Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики

Экз. №

На правах рукописи

Синев Валерий Евгеньевич

Методы построения и разработка практичных протоколов групповой подписи и алгебраических алгоритмов защитных преобразований

05.13.19 – методы и системы защиты информации, информационная безопасность

Диссертация на соискание ученой степени

кандидата технических наук

Научный руководитель
доктор технических наук
профессор Молдовян Н.А.

Санкт-Петербург

2017

Оглавление

Введение

Актуальность темы исследования. Тенденции расширения областей применения информационных технологий, связанных с обработкой, хранением и передачей информации, представленной в цифровом формате, связаны с решением задач обеспечения требуемого уровня информационной безопасности и неотрекаемости (неотказуемости) от содержания электронных сообщений и документов. Решение последней задачи связано с применением электронной цифровой подписи (ЭЦП). Разнообразие информационных технологий, в которых требуется обеспечить неотрекаемость от информации, представленной в электронном виде, определило появление разнообразных типов алгоритмов и протоколов ЭЦП. В случае электронных сообщений и документов, порождаемых коллегиальными органами или коллективами пользователей задача обеспечения неотрекаемости решается с помощью протоколов мультиподписи, которые дают возможность снизить информационную избыточность, связанную с формированием ЭЦП как дополнительного сообщения, присоединяемого к электронному документу. Недостатком известных протоколов мультиподписи является использование нестандартной инфраструктуры открытых ключей и нарушение основополагающего принципа полного недоверия участников протокола ЭЦП друг к другу. Эти недостатки сужают функциональность протоколов ЭЦП, и как следствие, области их применения. Одним из базовых требований к протоколам ЭЦП является их безопасность, т.е. высокая вычислительная сложность подделки цифровой подписи при использовании лучших известных алгоритмов подделки и низкая вероятность появления в обозримом будущем прорывных способов подделки подписи. Для количественной оценки безопасности протоколов ЭЦП используется векторный показатель безопасности в виде пары значений, которые отражают обеспечиваемое значение стойкости и интегральный показатель безопасности, который определяется как отношение обеспечиваемой стойкости к вероятности появления прорывного алгоритма подделки подписи.

Применение алгебраических алгоритмов защитных преобразований для обеспечения информационной безопасности информационно-телекоммуникационных технологий для защиты от атак с принуждением пользователя к раскрытию ключа преобразования требует придания алгоритмам такого типа новых функциональных возможностей. В частности защита от указанных атак потенциально может быть обеспечена разработкой псевдовероятностных алгебраических алгоритмов защитных преобразований позволяющих неоднозначное восстановление преобразованной информации. Тема диссертационного исследования связана с устранением указанных недостатков протоколов обеспечения неотрекаемости и алгоритмов защитных преобразований, что определяет её актуальность.

Степень разработанности темы. В настоящее время теория цифровых подписей является развитой областью современной криптографии и в развитых странах приняты стандарты ЭЦП. Протоколы индивидуальной цифровой подписи нашли широкое применение в современных информационных технологиях. Достаточно хорошо исследован вопрос построения протоколов мультиподписи (групповых, коллективных, агрегированных подписей и др.), однако для их широкого применения требуется решить задачу построения протоколов таких типов с использованием имеющейся инфраструктуры открытых ключей и стандартов ЭЦП. Вопрос использования алгебраических операций в качестве примитивов защитных преобразований блочного типа и конечных некоммутативных групп в качестве примитива криптосхем с открытым ключом затрагивался различными исследователями, однако вопросы разработки псевдовероятностных алгоритмов защитных преобразований алгебраического типа и вопросы использования задачи скрытого дискретного логарифмирования для построения алгоритмов строгой аутентификации не затрагивались.

Цель и задачи исследования. Цель данной работы состоит в расширении функциональности и повышении уровня безопасности протоколов обеспечения неотрекаемости от электронных сообщений и документов и алгоритмов защитных

преобразований. Для достижения этой цели были сформулированы и решены следующие исследовательские задачи**:**

– Разработка метода и построение протокола утверждаемой групповой ЭЦП, обладающего повышенной безопасностью;

– Разработка метода и построение протокола утверждаемой групповой ЭЦП, свободной от использования вспомогательных открытых ключей;

– Построение протокола утверждаемой групповой подписи, функционирующего с использованием стандартной инфраструктуры открытых ключей;

– Разработка метода построения и протокола коллективной ЭЦП, в котором формируется единая подпись для произвольной совокупности групповых подписантов;

– Построение протокола коллективной ЭЦП, в котором формируется единая подпись для произвольной совокупности групповых подписантов и произвольной совокупности индивидуальных подписантов;

– Выполнение оценивания безопасности алгебраических алгоритмов защитных преобразований;

– Разработка метода и построение псевдовероятностных алгебраических алгоритмов защитных преобразований.

Научная новизна диссертационного исследования заключается в следующем:

1. Разработан протокол утверждаемой групповой ЭЦП, основанный на вычислениях по простому модулю и отличающийся выполнением вспомогательной операции возведения в целочисленную степень по трудно разложимому модулю и вычислением рандомизирующих экспонент, маскирующих открытые ключи подписантов, как значения однонаправленной функции в зависимости от открытых ключей подписантов и секретного ключа

руководителя группы подписантов, за счет чего обеспечивается повышение уровня безопасности, обеспечиваемого протоколом.

2. Разработан метод построения протоколов коллективной ЭЦП, отличающийся тем, что рандомизирующий параметр подписи формируется несколькими групповыми подписантами, благодаря чему обеспечивается возможность выработки единой ЭЦП, разделяемой несколькими групповыми подписантами, что дает практически важное расширение функциональности протоколов коллективной подписи.
3. Разработан метод построения протоколов комбинированной коллективной ЭЦП, отличающийся тем, что рандомизирующий параметр подписи формируется несколькими групповыми подписантами и несколькими индивидуальными подписантами, благодаря чему обеспечивается возможность выработки единой ЭЦП, разделяемой несколькими групповыми подписантами и несколькими индивидуальными подписантами, что дает практически важное дополнительное расширение функциональности протоколов групповой подписи.
4. Разработан способ повышения производительности алгебраических псевдовероятностных алгоритмов защитных преобразований, отличающийся представлением блоков преобразуемых данных в виде элементов конечного расширенного поля, заданного в явной векторной форме, благодаря чему обеспечивается повышение производительности алгоритма защитного преобразования.

Теоретическая и практическая значимость работы. Теоретическая значимость работы состоит в разработке новых типов мультиподписи – протоколов коллективной ЭЦП с участием групповых подписантов. Практическая значимость состоит в расширении функциональности протоколов мультиподписи и повышением уровня обеспечиваемой безопасности протоколами утверждаемой групповой ЭЦП. Результаты оценивания безопасности алгебраических алгоритмов защитных преобразований представляют интерес для выбора типа

защитных преобразований при решении практических задач информационной безопасности, а также в учебном процессе.

Методология и методы исследования. В работе использован аппарат и методы математической статистики, теории вероятности, алгебры, теории чисел, криптографии и вычислительные эксперименты. *Объектом* исследования являются информационные технологии; *предметом* – способы, алгоритмы и протоколы обеспечения неотрекаемости от информации, представленной в цифровом формате.

Положения, выносимые на защиту.

1. Метод построения и протокол утверждаемой групповой ЭЦП, построенный на основе вычислений по модулю простого числа с трудно разложимой функцией Эйлера, обеспечивает повышение уровня безопасности протоколов данного типа.

2. Метод построения и протокол коллективной ЭЦП, обеспечивающий формирование единой цифровой подписи, разделяемой произвольным числом групповых подписантов.

3. Метод построения и протокол коллективной ЭЦП, использующий стандартную инфраструктуру открытых ключей и обеспечивающий формирование единой цифровой подписи, разделяемой произвольным числом групповых подписантов и произвольным числом индивидуальных подписантов.

4. Способ псевдовероятностного защитного преобразования информации, отличающийся реализацией вычислений в конечных полях, заданных в явной векторной форме, благодаря чему обеспечивается повышение производительности алгоритма защитного преобразования информации.

Степень достоверности и апробация результатов. Обоснованность научных положений, выводов и практических рекомендаций, полученных в диссертационной работе, обеспечивается анализом состояния исследований в данной области на сегодняшний день, формальными доказательствами,

вычислительным экспериментом и апробацией результатов на следующих конференциях: VI Санкт-Петербургской межрегиональной конференции «Информационная безопасность регионов России (ИБРР-2009)» (Санкт-Петербург, 28 - 30 октября 2009), XI Санкт-Петербургской международной конференции «Региональная информатика-2008 (РИ-2008)» (Санкт-Петербург, 22 - 24 октября 2008), Всеармейской научно-практической конференции «Инновационная деятельность в Вооруженных силах Российской Федерации» (Санкт-Петербург, 25 - 26 ноября 2010), XII Санкт-Петербургской международной конференции «Региональная информатика (РИ-2010)» (Санкт-Петербург, 20 - 22 октября 2010 г), IX Санкт-Петербургской межрегиональной конференции (Санкт-Петербург, 28 - 30 октября 2015 г).

Диссертация включает 5 глав.

В первой главе были рассмотрены известные способы, алгоритмы и протоколы обеспечения неотказуемости от электронных сообщений и документов, используемые в современных информационных технологиях. Представлены известные схемы и протоколы мультиподписи. Выполнена постановка задач диссертационного исследования.

Вторая глава посвящена разработке протоколов слепой и утверждаемой групповой подписи, обладающих повышенным уровнем безопасности. Для построения протокола слепой подписи был использован ранее известный метод, который состоит в использовании простого модуля p, задаваемого в качестве одного из элементов открытого ключа подписанта и имеющего специальную структуру вида $p = 2n + 1$, где $n = qr$, q и r – простые числа, разрядность которых равна или превышает 512 бит и являющиеся элементами личного секретного ключа владельца открытого ключа (подписанта). Известный метод был дополнен механизмом маскирования слепой подписи с использованием операции возведения в степень по модулю n одного из маскирующих параметров.

Третья глава посвящена разработке протоколов коллективной подписи для групповых и индивидуальных подписантов. Достаточно реальным случаем

является разработка электронного документа несколькими организациями, выступающими в роли групповых подписантов. Использование известных протоколов групповой подписи для этого случая связано с формированием нескольких независимых цифровых подписей. С целью обеспечении возможности формирования единой цифровой подписи, по которой можно доказательно проверить, что все ответственные стороны действительно подписали документ, представляет интерес разработка протокола коллективной ЭЦП для групповых подписантов, в которых размер подписи не зависит от числа индивидуальных подписантов. Изучение схем построения коллективной подписи и утверждаемой групповой подписи (УГП) показало, что эти два типа подписей могут быть объединены в едином протоколе коллективной ЭЦП для групповых подписантов. То есть методом реализации протоколов коллективной ЭЦП для групповых подписантов может служить задание процедур формирования долей подписи, генерируемых каждым групповым подписантом, и объединение всех долей подписи в единую подпись в соответствии с механизмами известных протоколов коллективной ЭЦП.

В четвертой главе рассматриваются вопросы разработки алгебраических алгоритмов защитных преобразований для обеспечения информационной безопасности информационных технологий.

Рассмотрены достоинства и недостатки ряда алгебраических операций при их использовании в качестве примитивов защитных преобразований, включая матричное и векторное умножения. Для разработки алгебраических алгоритмов защитных преобразований блочного типа предложен метод комбинирования алгебраических операций из различных алгебраических структур. Описан предложенный способ псевдовероятностного защитного преобразования информации, отличающийся реализацией вычислений в конечных полях, заданных в явной векторной форме.

В пятой главе рассматриваются протоколы с открытым ключом, использующие матричное умножение. Показано, что задача поиска сопрягающего

элемента в конечном некоммутативном кольце матриц, возникающая при осуществлении криптоанализа двухключевого криптоалгоритма Cayley-Purser, решается за полиномиальное время путем сведения к решению системы линейных уравнений, которые легко записываются по значениям параметров открытого ключа данной криптосистемы. Этот результат дает обоснование заключения о том, что алгоритм открытого шифрования Cayley-Purser не удовлетворяет современным требованиям стойкости. При этом достигнуть приемлемого уровня его безопасности не представляется возможным даже при применении секретного и открытого ключа большой разрядности.

Глава 1. Примитивы современных алгоритмов защиты и аутентификации электронных документов и сообщений

В настоящее время вопросы обеспечения информационной безопасности современных информационно-вычислительных и телекоммуникационных систем имеют чрезвычайно важное значение для многих областей человеческой деятельности. Эти вопросы связаны как с защитой конфиденциальной информации, так и с реализацией современных систем электронного документооборота, требующего придания юридической значимости сообщениям и документам, представленным в электронной форме. Последнее обеспечивается с помощью электронной цифровой подписи (ЭЦП). Защита информации, передаваемой по открытым каналам связи, связана с применение криптографических преобразований, использующих секретные ключи [1-4]. Для решения задачи распределения секретных ключей практичным является применение протоколов открытого распределения ключей, основанных на двухключевых криптографических алгоритмах. Функционирование последних обеспечивается наличием пары ключей, связанных между собой, а именно, личного секретного ключа и открытого ключа. Процедуры генерации и проверки подписи в протоколах ЭЦП также основаны на двухключевых криптосистемах [5-7]. В настоящий момент практическое значение криптографии с открытым ключом возросло настолько, что во многих развитых странах внедрена инфраструктура открытых ключей, позволяющая решать задачу проверки подлинности (аутентификации) открытых ключей в достаточно широком масштабе. С технической точки зрения важным представляется обеспечение высокой стойкости алгоритмов шифрования данных, протоколов открытого распределения ключей и схем ЭЦП. Другим важным обстоятельством, имеющим массовый характер, является распространение мобильной связи, карманных персональных компьютеров и смартфонов. Вычислительные ресурсы таких устройств ограничены для выполнения криптографических преобразований. В последних областях применения необходимы прежде всего алгоритмы, которые

бы эффективно использовали память устройств, при этом были более быстрыми и экономичными в потреблении энергии [8-11]. Известно, что криптоалгоритмы открытым ключом работают медленнее по сравнению с симметричными криптоалгоритмами, поэтому при реализации первых в мобильных устройствах следует делать упор на повышении производительности.

Одна из широко применяемых ассиметричных криптосистем, а именно RSA [12,13], низкоэффективна для применения в устройствах с ограниченными вычислительными ресурсами из-за того, что требует оперирования с большими числами, а последнее часто трудно реализовать в виду ограниченности ресурсов. Другими недостатками RSA является довольно медленное выполнение вычислений, особенно при генерации ключей, и необходимость использования более длинных ключей, чем в других криптосистемах с открытым ключом. Это обусловило то, что в мобильных устройствах самые распространенные на данный момент остаются криптоалгоритмы и криптопротоколы, использующие вычисления на эллиптической кривой [14,15], требующие меньше ресурсов, чем RSA. Дальнейшее уменьшение сложности аппаратной реализации и повышение быстродействия криптосхем с открытым ключом все еще остаются направлениями исследований, имеющими важное практическое значение и привлекающими значительный интерес исследователей.

В связи с этим представляет интерес поиск новых примитивов для реализации криптосистемы, которые бы обеспечивали решение поставленной задачи. В качестве таких примитивов представляют интерес конечные некоммутативные группы [16-22]. В таких группах для задания вычислительно трудной задачи дискретного логарифмирования могут выделяться коммутативные подгруппы. Ранее изучались и применялись для построения двухключевых алгоритмов только коммутативные конечные группы. Некоммутативные конечные группы также могут быть применены для синтеза двухключевых алгоритмов, поскольку подгруппы определенного порядка в них являются коммутативными. При этом в силу некоммутативности полной группы можно

ожидать, что для них разработка субэкспоненциальных методов дискретного логарифмирования менее вероятна, чем для коммутативных групп. В связи с этим представляет интерес исследовать условия образования некоммутативных групп, а также разработать двухключевые алгоритмы на их основе.

Кроме того, в настоящее время в криптографии сформировалось направление [23-28] поиска двухключевых криптосистем, обладающих экспоненциальной стойкостью к атакам, основанным на использовании квантовых вычислителей, которые позволят решить за полиномиальное время как задачу нахождения разложения целого числа, так и задачу нахождения дискретного логарифма в конечных циклических группах [29]. Указанные две задачи используются в большинстве применяемых на практике двухключевых криптосхемах, поэтому при предполагаемом появлении в будущем практически действующих квантовых компьютеров, которые смогут быть применены для атаки на криптосистемы, стойкость которых станет полиномиальной, что обусловит неприемлемость их практического применения. Это обусловливает интерес к новым типам вычислительно трудных задач, пригодных для построения криптосистем с открытым ключом, в частности к трудным задачам формулируемым над некоммутативными группами [30-33].

1.1. Двухключевые криптосистемы

В настоящее время большую роль для обеспечения информационной безопасности современных информационных технологий и телекоммуникационных систем играют криптосистемы с открытым ключом. Эти криптосистемы предоставляют методы и алгоритмы решения задач открытого распределения секретных ключей между удаленными пользователями и аутентификации информации, представленной в электронной форме. Как и в любой криптосистеме в основе функционирования криптосистем с открытым ключом лежит использование некоторой секретной информации – секретного ключа. Однако в отличие от классических криптосистем, в которых секретный ключ должен быть известным двум или более пользователям, в криптосистемах с

открытым ключом секретный ключ известен только одному пользователю, который выработал секретный ключ. С этим секретным ключом связан открытый ключ, значение которого зависит от секретного ключа, причем по известному открытому ключу и алгоритму генерации соответствующих друг другу открытого и секретного ключа вычислительно невозможно за обозримое время вычислить секретный ключ. Пользователь, который выработал для себя пару секретного и открытого ключей считается владельцем открытого ключа. Открытый ключ предоставляется для использования всеми пользователями криптосистемы и фактически является общеизвестным. Возможность решения задач распределения общих секретных ключей удаленными пользователями, аутентификации информации и зашифровывания информации обеспечивается корректностью генерации открытого и закрытого ключей и специальными двухключевыми алгоритмами. Благодаря тому, что личный секретный ключ известен только пользователю, который является владельцем открытого ключа, имеется возможность построения эффективных схем цифровой подписи, являющихся алгоритмической основой систем придания юридической силы электронным документам.

Криптографические алгоритмы, в которых используются два ключа одновременно – открытый и закрытый (секретный) называют алгоритмами с открытым ключом или двухключевыми криптоалгоритмами. Подход, где один из ключей известен всем пользователям и потенциальному злоумышленнику, т.е. сама идея использования в криптографическом преобразовании открытого ключа представляется фундаментальной, в связи с чем двухключевые криптосистемы часто называют открытыми шифрами, а производимые защитные преобразования по открытому ключу – открытым шифрованием. Выделяют следующие основные направления применения криптографических преобразований с открытым ключом:

- выработка общего секрета или обмен ключами;

- Выработка и проверка электронной цифровой подписи (аутентификация информации, представленной в цифровом формате);
- аутентификация пользователей;
- построение систем «электронной наличности»;
- построение систем тайного электронного голосования;
- защита материальных объектов от подделки и др.

Протоколы электронной цифровой подписи (ЭЦП) способны служить гарантией, что то или иное сообщение было составлено конкретным абонентом (пользователем) криптосистемы. Строгая доказательность этого факта основана на том, что двухключевые криптосистемы работают при условиях, когда пользователь не передает свой секретный ключ второй стороне. При выработке подписи к электронному документу использование секретного ключа контролируется с помощью открытого ключа. Но для формирования правильной цифровой подписи открытого ключа недостаточно. При этом владельцу ключа нужно понимать, что сохранение в тайне секретного ключа и соблюдения правил его использования являются его личной зоной ответственности. Секретный ключ дает возможность вычислить сообщение со специфической внутренней структурой, которая связана с открытым ключом и подписываемым документом. Это сообщение и называется ЭЦП. С помощью открытого ключа проверяется, что ЭЦП имеет структуру, которая была сформирована с использованием секретного ключа. Вероятность принятия сообщения от нарушителя, за сообщения от пользователя системы ЭЦП, исключительно низка и составляет менее 10^{-20}.

Таким образом, процедура проверки ЭЦП с использованием открытого ключа даёт высокую степень гарантии, что принятое сообщение было составлено владельцем секретного ключа. Такие свойства двухключевых криптографических систем лежат в основе правовых актов, придающих юридическую силу ЭЦП. При вводе в действие таких правовых актов обеспечивается возможность придания юридической силы электронным документам с помощью протоколов ЭЦП.

Поскольку только владелец личного секретного ключа может вычислить цифровую подпись к некоторому сообщению, такую, что она легко может быть проверена любым заинтересованным лицом по открытому ключу владельца, то не возникает трудности в проверке подлинности цифровой подписи. Это дает принципиальную возможность предотвратить отказ от авторства электронного сообщения или документа, т.е. отрицать свою связь с посланным сообщением. Например, предотвращение отказа от авторства является одним из важнейших требований в технологиях электронной коммерции. Открытый общедоступный ключ вычисляется в зависимости от личного секретного так, что вычисление личного секретного ключа по открытому ключу владельца оказывается вычислительно сложной (практически невыполнимой) задачей.

В основе криптосистем с открытым ключом лежат различные вычислительно сложные задачи. Криптосистемы такого типа по определению не могут являться безусловно стойкими в понимании смысла данного термина в рамках модели нарушителя, обладающего бесконечными вычислительными ресурсами. Применение криптосистем с открытым ключом основано на том, что они облают практической стойкостью, т.е. их взлом с использованием любой известной атаки является вычислительно невыполнимым за обозримое время. Чтобы обеспечить выполнимость требования практической стойкости, параметры вычислительно трудной задачи, лежащей в основе работы двухключевого криптоалгоритма, выбираются достаточно большого размера. При этом с ростом размера (битовой длины, разрядности) используемых параметров снижается производительность (вычислительная эффективность) двухключевых криптосхем. Это обусловливает предпочтительность использования криптоалгоритмов, основанных на задачах, которые обеспечивают достаточно высокий уровень стойкости (криптостойкости) при достаточно низкой вычислительной сложности процедур, выполняемых при осуществления криптографических преобразований. Предпочтительным для построения криптосхем с открытым ключом является выбор вычислительно сложных задач, для которых зависимость вычислительной сложности криптографических преобразований полиномиально зависит от

размера параметров задачи, а сложность решения базовой вычислительной задачи зависит экспоненциально или, по крайней мере, сверхполиномиально (под сверхполиномиальной зависимостью понимается то, что при выборе достаточного размера параметров базовой задачи ее вычислительная сложность растет более быстро по сравнению с любым заранее заданным полиномом).

Наиболее широкое применение для разработки криптографических алгоритмов и протоколов получили следующие вычислительно трудные задачи:

- задача факторизации больших целых чисел, включающих два множителя, представляющих собой простые числа, удовлетворяющие требованиям «сильной простоты» [13,34];
- задача дискретного логарифмирования в конечных ассоциативных алгебраических структурах (полях, циклических группах) [35,36];
- задача дискретного логарифмирования на эллиптической кривой (т.е. в циклических группах точек эллиптической кривой) [37-40];
- задача извлечения квадратного корня по трудно разложимому модулю [41,42];
- задача извлечения корней большой простой степени в циклических группах, порядок которых делится на квадрат степени корня [43-46].

1.2. Схемы открытого согласования ключей

Под схемой открытого распределения ключей понимают криптографический протокол, в рамках которого используется двухключевой криптоалгоритм и который решает задачу формирования общего секретного ключа для двух или более удаленных пользователей при обмене некоторыми несекретными сообщениями по открытому каналу. Применяются два типа таких схем – протоколы открытого согласования ключей и протоколы открытого распределения ключей, хотя два последних названия часто используются как синонимы. Однако при более строгом использовании данных терминов

учитывается то, что эти варианты отличаются тем, как именно формируется и распределяется общий секретный ключ. В схеме открытого согласования ключа ни один из пользователей не знает заранее какое конкретное значение секретного ключа будет сформировано в результате выполнения протокола. Значение секретного ключа зависит от выбора некоторых случайных значений каждой из сторон протокола, причем все стороны, участвующие в протоколе, используя получаемые сообщения вырабатывают одно и то же секретное значение, которое принимается в качестве общего секретного ключа. Нарушитель, перехватывающий все передаваемые данные не может вычислить секретное значение.

В схеме открытого распределения ключей один из пользователей генерирует секретный ключ и, используя открытые ключи других пользователей, участвующих в протоколе, зашифровывает секретный ключ и рассылает полученные криптограммы другим пользователям. Рассмотрим конкретные реализации таких протоколов.

Система открытого согласования ключей Диффи-Хеллмана. Становление и бурное развитие двухключевой криптографии началось с появления статьи Диффи и Хеллмана [36], в которой была предложена криптосхема, основанная на новом подходе к решению задачи распределения секретных ключей. В основу предложенной ими криптосхемы была положена хорошо известная вычислительно трудная задача дискретного логарифмирования в мультипликативной группе конечного простого поля. В предложенной схеме личным секретным ключом, который не подлежит рассылке, является случайное число x достаточно большого размера, а открытый ключ y формируется, путем выполнения операции возведения в степень, равную большому натуральному числу x, по модулю большого простого числа

$$y = \alpha^x \bmod p,$$

где x – целое число, такое, что $1 < x < p–1$, p – простое k-битовое число, α – примитивный элемент по модулю p [47,48]. Оказалось, что используя данную формулу, имеется возможность построения практически стойких криптографических систем, в которых *не требуется передача секретного ключа*. Обеспечение удаленных пользователей одинаковым секретным ключом реализуется с применением только открытых несекретных сообщений.

Механизм согласования общего секретного ключа двух удаленных абонентов телекоммуникационной системы с открытыми каналами связи реализуется следующим образом. Каждый пользователь генерирует свой личный секретный ключ в виде случайного натурального числа x и по последнему вычисляет свой открытый ключ y, соответствующий выбранному секретному ключу, используя указанную выше формулу. Пользователи А и В формируют общий секретный ключ без передачи каких-либо секретных значений следующим путем. Пользователь А берет из справочника открытых ключей (доступного на сайте удостоверяющего центра) открытый ключ y_B пользователя В и, используя свой личный секретный ключ x_A, вычисляет общий секретный ключ:

$$Z_{AB} = (y_B)^{x_A} = (\alpha^{x_B})^{x_A} = \alpha^{x_B x_A} \bmod p.$$

Аналогичные действия выполняет и пользователь В:

$$Z_{AB} = (y_A)^{x_B} = (\alpha^{x_A})^{x_B} = \alpha^{x_B x_A} \bmod p.$$

В результате этих шагов оба пользователя сформировали одинаковый секретный ключ Z_{AB} без использования какого-либо заранее установленного общего секретного ключа. Используя сформированный общий секретный ключ в качестве ключа шифрования и некоторую одноключевую криптосистему, пользователи могут зашифровывать направляемые друг другу сообщения, т.е. установить секретную связь по открытым каналам. Данная процедура согласования общего секретного ключа имеет достаточно высокую вычислительную эффективность для достаточно больших длин чисел p, α, y и x (например, разрядность этих чисел может быть тысячи и десятки тысяч бит).

Для оценки стойкости данной криптосхемы следует рассмотреть возможные действия нарушителя. Ему известны значения $y_B = \alpha^{x_B} \bmod p$ и $y_A = \alpha^{x_A} \bmod p$, но для вычисления значения Z_{AB}, он должен решить задачу дискретного логарифмирования и определить либо x_A, либо x_B. Это означает. Что нарушитель должен выполнить операцию, обратную операции возведения в степень, т.е. ему надо решить задачу дискретного логарифмирования по достаточно большому простому модулю. Известно, что для достаточно больших простых чисел p, таких, что в разложении на множители числа p–1 содержится простой делитель q достаточно большой разрядности, задача дискретного логарифмирования является вычислительно сложной (практически неосуществимой). На данном уровне развития вычислительной техники задача дискретного логарифмирования вычислительно невыполнима за обозримое время при длине числа p более 1536 бит и длине числа q более 240 бит.

При данных размерах чисел p и q операция возведение в большую целочисленную степень по модулю p выполняется очень быстро при использовании алгоритма быстрого возведения в степень [41,49], что делает схему открытого согласования ключей Диффи-Хеллмана весьма практичной, поскольку она обладает достаточно высоким быстродействием и высокой криптостойкостью.

Авторы данной схемы ввели понятие справочника открытых ключей и указали на то, что не следует упускать из виду проблему аутентификации открытых ключей. Стойкость схемы открытого согласования ключа может быть обеспечена только в случае, если все открытые ключи в справочнике открытых ключей являются подлинными. Участники протокола должны выполнить процедуру проверки подлинности открытых ключей друг друга. Решение задачи аутентификации открытых ключей решается организационно-техническими мерами.

Обобщенная схема открытого распределения ключей. Об открытом распределении ключей говорят, когда секретный ключ генерируется у одного из

пользователей и потом зашифровывается по открытому ключу получателя и направляется получателю по открытому каналу. Получатель, используя свой личный секретный ключ, расшифровывает полученный шифртекст и тем самым извлекает из шифртекста секретный ключ. В криптосхемах такого типа используется некоторый алгоритм открытого шифрования, название которого связано с тем, что зашифровывание сообщений выполняется по открытому ключу получателя сообщения, а расшифровывание – по личному секретному ключу получателя. В этом случае имеется возможность направлять секретные сообщения всем пользователям, подлинные открытые ключи которых известны отправителю. Для расшифровывания полученных текстов требуются секретные ключи, соответствующие открытым ключам, использованным при зашифровывании сообщения. Пусть известен алгоритм открытого шифрования E, такой, что процедуры шифрования по открытому и по соответствующему секретному ключу являются взаимно обратными. Пусть также известен открытый ключ e некоторого пользователя, причем подлинность ключа e подтверждена некоторой выполненной процедурой аутентификации ключа e (например он извлечен из цифрового сертификата, подписанного удостоверяющим центром). Тогда некоторое секретное сообщение Z может быть передано по открытым каналам в виде криптограммы $C = E_e(Z)$ владельцу открытого ключа e. Значение Z извлекается, выполняя шифрование по личному секретному ключу d, т.е. осуществляя обратное преобразование:

$$E_d(C) = E_e^{-1}(C) = E_e^{-1}(E_e(Z)) = Z.$$

Поскольку значение d известно только получателю, то только он может извлечь из криптограммы значение Z. В качестве Z может быть направлен получателю некоторый секретный ключ. Получив некоторый секретный ключ, получатель должен убедиться, что он не был направлен ему от нарушителя, который также может воспользоваться общедоступным открытым ключом e. Для того, чтобы процедура аутентификации значения Z могла быть выполнена следует усилить рассматриваемую криптосхему. Это делается путем добавления шага

шифрования по личному секретному ключу отправителя. В результате этого протокол открытого распределения ключей между пользователями A, B и C приобретает следующий вид.

1. Один из пользователей, например, пользователь A, генерирует секретный ключ Z.

2. Используя открытые ключи пользователей B и C, пользователь A вычисляет криптограммы

$$C_{\mathrm{C}} = E_{d_{\mathrm{A}}}\left(E_{e_{\mathrm{C}}}(Z)\right) \text{ и } C_{\mathrm{B}} = E_{d_{\mathrm{A}}}\left(E_{e_{\mathrm{B}}}(Z)\right).$$

3. Пользователь A отправляет криптограмму C_{B} пользователю B, а криптограмму C_{C} пользователю C.

4. Получив криптограмму C_{B}, пользователь B вычисляет значение

$$E_{d_{\mathrm{B}}}\left(E_{e_{\mathrm{A}}}(C_{\mathrm{B}})\right) = Z.$$

5. Получив криптограмму C_{C}, пользователь C вычисляет значение

$$E_{d_{\mathrm{C}}}\left(E_{e_{\mathrm{A}}}(C_{\mathrm{C}})\right) = Z.$$

Результатом выполнения этого протокола является появление общего секретного ключа, сгенерированного пользователем A, у пользователей B и C. Причем они уверенны в следующих фактах:

- значение Z неизвестно никому другому, кроме пользователя A, если оно было направлено пользователем A;
- если значение Z действительно позволяет восстанавливать получаемые сообщения, то ключ Z, был действительно направлен пользователем A.

Некоторым недостатком последней версии протокола открытого распределения ключей является «отложенность» факта подтверждения подлинности отправителя. В следующей версии протокола устраняется этот недостаток.

Обобщенный протокол открытого распределения ключей

1. Один из пользователей, например, пользователь A, генерирует секретный ключ Z.

2. Используя открытые ключи пользователей B и C, пользователь A вычисляет криптограммы

$$C_{\mathrm{C}} = E_{d_{\mathrm{A}}}\left(E_{e_{\mathrm{C}}}(Z), e_{\mathrm{A}}\right) \text{ и } C_{\mathrm{B}} = E_{d_{\mathrm{A}}}\left(E_{e_{\mathrm{B}}}(Z), e_{\mathrm{A}}\right).$$

3. Пользователь A отправляет криптограмму C_{B} пользователю B, а криптограмму C_{C} пользователю C.

4. Получив криптограмму C_{B}, пользователь B вычисляет значение

$$E_{e_{\mathrm{A}}}(C_{\mathrm{B}}) = \left(E_{e_{\mathrm{B}}}(Z), e_{\mathrm{A}}\right),$$

убеждается, что правая часть сообщения промежуточного сообщения $\left(E_{e_{\mathrm{B}}}(Z), e_{\mathrm{A}}\right)$ представляет собой открытый ключ отправителя (или некоторый другой специфицированный идентификатор), т.е. убеждается в подлинности отправителя, а затем вычисляет секретный ключ Z:

$$E_{d_{\mathrm{B}}}\left(E_{e_{\mathrm{B}}}(C_{\mathrm{B}})\right) = Z.$$

5. Получив криптограмму C_{C}, пользователь C вычисляет значение

$$E_{e_{\mathrm{A}}}(C_{\mathrm{C}}) = \left(E_{e_{\mathrm{C}}}(Z), e_{\mathrm{A}}\right),$$

убеждается, что правая часть сообщения $\left(E_{e_{\mathrm{C}}}(Z), e_{\mathrm{A}}\right)$ представляет собой открытый ключ пользователя A (или некоторый другой специфицированный идентификатор), т.е. убеждается в подлинности пользователя A, а затем вычисляет секретный ключ Z:

$$E_{d_{\mathrm{B}}}\left(E_{e_{\mathrm{B}}}(C_{\mathrm{B}})\right) = Z.$$

Далее будет рассмотрена криптосистема RSA [12], которая предоставляет конкретную функцию E, которая может быть положена в основу конкретной реализации данного обобщенного протокола распределения секретных ключей по открытым каналам связи.

Двухключевые шифры по сравнению с одноключевыми криптосистемами дают скорость шифрования на несколько порядков ниже. Поэтому наибольшей эффективностью обладают гибридные криптосистемы, которые комбинируют симметричные и асимметричные криптосхемы следующим образом: информационные сообщения шифруются с помощью симметричных (одноключевых) криптоалгоритмов, а распределение ключей симметричного шифрования выполняются по открытому каналу, используя двухключевые криптосхемы. В частности, при использовании криптосистемы RSA [12,13], можно выполнить обмен сеансовым ключом с другим пользователем, зашифровав сеансовый ключ с помощью его открытого ключа. При этом безопасно передать зашифрованный сеансовый ключ по открытому каналу связи, так как секретный ключ необходимый для расшифровки есть только у пользователя, открытый ключ которого использовался для зашифровки. Двухключевые шифры для непосредственного засекречивания информации находят узкое применение.

1.3. Протоколы электронной цифровой подписи

Криптосистема RSA. Данная криптосистема впервые была обнародована в работе [12]. Эта криптосистема является первой широко известной и практически используемой системой цифровой подписи и открытого шифрования. В основе ее работы лежит теорема Эйлера [47], которая утверждает, что для любых двух взаимно простых натуральных чисел n и $M < n$ справедливо соотношение

$$M^{\varphi(n)} = 1 \bmod n,$$

где $\varphi(n)$ – функция Эйлера, значение которой определяется как количество чисел, взаимно простых с n и не превосходящих n. Модуль n выбирается таким образом, чтобы его факторизация была вычислительно невыполнимой операцией.

Делители n составляют часть личного секретного ключа пользователя, а его открытым ключом значение n и некоторое другое число e, взаимно простое с $\varphi(n)$. Значение n генерируется по формуле $n = pq$, где оба множителя являются сильными простыми числами. Требование сильной простоты делителей p и q [50] обеспечивает практическую невозможность факторизации числа n. Для генерации случайных сильных простых чисел большой разрядности (512, 1024, 2048 бит и более) известны вычислительно эффективные алгоритмы, поэтому пользователи без труда могут сгенерировать значение n, факторизовать которое будет практически невозможно, даже при использовании вычислительных ресурсов всего человечества. Основной операцией преобразования в криптосистеме RSA является операция возведения в степень по модулю n.

Рассмотрим процедуру открытого шифрования. Пусть требуется зашифровать сообщение M (на шифруемые сообщения накладывается требование $M < n$). Эта процедура выполняется по открытому ключу в виде пары чисел (n, e) по следующей формуле

$$C = E_e(M) = M^e \bmod n.$$

Расшифровывание криптограммы C состоит в извлечении корня e-ой степени по модулю n. Это выполняется как операция возведения в степень d (ключ расшифровывания), равную $e^{-1} \bmod \varphi(n)$:

$$E_d(C) = C^d \bmod n = (M^e \bmod n)^d \bmod n = M^{ed} \bmod n = M.$$

Непосредственно из теоремы Эйлера следует, что две процедуры возведения в степень по модулю n будут представлять собой взаимно обратные преобразования, если произведение степеней, используемых в этих процедурах, сравнимо с единицей по модулю, равному функции Эйлера от числа n: $ed \equiv 1 \bmod \varphi(n)$. Если выбрано некоторое значение e, то для расшифровывания криптограммы требуется вычислить параметр $d = e^{-1} \bmod \varphi(n)$. Для вычисления последнего значения требуется знать разложение числа n, поэтому по открытому ключу зашифровывания e никто другой, кроме владельца открытого ключа (n, e),

не имеет возможности вычислить секретную экспоненту d. Параметр d является элементом личного секретного ключа того пользователя, который сгенерировал открытый ключ (n, e). Для вычисления экспоненты d владелец открытого ключа поступает следующим образом. Сначала он вычисляет значение функции Эйлера от модуля n. Поскольку для простых значений p и q имеет место $\varphi(p) = p - 1$ и $\varphi(q) = q - 1$, а модуль равен $n = pq$, то, используя свойство мультипликативности функции Эйлера, владелец открытого ключа (n, e) легко вычисляет значение $\varphi(n) = \varphi(pq) = \varphi(p)\varphi(q) = (p - 1)(q - 1)$, а затем для любого числа e определяет требуемую экспоненту расшифровывания $d = e^{-1} \bmod \varphi(n)$, которую сохраняет в тайне (для последнего он использует расширенный алгоритм Евклида [51,52]).

Таким образом, в криптосистеме RSA предусматривается следующее. Каждый пользователь выбирает два многоразрядных простых числа p и q, разность которых также является многоразрядным числом, и находит их произведение $n = pq$. После этого он вычисляет значение функции Эйлера $\varphi(n)$ и генерирует случайное число e, такое, что НОД$(e, \varphi(n)) = 1$. Затем он вычисляет обратное (по модулю $\varphi(n)$) к e значение $d = e^{-1} \bmod \varphi(n)$. После этого он может уничтожить делители q и p, что будет способствовать сохранению их в тайне. *Секретным ключом* является число d. Пара чисел n и e является *открытым ключом*, который предоставляется всем абонентам криптосистемы RSA.

В этой криптосхеме процедуры зашифровывания и расшифровывания являются симметричными, поэтому, если выполнить расшифровывание некоторого сообщения, то это даст некоторое значение, из которого можно восстановить исходное сообщение по открытому ключу, что может сделать любой желающий, используя открытый ключ. С точки зрения обеспечения секретности пересылаемых сообщений возведение сообщения в секретную степень не имеет смысла (поскольку обратное преобразование выполняется по открытому ключу), однако значение которое даст такое преобразование не может быть получено по открытому ключу. При этом по открытому ключу можно легко проверить, что некоторое значение было получено путем возведения некоторого сообщения в

секретную степень d. Легко понять, что это свойство может быть использовано для аутентификации источника сообщения по открытому ключу. Другими словами, криптосхема RSA реализует не только возможность выполнения открытого шифрования, но и возможность вычисления некоторых чисел, которые фактически представляют собой цифровую подпись к сообщениям, представленным в цифровом виде. При этом подлинность подписи может быть проверена любым лицом, используя открытый ключ подписанта.

Процедура генерации ЭЦП. Пусть требуется сформировать ЭЦП к некоторому заданному сообщению $M < n$. Цифровая подпись может быть вычислена путем возведения числа M в степень d по модулю n в соответствии с формулой

$$S = M^d \bmod n.$$

Значение S есть ЭЦП, которую для заданного сообщения M может выработать только владелец секретного ключа.

Процедура проверки подлинности ЭЦП. Подлинность цифровой подписи S к сообщению M проверяется как возведение числа S в степень открытого ключа e по модулю n по формуле

$$M' = S^e \bmod n.$$

Если результат этой операции дает значение M' равное заданному сообщению M, то ЭЦП S признается подлинной.

Криптостойкость схемы RSA основана на сложности факторизации модуля. Если эту задачу решить, то тогда можно вычислить функцию Эйлера от модуля и затем определить секретный ключ по открытому. Однако до настоящего времени не предложены алгоритмы решения вычислительно сложной задачи факторизации модуля рассматриваемого вида, имеющие полиномиальную сложность при использовании имеющихся вычислительных ресурсов, если размер

модуля равен 1536 бит и более и если делители модуля p и q представляю собой сильные простые числа примерно одинаковой длины.

В системе RSA необходимо учитывать также возможность экзистенциальной подделки подписи [53,54], которая состоит в выборе произвольного значения в качестве цифровой подписи и формировании по этой подписи сообщения. На самом деле, взяв некоторое произвольное число S, можно легко вычислить значение M', удовлетворяющее условию $M' = S^e$, т. е. произвольное значение можно попытаться выдать за цифровую подпись к некоторому случайному сообщению. Обычно случайные сообщения подписывать нет смысла в большинстве практических случаях использования протоколов ЭЦП. Однако, в ряде приложений иногда требуется подписывать случайные значения. Для устранения возможности экзистенциальной подделки ЭЦП и обеспечения возможности вычисления подписи к сообщениям произвольного размера в практически применяемых вариантах криптосхемы RSA подпись формируется не непосредственно по значению сообщения, а по значению хэш-функции, которая вычисляется от сообщения по некоторому алгоритму хэширования, который специфицируется как элемент протокола ЭЦП.

В настоящее время наибольшее распространение получили криптосхемы, основанные на трудности задачи дискретного логарифмирования. Впервые эта задача была применена в схеме цифровой подписи Эль-Гамаля [55].

Схема ЭЦП Эль-Гамаля [55] включает следующую процедуру генерации ключей.

1. Выбрать случайное достаточно большое простое число p, такое, что задача дискретного логарифмирования в конечном простом поле $GF(p)$ является практически нерешаемой.

2. Вычислить примитивный элемент α поля $GF(p)$, т.е. число для которого имеет место $\alpha^{(p-1)} \equiv 1 \bmod p$, причем для всех нетривиальных делителей $r \mid p-1$ не выполняется неравенство $\alpha^{(p-1)/r} \bmod p \neq 1$.

3. Сформировать случайное натуральное число $x < p - 1$, x – личный секретный ключ.

4. Используя значение x, вычислить значение открытого ключа y по следующей формуле: $y = \alpha^x \bmod p$.

5. В качестве *открытых параметров* криптосхемы используются числа α и p. Секретный ключ x необходимо хранить в тайне.

Генерация подписи включает следующие шаги:

1. Сгенерировать случайное секретное число k, удовлетворяющее условию $0 < k < p - 1$ и НОД(k, p –1) = 1 (k играет роль разового секретного ключа).

2. Вычислить параметр рандомизации цифровой подписи $R = \alpha^k \bmod p$ (R играет роль разового открытого ключа и не является секретным).

3. Вычислить значение S из следующего уравнения:

$$M = xR + kS \bmod p - 1 \quad \Rightarrow \quad S = \frac{M - xR}{k} \bmod p - 1$$

4. В качестве цифровой подписи к сообщению M берется пара натуральных чисел (R, S).

Процедура проверки подлинности ЭЦП осуществляется в соответствии со следующей процедурой:

$$Verify_{(y,\alpha,p)}(m,(R,S)) = True, \text{ если } r < p \text{ и } y^R R^S \equiv \alpha^M \bmod p.$$

Корректность схемы ЭЦП Эль-Гамаля доказывается подстановкой сгенерированной ЭЦП в проверочное уравнение процедуры верификации ЭЦП:

$$\alpha^M = y^R R^S = (\alpha^x)^R (\alpha^k)^{\frac{M-xR}{k}} = \alpha^{xR} \alpha^M \alpha^{-xR} = \alpha^M \bmod p.$$

Для того, чтобы обеспечить возможность вычисления второго элемента подписи, а именно, значения S, необходимо обеспечить условие взаимной простоты чисел k и $p - 1$ (данное требование вытекает из известной теоремы из теории сравнений о существовании обратного элемента). В связи с этим при генерации числа k требуется обеспечить выполнимость условия НОД(k, p –1) = 1. Число k должно уничтожаться после формирования значения подписи, поскольку по известному k и известной подписи не составляет труда вычислить секретный ключ.

После опубликования статьи Эль-Гамаля, в которой был представлен алгоритм цифровой подписи, основанный на задаче дискретного логарифмирования, было предложено множество вариаций реализации его идеи использования рандомизирующего значения в процедуре генерации подписи. Наиболее значительными из них является схема цифровой подписи Шнорра [56]. Также представляют интерес схемы ЭЦП на основе трудности дискретного логарифмирования в конечных простых полях, в которых нет возможности выделения одного из элементов подписи как параметра рандомизации, предложенные в работах [57-59].

Схема ЭЦП Шнорра [56] имеет существенные особенности по сравнению с алгоритмом цифровой подписи Эль-Гамаля. Важным достоинством схемы Шнорра является достаточно малый размер подписи (320 бит при обеспечении 80-битового уровня стойкости, т.е. трудоемкости взлома не менее 2^{80} операций модульного умножения), что представляется важным для многих практических применений. Для обеспечения 80-битовой стойкости ЭЦП в схеме Эль-Гамаля требуется использовать 1024-битовый модуль, что дает размер ЭЦП, равный 2048 бит. Теоретически важной особенностью схемы Шнорра является то, что значение хэш-функции вычисляется принудительно только после генерации параметра рандомизации R. Благодаря этой особенности для схемы Шнорра может быть формально доказано, что существование эффективных алгоритмов подделки подписи означает существование эффективных алгоритмов решения задачи

дискретного логарифмирования [60], т.е. имеется возможность теоретического доказательства ее стойкости в смысле доказательства того, что сложность взлома не проще решения трудной задачи дискретного логарифмирования, положенной в основу криптосхемы.

Более компактное представление значения ЭЦП достигается в схеме Шнорра с помощью конструирования поля F_p, содержащего намного подгруппу заданного простого порядка q. В схеме ЭЦП Эль-Гамаля безопасной длиной параметра p считается 1024 бит, что определяется возможностью решения задачи дискретного логарифмирования методом вычисления индексов [49], имеющих субэкспоненциальную сложность. Поскольку трудоёмкость этого метода определяется размером характеристики поля и практически не зависит от размера порядка основания логарифма, то имеется возможность уменьшения размера порядка α до $|q| \approx 160$ бит. При росте размера $|p|$ потребуется увеличит размер $|q|$, чтобы обеспечить рост стойкости, однако при этом отношение размера $|p|$ к размеру $|q|$ будет возрастать, т.е. схема Шнорра при усилении требований по стойкости будет еще более предпочтительна. Рассмотрим схему ЭЦП Шнорра.

Схема ЭЦП Шнорра [61] включает следующую процедуру генерации ключей.

1. Сгенерировать простые числа p, q, такие что $q | p - 1$; $|p| \approx 1024$ бит, $|q| \approx 160$ бит.

2. Сгенерировать элемент $\alpha \in GF(p)$ порядка q, т.е. $\alpha^q \equiv 1 \bmod p$.

3. Выбрать криптографически стойкую функцию хэширования F_h.

4. Сгенерировать случайное число x (секретный ключ) такое, что для него выполняется соотношение $1 < x < q$.

5. Вычислить $y = \alpha^x \bmod p$.

Личным секретным ключом является x, а о*ткрытым* ключом – значение y. Системными параметрами схемы Шнорра являются значения p, q и α.

Генерация цифровой подписи к сообщению *M*

1. Вырабатывается случайное натуральное число *k*, удовлетворяющее условию $1 < k < q$.

2. Вычисляется значение параметра рандомизации $R = \alpha^k \bmod p$.

3. К подписываемому сообщению *M* присоединяется параметр рандомизации и вычисляется значение хэш-функции F_h от аргумента $M||R$:

$E = H(M||R)$ – первый элемент ЭЦП.

4. Вычисляется значение *S*, которое представляет собой второй элемент ЭЦП:

$$S = k + xE \bmod q.$$

Процедура проверки подлинности подписи включает следующие шаги:

1. Вычисляется значение R′: $R' = \alpha^S y^E \bmod p$

2. К сообщению *M* присоединяется число *R′* (т.е. получаем аргумент $M||R'$) и вычисляется значение хэш-функции $F_h(M||R')$: $E' = F_h(M||R)$

3. Проверяется равенство значений *E′* и *E*. Если последние два значения равны, то подпись признается подлинной.

Корректность работы схемы Шнорра показывается следующим образом. Пусть к сообщению *M* приложена подпись (*S, E*), полученная по правильному значению секретного ключа и в соответствии с процедурой генерации подписи, специфицированной для схемы Шнорра. Тогда имеем:

$$R' = \alpha^S y^E \bmod p = \alpha^{k+xE}\alpha^{-xE} = \alpha^k = R \bmod p.$$

Так как $R' = R$ *E′*, то $E' = E$, т.е. процедура проверки ЭЦП подтвердит подлинность подписи. Аналогично криптосхеме Эль-Гамаля, параметр *k* фактически используется в качестве разового секретного ключа и поэтому должен генерироваться как равновероятное случайное число, которое должно уничтожаться непосредственно после вычисления значения ЭЦП.

Одним из ключевых моментов при реализации схем ЭЦП является применение хэш-функций, которые служат для обеспечения контроля целостности подписываемых сообщений и документов [62]. Наиболее полно требованиям, предъявляемым к криптографически стойким методам хэширования информации, отвечают однонаправленные хэш-функции без секрета [52]. Различные виды хэш-функций описаны в международном стандарте ISO/IEC 10118, ряде национальных стандартов и криптографической литературе. Хэш-функция предназначена для сжатия подписываемого документа в криптографически стойкую контрольную сумму размером от 160 бит до 512 бит, причем для данного алгоритма вычисления значения хэш-функции размер ее значения является фиксированным (чем больше размер значения хэш-функции тем более высокая стойкость может быть достигнута).

В качестве аргумента хэш-функция $F_h(M)$ могут браться сообщения произвольной длины M, а результатом вычисления хэш-значения является битовая строка h фиксированной длины $F_h(M) = h$. Значение хэш-функции $F_h(M)$ зависит от каждого бита хэшируемого сообщения M и практически не позволяет получить два или более сообщений с одинаковым значением хэш-функции.

Хэш-функция должна удовлетворять ряду условий [63]:

- ее значение должно зависеть от каждого бита сообщения-аргумента M по псевдослучайному закону, т.е. она должна изменяться при удалении, исправлении, вставке и перестановке любых битов, слов, предложений и т.п.;

- должна обладать свойством вычислительной необратимости, т.е. задача вычисления документа M, который обладал бы заданным наперед случайным значением хэш-функции, должна быть вычислительно невыполнимой;

- должна удовлетворять требованию коллизионной стойкости, которое состоит в том, что вычислительно неосуществимо нахождение каких-либо двух сообщений, обладающих одинаковыми значениями хэш-функции;

- вероятность того, что значение хэш-функции двух различных документов совпадут, должна быть ничтожно мала.

1.4. Постквантовая криптография и трудные задачи над некоммутативными группами

В большинстве современных продуктов и стандартов криптографии применяются методы с открытым ключом, основанные на проблеме факторизации больших чисел (RSA) [13] и дискретного логарифмирования (стандарты DSA [64], ECDSA [65], ГОСТ Р 34.10-94 [66] и ГОСТ Р 34.10-2001 [67]). Подход [68,69,115] к синтезу двухключевых криптосистем на основе эллиптических кривых (используется трудность задачи дискретного логарифмирования на эллиптической кривой (ЭК), т.е. в конечной группе точек ЭК) обеспечивает эквивалентную защиту при меньшем числе разрядов открытого ключа и других параметров криптосхемы по сравнению с ранее разработанными протоколами. Сложность атаки, как правило, связывают с зависимостью количества операций, необходимых для решения трудной задачи, положенной в основу криптосхемы. По виду функции, описывающей нарастание сложности решения трудных задач, различают полиномиальную, субэкспоненциальную и экспоненциальную сложность. Под видом функции здесь понимается формула, описывающая зависимость роста вычислительной сложности трудной задачи от размера задачи [70] (размера параметров, входящих в формулировку вычислительной задачи).

Стойкость криптосхем на основе специально выбранных ЭК экспоненциально связана с длиной ключа. Стойкость же RSA, основанной на сложности факторизации – субэкспоненциальная. Это позволяет использовать открытые ключи и другие открытые параметры криптосхем на основе ЭК значительно меньшей длины, чем для криптосхемы RSA. Размер параметров, равный 160 бит в первом случае, обеспечивает примерно такой же уровень безопасности, как RSA с параметрами размером 1024 бит. Кроме того, увеличение

длины ключа для схем на основе ЭК в два раза приводит к значительно большему увеличению криптостойкости, чем увеличение в два раза длины параметров RSA, поэтому в будущем преимущества алгоритмов и протоколов эллиптической криптографии еще более только возрастут. Также увеличение длины ключа и уменьшение разрядности процессора приводит к увеличению преимущества по производительности криптосхем на основе ЭК. При этом аналогичная сравнительная картина имеет место при сопоставлении криптосхем, заданных над конечными полями (схема Эль-Гамаля [55], схема Шнорра [56]), по сравнению со схемами эллиптической криптографии. Однако, если давать сравнение стойкости различных схем к атакам, основанным на предполагаемой возможности использований квантового компьютера, то все перечисленные в данном параграфе криптосхемы окажутся нестойкими, т.е. непригодными для практического использования ни при каких размерах параметров, так как квантовый компьютер решает задачу факторизации и задачу дискретного логарифмирования в циклической подгруппе любого типа за полиномиальное время при использовании алгоритма Шора [29].

Последнее обстоятельство, а также ожидание, что в ближайшем будущем появятся реально действующие квантовые вычислители, а значит и возможность их применения для выполнения криптоанализа двухключевых криптосистем, обусловило большой интерес к другим типам задач, на основе которых могут быть разработаны протоколы ЭЦП [17,24,28], открытого распределения ключей [24], алгоритмы открытого и коммутативного шифрования [25,26,28], которые основаны на вычислительно трудных задачах, имеющих экспоненциальную сложность при выполнении вычислений на квантовом компьютере.

Большое внимание разработчиков привлекают трудные задачи, формулируемые над некоммутативными группами. Заслуживают внимание вычислительная задача поиска сопрягающего элемента в некоммутативных группах кос (группах переплетения) [21-23], а также вычислительно трудная задача дискретного логарифмирования в маскируемой циклической подгруппе,

которая представляет собой комбинирование задачи дискретного логарифмирования и задачи поиска сопрягающего элемента и задается над конечными некоммутативными группами матриц и векторов [26,28].

Задача поиска сопрягающего элемента состоит в следующем. Пусть известно значение $Y = X \circ G \circ X^{-1}$, где X – неизвестный элемент некоммутативной группы Г и G – элемент заданный элемент, принадлежащий Г, Причем X принадлежит некоторой заданной подгруппе группы Г, а элемент G имеет достаточно большой порядок. Задача состоит в том, чтобы найти значение X, который называется элементом, сопрягающим элементы Y и G. Трудность задачи поиска сопрягающего элемента используется в ряде протоколов открытого распределения ключей [17], ряде алгоритмов открытого шифрования [22], а также в схемах ЭЦП различного типа [30]. При этом ожидается, что использование трудности задачи поиска сопрягающего элемента в группах кос позволит построить криптосхемы, обладающие высокой (сверхполиномиальной) стойкостью к атакам с использованием вычислений на квантовом компьютере [25].

Вопрос появления реально действующего многокубитового квантового компьютера в обозримом будущем является дискуссионным и в настоящее время с целью построения протоколов цифровой подписи для их построения используется подход, состоящий в комбинировании в единой криптосхеме двух вычислительно сложных задач, а именно, задачи факторизации и задачи дискретного логарифмирования. При этом протокол ЭЦП строится таким образом, что для его взлома требуется одновременно решить обе эти задачи. В связи с этим вероятность взлома протокола за счет появления в ближайшем будущем прорывного непредвиденного алгоритма решения вычислительно трудной задачи, положенной в основу протокола, существенно снижается.

Для реализации данного подхода повышения уровня безопасности, обеспечиваемого протоколом ЭЦП предложен метод, использующий трудность разложения составного модуля специального вида [71,72], а также метод

использующий вычислительную трудность нахождения дискретного логарифма по трудно разложимому модулю [73-76].

1.5. Протоколы коллективной и групповой цифровой подписи

На практике потребность в протоколах коллективной ЭЦП имеет место при разработке документации для крупных проектов, требующих привлечения достаточно большого числа специалистов различного профиля. Каждый из них готовит, например, отдельный раздел документации. При этом отдельные разделы проектной документации или вся их совокупность должны быть подписаны всеми разработчиками. Для сокращения размера совокупной цифровой подписи и снижения вычислительной сложности проверки подлинности ЭЦП могут быть применены протоколы коллективной ЭЦП, предложенные и разработанные в работах [77-79]. Построение протоколов коллективной подписи оказалось весьма удачным, что позволило по аналогии с построением протоколов слепой индивидуальной подписи разработать протоколы слепой коллективной ЭЦП [80-82].

В отличии от мультиподписей других типов протоколы коллективной ЭЦП ориентированы на использование имеющейся на практике инфраструктуры открытых ключей, причем оказалось возможным реализация таких протоколов на основе процедур генерации и проверки ЭЦП, которые специфицируются стандартами ЭЦП России, Беларуси и Украины [83-86]. Также перечисленные стандарты ЭЦП могут быть использованы для построения протоколов слепой и слепой коллективной ЭЦП, т.е. функциональность данных стандартов может быть существенно расширена. Последнее потенциально обеспечивает более широкое применение этих стандартов и имеющейся инфраструктуры открытых ключей в практически используемых информационных технологиях.

Коллективная подпись обладает важным для практики свойством *внутренней целостности*, которое состоит в том, что по коллективной подписи вычислительно невозможно вычислить подпись, относящуюся к другой

совокупности подписантов. Целостность означает, что подпись едина и неделима: либо все подписанты подписали электронный документ, либо никто из них не подписывал этот документ. Размер такой подписи равен размеру одной обычной (индивидуальной) ЭЦП. Рассмотрим, как функционирует обобщённый протокол коллективной ЭЦП (КЭЦП). В нём вводится понятие *коллективного открытого ключа* некоторого произвольного задаваемого множества подписантов, например, включающего m субъектов. Коллективный открытый ключ Y представляет собой значение, вычисляемое по всем открытым ключам y_1, y_2,... y_m заданного множества: $Y = f(y_1, y_2, ..., y_m)$. Обобщенная схема формирования КЭЦП представлена на рис. 1.1.

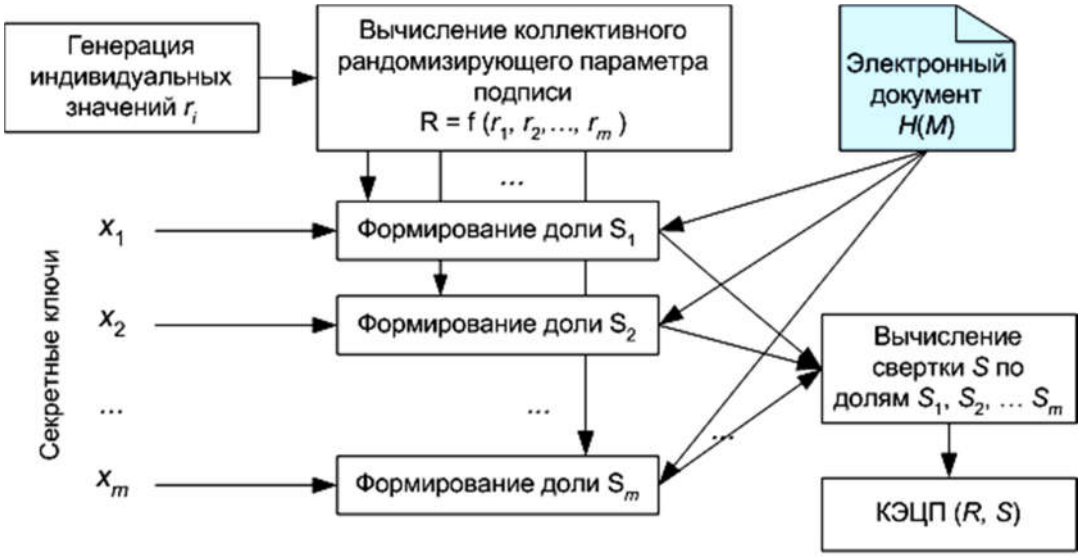

Рис. 1.1 Схема формирования коллективной подписи по протоколам [78,79]

При формировании КЭЦП каждый i-й подписант генерирует свой *разовый личный секретный ключ* k_i, и вычисляет по последнему свой разовый открытый ключ r_i (фактически параметр r_i представляет собой значение, через которое i-й подписант участвует в рандомизизации подписи). Все разовые открытые ключи r_i рассылаются каждому участнику протокола, после чего по ним вычисляется разовый коллективный открытый ключ R по формуле $R = f(r_1, r_2, ..., r_m)$, который по своей сути является интегральным параметром рандомизации, маскирующим все личные секретные ключи подписантов. При формировании коллективной подписи к электронному документу M произвольного размера используется

типовой приём представления документа M значением хэш-функции $H = F_H(M)$, где F_H – некоторая специфицированная функция хэширования. Значение R может быть взято в качестве первого элемента КЭЦП. Вторым элементом КЭЦП является число S, представляющее собой сумму долей S_i, $i = 1, 2, \ldots, m$, вычисляемых каждым подписантом индивидуально. Число S_i зависит от значения параметра R, значения хэш-функции H, значения личного секретного ключа подписанта x_i и его разового секретного ключа k_i. Если кто-либо из подписантов, участвовавших в формировании параметра рандомизации R, не предоставит правильно вычисленную долю подписи S_i, то нахождение правильного значения второго элемента КЭЦП вычислительно нереализуемо. Допустим все доли S_i вычислены правильно. Они рассылаются всем подписантам и любой из них может вычислить по всем долям их интегральное значение S, представляющее собой второй элемент КЭЦП. Полученное корректным способом значение КЭЦП в виде пары чисел (R, S) может быть использовано для доказательства того, что каждый из заданного множества подписантов действительно подписал электронный документ M.

Обобщенная схема процесса проверки подлинности КЭЦП представлена на рис. 1.2. Для проверки подлинности КЭЦП (R, S) к электронному документу M проверяющий считывает из справочника открытых ключей открытые ключи y_i подписантов (или из цифровых сертификатов подписантов). Коллективный открытый ключ Y для указанного множества подписантов вычисляется по формуле $Y = f(y_1, y_2, \ldots, y_m)$. Затем коллективный открытый ключ Y и коллективная подпись (R, S) подставляются в обычное проверочное уравнение, которое в частном случае $m = 1$ полностью совпадает с проверочным уравнением в протоколе индивидуальной ЭЦП.

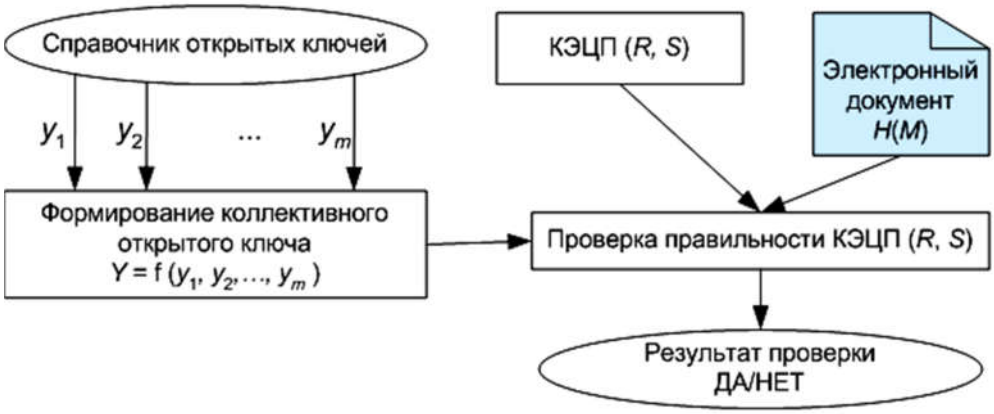

Рис. 1.2 Схема проверки подлинности коллективной ЭЦП по протоколам [78,79]

Важными преимуществами такого общего построения протоколов коллективной цифровой подписи является сравнительно малая длина подписи и использование имеющейся на практике инфраструктуры открытых ключей.

Другим интересным для практических приложений протоколов мультиподписи является групповая ЭЦП [87], который реализует возможность формирования ЭЦП от имени некоторого органа – группы подписантов, причем в группе подписантов выделяется лидер (руководитель). Предполагается, что протокол групповой подписи предоставляет следующие возможности: 1) подписать документ имеет возможность любой подписант или несколько подписантов из рассматриваемой группы; 2) руководитель и только он по значению ЭЦП и подписанному документу может идентифицировать подписантов, сформировавших данную подпись. Предложены различные варианты протоколов групповой ЭЦП [87], различающиеся дополнительными требованиями, использованными как критерии их разработки. Определенный интерес представляют протоколы пороговой групповой ЭЦП, спецификой которых является то, что для выработки групповой ЭЦП требуется участие не менее чем t подписантов, т.е. никакое их подмножество из $t-1$ субъектов вычислить правильное значение групповой подписи к какому-либо электронному документу не смогут. Существенными недостатками многих известных протоколов пороговой групповой ЭЦП являются следующие: 1) требуется

участие в протоколе некоторой доверенной стороны, которой подписанты передают свои личные секретные ключи; 2) для практической реализации протоколов требуется создание специфической инфраструктуры ключей.

В работе [88] предложен протокол утверждаемой групповой ЭЦП, который удовлетворяют следующим требованиям:

1) неразглашение личных секретных ключей подписывающих;

2) формирование групповой ЭЦП осуществляется путем вычисления предварительной цифровой подписи, после чего лидер по предварительной подписи вычисляет значение групповой ЭЦП (последнюю процедуру можно назвать утверждением подписанного документа);

3) предварительная подпись потенциально может быть создана каждым из подписывающих и произвольным их подмножеством;

4) руководитель и только он имеет возможность раскрыть групповую подпись к заданному документу без использования какой-либо дополнительной информации, т.е. используя значение подписи и сам электронный документ, к которому относится ЭЦП.

В работе [88] данный набор требований к протоколу групповой ЭЦП обосновывается тем, что свойства протокола обеспечивают близкую аналогию с процедурой подписывания и утверждения бумажных документов, которая реализуется на практике. Такая аналогия обусловливает практическую востребованность проколов утверждаемой групповой подписи, однако конкретный протокол такого типа, предложенный в работе [88] и использующий вычислительную трудность задачи дискретного логарифмирования в простом поле оставил открытыми следующие задачи:

1) реализация протокола утверждаемой групповой подписи с использованием вычислений на эллиптических кривых;

2) реализация протокола утверждаемой групповой подписи с использованием процедур формирования и проверки подлинности, регламентируемых российским стандартом ЭЦП ГОСТ Р 34.10-2012;

3) устранение использования внутренней инфраструктуры открытых ключей;

4) утверждение электронного документа двумя и более групповыми подписантами (например, документации к проекту, разработанного несколькими организациями) с помощью единой цифровой подписи при использовании имеющейся на практике инфраструктурой открытых ключей;

5) утверждение электронного документа единой подписью, подтверждающей, что документ подписан некоторой совокупностью групповых подписантов и некоторой группой индивидуальных подписантов.

Решение четвертой задачи означает разработку протоколов коллективной ЭЦП для групповых подписантов, что является построением нового типа протоколов мультиподписи, а решение пятой задачи – разработку протоколов комбинированной коллективной ЭЦП, что также относится к разработке протоколов мультиподписи нового типа.

Выводы к главе 1. Постановка задачи исследования

В современных технологиях электронного документооборота применение протоколов электронной цифровой подписи (ЭЦП) обеспечивает неотрекаемость от электронных сообщений и электронных документов. Применение таких протоколов лежит в основе придания юридической силы электронным документам и сообщениям. Огромное практическое значение протоколов ЭЦП обусловило принятие закона об электронной подписи и стандартов ЭЦП ГОСТ Р 34.10-1994, ГОСТ Р 34.10-2001 и ГОСТ Р 34.10-2012, а также создание развитой инфраструктуры открытых ключей, функционирующей в России. Все упомянутые стандарты специфицируют протоколы цифровой подписи формируемой индивидуальным подписантом, однако практика выдвигает новые задачи, для

решения которых требуется формирование цифровых подписей некоторой совокупностью индивидуальных подписантов (коллективная ЭЦП) или некоторым коллегиальным органом (групповая ЭЦП). Также востребована поддержка сценариев формирования единой ЭЦП со стороны нескольких групповых подписантов (коллективная ЭЦП для групповых подписантов). Также востребовано обеспечение возможности формирования единой цифровой подписи со стороны нескольких индивидуальных и нескольких групповых подписантов (комбинированная коллективная ЭЦП). Протоколы такого типа относятся к так называемым протоколам мультиподписи. В настоящее время предложены и исследованы протоколы групповой и коллективной ЭЦП, однако открытым остался вопрос разработки протоколов коллективной ЭЦП для групповых подписантов и комбинированной коллективной ЭЦП. При этом известные протоколы групповой подписи не удовлетворяют основополагающему принципу известности секретного ключа единственному субъекту и не позволяют использовать имеющуюся на практике инфраструктуру открытых ключей в неизменном виде.

Важным практическим вопросом информационной безопасности является обеспечение конфиденциальности информации, передаваемой по открытым каналам, реализуемое с помощью защитных преобразований информации. При этом в ряде практических случаях требуется разработка программных и программно-аппаратных модулей, реализующих такие преобразования на основе типовых алгебраических операций, что обеспечивает снижение затрат на реализацию таких модулей и упрощает их интеграцию в существующие информационно-телекоммуникационные технологии.

Учитывая перечисленные моменты в данной диссертационной работе были поставлены следующие задачи:

1. Разработать метод повышения уровня информационной безопасности протоколов утверждаемой групповой подписи, предложенной впервые в работе [88].

2. Разработать метод повышения уровня информационной безопасности протокола слепой подписи.

3. Разработать метод реализации протоколов утверждаемой групповой подписи с использованием вычислительной трудности задачи дискретного логарифмирования на эллиптической кривой.

4. Разработать метод построения протоколов утверждаемой групповой подписи с использованием процедур генерации и проверки подлинности ЭЦП, специфицируемых российским стандартом ГОСТ Р 34.10-2012.

5. Разработать метод реализации протоколов коллективной групповой подписи.

6. Разработать метод построения протоколов комбинированной коллективной подписи.

7. Разработать метод устранения необходимости использования внутренней инфраструктуры открытых ключей для обеспечения возможности раскрытия групповой подписи руководителем.

8. Разработать метод построения безопасных защитных преобразований информации с использованием алгебраических операций.

9. Разработать метод псевдовероятностных защитных преобразований, стойкий к атакам с принуждением отправителя и получателя сообщений к раскрытию ключа шифрования.

Глава 2. Построение протоколов слепой и групповой подписи, обладающих повышенным уровнем безопасности

Распространены протоколы слепой подписи, основанные на вычислительной трудности задачи разложения на множители составных чисел вида $n = qr$, где q и r – два сильных простых числа [5,13,42] и на задаче дискретного логарифмирования в конечной циклической группе большого простого порядка [5,36,55]. Для усложнения задачи взлома протоколов слепой подписи, имеет смысл разработка такого протокола слепой подписи, чтобы при попытке взлома пришлось одновременно решать две различные трудные вычислительные задачи.

В настоящей главе протоколы слепой подписи строятся по аналогии с протоколами обычной ЭЦП, основанными на двух различных трудных вычислительных задачах [89].

2.1. Метод построения и протокол слепой подписи, базирующийся на вычислительной сложности одновременного решения задачи разложения целого числа на множители и дискретного логарифмирования

Обеспечение повышенного уровня безопасности протоколов слепой подписи может быть достигнуто за счет того, чтобы задать необходимость одновременного решения задачи дискретного логарифмирования в простом поле и задачи факторизации при попытках взлома протокола. Для этой цели протокол слепой ЭЦП можно построить на основе конструктивной схемы ЭЦП, сочетающей механизмы, используемые в схеме ЭЦП Шнорра [56] и в схеме ЭЦП, предложенной в работе [90]. Метод построения такого протокола связан с использованием в качестве модуля простого числа p, имеющего структуру вида $p = 2n + 1$, где $n = qr$, q и r – большие простые числа размером не менее 512 бит являющиеся элементами личного секретного ключа подписанта. При этом число p

задается в качестве одного из элементов открытого ключа подписанта и в качестве второго элемента открытого ключа задается значение параметра α, имеющего секретное значение порядка (по модулю p), благодаря чему подпись требуемой длины может быть сгенерирована только владельцем открытого ключа, причем в ходе формирования ЭЦП предполагается использование ослепляющих параметров для обеспечения анонимности пользователя, предоставляющего документ для подписывания. В основу протокола цифровой подписи [90] положено следующее проверочное уравнение

$$r = F(\alpha^{HSr} \bmod p),$$

где три числовых значения (p, α, λ) представляют собой открытый ключ, а два числа (r, S) – это цифровая подпись, при этом разрядность числа S удовлетворяет условию $|S| \le \lambda$; параметр H – это значение хэш-функции, вычисляемое от документа M; F – это некоторая специфицированная однонаправленная сжимающая функция (например, в качестве F можно использовать хэш-функцию F_H, используемую для нахождения хэш-значения $H = F_H(M)$ от подписываемого сообщения); α – это число, порядок которого по модулю p равен простому числу q, где q – личный секретный ключ. Параметр λ представляет собой битовую разрядность числа q, а q - простой делитель числа n. При таком соглашении уравнение генерации параметра S принимает следующий вид

$$S = k(Hr)^{-1} \bmod q.$$

В последнем выражении предполагается, что число r предварительно вычисляется по формуле $r = F(\alpha^k \bmod p)$ по значению k которое является разовым секретным ключом. При выборе 1024-битового простого значения p и сжимающей функции F, значение разрядности которой равно 160 битам, битовая разрядность цифровой подписи примерно равна $|F| + |q| \approx 160 + 512 \approx 672$ бит. Проверка выполнимости условия $|S| \le \lambda$ является одним из достаточно важных шагов верификации подлинности ЭЦП. Это связано с тем, что цифровая подпись с параметрами (r, S') при разрядности $|S'| \approx 1024$ бит (при $|p| \approx 1024$ бит) может

быть достаточно просто вычислена по открытому ключу, т.е. без знания секретного числа q. Несмотря на то, что подпись (r, S') и будет удовлетворять проверочному соотношению, добиться выполнения условия $|S'| \le \lambda$ настолько вычислительно сложно, насколько вычислительно сложна задача разложения на множители модуля n.

Алгоритм ЭЦП, предложенный в работе [90], может быть использован в качестве прототипа схемы ЭЦП, для взлома которой необходимо решения двух трудных задач – факторизации числа n и дискретного логарифмирования в простом поле характеристики p. Отметим, что в последнем случае мы делаем отличие между задачами дискретного логарифмирования в конечном поле, решаемой методом вычисления индексов, который имеет субэкспоненциальную сложность [5] и задачей дискретного логарифмирования в подгруппе этого поля, решаемой общими методами дискретного логарифмирования, которые имеют экспоненциальную сложность. Разберём следующую схему ЭЦП с открытым ключом, представленную в виде чисел (p, α, λ, y). Первые три числа задаются по аналогичной схеме ЭЦП [90], а параметр y вычисляется по формуле $y = \alpha^x \bmod p$, где x – еще один элемент секретного ключа. Разработанный протокол ЭЦП создан по принципу алгоритма ЭЦП Шнорра, где подпись к сообщению M формируется по следующей процедуре:

1. Сгенерировать $R = \alpha^k \bmod p$, где k - случайно генерируемое значение, при условии $k < q$.

2. Вычислить $E = F_H(M||R)$.

3. Вычислить $S = k + xE \bmod q$, обеспечивающее выполнение условия $R = \alpha^S y^{-E} \bmod p$. Пара параметров *(R, S)* принимают в качестве значения ЭЦП.

Проверка подлинности подписи *(R, S):*

1. Если $|S| > \lambda$, то подпись отклоняется как неверная. В противоположном случае вычисляется $R^* = \alpha^S y^{-E} \bmod p$.

2. Вычисляется значение м-функции от сообщения M, к которому присоединили значение $E^* = F_H(M||R^*)$.

3. Если значения E^* и *E равны, то подпись считается подлинной.*

На основе алгоритма представленного выше генерация ЭЦП использует случайные ослепляющие параметры τ и ε, и выполняется следующим образом.

1. Подписант генерирует значение $R' = \alpha^k \bmod p$, , где k - случайно генерируемое значение, при условии $k < q$.

2. Пользователь А генерирует случайные значения τ и ε, разрядность которых λ бит. Потом он вычисляет значения $R = R'\alpha^{\varepsilon} y^{-\tau} \bmod p$, $E = H(M||R)$ и $E' = E - \tau$, после чего передает подписывающему вычисленное значение E.

3. Подписант вычисляет значение $S' = k + xE' \bmod q$, обеспечивающее условия $R' = \alpha^{S'} y^{-E'} \bmod p$. Пользователю А пересылает S' (в данном случае параметрам слепой подписи являются (R', S')).

4. Пользователь А вычисляет значение $S = S' + \varepsilon$. И в итоге получает подлинную подпись подписанта к сообщению M, представленную парой чисел (R, S).

Сгенерированная подпись *(R , S)* должна успешно пройти проверку на достоверность, тем самым будет доказана корректность описанного протокола слепой подписи.

Для этого из $R' = \alpha^{S'} y^{-E'} \bmod p$ => $R'\alpha^{\varepsilon} y^{-\tau} = \alpha^{S'+\varepsilon} y^{-(E+\tau)} \bmod p$ и $R' = \alpha^{S'} y^{E'} \bmod p$. При этом вопрос об авторстве отпадает, так как любую тройку чисел (R', S', E'), сформированную подписывающим, можно сопоставить с подписью *(E, S)* для данного документа M. В итоге:

$$R = \alpha^{S} y^{-E} \bmod p \text{ и } R' = \alpha^{S'} y^{-E'} \bmod p\} \Rightarrow \{R/R' \equiv \alpha^{S-S'} y^{-E+E'} \equiv \alpha^{\varepsilon} y^{-\tau} \bmod p,$$

отсюда при условии случайном равновероятном выборе «ослепляющих» слагаемых τ и ε, считается, что подпись (E', S') была создана одной из троек,

которые были сформированы подписывающим при вычислении параметров слепой подписи.

Стоит отметить, что на первом шаге функции проверки подлинности подписи, одним из важных условий является требование к размеру подписи, которое состоит в том, что размер подлинной подписи не превышает размер секретного ключа $|q| = \lambda$. Важность этого требования связана с тем, что после потенциально возможного появления прорывного решения вычислительно трудной задачи нахождения дискретного логарифма по модулю p у потенциального нарушителя появится возможность вычислить значение подписи, для которой выполняется условие $|S| \leq |n|$ и которая удовлетворяет проверочному соотношению. Для этого параметр S вычисляется нарушителем по формуле $S = k - xE \bmod n$. При условии, что для решения задачи дискретного логарифмирования в конечном поле найден прорывной полиномиальный алгоритм и вычисление секретного значения x становится осуществимым практически, формирование элемента подписи S длины $|S| \leq \lambda$ всё ещё является вычислительно трудной задачей, поскольку для выполнения последнего неравенства необходимо решить задачу факторизации целого числа n.

При взломе описанной схемы подписи требуется решить две задачи одновременно: факторизации, позволяющую найти значение q, необходимое для вычисления значения параметра S, размер которого не превысит число $\lambda = |q|$, и дискретного логарифмирования, позволяющего найти закрытый (секретный) ключ x. Следует отметить, что одновременное решение двух выше обозначенных сложных задач не является обязательным условием для осуществления взлома. Секретные параметры системы можно получить, решив одну только задачу дискретного логарифмирования. Для этого можно воспользоваться следующим алгоритмом:

1. Произвольно выбирается число t, при условии, что его битовая длина не превышает $\lambda - 1$.

2. Вычисляется значение $Z = \alpha^t \bmod p$.

3. Методом вычисления индексов находится $T = \log_\alpha Z$. Это значение T, вычисленное по модулю $n = p - 1$. Размер этого значения с вероятностью почти равной единице: $|T| \approx |n| > |t|$. Так как по модулю p число α имеет порядок q, получаем $T = t \bmod q$, поэтому число q делит нацело на разность $T - t$. Отсюда возможно вычислить параметр q для этого достаточно выполнить факторизацию числа $T - t$. При этом достаточно высока вероятность, что задача разложения числа $T - t$ будет иметь достаточно низкую вычислительную сложность. Поэтому если выполнить несколько раз данную процедуру можно найти легко значение $T - t$, которое может быть сравнительно легко разложено на простые множители.

Поэтому требуется модифицированная схема ЭЦП, для взлома которой требуется решение двух трудных задач одновременно. Рассмотрим следующую модифицированную схему для осуществления взлома которой нужно решение двух сложных задач – факторизации числа n и логарифмирования в конечном поле характеристики p. Для этого при создании схемы ЭЦП выбирается параметр α с порядком n, а в уравнение проверки значения S вводится S^2. Теперь подпись к сообщению M формируется по следующему алгоритму.

1. Сгенерировать $R = \alpha^k \bmod p$, где k - случайно генерируемое значение, при условии $k < q$.

2. Вычислить $E = F_H(M||R)$.

3. Вычислить $S^2 = k - xE \bmod n$, при условии $R = \alpha^{S^2} y^E \bmod p$. Значение (R, S) принимается в качестве ЭЦП.

Проверка подлинности подписи (R, S):

1. Если $|S| > \lambda$, то подпись отклоняется как неверная. В противоположном случае вычисляется $R^* = \alpha^S y^{-E} \bmod p$.

2. Вычисляется значение хэш-функции от сообщения M, к которому присоединили значение $E^* = F_H(M||R^*)$.

3. Если значения E^* и E равны, то подпись считается подлинной.

Для взлома модифицированной схемы нельзя обойтись только решением задачи нахождения дискретного логарифма по простому модулю. Для подделки подписи после модифицирования криптосхемы потребуется знание разложения числа n. Решение задачи дискретного логарифмирования в итоге дает возможность вычисления секретного ключа x и возможности вычислить значение $k - xE \bmod n$. Однако для вычисления элемента подписи S необходимо также извлечь квадратный корень из последнего значения, что не менее сложно, чем разложение модуля n на простые множители.

Представленный алгоритм ЭЦП использует две вычислительно трудные задачи. Его можно использовать в качестве основы для алгоритма построения слепой подписи, по подобной схеме, основанной на алгоритме ЭЦП Шнорра, представленного в работе [7]. Попытка найти лазейку для взлома системы слепой подписи, когда одновременно требуется решение двух сложных задач, пока не имела успеха, так как пользователь, готовящий документ для получения к нему подписи вслепую, может факторизовать число n за счет того, что извлечение квадратного корня по модулю n дает четыре различных значений. Поэтому было принято решение в проверочном уравнении выполнять возведение параметра ЭЦП S в степень e, значение которого является взаимно простым со значением функции Эйлера по модулю.

Последний механизм был использован для разработки протокола слепой подписи с использованием схемы слепой подписи, описанной в работе [95] и включающей следующие шаги:

1. У подписывающего имеется открытый ключ $y = \alpha^k \, mod \, p$. Он генерирует случайное значение $k < q$, вычисляет число $\rho = \alpha^k \, mod \, p$ и передает его пользователю А. Пользователь А обладает некоторым электронным сообщением M и намерен получить для него слепую ЭЦП подписывающего, из которого пользователь А будет иметь возможность самостоятельно найти

значение подписи, которое удовлетворяет проверочному уравнению, используемому в стандарте ГОСТ Р 34.10–94 (данная схема слепой подписи предложена в [95] как вариант потенциального расширения функциональности указанного стандарта).

2. Пользователь А вырабатывает случайные равновероятные числа $\mu, \varepsilon \in \{1, 2, \ldots, q-1\}$, вычисляет значения $\rho' = \rho y^{\mu}\alpha^{\varepsilon} \bmod p$, $R' = \rho' \bmod q$ и $R = R'/H + \mu \bmod q$, где H– хэш-значение от подписываемого документа, вычисляемое в соответствии со стандартом ГОСТ Р 34.11–94. Значение R' неизвестно подписанту и представляет собой первый элемент подлинной ЭЦП, а R – первый параметр слепой подписи.

3. Пользователь А направляет число R подписанту. По значению R нельзя вычислить R', ввиду случайного выбора неизвестных для подписанта значений параметров μ и ε, которые связывают значение R' с числом R.

4. Подписант вычисляет второй параметр слепой подписи $S = k + zR \bmod q$ (z – его секретный ключ) и передаёт его пользователю А.

5. Пользователь А вычисляет второй элемент подлинной цифровой подписи $S' = H(S + \varepsilon) \bmod q$.

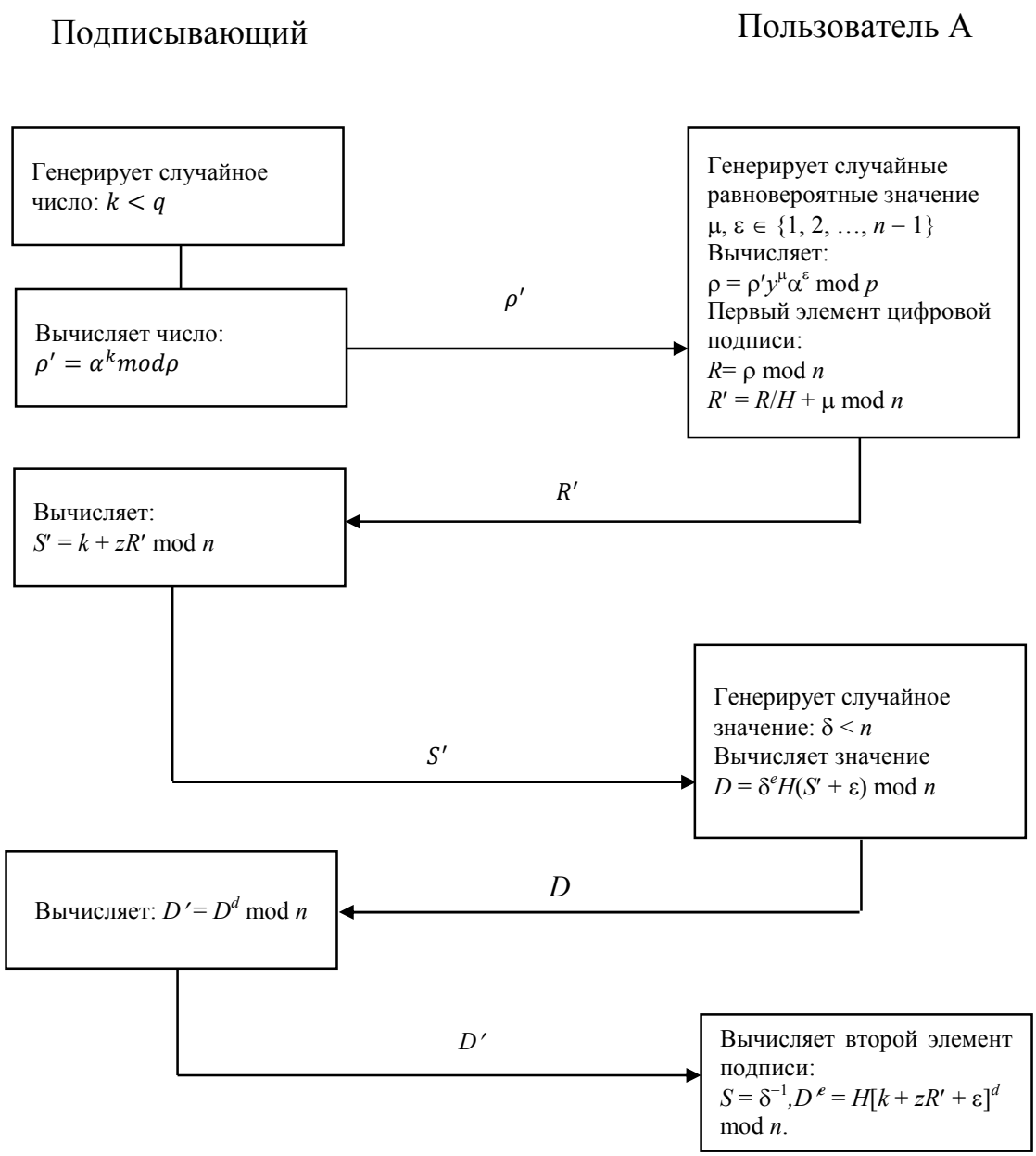

Рис. 2.1 Протокол слепой ЭЦП

Этот протокол использует в качестве проверочного соотношения уравнение проверки ЭЦП, специфицируемое стандартом ЭЦП ГОСТ Р 34.10–94:

$$R = (\alpha^{S/H} y^{-R/H} \bmod p) \bmod q \qquad (1).$$

Полученная в соответствии с описанным протоколом генерации подписи (R, S) вслепую последняя является подлинной, если она удовлетворяет уравнению проверки подлинности ЭЦП, указанному в п. 1. Для того, чтобы взлом схемы ЭЦП с таким уравнением потребовал одновременного решения двух разных

вычислительно сложных задач - нахождения дискретного логарифма по простому модулю и разложения числа специального вида, в качестве модуля следует взять простое число, обладающее следующей структурой $p = 2n + 1$, где $n = qr$, q и r – сильные простые числа размером не менее 512 бит. При этом в качестве проверочного уравнения выбирается следующее

$$R = (\alpha^{S^e/H} y^{-R/H} \bmod p) \bmod n \,, \qquad (2).$$

где значение e – число, взаимно простое с произведением $(p - 1)(q - 1)=\varphi(n)$. Значение e является элементом открытого ключа подписанта, которое выбирается им и по которому он вычисляет секретное значение $d = e^{-1} \bmod \varphi(n)$. При этом число α имеет порядок по модулю p, равный n.

С учетом указанных модификаций протокол слепой подписи имеет следующий вид (рис. 2.1):

1. Подписант генерирует равновероятное случайное число $k < q$, вычисляет значение $\rho' = \alpha^k \bmod p$ и направляет последнее пользователю А.

2. Пользователь А формирует случайные равновероятные значения маскирующих параметров $\mu, \varepsilon \in \{1, 2, \ldots, n - 1\}$, вычисляет значения $\rho = \rho' y^{\mu} \alpha^{\varepsilon} \bmod p$, $R = \rho \bmod n$ и $R' = R/H + \mu \bmod n$, где H– хэш-значение от подписываемого документа, вычисленное по некоторому специфицированному алгоритму хэширования (например, в соответствии с алгоритмом хэширования, заданным стандартом ГОСТ Р 34.11–94). Значение R является неизвестным подписанту и представляет собой первый элемент подлинной цифровой подписи. Число R' представляет собой значение первого элемента слепой подписи.

3. Пользователь А передает подписанту значение R'.

4. Подписант вычисляет значение $S' = k + zR' \bmod n$, где z– его секретный ключ, передает значение S' (второй элемент слепой подписи) пользователю А.

5. Пользователь А генерирует случайное число $\delta < n$ и вычисляет значение $D = \delta^e H(S' + \varepsilon) \bmod n$, которое направляет подписанту.

6. Подписант вычисляет значение $D' = D^d \bmod n$, где d является образным значением к e по модулю $\varphi(n)$: $d = e^{-1} \bmod (p-1)(q-1)$. Затем он направляет значение D' пользователю А.

7. Пользователь А вычисляет второй элемент подписи $S = \delta^{-1} D^{e} = H[k + zR' + \varepsilon]^d \bmod n$.

Процедура проверки подлинности ЭЦП (R, S) к документу M выполняется следующим образом (Рис 2.2):

1. Вычисляется хэш-значение H от документа M и число $R^* = (y^{-R/H} \alpha^{S^e/H} \bmod p) \bmod n$.

2. Если имеет место $R^* = (R, S)$, то подпись (R, S) принимается как подлинная.

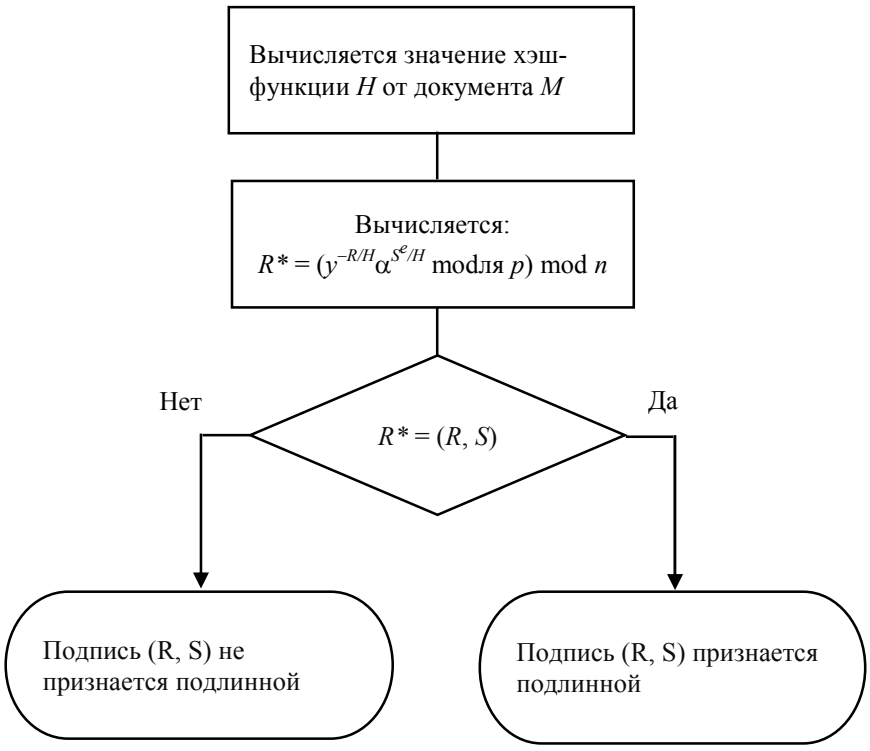

Рис. 2.2 Процедура проверки подлинности ЭЦП (R, S) к документу M

В итоге по описанному выше протоколу для генерации подписи вслепую получили ЭЦП с параметрами (R, S). Она является подлинной, если она вместе со значением хэш-функции H от сообщения M проходит уравнение проверки ЭЦП, указанное в п. 1, как верная (подлинная) подпись. Корректность работы протокола слепой ЭЦП для взлома которой необходимы решения одновременно двух вычислительно сложных задач дискретного логарифмирования в простом поле и факторизации можно доказать.

На шаге 4 вычисляет второй элемент слепой подписи $S' = k + zR' \bmod n$. Из данной формулы с учетом, что порядок числа α (по модулю p) равен n, следует справедливость сравнения $\alpha^{S'} \equiv \alpha^{k}\alpha^{-zR'} \bmod p$, из которого вытекает следующее сравнение $\rho' \equiv \alpha^{k} \equiv \alpha^{S'}\alpha^{-z'R} \bmod p$. Поскольку $R = H(R' - \mu) \bmod q$, то при вычислении правой части проверочного соотношения (2) для значения подписи (R, S) и значения хэш-функции H получаем следующее:

$$
\begin{aligned}
\rho^{*} &\equiv y^{-\frac{R}{H}}\alpha^{\frac{S^{e}}{H}} \equiv y^{-\frac{H(R'-\mu)}{H}}\alpha^{\frac{\left\{\left[H(k+zR'+\varepsilon)\right]^{d}\right\}^{e}}{H}} \equiv \\
&\equiv y^{-R'}y^{\mu}\alpha^{k}\alpha^{zR'}\alpha^{\varepsilon} \equiv y^{-R'}y^{\mu}\alpha^{k}y^{R'}\alpha^{\varepsilon} \equiv \\
&\equiv \rho' y^{\mu}\alpha^{\varepsilon} \equiv \rho \bmod p \Rightarrow \\
&\Rightarrow \rho^{*} = \rho \Rightarrow R^{*} = R\,.
\end{aligned}
\qquad (3)
$$

Последнее равенство означает, что подпись (R, S) к сообщению M успешно проходит процедуру проверки ЭЦП, т.е. является корректной.

Данный протокол создаёт условия анонимности пользователя, который предоставляет сообщения для получения подписи вслепую, т.е. точно определить пользователя приславшего данное сообщения для формирования слепой ЭЦП подписывающий не может (число подписанных сообщений с помощью протокола слепой подписи данным подписывающим $N > 1$). При наличии у подписывающего подлинной подписи (R', S') к сообщению M он не сможет идентифицировать пользователя, приславшего ему документ на подпись с вероятностью выше

значения d/N, где N – количество документов, подписанных (данным подписывающим) с помощью протокола слепой подписи; d– число документов, представлявшихся данным пользователем, так как любая подлинная подпись (R', S') с равной вероятностью может быть отнесена к каждой из N вычисленных значений слепой ЭЦП.

Подписывающему известны все тройки значения (ρ, R, S), из выполненных им N процедур, когда он подписывал сообщение вслепую. Любые из троек можно ассоциировать с произвольной подлинной (R', S'), относящейся к некоторому сообщению, которая представлена значением хэш-функции H. Так как тройки подлинной ЭЦП (ρ, R, S) и формируемая слепой подписи описанным (R', S', H) связаны сгенерированными равновесными значениями μ и ε, в соответствии с описанным ранее протоколом между подписывающим и пользователем, предоставляющим сообщение для получение слепой подписи. Поэтому тройка элементов подлинной ЭЦП (R', S', H) с равной вероятностью могла бы быть вычислена из любой тройки значений (ρ, R, S), которые формировались в ходе каждой из N процедур генерации слепой ЭЦП.

Если подписант фиксирует и хранит значение D, сформированное пользователем А, то для идентификации пользователя А подписант не может воспользоваться соотношением

$$\delta = \frac{D^d}{\left[H(S' + \varepsilon)\right]^d} \bmod n$$

или, что тоже самое, соотношением

$$\delta^e = \frac{D}{H(S' + \varepsilon)} \bmod n,$$

поскольку любые наборы переменных, входящих в правую часть последних двух формул, соответствуют некоторому случайному значению δ, которое неизвестно подписанту.

2.2. Метод повышения уровня безопасности протокола групповой подписи, основанного на маскировании открытых ключей подписантов

Криптографические алгоритмы и протоколы с открытым ключом широко применяются в современных информационно-телекоммуникационных системах для защиты информации и формирования электронных цифровых подписей (ЭЦП) к электронным документам, что предопределяет пристальное внимание к безопасности их использования. Понятие безопасности криптографических протоколов предполагает выполнение двух требований: 1) взлом протокола с помощью лучшего известного алгоритма взлома является вычислительно невыполнимым (это определяет стойкость протокола) и 2) вероятность появления прорывного алгоритма взлома в обозримом будущем является достаточно малой (это определяет возможность безопасного использования протокола).

Таким образом, высокая стойкость является только одним из двух основных требований, предъявляемых к протоколам с открытым ключом. Значение стойкости количественно выражается вычислительной трудоемкостью W лучшего известного алгоритма взлома протокола и измеряется в количестве операции определенного типа.

Вторым важным условием является значение вероятности P появления прорывного алгоритма взлома. В работах [74,75] предложено внедрить в криптосхему в качестве элемента безопасности соотношение W/P. Оно обеспечивает наглядность для двух важнейших требований безопасности при работе с криптографическими протоколами и алгоритмами. Для обеспеченья более высокого уровня безопасности есть две возможности:

1. Снижение вероятности взлома криптопротокола за счет появления прорывного решения указанной вычислительно сложной задачи.
2. Увеличить стойкость к атакам за счет размера задачи положенной в основу протокола

Первая возможность сводится к построению двухключевых криптосхем. Для их взлома требуется решение одновременно двух независимых вычислительно сложных задач – разложения на простые множители чисел специального вида и нахождения дискретного логарифма по простому модулю [74,75]. Разработаны протоколы ЭЦП, слепой ЭЦП, открытого согласования ключа и открытого шифрования [71-76], относящихся к криптосхемам последнего типа. Представляет интерес разработка протоколов утверждаемой групповой подписи (УГП), основанной на вычислительной сложности одновременного решения двух разных задач - задачи факторизации (ЗФ) и задачи дискретного логарифмирования (ЗДЛ). Предложенный впервые в работе [88] протокол УГП является основанным на двух вычислительно трудных задачах – ЗДЛ и ЗФ. Однако использование двух указанных задач в протоколе [88] не приводит к повышению значение интегральной безопасности, поскольку появление прорывного решения какой-либо одной из этих двух задач приводит к компрометации этого протокола УГП, характеризующегося использованием механизма маскирования открытых ключей подписантов.

Протоколы УГП, основанные на механизме маскирования открытых ключей представляют значительный практический интерес ввиду наличия ряда существенных преимуществ перед протоколами групповой подписи других типов. В связи с этим представляет интерес построение протокола УГП, взлом которого связан с одновременным решением рассматриваемых двух трудных задач -- ЗДЛ и ЗФ. Для решения этой задачи предлагается метод построения, состоящий в использовании механизма маскирования открытых ключей подписантов с вычислением самих открытых ключей по простому модулю p вида $p = dn + 1$, где n – трудно разложимое составное число d – четное число, в частности $d = 2$, и выполнении вычислений в мультипликативной циклической подгруппе конечного поля, генерируемой числом α, имеющим по модулю p порядок, равный n. На основе данного метода разработан протокол УГП, описываемый в следующем разделе.

2.3. Протокол утверждаемой групповой подписи, базирующийся на вычислительной сложности одновременного решения задачи разложения целого числа на множители и задачи дискретного логарифмирования

2.3.1 Требования к протоколу утверждаемой групповой подписи

Отметим различие понятий коллективной ЭЦП и групповой ЭЦП. Коллективной цифровой подписью называется ЭЦП, генерируемая заданным множеством подписантов и означающая, что каждый из них подписал заданный электронный документ. В отличие от этого, групповой подписью называют ЭЦП, означающая, что заданный электронный документ подписан коллегиальным органом, различные подмножества сотрудников которого могут сформировать групповую подпись. При проверке подлинности коллективной подписи используются открытые ключи всех подписантов. При проверке подлинности групповой подписи используется открытый ключ коллегиального органа.

Обычно рассматриваются протоколы групповой подписи, в которых реализуются следующие требования: 1) подписать документ может любой штатный сотрудник коллегиального органа; 2) руководитель этого органа значению групповой ЭЦП может определить лицо, сформировавшее данную конкретную подпись; 3) внешние лица не могут установить, кто из штатных сотрудников органа выработал данную конкретную групповую ЭЦП. В схеме групповой ЭЦП [88] реализованы следующие требования, представляющие практический интерес: 1) подписанты в процессе формирования групповой подписи используют личные секретные ключи, которые неизвестны руководителю; 2) выработка групповой ЭЦП выполняется в два этапа; на первом этапе вычисляется предварительная подпись, а на втором этапе руководитель по значению предварительной подписи формирует групповую ЭЦП (второй этап может рассматривается как процедура утверждения подписанного документа); 3) любое непустое подмножеством подписантов может сформировать

предварительную подпись; 4) по групповой ЭЦП и по документу, к которому относится эта подпись руководитель и только он может идентифицировать подмножество подписантов, участвовавших в формировании заданной подписи. Процедура такой идентификации называется раскрытием групповой подписи. Для практических приложений представляется важным, чтобы протокол предусматривал возможность раскрытия групповой подписи со стороны руководителя, причем для выполнения такой идентификации подписантов не должно требоваться никаких дополнительных данных, кроме самого значения групповой подписи и документа, который подписан этой подписью.

Заметим, что для подделки групповой подписи в протоколе [88] достаточно уметь вычислять дискретные логарифмы по простому модулю, а для раскрытия групповой подписи – факторизовать число n. Как первое, так и второе означает компрометацию протокола. Протокол можно считать безопасным до тех пор, пока не появится вычислительно эффективный алгоритм решения одной из двух задач, ЗДЛ или ЗФ.

2.3.2 Протокол утверждаемой групповой подписи повышенной безопасности

Для построения протокола УГП, взлом которого требует одновременного решения ЗФ и ЗДЛ, используется простое число p, имеющее вид $p = 2n + 1$, где n – трудно разложимое составное число, и протокол из статьи [88] в качестве прототипа. Простое число p используется в качестве одного из элементов открытого ключа руководителя, а значит и коллегиального органа, осуществляющего подписывание электронных документов. Таким образом, в предлагаемом протоколе предполагается, что руководитель генерирует случайные 512-битовые сильные [50] простые числа q и r, такие, что число $p = 2n + 1$ также является простым. Затем он вырабатывает число α, имеющее по модулю p порядок, равный n, выбирает случайное число z, имеющее разрядность не менее 256 бит, вычисляет значения $L = \alpha^z \bmod p$ и $\varphi(n) = (q - 1)(r - 1)$, генерирует случайное 32-битовое значение e и вычисляет число $d = e^{-1} \bmod \varphi(n)$. После этого

он уничтожает значения q и r и предоставляет четверку чисел (p, α, L, e) как открытый ключ коллегиального органа. Значения z и d составляют личный секретный ключ руководителя. Вычислительная невозможность нахождения значения d по значению e связана с необходимостью факторизации числа n для вычисления значения $\varphi(n)$ и обосновывается также как и в случае криптосистемы RSA [5,13].

Каждый j-тый подписант ($j = 1,2, \ldots, g$) генерирует свой личный секретный ключ x_j и вычисляет свой открытый ключ y_j по формуле

$$y_j = \alpha^{x_j} \bmod p.$$

Важно отметить, что платой за повышение уровня безопасности является то, что открытые ключи подписантов оказываются привязанными к открытому ключу руководителя, хотя, естественно, последний не знает личных секретных ключей подписантов. Также существенным отличием предложенного протокола является использование метода формирования рандомизирующей экспоненты λ_i с двукратным вычислением значений хэш-функции от открытого ключа y_i, к которому присоединяется первый раз личный секретный ключ, а второй раз – первое хэш-значение. Этот метод позволяет руководителю возможность доказать другим лицам, что идентифицированные при раскрытии некоторой групповой ЭЦП подписанты действительно участвовали в процедуре генерации этой ЭЦП. Процедура формирования групповой подписи к электронному документу M , выполняемая m подписантами, описывается следующими шагами (Рис 2.3):

1. Руководитель, используя предписанную хэш-функцию F_H вычисляет рандомизирующие экспоненты λ_i по формуле $\lambda_i = F_H(H||P_i||F_H(H||P_i||d))$, где знак || обозначает операцию конкатенации битовых строк ($i = 1,2, \ldots, m$). Затем он направляет каждое значение λ_i (предварительно зашифровав его по открытому ключу i-того подписанта, например с помощью алгоритма открытого шифрования

Эль-Гамаля [55]) только *i*-тому подписанту и вычисляет первый элемент групповой ЭЦП в виде значения $U = \prod_{i=1}^{m} y_i^{\lambda_i} \bmod p$.

2. Каждый *i*-ый подписант выбирает случайное $t_i < n$, вычисляет $R_i = \alpha^{t_i} \bmod p$ и отправляет значение R_i руководителю.

3. Руководитель генерирует случайное $T < n$ и вычисляет значения $R' = \alpha^{T} \bmod p$ и $R = R'(R_1 R_2 ... R_m) \bmod p = \alpha^{T + t_1 + t_1 + .. + t_m} \bmod p$ и $E = F_H(M||R||U)$, где E – второй элемент групповой ЭЦП. Руководитель направляет значение R каждому *i*-тому подписанту ($i = 1,2, \ldots, m$).

4. Каждый *i*-ый подписант вычисляет значение своей доли подписи $S_i = t_i + x_i \lambda_i E \bmod n$, где $i = 1,2, \ldots, m$, и направляет руководителю значение S_i.

6. Руководитель проверяет корректность каждой из долей S_i ($i = 1,2, \ldots, m$) путем проверки выполнимости равенства $R_i = y_i^{-\lambda_i E} \alpha^{S_i} \bmod p$. Если все доли S_i вычислены корректно, то руководитель вычисляет свою долю подписи $S' = T + zE \bmod n$ и значение $S = (S' + S_1 + S_2 + ... + S_m)^d \bmod n$ в качестве третьего элемента групповой ЭЦП.

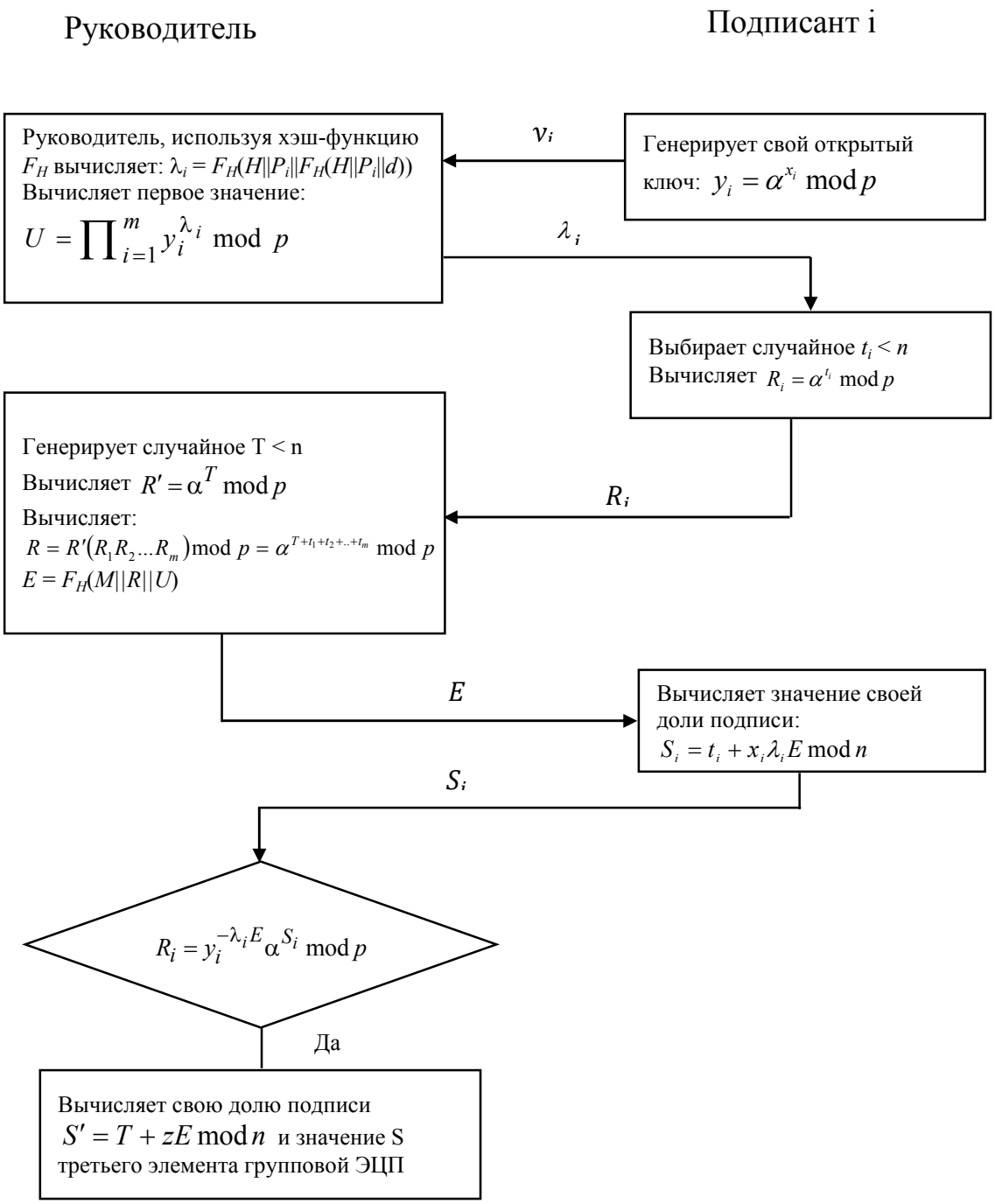

Рис. 2.3 Процедура формирования групповой подписи к электронному документу M , выполняемая m подписантами

Проверка подлинности групповой ЭЦП (*U, E, S*) к документу *M* выполняется следующим образом:

1. Вычислить хэш-значение от документа: $H = F_H(M)$.

2. По открытому ключу (p, α, L, e) и подписи (*U, E, S*) найти значение

$$\widetilde{R} = (UL)^{-E} \alpha^{\left(S^e \bmod \frac{p-1}{2} \right)} \bmod p .$$

3. Вычислить значение $\tilde{E} = F_H\left(M \,||\, \tilde{R} \,||\, U\right)$.

4. Выполнить проверку равенства значений E и $\tilde{E}$. Если $\tilde{E} = E$, то групповая подпись (*U, E, S*) принимается как подлинная, иначе подпись отвергается (Рис. 2.4).

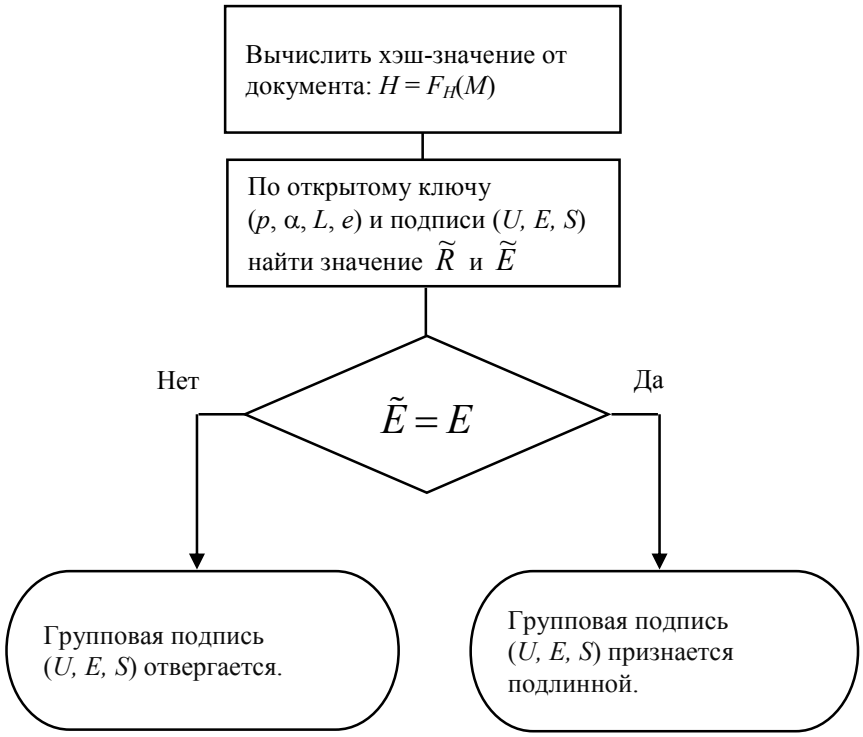

Рис. 2.4 Проверка подлинности групповой ЭЦП (*U, E, S*) к документу *M*

Существенное отличие разработанного протокола по сравнению с прототипом является то, что при подстановке значения третьего элемента подписи *S* в проверочное соотношение последний возводится в степень *e* по модулю $\frac{p-1}{2} = n$. Поэтому в случае появления прорывного алгоритма решения ЗДЛ по простому модулю и вычислительно эффективной возможности вычисления секретных ключей *z* и x_j ($i = 1,2, \ldots, m$) для генерации подписи останется необходимым вычисление корня *e*-ой степени по трудно разложимому модулю *n*, т.е. для взлома предложенного протокола потребуется решить как ЗДЛ

по модулю p, так и задачу факторизации модуля n. Лучшие известные алгоритмы решения этих задач имеют субэкспоненциальную трудность, причем при одинаковом размере чисел p и n трудоемкость этих алгоритмов имеет один порядок.

При одинаковом размере простого модуля p предложенный протокол имеет стойкость в два раза более высокую по сравнению со стойкостью протокола, использованного в качестве прототипа, но это увеличение стойкости не имеет практического значения в повышении уровня интегральной безопасности. Существенный рост значения последнего параметра достигается за счет того, что вероятность появления прорывного алгоритма взлома для предложенного протокола имеет существенно более низкое значение. Действительно, указанные две вычислительно трудные задачи являются независимыми, поэтому вероятность появления прорывных алгоритмов решения обеих задач существенно ниже вероятности появления прорывного решения одной из этих задач.

Использование в предложенном протоколе другого механизма формирования рандомизирующих экспонент λ_i, маскирующих открытые ключи подписантов, потребовалось для того, чтобы появление прорывного алгоритма решения ЗФ не приводило к компрометации протокола в плане возникновения возможности раскрытия групповой подписи несанкционированными лицами. Действительно факторизации числа n дает возможность вычислить значение функции Эйлера $\varphi(n)$ от числа n и затем – возможность вычислить личный секретный ключ руководителя, используя формулу $d = e^{-1} \bmod \varphi(n)$. После этого раскрытие групповой подписи осуществляется таким же способом, как это делает руководитель при возникновении такой потребности.

Использование двукратного вычисления хэш-функции при формировании маскирующих экспонент обеспечивает возможность доказать без раскрытия личного секретного ключа d любым другим лицам, что руководитель при раскрытии групповой подписи правильно идентифицирует подписантов. Для этого он предъявляет набор маскирующих экспонент λ_i и для каждого из

последних – уникальное значение δ_i, такое, что выполняется соотношение $\lambda_i = F_H(H||P_i||\delta_i) \neq F_H(H||P_i)$. В ходе такого доказательства руководителю нет необходимости раскрытия своего секретного ключа d, что потребовалось бы в случае вычисления маскирующих экспонент по формуле $\lambda_i = F_H(H||P_i||d)$. По представленным маскирующим экспонентам вычисляются модифицированные открытые ключи подписантов и их коллективный открытый ключ U, равный соответствующему значению в раскрываемой групповой подписи и зависящий от набора маскирующих коэффициентов в соответствии с формулой

$$U = \prod_{i=1}^{m} y_i^{\lambda_i} \bmod p.$$

Поскольку хэш-функция является вычислительно необратимой, набор маскирующих коэффициентов для данного значения хэш-функции, вычисленной от подписанного документа, могут быть восстановлены руководителем только в случае, если они действительно вычислялись до момента раскрытия подписи, а именно, в ходе формирования раскрываемой групповой подписи. При этом требуется восстановить уникальное значение δ_i. Руководитель и только он имеет возможность это сделать, поскольку последний параметр вычислялся с использованием его секретного ключа по формуле $\delta_i = F_H(H||P_i||d)$. Предоставление руководителем значения δ_i при раскрытии подписи доказывает то, что именно по этому значению вычислялось значение λ_i при выполнении процедуры формирования групповой подписи.

Процедура идентификации подмножества подписантов включает в себя вычисление коллективных открытых ключей подписантов для всех возможных подмножеств, пока не будет получен коллективный открытый ключ, фигурирующий в значении групповой подписи в качестве одного из ее трех элементов.

Определенным недостатком предложенного протокола по сравнению с протоколом-аналогом является увеличение размера групповой ЭЦП до 2208 бит

при обеспечении 80-битовой стойкости. При таком уровне стойкости размер подписи протоколе, использованном в качестве прототипа, размер подписи составляет 1344 бит. Для приложений, в которых первостепенным требованием является обеспечение высокого уровня безопасности данный недостаток не представляется существенным.

Выводы к главе 2

1. Разработаны методы построения протоколов слепой ЭЦП и групповой ЭЦП, для взлома которых требуется решения одновременно двух вычислительно сложных задач – ЗФ и ЗДЛ по простому модулю.

2. Разработан протокол слепой ЭЦП, для взлома которого требуется решить одновременно следующие две вычислительно сложные задачи – ЗФ и ЗДЛ по простому модулю.

3. Предложен метод формирования маскирующих коэффициентов, не требующий выполнения операции модульного возведения в многоразрядную степень при сохранении возможности доказательной идентификации всех подписантов по значению подписи.

Глава 3. Построение протоколов коллективной подписи для групповых и индивидуальных подписантов

Описанный в предыдущей главе протокол утверждаемой групповой подписи включает процедуру формирования групповой подписи, которая имеет близкую аналогии с процессом подготовки утверждаемых бумажных документов. В качестве одного из механизмов, позволяющих формировать групповую подпись различным подмножествам подписантов является механизм формирования коллективного открытого ключа, вычисляемого по их индивидуальным открытым ключам, который перенесен из протоколов коллективной подписи. При этом маскировка индивидуальных открытых ключей заданного подмножества подписантов обеспечивается тем, что при формировании коллективного открытого ключа используются модифицированные индивидуальные открытые ключи. Тем не менее, суть механизма формирования предварительной подписи как коллективной подписи заданного подмножества подписантов сохраняется. Это позволяет сделать предположение, что описанное в главе 2 построение протокола групповой подписи может быть реализовано для многих других типов уравнений проверки ЭЦП, для которых построены протоколы коллективной подписи. Учитывая такое соответствие можно предположить, что для групповых подписантов тоже может быть построен протокол коллективной подписи, т.е. электронный документ может быть подписан единой групповой подписью, означающей, что каждый из заданного подмножества коллегиальных органов (различных групп подписантов) действительно подписал этот документ. Это соответствует построению протокола коллективной ЭЦП для групповых подписантов. Практический интерес к проколам такого типа достаточно легко просматривается. Более того имеют практическую актуальность также и сценарии, когда единой подписью нужно удостоверить тот факт, что некоторый данный электронный документ подписан некоторым подмножеством групповых подписантов и некоторым подмножеством индивидуальных подписантов.

Протокол обеспечивающий решение такой задачи может быть назван протоколом комбинированной коллективной ЭЦП.

В следующих параграфах представлена разработка указанных новых типов протоколов коллективной подписи.

3.1. Метод построения и протокол коллективной ЭЦП для групповых подписантов

Достаточно реальным случаем является разработка электронного документа несколькими организациями, выступающими в роли групповых подписантов. Использование известных протоколов групповой подписи для этого случая связано с формированием нескольких независимых цифровых подписей. С целью обеспечения возможности формирования единой цифровой подписи, по которой можно доказательно проверить, что все ответственные стороны действительно подписали документ, представляет интерес разработка протокола коллективной ЭЦП для групповых подписантов по аналогии с протоколами коллективной ЭЦП [79,81], в которых размер подписи не зависит от числа индивидуальных подписантов. Изучение схем построения коллективной подписи и УГП показало, что эти два типа подписей могут быть объединены в едином протоколе коллективной ЭЦП для групповых подписантов. То есть методом реализации протоколов коллективной ЭЦП для групповых подписантов может служить задание процедур формирования долей подписи, генерируемых каждым групповым подписантом, и объединение всех долей подписи в единую подпись в соответствии с механизмами коллективной ЭЦП [79,81]. В соответствии с таким методом протокол коллективной ЭЦП для групповых подписантов может быть построен путем модифицирования известного протокола УГП [88], заключающейся в замене в нем механизма маскирования открытых ключей подписантов, основанного на схеме открытого шифрования RSA, на маскирующий механизм, основанный на вычислении хэш-функции от аргумента, зависящего от секретного ключа руководителя группы подписантов, описанного в

главе 2. Такая замена позволяет использовать один и тот же модуль для различных групповых подписантов при выполнении ими модульных вычислений для генерации ЭЦП и применить модифицированный протокол в качестве основы для протокола коллективной подписи для групповых подписантов.

Применяя такой метод построения, был разработан протокол коллективной ЭЦП для групповых подписантов, который описывается следующим образом. В протоколе используются следующие параметры:

1) достаточно большое простое число p (разрядностью не менее 2464 бит), такое, что число $p - 1$ содержит простой делитель q размером не менее 256 бит;

2) число α, порядок которого по модулю p равен значению q. Каждый представитель группы подписывающих генерирует свой секретный личный ключ в виде случайного числа x размер которого не менее 256 бит и свой открытый ключ $y = \alpha^x \bmod p$.

Открытый ключ руководителя L вычисляется по формуле $L = \alpha^X \bmod p$, где X – его секретный ключ. Значение L одновременно является открытым ключом группы, по которому проверяется подлинность групповой ЭЦП. Предполагается, что открытые ключи всех подписантов и руководителя регистрируются в удостоверяющем центре. Таким образом, они могут подписывать электронные документы не только в рамках протокола УГП, но как индивидуальные подписанты.

Пусть m подписантов, являющихся представителями одной из групп, обладают открытыми ключами $y_i = \alpha^{x_i} \bmod p$, где $i = 1,2, \ldots, m$; x_i – личный секретный ключ i-го подписанта. Формирование групповой подписи указанной группы подписантов к документу M выполняется по следующему алгоритму, входящему составной частью в протокол коллективной ЭЦП для групповых подписантов.

1. Руководитель вычисляет рандомизирующий параметр λ_i для каждого лица, являющегося внутренним подписантом документа M, по формуле $\lambda_i = F_H\left(H \| y_i \| F_H\left(H \| y_i \| X\right)\right)$, где $\|$ – операция конкатенации; F_H – некоторая специфицированная хэш-функция; $H = F_H(M)$, и передает каждое значение λ_i только i-тому подписанту. Затем руководитель вычисляет первый элемент групповой ЭЦП в виде значения

$$U = \prod_{i=1}^{m} y_i^{\lambda_i} \bmod p.$$

2. Каждый i-ый подписывающий (i = 1,2, …, m) генерирует случайное число $k_i < n$ и вычисляет значение $R_i = \alpha^{k_i} \bmod p$ и передает его руководителю.

3. Руководитель формирует случайное число $K < q$ и вычисляет $R' = \alpha^K \bmod p$ и значения

$$R = R' \prod_{i=1}^{m} R_i \bmod p = \alpha^{K + \sum_{i=1}^{m} k_i} \quad \text{и}$$

$$E = F_H(M||R||U),$$

где E – второй элемент групповой ЭЦП.

4. Каждый i-ый подписант (i = 1,2, …, m) вычисляет свою долю подписи $S_i = t_i - x_i \lambda_i E \bmod q$, где i = 1,2, …, m, и направляет руководителю значение S_i. и направляет ее руководителю.

5. Лидер проверяет корректность каждой доли путем проверки выполнимости равенства

$$R_i = y_i^{\lambda_i E} \alpha^{S_i} \bmod p.$$

При условии, что все доли цифровой подписи вычислены корректно, то он вычисляет свою долю подписи $S' = K + XE \bmod q$, а затем – третий элемент групповой подписи

$$S = S' + \sum_{i=1}^{m} S_i \bmod q.$$

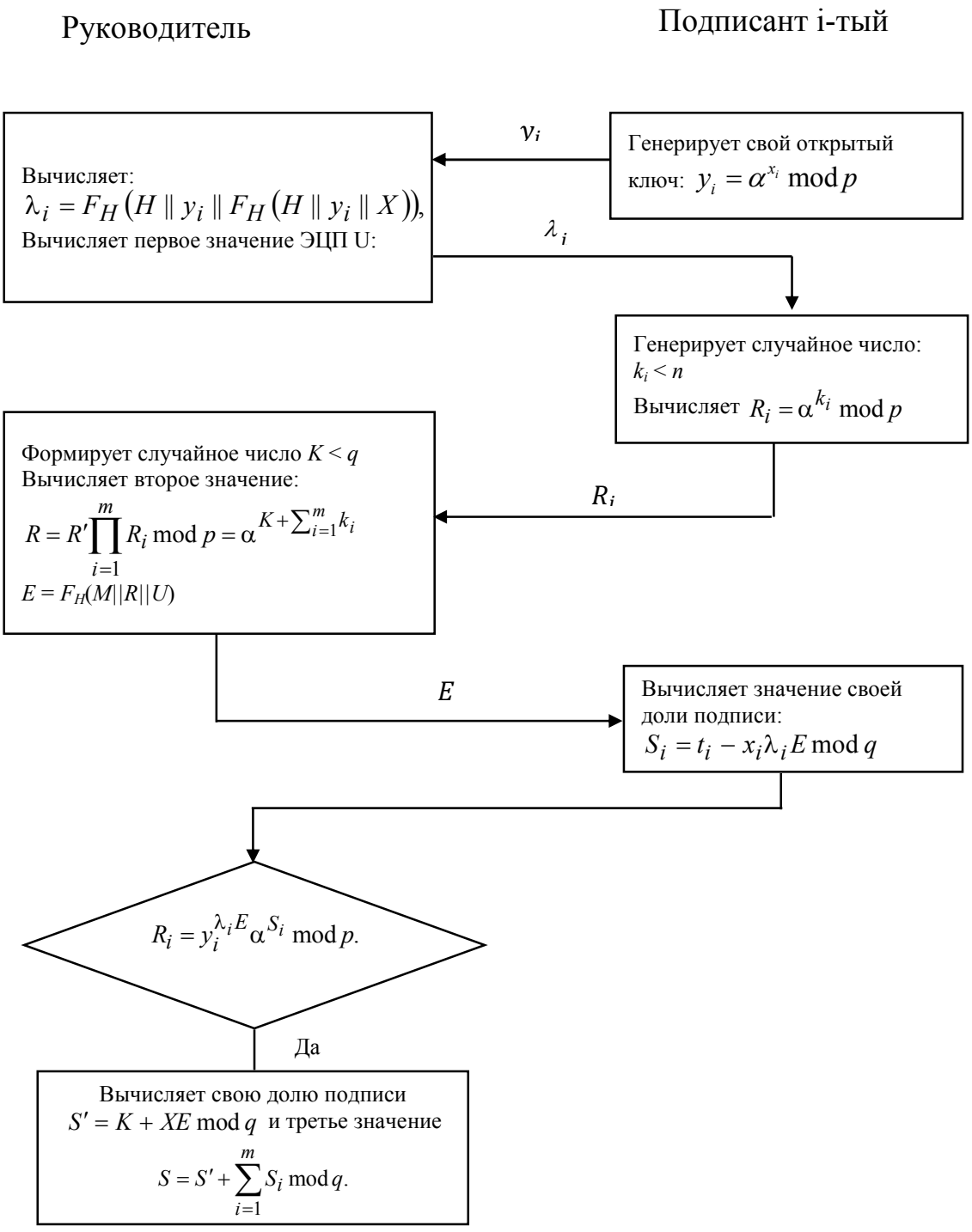

Рис. 3.1 Алгоритм вычисления долевого значения отдельного группового подписанта в коллективной групповой подписи

В ходе выполнения протокола формирования коллективной групповой ЭЦП данный алгоритм выполняется по частям в следующей последовательности:

1) каждый j-ый ($j = 1,2, \ldots, g$; g – количество групповых подписантов) групповой подписант выполняет независимо шаги 1–3 алгоритма, за исключением вычисления значения $E = F_H(M||R_j||U_j)$;

2) каждый все групповые подписанты рассылают свое значение параметра R_j и свое значение параметра U_j друг другу, а затем вычисляется произведение R (по модулю p) всех значений первого параметра

$$R = R_1 R_2 \ldots R_g \bmod p$$

и произведение (по модулю p) всех значений второго параметра;

$$U = U_1 U_2 \ldots U_g \bmod p$$

3) вычисляется общее значение $E = F_H(M||R||U)$;

4) по общему значению E каждый j-ый групповой подписант вычисляет значение своей доли подписи

$$S_j = S'_j + \sum_{i=1}^{m} S_{ji} \bmod q,$$

где S_{ji} доля подписи i-го ($i = 1,2, \ldots, m_j$) индивидуального подписанта в доле подписи j-го группового подписанта; S'_j -- доля подписи руководителя j-ой группы индивидуальных подписантов;

5) вычисляется общее значение параметра коллективной групповой ЭЦП:

$$S = \sum_{j=1}^{k} S_j \bmod q.$$

Коллективной групповой ЭЦП является тройка чисел (U, E, S).

При выполнении приводимого далее протокола коллективной ЭЦП для групповых подписантов выполняется проверка корректности долей подписи, генерируемых групповыми подписантами. Проверка подлинности доли S_j в коллективной групповой ЭЦП (*U, E, S*) к документу *M* выполняется следующим образом (Рис. 3.2):

1. Используя открытый ключ L_j заданного группового подписанта и значения U_j, E и S_j вычислить значение

$$\widetilde{R}_j = \left(UL_j\right)^E \alpha^{S_j} \bmod p.$$

2. Вычислить значение R_j.

3. Если $\widetilde{R}_j = R$, то доля (U_j, E, S_j) в коллективной групповой подписи признается подлинной и это значение используется для генерации значения коллективной групповой подписи, в противном случае эта доля отвергается и *j*-ый групповой подписант должен скорректировать значение своей доли.

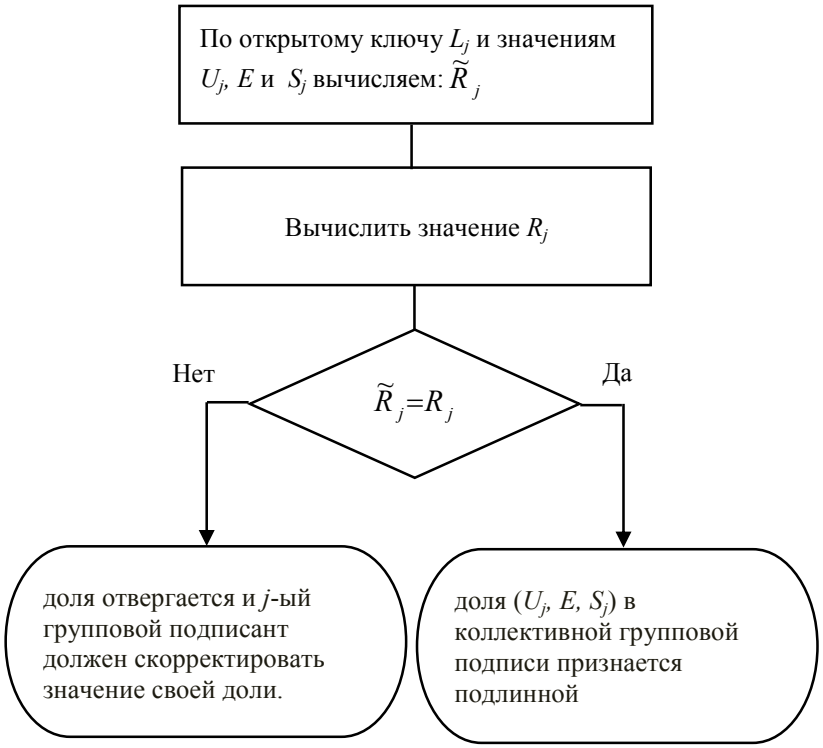

Рис. 3.2 Проверка корректности долей подписи, генерируемых групповыми подписантами

Легко заметить, что значение $D = \sum_{i=1}^{m} S_i \bmod q$, которое является предварительной подписью, утверждаемой руководителем путем сложения по модулю q со значением S', фактически представляет собой коллективную подпись всех подписантов, для формирования которой они использовали модифицированные открытые ключи. В разработанном протоколе схема коллективной подписи используется двояким образом – для формирования предварительной подписи и для формирования коллективной подписи для некоторого числа групповых подписантов.

Предлагаемый протокол коллективной ЭЦП для групповых подписантов описывается следующим образом. Пусть g групповых подписантов, обладающих открытыми ключами $L_j = \alpha^{X_j} \bmod p$, где $j = 1,2, \ldots, k$; X_j – личный секретный ключ руководителя j-й группы подписантов. Формирование коллективной групповой ЭЦП к документу M выполняется следующим образом.

1. В рамках описанного ранее алгоритма вычисления доли в коллективной групповой ЭЦП руководитель каждой j-ой группы подписантов ($j = 1,2, \ldots, g$) организует выработку маскирующих параметров для своих подписантов и j-ой доли в первом элементе коллективной групповой подписи, т.е. значения U_j, а также выработку рандомизирующего параметра $R_j = \alpha^{K_j} \bmod p$. Затем он направляет значения U_j и R_j всем другим руководителям.

2. Каждый j-ый руководитель (руководитель j-ой группы подписантов) вычисляет значения

$$U = \prod_{j=1}^{g} U_j \bmod p,$$

$$R = \prod_{j=1}^{g} R_j \bmod p = \alpha^{\sum_{j=1}^{g} K_j} \bmod p \text{ и}$$

$$E = F_H(M||R||U),$$

где U и E – первый и второй элементы коллективной групповой ЭЦП, соответственно.

3. Каждый j-ый руководитель (j = 1,2, …, g) вычисляет свою долю подписи S_j и рассылает значение S_j остальным руководителям. При этом для правильно вычисленной доли S_j выполняется соотношение $R_j = \left(U_j L_j\right)^E \alpha^{S_j} \bmod p$ (по которому выполняется проверка корректности доли подписи предоставленной j-ым руководителем). Затем он рассылает значение S_j остальным пользователям, которые могут проверить правильность значения по последней формуле.

4. Если проверка всех долей S_j подтвердила их правильность, то выполняется вычисление третьего элемент групповой подписи по формуле

$$S = \sum_{j=1}^{g} S_j \bmod q.$$

Проверка подлинности групповой ЭЦП (U, E, S) к документу M выполняется следующим образом (Рис. 3.3):

1. Вычисляется коллективный открытый ключ как произведение открытых ключей всех групповых подписантов:

$$L = \prod_{j=1}^{g} L_j \bmod p = \alpha^{\sum_{j=1}^{g} X_j} \bmod p.$$

2. Вычислить значение $\widetilde{R} = \left(UL\right)^E \alpha^S \bmod p$

3. Вычислить значение $\widetilde{E} = F_H\left(M \,||\, \widetilde{R} \,||\, U\right).$

4. выполнить проверку равенства значений E и $\widetilde{E}$. Если $\tilde{E} = E$, то коллективная групповая подпись (U, E, S) признается подлинной, иначе подпись отвергается.

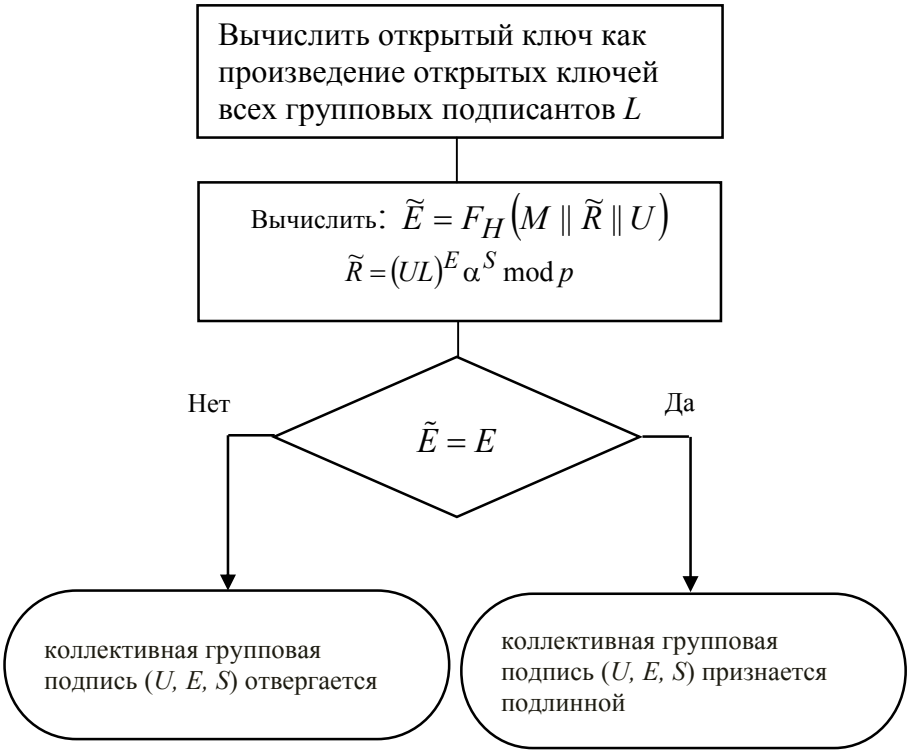

Рис. 3.3 Проверка подлинности групповой ЭЦП (U, E, S) к документу M

В предложенном протоколе коллективной групповой подписи восстановление лиц, участвовавших в формировании некоторой заданной коллективной подписи (идентификация индивидуальных подписантов) в одной или всех группах подписывающих требует участия всех руководителей. Процедура идентификации (раскрытие групповой подписи) выполняется по аналогии с процедурой раскрытия групповой подписи в протоколе УГП, описанном во второй главе.

3.2. Метод построения и протокол комбинированной коллективной ЭЦП

Также достаточно реальным случаем является разработка электронного документа несколькими индивидуальными подписантами и несколькими организациями, выступающими в роли групповых подписантов. Построение протокола коллективной ЭЦП для данного случая может быть выполнено в полном соответствии с протоколом коллективной ЭЦП для групповых подписантов, описанным в разделе 3.1. В качестве метода преобразования

протокола коллективной ЭЦП для групповых подписантов в протокол комбинированной коллективной ЭЦП предлагается принятие соглашения о том, что для индивидуальных подписантов значение доли в первом элементе коллективной подписи равно единице, т.е., если *j*-ый подписант является индивидуальным, то имеем $U_j = 1$.

В процедуре идентификации индивидуальных подписантов индивидуальные подписанты, очевидно, не принимают участие. Они также как и руководители групповых подписантов указываются непосредственно в списке подписавших, присоединяемом вместе со значением подписи в подписанному электронному документу.

Первый элемент коллективной подписи U содержит в себе информацию, позволяющую идентифицировать руководителю каждой группы подписантов всех индивидуальных подписантов его группы, участвовавших в протоколе. Однако, для выполнения процедуры раскрытия коллективной групповой подписи требуется участие всех руководителей в процедуре идентификации. При идентификации индивидуальных подписантов все руководители, участвующие в протоколе коллективной групповой подписи, согласованно выполняют полный перебор всех возможных вариантов модифицированных открытых ключей, в котором вычисляется набор модифицированных открытых ключей, произведение которых равно элементу U коллективной групповой подписи. Владельцы открытых ключей, вошедших в вычисленный набор, являются теми лицами, которые участвовали в процессе формирования заданной подписи. Таким образом, в ходе процесса идентификации индивидуальных подписантов происходит раскрытие всех подписантов. При этом в ходе этого процесса каждый руководитель восстанавливает имена подписантов, входящих только в его группу. Это обеспечивается тем, что параметры, используемые для маскирования открытых ключей, вычисляются в зависимости от личного секретного ключа руководителя.

Рассмотренные типы протоколов групповых подписей, которым придана функциональность схем коллективной ЭЦП, могут быть также рассмотрены как протоколы коллективной ЭЦП, которым придана функциональность утверждаемой групповой подписи. Этот факт связан со сходством механизмов формирования коллективных параметров в протоколах коллективной ЭЦП и протоколах утверждаемой групповой подписи.

3.3. Протокол коллективной цифровой подписи для групповых подписантов на основе процедур генерации и проверки подлинности цифровой подписи по стандарту ГОСТ Р 34.10–2012

Стандарты ЭЦП ведущих стран мира основаны на вычислительной трудности задачи нахождения дискретного логарифма на эллиптической кривой (ЭК). В криптографии используются ЭК, заданные над конечными полями. Они представляет собой конечные множества пар элементов (x, y) конечного поля $GF(p^s)$ (где s – параметр, называемый степенью расширения; p – простое число, называемое характеристикой поля), удовлетворяющих уравнению третьей степени. Над таким множеством пар (x, y), называемых точками ЭК, задается операция сложения точек, которая обладает свойствами ассоциативности и коммутативности (иногда эту операцию называют суперпозицией точек ЭК). Значение суммы точек $A = (x_A, y_A)$ и $B = (x_B, y_B)$ представляет собой точку $C = (x_C, y_C)$, координаты которой вычисляются по сравнительно простым порядок которых делится на достаточно большое простое число q формулам, в которые входят значения $x_A, y_A, x_B, y_B \in GF(p^s)$. Вид этих формул и вид уравнения ЭК зависит от значения характеристики p. Российский стандарт цифровой подписи ГОСТ Р 34.10–2012 регламентирует использование ЭК, заданным над простым полем $GF(p)$ с достаточно большим значением характеристики p и уравнение ЭК вида

$$y^2 = x^3 + ax + b, \text{ где } a,\ b \in GF(p).$$

Для таких ЭК сумма точек A и B вычисляется по формулам [39]:

$$x_C = k^2 - x_A - x_B \bmod p \text{ и}$$

$$y_C = k(x_A - x_C) - y_A \bmod p,$$

где $k = \frac{y_B - y_A}{x_B - x_A} \bmod p$, если точки A и B не равны, и $k = \frac{3x_A + a}{2y_A} \bmod p$, если точки A и B равны. Точки $A = (x_A, y_A)$ и $-A = (x_A, -y_A)$ называются противоположными. Сумма противоположных точек по определению равна бесконечно удаленной точке, обозначаемой буквой O, которая считается принадлежащей ЭК. Умножение точки A на натуральное число n по определению равно значению результата n-кратное сложения токи A:

$$nA = A + A + ... + A \quad (n \text{ раз}).$$

В качестве результата умножения любой точки ЭК на нуль берется точка O (бесконечно удаленная точка). Умножение на целое отрицательное число $-n$ определяется по формуле $(-n)\mathrm{A} = n(-\mathrm{A})$. Эллиптическая кривая над конечным полем представляет собой конечную коммутативную группу, в которой нейтральным элементом является бесконечно удаленная точка O. Вычисление неизвестного $h \in GF(p)$ в уравнении $P = hG$, где P и G – заданные точки ЭК, называется задачей дискретного логарифмирования на ЭК [39]. Число точек на ЭК называется ее порядком и обозначается как $\#E$.

В соответствии со стандартом ГОСТ Р 34.10–2012 следует использовать ЭК, имеющую значение порядка, которое содержит в своем разложении достаточно большое простой множитель q, имеющий разрядность не менее 256 бит или 512 бит в зависимости требуемого уровня стойкости. Для построения протокола коллективной ЭЦП для групповых подписантов воспользуемся методом аналогии, а именно, поступим следующим образом. Сначала выполним разработку протокола УГП, основанного на процедурах генерации и проверки подлинности ЭЦП, регламентируемых стандартом ГОСТ Р 34.10–2012, а затем преобразуем его в протокол коллективной ЭЦП по аналогии с тем, как это выполнено в разделе 3.2.

Для генерации открытых ключей следует задать некоторую точку G, порядок которой равен значению q. В протоколе УГП личный секретный ключ рядовых подписантов формируется в виде случайного числа t, а соответствующий ему открытый ключ – в виде точки P, вычисляемой по формуле $P = tG$. Руководитель вычисляет свой открытый ключ L по формуле $L = zG$, где z – его личный секретный ключ. Разработанный в данном диссертационном исследовании протокол УГП на основе российского стандарта ГОСТ Р 34.10-2012 описывается следующим образом.

1. Руководитель вычисляет маскирующий параметр λ_i для каждого лица, являющегося внутренним подписантом документа M, по формуле $\lambda_i = F_H\left(H \| x_{P_i} \| F_H\left(H \| x_{P_i} \| z\right)\right)$, где P_i – открытый ключ i-того подписанта; F_H – хэш-функция, специфицированная в стандарте ГОСТ Р 34.11–2012; $H = F_H(M)$, и передает каждое значение λ_i только i-ому подписанту. Затем руководитель вычисляет первый элемент групповой ЭЦП в виде значения

$$U = \lambda_1 P_1 + \lambda_2 P_2 + \ldots + \lambda_m P_m.$$

2. Каждый i-ый подписант генерирует случайное число $\rho_i < q$ и вычисляет точку $R_i = \rho_i G$ и отправляет ее руководителю.

3. Руководитель генерирует случайное число $\rho' < q$ и вычисляет точки $R' = \rho' G$, и $R = R' + R_1 + R_2 + \ldots + R_m$. Затем он вычисляет второй элемент групповой ЭЦП $e = F_H(M||x_R||x_U)$, где x_R и x_U – абсциссы точек R и U. Координаты точки R руководитель передает подписантам из его группы.

4. Каждый i-ый подписант ($i = 1,2, \ldots, m$) вычисляет значение $e = F_H(M||x_R||x_U)$ и свою долю подписи $s_i = e\rho_i + \lambda_i r t_i \bmod q$ и направляет ее руководителю.

5. Руководитель проверяет корректность каждой доли путем проверки выполнимости равенства $R_i = (s_i/e)G - (r\lambda_i/e)P_i$ для всех значений $i = 1,2, \ldots, m$. При условии, что все доли подписи вычислены корректно, то он вычисляет свою

долю подписи по формуле $s' = e\rho' + rz \bmod q$, а затем вычисляет третий элемент групповой подписи $s = s' + s_1 + s_2 + \ldots + s_m$.

Процедура проверки подлинности групповой ЭЦП (*U, e, s*) включает следующие шаги.

1. По открытому ключу группы подписантов и по групповой ЭЦП (*U, r, s*) к документу *M* вычислить точку $\tilde{R} = (s/e)G - (r/e)(U + L)$, где $e = F_H(M||x_R||x_U)$

2. Вычислить значение $\tilde{r} = x_{\tilde{R}} x_U \bmod q..$

3. Сравнить значения $\tilde{r}$ и *r*. Если выполняется равенство $\tilde{r} = r$, то ЭЦП признается подлинной.

Пусть имеются *g* групповых подписантов, обладающих открытыми ключами $L_j = z_j G$, где $j = 1,2, \ldots, g$; z_j – личный секретный ключ руководителя *j*-й группы подписантов. Формирование коллективной УГЭЦП к документу *M* выполняется следующим образом.

1. Руководитель каждой *j*-й группы подписантов ($j = 1,2, \ldots, g$) организует выработку маскирующих параметров для своих подписантов и *j*-ой доли в первом элементе коллективной групповой подписи, т.е. значения U_j, а также выработку рандомизирующего параметра R_j. Затем он направляет значения U_j и R_j всем другим руководителям.

2. Каждый *j*-ый руководитель вычисляет точки

$$U = \sum_{j=1}^{g} U_j, \quad R = \sum_{j=1}^{g} R_j$$

и число $r = F_H(x_R||x_U)$, где *U* и *r* – первый и второй элементы коллективной групповой ЭЦП, соответственно.

3. Каждый *j*-ый руководитель ($j = 1,2, \ldots, g$) вычисляет свою долю подписи s_j и рассылает значение s_j остальным руководителям. Для правильно вычисленной

доли s_j выполняется соотношение $R_j = (s_j / e)G - (r / e)(U_j + L_j)$. Затем он рассылает значение s_j остальным пользователям, которые могут проверить правильность значения s_j по последней формуле.

4. Если проверка всех долей s_j подтвердила их правильность, то выполняется вычисление третьего элемента групповой подписи по формуле

$$s = \sum_{j=1}^{g} s_j .$$

Групповой подписью к электронному документу M является тройка (U, r, s), где U – точка ЭК; e, s – натуральные числа. Процедура проверки подлинности групповой ЭЦП включает следующие шаги.

1. Вычислить коллективный открытый ключ

$$L = \sum_{j=1}^{g} L_j .$$

2. Вычислить значение $e = F_H(M||x_R||x_U)$ и точку $\tilde{R} = (s / e)G - (r / e)(U + L)$.

3. Вычислить значение $\tilde{r} = x_{\tilde{R}} x_U \bmod q.$.

4. Сравнить значения $\tilde{r}$ и r. Если выполняется равенство $\tilde{r} = r$, то коллективная ЭЦП для групповых подписантов признается подлинной.

В третьей главе описаны результаты по разработке нового направления в построении протоколов коллективной и групповой подписи, которое состоит в объединении свойство этих двух типов протоколов в едином протоколе. В сравнении с направлением разработки протоколов групповой подписи, основанных на вычислениях в решетках [116-119], на кодах [120-121] и на квантовой телепортации [122] предложенные протоколы выгодно отличаются тем, что они основаны на вычислительно сложной задаче дискретного

логарифмирования, благодаря чему они могут быть применены на практике при использовании уже имеющейся инфраструктуры открытых ключей. Последнее является преимуществом, которое представляется более существенны, чем достижение частных преимуществ протоколов групповой подписи, описанных в работах [123,124], например, по обеспечении динамичности подписантов [125] и уменьшению размера подписи [126].

Перечисленные направления развития протоколов групповой подписи, предложенные зарубежными исследователями в упомянутых работах [116-126], также представляют существенный интерес, однако на текущий момент и в ближайшем будущем предложенные в ходе диссертационного исследований методы построения и разработанные на их основе протоколы групповой и коллективной подписи представляются более интересными для практического использования, так как они согласуются с используемой на практике технологией ЭЦП в плане возможности использования алгоритмов ЭЦП, специфицируемых стандартами, и существующей инфраструктуры открытых ключей.

Выводы к главе 3

Впервые разработаны протоколы коллективной цифровой подписи для групповых и индивидуальных подписантов, независимо от соотношения первых и вторых. В случае, когда все подписанты являются индивидуальными, разработанные протоколы вырождаются в протоколы коллективной подписи, подобные протоколам, описанным в работах [78-80].

Достоинствами протоколов коллективной подписи для групповых подписантов и комбинированной коллективной подписи являются следующие:

- возможность реализации на основе процедур формирования и проверки подлинности ЭЦП на основе стандарта ГОСТ Р 34.11–2012;
- фиксированный размер подписи независимо от числа групповых и индивидуальных подписантов;

– для практического применения протокола можно использовать имеющуюся на практике инфраструктуру открытых ключей.

Глава 4. Алгоритмы шифрования с использованием алгебраических операций

4.1. Подход и метод построения блочных шифров на базе операции умножения матриц

Известные способы симметричного шифрования с использованием операции матричного умножения как базовой операции криптографического преобразования [91] достаточно прямолинейны, из-за чего они подверглись со стороны различных исследователей серьезной критике. В известных построениях шифров на основе матричного умножения по какой-то причине разработчики полностью доверились хорошим рассеивающим свойствам операции матричного умножения и практически не принимали во внимание атаки на основе алгебраических подходов, когда не рассматриваются влияние отдельных входных битов на отдельные выходные биты или группы битов, а рассматриваются подблоки данных и подключи как элементы некоторой конечной алгебраической структуры. При таком подходе криптоанализ предложенных шифров, базирующихся на матричном умножении, сводится к решению алгебраических уравнений, причем достаточно простых какими являются, например, системы линейных уравнений.

Указанные простые построения блочных шифров с использованием операции матричного умножения как операции шифрования включают представление сообщения в виде элемента конечного векторного пространства, определенного над конечным полем, и умножение вектора-сообщения на секретную матрицу-ключ, которая определена над тем же конечным полем. В частности удобно использовать задание векторов над двоичным полем $GF(2^s)$, где s – степень расширения простого поля $GF(2)$, поскольку в этом случае легко построить достаточно быстродействующие устройства шифрования, которые имеют достаточно низкую схемотехническую сложность реализации. В целом такой подход обеспечивает достаточно высокую производительность процедуры

шифрования и простоту реализации как при программной, так и при аппаратной реализации в виде устройств шифрования. При достаточно большой разрядности порядка конечного поля, над которым заданы матрицы и векторы известные способы шифрования обеспечивают приемлемую стойкость к криптоаналитическим атакам на основе известного шифртекста (при этом также известным считается алгоритм шифрования).

Однако известные способы шифрования, использующие матричное умножения, не обеспечивают приемлемой для практики стойкости к атакам на основе известных нарушителю или специально им выбираемых исходных текстов. Важность обеспечения стойкости к атакам последнего типа требуется для большинства приложений блочных шифров в средствах защиты информации от несанкционированного доступа. Стойкость алгоритмов защитных преобразований к атакам последних типов в настоящее время является стандартным требованием к алгоритмам симметричного шифрования [5,53,62]. Основным недостатком простых шифров на основе матричного умножения является то, что атака на основе известных или подобранных текстов практически по определению связана с наличием у атакующего большого числа линейных уравнений, связывающих координаты входного вектора данных с координатами выходного вектора данных через элементы матрицы-ключа, которые являются неизвестными.

При этом эти связи являются линейными, что является решающим моментом в пользу успешного выполнений криптоаналитической атаки указанного типа. Легко записываемые линейные уравнения образуют единую систему совместимых уравнений. Такая система в рамках каждой из рассматриваемых атак разрешима. Вопрос состоит только в получении такой системы уравнений, когда имеется число независимых уравнений, равное или превышающее число элементов, присутствующих в матрице-ключе. Рассматриваемые атаки на основе известных или подобранных исходных текстов не предполагают каких-либо существенных ограничений по объему известных

входных текстов, поэтому условие единственности решения указанной системы линейных уравнений практически всегда может быть обеспечено.

Таким образом, наличие хороших рассеивающих свойств матричного умножения, хотя и является предпосылкой для разработки стойких блочных шифров, но не представляет собой достаточным для этого преобразованием. Причем основной предпосылкой для атак на простые шифры с матричным умножением является то, что шифр работает в пределах одной алгебраической структуры, т.е. выполняются операции сложения и умножения в выбранном конечном поле. Приведенные ранее замечания дают основание сделать заключение об актуальности задача разработки стойких способов защитного преобразования информации, основанных на использовании матричного умножения в качестве базовой операции криптографического преобразования, При этом предполагается обеспечить стойкость к атакам на основе известных и специально подобранных текстов. Решение этой задачи требует включения в алгоритм защитного преобразования также и других дополнительных операций преобразования. В частности могут быть использованы операции, относящиеся к другим типам алгебраических структур. При соответствующем выборе вспомогательных операций можно обеспечить построение вычислительно эффективного шифрующего преобразования, для которого алгебраические подходы к криптоанализу окажутся вычислительно неэффективными для взлома шифра. С учетом того, что будет применяться комбинирование матричного умножения с другими типами операций потенциально имеется возможность использования достоинств матричного умножения как базовой операции для построения стойких блочных шифров.

Таким образом, для синтеза алгоритмов симметричного криптографического преобразования существенный интерес представляет *метод комбинирования операций преобразования из различных алгебраических структур*, состоящий в том, что вместе с операцией матричного умножения требуется использование дополнительных операций, в качестве которых можно

применить алгебраические операции, относящиеся к другим типам по сравнению с матричным умножением или дополнительных операций, выполняемых на битовом уровне, например, операций подстановок размером 4x4 или 8x8, поразрядных логических операций, операций битовых сдвигов (или перестановок битов другого типа).

4.2. Достоинства матричного умножения как примитива блочных шифров

Основными моментами, определяющими интерес к применению операции матричного умножения в качестве базовой операции алгоритмов блочного шифрования, являются следующие:

1. Сравнительная простота строения процедур, задающих блочное шифрование данных.

2. Относительно малый размер машинного кода при программной и микропрограммной реализации алгоритма шифрования.

3. Возможность выбора такого конечного поля для задания матрицы, при котором обеспечивается высокая скорость шифрования и достаточно низкая сложность аппаратной реализации на различных технических платформах (в заказных СБИС, а также при использовании программируемых логических матриц) для случаев различных размеров входного блока данных.

4. Использование стандартных команд контроллеров и микропроцессоров, выполнение которых требует малое число машинных тактов (поразрядное сложение по модулю два, арифметическое сложение, вычитание, арифметические сдвиги, циклические сдвиги и умножение).

5. Достаточно хорошие шифрующие свойства операции матричного умножения, вносящие существенный вклад в рассеивание и перемешивание блочного шифрующего преобразования.

6. Возможность легко распараллелить процесс выполнения операции матричного умножения как при аппаратной реализации, так и при программный реализации при использовании многоядерных микропроцессоров.

7. Потенциальная возможность построения шифров, обеспечивающих высокую стойкость к нападениям на основе известных и специально выбранных текстов, а также к атакам на основе связанных ключей (последнее обусловлено многократным использованием каждого элемента матрицы-ключа в операциях умножения в поле, над которым заданы матрицы).

8. Возможность построения блочных алгоритмов шифрования, использующих простую процедуру формирование раундовых ключей шифрования в зависимости от секретного ключа, и блочных шифров свободных от использования какой-либо процедуры усложнения ключа, что представляет интерес для шифров, ориентированных на применение в условиях частой смены ключей шифрования, например, в криптографических маршрутизаторах .

Представляет интерес обоснование использования двух операций матричного умножения блока данных, представленного в виде матрицы M, на ключевую матрицу K и на матрицу обратную ключевой матрице K^{-1} по формулам

$C = KMK^{-1}$ или $C = K^{-1}MK$.

Этот интерес связан с результатами, полученными в области криптосистем с открытым ключом, построенных с использованием конечных некоммутативных групп. В таких криптосхемах используется секретный ключа для задания преобразования, реализующего автоморфное отображение некоммутативной группы [92-94]. При этом при использовании секретного ключа в качестве сопрягающей матрицы автоморфного преобразования его вычисление по известным матрицам C и M связано с вычислительной задачей поиска сопрягающего элемента некоммутативной группы матриц [18].

При выполнении некоторой достаточно простой операции, задающий биективное преобразование матрицы M до выполнения указанного автоморфного

преобразования обеспечивается существенное повышение вычислительной сложности нахождения секретного ключа. В качестве такой дополнительной операции, например, можно использовать операцию циклического сдвига битов в диагональных элементах матрицы *M*. Это показывает, что при выполнении двух операций матричного умножения в принципе может быть получена приемлемая стойкость алгоритма блочного шифрования к атакам на основе известных и специально подобранных текстов при соответствующем задании конечной группы матриц, связанным с выбором размерности матриц и конечного поля, над которым определяются матрицы. При этом не потребуется использовать конечные поля, значение порядка которых равно числу большой разрядности как это имеет место в случае криптосистем с открытым ключом. При этом в блочных алгоритмах защитного преобразования это матричное уравнение «разрушается» путем включения дополнительных операций из других алгебраических структур и многократного выполнения процедур преобразования, выполняемых над заданным блоком исходных данных.

Следует отметить, что в случае разработки криптосистем с открытым ключом такой путь повышения стойкости не представляется возможным, так как в криптосистемах с открытым ключом требуется обеспечить взаимную коммутативность всех операций, используемых для формирования открытого ключа по секретному ключу. Если последнее не обеспечивается, то «разрушается» сама двухключевая криптосхема.

Таким образом, в случае синтеза блочных шифров отсутствует требование взаимной коммутативности используемых алгебраических операций, поэтому операцию автоморфного преобразования (умножение справа и слева на две взаимно обратные матрицы) можно комбинировать с операциями циклического сдвига или операциями умножения в полях, отличных от конечных полей, использованных для задания матриц, а также с операциями матричного умножения в других конечных полях. Эти общие предпосылки, естественно, требуют конкретных реализаций в виде конкретных решений по выбору

параметров процедуры шифрования в рамках синтеза конкретного алгоритма блочного шифрования. Однако наличие широких возможностей по выбору конечных полей для задания матриц, размера входных блоков данных и дополнительных операций преобразования потенциально создают предпосылки для успешного синтеза различных блочных шифров, использующих в качестве базового криптографического примитива матричное умножение.

Далее будет рассматриваться случай, когда входной блок данных, также как и выходной, представляется в виде набора подблоков данных, являющихся элементами некоторой матрицы, заданной над конечным полем. Это даст возможность использовать операцию автоморфного отображения как умножения справа и слева на матрицу-ключ и на ее обратное значение.

4.3. Выбор конечного поля для задания матриц и их размерности

Для задания конечных мультипликативных групп матриц следует учесть то, что входной блок данных должен быть представлен в виде матрицы, элементы которой имеют фиксированный размер. При этом для экономичного использования регистров микропроцессора или микроконтроллера и памяти вычислительного устройства рациональным является использование размера подблоков данных, которые задают элементы матрицы или элементы векторов, кратного 8 битам, т.е. размер, равный значениям s = 8, 16, 24, 32, 40, 48, 56 или 64 бит. С учетом разрядности массово доступных процессоров для разработки программно-ориентированных блочных шифров наибольший интерес представляют значения s = 32 бит и s = 64 бит.

Для того, чтобы блок сообщения, разбитый на подблоки размера s, мог бы быть преобразован с помощью матричного умножения «без потери информации» следует выбрать конечное поле вида $GF(2^s)$. Элементы таких полей представляют собой двоичные многочлены степени не более $s - 1$, а операция умножения в поле – это умножение многочленов по модулю неприводимого многочлена степени s.

Нарушение этого условия приводит к тому, что «элементарная» единица, участвующая в преобразовании, т.е. при выполнении операций преобразования, не может быть полностью размещена в одном регистре микропроцессора или микроконтроллера, что приведет к необходимости выполнения дополнительных элементарных операций микропроцессора по «сборке» частей «элементарной» единицы данных. Это может приводить к существенной потере в производительности алгоритма блочного шифрования.

В настоящее время считается, что для обеспечения стойкого блочного шифрования размер входного блока данных не должен быть менее 64 бит. При этом рассматривается вполне допустимым использование следующих размеров входного блока данных, равных 128, 256 и 512 бит. С учетом указанных предпочтительных размеров входного блока данных и размеров их подблоков можно найти различные решения по выбору вариантов задания матриц. В табл. 4.1 и 4.2 приведены примеры приемлемых вариантов выбора размера элементов матрицы и ее размерности для различных размеров входного блока при его преобразовании как вектора *V* или матрицы *M*. (При преобразовании блока данных как вектора *V* выполняется умножение последнего на матрицу ключ *K*.)

Таблица 4.1. Возможные варианты выбора размера элементов матриц *M* и *K* и их размерности (задание входного блока данных в виде матрицы)

№ п/п	Размер элементов матриц *M* и *K*, бит	Размерность матриц *M* и *K*	Размер входного блока данных *M*, бит
1	16	2×2	64
2	8	4×4	128
3	32	2×2	128
4	64	2×2	256

5	16	4×4	256
6	32	4×4	512
7	128	2×2	512

Таблица 4.2. Возможные варианты выбора размера элементов матриц *M* и *K* и их размерности (задание входного блока данных в виде вектора *V*)

№ п/п	Размер элементов матриц *K* и вектора *V*, бит	Размерность матрицы *K* (вектора *V*)	Размер входного блока данных *V*, бит
1	8	8×8 (8)	64
2	16	4×4 (4)	64
3	32	2×2(2)	64
4	8	16×16 (16)	128
5	16	8×8(8)	128
6	32	4×4(4)	128
7	64	2×2(2)	128
8	16	16×16(16)	256
9	32	8×8(8)	256
10	64	4×4(4)	256
11	128	2×2(2)	256
12	16	32×32(32)	512
13	32	16×16(16)	512

14	64	8×8(8)	512
15	128	4×4(4)	512

В случае задания матриц над конечным полем $GF(2^8)$ операция матричного умножения может быть выполнена достаточно быстро с использованием ЭВМ, если полную таблицу умножения элементов в таких полях разместить в оперативной памяти вычислительного устройства при запуске программы, реализующей алгоритм шифрования. Размер такой таблицы сравнительно мал и составляет 64 Кбайт. Эта таблица может быть вычислена заранее и размещена в саму шифрующую программу или может храниться в виде массива данных, переписываемого в оперативную память при вызове процедуры шифрования. После этого одно умножение в поле $GF(2^8)$ выполняется за одно обращение к памяти. Для современных вычислительных устройств это требование по памяти в большинстве случаев не представляется существенным и вполне приемлемо. В связи с этим выбор полей $GF(2^8)$ для задания матричного умножения представляет значительный интерес, поскольку обеспечивается возможность эффективного выполнения операции умножения в этом поле. Операция сложения будет представлять собой простую операцию поразрядного суммирования по модулю два, выполняемую над 8-битовыми подблоками данных или подблоком данных и подключом.

При задании матриц и векторов над расширенными полями $GF(2^{16})$, $GF(2^{32})$, $GF(2^{64})$ и $GF(2^{128})$ требуется выполнить выбор неприводимого двоичного многочлена степени $s = 16$, 32, 64 и 128, соответственно. Операция умножения в указанных полях будет выполняться по модулю выбранного неприводимого. Поскольку операция умножения двоичных многочленов по модулю неприводимого двоичного многочлена выполняется путем выполнения операции обычного умножения двоичных многочленов, после чего результат перемножения многочленов делится на неприводимый многочлен, то в качестве неприводимого

двоичного многочлена целесообразно брать многочлены с минимальным числом ненулевых коэффициентов. В этом случае существенно снижается сложность операции модульного умножения, поскольку при малом числе ненулевых коэффициентов становится возможным выполнение этой операции без выполнения операции деления многочленов, например по способу [95], записанному для случая использования двоичных трехчленов в качестве модуля. Существуют таблицы [37,96] неприводимых двоичных трехчленов различных степеней, однако, для случаев $s = 16$, 32, 64 и 128 неприводимых двоичных трехчленов, поэтому следует брать неприводимые пятичленны.

При умножении двоичных многочленов степени $s - 1$ по модулю неприводимого двоичного пятичлена вида

$$\eta(x) = x^n + x^k + x^h + x^g + 1,$$

где k мало по сравнению с n ($k < n/2$), модульное умножение также может быть выполнено без выполнения операции деления многочленов по аналогии со способом [95], описанным для случая неприводимых двоичных трехчленов. Для генерации неприводимых двоичных пятичленов можно использовать алгоритм, предложенный в работе [97], который позволяет генерировать двоичные многочлены, степени которых равны значениям до $s = 2048$ и более.

Известно, что в общем случае расширенные конечные поля $GF(p^s)$ как расширения простых полей $GF(p)$ могут быть заданы различными способами, т.е. существует большое число различных конкретных вариантов операции умножения многочленов по модулю неприводимого многочлена, при которых множество всех многочленов заданной степени, заданных над простым полем $GF(p)$, образуют конечное поле. Все конечные поля одинакового порядка являются изоморфными, что указывает на допустимость выбора различных вариантов задания конечных расширенных полей без снижения уровня безопасности, обеспечиваемой блочным алгоритмом шифрования, при условии, что другие элементы алгоритма шифрования остаются неизменными. Это

показывает на предпочтительность использования таких вариантов расширенных конечных конечных полей, для которых обеспечивается более низкая временная сложность операции модульного умножения многочленов. Последнее может быть обеспечено выбором в качестве модуля неприводимого многочлена с малым числом ненулевых коэффициентов.

Поскольку операция умножения по модулю неприводимого многочлена включает по определению операцию арифметического умножения многочленов и деление полученного результата на неприводимый многочлен, то вопрос снижения сложности умножения в расширенном поле сводится к выбору такого неприводимого многочлена, для которого операция арифметического деления будет иметь достаточно малую временную сложность. В случае выбора неприводимых многочленов общего вида, когда число ненулевых коэффициентов многочлена сравнительно велико сложность операции деления достаточно высока – существенно выше сложности операции арифметического умножения многочленов. Однако, если неприводимый многочлен содержит малое число (например, три или пять) коэффициентов, не равных нулю, включая ненулевой коэффициент при старшей степени формальной переменной, то вычислительная сложность операции арифметического деления многочленов примерно равна сложности операции арифметического умножения многочленов. Следовательно следует выбирать поля многочленов заданные по модулю неприводимого многочлена, содержащего минимизированное число ненулевых коэффициентов (включая старший коэффициент, равный единице. Если такой возможности нет, то в случае использования неприводимого многочлена произвольного вида можно использовать метод Монтгомери для выполнения модульного умножения многочленов [95]. Однако это оправдано в случае, когда число умножений достаточно велико, поскольку переход от обычного умножения по модулю к умножению по Монтгомери включает операции предвычислений и поствычислений [41,95]. Видимо в случае блочных шифров умножение по Монтгомери не может обеспечить повышение скорости шифрования. Поэтому

основным способом повышения производительности рассматриваемых шифров является выбор неприводимых многочленов специального вида.

Кроме способа снижения сложности умножения в расширенном конечном поле, связанного с выбором неприводимых многочленов с малым числом ненулевых коэффициентов (двучленов, трехчленов или пятичленов), имеется также способ уменьшения временной сложности операции умножения, связанный с уменьшение временной сложности операции умножения в поле $GF(p)$, над которым заданы многочлены. Операция умножения в поле $GF(p)$ является элементарной операцией в процедуре, описывающей операцию арифметического умножения многочленов, вычислительная сложность которой пропорциональна вычислительной сложности операции умножения в поле $GF(p)$. Последнее означает, что минимизация вычислительной сложности операции умножения в поле $GF(p)$ также играет важную роль, аналогично уменьшению вычислительной сложности операции умножения в конечном поле многочленов за счет выбора неприводимых многочленов специального вида. Перейдем к рассмотрению подходов к уменьшению временной сложности умножения в поле $GF(p)$.

Операция последнего вида реализуется процедурно как операция арифметического умножения чисел, принадлежащих множеству $\{0, 1, 2, \ldots, p - 1\}$, и арифметического деления полученного результата на простое число p. При этом арифметическое деление имеет временную сложность примерно в $\log_2 p$ больше, чем временная сложность арифметического умножения. Уменьшение временной сложности умножения в поле $GF(p)$, т.е. умножения по модулю простого числа p связано с использованием простых чисел p, имеющих специальную структуру в двоичном представлении (например, имеющих два или три ненулевых значения в двоичном представлении) или реализации модульного умножения по Монтгомери. Если используется первый метод, то нет смысла использовать второй. Однако, если требуется применить простые числа общего вида (как это имеет место в случае криптосистем RSA и Рабина), то применение способа умножения по Монтгомери является

оправданным в ряде случаев, поскольку число умножений в поле $GF(p)$ должно быть выполнено достаточно много раз для выполнения одного умножения в расширенном поле многочленов.

Умножение в полях $GF(p^s)$, а также умножение по простому модулю p представляет интерес для использования в качестве дополнительной операции при построении блочных шифров на базе операции умножения матриц. При этом итеративное построение алгоритма блочного шифрования может быть построено таким образом, что в нем могут быть применены операции преобразования, которые необратимы, например, как в случае построения блочного шифра по схеме построения сети Фейстеля []. Это расширяет разнообразие подходов к выбору вспомогательных операций, имеющих сравнительно низкую временную сложность и вносящих существенный вклад в стойкость блочного шифра.

Таким образом, существуют достаточно много вариантов задания матриц, элементами которых являются подблоки преобразуемого блока данных, имеющих размер 64, 128, 256 и 512 бит. При этом подблоки данных рассматриваются как элементы расширенного поля, над которым заданы матрицы, а выбором конкретного расширенного конечного поля $GF(2^s)$ может быть обеспечено существенное снижение вычислительной сложности умножения в $GF(2^s)$, а тем самым и снижение сложности матричного умножения. Также имеются широкие возможности по выбору вспомогательных операций при построении алгоритма блочного шифрования.

4.4. Итеративный блочный шифр с использованием вспомогательной операции в виде умножения по простому модулю

Рассмотрим конечную некоммутативную группу Г, групповую операцию в которой обозначим знаком «◦». Предположим, что эта группа имеет достаточно большой порядок, причем в ней содержатся элементы, порядок которых делится на некоторое достаточно большое простое число q (согласно теории групп такие элементы существуют в любой конечной группе). Один из таких элементов

обозначим как элемент M. Рассмотрим некоторый другой элемент X рассматриваемой группы Г задающий отображение элемента M в элемент $C \in \Gamma$ в соответствии с выражением

$$C = X \circ M \circ X^{-1} \qquad (4.1)$$

Это выражение может быть рассмотрено как заданное в группе Г уравнение, в котором неизвестным значением является элемент X. Например, такое уравнение возникает при попытке вычислить секретный ключ, задающий операцию преобразования значения M по формуле (4.1). В литературе [22,25] имеются предположения, что при выборе некоторых типов некоммутативных групп нахождение неизвестного в уравнении (4.1) представляет собой вычислительно сложную задачу (предположительно в группах переплетения [22,25]). Элемент X в формуле (4.1) называется элементом, сопрягающим элементы C и M, поэтому задача решения этого уравнения называется задачей поиска сопрягающего элемента. Формула (4.1) как операция преобразования интересна для построения алгоритмов шифрования тем, что она задает автоморфное отображение (автоморфизм) конечной группы Г. С этим связано и другое название задачи решения уравнения (4.1) при заданных известных элементах C и M (представляющих собой, например в задаче взлома алгоритма шифрования на основе известного входного текста, криптограмму и исходное сообщение) - нахождение автоморфизма отображающего заданное значение M в заданное значение C. При наличии нескольких пар элементов некоммутативной группы, сопряженных через один и тот же неизвестный элемент нахождение последнего сводится к решению системы линейных уравнений.

Последнее означает, что, оставаясь в рамках использования операций над матрицами, нахождение приемлемой формулы шифрования связано с составлением такого соотношения над матрицами, которое при соответствующей записи в виде системы уравнений (в конечным полем $GF(p^s)$) с многими неизвестными, содержало неизвестные, возводимые в некоторые степени и/или содержало в качестве слагаемых произведения двух и более неизвестных. Этого

можно достигнуть, например, путем задания ключа шифрования в виде двух или нескольких матриц. Используя три случайные матрицы X_1, X_2 и X_3 в качестве подключей (элементов секретного ключа), можно задать следующее преобразование матрицы M:

$$C = X_3 \circ (M \circ X_1 + X_2) \circ X_3^{-1}. \qquad (4.2)$$

Матричное уравнение (4.1) можно представить системой уравнений в конечном поле $GF(p^s)$, в котором неизвестными являются элементы матриц X_1, X_2 и X_3. Запись уравнения (4.2) в виде выражения $C \circ X_3 = X_3 \circ (M \circ X_1 + X_2)$ показывает, что в упомянутой системе уравнений будут присутствовать произведения различных наборов неизвестных, поэтому для решения данной системы уравнений в поле $GF(p^s)$ не могут быть непосредственно использованы методы линейной алгебры. В частности способ решения путем последовательного исключения неизвестных ведет к проблеме решения уравнений высоких степеней в поле $GF(p^s)$.

Ввиду того, что при размерах входных блоков, приведенных в табл. 4.1 найти решение переборным методом практически невозможно, то можно ожидать, выполнение шифрования по формуле (4.2) обеспечит стойкость к криптоанализу на основе известных исходных текстов и к атакам на основе выбранных исходных текстов и выбранных шифртекстов. Однако данные конструктивные решения по синтезу блочных шифров оставляют алгебраические атаки в рамках уравнений над полем $GF(p^s)$, т.е. в рамках единственной алгебраической структуры. Последнее представляет собой предпосылку для выполнения успешных алгебраических атак.

Для более полного противодействия потенциальным алгебраическим методам криптоаналитических атак на алгоритм шифрования, базирующийся на использовании матричного умножения, интерес представляет включение в процедуру шифрования операций из конечных алгебраических структур других типов. Поскольку включение дополнительных операций носит вспомогательный

характер, то в целях устранения существенного снижения производительности алгоритма шифрования представляет интерес использование достаточно простых дополнительных операций, например использование операции сложения из алгебраической структуры, отличной от поля $GF(p^s)$. В частности интерес представляет замена операции сложения присутствующей с скобках соотношения (4.2) на операцию сложения матриц, определенную как сложение их соответствующих элементов по модулю 2^s. Данное модифицирование не вносит существенного роста вычислительной сложности процедуры шифрования и является приемлемым способом комбинирования операций, относящихся к алгебраическим структурам двух разных типов. При расшифровывании криптограммы потребуется осуществление операции вычитания по модулю 2^s, однако вычитание и сложение по такому модулю реализуются микропроцессором общего назначения достаточно быстро.

Еще один способ «мгновенного» перехода (такой переход можно назвать переходом от одной алгебраической структуры к другой с «нулевыми временными затратами») от одной алгебраической структуры к другой состоит в комбинировании операций умножения в различных конкретных вариантах задания одного и того же поля $GF(2^s)$, отличающихся выбором различных неприводимых многочленов для задания модульного умножения многочленов (каждый из многочленов выбирается с учетом требования минимизации временной сложности операции умножения в поле $GF(2^s)$).

В следующем параграфе рассматриваются некоторые другие варианты комбинирования операций из разных конечных алгебраических структур. При этом предлагаемые варианты сформированы с учетом обеспечения практически «нулевых временных затрат» на переход от одной алгебраической структуры к другой (переход состоит в простой смене интерпретации блоков или подблоков даны как элементов разных алгебраических структур).

4.5. Комбинирование матричного умножения с операциями из других алгебраических структур

Операция умножения матриц, элементами которой являются элементы конечного двоичного поля $GF(2^s)$, обладает выраженными рассеивающими свойствами. Включение нескольких операций такого типа в процедуру итеративного блочного шифрования достаточно эффективно противодействует атакам, ориентированным на поиск статистических взаимосвязей между битами блоков исходного текста и шифртекста (см. результаты статистического тестирования, полученные в работе [98] для конкретных вариантов блочных шифров на основе матричного умножения).

Однако, в рамках атак алгебраического типа блоки и подблоки преобразуемых данных принимаются в качестве элементов поля $GF(2^s)$. При этом осуществление криптоанализа на основе специально выбранных исходных текстов или выбранных шифртекстов заключается в получении и решении уравнений над полем $GF(2^s)$ для нахождения значений неизвестных, которыми являются подключи. Для противодействия алгебраическим атакам можно использовать следующие способы:

1. Увеличение числа выполняемых операций матричного умножения, операндами которых являются различные матрицы-ключи, и чередование левых и правых умножений на матрицы-ключи.

2. Выполнение в рамках процедуры раундового шифрования сравнительно малого числа матричных умножений при их комбинировании со вспомогательными быстрыми и обратимыми операциями из разных конечных алгебраических структур.

Второй способ требует выполнения меньшего числа операций, благодаря чему может быть достигнута более высокая скорость шифрования. Однако его применение для построения шифров блочного типа связано с обеспечением согласования битовой длины входного блока данных, подблоков данных и

элементов из различных алгебраических структур. Для обеспечения указанной согласованности следует комбинировать в единой процедуре шифрования операции из различных алгебраических структур, которые включают в качестве своих элементов, все возможные варианты битовых цепочек заданного фиксированного размера. В этом случае шифр не будет вносить ограничений на возможные значения подблоков данных. Отсутствие таких ограничений является естественным требованием, поскольку предполагается, что зашифровыванию могут подвергаться и случайные данные, которые впоследствии требуется правильно и однозначно расшифровать.

В качестве основной алгебраической структуры, используемой в шифрах, основанных на операции матричного умножения, является конечное двоичное поле $GF(2^s)$. В этом случае встраивание в процедуру шифрования дополнительных операций из алгебраических структур другого типа предполагает рассмотрение блоков и подблоков преобразуемых данных в качестве элементов алгебраической структуры, которая отличается от поля $GF(2^s)$. При этом между элементами вспомогательной алгебраической структуры и всевозможными битовыми цепочками некоторой выбранной длины h должно иметь место взаимно однозначное соответствие. Такое условие существенно ограничивает число разных вариантов построения блочных шифров по второму способу. Тем не менее, в рамках второго способа имеются достаточное пространство для построения разнообразных шифров, интересных для практического применения. В качестве вспомогательных операций представляют интерес следующие:

Умножение по простому модулю вида $p = 2^{16} + 1$. Числа от 1 до 2^{16} составляют конечную мультипликативную группу поля $GF(2^{16} + 1)$. При этом в двоичной системе числа от 1 до $2^{16} - 1$ записываются в виде 16-битовых цепочек, каждую из которых естественно рассматривать как двоичное число, запись которого они представляют. Имеется единственное число 2^{16}, записываемое в виде 17-битовой цепочки, однако эта цепочка содержит единственный ненулевой старший бит, после которого присутствуют 16 нулевых битов. Поскольку нуль не

входит в мультипликативную группу, то битовую цепочку, содержащую 16 нулей, можно трактовать как число 2^{16}. Это обеспечивает нужное взаимно однозначное соответствие элементов множества 16-битовых цепочек и элементов указанной конечной мультипликативной группы. В последней групповой операцией является умножения по модулю $2^{16}+1$, которое обладает хорошими перемешивающими свойствами и выполняется очень быстро. Действительно, умножение двух 16-битовых цепочек a и b, таких, что либо $a \neq 2^{16}$, либо $b \neq 2^{16}$, может быть выполнено следующим образом. Пусть $ab = v_1 || v_2$, где v_2 есть 16-битовое число, тогда получаем

$$ab = v_1 \cdot 2^{16} + v_2 = v_1 \cdot 2^{16} + v_1 + v_2 - v_1 = v_1(2^{16}+1) + v_2 - v_1, \qquad (4.3)$$

$$ab \bmod p = v_2 - v_1, \text{ если } v_2 > v_1,$$

$$ab \bmod p = p + v_2 - v_1, \text{ если } v_2 < v_1, \qquad (4.4)$$

Таким образом, умножение двух 16-битовых чисел по модулю $2^{16}+1$ сводится к выполнению одного арифметического умножения, двух сравнений 16-битовых цепочек, одного сложения и одного вычитания. Благодаря тому, что устраняется необходимость выполнения арифметического деления обеспечивается значительное сокращение времени выполнения модульного умножения. Модульное умножение в рассматриваемой конечной группе целесообразно включить в процедуру шифрования в виде операции умножения подблоков преобразуемых данных на 16-битовый подключ K. Соответствующая обратная операция представляет собой умножению на обратное значение этого подключа K^{-1} (по $\bmod p$), которое может быть вычислено до начала выполнения процедуры шифрования, например на этапе загрузки программы шифрования.

Операции вращения. Эти операция представляют собой циклический сдвиг (вправо или влево) битовой цепочки некоторой заданной длины. Эти операции обратимы и выполняются за один машины такт над 8-, 16- и 32-битовыми цепочками.

Битовые перестановки. Данный вид операций в настоящее время представляет интерес в случае аппаратной реализации алгоритма шифрования. Однако при внедрении команды универсальной перестановки в состав команд универсальных микропроцессоров массового применения, которая описана в работах [99,100], этот вид операций будет иметь существенный интерес для разработки программно-ориентированных алгоритмов шифрования в том числе для построения шифров, основанных на выполнении битовых перестановок, зависящие от текущих значений шифруемых подблоков данных [101,102].

Рассмотренные варианты дополнительных операций хорошо иллюстрируют возможности проектирования в рамках второго подхода, основанного на комбинировании операций преобразования, относящимся к различным алгебраическим структурам.

4.6. Блочные шифры с использованием операций векторного умножения

Принцип комбинирования алгебраических операций из различных алгебраических структур, который использован в разделах 4.2 – 4.5 для синтеза блочных шифров на основе матричного умножения, представляется интересным для применения совместно и с другими типами алгебраических операций, используемых в качестве базового криптографического примитива. В качестве последних может быть использована операция умножения в конечных полях и кольцах. Например, умножение в простых конечных полях представляет собой умножение чисел по модулю простого числа, а в расширенных конечных полях – умножение многочленов по модулю многочлена, который является неприводимым. В случае использования конечных колец операция умножения также будет представлять собой модульное умножение, однако в этом случае мы имеем дело с умножением по модулю составного числа или приводимого многочлена. Умножение в полях и кольцах может быть задано и как умножение векторов, над конечным полем. Задание полей и колец в явной векторной форме

достаточно подробно представлено в работах [46,103]. При этом конечные кольца векторов могут быть заданы таким образом, что они будут некоммутативными [26,28,104]. Это означает, что принципиальный подход к синтезу блочных шифров на основе КГНМ может быть перенесен на случай использования некоммутативных конечных колец векторов без существенных изменений.

Переход к использованию векторного умножения в качестве базовой операции преобразования для разработки блочных шифров сохраняет принцип комбинирования левых и правых умножение на ключи шифрования, комбинирование векторных умножений различного типа (задаваемых по ТУБВ различного типа, или задаваемых через операции умножения многочленов по модулю неприводимого многочлена различного вида), комбинирование с дополнительными алгебраическими операциями: умножением по модулю $2^h + 1$, битовыми перестановками, сложением по модулю 2^h.

При этом использование векторного умножения позволяет иметь другие варианты реализации разбиения входного блока данных на подблоки по сравнению со случаем использования матричного умножения.

Способы задания конечных некоммутативных групп векторов и матриц над конечными полями описаны в работах [46,103]. Для устранения эффекта «урезания» информации при экономичном использовании регистров и памяти микропроцессора (микроконтроллера) представляется целесообразным выбирать размер координат векторов, равный $s = 8$, 16, 32 или 64 бит, в зависимости от разрядности вычислительного устройства, и использовать соответствующие значения порядка конечного поля над которым задается конечная некоммутативная группа. Для того, чтобы блок сообщения, разбитый на подблоки такого размера, мог бы быть преобразован с помощью матричного умножения «без потери информации» следует выбрать конечное поле вида $GF(2^s)$. Элементы таких полей представляют собой многочлены степени не более $s - 1$, а операция умножения в поле – это умножение многочленов по модулю неприводимого двоичного многочлена степени s. При этом для заданной степени s могут быть

найдены различные неприводимые многочлены. Аналогичное обоснование выбора алгебраической структуры для представления блоков и подблоков преобразуемых данных уже обсуждалось при рассмотрении подходов к синтезу блочных шифров на основе матричного умножения. Случай использования векторного умножения отличается возможностью выбора других соотношений между размером входного блока данных и его подблоками. Это связано с тем, что число подблоков растет кратно размерности вектора, тогда как в случае матриц мы имеем квадратичный рост, приводящий к ограничению приемлемых вариантов указанного соотношения.

В табл. 4.2 приведены примеры приемлемых вариантов выбора размера координат векторов, размерности векторов и соответствующего размера входного блока.

Таблица 4.2. Варианты выбора размера координат векторов и их размерности

№ п/п	Размер координат, бит	Размерность векторов, бит	Размер входного блока данных, бит
1	8	32	256
2	16	10	160
3	16	16	256
4	32	6	192
5	32	8	256
6	32	16	512
7	64	4	256
8	64	6	384
9	64	8	512
10	64	2	128

11	32	4	128
12	16	8	128
13	32	2	64
14	16	4	64
15	8	8	64

При построении функции блочного шифрования можно использовать представление входного блока данных и ключа как элементов двух различных групп, заданных над двумя различными изоморфными полями типа $GF(2^s)$, которые различаются выбранными вариантами операции умножения. Умножение в этих полях задается по модулю различных неприводимых многочленов одной и той же степени s. В соответствие с этим конечные группы имеют различные групповые операции (обозначим их знаками $\otimes$ и $\times$), т.е. являются различными алгебраическими структурами (хотя эти структуры могут быть изоморфными). Чередуя выполнение двух различных групповых операций обеспечивается стойкость функции шифрования к алгебраическим атакам. Стойкость к криптоанализу на основе известных исходных текстов и к атакам на основе специально подобранных текстов обеспечивается соответствующим комбинированием алгебраических операций из алгебраических структур различного типа.

Следует отметить, что операция векторного умножения обладает более выраженным лавинным эффектом по сравнению с операцией матричного умножения при заданном фиксированном размере входного блока данных. Это определяется тем, что число координат вектора в этом случае меньше числа элементов матрицы, а количество умножений в поле, над которым заданы вектора и матрицы, выполняемых для вычисления координаты вектора-результата и элемента матрицы-результата, одинаково в обоих рассматриваемых случаях. Это

означает в случае векторов умножение выполняется над операндами большего размера, а это определяет более выраженный лавинный эффект.

Для снижения временной сложности операции векторного умножения могут быть применены те же подходы, которые предложены в разделе 4.3 для минимизации сложности матричного умножения.

4.7. Особенности модульного умножения как вспомогательного примитива алгебраических блочных шифров

В общем случае для осуществления операции умножения по модулю требуются две арифметические операции –деления и умножения. Они дают неплохой лавинный эффект, при изменении одного из бита операнда, в этом случае их роль выполняют подключи и подблоки данных. Но следует учитывать, что вычислительная сложность операции арифметического деления существенно превышает вычислительную сложность операции арифметического умножения, поэтому необходимо при создании блочных алгебраических шифров правильно подобрать значения модулю, при которых есть возможность не пользовать операцией деления или снизить её временную сложность. В первом случае в качестве модуля выбирается число $m = 2^k + 1$, во втором – $m = 2^k$, где k – натуральное число. В первом случае сохраняется лавинный эффект для всех битов операндов, а для второго этот эффект значительно уменьшается от младших к старшим битам операндов. В случае снижения временной сложности операции деления ($m = 2^k$), для «выравнивания» лавинного эффекта можно воспользоваться операцией циклического сдвига на $\lceil k/2 \rceil$ бита и повторной операцией модульного умножения.

Выбор в качестве модуля других видов чисел имеет недостатки: необходимо выполнить операцию деления. При этом, когда разрядность блоков данных фиксирована возможна, либо неоднозначность обратного преобразование подблоков данных, либо необходимо построение блочных шифров, в которых формируется шифртекст, имеющий размер превышающий размер входного блока

данных. А это сокращает области применения блочных шрифтов в обоих описанных случаях.

В случае умножения по модулю $m = 2^k + 1$ подбирается размер подблоков данных равный k битам, при этом ненулевые подблоки рассматриваются как двоичные числа, а подблок, который включает k нулевые биты рассматривается как число $m = 2^k$. Для осуществления одной операции умножения по модулю $m = 2^k + 1$ требуется выполнить одну операцию арифметического умножения k-битовых чисел и не более трех операций сравнения. В случае, когда либо $a \neq 2^k$, либо $b \neq 2^k$, результат арифметического произведения ab представляет собой $2k$-битовое число. Представим последнее число в виде конкатенации двух k-битовых чисел w_1 и w_2, т.е. $ab = w_1 \| w_2$. Запишем это выражение как:

$$ab = 2^k w_1 + w_2 = \left(2^k + 1\right)w_1 + w_2 - w_1 \;\Rightarrow\; ab \equiv w_2 - w_1 \bmod\left(2^k + 1\right). \qquad (4.5)$$

Из последнего соотношения следует

$$ab \bmod\left(2^k + 1\right) = \begin{cases} w_2 - w_1, & \text{если} \quad w_2 > w_1 \\ 2^k + 1 + w_2 - w_1, & \text{если} \quad w_2 < w_1 \end{cases}. \qquad (4.6)$$

В случае, когда значения обоих k-битовых равно 2^k, то в качестве произведения берется значение, равное единице. Так для осуществления операции умножения по модулю $m = 2^k + 1$ потребуется выполнить одну операцию сложения, одну операцию вычитания, не более трех операций сравнения и одно арифметическое умножение. Таким образом, по данному модулю операция умножение осуществляется довольно быстро, при этом для всех возможных значений k-битовых операндов она обратима. Хотя и имеется недостаток: при синтезе шифров перемножения k-битовых подблоков данных имеет достаточно высокую временную сложность, так как для обратной операции необходимо вычислить обратного значения по модулю. Всё это приводит к тому, что скорость шифрования существенно снижается. Кроме того, при значение числа $m = 2^k + 1$,

не являющегося простым, в случае, когда один из операндов не является взаимно простым с модулем, обратное преобразование дает неоднозначный результат. Последнее обусловливает некорректность работы алгоритм шифрования в случае, когда в качестве каждого из двух операндов используются подблоки преобразуемых данных.

С учетом данных недостатков при выполнении этапа генерации ключа следует осуществить проверку каждого из всех подключей на взаимную простоту его значения с модулем. Для осуществления операции проверки на взаимную простоту значений возможно использование расширенного алгоритма Евклида. Это позволит одновременно найти и обратные значения для каждого из подключей. Данная процедура может быть выполнена предварительно, т.е. до осуществления процесса непосредственного шифрования данных, т.е. ее наличие не будет обусловливать снижение скорости алгоритма работы алгоритма шифрования. Для значений степени $k = 32; 64; 128$ число $m = 2^k + 1$ в своем разложении содержит два достаточно больших простых числа q и p и сокращением числа допустимых значений подключей можно пренебречь (см. табл. 4.1). Действительно, количество недопустимых значений равно $n = m - \varphi(m) = pq - (p-1)(q-1)$, где $\varphi(m)$ – значение функции Эйлера от числа *3*. Доля d необратимых значений может быть оценена по формуле

$$d = n/m = \frac{1}{p} + \frac{1}{q} - \frac{1}{pq} < \frac{1}{p} + \frac{1}{q}. \qquad (4.7)$$

Таблица 4.3. Число значений, которые не превышают m и необратим по модулю m, и оценка их доли по формуле (4.7) (знаком «\\» обозначен перенос записи в следующую строку).

k	m	q	p	n	доля,
32	$2^{32}+1$	641	6700417	6701057	0,15 %
64	$2^{64}+1$	274177	67280421310721	67280421584897	$3\cdot10^{-4}$ %

128	$2^{128}+1$	596495891\\ 27497217	570468920068\\ 5129054721	57047488502742\\ 56551937	$1{,}7\cdot10^{-15}$ %

Обозначенные выше замечания о возможности однозначного выполнения обратной операции также сохраняются и при умножении по модулю $m = 2^k$, таким образом необходимо пользоваться подключами с нечетными значениями, что сократит число возможных значений операнда-подключа в два раза. И получается существенная экономия подключевого пространства по сравнению с предыдущем случаем, что можно добавить дополнительное число независимых подключей в алгоритме шифрования. Но не смотря на это у данного случая есть недостаток, заключающийся в двух умножениях по модулю $m = 2^k$ для «выравнивания» лавинного эффекта.

Выполнение умножения по модулю может быть осуществлена без операции арифметического деления. Такая операция подходит для агрегирования блочных шифров. Но при этом обратная операция возможна не для всех значений подблоков. Например, когда используемые в качестве второго операнда подключи являются с модулем взаимно простыми. Про это не следует забывать при построение алгоритма шифрования. В этом случае операцию можно использовать при построении раундовой функции в итеративных шифрах на базе сети Фейстеля или на базе аналогичных сетей [52].

4.8. Задание матриц над конечными полями векторов

В данном разделе рассматривается метод минимизации временной сложности операции матричного умножения при задании конечной группы невырожденных матриц (КГНМ) над конечными расширенными полями с характеристикой произвольного вида. Данный подход представляет интерес для применения при построении двухключевых криптосхем на основе вычислительной трудности задачи скрытого дискретного логарифмирования, задаваемой над КГНМ .

Как показано в работах, в случае, когда между вектором с размерностью m и порядком поля p выполняются некоторые соотношения, операция умножения векторов может быть задана таким образом, что векторное пространство составит конечное поле $GF(p^m)$. Проведём сравнения по сложности операции умножения п поле данного типа по сравнению с простым полем $\boldsymbol{Z}_{p'}$, где $|p'| = m|p|$ и $|p|$ - битовая длина p (для полей с одинаковым размером порядка). В поле $GF(p^m)$ операция умножения элементов составляет m^2 операции умножения в поле $GF(p)$, при этом в поле $GF(p)$ сложность операции пропорциональна $|p|^2$, отсюда в $GF(p^m)$ при прямолинейном выполнении операции умножения, которое представлено в форме вектора, ее сложность примерно равно сложности умножения в поле $\boldsymbol{Z}_{p'}$, при этом арифметические сложения в расчет не берется, так как их участие мало.

При этом сложность умножения элементов поля $GF(p^m)$ возможно снизить. Для этого соответствующие пары значений координат векторов-операндов перемножаются с помощью операции арифметические умножения, результаты выполнения этих операций умножения суммируются, а потом выполняется деление на значение модуля p. Таким образом число арифметических умножений остается прежним (m^2), а число делений станет равной m, т.е. уменьшиться в два раза. При этом, естественно, в m раз возрастет делимое, но сложность операции деления это повлияет незначительно, так как длина делимого увеличится всего на несколько битов. А в случае практически используемых разрядностей координат векторов это перестает вообще играть существенную роль. Размер координат определяется разрядностью модуля, которая лежит в диапазоне от $|p| = 16$ до $|p| = 200$ бит при значении размерности векторов от $m = 13$ до $m = 3$, соответственно. Сложность операции арифметического деления существенно превышает сложность операции арифметического умножения, что ведет к тому, что при указанном способе сложность операции умножения в поле $GF(p^m)$ снижается примерно пропорционально размерности m. Для создания условий формирования полей в конечных векторных пространствах используются дополнительные операции умножения на коэффициенты растяжения, которые

практически не влияют на общую сложность операций умножения, так как размер коэффициентов можно взять всего в два-три бита.

В конечном поле $GF(p^m)$, заданном в виде множества всевозможных многочленов, имеющих степени не выше значения $m-1$ операция умножения (модульное умножение двух многочленов) включает выполнение m^2 операций арифметического умножения $|p|$-битовых чисел и m операций деления $2|p|$-битовых чисел на модуль p. По сравнению с вычислительной сложностью операции умножения сложность операции сложения достаточно мала и ею можно пренебречь. В результате выполнения операции арифметического умножения двух многочленов в качестве результата получаем многочлен степени $2m-2$. Завершается выполнение операции умножения в поле $GF(p^m)$ арифметическим делением последнего многочлена на неприводимый многочлен. Из-за последней операции эффективное распараллеливание многочленов в поле многочленов существенно затруднено. При осуществлении арифметического деления на неприводимый многочлен осуществляется примерно m^2 арифметических умножений $|p|$-битовых чисел и m арифметических делений $2|p|$-битовых значений на число p. Отсюда можно сделать вывод, что операция умножения в поле $GF(p^m)$, заданном в явной векторной форме, имеет вычислительную сложность примерно вдвое меньшую, чем вычислительная сложность операции умножения в поле $GF(p^m)$, заданном в виде множества многочленов.

Даже при использовании устройства с одним процессором новая форма задания расширенных конечных полей более эффективна при вычислении. Стоит отметить, что в поле $GF(p^m)$, заданном в конечном векторном пространстве, операцию умножения можно при необходимости разделить на m параллельных процессов, поэтому при возможности реализовать процесс с использованием более сложном вычислительном устройстве, можно сократить время выполнения операции умножения в поле $GF(p^m)$ примерно до m^2 раз в сравнении с простым полем и до $2m$ раз в сравнении с конечным полем многочленов. Есть возможность распараллеливания операции умножения (выполняемые до деления на

неприводимый многочлен) и в поле многочленов, это естественно усложнит аппаратную реализацию, и при этом уменьшит время вычисления лишь в $2m(m+1)^{-1}$ раз. Для сравнения, в векторном поле такой способ дает сокращение временя в m раз.

4.9. Способ совместного шифрования произвольных пар сообщений

Алгоритмы шифрования по двум независимым ключам известны как алгоритмы отрицаемого шифрования. Они впервые были предложены как механизм криптографического преобразования, который обеспечивает высокую стойкость к принуждающим атакам [105]. Предполагается, что в модели принуждающей атаки атакующий обладает некоторым ресурсом воздействия на законного пользователя (например, на отправителя, получателя или хранителя криптограммы), который вынуждает владельцев криптограммы представить ключ расшифровывания криптограммы. Стойкость к атакам с принуждением достигается тем, что конфиденциальные сообщения шифруются совместно с ложными сообщениями, содержащими правдоподобную информацию, которая будет восприниматься атакующим (при получении им доступа к ней) как информация, относящаяся к криптограмме. Конфиденциальные и ложные сообщения шифруются по двум различным ключам. При этом процедура расшифровывания осуществляется по алгоритму, который не дает оснований атакующему предполагать, что в шифртексте могут содержаться и другие сообщения, а в процессе шифрования использовались два различных ключа зашифровывания.

В общем случае размер криптограммы не может быть менее суммы размеров шифруемых совместно сообщений, поэтому в качестве признаков наличия в криптограмме других сообщений рассматриваются признаки отличия криптограммы от шифртекста, получаемого вероятностным шифрованием раскрытого сообщения, которое также приводит к увеличению размера

зашифрованного сообщения. Неразличимость отрицаемого шифрования от вероятностного шифрования считается достаточным, чтобы снять требования атакующего предоставить еще какой-то дополнительный ключ.

Вероятностное шифрование эффективно используются для защиты передаваемых данных. Оно существенно увеличивает процесс вычисления ключа и расшифровки сообщения для потенциального злоумышленника. А признак того, что размер расшифрованного текста меньше криптограммы не должно быть признаком наличия в ней «лишних» сообщений [105]. При этом атакующего ставят в известность, что к сообщению был применен метод вероятностного шифрования, т.е. криптограмма порождена данным алгоритмом. А для подобных шрифтов свойственно увеличение размера шифртекстов, в сравнении с открытыми текстами.

Признаками того, что криптограмме содержатся дополнительные сообщения для атакующего могут стать [105]: 1) наличие в процессе расшифровывания ветвлений, которые управляются ключом; 2) не вся криптограмма участвовала в расшифровки; 3) нарушение единообразия процедуры расшифровывания криптограммы при использовании различных значений ключа шифрования; 4) в процессе расшифровки идёт сортировка битов криптограммы; 5) биты криптограммы неравномерно влияют ни биты расшифрованного текста.

Задание равновероятного влияния битов шифртекста на биты восстановленного текста является важным требованием, которое предъявляется к процедурам отрицательного шифрования и состоит в том, что при условии изменения любого бита в криптограмме должно автоматически приводить к инвертированию бита в расшифрованном тексте с вероятностью примерно равной одной второй. Другими словами, алгоритмы шифрования должны обеспечивать единообразие расшифровывания по различным значениям ключа.

Механизм отрицаемого шифрования перспективен для использования в качестве специального механизма защиты информации от несанкционированного

доступа, позволяющего вводить потенциальных нарушителей в заблуждение, применяя способ регулируемой защищенности ключей, на которых зашифрованы сообщении содержащие дезинформацию. Применение алгоритмов отрицаемого шифрования в системах защиты информации от несанкционированного доступа связано с задачей разработки достаточно производительных алгоритмов данного типа. В настоящем разделе разрабатывается алгоритм отрицаемого шифрования, основанный на вычислениях в полях векторов, благодаря чему обеспечивается повышение их производительности по сравнению со способом, предложенным в работе [105].

Алгебраический алгоритм отрицаемого шифрования.

Шифруемый исходный текст будем разбивать на блоки данных, имеющих размер 1024 бит. Битовую строку, соответствующую некоторому блоку данных, обозначим как M и будем ее трактовать как вектор $M = (m_1, m_2, ..., m_h)$, координаты которого являются значениями в простом конечным полем $GF(p)$. Дополнительное осмысленное сообщение, которое шифруется совместно с конфиденциальным сообщением также разбивается на 1024-битовые блоки, рассматриваемые как вектора. Обозначим блок дополнительного сообщения вектором $D = (d_1, d_2, ..., d_h)$. При этом операция умножения векторов определена так, что множество всех возможных значений векторов размерности h образуют расширенное поле $\mathrm{GF}(p^h)$ [106], порядок мультипликативной группы которого содержит в качестве своего делителя простое число достаточно большой разрядности (не менее 256 бит). Для выполнения отрицаемого шифрования будем использовать два различных секретных ключа K_Q и K_W, каждый из которых представляет собой пару векторов и пару чисел (e, d): $K_Q = (Q_1, Q_2, e_1, d_1)$ и $K_W = (W_1, W_2, e_2, d_2)$. В качестве числа e выбирается случайное число e, взаимно простое со значением $p^h - 1$, где h – разрядность векторов. Второе число d вычисляется как значение, обратное к числу e по модулю $p^h - 1$.

Предлагаемый алгоритм отрицательного шифрования включает следующие шаги:

1. Генерация вектора R_1 по формуле

$$R_1 = M^{e_1}.$$

2. Генерация вектора R_2 по формуле

$$R_2 = M^{e_2}.$$

3. Вычисление блока криптограммы в виде пары векторов $C = (C_1, C_2)$, которая представляет собой решение системы уравнений следующего вида

$$\begin{cases} Q_1 C_1 + Q_2 C_2 = R_1 \\ W_1 C_1 + W_2 C_2 = R_2 \end{cases}.$$

Алгоритм расшифровывания криптограммы $C = (C_1, C_2)$ по ключу $K_Q = (Q_1, Q_2, e_1, d_1)$ состоит в выполнении следующих шагов:

1. Вычисление значения R по формуле

$$R = Q_1 C_1 + Q_2 C_2 = R_1.$$

2. Вычисление расшифрованного блока сообщения N по формуле

$$N = R^{d_1} = {R_1}^{d_1} = M.$$

Расшифровывание криптограммы $C = (C_1, C_2)$ по ключу $K_W = (W_1, W_2, e_2, d_2)$ состоит в выполнении следующих шагов:

1. Вычисление значения R по формуле

$$R = W_1 C_1 + W_2 C_2 = R_2.$$

2. Вычисление расшифрованного блока сообщения N по формуле

$$N = R^{d_2} = {R_2}^{d_2} = D.$$

Таким образом, одна и та же криптограмма расшифровывается в различные сообщения в зависимости от использованного ключа. При этом процесс расшифровывания выполняется в единообразной манере, в которой каждый бит криптограммы преобразуется одинаковым способом независимо от значения ключа расшифровывания, т.е. криптограмма неотличима от криптограммы, полученной с помощью процесса вероятностного шифрования, в котором используются случайные параметры Q_1, Q_2, R_1. При этом случайные параметры процесса зашифровывания не восстанавливаются однозначно в процессе расшифровывания.

Выводы к главе 4

1. Показаны общие недостатки блочных шифров на основе матричного умножения, предложенных ранее в литературных источниках и имеющих прямолинейное построение процедуры шифрования.

2. Показаны достоинства матричного умножения как криптографического примитива, принципиально позволяющие построить стойкие скоростные блочные шифры на его основе.

4. Предложен и обоснован общий подход к синтезу блочных шифров, использующих в качестве базового криптографического примитива матричное умножение, основанный на свойствах некоммутативности матричного умножения и комбинировании операций из алгебраических структур различного типа.

5. Сформулирован критерий комбинирования базовой алгебраической операции с дополнительными операциями из алгебраических структур различного типа и предложен конкретный ряд таких операций для комбинирования с матричным умножением.

6. Показаны основные подходы к заданию матриц, обеспечивающие минимизацию временной сложности матричного умножения.

7. Показана перспективность использования векторного умножения в качестве базового примитива блочных шифров и рассмотрены подходы к заданию векторов над конечными простыми полями для синтеза блочных шифров, обеспечивающие минимизацию временной сложности операции умножения векторов.

8. Предложен способ снижения операции матричного умножения путем определения конечных колец матриц над конечными расширенными полями, заданными в виде векторного пространства.

9. Предложен новый способ отрицаемого шифрования, отличающийся реализацией вычислений в конечных полях, заданных в явной векторной форме, благодаря чему обеспечивается повышение производительности.

Глава 5. Протоколы с открытым ключом, использующие матричное умножения

5.1. Оценка безопасности алгоритма Cayley-Purser

Алгоритм Cayley-Purser [wwhyte@baltimore.ie] предложен для выполнения быстрого открытого шифрования. Предполагалось, что его целесообразно использовать при создании практических приложений, благодаря его повышенной скорости шифрования, которая сохраняется при достаточно большой разрядности параметров вычислительно сложной задачи, лежащей в его основе. Основная алгебраическая операция, которая используется в криптосхеме Cayley-Purser, - это операция умножения матриц размера 2x2, заданных над кольцом $\boldsymbol{Z}_n$, где n – трудно разложимое на множители число, которое является одним из элементов открытого ключа. Для вычисления $n = pq$ пользователю необходимо сгенерировать сильные простые числа p и q. Значение этих чисел является секретным. Число n должно иметь разрядность 1024 бит или более. Высокая скорость защитного преобразования данных, обеспечиваемая алгоритмом Cayley-Purser за счёт того, что как процедура зашифровывания, так и процедура расшифровывания осуществляется всего за две операции умножения матриц. Это даёт преимущество перед такими распространенными алгоритмами открытого шифрования RSA и Эль-Гамаля, где необходимо выполнить операции возведения в большую степень по модулю 1024 битового числа. Далее разберем основные принципы работы криптосхемы Cayley-Purser.

Размер блоков шифруемого сообщения, необходимо сделать таким, чтобы его возможное максимальное двоичное значение не превышала числа $n - 1$. В четыре позиции матрицы 2x2 (помечаем её как m) помещаем четыре блока сообщений. Зашифрованную матрицу-сообщение m посылаем пользователю, где она расшифровывается, процедурой, обратной процедуре зашифровывания.

Генерация секретного ключа состоит в следующем:

1. Генерируется два сильных простых числа p и q.

2. Вычисляется модуль $n = pq$.

3. Генерируются неперестановочные матрицы $(k) \in \mathrm{GL}(2, \mathbf{Z}_n)$ и $(a) \in \mathrm{GL}(2, \mathbf{Z}_n)$.

Генерация открытого ключа состоит в следующем.

1. Вычисляется матрица $(b) = (k)^{-1}(a)^{-1}(k)$.

2. Вычисляется матрица $(g) = (k)^r$, где r – случайное число.

Открытым ключом является модуль n и матрицы $(b),\ (a), (g)$.

Процедура зашифровывания матрицы-сообщения (m) состоит в следующем:

1. Отправитель сообщения (m) генерирует случайное число t.

2. Затем он вычисляет матрицу $(s) = (g)^t$.

3. Затем вычисляет матрицу $(e) = (s)^{-1}(a)(s)$ и матрицу $(c) = (s)^{-1}(b)(s)$.

4. Отправитель шифруете матрицу-сообщение (m) в соответствии с формулой $(m') = (c)(m)(c)$ и отправляет шифртекст (m') и матрицу (e) владельцу открытого ключа, выступающего в роли получателя сообщения.

Процедура расшифровывания состоит в следующем:

1. Получатель, приняв матрицу (m') и матрицу (e), вычисляет матрицу $(h) = (k)^{-1}(e)(k)$.

2. Затем он расшифровывает шифртекст по формуле $(m) = (h)(m')(h)$.

Рассматриваемая криптсхема работает корректно. Действительно, при расшифровывании выполняются следующие вычисления:

$$(h) = (k)^{-1}(e)(k) = (k)^{-1}\left((s)^{-1}(a)(s)\right)(k) = (s)^{-1}\left((k)^{-1}(a)(k)\right)(s).$$

В последнем преобразовании использована коммутативность матрицы (s), являющейся степенью матрицы (e), с матрицей (e). Далее имеем

$$(h)=(s)^{-1}\left((k)^{-1}(a)(k)\right)(s)=(s)^{-1}\left((k)^{-1}(a)^{-1}(k)\right)^{-1}(s)=$$
$$=(s)^{-1}\left(b\right)^{-1}(s)=\left((s)^{-1}\left(b\right)(s)\right)^{-1}=(c)^{-1}.$$

То есть получено обратное значение ключа шифрования (c), использованного отправителем для шифрования сообщения. Поскольку расшифрование выполняется по формуле $(m)=(h)(m')(h)$, то получаем

$$(h)(m')(h)=(h)\left((c)(m)(c)\right)(h)=(c)^{-1}\left((c)(m)(c)\right)(c)^{-1}=$$
$$=\left((c)^{-1}(c)\right)(m)\left((c)(c)^{-1}\right)=(m).$$

Таким образом, схема открытого шифрования Cayley-Purser работает корректно.

Рассмотрим вопрос стойкости данного способа открытого шифрования. Безопасность криптоалгоритма Cayley-Purser базируется на предположении, что решения задачи поиска сопрягающей матрицы является вычислительно сложной (практически нереализуемой) задачей. В работе [107] было показано, что в случае криптосхемы MOR [108] также используется предполагаемая вычислительная трудность этой задачи, однако оказалось, что она может быть решена за полиномиальное время. Этот известный результат дал основание поставить задачу нахождения вычислительно эффективного способа решения базовой задачи алгоритма Cayley-Purser.

Далее приводится анализ стойкости криптосхемы Cayley-Purser и показывается, что решение задачи сопрягающего элемента позволяет вычислить секретный ключ и не имеет сверхполиномиальной стойкости, т.е. криптосхема Cayley-Purser не является безопасной даже для значений *n* очень большого размера (10000 бит и более).

Анализ безопасности алгоритма Cayley-Purser

В криптосхеме Cayley-Purser открытым ключом является модуль n и матрицы $(b),\ (a), (g)$. Для расшифровывания шифртекста достаточно знать секретную матрицу (k). Имеется следующая связь между секретной матрицей (k) и матрицами открытого ключа $(b),\ (a), (g)$:

$$(b) = (k)^{-1}(a)^{-1}(k), \tag{5.1}$$

причем известна матрица (g), которая перестановочна с матрицей (k), поскольку (g) вычисляется по формуле $(g) = (k)^r$ для некоторого неизвестного значения *r*. Значение r не будет использоваться в анализе приводимом ниже. Будет использован факт перестановочности матриц (g) и (k). Использование данного факта выражается в том, что матрица (g) задает циклическую (коммутативную) подгруппу матриц, которая содержит секретную матрицу (k). Пусть

$$(b) = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix}; \quad (a)^{-1} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}; \ (k) = \begin{pmatrix} x & y \\ z & w \end{pmatrix}.$$

Покажем, что неизвестные элементы *x*, *y*, *z* и *w* могут быть вычислены путем решения системы линейных уравнений над кольцом $\mathbb{Z}_n$.Матричное уравнение (2.15) перепишем в виде $(k)(b) = (a)^{-1}(k)$, а затем в виде

$$\begin{pmatrix} x & y \\ z & w \end{pmatrix}\begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}\begin{pmatrix} x & y \\ z & w \end{pmatrix} \ \Rightarrow$$

$$\begin{pmatrix} xb_{11} + yb_{21} & xb_{12} + yb_{22} \\ zb_{11} + wb_{21} & zb_{12} + wb_{22} \end{pmatrix} = \begin{pmatrix} xa_{11} + za_{12} & ya_{11} + wa_{12} \\ xa_{21} + za_{22} & ya_{21} + wa_{22} \end{pmatrix}. \tag{5.2}$$

Выражение (5.2) позволяет записать систему линейных уравнений, в которой неизвестными являются элементы сопрягающей матрицы (k), следующего вида:

$$\begin{cases} (b_{11}-a_{11})x + b_{21}y - a_{12}z + 0\cdot w \equiv 0 \bmod n \\ b_{12}x + (b_{22}-a_{11})y + 0\cdot z - wa_{12} \equiv 0 \bmod n \\ -a_{21}x + 0\cdot y + (b_{11}-a_{22})z + wb_{21} \equiv 0 \bmod n \\ 0\cdot x - a_{21}y + b_{12}z + (b_{22}-a_{22})w \equiv 0 \bmod n \end{cases}. \quad (5.3)$$

Поскольку система однородных линейных уравнений (5.3) имеет ненулевые решения, причем не менее $\approx n$ решений, что следует из того, что при наличии решения $\begin{pmatrix} x & y \\ z & w \end{pmatrix}$ сразу же получаем множество решений $\begin{pmatrix} \lambda x & \lambda y \\ \lambda z & \lambda w \end{pmatrix} = \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}\begin{pmatrix} x & y \\ z & w \end{pmatrix}$, определяемых всеми возможными значениями $\lambda \in \mathbf{Z}_n$, такими, что НОД(λ, n) = 1. Получаем случай аналогичный рассмотрению задачи поиска сопрягающего элемента, имевшей место при рассмотрении безопасности криптосхемы MOR в работе [107]. Учитывая уравнение расшифровывания криптосхемы Cayley-Purser, легко видеть, что все указанные решения эквиваленты как ключ расшифровывания, т.е. нахождение любого из них означает взлом этой криптосхемы.

Действительно, расшифровывание шифртекста выполняется с использованием матрицы (h), вычисляемой по матрицам (m') и (e) следующим путем по любой из матриц $\lambda(k)$:

$$(h) = (\lambda(k))^{-1}(e)(\lambda(k)) = \lambda^{-1}(k)^{-1}(e)\lambda(k) = \\ = (k)^{-1}(e)\lambda(k).$$

Последнее соотношение показывает эквивалентность всех решений, соответствующих разным значениям $\lambda \in \mathbf{Z}_n$, таким, что НОД(λ, n) = 1.

Однако ранг определителя системы уравнений (5.3) не превышает значения 2, т.е. эта система содержит не более двух линейно независимых уравнений, поэтому имеется не менее n^2 решений (учитываются решения для случаев НОД(λ, n) $\neq$ 1). Легко заметить, что любые два уравнений в системе (5.3) линейно

независимы, поэтому на самом деле имеется n^2 решений. Ввиду большого размера значения n найти одно из эквивалентных решений только по системы (5.3) практически невозможно. Требуется найти другие линейно независимые уравнения, в которые в качестве неизвестных входят элементы секретной матрицы (k). Такие уравнения могут быть сгенерированы из того условия, что (k) перестановочна с (g). Для этого сгенерируем случайное натуральное число μ и вычислим матрицу (b_μ):

$$(b_\mu) = (g)^{-\mu}(a)^{-1}(g)^{\mu}, \tag{5.4}$$

и матрицу (z_μ) по аналогичной формуле:

$$(z_\mu) = (g)^{-\mu}(b)(g)^{\mu}. \tag{5.5}$$

Вычисленная матрица (z_μ) равна матрице (z'_μ), которая могла бы быть вычислена по секретной матрице (k) и матрице (b_μ) по формуле:

$$(z'_\mu) = (k)^{-1}(b_\mu)(k). \tag{5.6}$$

Действительно, имеем

$$\begin{aligned}(z_\mu) &= (g)^{-\mu}(b)(g)^{\mu} = (g)^{-\mu}\left((k)^{-1}(a)^{-1}(k)\right)(g)^{\mu} = \\ &= \left((g)^{-\mu}(k)^{-1}\right)(a)^{-1}\left((k)(g)^{\mu}\right) = \left((k)^{-1}(g)^{-\mu}\right)(a)^{-1}\left((g)^{\mu}(k)\right) = \\ &= (k)^{-1}\left((g)^{-\mu}(a)^{-1}(g)\right)(k) = (k)^{-1}\left(\left(b_\mu\right)\right)(k) = (z'_\mu).\end{aligned}$$

В результате получаем еще одно матричное уравнение с неизвестной матрицей (k), которое имеет вид

$$(z_\mu) = (k)^{-1}(b_\mu)(k) \text{ или } (k)(z_\mu) = (b_\mu)(k). \tag{5.7}$$

Выбирая различные значения $\mu \in \{1, 2, 3, ...\}$, можно получить множество различных матричных уравнений вида (5.27, которому соответствует система линейных уравнений

$$\begin{cases} \left(z_{\mu 11}-b_{\mu 11}\right)x+z_{\mu 21}y-b_{\mu 12}z+0\cdot w\equiv 0 \bmod n \\ z_{\mu 12}x+\left(z_{\mu 22}-b_{\mu 11}\right)y+0\cdot z-b_{\mu 12}w\equiv 0 \bmod n \\ -b_{\mu 21}x+0\cdot y+\left(z_{\mu 11}-b_{\mu 22}\right)z+z_{\mu 21}w\equiv 0 \bmod n \\ 0\cdot x-a_{21}y+z_{\mu 12}z+\left(z_{\mu 22}-a_{22}\right)w\equiv 0 \bmod n \end{cases}, \tag{5.8}$$

в которой коэффициенты представляют собой элементы вычисленных матриц (b_μ) и (z_μ). Очевидно, что нет смысла генерировать много дополнительных уравнений, поскольку более трех линейно независимых уравнений не может быть получено. Последнее связано с тем, что расширение исходной системы линейных уравнений присоединением к ней произвольного числа дополнительных уравнений, получаемых при различных значениях μ, не даст возможности получить только одно решение. Расширенной системе будет удовлетворять любое решение из множества ранее упомянутых эквивалентных решений вида $\begin{pmatrix} \lambda x & \lambda y \\ \lambda z & \lambda w \end{pmatrix}=\begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}\begin{pmatrix} x & y \\ z & w \end{pmatrix}$, что непосредственно видно из перестановочности диагональной матрицы $\begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix}$ с любой другой матрицей. Но может ли иметь место случай, что дополнительные уравнений системы (5.8) не дадут ни одного нового линейно независимого уравнения к системе (5.7)? Такая возможность сомнительна, поскольку при получении матричных уравнений (5.7) существенно использовался тот факт, что матрица (k) перестановочна с матрицей (g). То есть появляющиеся уравнения системы (5.8) должны ограничить число решений, но всегда оставляя решения, дающие матрицы, которые неперестановочны с (g). Такое ограничение связано с появлением дополнительных уравнений, которые линейно не могут быть выражены через четыре уравнения системы (5.3). Но тогда появление хотя бы одного нового линейно независимого уравнения приведет к тому, что число решений уменьшится в n раз, т.е. из n^2 решений останутся n решений. Если из указанных n

решений исключить чрезвычайно малую долю решений, соответствующих условию НОД(λ, n) $\neq 1$, то останутся только эквивалентные решения.

Поскольку решение систем линейных уравнений имеет полиномиальную сложность, то вычисление матрицы, эквивалентной секретно матрице (k) в криптосхеме Cayley-Purser осуществимо за достаточно короткое время даже при очень больших размерах числа n (десятки и сотни тысяч бит). При этом решение задачи факторизации для взлома этой криптосистемы совсем не требуется. Использование в ней матриц, заданных простым конечным полем обеспечивает примерно такую же стойкость, как и задание матриц над кольцом $\mathbf{Z}_n$.

5.2. Экспериментальное подтверждение результативности атаки на криптосхему Cayley-Purser

Рассмотренная в разделе 5.1 атака на криптосистему Cayley-Purser использует повышение ранга матриц коэффициентов системы линейных уравнений при увеличении числа последних за счет добавления линейных уравнений, определяемых процедурами генерации открытых ключей и общего секретного ключа. Это предположение было обосновано из общих соображений. В данном параграфе приводится схема выполненного вычислительного эксперимента на примере конкретного частного случая. В рамках эксперимента были осуществлены аналогичные опыты при различных значениях модуля и различных значениях параметров задания задачи поиска сопрягающего элемента в некоммутативных группах матриц размерности 2x2.

Результаты вычислений при значении модуля p = 206479379.

Использованная формула для генерации открытого ключа имеет вид $(y) = (x)^{-1}(g)(x)$. Использованная формула для генерации общего секрета имеет вид $(z) = (x)^{-1}(y')(x)$, где открытый ключ (y') генерировался по формуле $(y') = (x')^{-1}(g)(x')$, в которой (x') - личный секретный ключ, соответствующий открытому ключу (y'). Были сгенерированы следующие матрицы:

$$(g) = \begin{pmatrix} 13456345 & 12234257 \\ 63455762 & 5325465 \end{pmatrix};\ (x) = \begin{pmatrix} 60598656 & 169593471 \\ 147729601 & 180227714 \end{pmatrix};$$

$$(x)^{-1} = \begin{pmatrix} 46528364 & 175149153 \\ 72006571 & 79988944 \end{pmatrix};\ (y) = \begin{pmatrix} 206311937 & 153292203 \\ 176927485 & 18949252 \end{pmatrix};$$

$$(y') = \begin{pmatrix} 68736584 & 204193853 \\ 141537838 & 156524605 \end{pmatrix}.$$

Матрица коэффициентов системы линейных уравнений, составленной из матричного уравнения для открытого ключа (y) имеет следующий вид:

$$(M_1) = \begin{pmatrix} 192855592 & 143023617 & 153292203 & 0 \\ 176927485 & 0 & 5492907 & 143023617 \\ 194245122 & 200986472 & 0 & 153292203 \\ 0 & 176927485 & 194245122 & 13623787 \end{pmatrix}.$$

Ранг данной матрицы равен числу 2.

Матрица коэффициентов системы линейных уравнений, составленной из матричного уравнения для общего секрета (z) имеет следующий вид:

$$(M_2) = \begin{pmatrix} 174008790 & 64941541 & 163256695 & 0 \\ 101464625 & 0 & 120258610 & 64941541 \\ 2285526 & 86220769 & 0 & 163256695 \\ 0 & 101464625 & 2285526 & 32470589 \end{pmatrix}.$$

Ранг данной матрицы равен числу 2.

Матрица коэффициентов системы линейных уравнений, составленной из матричного уравнения для открытого ключа (y) и матричного уравнения для общего секрета (z) имеет следующий вид:

$$(M_{\text{рез}}) = \begin{pmatrix} 192855592 & 143023617 & 153292203 & 0 \\ 176927485 & 0 & 5492907 & 143023617 \\ 194245122 & 200986472 & 0 & 153292203 \\ 0 & 176927485 & 194245122 & 13623787 \\ 174008790 & 64941541 & 163256695 & 0 \\ 101464625 & 0 & 120258610 & 64941541 \\ 2285526 & 86220769 & 0 & 163256695 \\ 0 & 101464625 & 2285526 & 32470589 \end{pmatrix}.$$

Вычисление показывает, что ранг последней матрицы равен числу 3. Аналогичные результаты были получены в многочисленных экспериментах с различными значениями модуля, что подтверждает практическую реализуемость эффективность атаки на криптосистему Cayley-Purser, обоснованной из общих соображений в предыдущих двух разделах.

5.3. Схемы аутентификации с использованием задачи дискретного логарифмирования в скрытой подгруппе

5.3.1. Задача дискретного логарифмирования в скрытой подгруппе некоммутативной группы

В разделе 5.1 было показано, что конкретные варианты задачи нахождения сопрягающего элемента в конечных некоммутативных группах, используемые в двухключевом криптоалгоритме Cayley-Purser, не являются вычислительно сложными. Можно предположить с достаточным основанием, что построение аналогичных криптосхем с использованием конечных некоммутативных групп не приведет к получению стойких алгоритмов открытого шифрования, если уравнение с неизвестным сопрягающим элементом может быть записано в виде системы линейных уравнений. Поиск конечных групп, для которых такая запись не может иметь места, представляется проблематичным. В связи с этим представляет интерес формулировка скрытой задачи поиска сопрягающего элемента, которая состоит в вычислении сопрягающего элемента W и некоторого натурального значения x из следующего уравнения:

$$Y = W \circ G^x \circ W^{-1}, \qquad (5.9)$$

где Y и G – заданные элементы некоторой некоммутативной группы Г. При известном x имеем задачу поиска сопрягающего элемента, но в исходном уравнении x неизвестно, поэтому требуется одновременно найти элемент X и число x. Данная задача получила название задачи нахождения логарифма в скрытой циклической подгруппе [26]. Нетрудно заметить, что, если пара значений X и x есть некоторое решение, то для произвольного элемента Λ группы Г, являющегося перестановочным со всеми другими элементами группы, пара $(\Lambda X, x)$ удовлетворяет уравнению (5.9), т.е. также является решением рассматриваемой задачи.

Из соотношения (5.9) видно, что операция автоморфного отображения (умножение слева и справа на взаимно-обратные значения) является коммутативной с операцией возведения в степень, т.е. имеет место равенство

$$(W \circ G \circ W^{-1})^x = W \circ (G^x) \circ W^{-1}.$$

Однако, для построения криптосхем с открытым ключом еще требуется обеспечить взаимную коммутативность автоморфных отображений при различных допускаемых значениях элемента W. В частных случаях это свойство не выполняется, поэтому при задании криптосхемы следует указать такие условия выбора элемента W, при которых обеспечивается свойство коммутативности автоморфного отображения $\varphi_W(G) = W \circ G \circ W^{-1}$ группы Г (здесь предполагается, что элемент G пробегает все значения группы Г).

Взаимная коммутативность автоморфных отображений может быть обеспечена условием выбора элемента W из некоторой специфицированной коммутативной подгруппы, содержащейся в группе Г. В работах [26,28] представлены несколько способов вычислительно эффективного задания выбора элементов из одной и той же коммутативной подгруппы. В этих способах подгруппа специфицируется косвенно, т.е. специфицируется процедура случайно генерации элемента W, при которой все генерируемые элементы коммутативны

между собой. Этого оказывается достаточным при построении конкретных криптосхем. При таком косвенном задании коммутативной подгруппы основным требованием является потенциальная возможность генерации достаточно большого числа различных значений W.

Рассмотрим механизм открытого согласования ключа на основе задачи дискретного логарифмирования в скрытой подгруппе. Пусть секретным ключом первого пользователя является пара (W_1, x_1), а второго пользователя – пара (W_2, x_2). Первый и второй пользователи генерируют свои открытые ключи Y_1 и Y_2 по формулам

$$Y_1 = W_1 \circ G^{x_1} \circ W_1^{-1} \quad \text{и} \quad Y_2 = W_2 \circ G^{x_2} \circ W_2^{-1}.$$

В силу выбора элементов W_1 и W_2 из коммутативной подгруппы группы Г и взаимной коммутативности процедуры автоморфного отображения и операции возведения в степень оба пользователя могут сформировать общий секретный ключ K_{12}, обмениваясь друг с другом по открытому каналу их открытыми ключи. При этом каждый из них использует только свой личный секретный ключ и открытый ключ другой стороны сеанса связи. Первый пользователь вычисляет общий секретный ключ следующим образом:

$$\begin{aligned} K_{12} &= W_1 \circ Y_2^{x_1} \circ W_1^{-1} = W_1 \circ \left(W_2 \circ G^{x_2} \circ W_2^{-1}\right)^{x_1} \circ W_1^{-1} = \\ &= W_1 \circ W_2 \circ G^{x_2 x_1} \circ W_2^{-1} \circ W_1^{-1}, \end{aligned}$$

Второй пользователь выполняет следующие вычисления:

$$\begin{aligned} K'_{12} &= W_2 \circ Y_1^{x_2} \circ W_2^{-1} = W_2 \circ \left(W_1 \circ G^{x_1} \circ W_1^{-1}\right)^{x_2} \circ W_2^{-1} = \\ &= W_2 \circ W_1 \circ G^{x_2 x_1} \circ W_1^{-1} \circ W_2^{-1} = W_1 \circ W_2 \circ G^{x_2 x_1} \circ W_2^{-1} \circ W_1^{-1} = \\ &= K_{12}. \end{aligned}$$

Таким образом, имеет место равенство $K_{12} = K'_{12}$, т.е. оба пользователя вычислили одинаковое секретное значение (ключ) в виде некоторого элемента конечной группы Г. Этот общий секрет может быть использован для шифрования сообщений или шифрования сеансовых ключей. На основе этого механизма

нетрудно составить алгоритм открытого шифрования [26] и алгоритм коммутативного шифрования [104].

Представляет интерес также и разработка схем строгой аутентификации на основе задачи дискретного логарифмирования в скрытой подгруппе. Далее разрабатывается протокол строгой взаимной аутентификации двух уделенных абонентов и протокол аутентификации с нулевым разглашением секрета.

5.3.2. Схема строгой аутентификации

Существует два основных вида протоколов аутентификации удаленных пользователей – строгая и нестрогая аутентификация. Строгая аутентификация состоит в выполнении процедуры подтверждения подлинности некоторого субъекта по факту знания им некоторого секретного ключа, при которой сам секрет в открытом виде не предъявляется. Пусть пользователи участвующие в процедуре взаимной аутентификации знают подлинные открытые ключи друг друга, сформированные по формулам

$$Y_1 = Q_1^{w_1} \circ G^{x_1} \circ Q_1^{-w_1} \quad \text{и} \quad Y_2 = Q_2^{w_2} \circ G^{x_2} \circ Q_2^{-w_2}, \tag{5.10}$$

где Q – некоторый специфицированный элемент группы Г, который обладает достаточно большим простым порядком. Задание параметра Q фактически задает способ выбора секретного элемента X как элемента циклической подгруппы генерируемой элементом Q. Механизм генерации открытых ключей формулам (5.10) может быть распространен на произвольное число пользователей. При этом можно использовать некоторый стандартный механизм распределения открытых ключей, например, с использованием удостоверяющих центров.

В протоколе будем применим также и некоторую специфицированную хэш-функцию F_h, в качестве аргумента которой могут быть использованы элементы группы Г. При наличии у пользователей аутентичных открытых ключей протокол взаимной аутентификации двух удаленных пользователей выглядит следующим

образом, где знак «∘» обозначает групповую операцию в используемой конечной некоммутативной группе Г (рис 5.1).

1. Первый пользователь генерирует случайный элемент R_1 группы Г и направляет R_1 второму пользователю.

2. Второй пользователь, используя полученный элемент R_1, вычисляет следующее: элемент $K = Q^{w_2} \circ Y_1^{x_2} \circ Q^{-w_2}$, значения $h_1 = F_h(R_1), h_2 = F_h(K), h_3 = F_h(KR_1)$ и элемент $Z = K^{h_1} \circ R_1^{h_2} \circ K^{h_3}$. Затем он генерирует случайный элемент R_2 группы Г и отправляет элементы Z и R_2 первому пользователю.

3. Первый пользователь, используя сгенерированный им случайный элемент R_1 группы Г, вычисляет следующее: элемент $K = Q^{w_1} \circ Y_2^{x_1} \circ Q^{-w_1}$, значения $h_1 = F_h(R_1), h_2 = F_h(K), h_3 = F_h(KR_1)$ и элемент $Z' = K^{h_1} \circ R_1^{h_2} \circ K^{h_3}$. Затем он сравнивает элементы Z и Z'. Если выполняется равенство $Z = Z'$, то он делает вывод о подлинности второго пользователя.

4. Первый пользователь, используя полученный элемент R_2 и уже вычисленный элемент $K = Q^{w_1} \circ Y_2^{x_1} \circ Q^{-w_1}$, находит следующее: значения $h_1' = F_h(R_2), h_2' = F_h(K), h_3' = F_h(KR_2)$ и элемент $Z'' = K^{h_1'} \circ R_2^{h_2'} \circ K^{h_3'}$. Затем он и отправляет элементы Z'' второму пользователю.

5. Второй пользователь, используя сгенерированный им случайный элемент R_2 группы Г и уже вычисленный элемент $K = Q^{w_2} \circ Y_1^{x_2} \circ Q^{-w_2}$, находит значения $h_1' = F_h(R_2), h_2' = F_h(K), h_3' = F_h(KR_2)$ и элемент $Z^* = K^{h_1'} \circ R_2^{h_2'} \circ K^{h_3'}$. Затем он сравнивает элементы Z'' и Z^*. Если выполняется равенство $Z'' = Z^*$, то он делает вывод о подлинности второго пользователя.

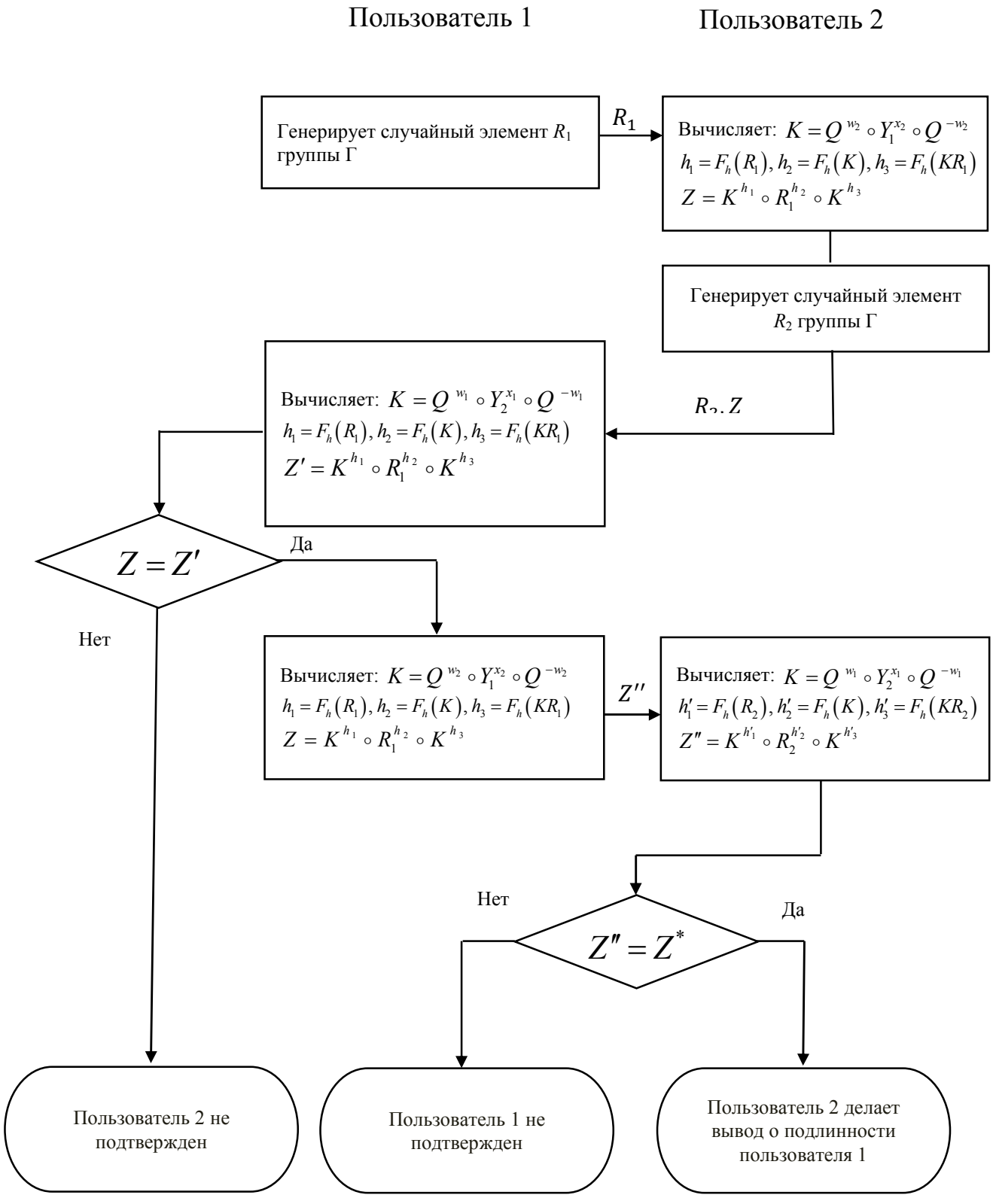

Рис. 5.1 Протокол строгой аутентификации удаленных абонентов

Из протокола видно, что секретные значения не отправляются по открытому каналу связи, т.е. он относится к типу протоколов строгой аутентификации. При этом аутентификация выполняется по открытым ключам, поэтому не требуется

предварительное использование защищенных каналов для обмена секретными значениями. Вычисление секретного ключа по открытому ключу относится к типу силовых атак на двухключевые криптсхемы. Данный вариант атаки на рассмотренный протокол строгой аутентификации связан с решением задачи дискретного логарифмирования в скрытой подгруппе, которая имеет высокую вычислительную трудность при выборе соответствующих размеров параметров, используемых в протоколе.

Возможен и второй тип атаки на протокол, целью которого является вычисление общего секретного ключа первого и второго пользователей, генерируемых каждым из них по своему секретному ключу и открытому ключу другого пользователя. Это теоретически может быть сделано по передаваемым по открытому каналу значениям. Например, по значениям Z и R_1 из уравнения $Z = K^{h_1} \circ R_1^{h_2} \circ K^{h_3}$ или по значениям Z^* и R_2 из уравнения $Z^* = K^{h'_1} \circ R_2^{h'_2} \circ K^{h'_3}$. Однако данная задача является вычислительно более сложной, чем задача дискретного логарифмирования в скрытой подгруппе, поскольку в последних двух уравнениях имеются четыре неизвестных элемента.

5.3.3. Протокол с нулевым разглашением

Криптосхемы с нулевым разглашением [109,110] секрета лежат в основе процедур строгой аутентификации удаленных пользователей в информационно-телекоммуникационных системах. Протоколы такого типа основаны на некоторых вычислительно трудных задачах. Протоколы данного типа реализуют следующую обобщенную схему аутентификации удаленного пользователя А (доказывающий), который в ходе выполнения протокола демонстрирует знание решения некоторой вычислительно сложной задачи при некотором конкретном выборе значений ее параметров, которые играют роль открытого ключа (ОК) пользователя А. В ходе осуществления протокола пользователь А доказывает другому пользователю Б (который называется проверяющим), что он знает личный секретный ключ связанный с ОК, владельцем которого является

пользователь А (ОК пользователь Б узнает из справочника открытых ключей или цифрового сертификата, подписанных удостоверяющим центром). Имеются следующие виды построения протоколов с нулевым разглашением секрета: 1) реализация в форме многораундовой интерактивной процедуры, 2) в виде двухшаговой процедуры и 3) в виде трехшаговой процедуры. В протоколах каждого из этих типов проверяющий с заданным уровнем гарантии (задается низкое значение вероятности того, что нарушитель может обмануть проверяющего) убеждается, что доказывающий знает секрет, связанный с ОК пользователя А. Это соответствует тому, что удаленный пользователь А доказал свою подлинность. При этом в ходе протокола пользователь А не передает какой-либо информации о своем личном секрете (это подчеркивается термином нулевое разглашение).

Для построения известных протоколов с нулевым разглашением секрета использованы следующие вычислительно сложные задачи 1) нахождение дискретного логарифма в конечной циклической группе или конечном поле, 2) разложение целых чисел на простые множители, 3) извлечение квадратных корней по модулю, в качестве которого задается трудно разложимое число, 4) нахождение сопрягающего элемента в конечных некоммутативных группах и другие.

В данном разделе предлагается реализация протокола с нулевым разглашением секрета с использованием вычислительной трудности задачи нахождения дискретного логарифма в скрытой циклической подгруппе некоторой конечной некоммутативной группы Г матриц размера 2x2, определенных над простым полем $GF(p)$, характеристика которого (число p) имеет достаточно большой размер.

Пусть заданы две неперестановочные между собой матрицы $G \in \Gamma$ и $U \in \Gamma$, причем порядок каждой из них является простым числом достаточно большого размера. В качестве личного секретного ключа генерируется пара случайных чисел (w, x), где $w < \omega$; $x < \omega$; ω – значение порядка матриц G и U. В качестве ОК,

связанного с секретным ключом (w, x), задается матрица Y, которая вычисляется по формуле $Y = U^w \circ (G^x) \circ U^{-w}$.

Пусть ОК доказывающего является матрица Y, а его личным секретным ключом является пара чисел (w, x). Предлагаемый протокол включает z-кратное выполнение следующего трехшагового раунда (Рис. 5.2):

1. Доказывающий формирует равновероятные случайные значения $k \le p - 1$ и $u \le p - 1$, затем вычисляет числа $k' = kx \bmod \omega$, $u' = ux \bmod \omega'$ и матрицу $Q = U^{u'} \circ (G^{k'}) \circ U^{-u'}$ и направляет матрицу Q проверяющему.

2. Проверяющий направляет доказывающему случайный равновероятный бит r (т.е. с вероятностью 0,5 имеем $r = 1$ и с вероятностью 0,5 имеем $r = 0$).

3. При получении значения $r = 1$ доказывающий направляет проверяющему пару чисел (u', k'), а при получении значение $r = 0$ – пару чисел (u, k).

Проверяющий делает вывод о правильности ответа на его однобитовый запрос, если имеет место $Q = U^{-u'} \circ (G^{k'}) \circ U^{-u'}$ (в случае $r = 1$) или $Q = U^{u} \circ (Y^{k}) \circ U^{-u}$ (в случае $r = 0$). Вероятность того, что нарушитель может выдать себя за пользователя А в одном раунде, равна 0,5. Вероятность того, что нарушитель успешно ответит на случайные запросы проверяющего в каждом из z раундов, равна 2^{-z}. Выбором достаточно большого значения числа раундов можно добиться сколь угодно малой вероятности того, что нарушителя не удастся обнаружить.

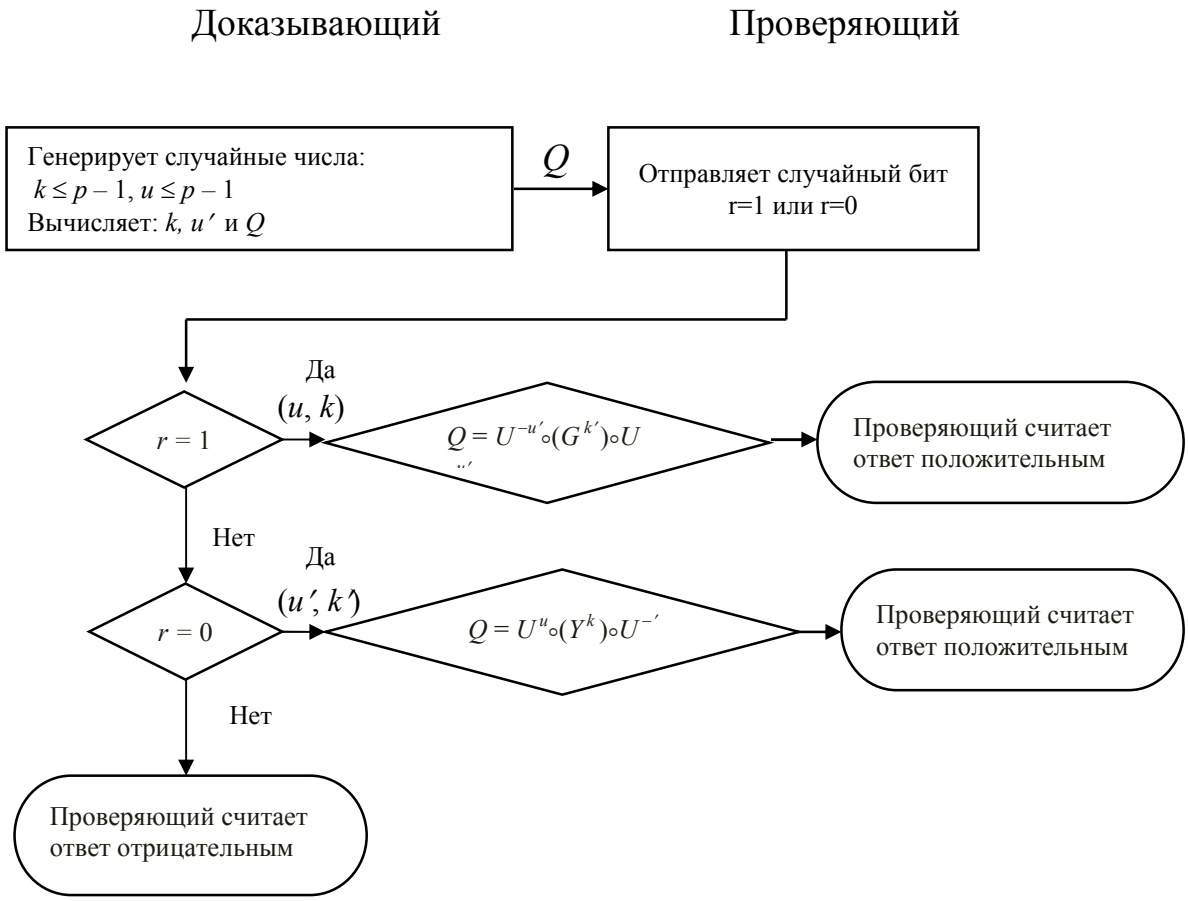

Рис. 5.2 Протокол аутентификации удаленного пользователя с нулевым разглашением секрета

Нулевая утечка информации о секретном ключе доказывающего в ходе протокола обеспечивается тем, что получаемые проверяющим пары чисел являются равновероятными случайными значениями. Действительно, множество соотношений между случайными значениями, аналогичные соотношениям, получаемым проверяющим от доказывающего, доказывающий может сгенерировать самостоятельно.

Для построения протоколов аутентификации удаленных пользователей, обеспечивающих нулевую утечку информации о секретном ключе представляет интерес задача «факторизации» элементов конечной некоммутативной группы Г. Эта задача состоит в представлении некоторого заданного элемента Y в виде произведения элементов $V \in \Gamma' \subset \Gamma$, $W \in \Gamma'' \subset \Gamma$ и $G \in \Gamma$, где G – элемент, являющийся известным параметром протокола; Γ' и Γ'' – две коммутативные

подгруппы достаточно большого порядка, такие что элементы одной подгруппы являются неперестановочными с элементами второй подгруппы. То есть рассматриваемая задача «факторизации» состоит в решении уравнения $Y = WGV$ относительно неизвестных W и V. Эта задача представляется не менее сложной чем задача поиска сопрягающего элемента, поскольку вторая решается как частный случай первой задачи.

Рассмотрим протокол с нулевым разглашением, основанный на предполагаемой трудности задачи «факторизации» элементов конечной некоммутативной группы Г задание коммутативных подгрупп Г′ и Г″ можно реализовать, с помощью двух неперестановочных между собой элементов Q и P достаточно больших простых порядков q и r, таких, что каждый из этих элементов неперестановочен с G. Также как и в предыдущем варианте протокола элементы W и V группы Г формируются путем выбора двух случайных чисел $w < q$ и $x < r$ и вычисления $W = Q^w$ и $V = P^x$. При таком задании подгрупп Г′ и Г″ описанный выше протокол преобразуется к следующему конкретному варианту реализации.

Секретным ключом пользователя А является пара случайных чисел w и x, а открытый ключ вычисляется по формуле $Y = Q^w G P^x$. В процессе выполнения протокола многократно (z раз) осуществляется следующий трехшаговый раунд:

1. Доказывающий (пользователь А) генерирует случайные значения $w < q$ и $x < r$, вычисляет элемент $R = Q^t G P^u$. Затем он отправляет проверяющему значение R.

2. Проверяющий направляет доказывающему запрос в виде равновероятного случайного бита r ($r = 1$ или $r = 0$).

3. При получении значения $r = 1$, доказывающий отправляет в ответ числа $T' = TV$ и $U' = UW$. При получении значения $r = 0$ доказывающий направляет в ответ элементы T и U.

Для обоснования данного протокола полезно следующее утверждение (обобщение теоремы об уникальности открытых ключей [26]), где $^\vee$ - знак, обозначающий неперестановочность двух элементов некоммутативной группы.

Теорема 3.1. Пусть $Q^\vee P$, $Q^\vee G$, $P^\vee G$, , где порядками элементов Q, P и G являются простые числа q, r и g соответственно. Пусть также имеет место условие $Q^\vee(GPG^{-1})$. Тогда для всех различных пар значений i и j элементы $Z_{i,j} = Q^i GP^j$, где $i \in \{1,2,\ldots, q\}$, $j \in \{1,2,\ldots, r\}$, попарно различны.

Доказательство: Допустим противное: $Q^i PG^j = Q^{i'} PG^{j'}$ для некоторых пар значений (i, j) и (i', j'), таких, что $i \neq i'$ и $j \neq j'$. Тогда $Q^{i-i'}G = GP^{j'-j}$, следовательно $Q^{i-i'} = GP^{j'-j}G^{-1}$, где $z = i - i'$. Поскольку $i - i' \in \{1,2,\ldots, q\}$, то НОД$(q,\ i - i') = 1$ и существует целое $h = z^{-1} = (i - i')^{-1} \bmod q$, при котором имеем $Q = GP^{h(j-j')}G^{-1}$. Поскольку элемент P имеет простой порядок, то при некотором значении $d = [h(j' - j)]^{-1} \bmod q$ имеем $Q^d = GPG^{-1}$. Из последней формулы следует, что $Q^\wedge(GPG^{-1})$. Полученное противоречие доказывает утверждение. (Для случая $(i, j) \neq (i', j')$ и $i = i'$, а также случая $(i, j) \neq (i', j')$ и $j \neq j'$ доказательство тривиально.)

Утверждение 2 имеет значение для криптосхемы с формированием открытых ключей вида $Y = Q^w GP^x$. Его практическое значение состоит в том, что при выборе элементов Q, P и G, удовлетворяющих условию утверждения, устанавливается размер ключевого пространства, т.е. дает гарантии того, что пространство открытых ключей достаточно велико и не существуют пары секретных ключей, порождающих одинаковые открытые ключи (свойство уникальности открытых ключей).

5.3.4. Выбор конечных групп матриц

В описанных в предыдущих разделах криптосхемах используется некоммутативная группа Г. Для реализации стойкого протокола группа Г должна удовлетворять следующим требованиям: порядок ее должен быть достаточно

большим, она должна содержать в себе коммутативные подгруппы достаточно большого порядка. С практической точки зрения важным является требование вычислительной эффективности групповой операции. Эти требованиям удовлетворяют конечные группы невырожденных матриц различной размерности.

Порядок группы матриц, заданных над конечным полем $GF(p)$, размерности $n \times n$ выражается следующей формулой:

$$\Omega = (p^n - 1)(p^n - p)(p^n - p^2)\ldots(p^n - p^{n-1}) = \prod_{i=0}^{n-1} p^i \left(p^{n-i} - 1\right).$$

Известно, что некоммутативные группы Г содержат подгруппы, порядок которых равен каждому простому делителю порядка группы Г. Для заданного значения n представляет интерес получение максимального значения размера $|q|$ простого делителя $q|\Omega$ при заданном размере $|p|$ или минимального размера $|p|$ при заданном значении $|q|$. Для простых значений n потенциально возможное значение q содержится в множителе $(p^n - 1)$, который в этом случае разлагается на множители так:

$$(p^n - 1) = (p - 1)(p^{n-1} + p^{n-2} + p^{n-3} + \ldots + 1).$$

Для заданной разрядности $|p|$, выбирая перебирая различные простые числа p, можно добиться, чтобы вторая скобка в последнем выражении была равна nq, где q – простое число размера $|q| \approx (n - 1) \cdot |p|$. Это показывает, что для конечных групп матриц большой простой размерности коммутативные подгруппы большого простого порядка могут быть получены даже в случае сравнительно малых значений размера числа p. В случае матриц малой размерности следует выбирать значения $|p| \geq 160$ бит.

Представляет интерес оценка влияния размерности матриц на размер открытого ключа, обеспечивающего заданный уровень стойкости. Это связано с тем, что с ростом размера матриц растет в общем случае и временная сложность матричного умножения. Размер открытого ключа определяется битовым размером s матриц в заданной КГНМ, который равен $s = n^2|p|$. Учитывая, что значение

стойкости определяется размером наибольшего простого делителя порядка КГНМ и то, что при простых значениях n максимальный простой делитель порядка КГНМ не превышает $|q| \approx (n-1)\cdot|p|$, т.е. $|p| \geq |q|\cdot(n-1)^{-1}$, получаем оценку

$$s = n^2|p| \geq |q|\cdot n^2(n-1)^{-1} = \frac{n^2\,|\,q\,|}{n-1} \approx n\cdot|\,q\,|.$$

Последнее соотношение показывает, что минимальный размер матриц растет примерно пропорционально n, т.е. числу строк в матрицах используемой КГНМ. Это означает, что наибольший интерес представляет реализация разработанных и известных криптосхем, использующих задачи формулируемые над некоммутативными конечными группами, над конечными группами матриц размерности 2x2.

Для повышения вычислительной эффективности разработанных криптосхем представляет интерес выбор других типов конечных некоммутативных групп. Однако такой выбор является достаточно ограниченным. Наибольший интерес представляют конечные некоммутативные группы векторов, предложенные в работах [26,28,104], которые вводятся следующим образом.

Векторы конечного m-мерного векторного пространства могут быть представлены в виде $(a, b,\ldots, q)$ или в виде $a\mathbf{e} + b\mathbf{i} + \ldots + q\mathbf{w}$, где e, i, …, w – некоторые формальные базисные векторы; $a, b,\ldots, q \in GF(p^s)$ – координаты вектора. Операция сложения двух векторов задается следующей формулой:

$$(a,b,...,q) + (x,y,...,z) = (a+x, b+y, ..., q+z).$$

Векторы вида $\varepsilon\mathbf{v}$, где $\varepsilon \in GF(p^s)$ и $\mathbf{v}$ – некоторый формальный базисный вектор, представляют собой компоненты вектора и по сути являются одномерными векторами. Операция умножения двух векторов определяется как попарное перемножения всех компонентов векторов-операндов (сомножителей) в соответствии с формулой

$$(a\mathbf{e} + b\mathbf{i} + ... + q\mathbf{w}) \circ (x\mathbf{e} + y\mathbf{i} + ... + z\mathbf{w}) = ax\mathbf{e}\circ\mathbf{e} + ay\mathbf{e}\circ\mathbf{i} + ... + az\mathbf{e}\circ\mathbf{w} +$$
$$+ bx\mathbf{i}\circ\mathbf{e} + by\mathbf{i}\circ\mathbf{i} + ... + bz\mathbf{i}\circ\mathbf{w} + ... + qx\mathbf{w}\circ\mathbf{e} + qy\mathbf{w}\circ\mathbf{i} + ... + qz\mathbf{w}\circ\mathbf{w}\ ,$$

в которой предполагается выполнить замену произведений пар базисных векторов на некоторый однокомпонентный вектор, который задается так называемой таблицей умножения базисных векторов (ТУБВ). Координата этого однокомпонентного вектора называется структурным коэффициентом. Если ТУБВ придает операции умножения векторов свойство ассоциативности, то конечное векторное пространство с определенной таки способом операцией умножения векторов будет представлять собой конечную ассоциативную алгебру [112], которая в частных случаях является векторным конечным кольцом (ВКК) или конечным полем, заданным в явной векторной форме.

Для повышения быстродействия протоколов открытого шифрования, коммутативного шифрования, открытого согласования ключа и аутентификации с нулевым разглашением секрета представляет интерес использованием КГНМ, элементами которых являются матрицы размерности 2x2, заданные над конечным полем, заданным в явной векторной форме (см. раздел 4.8). Этот способ повышения быстродействия представляет существенный интерес при реализации протоколов на многопроцессорных вычислителях.

Выводы к главе 5

1. Задача поиска сопрягающего элемента в конечной некоммутативной группе, возникающая в криптоанализе криптосистемы Cayley-Purser, решается за полиномиальное время, осуществляя сведение этой задачи к нахождению решения системы линейных уравнений, которые легко получаются из параметров открытого ключа этой криптосистемы.
2. Алгоритм открытого шифрования Cayley-Purser не отвечает требованиям безопасности даже при достаточно больших значениях параметров секретного и открытого ключей.
3. На основе трудности задачи нахождения логарифма в скрытой циклической подгруппе разработан протокол строгой аутентификации.

4. На основе трудности задачи нахождения логарифма в скрытой циклической подгруппе разработан протокол с нулевым разглашением.

5. На основе трудности задачи «факторизации» элементов некоммутативной группы по заданным коммутативным подгруппам разработан протокол с нулевым разглашением. Доказана теорема об уникальности открытых ключей, задаваемых процедурой их генерации по данному протоколу.

6. Заключение

В диссертационной работе получено решение актуальной задачи разработки методов построения протоколов ЭЦП, обладающих расширенной функциональностью и повышенным уровнем обеспечиваемой ими безопасности, на основе которых разработаны практичные протоколы цифровой подписи, и способа псевдовероятностного защитного преобразования информации, обеспечивающего повышение производительности процедур такого типа, в том числе получены следующие основные результаты:

1. Разработан метод повышения уровня информационной безопасности и протокол утверждаемой групповой подписи, отличающийся использованием вычислений в конечном поле $GF(p)$, где число $p-1$ является трудно факторизуемым, что обеспечивает безопасность протокола при появлении прорывного решения одной из следующих двух трудных вычислительных задач: дискретного логарифмирования по простому модулю и факторизации. На основе метода разработан

2. Разработан метод повышения уровня информационной безопасности и протокол слепой подписи, отличающийся использованием вычислений в конечном поле $GF(p)$, где число $p-1$ является трудно факторизуемым

3. Разработан метод реализации и протокол утверждаемой групповой подписи, основанный на вычислительной сложности задачи нахождения дискретного логарифма на эллиптической кривой и отличающийся вычислением открытого ключа в виде суммы точек эллиптической кривой и генерацией маскирующих коэффициентов в виде криптографической контрольной суммы зависящей от секретного ключа руководителя группы подписантов.

4. Разработан метод построения и протокол утверждаемой групповой подписи, отличающийся использованием процедур генерации и проверки подлинности ЭЦП, специфицируемых российским стандартом ГОСТ Р 34.10-2012.

5. Разработан метод реализации и протокол коллективной групповой подписи.

6. Разработан метод построения и протокол комбинированной коллективной подписи.

7. Разработан метод устранения необходимости использования внутренней инфраструктуры открытых ключей для обеспечения возможности раскрытия групповой подписи руководителем, отличающийся тем, что параметры, маскирующие открытые ключи подписантов, вычисляются с использованием двухкратного вычисления хэш-функции, зависящей от секретного ключа руководителя группы подписантов.

8. Разработан метод построения безопасных защитных преобразований информации с использованием алгебраических операций, отличающихся комбинированием операций из конечных алгебраических структур различного типа и разбиением входного блока данных на подблоки различного размера.

9. Разработан метод псевдовероятностных защитных преобразований, стойкий к атакам с принуждением отправителя и получателя сообщений к раскрытию ключа шифрования, отличающийся реализацией вычислений в конечных полях, заданных в явной векторной форме, благодаря чему обеспечивается повышение производительности.

Перспективы развития выполненного исследования состоят в разработке протоколов групповой ЭЦП , коллективной ЭЦП для групповых подписантов, в которых подпись свободна от включения третьего дополнительного параметра U, что упростит строение этих протоколов, их реализацию и использование имеющейся на практике инфраструктуры открытых ключей при практическом применении протоколов данных типов. Дальнейшее развитие направления разработки способов алгебраических алгоритмов защитных преобразований можно связать с их построением на основе некоммутативных конечных ассоциативных алгебр размерности 3, что позволит повысить производительность алгоритмов защитного преобразования алгебраического типа.

Список опубликованных работ по теме диссертационного исследования

Публикации в ведущих рецензируемых журналах, рекомендуемых ВАК для публикации результатов диссертационных исследований:

1. Синев В.Е. Повышение уровня безопасности протокола групповой цифровой подписи, основанного на механизме маскирования открытых ключей// Известия СПбГЭТУ «ЛЭТИ». 2016. № 6. С. 21-25.
2. Молдовян А.А., Галанов А.И., Синев В.Е. Утверждаемая групповая подпись: новые протоколы // Вопросы защиты информации. 2016. № 2. С. 44-50.
3. Галанов А.И., Захаров Д.В., Молдовян Д.Н., Синев В.Е. Протоколы слепой подписи на основе двух вычислительно трудных задач // Вопросы защиты информации. 2009. № 4. С.2-7.
4. Доронин С.Е., Молдовян Н.А., Синев В.Е. Конечные расширенные поля для алгоритмов электронной цифровой подписи // Информационно-управляющие системы. 2009. № 1. С. 33-40.
5. Доронин С.Е., Молдовяну П.А., Синев В.Е. Векторные конечные поля: задание умножения векторов большой четной размерности // Вопросы защиты информации. 2008. № 4(83). С.2-7.

Публикации в других изданиях:

6. Дернова Е.С., Костина А.А., Синев В.Е. Выбор характеристики и степени расширения конечных полей для схем цифровой подписи на основе конечных групп матриц // Материалы VI Санкт-Петербургской межрегиональной конференции «Информационная безопасность регионов России (ИБРР-2009)». Санкт-Петербург, 28-30 октября. СПб.: СПОИСУ, 2009. С.96
7. Дернова Е.С., Костина А.А., Молдовяну П.А., Синев В.Е. Примитивы алгоритмов цифровой подписи: конечные группы матриц над векторными полями // XIC-

Петербургская международная конф. «Региональная информатика-2008 (РИ-2008)» СПб, 22-24 октября 2008 г. / Материалы конференции. СПб, 2008. с. 96-97.

8. Дернова Е.С., Костина А.А., Синев В.Е. Выбор параметров задания конечных групп матриц для построения алгоритмов электронной цифровой подписи // Научно-технические проблемы в промышленности. СПб, 12-14 ноября 2008 г. Материалы конференции. СПб, 2008. с. 40-41.
9. Костина А.А., Аль-Рахми Р.Я., Синев В.В. Подходы к разработке шифров на основе операций матричного умножения // Инновационная деятельность в Вооруженных силах Российской Федерации: Труды всеармейской научно-практической конференции. 25-26 ноября 2010, г. Санкт-Петербург. СПб.: ВАС, 2010. С. 57-62.
10. Борков П.В., Костина А.А., Аль-Рахми Р.Я., Синев В.В. Блочное шифрование с использованием некоммутативных операций // XII Санкт-Петербургская международная конференция Региональная информатика «РИ-2010» СПб, 20-22 октября 2010г. Материалы конференции. СПб, 20. сс. 94-95
11. Дернова Е.С., Костина А.А., Синев В.Е. Выбор параметров задания конечных групп матриц для построения алгоритмов электронной цифровой подписи // Труды конф. «Научно-технические проблемы в промышленности», Санкт-Петербург, 12-14 ноября 2008 г. СПб., 2008. С. 291-295.
12. Гурьянов Д. Ю., Мирин А.Ю., Синев В.Е. Метод решения задачи дискретного логарифмирования в конечных группах двухмерных векторов // Инновационная деятельность в Вооруженных силах Российской Федерации: Труды всеармейской научно-практической конференции. 10-11 декабря 2009 г., Санкт-Петербург. СПб.: ВАС, 2009. С. 228-233.
13. Дернова Е.С., Костина А.А., Синев В.Е. Выбор порождающего элемента в схемах цифровой подписи на основе конечных групп матриц и векторов // Материалы VI Санкт-Петербургской межрегиональной конференции «Информационная безопасность регионов России (ИБРР-2009)». Санкт-Петербург, 28-30 октября. СПб.: СПОИСУ, 2009. С.96-97.

14. Галанов А.И., Березин А.Н., Синёв В.Е. Протокол утверждаемой групповой цифровой подписи на основе двух трудных задач // IX Санкт-Петербургская международная конференция Информационная безопасность регионов России «ИБРР-2015» СПб, 28-30 октября 2015г. Материалы конференции. СПб, 2015. сс. 101-102.

Список использованной литературы

1. Шаньгин В. Ф. Защита информации в компьютерных системах и сетях — М.: ДМК, 2012. — 592 с.
2. Иванов М. А., Чугунков И.В. Криптографические методы защиты информации в компьютерных системах и сетях.- М.: НИЯУ МИФИ, 2012.-400с.
3. Молдовян Д.Н., Молдовян Н.А. Введение в теоретические основы криптоситем с открытым ключом.-- СПб.: Изд–во ГУМРФ им. адмирала С.О. Макарова, 2016. — 68 с..
4. Лось А.Б., Нестеренко А.Ю., Рожков М.И. Криптографические методы защиты информации. –М.: Юрайт, 2016.-474с
5. Menezes A.J., Van Oorschot P.C., Vanstone S.A. Handbook of Applied Cryptography. CRC Press, Boca Raton, FL, 1997.- 780 p.
6. Алферов А. П., Зубов А. Ю., Кузьмин А. С., Черемушкин А. В. Основы криптографии. - М.: Гелиос АРВ. – 2005. – 480с.
7. Романец Ю. В., Тимофеев П. А., Шаньгин В. Ф. Защита информации в компьютерных системах и сетях. — М.: Радио и связь, 2001. — 376 с.
8. Молдовян А.А., Молдовян Н.А., Гуц Н.Д., Изотов Б.В. Криптография: скоростные шифры. – СПб, БХВ-Петербург. 2002. – 495 с.
9. Moldovyan N.A., Moldovyan A.A. Data-Driven Block Ciphers for Fast Telecommunication Systems. – Auerbach Publications. Talor & Francis Group. New York, London. 2007.- 185 p.
10. Moldovyan N.A., Moldovyan A.A., Eremeev M.A. A class of data-dependent operations // International Journal of Network Security. 2006. Vol. 2. No. 3. P. 187-204.
11. Moldovyan A.A., Moldovyan N.A., Moldovyanu P.A. Architecture Types of the Bit Permutation Instruction for General Purpose Processors // Springer LNGC. 2007.

Vol. XIV. P. 147-159 / 3d Int. Workshop IF&GIS'07 Proc. St.Petersburg, May 28-29, 2007.

12. Rivest R., Shamir A., Adleman A. A method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communication of the ACM. — 1978. —V. 21. — N. 2. — P. 120–126.

13. Коутинхо С. Введение в теорию чисел. Алгоритм RSA. – М.: Постмаркет, 2001. – 323 с.

14. V.S. Miller. Use of elliptic curves in cryptography, Cryptology, CRYPTO'85, LNCS 218, 1985, pp.417-426.

15. K. Koblitz. Elliptic curve cryptosystems, Mathenmatics of Computation 48(177), 1987, pp.203-209.

16. Anshel I., Anshel M., Goldfeld D. An Algebraic Method for Public Key Cryptography // Mathematical Research Letters. 1999. Vol. 6. P. 287–291.

17. Ki Hyoung Ko , Sang-Jin Lee , Jung Hee Cheon , Jae Woo Han , Ju-Sung Kang , Choonsik Park. New Public-Key Cryptosystem Using Braid Groups // Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology, p.166-183, August 20-24, 2000.

18. Seong-Hun Paeng , Kil-Chan Ha , Jae Heon Kim , Seongtaek Chee , Choonsik Park. New Public Key Cryptosystem Using Finite Non Abelian Groups // Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology, p.470-485, August 19-23, 2001.

19. In-Sok Lee, Woo-Hwan Kim, Daesung Kwon, Sangil Nahm, Nam-Seok Kwak, and Yoo-Jin Baek. On the Security of MOR Public Key Cryptosystem // Proceedings of the ASIACRYPT 2004, LNCS. 2004. Vol. 3329 pp. 387-400.

20. Mahalanobis A. The Diffie-Hellman key exchange protocol and non-abelian nilpotent groups // Israel Journal of Mathematics. 2008. V. 165. P. 161-187.

21. Ko K.H., Lee S.J., Cheon J.H., Han J.W., Kang J.S., Park C. New Public-Key Cryptosystems Using Braid Groups // / Proceedings of the international conference Advances in Cryptology – Crypto 2000 / Lecture Notes in Computer Science. Springer-Verlag, 2000. Vol. 1880. P. 166–183.

22. Lee E., Park J.H. Cryptanalysis of the Public Key Encryption Based on Braid Groups // In: Advances in Cryptology -- EUROCRYPT 2003 / LNCS. Springer, Heidelberg. 2003. V. 2656. P. 477-489.

23. Myasnikov A., Shpilrain V., Ushakov A. A Practical Attack on a Braid Group Based Cryptographic Protocol // In: Advances in Cryptology -- CRYPTO'05 / LNCS. Springer, Heidelberg. 2005. V. 3621. P. 86--96.

24. Seong-Hun Paeng, Kil-Chan Ha, Jae Heon Kim, Seongtaek Chee and Choonsik Park, New public key cryptosystem using finite non-abelian groups, Crypto 2001, LNCS 2139, 470-485.

25. Lee E., Park J.H. Cryptanalysis of the Public Key Encryption Based on Braid Groups // Advances in Cryptology – Eurocrypt 2003 / Lecture Notes in Computer Science. Springer-Verlag, 2003. Vol. 2656. P. 477–489.

26. Moldovyan D.N., Moldovyan N.A. A New Hard Problem over Non-Commutative Finite Groups for Cryptographic Protocols // Proceedings of the international conference MMM-ACNS 2010 / I. Kotenko and V. Skormin (Eds.): MMM-ACNS 2010, LNCS. Springer, Heidelberg. 2010. V. 6258. P. 183-194.

27. Moldovyan A.A., Moldovyan N.A., Shcherbacov V.A. Non-commutative finite rings with several mutually associative multiplication operations // The Fourth Conference of Mathematical Society of the Republic of Moldova dedicated to the centenary of Vladimir Andrunachievici (1917-1997), June 28 - July 2, 2017, Chisinau, Proceedings CMSM4, 2017, p. 133-136.

28. Молдовян Д.Н. Конечные некоммутативные группы как примитив криптосистем с открытым ключом // Информатизация и связь. 2010. № 1. С.62-65.

29. Shor P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on quantum computer // SIAM Journal of Computing. 1997. Vol. 26. P. 1484–1509.

30. Verma G.K. A Proxy Blind Signature Scheme over Braid Groups // Int. Journal of Network Security. 2009. V.9, No 3. P. 214–217.

31. Jozsa R. Quantum algorithms and the fourier transform // Proc. Roy. Soc. London. 1998. Ser A. V. 454. P. 323-337.

32. Ekert, A., Jozsa, R. Quantum computation and Shor's factoring algorithm // Rev. Mod. Phys. 1996. V. 68. P. 733.

33. Ettinger M., Hoyer P. On quantum algorithms for noncommutative hidden subgroups // Proc. 16th STACS. 1999. P. 478-487.

34. Соловьев Ю.П., Садовничий В.А., Шавгулидзе Е.Т., Белокуров В.В. Эллиптические кривые и современные алгоритмы теории чисел.-М., Ижевск. Институт компьютерных исследований, 2003.- 191 с.

35. N. Koblitz, A.J. Menezes. Another Look at Provable Security // Journal of Cryptology. 2007. V. 20. P. 3-38.

36. Diffie W., Hellman M.E. New Directions in Cryptography // IEEE Transactions on Information Theory. 1976, Vol. IT-22. pp. 644 – 654.

37. Болотов А. А., Гашков С. Б., Фролов А. Б., Часовских А. А. Элементарное введение в эллиптическую криптографию: Алгебраические и алгоритмические основы. – М.: КомКнига/URSS, 2006. – 328 с.

38. Болотов А. А., Гашков С. Б., Фролов А. Б., Часовских. Элементарное введение в эллиптическую криптографию: Протоколы криптографии на эллиптических кривых. – М.: КомКнига/URSS, 2006. – 274 с.

39. Menezes A.J. and Vanstone S.A. Elliptic Curve Cryptosystems and Their Implementation. // Journal of cryptology. 1993. V. 6. No 4. P. 209-224.

40. N. Koblitz. A Course in Number Theory and Cryptography.- Springer-Verlag. Berlin, 2003. - 236 p.

41. Смарт Н. Криптография. – М.: Техносфера, 2005. – 528 с.

42. Rabin M.O. Digitalized signatures and public key functions as intractable as factorization. – Technical report MIT/LCS/TR-212, MIT Laboratory for Computer Science, 1979.

43. Молдовян Н.А. Извлечение корней по простому модулю как криптографический примитив // Вестник СПбГУ. Сер. 10, 2008. Вып. 1. С. 100-105.

44. Молдовян А.А., Молдовян Н.А. Новые алгоритмы и протоколы для аутентификации информации в АСУ // Автоматика и телемеханика. 2008. № 7. С.157-169.

45. Гурьянов Д.Ю., Дернова Е.С., Избаш В.И., Молдовян Д.Н. Алгоритмы электронной цифровой подписи на основе сложности извлечения корней в конечных группах известного порядка // Информационно-управляющие системы. 2008. № 5. С.33-40.

46. Молдовян Н.А. Теоретический минимум и алгоритмы цифровой подписи. – СПб: БХВ – Петербург, 2010. – 304 с.

47. Бухштаб А. А. Теория чисел. – М.: Просвещение, 1966. – 384 с.

48. Виноградов И. М. Основы теории чисел. – М.: Наука, 1972. – 167 с.

49. Shcherbacov V.A. Generating Cubic Equations as a Method for Public Encryption // Buletinul Academiei de Stiinte a Republicii Moldova. Matematica. 2015. N. 3 (79). P. 60-71.

50. Gordon J. Strong primes are easy to find, Advances in cryptology – EUROCRYPT’84, Springer-Verlag LNCS, 1985, vol. 209, pp. 216-223.

51. Акритас А. Основы компьютерной алгебры с приложениями. – М.: Мир, 1994. – 544 с.

52. Молдовян Н.А. Введение в криптосистемы с открытым ключом. – СПб: БХВ – Петербург, 2005. –286 с.

53. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке СИ. – М.: ТРИУМФ, 2002. – 816 с.

54. Венбо Мао. Современная криптография. Теория и практика. — М., СПб., Киев: Издательский дом «Вильямс», 2005. — 763 с.

55. ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms // IEEE Transactions on Information Theory. 1985. V. IT-31, No. 4. P. 469 – 472.

56. Schnorr C.P. Efficient signature generation by smart cards // Journal of Cryptology. 1991. V. 4. P. 161-174.

57. Молдовян Д.Н. Новый механизм формирования подписи в схемах ЭЦП, основанных на сложности дискретного логарифмирования и факторизации. // Вопросы защиты информации. 2005. №4. С. 2-7.

58. Moldovyan A.A., Moldovyan D.N., and Gortinskaya L.V. Cryptoschemes based on new signature formation mechanism // Computer Science Journal of Moldova. 2006. Vol. 14. No 3(42). P.397-411.

59. Молдовян Д.Н., Молдовян Н.А. Новые схемы ЭЦП с сокращенной длиной подписи. // Вопросы защиты информации 2006. №3 (74). С. 7-12.

60. D. Pointcheval, J. Stern. Security Arguments for Digital Signatures and Blind Signatures // Journal of Cryptology. 2000. V. 13. P. 361-396.

61. Schnorr C.P. Efficient identification and signatures for smart cards // Advances in cryptology – CRYPTO’89 / Springer-Verlag LNCS. 1990. V. 435. P. 239-252.

62. Pieprzyk J., HardjonoTh., Seberry J. Fundumentals of Computer Security. – Springer-verlag. Berlin, 2003. – 677 p.

63. Молдовян Н. А., Молдовян А. А., Еремеев М. А. Криптография: от примитивов к синтезу алгоритмов. – СПб.: БХВ-Петербург, 2004. – 446 с.

64. National Institute of Standards and Technology, NIST FIPS PUB 186, "Digital Signature Standard", U.S. Department of Commerce, 1994.

65. International Standard ISO/IEC 14888-3:2006(E). Information technology – Security techniques – Digital Signatures with appendix – Part 3: Discrete logarithm based mechanisms.

66. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. ГОСТ Р 34.10-94. – Госстандарт России. М., Издательство стандартов. – 18 с.

67. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. ГОСТ Р 34.10-2001. – Госстандарт России. М., ИПК Издательство стандартов. – 12 с.

68. Ростовцев А.Г., Маховенко Е.Б. Теоретическая криптография – СПб., АНО НПО «Профессионал», 2005. – 480 с.

69. ANSI X9.62 and FIPS 186-2. Elliptic curve signature algorithm, 1998.

70. Rothe J. Complexity Theory and Cryptology.- Berlin, Heidelberg: Springer-Verlag, 2005.- 479 p.

71. Tahat N. M. F., Ismail E. S., Ahmad R. R. A New Blind Signature Scheme Based On Factoring and Discrete Logarithms // International Journal of Cryptology Research. 2009. No 1 (1). P. 1–9.

72. Tahat N.M.F., Shatnawi S.M.A., Ismail E.S. A New Partially Blind Signature Based on Factoring and Discrete Logarithms // Journal of Mathematics and Statistics. 2008. No 4(2). 124–129.

73. Minh N. H., Binh D. V., Giang N. T., Moldovyan N. A. Blind Signature Protocol Based on Difficulty of Simultaneous Solving Two Difficult Problems // Applied Mathematical Sciences. 2012. V. 6. No 139. P. 6903– 6910.

74. 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.

75. 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.

76. 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.

77. Молдовян Н.А., Галанов И.А., Еремеев М.А. Множественная подпись: новые решения на основе понятия коллективного открытого ключа // Информационно-управляющие системы. 2008. № 1. С. 34-36.

78. Ананьев М.Ю., Гортинская Л.В., Молдовян Н.А. Протоколы коллективной подписи на основе свертки индивидуальных параметров // Информационно-управляющие системы. 2008. № 2. С.22-27.

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

80. Молдовян Н.А., Хо Нгок Зуй, Сухов Д.К., Протокол слепой коллективной подписи на основе сложности задачи дискретного логарифмирования // Известия СПБГЭТУ «ЛЭТИ». 2011. № 3. С. 19-24.

81. Moldovyan A.A., Moldovyan N.A. Blind Collective Signature Protocol Based on Discrete Logarithm Problem // Int. Journal of Network Security. 2010. Vol. 11, No 2. P. 106-113.

82. Moldovyan N.A. Blind Collective Signature Protocol // Computer Science Journal of Moldova. 2011. Vol. 19. No. 1. P. 80–91.

83. Moldovyan N.A. Blind Signature Protocols from Digital Signature Standards // Int. Journal of Network Security. 2011. Vol. 13, No 1. P. 22-30.

84. Nguyen M.H., Ho D.N., Luu D.H., Moldovyan A.A., Moldovyan N.A. On Functionality Extension of the Digital Signature Standards // Advanced Technologies for Communications (ATC) Proceedings of the 2011 International Conference on Advanced Technologies for Communications. Vietnam, Da Nang, August 3 – 5, 2011. P. 6 - 9 (2011).

85. Молдовян Н.А., Дернова Е.С., Молдовян Д.Н. Расширение функциональности стандартов электронной цифровой подписи России и Беларуси // Вопросы защиты информации. 2011. № 2. С. 8-14.

86. Молдовян Н.А., Дернова Е.С., Молдовян Д.Н. Протоколы слепой и коллективной подписи на основе стандарта ЭЦП ДСТУ 4145-2002 // Вопросы защиты информации. 2011. № 2. С. 14-18.

87. Pieprzyk J., HardjonoTh., Seberry J. Fundumentals of Computer Security. Springer-verlag. Berlin, 2003. - 677 p.

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

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

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

91. Konheim A.G. Cryptography // John Wiley & Sons. New York. 1981.- 360 p.

92. Б.Л. ван дер Варден. Алгебра.- СПб, М., Краснодар. Лань, 2004.- 623 с.

93. Каргаполов М. И., Мерзляков Ю.И. Основы теории групп. – М.: Физматлит, 1996. – 287 с.

94. Кострикин А. И. Введение в алгебру. Основы алгебры. – М.: Физматлит, 1994. – 320 с.

95. Молдовян А.А., Молдовян Д.Н., Левина А.Б. Протоколы аутентификации с нулевым разглашением секрета.-- СПб.: Изд–во Университета ИТМО, 2016. — 53 с.

96. Zierler N. Primitive trinomials whose degree is a Mersenne exponent // Information and Control. 1969. vol. 15, no. 1, pp. 67-69.

97. Молдовян Н.А., Рахья Р.Я. Синтез алгебраических блочных шифров с использованием операций над двоичными многочленами // Вопросы защиты информации. 2012. № 1. С. 2-7.

98. Молдовян Н.А., Аль-Рахми Р.Я. Синтез блочных шифров на основе операций матричного умножения // Вопросы защиты информации. 2011. № 2. С. 2-8.

99. Moldovyan A.A., Moldovyan N.A., Moldovyanu P.A.. Architecture Types of the Bit Permutation Instruction for General Purpose Processors // Springer LNGC. 2007. Vol. XIV. pp. 147-159 / 3d Int. Workshop IF\&GIS'07 Proc. St.Petersburg, May 28-29, 2007. St. Petersburg, Russia.

100. Moldovyan N.A., Moldovyan A.A. Data-driven block ciphers for fast telecommunication systems. Auerbach Publications. Talor & Francis Group. New York, London. 2008.- 185 p.

101. Moldovyan N.A., Moldovyanu P.A., Summerville D.H. On Software Implementation of Fast DDP-Based Ciphers. // International Journal of Network Security. 2007. vol. 4, no. 1. P.81-89.

102. Moldovyan N.A., Moldovyan A.A. Innovative cryptography.- Charles River Media, Boston, Massachusetts, 2006.- 386 pp.

103. Moldovyan N.A. Fast Signatures Based on Non-Cyclic Finite Groups // Quasigroups and related systems. 2010. V.18. P. 83-94.

104. Moldovyan D.N., Moldovyan N.A. Cryptoschemes over hidden conjugacy search problem and attacks using homomorphisms // Quasigroups and Related Systems. 2010. Vol. 18. P. 177-186.

105. Березин А. Н., Биричевский А. Р., Молдовян Н. А., Рыжков А. В. Способ отрицаемого шифрования // Вопросы защиты информации. 2013. № 2. С. 18-21.

106. Moldovyan, N.A., Moldovyanu, P.A.: New Primitives for Digital Signature Algorithms: Vector Finite Fields // Quasigroups and Related Systems. 2009 Vol. 17. P. 271-282.

107. Куприянов И.А. Методы защиты информации на основе вычислений в конечных группах матриц // Автореф. дисс. …канд. тех. наук. СПб, 2013.

108. Seong-Hun Paeng, Kil-Chan Ha, Jae Heon Kim, Seongtaek Chee, Choonsik Park. New public key cryptosystem using finite non-abelian groups // Crypto 2001 (J. Kilian, ed.), LNCS, vol. 2139, Springer-Verlag, 2001, pp. 470–485.

109. Демьянчук А.А., Молдовян Н.А., Рыжков А.В. Выбор «идеальных» параметров в схеме двухшаговой аутентификации и коммутативном шифре // Известия СПбГЭТУ «ЛЭТИ». 2013. № 8. С. 15-18.

110. Демьянчук А.А, Молдовян Д.Н., Молдовян А.А. Алгоритмы открытого шифрования в протоколах с нулевым разглашением секрета // Вопросы защиты информации. 2013. № 2. С. 22-27.

111. Демьянчук А.А., Молдовян Д.Н., Новикова Е.С., Гурьянов Д.Ю. Подход к построению криптосхем на основе нескольких вычислительно трудных задач // Информационно-управляющие системы. № 2. 2013. С. 60-66.

112. Кузьмин А.А., Марков В.Т., Михалев А.А., Михалев А.А., Михалев А.В., Нечаев А.А. Криптографические алгоритмы на группах и алгебрах // Фундаментальная и прикладная математика, 2015, том 20, № 1, с. 205—222.

113. Васильев, П.Н. Криптографические протоколы. Схемы групповой подписи : учеб. пособие / П.Н. Васильев, Е.Б. Маховенко (Александрова). – СПб: Изд-во Политехнического ун-та, 2012. – 68 с.

114. Александрова, Е.Б. Применение постквантовой и гомоморфной криптографии в задачах кибербезопасности / Е.Б. Александрова, Н.Н. Шенец // Неделя науки СПбПУ: материалы научного форума с международным участием. Междисциплинарные секции и пленарные заседания институтов. – СПб., 2015. – С. 9–17.

115. Маховенко (Александрова), Е.Б. Использование эллиптических кривых с комплексным умножением для реализации ГОСТ Р 34.10−2001 в маломощных вычислительных устройствах / Е.Б. Маховенко (Александрова), Д.С. Павлов. // Сб. трудов Научной сессии МИФИ-2009. XVI Всероссийская научная конференция. Проблемы информационной безопасности в системе высшей школы. – М., 2009.

116. Libert B., Mouhartem F., Nguye Kh. A Lattice-Based Group Signature Scheme with Message-Dependent Opening // International Conference on Applied Cryptography and Network Security. Proceedings ACNS 2016: Applied Cryptography and Network Security pp 137-155.

117. Laguillaumie F., Langois A., Libert B., Stehlé D. Lattice-based group signature scheme with logarithmic signature size // Proc. of 19th International Conference on the Theory and Application of Cryptology and Information Security (December 1-5, 2013). P.41-61.

118. Phong Q. Nguyen, Jiang Zhang, Zhenfeng Zhang, Simpler efficient group signature from lattices // Proc. of 18th IACR International Conference on Practice and Theory in Public-Key Cryptography March 30-April 1, 2015). P.401-426.

119. Langlois A., San Ling, Khoa Nguyen, Huaxiong Wang. Lattice-based group signature scheme with verifier-local revocation // Proc. of 17th International Conference on Practice and Theory in Public-Key Cryptography (May 26-28, 2014). P.345-361.

120. Alamélou Q,, Blazy O,, Cauchie S., Gaborit Ph. A code-based group signature scheme // Designs, Codes and Cryptography. 2017, Vol. 82. No 1-2. P. 469–493.
121. Shah F., Patel H. A Survey of Digital and Group Signature // International Journal of Computer Science and Mobile Computing, Vol.5 Issue.6, June- 2016, pg. 274-278
122. Qi Su, Wen-Min Li, Improved Group Signature Scheme Based on Quantum Teleportation // International Journal of Theoretical Physics. 2016. Vol. 53. No. 4. P. 1208.
123. Yanhua Zhang, Yupu Hu, Wen Gao, Mingming Jiang. Simpler Efficient Group Signature Scheme with Verifier-Local Revocation from Lattices // KSII Transactions on Internet and Information Systems (Monthly Online Journal (eISSN: 1976-7277)). 2016. Vol. 10, No.1. (DOI 10.3837/tiis.2016.01.024).
124. Libert B, Peters Th., Yung M. Short Group Signatures via Structure Preserving Signatures: Standard Model Security from Simple Assumptions // Proc. of 35th Annual Annual Cryptology Conference (August 16-20, 2015) P.296-316.
125. Run Xie, Chunxiang Xu, Chanlian He, Xiaojun Zhang. A new group signature scheme for dynamic membership // International Journal of Electronic Security and Digital Forensics (archive). 2016. Vol. 8. No. 4 (DOI 10.1504/IJESDF.2016.079446).
126. San Ling, Khoa Nguyen, Huaxiong Wang, Group signature from lattices: simpler, tighter, shorter, ring-based // Proc. of 18th IACR International Conference on Practice and Theory in Public-Key Cryptography (March 30-April 1, 2015). P.427-449.