытого ключа $y = \prod_{i=1}^{m} y_i \bmod n$.
2. Вычисляются значения $\tilde{R} = y^{-E}\alpha^{S} \bmod n$ и $\tilde{E} = \mathsf{F}_H(\tilde{R}||M||y)$.
3. Сравниваются значения $\tilde{E}$ и E. Если $\tilde{E} = E$, то ЭЦП признается подлинной.

Схема генерации коллективного открытого ключа, которая будет использована далее в протоколах слепой коллективной ЭЦП и утверждаемой групповой ЭЦП, представлена на рисунке 4.5. Общая схема протокола коллективной ЭЦП представлена на рисунке 4.6, блок схемы алгоритмов - на рисунке 4.7. Алгоритм генерации системных параметров (α, n, γ) обозначим $Gen()$, алгоритм генерации пары ОК и СК на основе системных параметров (α, n, γ) обозначим $GenK(a,n,\gamma)$.

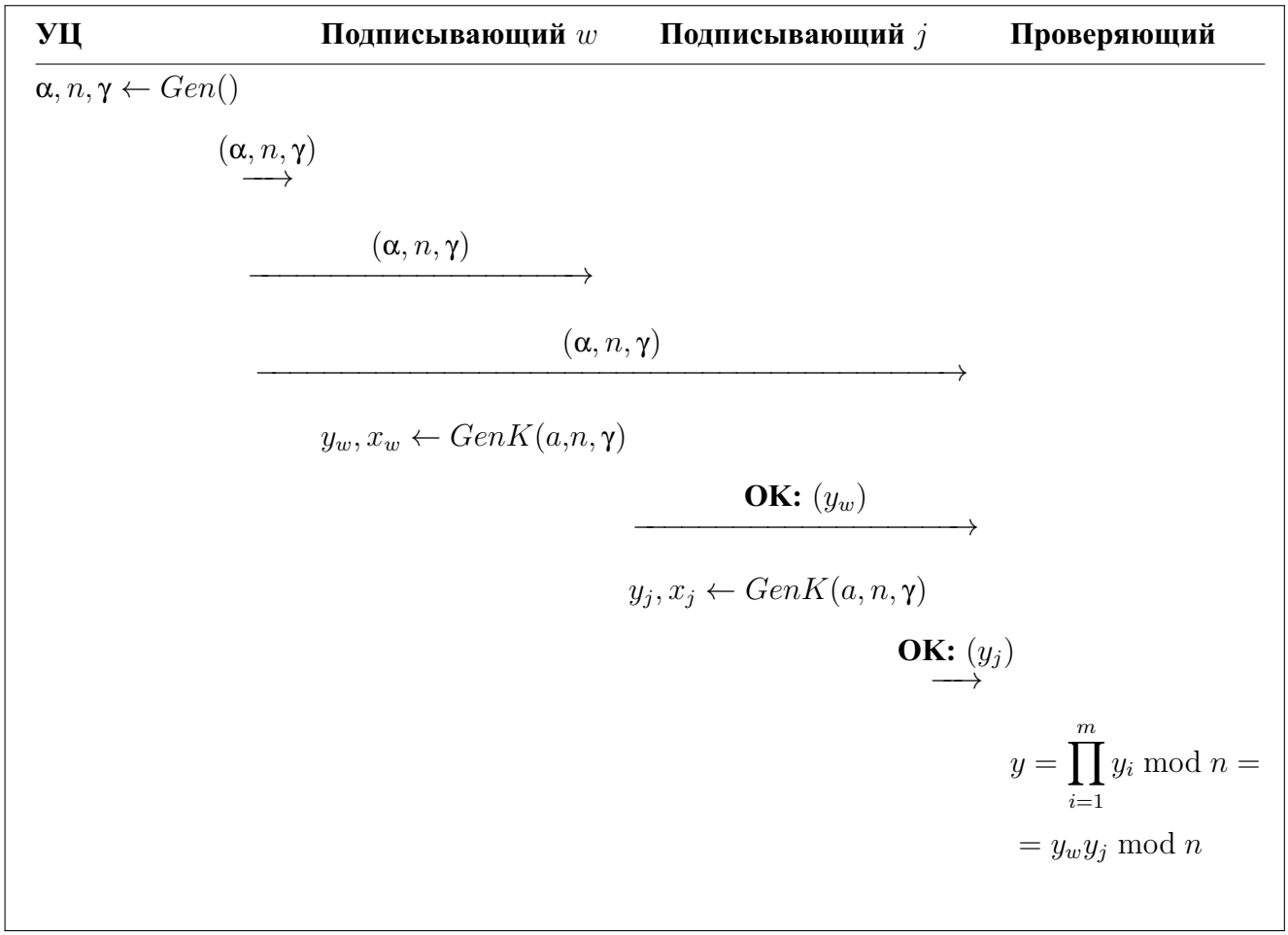

Рисунок 4.5 — Генерация общего открытого ключа в протоколах коллективных ЭЦП, для двух участников (m = 2)

Подписывающий w		**Подписывающий** j	**Проверяющий**
..... Подписывание			
$k_w \leftarrow_{\$} (\mathbb{N}, \lvert k_w\rvert = \lvert\gamma\rvert,$		$k_j \leftarrow_{\$} (\mathbb{N}, \lvert k_j\rvert = \lvert\gamma\rvert,$	
$k_w < \gamma))$		$k_j < \gamma))$	
$R_w = \alpha^{k_w} \bmod n$		$R_j = \alpha^{k_j} \bmod n$	
	$\xrightarrow{R_w}$		
	$\xleftarrow{R_j}$		
$R = \prod_{i=1}^{m} R_i \bmod n =$		$R = \prod_{i=1}^{m} R_i \bmod n =$	
$= R_j R_w \bmod n$		$= R_w R_j \bmod n$	
$y = \prod_{i=1}^{m} y_i \bmod n =$		$y = \prod_{i=1}^{m} y_i \bmod n =$	
$= y_w y_j \bmod n$		$= y_j y_w \bmod n$	
$E = \mathsf{F}_H(R\|\|M\|\|y)$		$E = \mathsf{F}_H(R\|\|M\|\|y)$	
$S_w = k_w + x_w E \bmod \gamma$		$S_j = k_j + x_j E \bmod \gamma$	
	$\xrightarrow{S_w}$		
	$\xleftarrow{S_j}$		
$S = \sum_{i=1}^{m} S_i \bmod \gamma$		$S = \sum_{i=1}^{m} S_i \bmod \gamma$	
$= S_w + S_j \bmod \gamma$		$= S_j + S_w \bmod \gamma$	
Подпись (E, S)	$=$	Подпись (E, S)	
..... Проверка			
	$\xrightarrow{M, (E,S)}$		$y = \prod_{i=1}^{m} y_i \bmod n$
			$= y_w y_j \bmod n$
			$\tilde{R} = y^{-E} \alpha^{S} \bmod n$
			$\tilde{E} = \mathsf{F}_H(\tilde{R}\|\|M\|\|y)$
			if $E = \tilde{E}$
			then подпись верна
			else подпись отвергнута

Рисунок 4.6 — Протокол коллективной ЭЦП, для двух участников (m = 2)

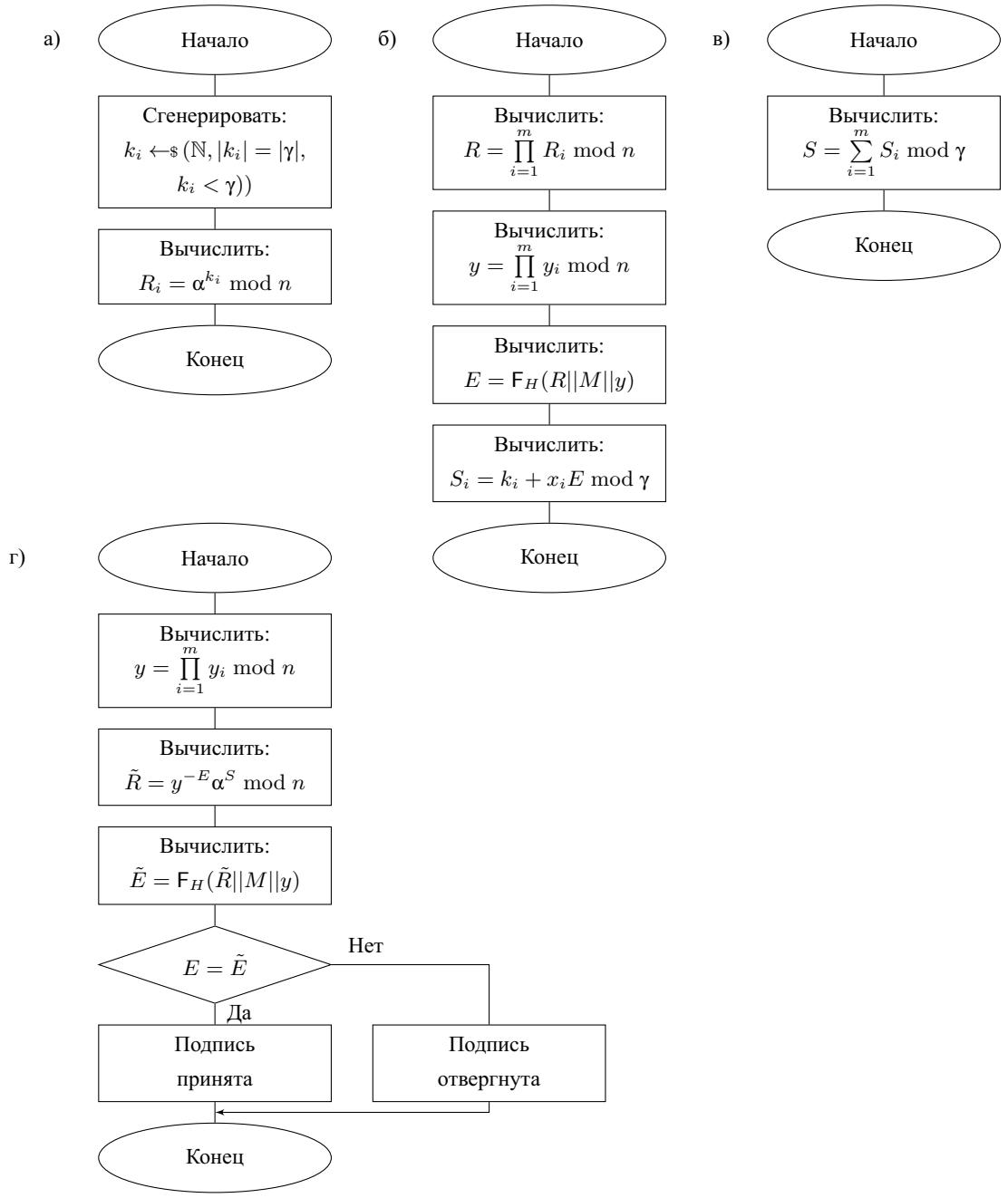

Рисунок 4.7 — Алгоритмы коллективной ЭЦП: а) генерация рандомизирующих параметров; б) вычисление первого элемента подписи E и вычисление индивидуальной части второго элемента подписи S_i; в) вычисление второго элемента подписи S; г) проверка коллективной ЭЦП

Доказательство корректности протокола коллективной ЭЦП.

$$\tilde{R} = y^{-E}\alpha^S = \left(\prod_{i=1}^{m} y_i\right)^{-E} \alpha^{\sum\limits_{i=1}^{m} S_i} = \left(\alpha^{\sum\limits_{i=1}^{m} x_i}\right)^{-E} \alpha^{\sum\limits_{i=1}^{m} S_i} =$$

$$= \alpha^{-E\sum\limits_{i=1}^{m} x_i} \alpha^{\sum\limits_{i=1}^{m} k_i + E\sum\limits_{i=1}^{m} x_i} = \alpha^{-E\sum\limits_{i=1}^{m} x_i} \alpha^{E\sum\limits_{i=1}^{m} x_i} \alpha^{\sum\limits_{i=1}^{m} k_i} =$$

$$= \alpha^{\sum\limits_{i=1}^{m} k_i} = \prod_{i=1}^{m} \alpha^{k_i} = \prod_{i=1}^{m} R_i = R \,(\mathrm{mod} n) \;\Rightarrow\; \tilde{R} = R \;\Rightarrow\; \tilde{E} = E.$$

У предложенного протокола коллективной ЭЦП существует единственный аналог, описанный в работе [117], взлом которого требует одновременного решения ЗФ и ЗДЛ по простому модулю. Он отличается использованием ЗДЛ по двойному основанию, требует две операции модульного возведения в степень при генерации подписи, формируемая ЭЦП в 1,5 раза больше по сравнению с предложенным протоколом коллективной ЭЦП, проверка подписи требует три операции модульного возведения в степень. Сравнение предложенного протокола и аналога приведено в таблице 11.

Таблица 11 — Сравнение характеристик предложенного протокола коллективной ЭЦП с аналогом, взлом которого требует одновременного решения ЗФ и ЗДЛ, для 100 битовой стойкости

Протокол ЭЦП	Размер подписи (бит)	Трудоёмкость генерации ЭЦП (возв. в ст. по мод.)	Трудоёмкость проверки ЭЦП (возв. в ст. по мод.)
предложенный	400	1	2
Латышев 2013 [117]	600	2	3

4.2.2 Слепая коллективная электронная цифровая подпись

Протокол слепой коллективной подписи на основе трудности одновременного решения ЗФ и ЗДЛ может быть построен путем объединения рассмотренных ранее протоколов слепой (раздел 4.1.2) и коллективной подписи (раздел 4.2.1) по аналогии со схемой построения, использованной в работе [135]. При этом также как и в протоколе коллективной подписи, системные параметры n, α, γ должны генерироваться доверительным центром (алгоритм генерации описан в разделе 3.1.2).

Пусть коллектив подписывающих включает m пользователей, владеющих открытыми ключами $y_i = \alpha^{x_i} \bmod n$, где x_i – секретный ключ i-го пользователя (i = 1, 2, …, m). Слепая коллективная подпись к сообщению M формируется следующим образом.

1. Каждый i-ый подписывающий (i = 1, 2, …, m) генерирует случайное число $k_i \leftarrow_{\$} (\mathbb{N}, |k_i| = |\gamma|, k_i < \gamma))$, вычисляет значение $\bar{R}_i = \alpha^{k_i} \bmod n$ и отправляет $\bar{R}_i$ запрашивающему.
2. Запрашивающий вычисляет значение коллективного рандомизирующего параметра $\bar{R} = \prod_{i=1}^{m} \bar{R}_i \bmod n$.
3. Вычисляется значение коллективного открытого ключа $y = \prod_{i=1}^{m} y_i \bmod n$.
4. Запрашивающий генерирует два случайных числа $\varepsilon \leftarrow_{\$} (\mathbb{N}, |\varepsilon| = |\gamma|, \varepsilon < \gamma)$ и $\mu \leftarrow_{\$} (\mathbb{N}, |\mu| = |\gamma|, \mu \neq \varepsilon, \mu < \gamma)$ (которые называются «маскирующими» параметрами) и вычисляет $R = \bar{R}\, y^{\mu} \alpha^{\varepsilon} \bmod n$. Затем используя специфицированную хэш-функцию F_H ($|\mathsf{F}_H| = |\gamma|$), он вычисляет $E = \mathsf{F}_H(R||M||y)$, и $\bar{E} = E + \mu \bmod \gamma$. Значение E – первый элемент подписи к документу M, а является первым элементом слепой подписи. Запрашивающий отправляет $\bar{E}$ подписывающим.
5. Каждый i-ый подписывающий, используя свой личный секретный ключ x_i и значения k_i и R вычисляет свою долю подписи в виде значения $\bar{S}_i$: $\bar{S}_i = k_i + x_i E \bmod \gamma$, и отправляет запрашивающему.
6. Запрашивающий вычисляет второй элемент слепой коллективной ЭЦП: $\bar{S} = \sum_{i=1}^{m} \bar{S}_i \bmod \gamma = \sum_{i=1}^{m} k_i + E \sum_{i=1}^{m} x_i \bmod \gamma$.
7. Запрашивающий “демаскирует” параметр $\bar{S}$, то есть вычисляет второй элемент подписи (E, S) к сообщению M: $S = \bar{s} + \varepsilon \bmod \gamma$.

Коллективной подписью является пара чисел (E, S). Процедура проверки слепой коллективной ЭЦП к документу M осуществляется следующим образом.

1. Вычисляется значение коллективного открытого ключа $y = \prod_{i=1}^{m} y_i \bmod n$.
2. Вычисляются значения $\tilde{R} = y^{-E} \alpha^{S} \bmod n$ и $\tilde{E} = \mathsf{F}_H(\tilde{R}||M||y)$.
3. Сравниваются значения $\tilde{E}$ и E. Если $\tilde{E} = E$, то ЭЦП признается подлинной.

Общая схема протокола представлена на рисунках 4.8 и 4.9, блок схемы алгоритмов - на рисунке 4.10.

Запрашивающицй		**Подписывающий** w	**Подписывающий** j
..... Подписывание			
		$k_w \leftarrow_{\$} (\mathbb{N}, \lvert k_w\rvert = \lvert\gamma\rvert,$	$k_j \leftarrow_{\$} (\mathbb{N}, \lvert k_j\rvert = \lvert\gamma\rvert,$
		$k_w < \gamma))$	$k_j < \gamma))$
		$\bar{R}_w = \alpha^{k_w} \bmod n$	$\bar{R}_j = \alpha^{k_j} \bmod n$
	$\xleftarrow{\bar{R}_w}$		
	$\xleftarrow{\qquad\qquad\bar{R}_j\qquad\qquad}$		
$\bar{R} = \prod_{i=1}^{m} \bar{R}_i \bmod n =$			
$= \bar{R}_w \bar{R}_j \bmod n$			
$\varepsilon \leftarrow_{\$} (\mathbb{N}, \lvert\varepsilon\rvert = \lvert\gamma\rvert,$			
$\varepsilon < \gamma)$			
$\mu \leftarrow_{\$} (\mathbb{N}, \lvert\mu\rvert = \lvert\gamma\rvert,$			
$\mu \neq \varepsilon, \mu < \gamma)$			
$R = \bar{R} y^{\mu} \alpha^{\varepsilon} \bmod n$			
$y = \prod_{i=1}^{m} y_i \bmod n =$			
$= y_w y_j \bmod n$			
$E = \mathsf{F}_H(R\|\|M\|\|y)$	$\xrightarrow{\bar{E}}$		
	$\xrightarrow{\qquad\qquad\bar{E}\qquad\qquad}$		
		$\bar{S_w} = k_w + x_w E \bmod \gamma$	$\bar{S}_j = k_j + x_j E \bmod \gamma$
	$\xleftarrow{S_w}$		
	$\xleftarrow{\qquad\qquad S_j\qquad\qquad}$		
$\bar{S} = \sum_{i=1}^{m} \bar{S}_i =$			
$= S_w + S_j \bmod n$			
Подпись (E, S)			

Рисунок 4.8 — Протокол слепой коллективной ЭЦП, для двух участников ($m = 2$)

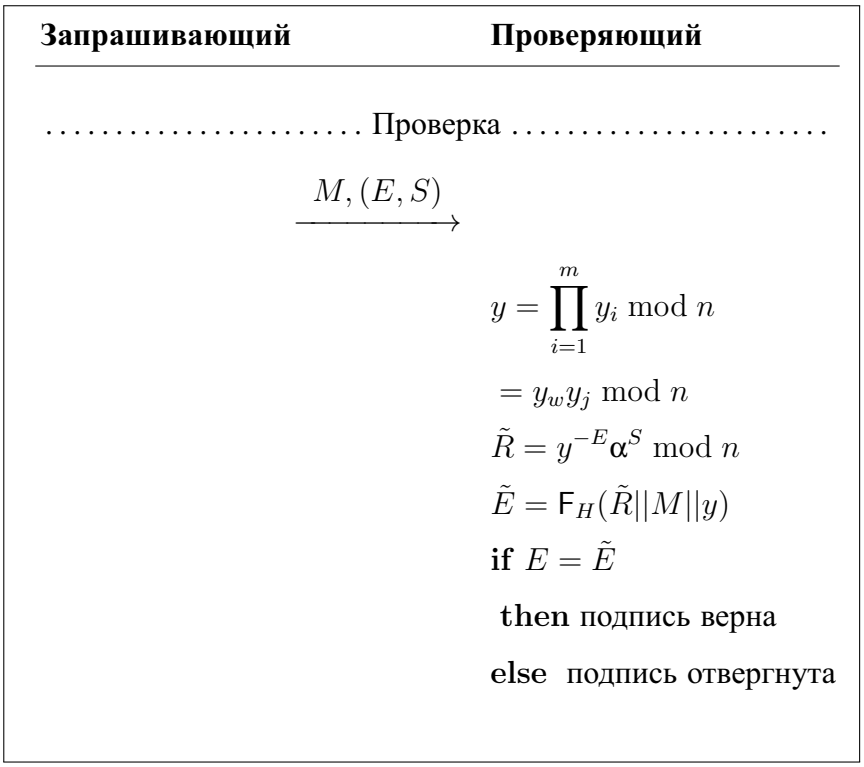

Рисунок 4.9 — Протокол проверки слепой коллективной ЭЦП, для двух участников (m = 2)

Доказательство корректности протокола слепой ЭЦП.

$$\tilde{R} = y^{-E}\alpha^S = \left(\prod_{i=1}^{m} y_i\right)^{-E}\alpha^S = \left(\prod_{i=1}^{m}\alpha^{x_i}\right)^{-E}\alpha^S = \prod_{i=1}^{m}\alpha^{-Ex_i}\alpha^{\bar{S}+\varepsilon} =$$

$$= \prod_{i=1}^{m}\alpha^{-Ex_i}\alpha^{\sum\limits_{i=1}^{m}\bar{S}_i+\varepsilon} = \alpha^{-(\bar{E}-\mu)\sum\limits_{i=1}^{m}x_i}\alpha^{\sum\limits_{i=1}^{m}k_i+E\sum\limits_{i=1}^{m}x_i+\varepsilon} =$$

$$= \alpha^{-\bar{E}\sum\limits_{i=1}^{m}x_i+E\sum\limits_{i=1}^{m}x_i+\mu\sum\limits_{i=1}^{m}x_i}\alpha^{\sum\limits_{i=1}^{m}k_i}\alpha^{\varepsilon} = \left(\alpha^{\sum\limits_{i=1}^{m}x_i}\right)^{\mu}\alpha^{\sum\limits_{i=1}^{m}k_i}\alpha^{\varepsilon} =$$

$$= y^{\mu}\prod_{i=1}^{m}\bar{R}_i\alpha^{\varepsilon} = \bar{R}y^{\mu}\alpha^{\varepsilon} = R(\mathrm{mod}n)$$

$$\Rightarrow \left\{\tilde{R} = R;\ \tilde{E} = F_H(\tilde{R}||M||y)\ ;\ E = F_H\left(R||M||y\right)\right\} \Rightarrow \tilde{E} = E.$$

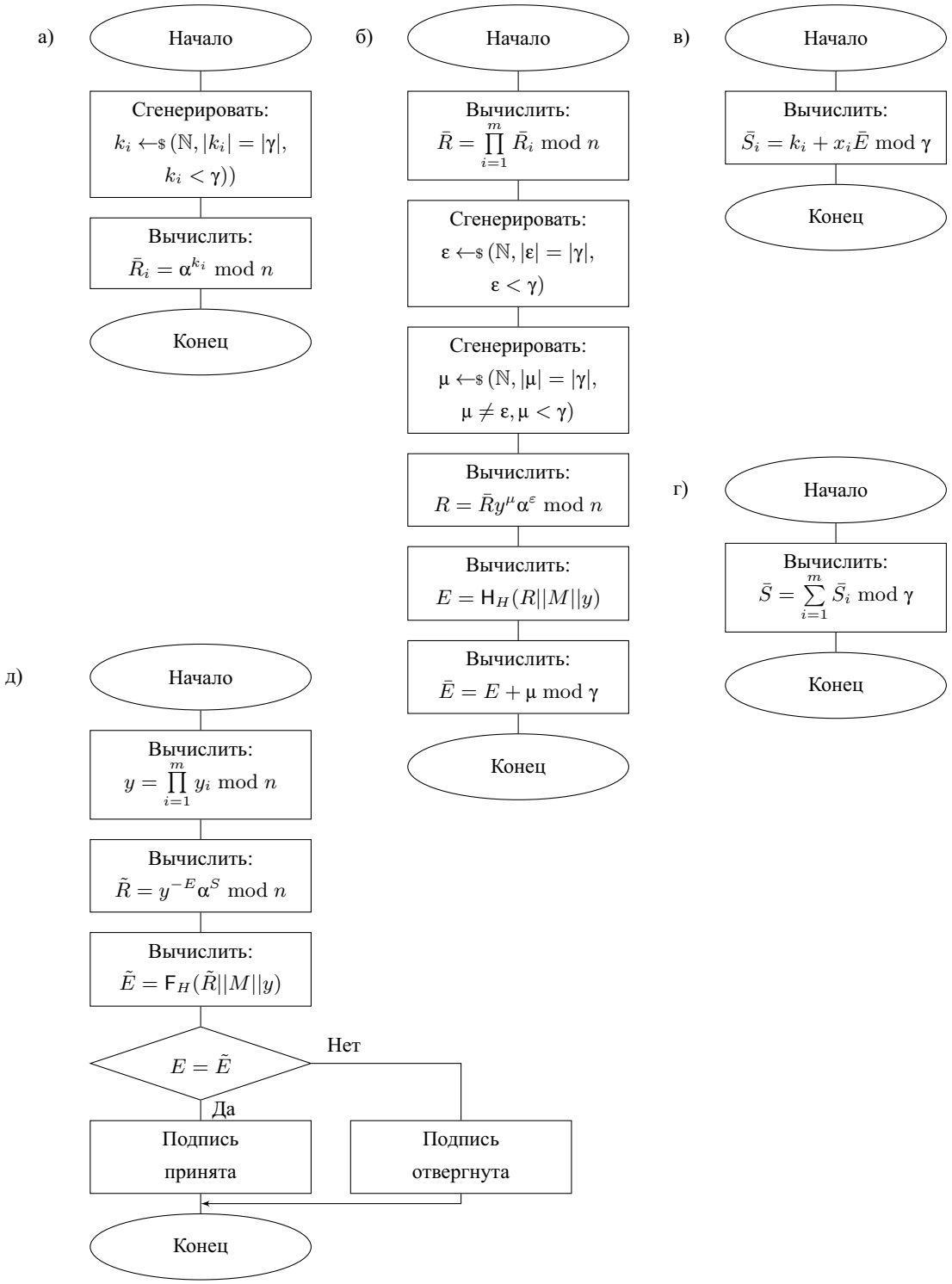

Рисунок 4.10 — Алгоритмы слепой коллективной ЭЦП: а) генерация рандомизирующих параметров; б) генерация ослепляющих параметров и первого элемента подписи E; в) вычисление индивидуальной части второго элемента подписи $\bar{S}_i$; г) вычисление второго элемента подписи S; д) проверка слепой коллективной ЭЦП

4.2.3 Утверждаемая групповая электронная цифровая подпись

Наиболее близким к практике подготовки, визирования и утверждения бумажных документов в организации является протокол утверждаемой групповой ЭЦП (УГЭЦП). Данный протокол обеспечивает формирование ЭЦП руководителем и подмножеством работников от имени некоторого коллегиального органа. В протоколах данного типа руководитель распределяет работу по подготовке электронного документа, после подготовки, исполнители формируют свою групповую предподпись к электронному документу и отправляют документ на утверждение руководителю. После утверждения руководитель формирует ЭЦП коллегиального органа на основе своего секретного ключи и групповой предподписи. Причём, разглашения личных секретных ключей исполнителей не происходит, и идентифицировать лиц, которые подписали данный электронный документ, может только руководитель. Впервые протокол УГЭЦП был предложен в работе [136]. В протоколе использовалась вычислительная сложность решения ЗДЛ и ЗФ, однако для компрометации протокола достаточно было решить одну из указанных задач, таким образом, он не обеспечивает повышения интегрального показателя безопасности.

Выделим основные отличия УГЭЦП от коллективной ЭЦП:

1) сотрудники при формировании своей части УГЭЦП (предподпись) используют свои личные секретные ключи, которые неизвестны руководителю;
2) готовить групповую предподпись может любое подмножество сотрудников коллегиального органа, от имени которого подписывается документ;
3) только руководитель преобразует предподпись в конечную УГЭЦП;
4) по значению УГЭЦП и документу, к которому она сгенерирована, только руководитель может определить перечень лиц, которые сформировали предподпись к документу, то есть внешние лица не могут установить кто из сотрудников участвовал в разработке и утверждении данного документа.

Для построения протокола УГЭЦП воспользуемся механизмами предложенными в работе [137].

Для генерации системных параметров n, α, γ можно использовать алгоритмы, описанные в разделах 3.1.1 и 3.1.2.

Генерация ОК коллегиального органа.

1. Руководитель или УЦ генерирует системные параметры протокола (n, α, γ) $\leftarrow Gen()$.
2. Руководитель генерирует личный СК $x_0 \leftarrow_{\$} (\mathbb{N}, x_0 < \gamma, |x_0| \approx |\gamma|)$.
3. Затем вычисляет элемент ОК коллегиального органа $y_0 = \alpha^{x_0} \bmod n$.
4. Отправляет значения (n, α, γ) подчинённым.

Генерация ОК подчинённых.

1. Каждый i-тый подчинённый ($i = 1,2,3,...$) генерирует личный СК $x_i \leftarrow_{\$} (\mathbb{N}, x_i < \gamma, |x_i| \approx |\gamma|)$.
2. Затем вычисляет личный ОК $y_i = \alpha_i^{x_i} \bmod n$.

Формирование групповой предподписи к документу M.

1. Руководитель вычисляет $H = \mathsf{F}_H(M)$, где F_H некоторая стойкая хэш-функция ($|\mathsf{F}_H| = |\gamma|$), $\lambda_i = \mathsf{F}_H(H||y_i||\mathsf{F}_H(H||y_i||x_0))$, где y_i - ОК i-го подчинённого. Отправляет подчинённому i значение λ_i, зашифровав его с использованием открытого ключа подчинённого (например, с использованием алгоритма шифрования по ОК, описанного в разделе 3.3). Значение λ_i руководитель генерирует для всех m подчинённых, ответственных за разработку документа M.
2. Затем он вычисляет первый элемент ЭЦП $U = y_0^{\lambda_0 - 1} \prod_{i=1}^{m} y_i^{\lambda_i} \bmod n$, где $\lambda_0 = \mathsf{F}_H(H||y_o||\mathsf{F}_H(H||y_o||x_0))$.
3. Каждый i-тый подчинённый ($i = 1,2,3,...m$) генерирует случайное значение $t_i \leftarrow_{\$} (\mathbb{N},|t_i| \approx |\gamma|, t_i < \gamma)$, вычисляет $R_i = \alpha^{t_i} \bmod n$ и отправляет значение R_i руководителю.
4. Руководитель генерирует случайное $t_0 \leftarrow_{\$} (\mathbb{N},|t_0| \approx |\gamma|, t_0 < \gamma)$ и вычисляет $R_0 = \alpha^{t_0} \bmod n$. Вычисляет $R = R_0 \prod_{i=1}^{m} R_i \bmod n$ и формирует второй элемент ЭЦП по формуле $E = \mathsf{F}_H(M||R||U)$. Значение E рассылает каждому подчинённому.
5. Каждый i-тый подчинённый ($i = 1,2,3,...m$) вычисляет свою долю подписи $S_i = t_i + x_i \lambda_i E \mod \gamma$ и отправляет значение S_i руководителю.
6. Руководитель проверяет корректность вычисления всех долей подписи $R_i = y_i^{-\lambda_i E} \alpha^{S_i} \bmod n$. Если все доли подписи вычислены корректно, то

он вычисляет $S = t_0 + x_0 E \bmod \gamma$ и утверждает ЭЦП путём вычисления третьего элемента ЭЦП по формуле $S = \sum_{i=0}^{m} S_i \bmod \gamma$.

Подписью к документу M является тройка чисел (U, E, S).

Проверка подписи осуществляется следующим образом.

1. Проверяющий с использованием ОК коллегиального органа (y_o, n, α) и подписи (U, E, S) к документу M вычисляет $\tilde{R} = (U y_o)^{-E} \alpha^S \bmod n$.
2. Затем он вычисляет $\tilde{E} = \mathsf{F}_H(M||\tilde{R}||U)$.
3. Если $E = \tilde{E}$, то подпись признаётся верной, иначе подпись отвергается.

Определение руководителем лиц, которые подписали данный документ.

1. Руководитель вычисляет $\lambda_i = \mathsf{F}_H(H||y_i||\mathsf{F}_H(H||y_i||x_0))$.
2. Затем перебирая все возможные подмножества подписывающих ищет $\tilde{U} = y_0^{\lambda_0 - 1} \prod_{i=1}^{m} y_i^{\lambda_i} \bmod n$, до тех пор пока $\tilde{U} \neq U$.
3. Если $U = \tilde{U}$, то руководитель определил лиц, подписавших этот документ.

Доказательство стороннему проверяющему, что именно эти лица подписывали документ.

1. Руководитель предъявляет набор значений y_i, $\mathsf{F}_H(H||y_i||x_0)$ лиц, которые подписывали данный документ.
2. Проверяющий вычисляет $\lambda_i = \mathsf{F}_H(H||y_i||\mathsf{F}_H(H||y_i||x_0))$ для каждой пары предъявленных значений.
3. Затем вычисляет $\tilde{U} = y_0^{\lambda_0 - 1} \prod_{i=1}^{m} y_i^{\lambda_i} \bmod n$
4. Если $U = \tilde{U}$, то действительно документ подписали лица, чьи открытые ключи y_i предоставил руководитель.

Вычисление рандомизирующих экспонент по формуле $\lambda_i = \mathsf{F}_H(H||y_i||\mathsf{F}_H(H||y_i||x_0))$ с использованием двойного хэширования $\mathsf{F}_H()$ позволяет руководителю доказывать проверяющему, что именно эти лица подписали документ, без разглашения своего секретного ключа x_0. Общая схема протокола УГЭЦП представлена на рисунке 4.11, протокол проверки УГЭЦП представлен на рисунке 4.12.

Доказательство корректности протокола УГЭЦП.

$$\tilde{R} = (U y_o)^{-E} \alpha^S = (\prod_{i=0}^{m} y_i^{\lambda_i})^{-E} \alpha^{\sum\limits_{i=0}^{m} S_i} =$$
$$= \alpha^{\sum\limits_{i=0}^{m} -E x_i \lambda_i} \alpha^{\sum\limits_{i=0}^{m} t_i + x_i \lambda_i E} = \alpha^{\sum\limits_{i=0}^{m} t_i} = R_0 \prod_{i=1}^{m} R_i = R \ (\mathrm{mod} n)$$

Руководитель		**Подписывающий** w	**Подписывающий** j
$n, \alpha, \gamma \leftarrow Gen()$			
	$\xrightarrow{(n,\alpha,\gamma)}$		
	$\xrightarrow{(n,\alpha,\gamma)}$		
OK: $y_o = \alpha^{x_0} \bmod n$		**OK:** $y_w = \alpha^{x_w} \bmod n$	**OK:** $y_j = \alpha^{x_j} \bmod n$
	$\xleftarrow{y_w}$		
	$\xleftarrow{y_j}$		
..... Подписывание			
$H = \mathsf{F}_H(M)$			
$\lambda_i = \mathsf{F}_H(H\|\|y_i\|\|$			
$\|\|\mathsf{F}_H(H\|\|y_i\|\|x_0))$	$\xrightarrow{\lambda_w}$		
	$\xrightarrow{\lambda_j}$		
$U = y_0^{\lambda_0 - 1} \prod_{i=1}^{m} y_i^{\lambda_i} \bmod n =$			
$= y_0^{\lambda_0 - 1} y_w^{\lambda_w} y_j^{\lambda_j} \bmod n$			
$t_0 \leftarrow_\$ (\mathbb{N}, \|t_0\| \approx \|\gamma\|,$		$t_w \leftarrow_\$ (\mathbb{N}, \|t_w\| \approx \|\gamma\|,$	$t_j \leftarrow_\$ (\mathbb{N}, \|t_j\| \approx \|\gamma\|,$
$t_0 < \gamma)$		$t_w < \gamma)$	$t_j < \gamma)$
$R_0 = \alpha^{t_0} \bmod n$		$R_w = \alpha^{t_w} \bmod n$	$R_j = \alpha^{t_j} \bmod n$
	$\xleftarrow{R_w}$		
	$\xleftarrow{R_j}$		
$R = R_0 \prod_{i=1}^{m} R_i \bmod n =$			
$= R_0 R_w R_j \bmod n$			
$E = \mathsf{F}_H(M\|\|R\|\|U)$	$\xrightarrow{E}$		
	$\xrightarrow{E}$		
	$\xleftarrow{S_w}$	$S_w = t_w + x_w \lambda_w E \mod \gamma$	$S_j = t_j + x_j \lambda_j E \mod \gamma$
	$\xleftarrow{S_j}$		
if $R_i = y_i^{-\lambda_i E} \alpha^{S_i} \bmod n$			
then END			
else			
$S = t_0 + x_0 \lambda_0 E \bmod \gamma$			
$S = \sum_{i=0}^{m} S_i \bmod \gamma =$			
$= S_0 + S_w + S_j \bmod \gamma$			
Подпись (U, E, S)			

Рисунок 4.11 — Протокол утверждаемой групповой ЭЦП, для двух участников ($m = 2$) и руководителя

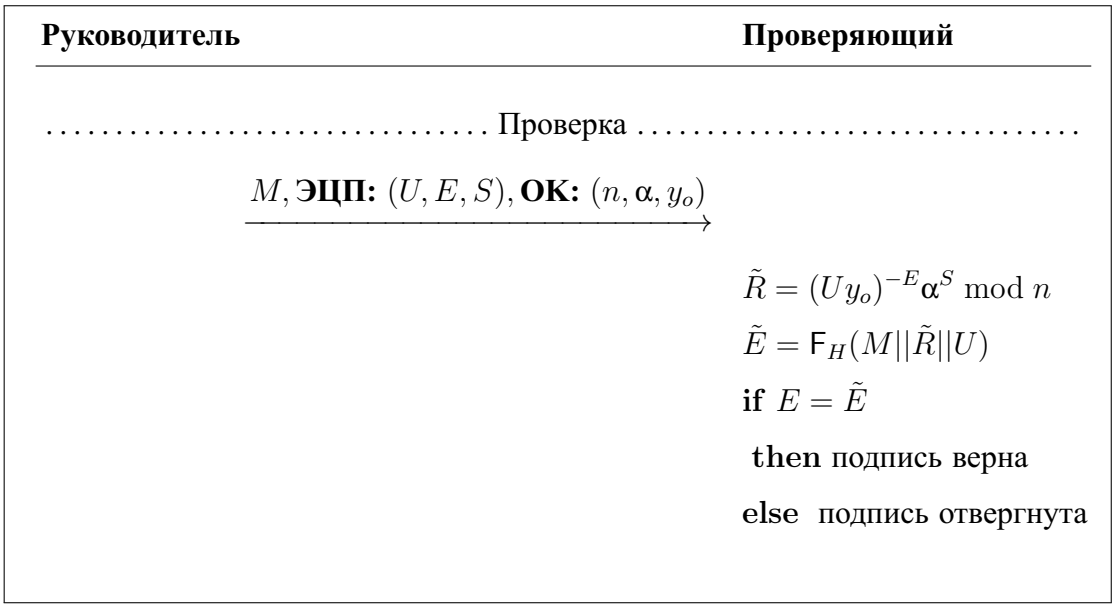

Рисунок 4.12 — Протокол проверки утверждаемой групповой ЭЦП

4.3 Выводы к четвёртой главе

1. Разработаны протоколы индивидуальных ЭЦП, взлом которых требует одновременного решения двух вычислительно трудных задач факторизации и дискретного логарифмирования:
    а) протокол простой индивидуальной ЭЦП;
    б) протокол слепой индивидуальной ЭЦП.
2. Разработаны протоколы совместных ЭЦП, взлом которых требует одновременного решения двух вычислительно трудных задач факторизации и дискретного логарифмирования:
    а) протокол коллективной ЭЦП;
    б) протокол слепой коллективной ЭЦП;
    в) протокол утверждаемой групповой ЭЦП.
3. Размер подписи в разработанных протоколах совместных ЭЦП не зависит от количества участников протокола.
4. Формирование частей подписи, выполняемых участниками в разработанных протоколах совместных ЭЦП, происходит параллельно, благодаря чему длительность вычисления подписи не зависит от количества участников протокола.
5. В протоколе утверждаемой групповой ЭЦП только руководитель сможет определить и доказать стороннему проверяющему, что именно данное подмножество работников подписывали этот документ.

Глава 5. Специальные алгоритмы для защиты информации в информационно-телекоммуникационных системах

5.1 Протоколы аутентификации удалённых пользователей информационно-телекоммуникационных систем с 0 разглашением

Для выполнения процедуры аутентификации удалённых пользователей во многих практических приложениях предпочтительными для использования являются протоколы строгой аутентификации, которые характеризуются тем, что в ходе выполнения протокола не происходит явного разглашения секрета. Из этого класса протоколов отдельное место занимают протоколы с нулевым разглашением секрета, в которых подлинность пользователей проверяется по их открытым ключам [9; 12; 138]. К ним относятся такие протоколы, для которых можно доказать, что в ходе выполнения протокола никакой утечки информации о секрете не происходит. В протоколах с 0 разглашением две роли: проверяющий и доказывающий. Проверяющий осуществляет проверку доказывающего на предмет обладания СК, связанным с проверяемым ОК. Доказывающий подтверждает проверяющему, что он действительно знает СК, связанный с проверяемым ОК.

5.1.1 Интерактивный протокол аутентификации субъекта

В основу итеративного протокола положено умение владельца ОК определять принадлежность случайных чисел подгруппе, генерируемой всевозможными натуральными степенями числа α по составному модулю n.

Рассмотрим построение протокола с нулевым разглашением секрета, основанного на вычислительной сложности одновременного решения ЗФ и ЗДЛ по простому модулю и использующего ОК вида (n, α), где $n = pq$ и α число, порядок которого по модулю n равен γ (числа p, q и γ являются СК). Для генерации системных параметров и пары ОК, СК можно использовать алгоритмы, описанные в разделах 2.4 и 3.1.

Только владелец ОК (доказывающий) может точно определять принадлежность случайных чисел подгруппе генерируемой всевозможными натуральными степенями числа α по модулю n. Аутентификация владельца ОК реализуется

итеративным протоколом, в котором многократно выполняется следующий двухшаговый раунд:

1. Проверяющий генерирует случайные числа $k \leftarrow_{\$} (\mathbb{N}, |k| = |\gamma|, k < \gamma)$, $R \leftarrow_{\$} (\mathbb{N}, |R| < |n|)$ и случайный бит $e \leftarrow_{\$} \{0,1\}$. Если $e = 0$, то он направляет запрос $U = R$. Иначе, он вычисляет значение $U = \alpha^k \bmod n$ и направляет значение U доказывающему в качестве запроса.
2. Доказывающий вычисляет значение $Z = U^{\gamma} \bmod n$. Если $Z = 1$, то в качестве своего ответа, он отправляет $r = 1$ проверяющему. Если $Z > 1$, то в качестве своего ответа направляет $r = 0$ проверяющему.

В каждом раунде доказывающий, если он является подлинным, даёт правильный ответ с вероятностью, равной единице, а нарушитель может дать правильный ответ с вероятностью 0,5. Выполняя z таких раундов и выбирая значение z из интервала $40 \leq z \geq 160$, можно понизить вероятность обмана до пренебрежимо малой величины. Общая схема протокола представлена на рисунке 5.1, используемые алгоритмы на рисунке 5.2.

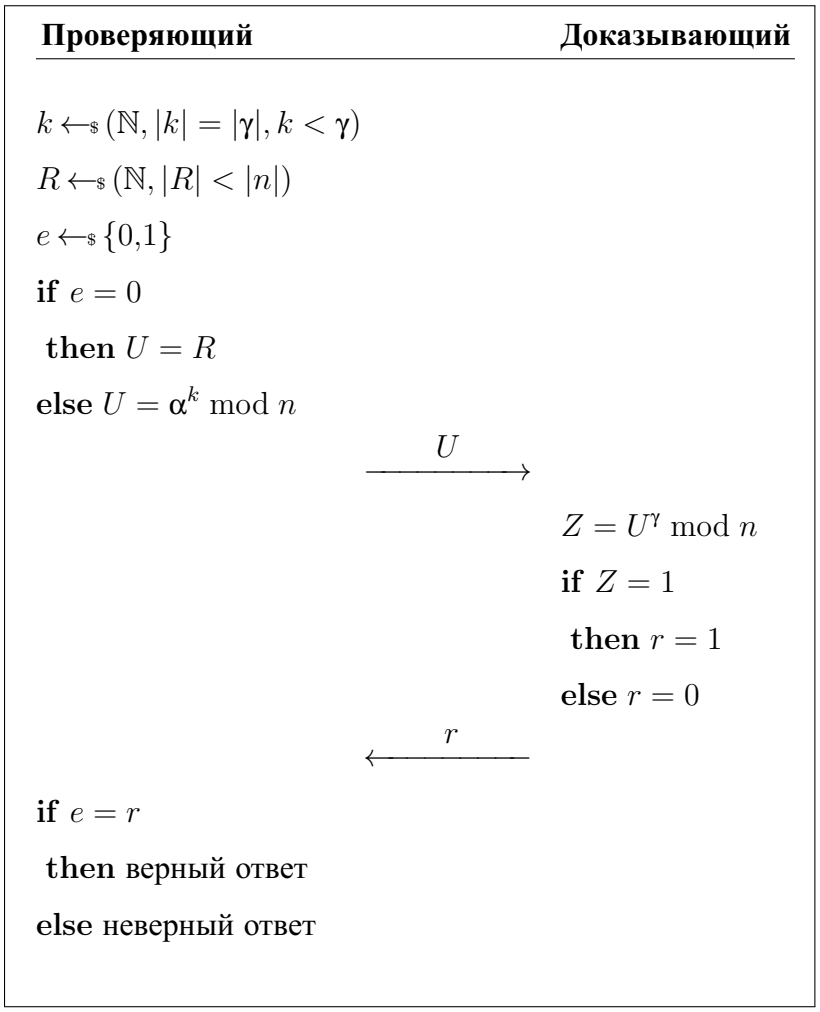

Рисунок 5.1 — Интерактивный протокол аутентификации субъекта

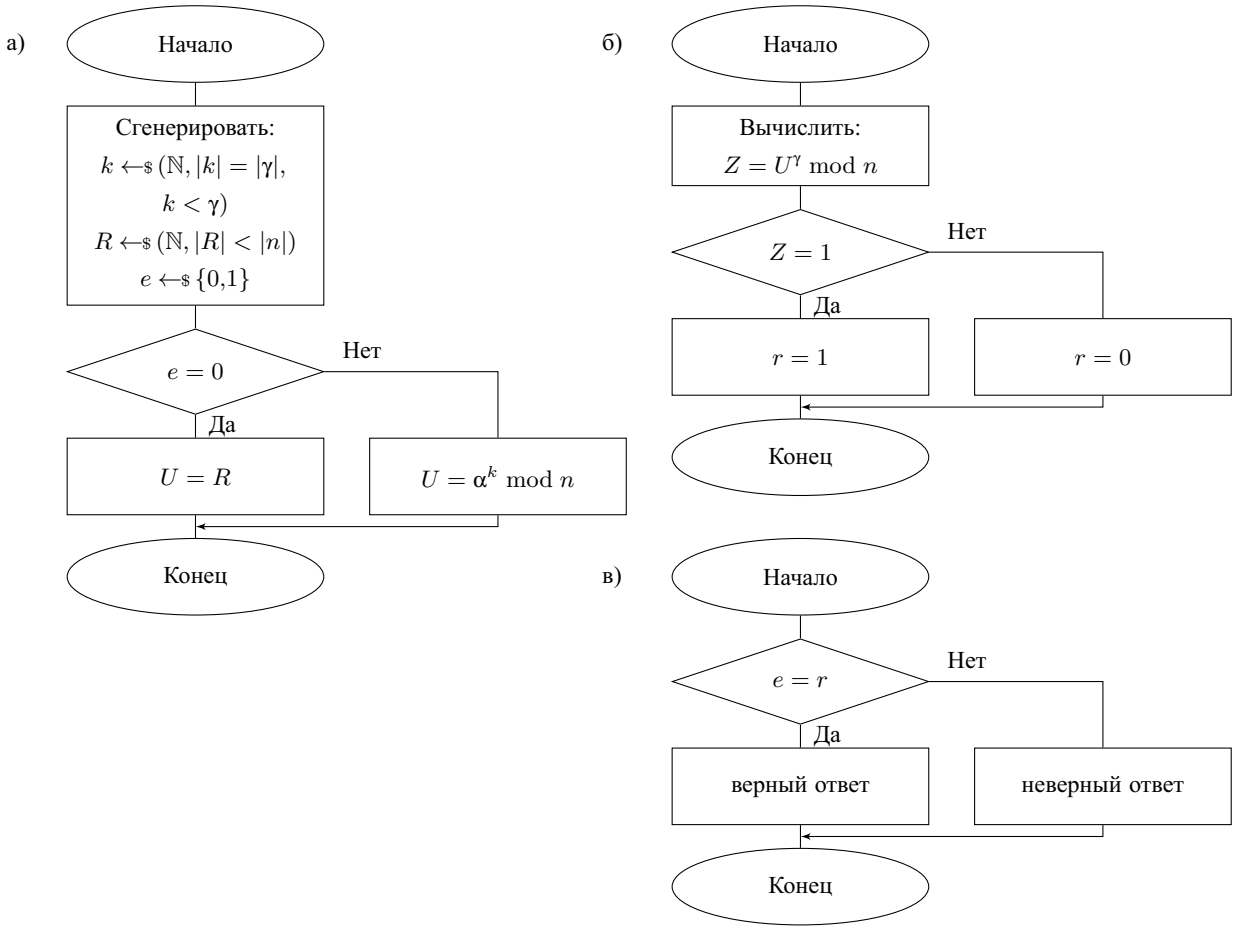

Рисунок 5.2 — Интерактивный протокол аутентификации субъекта: а) алгоритм генерации запроса; б) алгоритм генерации ответа; в) алгоритм проверки

5.1.2 Двухшаговый протокол аутентификации субъекта

Используя вычислительную сложность решения ЗДЛ по трудно факторизуемому модулю и ОК вида (n, α, y), где $y = \alpha^x \bmod n$, $n = pq$, а тройка значений (p, q, x) являются СК, можно предложить следующий протокол (для генерации системных параметров и пары ОК, СК можно использовать алгоритмы, описанные в разделах 2.4 и 3.1):

1. Проверяющий генерирует случайное число $k \leftarrow_\$ (\mathbb{N}, |k| = |\gamma|, k < \gamma)$ и вычисляет значения $U = \alpha^k \bmod n$ и $Z = y^k \bmod n$, где y, n и α - ОК доказывающего. Затем в качестве запроса отправляет значение U доказывающему.
2. Доказывающий проверяет выполнимость соотношения $U^\gamma \bmod n = 1$. Если соотношение не выполняется, то он посылает проверяющему ответ "некорректный запрос"и прекращает обмен. В противном случае, он

вычисляет значение $Z' = U^x \bmod n$ и направляет Z' в качестве своего ответа проверяющему.

Если $Z = Z'$, то проверяющий делает вывод, что доказывающий действительно является владельцем СК, связанного с проверяемым ОК.

Нулевая утечка обеспечивается тем, что проверяющий получает в ответ значение, которое он уже знает и вычислил самостоятельно. Проверка значения U является принципиальным моментом. Если эта проверка не делается, то проверяющий может в качестве запроса U выбирать случайные числа и в качестве ответа получать значения, которые ему не были заранее известны, так как вероятность выполнимости соотношения $U^\gamma \bmod n = 1$ является пренебрежимо малой. То есть в этом случае нельзя говорить о нулевой утечке информации о СК. Общая схема протокола представлена на рисунке 5.3, используемые алгоритмы на рисунке 5.4.

Проверяющий		**Доказывающий**
$k \leftarrow_\$ (\mathbb{N}, \lvert k\rvert = \lvert\gamma\rvert, k < \gamma)$		
$U = \alpha^k \bmod n$		
$Z = y^k \bmod n$		
	$\xrightarrow{U}$	
		if $U^\gamma \bmod n \neq 1$
		then END
		else $Z' = U^x \bmod n$
	$\xleftarrow{Z'}$	
if $Z = Z'$		
then верный ответ		
else неверный ответ		

Рисунок 5.3 — Двухшаговый протокол аутентификации субъекта

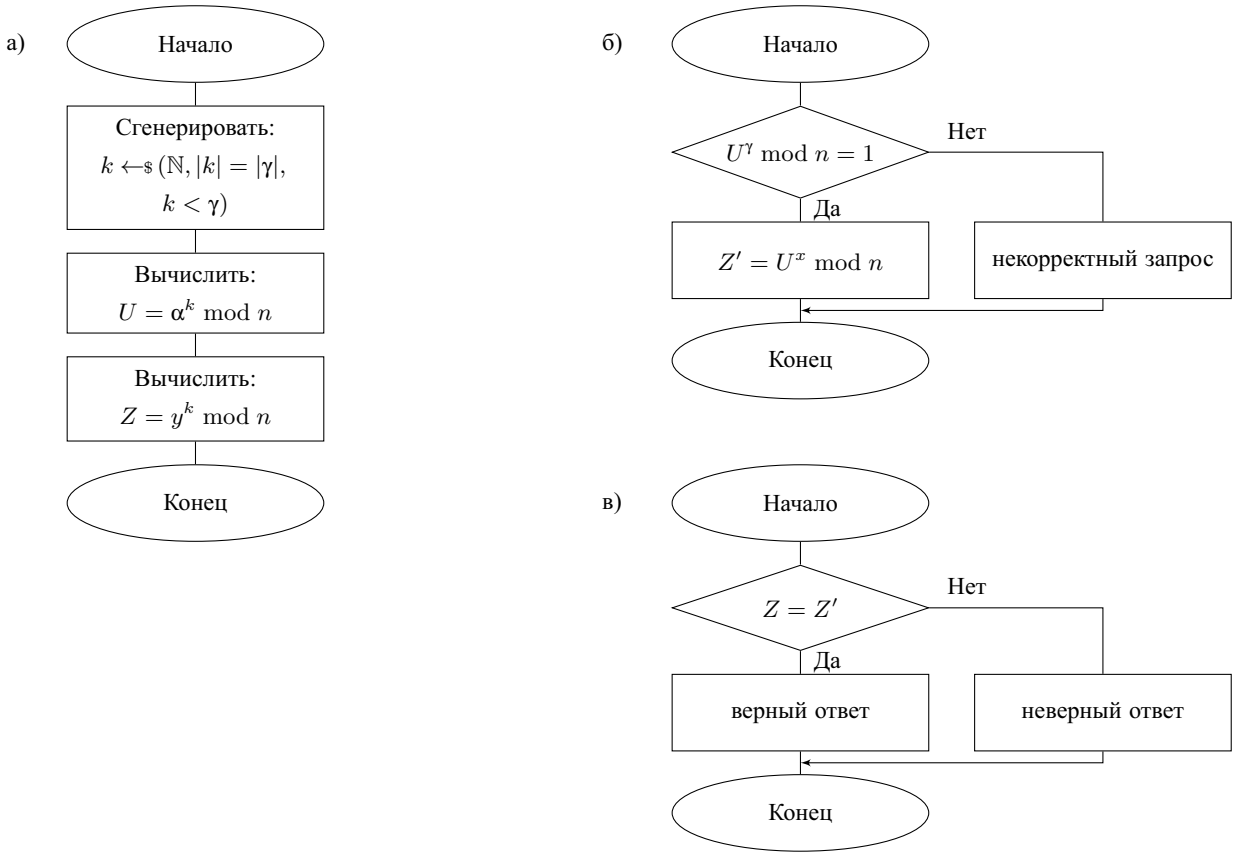

Рисунок 5.4 — Двухшаговый протокол аутентификации субъекта: а) алгоритм генерации запроса; б) алгоритм генерации ответа; в) алгоритм проверки

Сравнение разработанного протокола аутентификации с аналогами представлено в таблице 12. Отметим, что вероятность выдачи нарушителем себя за легального пользователя зависит от стойкости вычислительно трудных ЗФ и ЗДЛ, используемых во всех приведённых в сравнении протоколах. В протоколах аналогах есть одна существенная особенность, в работах указаны только 3 посылки по сети, но инициатором протокола является доказывающий, и перед инициализацией он должен узнать кому доказывать, следовательно потребуется дополнительная посылка по сети для инициализации протокола. В результате для всех протоколов аналогов потребуется 4-ре посылки по сети.

Таблица 12 — Сравнение характеристик предложенного протокола аутентификации с аналогами

Протокол	Кол-во посылок по сети	Трудоёмкость выч. доказывающего (возв. в ст. по мод.)	Трудоёмкость выч. проверяющего (возв. в ст. по мод.)
предложенный	2	2	2
Поинтчеваль 2000 [92]	4	1 + хэш	2 + хэш
Бриккель 1992 [68]	4	1	2
Жиральт 1991 [66]	4	1	2

5.2 Коммутативное шифрование

Коммутативное шифрование позволяет производить многократное зашифрование и расшифрование сообщения на различных ключах в различной последовательности без использования общих или открытых ключей, так же класс таких шифров называют протоколами бесключевого шифрования [8;9;12;15;16].

Пусть $C = E_e(M)$, где $E_e(M)$ - функция шифрования сообщения M по ключу e, C - криптограмма, $D_d(C)$ - функция расшифрования шифртекста C по ключу d. Каждый i-тый пользователь протокола генерирует свою пару ключей (e, d), являющихся ключами зашифрования и расшифрования соответственно. Свойство коммутативности выражается в следующем.

$$C = E_{e_j}(E_{e_i}(M)) = E_{e_i}(E_{e_j}(M))$$
$$M = D_{d_j}(D_{d_i}(C)) = D_{d_i}(D_{d_j}(C))$$
$$M = D_{d_j}(E_{e_i}(D_{d_i}(E_{e_j}(M)))) = D_{d_i}(E_{e_j}(D_{d_j}(E_{e_i}(M))))$$

То есть от изменения порядка зашифрования и расшифрования результат не меняется.

Самым известным представителем протоколов коммутативного шифрования является трёхпроходный протокол Шамира [12], в основе которого лежит алгоритм Полига-Хеллмана [139], так же известна его модификация, использующая конечное двоичное поле - протокол Месси-Омура [140].

Трёхпроходный протокол Шамира позволяет двум пользователям обменяться зашифрованным сообщением по открытому каналу связи без использования общих или открытых ключей. Он использует вычислительную сложность решения задачи ЗДЛ. Каждый из пользователей генерирует свою пару ключей зашифрования и расшифрования $ed = 1 \bmod \varphi(p)$, где p большое простое число являющееся общесистемным параметром, $\varphi(p)$ - функция Эйлера.

Пусть пользователь (отправитель) хочет отправить сообщение другому пользователю (получателю) с использованием трёхпроходного протокола Шамира. Для этого необходимо выполнить следующие шаги.

1. Отправитель генерирует свою пару ключей зашифрования и расшифрования $e, d \leftarrow_{\$} (\mathbb{N}, ed = 1 \bmod \varphi(p))$. Зашифровывает сообщение M по формуле $C = E_e(M) = M^e \bmod p$ и отправляет криптограмму C получателю.

2. Получатель генерирует свою пару ключей зашифрования и расшифрования $e_1, d_1 \leftarrow_{\$} (\mathbb{N}, e_1 d_1 = 1 \bmod \varphi(p))$. Зашифровывает криптограмму C по формуле $C_1 = E_{e_1}(C) = C^{e_1} \bmod p$ и отправляет её отправителю.

3. Отправитель расшифровывает полученную криптограмму C_1 по своему ключу d по формуле $C_2 = D_d(C_1) = C_1^d \bmod p$ и отправляет C_2.

Получатель расшифровывает сообщение M из криптограммы C_2 по своему ключу d_1 по формуле $M = D_{d_1}(C) = C^{d_1} \bmod p$. Общая схема протокола представлена на рисунке 5.5.

Отправитель		**Получатель**
$e, d \leftarrow_{\$} (\mathbb{N}, ed = 1 \bmod \varphi(p))$		
$C = E_e(M) = M^e \bmod p$		
	$\xrightarrow{C}$	
		$e_1, d_1 \leftarrow_{\$} (\mathbb{N}, e_1 d_1 = 1 \bmod \varphi(p))$
		$C_1 = E_{e_1}(C) = C^{e_1} \bmod p$
	$\xleftarrow{C_1}$	
$C_2 = D_d(C_1) = C_1^d \bmod p$		
	$\xrightarrow{C_2}$	
		$M = D_{d_1}(C_2) = C_2^{d_1} \bmod p$

Рисунок 5.5 — Трёхпроходный протокол Шамира

Использование составного трудно факторизуемого модуля n представляющему собой произведение двух больших сильных простых чисел p и q ($n = pq$) вместо простого модуля p напрямую в этом протоколе имеет следующие особенности:

1) вычисления по различным модулям не обладают свойством коммутативности;
2) коммутативное шифрование будет работать корректно только для сообщений представимых в виде $\alpha^i \bmod n$, то есть значение шифруемого сообщение должно быть элементом мультипликативной подгруппы, порождаемой генератором α.

Из первой особенности следует, что для построения протокола бесключевого шифрования потребуется чтобы составной модуль n был общесистемным параметром. Роль по генерации такого модуля $n = pq$ может выполнять доверительный центр, который после генерации уничтожит делители p, q. Однако из-за уничтожения делителей pq пользователи не смогут вычислить функцию Эйлера $\varphi(n) = (p-1)(q-1)$ из-за чего становится невозможно вычислить ключи зашифрования и расшифрования $ed = 1 \bmod \varphi(n)$. В случае генерации модуля $n = pq$ отправителем, получатель не сможет вычислить свою пару ключей. Если значение $\varphi(n)$ будет предоставляться совместно с модулем, то возможно факторизовать модуль следующим способом [9].

$$\varphi(n) = (p-1)(q-1) = pq - (p+q) + 1 = n - (p+q) + 1$$
$$(p-q)^2 = (p+q)^2 - 4pq = (p+q)^2 - 4n$$
$$p = (p+q) + (p-q)/2$$
$$q = (p+q) - (p-q)/2$$

Если значения e_i, d_i для каждого i-го будет генерировать доверительный центр, то два пользователя i, j могут обменяться своими значениями e_i, d_i, e_j, d_j, что позволит вычислить функцию Эйлера от модуля $\varphi(n)$ и факторизовать, вышеописанным способом, модуль n. В случае генерации отправителем пары ключей e, d, потребуется доверенный канал передачи их получателю.

Для нейтрализации вышеописанных атак предлагается использовать схему генерации α с простым порядком γ, раскрытие которого не позволяет факторизовать модуль n, описанную в разделе 3.1.

Вторая особенность устанавливает ограничение на сообщения которые можно шифровать. В случае если таких сообщений сравнительно небольшое количество, то их можно закодировать значениями элементов мультипликативной подгруппы, порождаемой генератором α. Однако для шифрования любых сообщений такой метод неприменим непосредственно. Для решения этой проблемы был разработан механизм расщепления сообщений.

5.2.1 Механизм расщепления сообщений

Механизм расщепления сообщений был впервые предложен в работе [141] и позволяет снять ограничения на шифруемые сообщения. В его основе алгебраический шифр, который в качестве ключа использует значения элементов мультипликативной подгруппы, порождаемой генератором α, так как значения всех элементов невозможно перебрать (Для 128 битовой стойкости, размер мультипликативной подгруппы равен 2^{256}).

Пусть требуется преобразовать сообщение M в значение элемента мультипликативной подгруппы. Системными параметрами являются n, α и γ (для генерации можно использовать алгоритмы, описанные в 2.4). Для преобразования сообщения M необходимо выполнить следующие шаги.

1. Сгенерировать случайное число $k \leftarrow_{\$} (\mathbb{N}, |k| = |\gamma|, k < \gamma)$.
2. Вычислить $K = \alpha^k \bmod n$.
3. Зашифровать сообщение M по формуле $C = (M + K)K \bmod n$.

Сообщение M можно восстановить из двух значений C и K. В данном случае можно говорить о расщеплении сообщения M на два значения, C является числом произвольного вида, а K - элементом мультипликативной подгруппы. Блок схема алгоритма расщепления сообщения M представлена на рисунке 5.6.

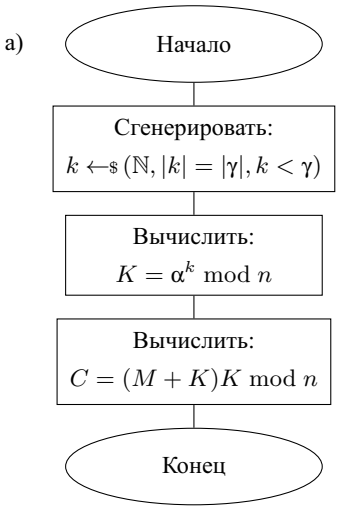

Рисунок 5.6 — Алгоритм расщепления сообщения M

Использование более простых формул $C = M + K \bmod n$ и $C = MK \bmod n$ в данном механизме позволяет нарушителю раскрывать сообщения из заранее известного набора M_1,M_2,...M_i,... или короткие сообщения, значения которых можно перебрать. В качестве критерия распознавания истинного сообщения используется равенство порядка значений $K_i = C - M_i$ и $K_i = CM_i^{-1}(\bmod\ n)$, соответственно, порядку γ, то есть проверка выполнимости соотношения $K_i \equiv 1 \bmod n$.

5.2.2 Коммутативное шифрование информации с использованием трудно факторизуемого модуля

После расщепления сообщения M на пару значений (C, K) операции коммутативного шифрования можно проводить только над значением K. Тогда операция шифрования может быть выполнена по формуле $S = K^e \bmod n$, а операция расшифрования - $M = S^d \bmod n$. Значение C не подвергается какому-либо преобразованию в процессе шифрования. Получаем следующую схему коммутативного шифрования.

1. При первичном шифровании сообщения M выполняется его расщепление, т.е. представление сообщения M в виде (C, K), и шифрование значения K: $S = K^e \bmod n$. На выходе первичной процедуры шифрования имеем пару значений (C, S).
2. Последующие шаги зашифрования/расшифрования выполняются над значением S.

3. Завершающий шаг расшифрования выполняется как расшифрование значения S и приводит к восстановлению значения K, выбранного на шаге первичного шифрования сообщения, и восстановление значения M по формуле $M = CK^{-1} - K \bmod n$.

Получаем следующий протокол бесключевого шифрования.

1. Отправитель генерирует случайное число $k \leftarrow_{\$} (\mathbb{N}, |k| = |\gamma|, k < \gamma)$. Вычисляет $K = \alpha^k \bmod n$. Зашифровывает сообщение M по формуле $C = (M + K)K \bmod n$. Генерирует ключи зашифрования и расшифрования $e, d \leftarrow_{\$} (\mathbb{N}, ed = 1 \bmod \gamma)$. Зашифровывает значение K по формуле $S = K^e \bmod n$. И отправляет получателю пару (C,S).
2. Получатель генерирует ключи зашифрования и расшифрования $e_1, d_1 \leftarrow_{\$} (\mathbb{N}, e_1 d_1 = 1 \bmod \gamma)$. Зашифровывает значение S по формуле $S_1 = S^{e_1} \bmod n$. И отправляет отправителю значение S_1.
3. Отправитель расшифровывает значение S_2 по формуле $S_2 = S_1^d \bmod n$. И отправляет его получателю.

Получатель расшифровывает значение K по формуле $K = S_2^{d_1} \bmod n$ и восстанавливает сообщение M по формуле $M = CK^{-1} - K \bmod n$. Общая схема протокола представлена на рисунке 5.7.

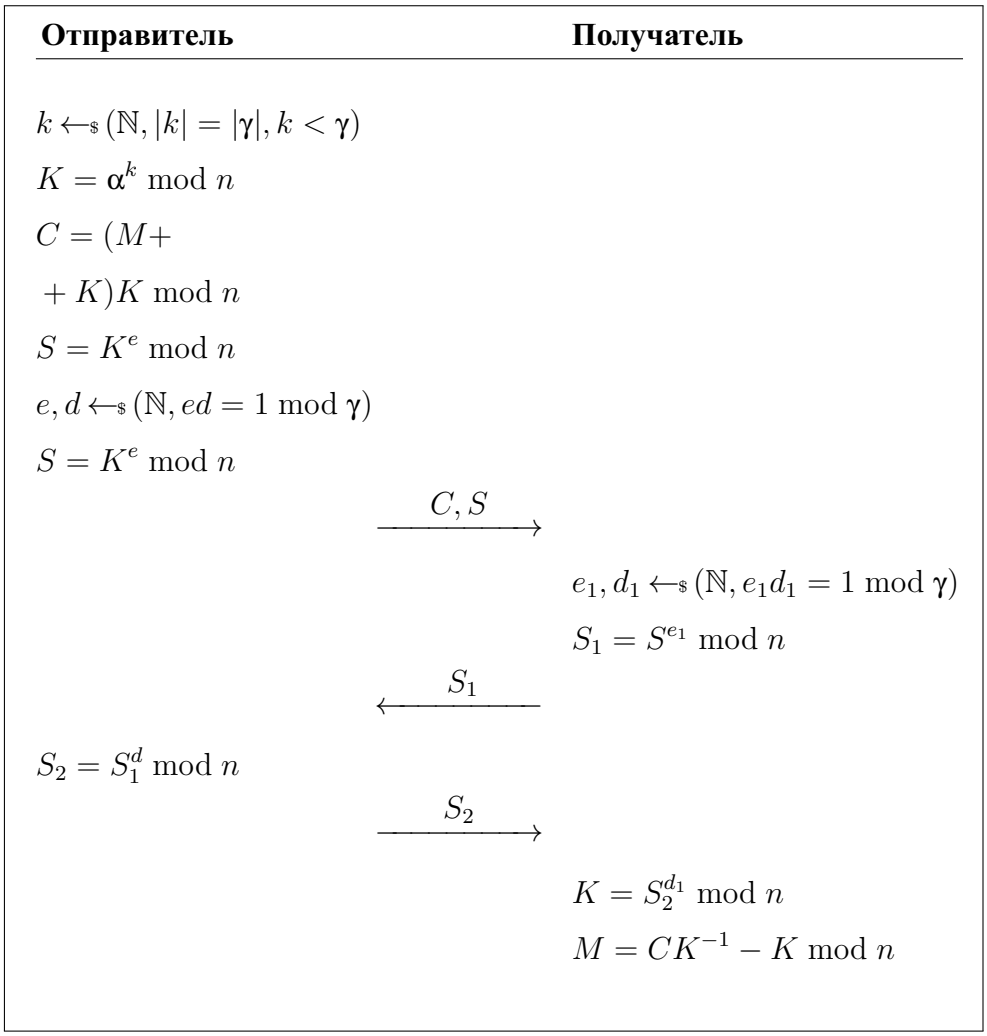

Рисунок 5.7 — Трёхпроходный протокол бесключевого шифрования, взлом которого требует решения двух независимых вычислительно трудных задач

Протокол для произвольного количества участников равного z представлен на рисунке 5.8. Для упрощения восприятия операции зашифрования и расшифрования выполняются последовательно, хотя протокол обладает свойством коммутативности и абсолютно не имеет значения в каком порядке выполняются операции расшифрования и зашифрования.

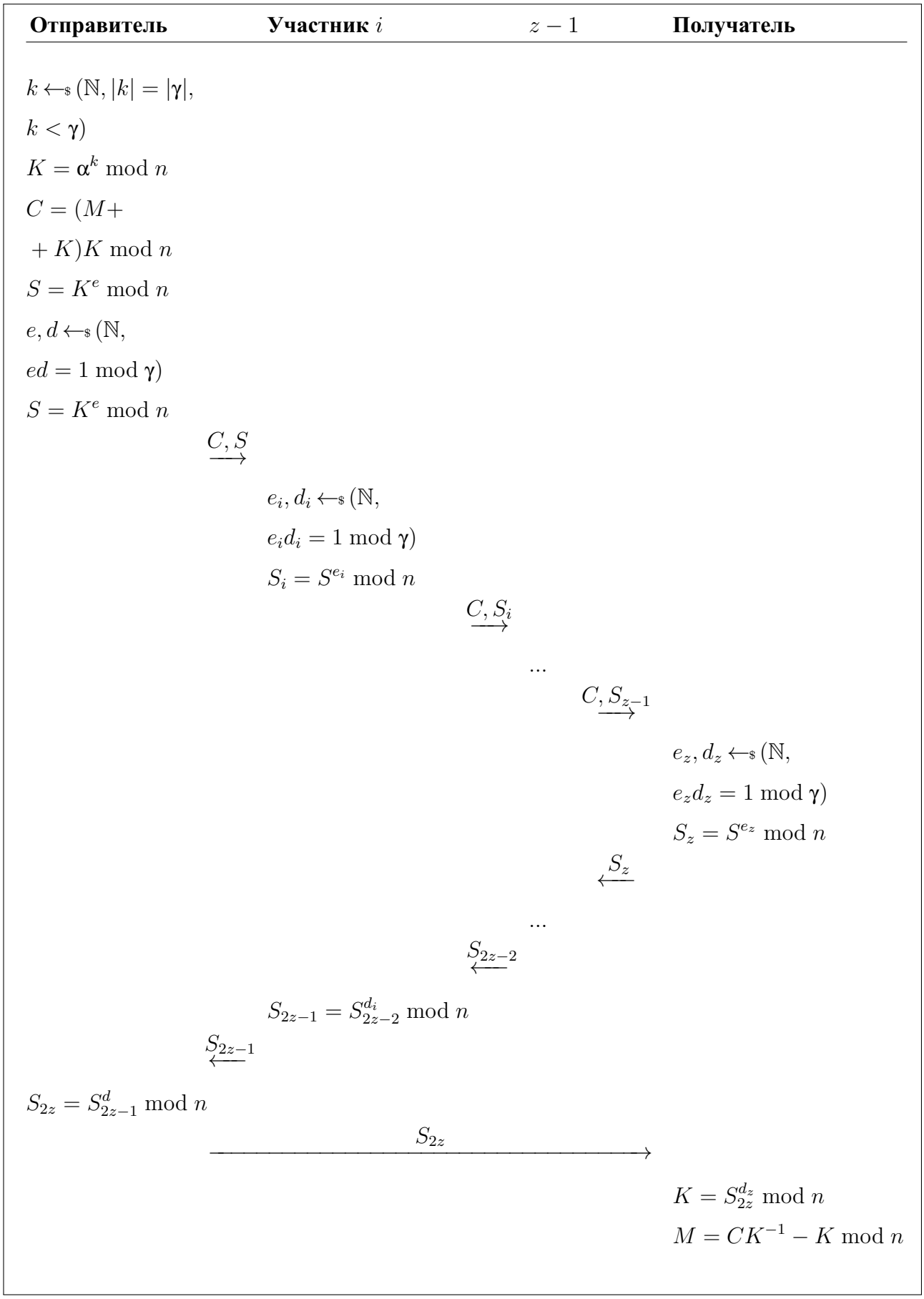

Рисунок 5.8 — Протокол бесключевого шифрования, взлом которого требует решения двух независимых вычислительно трудных задач, для z участников

Сравнение разработанного протокола бесключевого шифрования с протоколами, предложенными в работах [139; 141], представлено в таблице 13.

Таблица 13 — Сравнение характеристик предложенного протокола бесключевого шифрования с аналогами

Протокол	Коэффициент увеличения криптограммы	Трудоёмкость протокола (возв. в ст. по мод.)	Используемые трудные задачи
предложенный	2	4	ЗФ+ЗДЛ
Молдовян 1 2014 [141]	2	8	ЗФ+ЗДЛ
Молдовян 2 2014 [141]	2	4	ЗФ
Похлиг-Хеллман 1984 [139]	1	4	ЗДЛ

5.3 Стойкое шифрование по ключу малого размера

Классические протоколы и алгоритмы шифрования обеспечивают гарантированную стойкость при длине ключа не ниже некоторого значения, обычно 256 бит. Но на практике возникает задача передачи секретного сообщения в условиях ограниченности ключевого материала, то есть абоненты имеют ключ только малого размера (например, 32 бита). Ключи таких малых размеров нельзя использовать в алгоритмах для шифрования секретного сообщения напрямую, так как атакующий может легко найти его путём перебора по ключевому пространству. Для решения данной задачи впервые в работах [142; 143] предлагается концепция шифрования информации по ключу малого размера.

5.3.1 Обоснование концепции стойкого шифрования по ключу малого размера

Основной проблемой бесключевого шифрования является атака "человек посередине". Отправитель и получатель не могут быть уверены во второй стороне, так как в протоколах бесключевого шифрования не предусмотрено средств

аутентификации. Поэтому требуется встроить в протокол бесключевого шифрования алгоритм аутентификации сторон. Для решения этой задачи подходит разделяемый ключ малого размера Q. Причём его применение принципиально отличается от его применения в алгоритмах шифрования. У атакующего будет всего лишь одна попытка угадать ключ, в то время как, при шифровании у него имеется возможность многократной проверки различных значений ключа. Для 32 битного ключа, вероятность его угадывания атакующим составит 2^{-32}, что является достаточным даже для критичных применений. В качестве алгоритмов аутентификации по ключу малого размера возможно использование симметричных шифров. Пусть $G_Q(M)$ - функция симметричного шифрования сообщения M по ключу малого размера Q, $G_Q^{-1}(S)$ функция расшифрования шифртекста S по ключу Q.

В трёхпроходном протоколе Шамира, описанном в разделе 5.2, значения шифротекстов C_1, C_2, C_3 являются вычислительно неотличимыми от случайных значений. Их зашифрование с использованием короткого ключа не позволяет злоумышленнику найти его значение, так как у него не будет критерия отбраковки неверных значений. При использовании данного метода протокол стойкого шифрования по ключу малого размера с использованием трёхпроходного протокола Шамира выглядит следующим образом.

1. Отправитель генерирует свою пару ключей зашифрования и расшифрования $e, d \leftarrow_{\$} (\mathbb{N}, ed = 1 \bmod \varphi(p))$. Зашифровывает сообщение M по формуле $C = E_e(M) = M^e \bmod p$. Зашифровывает полученную криптограмму C с использованием алгоритма симметричного шифрования $S = G_Q(C)$ по разделяемому ключу малого размера Q, отправляет криптограмму S получателю.

2. Получатель генерирует свою пару ключей зашифрования и расшифрования $e_1, d_1 \leftarrow_{\$} (\mathbb{N}, e_1 d_1 = 1 \bmod \varphi(p))$. Расшифровывает полученную криптограмму $C = G_Q^{-1}(S)$ по разделяемому ключу малого размера Q. Зашифровывает криптограмму C по формуле $C_1 = E_{e_1}(C) = C^{e_1} \bmod p$. Зашифровывает полученную криптограмму C_1 с использованием алгоритма симметричного шифрования по формуле $S_1 = G_Q(C_1)$ и отправляет криптограмму S_1 отправителю.

3. Отправитель расшифровывает полученную криптограмму S_1 по разделяемому ключу малого размера Q и получает $C_1 = G_Q^{-1}(S_1)$. Затем, расшифро-

вывает C_1 по своему ключу d по формуле $C_2 = D_d(C_1) = C_1^d \bmod p$ и отправляет C_2 полученную криптограмму получателю.

Получатель расшифровывает сообщение M из криптограммы C_2 по своему ключу d_1 по формуле $M = D_{d_1}(C) = C^{d_1} \bmod p$.

Как видно из протокола, разделяемый ключ используется 2 раза в ходе протокола для шифрования случайных значений. Если в процессе выполнения протокола, атакующий будет выдавать себя за отправителя или получателя сообщения, нарушится свойство коммутативности, из-за различных значений разделяемого ключа малого размера. Для шифрования случайных сообщений M, возможно использовать дополнительно контрольную сумму, которая перед шифрованием будет присоединяться к сообщению, и при расшифровании получатель сможет проверить, было ли вмешательство атакующего или нет. Произвести атаку на используемый алгоритм симметричного шифрования по короткому ключу не представляется возможным, так как при попытке перебора по ключевому пространству у атакующего отсутствуют критерий выбора правильного ключа. Общая схема протокола представлена на рисунке 5.9.

Отправитель		**Получатель**
$e, d \leftarrow_\$ (\mathbb{N}, ed = 1 \bmod \varphi(p))$		
$C = E_e(M) = M^e \bmod p$		
$S = G_Q(C)$		
	$\xrightarrow{S}$	
		$e_1, d_1 \leftarrow_\$ (\mathbb{N}, e_1 d_1 = 1 \bmod \varphi(p))$
		$C = G_Q^{-1}(S)$
		$C_1 = E_{e_1}(C) = C^{e_1} \bmod p$
		$S_1 = G_Q(C_1)$
	$\xleftarrow{S_1}$	
$C_1 = G_Q^{-1}(S_1)$		
$C_2 = D_d(C_1) = C_1^d \bmod p$		
	$\xrightarrow{C_2}$	
		$M = D_{d_1}(C) = C^{d_1} \bmod p$

Рисунок 5.9 — Протокол стойкого шифрования по ключу малого размера

5.3.2 Протокол стойкого шифрования информации по ключу малого размера, основанный на двух трудных задачах

Для построения алгоритма стойкого шифрования по ключу малого размера используем трёхшаговый протокол бесключевого шифрования, предложенный в разделе 5.2.

В качестве средства аутентификации применение симметричного шифра невозможно. Это связано с особенностями работы механизма расщепления сообщений. Пересылаемое сообщение M расщепляется на два числа K и C. После чего все операции шифрования и расшифрования выполняются над значением K. Значение C является вычислительно неотличимым от случайного и может быть зашифровано с использованием алгоритма симметричного шифрования по короткому ключу, но оно участвует в протоколе только один раз при отправке получателю, и не может использоваться в двухсторонней аутентификации отправителя и получателя. Значения S, S_1, S_2 не являются вычислительно неотличимыми от случайных значений, так как они все являются элементами мультипликативной группы, порождаемой генератором α. Шифрование значений S, S_1, S_2 с использованием алгоритма симметричного шифрования по короткому ключу, позволит атакующему найти значение короткого ключа. Для этого будет достаточно перебрать все возможные значения короткого ключа и проверить расшифровываемые значения на принадлежность к группе, путём возведения в степень γ и сравнении с 1 по модулю n. Это можно сделать достаточно быстро при использовании короткого ключа.

Для решения задачи разработки стойкого шифрования информации по ключу малого размера предлагается включить дополнительные возведения генератора группы α в степень равную значению разделяемого короткого ключа Q при вычислении шифротекстов S, S_1, S_2. В таком случае у атакующего не будет никакой возможности вычислить значение короткого ключа и выдать себя за одного из участников протокола. Тогда протокол бесключевого шифрования, взлом которого потребует решения двух независимых вычислительно трудных ЗФ и ЗДЛ, выглядит следующим образом.

Пусть отправитель и получатель обладают разделяемым ключом Q малого размера. Системные параметры n, α и γ были сгенерированы некоторым довери-

тельным центром или отправителем сообщения. Тогда для безопасной передачи сообщения M необходимо выполнить следующие шаги.

1. Отправитель генерирует случайное число $k \leftarrow_{\$} (\mathbb{N}, |k| = |\gamma|, k < \gamma)$. Вычисляет $K = \alpha^k \bmod n$. Зашифровывает сообщение M по формуле $C = (M + K)K \bmod n$. Генерирует ключи зашифрования и расшифрования $e, d \leftarrow_{\$} (\mathbb{N}, ed = 1 \bmod \gamma)$. Зашифровывает значение K по формуле $S = K^e \alpha^Q \bmod n$. И отправляет получателю пару (C,S).
2. Получатель генерирует ключи зашифрования и расшифрования $e_1, d_1 \leftarrow_{\$} (\mathbb{N}, e_1 d_1 = 1 \bmod \gamma)$. Зашифровывает значение S по формуле $S_1 = (S\alpha^{-Q})^{e_1} \alpha^Q \bmod n$. И отправляет отправителю значение S_1.
3. Отправитель расшифровывает значение S_1 по формуле $S_2 = (S_1 \alpha^{-Q})^d \alpha^Q \bmod n$. И отправляет его получателю.

После получения S_1 получатель расшифровывает значение K по формуле $K = (S_2 \alpha^{-Q})^{d_1} \bmod n$ и восстанавливает сообщение M по формуле $M = CK^{-1} - K \bmod n$. Общая схема протокола представлена на рисунке 5.10.

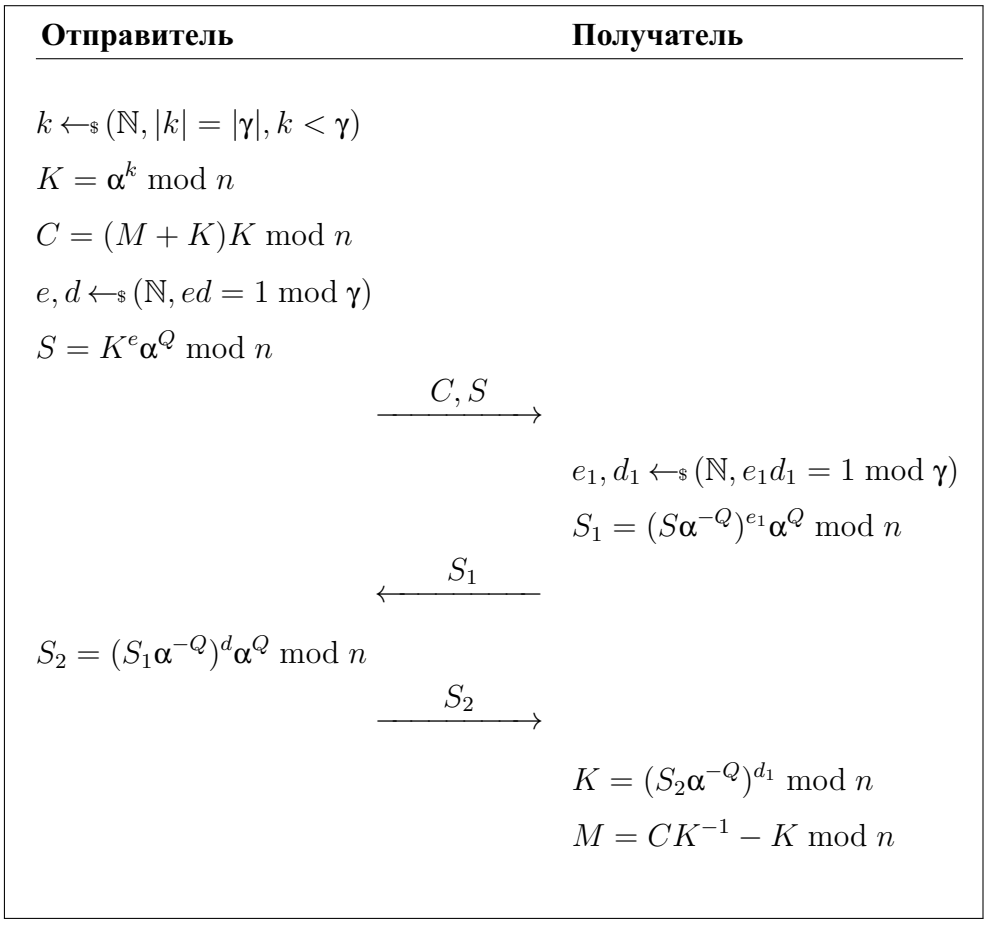

Рисунок 5.10 — Трёхпроходный протокол стойкого шифрования по ключу малого размера с повышенным уровнем безопасности

Доказательство корректности расшифрования.

$$(((((K^e\alpha^Q)\alpha^{-Q})^{e_1}\alpha^Q)\alpha^{-Q})^d\alpha^Q)\alpha^{-Q})^{d_1} \bmod n =$$
$$((((K^{ee_1}\alpha^Q)\alpha^{-Q})^d\alpha^Q)\alpha^{-Q})^{d_1} \bmod n =$$
$$((K^{ee_1d}\alpha^Q)\alpha^{-Q})^{d_1} \bmod n =$$
$$K^{e_1d_1} \bmod n = K \bmod n$$

5.4 Выводы к пятой главе

1. Разработаны протоколы аутентификации удалённых пользователей в информационно-телекоммуникационных системах:
   а) интерактивный протокол аутентификации;
   б) двухшаговый протокол аутентификации.
2. Подобран метод расщепления сообщений, позволяющий снять ограничения на шифруемые сообщения с использованием коммутативного шифрования по составному модулю.
3. На основе рассмотренного метода расщепления сообщений, разработано коммутативное шифрование информации, взлом которого потребует одновременного решения двух вычислительно сложных задач: ЗФ и ЗДЛ.
4. Разработан протокол стойкого шифрования информации по ключу малого размера, взлом которого потребует одновременного решения двух вычислительно сложных задач: ЗФ и ЗДЛ.
5. Взлом всех разработанных алгоритмов и протоколов потребует от атакующего умения одновременного решения двух независимых вычислительно сложных ЗФ и ЗДЛ.

Заключение

Основные результаты работы заключаются в следующем.

1. Впервые предложен универсальный метод построения алгоритмов и протоколов, взлом которых требует одновременного решения двух вычислительно сложных задач, отличающийся использованием задачи дискретного логарифмирования (ЗДЛ) по трудно факторизуемому модулю n, размер множителей которого выбирается таким образом, что по крайней мере решение ЗДЛ по модулю одного из делителей труднофакторизуемого модуля имеет вычислительную сложность не ниже заданного уровня стойкости.
2. На основе нового метода разработаны новые алгоритмические средства аутентификации объектов и субъектов информационных процессов, обладающие повышенным уровнем безопасности:
   I. Протокол электронной цифровой подписи (ЭЦП), отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры, благодаря чему достигнуто снижение размера, вычислительной сложности процедур генерации и проверки ЭЦП;
   II. Протокол коллективной ЭЦП, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры, благодаря чему достигнуто снижение размера, вычислительной сложности процедур генерации и проверки ЭЦП.
   III. Протокол утверждаемой групповой ЭЦП, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры и механизма маскирования ключей, благодаря чему руководитель и только он может доказывать стороннему проверяющему список лиц, которые подписывали документ, без разглашения секретных ключей подчинённых и своего собственного, и никто другой не сможет его определить.
   IV. Протокол интерактивной аутентификации субъекта, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры.

V. Протокол двухшаговой аутентификации субъекта, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры и элемента группы в качестве запроса, благодаря чему достигнута возможность безопасной аутентификации субъекта за два шага.

3. На основе нового метода разработаны новые алгоритмические средства защиты информации, обладающие повышенным уровнем безопасности:
   I. Протоколы обмена ключами, отличающиеся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры и рандомизирующего параметра, благодаря чему обеспечивается случайность значений ключа, формируемого в ходе протокола.
   II. Протокол защитного преобразования информации, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры.
   III. Протокол коммутативного защитного преобразования информации, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры и механизма расщепления сообщений, благодаря чему обеспечивается возможность выполнения защитных преобразований для произвольных сообщений.
   IV. Протокол стойкого защитного преобразования информации с использованием ключа малого размера, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры и процедуры бесключевого защитного преобразования совместно с аутентификацией по коротким ключам, благодаря чему возможно задать необходимую стойкость протокола, для малых длин ключа.
4. На основе нового метода разработаны новые алгоритмические средства обеспечения анонимности, обладающие повышенным уровнем безопасности:
   I. Протокол слепой ЭЦП, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры, благодаря чему достигнуто снижение размера, вычислительной сложности процедур генерации и проверки ЭЦП.

II. Протокол слепой коллективной ЭЦП, отличающийся использованием ЗДЛ по трудно факторизуемому модулю специальной структуры, благодаря чему достигнуто снижение размера, вычислительной сложности процедур генерации и проверки ЭЦП.

Данная работа описывает вопросы, связанные с применением разработанных алгоритмов и протоколов в алгоритмических средствах защиты информации, обрабатываемой с использованием информационно-телекоммуникационных технологий, в процессе её сбора, хранения, обработки, передачи и распространения. Разработанные алгоритмы и протоколы позволяют обеспечить важные аспекты защиты информации, такие как конфиденциальность, аутентичность, анонимность, неотказуемость, достоверность, целостность, подотчётность.

Список работ, опубликованных автором по теме диссертации

В изданиях, рекомендованных ВАК Минобрнауки РФ:

1. **Березин А.Н.** Протокол стойкого шифрования по ключу малого размера, взлом которого требует решения задач факторизации и дискретного логарифмирования // Вопросы защиты информации. — 2016. — № 2. — С. 3-8.
2. **Березин А.Н.**, Молдовян Н. А., Латышев Д. М. Протокол 240–битовой коллективной подписи над нециклической конечной группой // Вопросы защиты информации. — 2013. — № 3. — С. 81–85.
3. **Березин А.Н.**, Молдовян Н. А. Построение криптосхем на основе задачи дискретного логарифмирования по трудно разложимому модулю // Известия СПбГЭТУ «ЛЭТИ». — 2013. — № 7. — С. 54–59.
4. **Березин А.Н.**, Молдовян Н. А., Щербаков В.А. Общий метод построения криптосхем, основанных на трудности одновременного решения задач факторизации и дискретного логарифмирования // Вопросы защиты информации. — 2014. — №2. — С. 3–11.
5. **Березин А.Н.**, Молдовян Н.А., Рыжков А.В. Коммутативные шифры на основе трудности одновременного решения задач факторизации и дискретного логарифмирования // Информационно управляющие системы. — 2014. — №4. — С. 106–110.

Публикации в прочих изданиях, научных сборниках, тезисы докладов

6. **Berezin A.N.**, Moldovyan N.A., Shcherbakov V.A. Cryptoschemes Based on Difficulty of Simultaneous Solving Two Different Difficult Problems // Computer Science Journal of Moldova. — 2013. — V. 21. — № 2(62). — P. 280–290.
7. **Березин А.Н.**, Биричевский А.Р., Молдовян Н.А. Особенности задачи дискретного логарифмирования по составному модулю как криптографического примитива // Труды VII Санкт–Петербургской межрегиональной конференции «Информационная безопасность регионов России (ИБРР–2011)». — Санкт–Петербург, 26–28 октября 2012 г. / СПб.: СПОИСУ, 2012. — С. 104–108.

8. **Березин А.Н.**, Васильев И.Н., Молдовян Н.А. Обоснование криптосхем на основе задачи дискретного логарифмирования по трудно разложимому модулю// 65–я научно–техническая конференция профессорско–преподавательского состава университета. — Санкт-Петербург, 24 января – 4 февраля 2012 г. / Труды конференции. — СПб.:СПбГЭТУ «ЛЭТИ», 2012. — С. 120–124.
9. **Березин А.Н.**, Демьянчук А.А., Молдовян Д.Н., Рыжков А. В. Протоколы аутентификации с нулевым разглашением секрета: приложения, повышение безопасности и новые реализации // XIII Санкт–Петербургская международная конференция «Региональная информатика – 2012», — Санкт–Петербург, 24–26 октября 2012 г. / Труды конференции. — СПб.:СПОИСУ, 2013. — С. 82–83.
10. **Березин А.Н.** Подходы к построению криптосхем на основе трудности одновременного решения задач факторизации и дискретного логарифмирования // Инновационная деятельность в Вооруженных силах Российской Федерации: Труды всеармейской научно-практической конференции. — Санкт-Петербург, 21-22 ноября 2013 г. / СПб.:ВАС, 2013. — С. 72–76.
11. **Березин А.Н.** Подходы к повышению безопасности криптографических алгоритмов и протоколов // 68–я научно–техническая конференция профессорско–преподавательского состава университета. — Санкт-Петербург, 28 января – 5 февраля 2015 г. / Труды конференции. — СПб.:СПбГЭТУ «ЛЭТИ», 2015. С. — 105–108.
12. **Березин А.Н.**, Хо Н.З., Синев В.Е. Методические аспекты обоснования задачи дискретного логарифмирования по трудно разложимому модулю в дисциплине «Криптографические протоколы» // XVIII международная научно–методическая конференция «Современное образование: содержание, технологии, качество». — Санкт–Петербург, 18 апреля 2012 г. / Материалы конференции. — Т. 1. — СПб.:СПбГЭТУ «ЛЭТИ», 2012. — С. 246–248.
13. **Березин А.Н.**, Демьянчук А.А., Краснова А.И. Протоколы с нулевым разглашением на основе трудности вычисления порядка элементов конечной группы // XIII Санкт–Петербургская международная конферен-

ция «Региональная информатика – 2012». — Санкт–Петербург, 24–26 октября 2012 г. / Материалы конференции. — СПб.:СПОИСУ, 2012. — С. 82–83.

14. **Березин А.Н.**, Рыжков А.В. Подход к повышению безопасности процедуры коммутативного шифрования // XIII Санкт–Петербургская международная конференция «Региональная информатика – 2012». — Санкт–Петербург, 24–26 октября 2012 г. / Материалы конференции. — СПб.:СПОИСУ, 2012. — С. 123.
15. **Березин А.Н.**, Демьянчук А.А., Кишмар Р.В. Двухпроходные протоколы аутентификации с нулевым разглашением // XIII Санкт–Петербургская международная конференция «Региональная информатика – 2012». — Санкт–Петербург, 24–26 октября 2012 г. / Материалы конференции. — СПб.:СПОИСУ, 2012. — С. 89.
16. **Березин А.Н.**, Демьянчук А.А. Расширенное изложение протоколов с нулевым разглашением секрета в дисциплине “Криптографические протоколы” // XIX международная научно–методическая конференция «Современное образование: содержание, технологии, качество». — Санкт–Петербург, 24 апреля 2013 / Материалы конференции. — Т. 1. — СПб.:СПбГЭТУ «ЛЭТИ», 2013. — С. 138–140.
17. **Березин А.Н.** Подходы к построению криптосхем на основе задач факторизации и дискретного логарифмирования // Материалы VIII СПб межрегиональная конференция «Информационная безопасность регионов России (ИБРР-2013)». — Санкт–Петербург, 23-25 октября 2013 / СПб.:СПОИСУ, 2013. — С. 80–81.
18. **Березин А.Н.** Варианты задачи дискретного логарифмирования по составному модулю // Материалы VIII СПб межрегиональная конференция «Информационная безопасность регионов России (ИБРР-2013)». — Санкт–Петербург, 23-25 октября 2013 г. / СПб.:СПОИСУ, 2013. — С. 81–82.
19. **Березин А.Н.**, Демьянчук А.А., Рыжков А.В. Протоколы с нулевым разглашением, использующие алгоритмы открытого шифрования // Материалы VIII СПб межрегиональная конференция «Информационная без-

опасность регионов России (ИБРР-2013)». — Санкт–Петербург, 23-25 октября 2013 г. / СПб.:СПОИСУ, 2013. — С. 82–83.

20. **Березин А.Н.** Новый способ повышения уровня безопасности криптографических механизмов обеспечения информационной безопасности // Восемнадцатая Санкт-Петербургская ассамблея молодых учёных и специалистов. — Санкт-Петербург, 13 декабря 2013 г. / СПб.:ЦОП РГГМУ, 2013. — С. 34.
21. **Березин А.Н.** Коммутативные шифры на основе двух трудных задач // XIV Санкт–Петербургская международная конференция «Региональная информатика – 2014». — Санкт–Петербург, 29–31 октября 2014 г. / Материалы конференции. — СПб.:СПОИСУ, 2014. — С. 121–122.
22. **Березин А.Н.** Расширение типов криптографических схем с повышенным уровнем безопасности // 19 Санкт-Петербургская ассамблея молодых учёных и специалистов. — Санкт-Петербург, 21 декабря 2014 г. / СПб.:ЦОП РГГМУ, 2014 — С. 211 .
23. **Березин А.Н.** Протоколы стойкого шифрования по ключу малого размера, основанные на коммутативных преобразованиях // 20 Санкт-Петербургская ассамблея молодых учёных и специалистов. — Санкт-Петербург, 18 декабря 2015 г. / СПб.:ЦОП РГГМУ. — С. 41.
24. **Березин А.Н.**, Галанов А.И., Синев В.Е. Протокол утверждаемой групповой цифровой подписи на основе двух трудных задач //Информационная безопасность регионов России (ИБРР-2015). IX Санкт-Петербургская межрегиональная конференция. — Санкт-Петербург, 28-30 октября 2015 г. / Материалы конференции. — СПб.:СПОИСУ, 2015. — С. 101-102.
25. **Березин А.Н.**, Молдовян Н.А., Муравьев А.В. Протокол шифрования на основе двух трудных задач по ключу малого размера //Информационная безопасность регионов России (ИБРР-2015). IX Санкт-Петербургская межрегиональная конференция. — Санкт-Петербург, 28-30 октября 2015 г. / Материалы конференции. — СПб.:СПОИСУ, 2015. — С. 116-117.

Список литературы

1. Федеральный закон от 27.07.2006 N 149-ФЗ (ред. от 31.12.2014) "Об информации, информационных технологиях и о защите информации"(с изм. и доп., вступ. в силу с 01.09.2015). — 2015.

2. ГОСТ Р ИСО/МЭК ТО 13335-1-2006. Информационная технология. Методы и средства обеспечения безопасности. Часть 1. Концепция и модели менеджмента безопасности информационных и телекоммуникационных технологий. — М.: Стандартинформ, 2007. — 22 с.

3. ГОСТ 50.1. 053-2005 Информационные технологии. Основные термины и определения в области технической защиты информации. — М.: Стандартинформ, 2005. — 13 с.

4. Защита от несанкционированного доступа к информации. Термины и определения. — М.: ГТК РФ, 1992.

5. ГОСТ Р ИСО 7498-2-99. Информационная технология. Взаимосвязь открытых систем. Базовая эталонная модель. Часть 2. Архитектура защиты информации. — М.: Стандартинформ, 1999. — 39 с.

6. Криптография: скоростные шифры / А.А. Молдовян, Н.А. Молдовян, Б.В. Изотопов, Н.Д. Гуц. — 2002. — Т. 495. — С. 495.

7. ГОСТ Р 34.12–2015 Информационная технология. Криптографическая защита информации. Блочные шифры. — М.: Стандартинформ, 2015.

8. *Katz J., Lindell Y.* Introduction to Modern Cryptography / Ed. by Douglas Stinson. — Chapman Hall CRC, 2008.

9. *Молдовян Н. А.* Теоретический минимум и алгоритмы цифровой подписи. — БХВ-Петербург, 2010. — С. 304.

10. Информатика: Введение в информационную безопасность / М. А. Вус, В. С. Гусев, Д. В. Долгирев, А. А. Молдовян. — 2004.

11. *Diffie W., Hellman M. E.* New directions in cryptography // *Information Theory, IEEE Transactions on*. — 1976. — Vol. 22, no. 6. — Pp. 644–654.

12. *Menezes A. J., Van Oorschot P. C., Vanstone S. A.* Handbook of applied cryptography. — CRC press, 1996.

13. *McCurley K. S.* The discrete logarithm problem // Proc. of Symp. in Applied Math. — Vol. 42. — 1990. — Pp. 49–74.

14. *Василенко О. Н.* Теоретико-числовые алгоритмы в криптографии. — 2003. — С. 328.

15. *Shoup V.* A computational introduction to number theory and algebra. — Cambridge university press, 2009.

16. *Hoffstein J., Pipher J. C.and Silverman J. H.* An introduction to mathematical cryptography. — Springer, 2008. — P. 530.

17. *Merkle R. C., Hellman M. E.* Hiding information and signatures in trapdoor knapsacks // *Information Theory, IEEE Transactions on*. — 1978. — Vol. 24, no. 5. — Pp. 525–530.

18. *Shamir A.* A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystem // Advances in Cryptology / Springer. — 1983. — Pp. 279–288.

19. *Moldovyan D. N., Moldovyan N. A.* Cryptoschemes over hidden conjugacy search problem and attacks using homomorphisms // *Quasigroups and Related Systems*. — 2010. — Vol. 18.

20. *Moldovyan D. N.* Non-commutative finite groups as primitive of public key cryptosystems // *Quasigroups and Related Systems*. — 2010. — Vol. 18. — Pp. 165–176.

21. *Молдовяну П. А., Молдовян Д. Н., Дернова Е. С.* Гомоморфизм и многомерная цикличность конечных групп векторов в синтезе алгоритмов ЭЦП // *Вопросы защиты информации*. — 2009. — № 3. — С. 2–8.

22. *Березин А. Н., Гурьянов Д. Ю., Молдовян Д. Н.* Гомоморфизмы конечных групп векторов и выбор параметров криптосхем на их основе // *Известия ЛЭТИ*. — 2013. — С. 32.

23. *Глухов М. М.* К анализу некоторых систем открытого распределения ключей, основанных на неабелевых группах // *Математические вопросы криптографии*. — 2010. — Т. 1, № 4. — С. 5–22.

24. *Rivest R. L., Shamir A., Adleman L.* A method for obtaining digital signatures and public-key cryptosystems // *Communications of the ACM*. — 1978. — Vol. 21, no. 2. — Pp. 120–126.

25. *Rabin M. O.* Digitalized signatures and public-key functions as intractable as factorization // *MIT*. — 1979.

26. *Crandall R., Pomerance C. B.* Prime numbers: a computational perspective. — Springer Science, 2006. — Vol. 182.

27. *Bressoud D. M.* Factorization Primality Testing. — 1989.

28. *Pollard J. M.* A Monte Carlo method for factorization // *BIT Numerical Mathematics*. — 1975. — Vol. 15, no. 3. — Pp. 331–334.

29. *Vasilenko O. N.* Number-theoretic algorithms in cryptography. — American Mathematical Soc., 2007. — P. 232.

30. *Pollard J. M.* Theorems on factorization and primality testing // Mathematical Proceedings of the Cambridge Philosophical Society / Cambridge Univ Press. — Vol. 76. — 1974. — Pp. 521–528.

31. *Lehman R. S.* Factoring large integers // *Mathematics of Computation*. — 1974. — Vol. 28, no. 126. — Pp. 637–646.

32. *Morrison M. A., Brillhart J.* A method of factoring and the factorization of F7 // *Mathematics of computation*. — 1975. — Vol. 29, no. 129. — Pp. 183–205.

33. *Lenstra J., Hendrik W.* Factoring integers with elliptic curves // *Annals of mathematics*. — 1987. — Pp. 649–673.

34. *Dixon J. D.* Asymptotically fast factorization of integers // *Mathematics of computation*. — 1981. — Vol. 36, no. 153. — Pp. 255–260.

35. *Pomerance C.* A Tale of Two Sieves // *Notices of the AMS*. — 1996. — Vol. 43. — Pp. 1473–1485.

36. *Lenstra A. K.* Integer factoring // *Designs, codes and cryptography*. — 2000. — Vol. 19, no. 2. — Pp. 101–128.

37. Factorization of a 768-bit RSA modulus / T. Kleinjung, K. Aoki, J. Franke et al. // *Advances in Cryptology–CRYPTO 2010*. — 2010. — Pp. 333–350.

38. *ElGamal T.* A public key cryptosystem and a signature scheme based on discrete logarithms // Advances in cryptology / Springer. — 1985. — Pp. 10–18.

39. *Молдовян Н. А.* Введение в криптосистемы с открытым ключом. — БХВ-Петербург, 2005.

40. ГОСТ Р 34.10–2012 Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. — М.: Стандартинформ, 2012.

41. *Joux A., Odlyzko A., Pierrot C.* The past, evolving present, and future of the discrete logarithm // *Open Problems in Mathematics and Computational Science*. — 2014. — Pp. 5–36.

42. *Yan S. Y.* Quantum Computational Number Theory. — Springer, 2016. — P. 252.

43. *Shanks D.* The infrastructure of a real quadratic field and its applications // Proc. 1972 Number Theory Conf., Boulder, Colorado. — 1972. — Pp. 217–224.

44. *Pollard J. M.* Monte Carlo methods for index computation (mod p) // *Mathematics of computation*. — 1978. — Vol. 32, no. 143. — Pp. 918–924.

45. *Pohlig S. C., Hellman M. E.* An improved algorithm for computing logarithms over GF (p) and its cryptographic significance (Corresp.) // *Information Theory, IEEE Transactions on*. — 1978. — Vol. 24, no. 1. — Pp. 106–110.

46. *Schirokauer O.* Discrete logarithms and local units // *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. — 1993. — Vol. 345, no. 1676. — Pp. 409–423.

47. *Schirokauer O.* Using number fields to compute logarithms in finite fields // *Mathematics of Computation of the American Mathematical Society*. — 2000. — Vol. 69, no. 231. — Pp. 1267–1283.

48. *Adleman L. M., Huang M. A.* Function field sieve method for discrete logarithms over finite fields // *Information and Computation*. — 1999. — Vol. 151, no. 1. — Pp. 5–16.

49. *ECRYPT.* Yearly Report on Algorithms and Keysizes. — 2012.

50. *Shor P. W.* Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer // *SIAM review*. — 1999. — Vol. 41, no. 2. — Pp. 303–332.

51. A Compare between Shor's quantum factoring algorithm and General Number Field Sieve / S. M. Hamdi, S. T. Zuhori, F. Mahmud, B. Pal // *Electrical Engineering and Information and Communication Technology (ICEEICT), 2014 International Conference on*. — 2014. — Pp. 1–6.

52. *Smolin J. A., Smith G., Vargo A.* Oversimplifying quantum factoring // *Nature*. — 2013. — Vol. 499, no. 7457. — Pp. 163–165.

53. *Proos J.and Zalka C.* Shor's discrete logarithm quantum algorithm for elliptic curves // *arXiv preprint quant-ph/0301141*. — 2003.

54. *Lenstra A. K., Verheul E. R.* Selecting cryptographic key sizes // *Journal of cryptology*. — 2001. — Vol. 14, no. 4. — Pp. 255–293.

55. *Lenstra A. K.* Key lengs // Contribution to the handbook of information security. — 2006.

56. *Orman H., Hoffman P.* Determining strengths for public keys used for exchanging symmetric keys. — 2004. — URL: http://www.ietf.org/rfc/rfc3766.txt.

57. Recommendation for key management-part 1: General (revised) / E. Barker, W. Barker, W. Burr et al. // *Special Publication 800-57 Part 1 Rev. 3*. — 2012.

58. *Pailloux P.* Mecanismes cryptographiques - Regies et recommandations, Rev. 2.03 // ANSSI. — 2014.

59. *Margraf M.* Kryptographische Verfahren: Empfehlungen und Schlussellangen // TR-02102-1 v2015-1 BSI. — 2015.

60. *Shamir A., Tromer E.* Factoring large numbers with the TWIRL device // *Advances in Cryptology-CRYPTO 2003*. — 2003. — Pp. 1–26.

61. SHARK: a realizable special hardware sieving device for factoring 1024-bit integers / J. Franke, T. Kleinjung, C. Paar et al. // *Cryptographic Hardware and Embedded Systems CHES 2005*. — 2005. — Pp. 119–130.

62. *Berezin A.N., Moldovyan N.A., Shcherbacov V.A.* Cryptoschemes Based on Difficulty of Simultaneous Solving Two Different Difficult Problems // *Computer Science*. — 2013. — Vol. 21, no. 2. — P. 62.

63. *Shmuely Z.* Composite Diffie-Hellman public-key generating systems are hard to break. // *Technion Dept of Computer Science Technical Report*. — 1985. — P. 21.

64. *Shamir A., Rivest R.L., Adleman L.M.* Mental poker // The mathematical gardner. — Springer, 1981. — Pp. 37–43.

65. *McCurley K. S.* A key distribution system equivalent to factoring // *Journal of cryptology*. — 1988. — Vol. 1, no. 2. — Pp. 95–105.

66. *Girault M.* An identity-based identification scheme based on discrete logarithms modulo a composite number // Advances in Cryptology—EUROCRYPT'90 / Springer. — 1991. — Pp. 481–486.

67. *Schnorr C. P.* Efficient identification and signatures for smart cards // *Advances in cryptology—CRYPTO'89 proceedings*. — 1989. — Pp. 239–252.

68. *Brickell E. F., McCurley K. S.* An interactive identification scheme based on discrete logarithms and factoring // *Journal of Cryptology*. — 1992. — Vol. 5, no. 1. — Pp. 29–39.

69. *Harn L.* Public-key cryptosystem design based on factoring and discrete logarithms // *IEE Proceedings-Computers and Digital Techniques*. — 1994. — Vol. 141, no. 3. — Pp. 193–195.

70. *Lee N.Y., Hwang T.* Modified Harn signature scheme based on factorising and discrete logarithms // *IEE Proceedings-Computers and Digital Techniques*. — 1996. — Vol. 143, no. 3. — Pp. 196–198.

71. *He J., Kiesler T.* Enhancing the security of El Gamal's signature scheme // *IEE Proceedings-Computers and Digital Techniques*. — 1994. — Vol. 141, no. 4. — Pp. 249–252.

72. *Harn L.* Enhancing the security of El Gamal's signature scheme // Computers and Digital Techniques, IEE Proceedings / IET. — Vol. 142. — 1995. — P. 376.

73. *Lee N.Y., Hwang T.* The security of He and Kiesler's signature schemes // Computers and Digital Techniques, IEE Proceedings / IET. — Vol. 142. — 1995. — Pp. 370–372.

74. *Tiersma H. J.* Enhancing the security of El Gamal's signature scheme // *IEE Proceedings-Computers and Digital Techniques*. — 1997. — Vol. 144, no. 1. — Pp. 47–48.

75. *Li C., Chen Y.* Enhancing the Security of He-Kiesler Signature Schemes // *Journal of Beijing institute of technology*. — 2003. — Vol. 12, no. 3. — Pp. 326–328.

76. *Wei S.* A new digital signature scheme based on factoring and discrete logarithms // *Progress on Cryptography*. — 2004. — P. 107.

77. *Laih C. S., Kuo W. C.* New signature schemes based on factoring and discrete logarithms // *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*. — 1997. — Vol. 80, no. 1. — Pp. 46–53.

78. *Li L., Tzeng S., Hwang M.* Improvement of signature scheme based on factoring and discrete logarithms // *Applied Mathematics and Computation*. — 2005. — Vol. 161, no. 1. — Pp. 49–54.

79. *Qian H., Cao Z., Bao H.* Cryptanalysis of Li–Tzeng–Hwang's improved signature schemes based on factoring and discrete logarithms // *Applied mathematics and computation*. — 2005. — Vol. 166, no. 3. — Pp. 501–505.

80. *Shao Z.* Signature schemes based on factoring and discrete logarithms // *IEE Proceedings-Computers and Digital Techniques*. — 1998. — Vol. 145, no. 1. — Pp. 33–36.

81. *Li J., Xiao G.* Remarks on new signature scheme based on two hard problems // *Electronics letters*. — 1998. — Vol. 34, no. 25. — P. 2401.

82. *Lee N. Y.* Security of Shao's signature schemes based on factoring and discrete logarithms // *IEE Proceedings-Computers and Digital Techniques*. — 1999. — Vol. 146, no. 2. — Pp. 119–121.

83. *He W.* Digital signature scheme based on factoring and discrete logarithms // *Electronics Letters*. — 2001. — Vol. 37, no. 4. — Pp. 220–222.

84. *Hwang M. S., Yang C. C., Tzeng S. F.* Improved digital signature scheme based on factoring and discrete logarithms // *Journal of Discrete Mathematical Sciences and Cryptography*. — 2002. — Vol. 5, no. 2. — Pp. 151–155.

85. *Wang C. T., Lin C. H., Chang C. C.* Signature schemes based on two hard problems simultaneously // Advanced Information Networking and Applications, 2003. AINA 2003. 17th International Conference on / IEEE. — 2003. — Pp. 557–560.

86. *Tzeng S. F., Yang C. Y., Hwang M. S.* A new digital signature scheme based on factoring and discrete logarithms // *International Journal of Computer Mathematics*. — 2004. — Vol. 81, no. 1. — Pp. 9–14.

87. *Shao Z.* Digital signature schemes based on factoring and discrete logarithms // *Electronics Letters*. — 2002. — Vol. 38, no. 24. — Pp. 1518–1519.

88. *Chen T. H., Lee W. B., Horng G.* Remarks on some signature schemes based on factoring and discrete logarithms // *Applied mathematics and computation*. — 2005. — Vol. 169, no. 2. — Pp. 1070–1075.

89. *Pon S., Lu E., Jeng A. B.* Meta-He digital signatures based on factoring and discrete logarithms // *Applied mathematics and computation*. — 2005. — Vol. 165, no. 1. — Pp. 171–176.

90. *Shao Z.* Security of meta-He digital signature scheme based on factoring and discrete logarithms // *Applied mathematics and computation*. — 2005. — Vol. 170, no. 2. — Pp. 976–984.

91. *Qian H., Cao Z., Bao H.* Security of Pon–Lu–Jeng's Meta-He digital signature schemes // *Applied mathematics and computation*. — 2005. — Vol. 170, no. 1. — Pp. 724–730.

92. *Pointcheval D.* The composite discrete logarithm and secure authentication // Public Key Cryptography / Springer. — 2000. — Pp. 113–128.

93. *Молдовян Д. Н.* Новый механизм формирования подписи в схемах ЭЦП, основанных на сложности дискретного логарифмирования и факторизации // *Вопросы защиты информации*. — 2005. — № 4. — С. 71.

94. *Гортинская Л. В., Молдовян Д. Н., Молдовян А. А.* Требования к выбору параметров криптосхем на основе RSA-модуля // *Вопросы защиты информации*. — 2005. — № 3. — С. 34–38.

95. *Гортинская Л.В., Молдовян Д.Н.* Основанная на сложности факторизации схема ЭЦП с простым модулем // *Вопросы защиты информации*. — 2005. — № 4. — С. 7–11.

96. *Moldovyan A. A., Moldovyan D. N., Gortinskaya L. V.* Cryptoschemes based on new signature formation mechanism // *Computer Science Journal of Moldova*. — 2006. — Vol. 14, no. 3. — Pp. 397–411.

97. *Дернова Е.С., Молдовян А.А.* Увеличение стойкости алгоритмов на основе комбинирования двух независимых трудных задач // *Информационная безопасность регионов России (ИБРР-2007): Материалы конференции*. — 2007. — С. 23–25.

98. *Дернова Е.С., Молдовян Н.А.* Новый алгоритм ЭЦП, раскрытия которого требует одновременного решения двух трудных задач // *Инновационная деятельность в Вооруженных силах Российской Федерации: Труды всеармейской научно-практической конференции*. — 2007. — С. 22–23.

99. *Дернова Е.С., Молдовян Н.А.* Протоколы коллективной цифровой подписи, основанные на сложности решения двух трудных задач // *Безопасность информационных технологий*. — 2008. — № 2. — С. 79–85.

100. Схемы цифровой подписи, взлом которых требует решения двух трудных задач в одной конечной группе / Е.С. Дернова, Л. М. Нгуен, Костина А.А., Щербаков В.А. // *XI Санкт-Петербургская международная конф. Региональная информатика-2008 (РИ-2008)*. — 2008. — С. 22–24.

101. *Дернова Е.С.* Механизмы аутентификации информации, основанные на двух вычислительно трудных задачах : дис....канд. физ.-мат. наук : 05.13.19 /Дернова Евгения Сергеевна. — СПб., 2009. — С. 162.

102. Blind Signature Protocol Based on Difficulty of Simultaneous Solving Two Difficult Problems / N. H. Minh, D. V. Binh, N. T. Giang, N. A. Moldovyan // *Applied Mathematical Sciences*. — 2012. — Vol. 6, no. 139. — Pp. 6903–6910.

103. *Binh D. V., Minh N. H., Moldovyan N. A.* Digital Signature Schemes from Two Hard Problems // Multimedia and Ubiquitous Engineering. — Springer, 2013. — Pp. 817–825.

104. *Дернова Е.С., Молдовян Н.А.* Синтез алгоритмов цифровой подписи на основе нескольких вычислительно трудных задач // *Вопросы защиты информации*. — 2008. — № 1. — С. 22–26.

105. *Wei S.* Digital signature scheme based on two hard problems // *International Journal of Computer Science and Network Security*. — 2007. — Vol. 7, no. 12. — Pp. 207–209.

106. Security of two signature schemes based on two hard problems / J. Zheng, Z. Shao, S. Huang, T. Yu // *Communication Technology, 2008. ICCT 2008. 11th IEEE International Conference on*. — 2008. — Pp. 745–748.

107. *Lin H.F., Liu J., Chen C.Y.* Improved Shao's signature scheme // *Journal of information science and engineering*. — 2007. — Vol. 23, no. 1. — Pp. 285–298.

108. *Ismail E.S., Tahat N.M., Ahmad R.R.* A new digital signature scheme based on factoring and discrete logarithms // *Journal of mathematics and statistics*. — 2008. — Vol. 4, no. 4. — P. 222.

109. *Tahat N. M., Ismail E. S., Ahmad R. R.* A New Blind Signature Scheme Based On Factoring and Discrete Logarithms // *International Journal of Cryptology Research*. — 2009. — Vol. 1, no. 1. — Pp. 1–9.

110. *Ismail E. S., Tahat N. M.* A new signature scheme based on multiple hard number theoretic problems // *ISRN Communications and Networking*. — 2011. — Vol. 2011.

111. *Ismail E.S., Hijazi M.S.* New cryptosystem using multiple cryptographic assumptions // *Journal of Computer Science*. — 2011. — Vol. 7, no. 12. — P. 1765.

112. *Ismail E. S., Hijazi M. S.* A New Cryptosystem Based on Factoring and Discrete Logarithm Problems // *Journal of Mathematics and Statistics*. — 2011. — Vol. 7, no. 3. — Pp. 165–168.

113. *Verma S., Sharma B. K.* A New Digital Signature Scheme Based on Two Hard Problems // *Int. J. Pure Appl. Sci. Technol*. — 2011. — Vol. 5, no. 2. — Pp. 55–59.

114. *Al-Fayoumi M., Aboud S. J., Al-Fayoumi M. A.* A New Digital Signature Scheme Based on Integer Factoring and Discrete Logarithm Problem. // *IJ Comput. Appl.* — 2010. — Vol. 17, no. 2. — Pp. 108–115.

115. Построение схемы 240-битовой цифровой подписи / Д.Н. Молдовян, И.Н. Васильев, Д.М. Латышев, Д.К. Сухов // *Вопросы защиты информации*. — 2011. — № 3. — С. 6–11.

116. *Moldovyan A. N., Moldovyan N.A., Novikova E.* Blind 384-bit digital signature scheme // *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security*. — 2012. — Pp. 77–83.

117. *Березин А. Н., Молдовян Н. А., Латышев Д. М.* Протокол 240-битовой коллективной подписи над нециклической конечной группой // *Вопросы защиты информации*. — 2013. — Т. 102, № 3. — С. 81–85.

118. *Vishnoi S., Shrivastava V.* A new Digital Signature Algorithm based on Factorization and Discrete Logarithm problem // *International Journal of Computer Trends and Technology (IJCTT)*. — 2012. — Vol. 3.

119. *Chiou S., He Y.* Remarks on new Digital Signature Algorithm based on Factorization and Discrete Logarithm problem // *International Journal of Computer Trends and Technology (IJCTT)*. — 2013. — Vol. 4.

120. *Vijay A., Trika P., Madhur K.* A New Variant of RSA Digital Signature // *International Journal of Advanced Research in Computer Science and Software Engineering*. — 2012. — Vol. 2, no. 10. — Pp. 366–371.

121. *Madhur K.* Modified ElGamal over RSA Digital Signature Algorithm (MERDSA) // *International Journal of Advanced Research in Computer Science*. — 2013. — Vol. 4, no. 3.

122. *Ciss A. A., Youssef A.* A Factoring and Discrete Logarithm based Cryptosystem // *International Journal of Contemporary Mathematical Sciences*. — 2013. — Vol. 8, no. 11. — Pp. 511–517.

123. *Shao Z., Gao Y.* A Provably Secure Signature Scheme based on Factoring and Discrete Logarithms // *Appl. Math*. — 2014. — Vol. 8, no. 4. — Pp. 1553–1558.

124. *Nimbalkar A.B., Desai C.G.* A Modified Digital Signature Schemes based on Integer Factorization and Discrete Logarithms // *International Journal of Research in Computer Engineering and Electronics*. — 2014. — Vol. 2, no. 6.

125. *Chiou S. Y.* Novel Digital Signature Schemes based on Factoring and Discrete Logarithms // *International Journal of Security and Its Applications*. — 2016. — Vol. 10, no. 3. — Pp. 295–310.

126. *Молдовян Н. А.* Практикум по криптосистемам с открытым ключом. — БХВ-Петербург, 2007.

127. *Gordon J.* Strong primes are easy to find // Advances in Cryptology / Springer. — 1985. — Pp. 216–223.

128. *Biham E., Boneh D., Reingold O.* Breaking generalized Diffie–Hellman modulo a composite is no easier than factoring // *Information Processing Letters*. — 1999. — Vol. 70, no. 2. — Pp. 83–87.

129. *Schnorr C.P.* Efficient signature generation by smart cards // *Journal of cryptology*. — 1991. — Vol. 4, no. 3. — Pp. 161–174.

130. *Chaum D., Rivest R. L., Sherman A. T.* Blind Signature System // *Advances in cryptology Proceedings of CRYPTO*. — 1983. — Vol. 82. — Pp. 279–303.

131. *Tahat N. M., Shatnawi S. M., Ismail E. S.* A new partially blind signature based on factoring and discrete logarithms // *Journal of Mathematics and Statistics*. — 2008. — Vol. 4, no. 2. — P. 124.

132. *Молдовян А.А., Молдовян Н.А.* Коллективная ЭЦП—специальный криптографический протокол на основе новой трудной задачи // *Вопросы защиты информации*. — 2008. — № 1. — С. 14–18.

133. *Молдовян А.А., Молдовян Н.А.* Новые алгоритмы и протоколы для аутентификации информации в АСУ // *Автоматика и телемеханика*. — 2008. — № 7. — С. 157–169.

134. *Moldovyan N. A., Moldovyan A. A.* Blind Collective Signature Protocol Based on Discrete Logarithm Problem. // *IJ Network Security*. — 2010. — Vol. 11, no. 2. — Pp. 106–113.

135. *Молдовян Н. А.* Протоколы слепой коллективной подписи на основе стандартов цифровой подписи // *Вопросы защиты информации*. — 2010. — № 1. — С. 2–7.

136. Протокол групповой цифровой подписи на основе маскирования открытых ключей / А.А. Молдовян, Н.А. Молдовян, Д.М. Латышев, Д.А. Головачев // *Вопросы защиты информации*. — 2011. — № 3. — С. 2–6.

137. *Moldovyan N. A., Moldovyan A. A.* Group signature protocol based on masking public keys // *Quasigroups and related systems*. — 2014. — Vol. 22, no. 1. — Pp. 133–140.

138. *Брюс Ш.* Прикладная криптография. — Изд. ТРИУМФ, 2003. — С. 816.

139. *Hellman M. E., Pohlig S. C.* Exponentiation cryptographic apparatus and method. — 1984. — US Patent 4,424,414.

140. *Massey J.L., Omura J.K.* Method and apparatus for maintaining the privacy of digital messages conveyed by public transmission. — 1986. — US Patent 4,567,600.

141. *Молдовян А. А., Березин А. Н., Рыжков А. В.* Коммутативные шифры на основе трудности одновременного решения задач факторизации и дискретного логарифмирования // *Информационно-управляющие системы*. — 2014. — № 4 (71). — С. 106–110.

142. *Муравьёв А. В., Березин А. Н., Молдовян Д. Н.* Протокол стойкого шифрования сообщений с использованием коротких ключей // *Известия высших учебных заведений. Приборостроение*. — 2014. — Т. 57, № 11. — С. 68–72.

143. *Молдовян Н. А., Горячев А. А., Муравьёв А. В.* Протокол стойкого шифрования по ключу малого размера // *Вопросы защиты информации*. — 2015. — № 1. — С. 3–8.