Министерство науки и высшего образования Российской Федерации
Санкт-Петербургский политехнический университет Петра Великого
Высшая школа кибербезопасности и защиты информации

Работа допущена к защите
Директор Высшей школы
кибербезопасности и защиты
информации, д.т.н., проф.
_____ Д.П. Зегжда
«___»_____2020 г.

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ДИПЛОМНАЯ РАБОТА

ОТЗЫВ СО СВЯЗЫВАНИЕМ В СХЕМЕ КОЛЬЦЕВОЙ ПОДПИСИ НА РЕШЕТКАХ

по направлению подготовки (специальности)
10.05.04 Информационно-аналитические системы безопасности

Направленность (профиль)
10.05.04_01 Автоматизация информационно-аналитической деятельности

Выполнил
студент гр. 3651004/40101 И.Ш. Рехвиашвили

Руководитель
профессор ВШКиЗИ ИПММ,
д.т.н., доцент Е.Б. Александрова

Санкт-Петербург
2020

РЕФЕРАТ

На 43 с., 8 рисунков, 5 таблиц, 1 приложение.

КЛЮЧЕВЫЕ СЛОВА: РЕШЕТКИ, КОЛЬЦЕВАЯ ПОДПИСЬ, ОТЗЫВ ПРАВА ПОДПИСИ, ШИФРОВАНИЕ С ОТКРЫТЫМ КЛЮЧОМ, ТЕСТ НА РАВЕНСТВО

Предлагается подход к решению задачи отзыва права подписи у участника группы в схеме кольцевой подписи на решетках путем добавления уполномоченной сущности – центра отзыва, осуществляющего проверку наличия сертификата пользователя в списке отзыва.

THE ABSTRACT

43 pages, 8 pictures, 5 tables, 1 application

KEY WORDS: LATTICE, RING SIGNATURE, REVOCATION, PUBLIC KEY ENCRYPTION, EQUALITY TEST

An approach is proposed to solve the problem of revoking the right to sign from a group member in a lattice-based ring signature scheme by adding a revocation center that checks the presence of a member certificate in the revocation list.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В настоящее время электронная подпись является основным механизмом, позволяющим аутентифицировать данные и их источник. Одним из видов электронной подписи является кольцевая подпись. Такая подпись гарантирует, что сообщение было подписано одним из участников группы, однако не предоставляет возможности отследить, кем именно. Схемы кольцевой подписи находят свое применение в таких прикладных областях, как системы электронного голосования, электронной валюты, системы контроля для общественного транспорта, а также сервисы, предоставляющие свои услуги пользователям по подписке. В таких системах отсутствует необходимость уникально идентифицировать каждого пользователя, достаточно лишь убедиться, что ему разрешен доступ к ресурсу.

Так как в рассматриваемых системах требуется наличие возможности исключения пользователя из группы, важной составляющей протокола кольцевой подписи является процедура отзыва права подписи у пользователя группы. Организация отзыва оказывает влияние на эффективность схемы подписи. Так, разные механизмы отзыва отличаются объемом дополнительных вычислений на этапах генерации ключей, формирования и проверки подписи, необходимостью проведения обновлений для участников схемы, а также влиянием на размер подписи.

Традиционные схемы цифровой подписи, основанные на задаче разложения целых чисел на множители и задаче дискретного логарифмирования, не являются стойкими к атакам с использованием квантового компьютера. В связи с этим в настоящее время стоит задача разработки криптографических алгоритмов, стойких к подобного рода атакам. Одним из перспективных направлений в данной области является криптография на решетках.

Объектом исследования является схема кольцевой подписи на решетках.

Предметом исследования является возможность отзыва права подписи у участников схемы кольцевой подписи на решетках.

Цель исследования – организация отзыва права подписи в схеме кольцевой подписи на решетках.

Для достижения поставленной цели необходимо решить следующий ряд *задач*:

- проанализировать свойства схем кольцевой подписи на решетках и выбрать схему подписи в качестве прототипа;
- сравнить механизмы отзыва права подписи в схемах со многими участниками;
- разработать схему кольцевой подписи на решетках, обеспечивающую возможность отзыва права подписи у участников;
- выполнить программную реализацию и оценку времени работы процедур разработанной схемы кольцевой подписи на решетках с механизмом отзыва права подписи.

Теоретическая и методологическая базы исследования. Теоретической основой выпускной квалификационной работы послужили исследования в области криптографии на решетках и протоколов кольцевых подписей. В качестве методологии применялись общенаучные (анализ, обобщение, синтез) и частные (криптографические) методы исследования.

При подготовке выпускной квалификационной работы были использованы материалы таких учебных дисциплин, как «Криптографические методы защиты информации», «Теоретико-числовые методы в криптографии», «Алгебра и теория чисел», «Методы программирования».

Информационную базу исследования составили научные публикации по исследуемой теме, материалы научных конференций, материалы, собранные в процессе прохождения учебной и преддипломной практик.

Степень научной разработанности проблемы. Созданию и исследованию схем кольцевой подписи посвящено значительное число работ, основополагающей среди которых является работа Р. Ривеста, А. Шамира и Я. Тауман [1]. Исследования К. Джентри и К. Пейкерта [2], В. Любашевского [3] посвящены разработке схем подписи на решетках. Подход к построению кольцевой схемы

подписи на решетках был впервые предложен в работе З. Бракерски [4], после чего нашел развитие в работах П. Кейрела [5], Ч. Вана [6], Ш. Вана и Ж. Чжао [7], в которых предлагаются схемы кольцевой подписи на решетках. Проблема отзыва права подписи в групповых подписях рассматривается в работах Г. Атениза [8], Е. Брессона и Дж. Стерна [9], Д. Боне и Х. Шахама [10], Я. Камениша и А. Лысянской [11], Д. Сламанига, Р. Шпрайтцера и Т. Унтерлуггауэра [12].

Научная новизна выпускной квалификационной работы определяется тем, что для решения задачи организации отзыва права подписи предложено применить механизм отзыва со связыванием, а также решены проблемы безопасности разработанной схемы посредством применения белых и черных списков, неинтерактивного доказательства с нулевым разглашением и пороговой схемы разделения секрета.

Практическая значимость. В настоящее время схемы кольцевой подписи находят свое применение в системах электронного голосования, в которых они позволяют реализовать принцип тайного голосования. Также кольцевые подписи применяются в системах электронной валюты для реализации неотслеживаемых транзакций. Помимо этого, такие схемы могут применяться в различных системах контроля доступа к ресурсу, например в системах контроля общественного транспорта, а также системах предоставления пользователям доступа по подписке.

Предлагаемый в работе подход позволяет реализовать в указанных системах возможность исключать участников из группы посредством отзыва у них права подписи без необходимости повторной генерации ключей подписи для всех участников, а также обновления списков отзыва для пользователей системы.

Апробация результатов исследования.

Основные положения и результаты исследований докладывались и обсуждались на 47-й научной конференции с международным участием «Неделя

науки СПбПУ» (Санкт-Петербург, 2018 г.). По итогам участия был получен диплом за лучший доклад на секционном заседании.

По результатам выполнения научного проекта «Организация отзыва для схемы кольцевой подписи» в рамках конкурса грантов для студентов, аспирантов, молодых ученых, молодых кандидатов наук 2019 года была получена премия Правительства Санкт-Петербурга.

Результаты исследований по теме выпускной квалификационной работы отражены в публикации [13].

1. СХЕМА КОЛЬЦЕВОЙ ПОДПИСИ НА РЕШЕТКАХ

Цифровая подпись является механизмом аутентификации данных и их источника. Она представляет собой данные, присоединяемые к передаваемому сообщению и подтверждающие, что автор подписи составил и заверил данное сообщение. Получатель сообщения (проверяющий) с помощью этой подписи может убедиться, что автором сообщения является именно владелец подписи и что целостность данных не была нарушена в процессе передачи. Кроме того, подпись составляется таким образом, чтобы автор подписи не мог затем отрицать перед проверяющим факт подписания [14].

Дальнейшее развитие схем цифровой подписи привело к появлению класса подписей, в которых авторы подписей являются участниками некоторой группы. К такому классу относятся групповая, пороговая и кольцевая подпись, рассматриваемая в данном исследовании.

Безопасность цифровой подписи (то есть трудность подделки, невозможность отказа) обеспечивается сложностью вычислительно трудных задач. Современные исследования показывают, что схемы подписи, основанные на задачах разложения целого числа и дискретного логарифмирования, уязвимы к атакам на квантовом компьютере с использованием полиномиальных алгоритмов разложения числе на множители и дискретного логарифмирования. Это побудило к поиску других вычислительно трудных задачах, позволяющих строить стойкие к подобным атакам криптосистемы. Одним из направлений в решении этой задачи является криптография на решетках. Предполагается, что криптосистемы, основанные на решетках, являются стойкими к атакам на квантовом компьютере, поскольку квантовые алгоритмы решения вычислительно трудных задач на решетках имеют более чем полиномиальную сложность [15].

В данной главе рассматриваются основные свойства кольцевых подписей, характеристики решеток и вычислительно трудные задачи на решетках, а также описывается протокол кольцевой подписи на решетках.

1.1 Свойства кольцевых подписей

Схема кольцевой подписи была впервые предложена Р. Ривестом, А. Шамиром и Я. Тауман в 2001 году в работе «How to leak a secret» [1] в контексте задачи анонимного раскрытия секретных сведений, в которой один из участников группы (например, лицо, анонимно поставляющее сведения о нелегальных действиях в организации) намерен передать секретную информацию третьему лицу. Получатель должен быть уверен, в том, что сообщение было отправлено одним из участников группы, в то время как отправитель заинтересован в том, чтобы его личность не была раскрыта.

Кольцевая подпись является разновидностью групповой подписи. В классической схеме групповой подписи менеджер группы определяет участников группы и предоставляет им ключи, с помощью которых каждый участник может подписать сообщение от лица группы. При этом получатель не может определить по подписи, кем из участников она была создана, однако это может сделать менеджер группы.

Кольцевая подпись, так же как и групповая, позволяет каждому участнику подписывать сообщения от лица группы, однако не предоставляет менеджеру группы механизма аннулирования анонимности. Таким образом, данная схема гарантирует, что сообщение было подписано одним из участников группы, и при этом обеспечивает анонимность участника в пределах данной группы. Отсюда вытекают два важнейших свойства безопасности кольцевых схем подписи:

- невозможность подделки – заключается в неосуществимости операции подписи от лица группы при отсутствии одного из закрытых ключей;
- анонимность автора подписи – заключается в невозможности узнать, какой именно закрытый ключ был использован при создании подписи.

Группа из r участников называется кольцом. С каждым участником кольца ассоциируется пара: открытый ключ P_k и закрытый ключ S_k.

Простейшая схема кольцевой подписи состоит из следующих двух алгоритмов:

– Ring-sign$(m, P_1, P_2, \dots, P_r, S_k)$ – процедура формирования подписи σ для сообщения m по данным открытым ключам $P_1, P_2, \dots, P_r$ участников кольца и закрытому ключу S_k автора подписи.

– Ring-verify(m, σ) – процедура проверки подписи, для сообщения m и подписи σ, включающей в себя открытые ключи участников кольца, определяющая, является ли данная подпись действительной.

На рис. 1.1 представлено взаимодействие основных сущностей, участвующих в схеме кольцевой подписи.

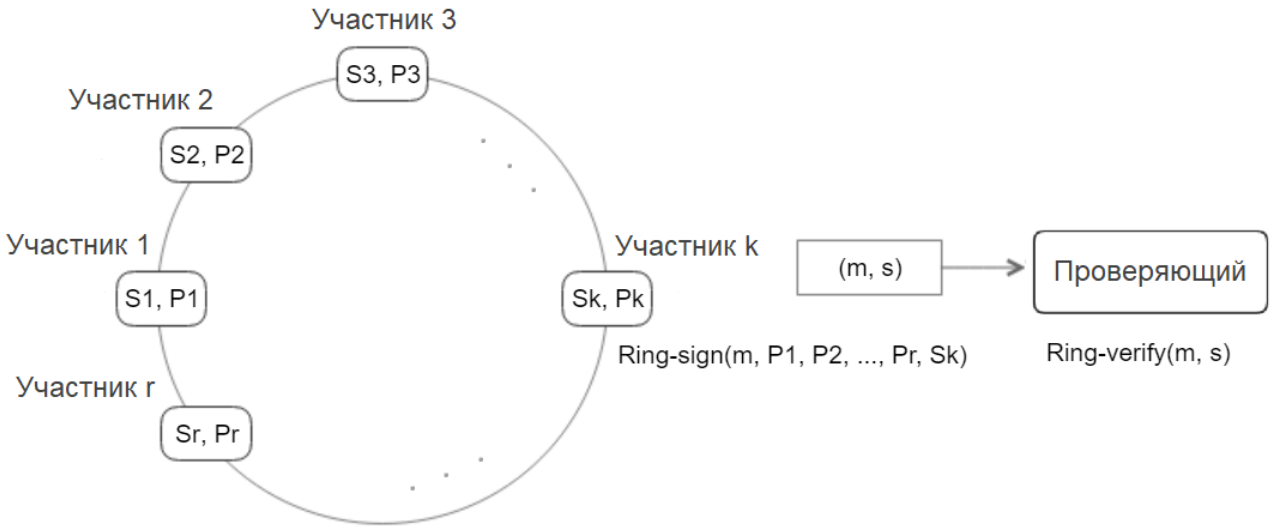

Рисунок 1.1 – Схема кольцевой подписи

В отличие от групповых подписей, в кольцевых подписях отсутствует необходимость взаимодействия между участниками: любой член группы может подписать сообщение, используя свой закрытый ключ и открытые ключи других участников, не запрашивая их подтверждения. Проверка подписи должна удовлетворять условию: для кольца размером r участников проверяющий не сможет определить настоящего автора подписи с вероятностью больше, чем $1/r$.

1.2 Криптография на решетках

Безопасность большинства схем подписи основывается на вычислительной сложности задачи разложения целого числа на множители и задачи дискретного логарифмирования. Однако с появлением в 1994 году алгоритма Шора, использующего квантовые вычисления и способного эффективно (за полиномиальное время) решать данные задачи, стало ясно, что традиционные криптографические протоколы, основанные на них, являются уязвимыми к атакам с использованием квантового компьютера. Таким образом, актуальной задачей в наше время является разработка криптографических алгоритмов, стойких к атакам квантового компьютера. Их изучением занимается постквантовая криптография.

Одной из перспективных математических структур постквантовой криптографии являются решетки. Они лежат в основе таких криптографических примитивов, как:

- шифрование с открытым ключом: NTRUEncrypt, GGH, RLWE-HOM;
- цифровая подпись: NTRUSign, GGH, RLWE-SIG;
- хеш-функции: SWIFFT, LASH.

1.2.1 Свойства решеток

Рассмотрим основные свойства решеток. Пусть $\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_m$ – набор линейно независимых векторов в $\mathbf{R}^n$. Решеткой называется набор линейных комбинаций этих векторов с целочисленными коэффициентами:

$$L(\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_m) = \left\{ \sum_{i=1}^{m} x_i \boldsymbol{b}_i \mid x_i \in \boldsymbol{Z} \right\} = \{B\boldsymbol{x},\ \boldsymbol{x} \in \boldsymbol{Z}^{\boldsymbol{m}}\},$$

где B – это матрица размерности $n \times m$ со столбцами $\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_m$.

Векторы $\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_m$ называются базисом решетки, а число m – размерностью (или рангом). При $m = n$ решетка называется полноранговой.

Матрицей Грама называется матрица

$$\Gamma(\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_m) = \begin{pmatrix} (\boldsymbol{b}_1, \boldsymbol{b}_1) & (\boldsymbol{b}_1, \boldsymbol{b}_2) & \cdots & (\boldsymbol{b}_1, \boldsymbol{b}_m) \\ (\boldsymbol{b}_2, \boldsymbol{b}_1) & (\boldsymbol{b}_2, \boldsymbol{b}_2) & \cdots & (\boldsymbol{b}_2, \boldsymbol{b}_m) \\ \dots & \dots & \dots & \dots \\ (\boldsymbol{b}_m, \boldsymbol{b}_1) & (\boldsymbol{b}_m, \boldsymbol{b}_2) & \cdots & (\boldsymbol{b}_m, \boldsymbol{b}_m) \end{pmatrix}.$$

Число $\det(L) = \sqrt{\det\left(\Gamma(\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_m)\right)}$ называется объемом (или определителем) решетки. Определитель решетки равен объему фундаментального параллелепипеда, построенного на векторах $\boldsymbol{b}_1, \boldsymbol{b}_2, \dots, \boldsymbol{b}_m$.

Для матрицы размерностью ≥ 2 существует бесконечное количество базисов. Матрица перехода между любыми двумя базисами является унимодулярной, то есть ее определитель равен ± 1, таким образом, объем решетки не зависит от выбора базиса.

Так как решетка дискретная, в ней содержится ненулевой вектор минимальной длины, который называется ее кратчайшим вектором. Евклидова норма данного вектора называется первым минимумом решетки и обозначается $\lambda_1(L)$ или $\|L\|$. Также могут использоваться и другие нормы.

1.2.2 Вычислительно трудные задачи

В криптографических системах применяются следующие вычислительно трудные задачи, основанные на решетках [16]:

1. Задача поиска кратчайшего вектора (shortest vector problem, SVP). По заданному базису решетки L найти вектор $\boldsymbol{u} \in L$ такой, что $\|\boldsymbol{u}\| = \|L\|$. Также может использоваться аппроксимированная версия задачи поиска кратчайшего вектора: найти вектор $\boldsymbol{v} \in L$ такой, что $\|\boldsymbol{v}\| \leq \gamma\|L\|$.
2. Задача поиска ближайшего вектора (closest vector problem, CVP). По заданному базису решетки L и вектору $\boldsymbol{v} \in \mathbf{R}^n$ найти ближайший к нему вектор решетки $\boldsymbol{u} \in L$, то есть такой, что $\|\boldsymbol{u} - \boldsymbol{v}\| \leq \|\boldsymbol{w} - \boldsymbol{v}\| \ \forall\, \boldsymbol{w} \in L$. Аппроксимированная версия задачи: найти вектор $\boldsymbol{u} \in L$ такой, что $\|\boldsymbol{u} - \boldsymbol{v}\| \leq \gamma\|\boldsymbol{w} - \boldsymbol{v}\| \ \forall\, \boldsymbol{w} \in L$.
3. Задача поиска наименьшего базиса (smallest basis problem, SBP). Для заданной решетки L найти базис с наименьшей длиной его максимального вектора, то есть найти

$$B: \forall B' \in \{B \in \boldsymbol{R}^{n\times m} \mid L = L(B)\} \ \max_{1\le i\le m}\{\|\boldsymbol{b}_i\|\} \le \max_{1\le i\le m}\{\|\boldsymbol{b}'_i\|\}.$$

4. Задача обучения с ошибками (learning with errors, LWE). Пусть решетка L задана базисом B следующим образом: $L = \{\boldsymbol{z} \in \boldsymbol{Z}^n : \boldsymbol{z} = B^T\boldsymbol{s}, \boldsymbol{s} \in \boldsymbol{Z}^n\}$. На решетке равномерно распределен шум $\boldsymbol{e}$. Задан некоторый исходный вектор без шума $\boldsymbol{s} \in \boldsymbol{Z}_q^n$ и соответствующее значение $B\boldsymbol{s} + \boldsymbol{e}$. Найти исходную точку в решетке (исключить шум) по некоторому множеству известных $B\boldsymbol{s_i} + \boldsymbol{e}_i$.
5. Задача поиска вектора по норме (short integer solution, SIS). Для заданной матрицы $A \in \boldsymbol{Z}_q^{n\times m}$ найти вектор $\boldsymbol{v} \in \boldsymbol{Z}^m \backslash \{0\}$ такой, что $A\boldsymbol{v} = \boldsymbol{0}$ и $\|\boldsymbol{v}\| \le \beta$ для некоторого заданного β.

1.3 Описание схемы кольцевой подписи на решетках

Схема кольцевой подписи на решетках, основанная на модели со случайным оракулом, предложенная в 2014 году Ш. Ваном и Ж. Чжао в работе [7], была выбрана в данной работе в качестве прототипа ввиду того, что она характеризуется меньшим размером сформированной подписи в сравнении с другими рассмотренными схемами кольцевой подписи на решетках [5, 6]. Данная схема является адаптацией схемы подписи, предложенной В. Любашевским в 2012 году в работе «Lattice signatures without trapdoors» [3]. Схема является стойкой к атакам на основе выбранного открытого текста, ее безопасность основывается на задаче поиска вектора по норме (SIS). В процессе формирования подписи сообщения используются только операции перемножения матриц, что делает алгоритм достаточно эффективным.

Используемые процедуры:

1) TrapGen(1^n) – односторонняя функция с лазейкой. Пусть $q = poly(n)$ – простое число, $m > 5n\log q$ – случайное положительное число. На входе: параметр безопасности n. На выходе: матрицы $\boldsymbol{A} \in \boldsymbol{Z}_q^{n\times m}$ и $\boldsymbol{B_A} \in \boldsymbol{Z}_q^{n\times m}$. Здесь$\boldsymbol{B_A}$ – это базис решетки $\Lambda_{\boldsymbol{q}}(\boldsymbol{A}) = \{\boldsymbol{v} \in \boldsymbol{Z}_q^m : \boldsymbol{A}\boldsymbol{v} = 0(mod\ q)\}$.

2) SamplePr($\boldsymbol{A}, \boldsymbol{B_A}, S, \boldsymbol{y}$). На входе $\boldsymbol{A} \in \mathbf{Z}_q^{n\times m}$ и $\boldsymbol{B_A} \in \mathbf{Z}_q^{n\times m}$, $S \geq \|\boldsymbol{B_A}\|\omega(\sqrt{\log n})$, $\boldsymbol{y} \in \mathbf{Z}_q^n$. На выходе: вектор $\boldsymbol{e} \in \{\boldsymbol{e} \in \boldsymbol{Z}^m : \|\boldsymbol{e}\| \leq \sigma\sqrt{m}\}$ такой, что $\boldsymbol{Ae} = \boldsymbol{y}(mod\ q)$.

Основные алгоритмы протокола:

1. Ring-gen – вероятностный алгоритм, принимающий на вход параметр безопасности n и выдающий на выходе пару ключей (pk_i, sk_i) за полиномиальное время. Шаги алгоритма:

1) Пусть $q \geq 3$ – простое, $n \in \mathbf{Z} \gg 64, m \in \mathbf{Z} > 5\mathrm{n}\log q$.
   $H: \{0,1\}^* \to \{\boldsymbol{v}: \boldsymbol{v} \in \{-1,0,1\}^k, \|\boldsymbol{v}\| \leq k'\}$ – стойкая к коллизиям хэш-функция, $k, k' \in \mathbf{Z} > 0$. Матрица $\boldsymbol{T}$ выбирается случайно из $\mathbf{Z}_q^{n\times m}$.
2) В кольце R из l участников для каждого i-го участника запускается алгоритм TrapGen(1^n), чтобы получить $\boldsymbol{A}_i \in \mathbf{Z}_q^{n\times m}$ и $\boldsymbol{B}_i \in \mathbf{Z}_q^{n\times m}$.
3) Для каждого i-го участника запускается алгоритм SamplePr($\boldsymbol{A}_i, \boldsymbol{B}_i, S, \boldsymbol{t}_j$), где вектор $\boldsymbol{t}_j$ – j-й столбец матрицы $\boldsymbol{T}$. На выходе алгоритм выдает вектор $\boldsymbol{s}_{ij} \in \boldsymbol{Z}^m$ такой, что $\boldsymbol{A}_i\boldsymbol{S}_i = \boldsymbol{T}$ и $\boldsymbol{S}_i \in \{-d, \dots, 0, \dots, d\}^{m\times k}$, где $\boldsymbol{S}_i = (\boldsymbol{s}_{j1}, \dots, \boldsymbol{s}_{jk})$. $\boldsymbol{A}_i$ – открытый ключ, $\boldsymbol{S}_i$ – закрытый ключ i-го участника.

2. Ring-sign – вероятностный алгоритм, принимающий на вход набор параметров ρ, сообщение m, ключ автора подписи sk_i и набор открытых ключей $L = \{pk_1, \dots, pk_l\}$ и выдающий на выходе подпись z за полиномиальное время.

На входе алгоритма дано сообщение m, кольцо R из l участников с открытыми ключами $L = \{\boldsymbol{A}_1, \dots, \boldsymbol{A}_l\}$, закрытый ключ автора подписи $\boldsymbol{S}_j$. Шаги алгоритма:

1) Для каждого $i, 1 \leq i \leq l$ автор подписи генерирует случайный вектор $\boldsymbol{y}_i \in \boldsymbol{Z}^m$;
2) Вычисляет значение хеш-функции $\boldsymbol{c} = H(\sum_i \boldsymbol{A_i} y_i, L, m)$
3) Для всех $i, 1 \leq i \leq l$ вычисляет

$$\boldsymbol{z_i} = \begin{cases} \boldsymbol{y}_i, i \neq j \\ \boldsymbol{S}_j c + \boldsymbol{y_j}, i = j \end{cases}.$$

Тогда $(\boldsymbol{z_i}: 1 \leq i \leq l, \boldsymbol{c})$ – кольцевая подпись сообщения m.

3. Ring-verify – детерминированный алгоритм, принимающий на вход набор параметров ρ, подпись z сообщения m и определяющий, является ли подпись корректной.

На входе: сообщение m, кольцо R из l участников с открытыми ключами $L = \{\boldsymbol{A}_1, \dots, \boldsymbol{A}_l\}$ и подпись $(\boldsymbol{z_i}: 1 \leq i \leq l, \boldsymbol{c})$.

Проверяющий принимает подпись тогда и только тогда, когда выполняются два условия:

1) $\|\boldsymbol{z_i}\| \leq 2\sigma\sqrt{m}$ для всех $i, 1 \leq i \leq l$;
2) $\boldsymbol{c} = H(\sum_i \boldsymbol{A_i z}_i - \boldsymbol{Tc}, L, m)$.

1.4 Выводы

Таким образом, в данной главе рассмотрены основные свойства схем кольцевых подписей, к которым относятся невозможность подделки и анонимность автора подписи, описаны субъекты системы, отношения между ними и используемые алгоритмы. Обоснован выбор решеток в качестве математического аппарата кольцевой подписи, описаны вычислительно трудные задачи, основанные на решетках, и описан протокол выбранной кольцевой подписи на решетках.

В следующей главе для данной схемы кольцевой подписи на решетках будет выбран и применен механизм отзыва права подписи.

2. ОТЗЫВ В СХЕМЕ КОЛЬЦЕВОЙ ПОДПИСИ

Важным свойством всех групповых подписей является возможность удаления участника из группы и отзыв у него права подписи. В существующих схемах применяются различные методы для организации отзыва. Так, у самого участника может быть отозвана возможность создания корректной подписи либо наоборот, проверяющий при проверке подписи производит дополнительные вычисления, чтобы убедиться, что ее автор не был удален из группы. При этом важным является влияние на эффективность схемы подписи.

2.1 Сравнение механизмов отзыва

Разные механизмы отзыва отличаются объемом дополнительных вычислений и обновлений, которые должны сделать как участники группы, так и проверяющие, а также влиянием на размер подписи. Рассмотрим существующие механизмы отзыва.

– Базовый подход. Наиболее простым методом является отзыв, при котором все участники, которые не должны быть удалены из группы, получают заново сгенерированные ключи (reissuance-based revocation, RBR) [8, 17]. Такой подход требует больших вычислительных затрат во время проведения процедуры отзыва права подписи, поэтому является неудобным в случае, если участников необходимо удалять часто.

– Черный список. Этот подход может основываться на списках отозванных сертификатов (Certificate-based blacklist revocation, BR-C) [9]. От автора подписи требуется предоставить доказательство с нулевым разглашением того, что он не состоит в списке отзыва. При этом возрастают вычислительные затраты для автора подписи и проверяющего при каждом формировании и проверке подписи, а также возрастает размер подписи. Аналогичный подход, основанный на использовании списков ключей подписи [14] приводит к линейному росту вычислительных затрат и размера подписи в зависимости от количества участников. Для организации черного списка можно использовать отзыв с ди-

намическим накоплением (Accumulator-based blacklist revocation, BR-A) [11, 19], когда для предоставления списка отзыва используются криптографические «аккумуляторы».

– Отзыв, локальный для проверяющего (verifier-local revocation, VLR) [10, 20, 21]. В данном методе записи отзыва не основываются на используемых ключах и обрабатываются только проверяющим, поэтому требуется, чтобы проверяющие обновляли список отзыва при каждом удалении участников. Проверка подписи значительно влияет на объем вычислений, производимых проверяющим. Другим недостатком данной схемы является то, что по подписям отозванных участников группы каждый проверяющий может определить, были они сформированы одним и тем же автором или нет, что нарушает одно из ключевых свойств схемы кольцевой подписи, заключающееся в анонимности автора подписи в пределах группы.

– Отзыв со связыванием (linking-based revocation, LBR) [12]. Данный подход основывается на том, что назначенная сущность обладает возможностью определить, были ли две подписи сформированы одним и тем же (анонимным) автором – участником группы. Данная сущность является онлайн-сервером, который хранит подписи отозванных участников группы. При проверке подписи проверяющий обращается к серверу. Сервер сравнивает полученную подпись со списком отозванных подписей, после чего отвечает проверяющему, состоит ли автор подписи в списке отзыва. Данный метод не требует от участников группы обновления ключей при отзыве одного из участников, а влияние на время формирования, проверки и размер подписи не зависит от размера группы.

Сравнение свойств данных механизмов, влияющих на эффективность схемы подписи, представлено в табл. 2.1. Буквой R обозначено число участников группы, прочерком обозначаются случаи, когда механизм отзыва не оказывает влияния на время работы указанных алгоритмов, на размер ключей и подписи или не требует обновлений. Как видно из таблицы, механизм с перевыпуском ключей требует значительных вычислительных затрат для генерации

новых ключей при отзыве права подписи у одного из участников, дополнительные вычислительные затраты линейно зависят от числа участников. Механизм с черными списками требует значительных дополнительных вычислений и на этапе формирования подписи, и на этапе проверки для доказательства того, что автор подписи не состоит в списке отзыва. Отзыв, локальный для проверяющего, не оказывает влияния на размер подписи, но требует частых обновлений от проверяющего для синхронизации списков отзыва между участниками. Механизм отзыва со связыванием также влияет на размер подписи и время ее формирования и проверки, но это влияние не растет с увеличением размера группы участников.

Таблица 2.1 – Сравнение механизмов отзыва

Тип	Память		Время		Обновления	
	Размер ключей	Размер подписи	Автор подписи	Проверяющий	Автор подписи	Проверяющий
RBR	-	-	-	-	O(R)	O(1)
BR-C	-	O(R)	O(R)	O(R)	O(R)	O(R)
BR-A	-	O(1)	O(1)	O(1)	O(1)	O(1)
VLR	-	-	-	O(R)	-	O(R)
LBR	O(1)	O(1)	O(1)	O(1)	-	-

Важным показателем эффективности подписи является необходимость обновления данных для участников группы и для проверяющих. Частые обновления для пользователей, право подписи которых не было отозвано, зачастую сложно реализовать на практике, так как отзыв должен происходить незамедлительно, следовательно, обновления не могут осуществляться с определенной периодичностью, необходима постоянная синхронизация данных между участниками группы и проверяющими. Схема отзыва со связыванием решает эту проблему за счет того, что проверяющий должен постоянно иметь доступ к онлайн-серверу.

2.2 Отзыв со связыванием

Так как механизм отзыва со связыванием, рассмотренный в разделе 2.1, позволяет производить отзыв права подписи без дополнительных вычислений для участников группы, а влияние выбранного механизма на размер подписи и на объем вычислений, производимых при формировании и проверке подписи, не зависит от числа участников группы, данный метод был выбран в качестве механизма отзыва для схемы кольцевой подписи на решетках. В данном разделе более подробно описываются свойства рассматриваемого метода и способы его реализации для применения к схеме кольцевой подписи на решетках.

2.2.1 Свойство контролируемой связываемости

Кольцевые подписи с контролируемой связываемостью отличаются наличием выделенной сущности – менеджера связывания, обладающего возможностью определить, были ли две подписи сформированы одним и тем же пользователем, без возможности идентифицировать этого пользователя. Ключ менеджера связывания (master linking key) обозначим как mlk. Такая схема содержит следующие алгоритмы [22]:

– GkGen(1^{λ}): На вход принимает параметр безопасности λ, на выходе генерирует открытые параметры схемы gpk и ключ менеджера связывания mlk.

– UkGen(1^{λ}): На вход принимает параметр безопасности λ, на выходе генерирует пары ключей пользователей (pk_i, sk_i).

– Sign(gpk, M, sk_i): На вход принимает открытые параметры системы gpk, сообщение M и секретный ключ пользователя sk_i, на выходе генерирует подпись σ.

– Verify(gpk, M, σ): На вход принимает открытые параметры системы gpk, сообщение M и подпись σ, на выходе возвращает true в случае, если подпись корректна, и false – иначе.

– Link($gpk, mlk, M, \sigma, M', \sigma'$): На вход принимает открытые параметры системы gpk, ключ менеджера связывания mlk, сообщения M и M' и соот-

ветствующие им подписи σ и σ'. На выходе алгоритм возвращает true в случае, если обе подписи были сформированы одним и тем же пользователем, и false – иначе.

К таким свойствам схемы подписи, как невозможность подделки и анонимность автора подписи, добавляется следующее свойство безопасности: ключ менеджера связывания не должен позволять получить какую-либо информацию, с помощью которой можно идентифицировать автора подписи.

Отзыв в схеме кольцевой подписи, основанный на идее контролируемой связываемости, организуется следующим образом. Проверяющий, в первую очередь, проверяет корректность подписи с помощью алгоритма Verify(), затем обращается к центру отзыва. Центр отзыва, владея ключом менеджера связывания mlk и списком отзыва, который в нашем случае представляет собой список подписей, сформированных пользователями, чье право подписи было отозвано, проверяет подпись, сравнивая ее с каждой записью в списке с помощью алгоритма Link(). Если какая-либо из подписей в списке была сформирована данным пользователем, значит, он был отозван. Затем центр отзыва отправляет проверяющему ответ, был автор подписи отозван или нет. Данный подход представлен на рис. 2.1.

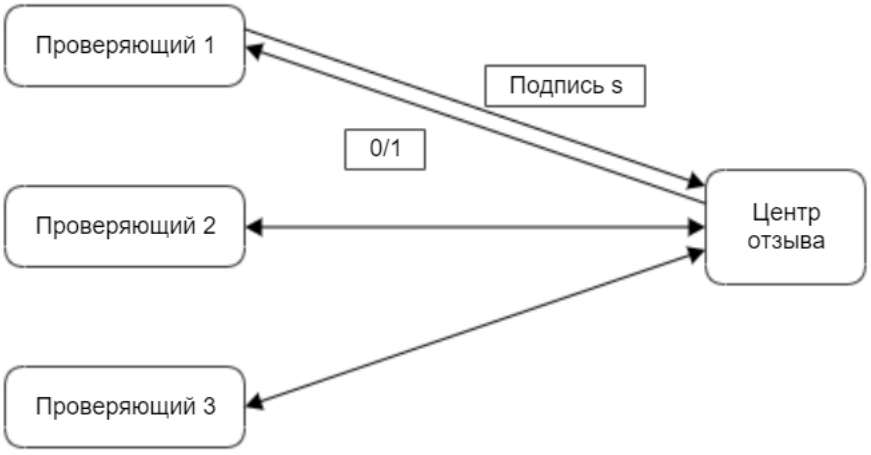

Рисунок 2.1 – Отзыв со связыванием

Кольцевая подпись с реализованным свойством связываемости должна состоять из следующих блоков:

1) схема подписи DS=(KeyGen$_s$, Sign, Verify);
2) схема шифрования с открытым ключом AE=(KeyGen$_e$, Enc, Dec).

2.2.2 Алгоритм шифрования с открытым ключом с тестами на равенство

Открытый ключ gpk (group public key) содержит в себе открытый ключ шифрования epk и открытый ключ, необходимый для проверки подписи, spk. Каждый новый участник кольца отправляет выпускающему центру значение $f(x_i)$, где f – это односторонняя функция, примененная к секретному значению x_i. Выпускающий центр в ответ отправляет подпись Sign$(sk, f(x_i))$ в качестве сертификата участника $cert$. Тогда кольцевая подпись, сформированная участником при подписании сообщения M, будет дополнена элементом T = Enc$(epk, cert)$.

Для доказательства связанности двух подписей используется шифрование с открытым ключом с тестами на равенство (All-or-Nothing Public Key Encryption with Equality Tests, AoN-PKEET) [23], позволяющее назначенной сущности, владеющей лазейкой, по двум шифртекстам определить, зашифровывают ли они один и тот же открытый текст, при этом не раскрывая его содержания.

Протокол шифрования с открытым ключом с тестами на равенство состоит из следующих алгоритмов:

1. PKEET-gen(1^λ): Алгоритм генерации ключей шифрования, принимающий на вход параметр безопасности λ и генерирующий на выходе пару ключей (epk,esk).

   Пусть G – мультипликативная группа простого порядка p, g – ее образующая, H_1, H_2, H_3 – хеш-функции: $H_1: \{0,1\}^* \to \{0,1\}^{m+d}$, $H_2: \{0,1\}^* \to \boldsymbol{Z}_p, H_3: \{0,1\}^* \to \{0,1\}^k$. Алгоритм вычисляет закрытый ключ $esk = (x, y)$, где $x, y \in \boldsymbol{Z}_p$ выбираются случайно, и соответствующий открытый ключ $epk = (g^x, g^y)$.

2. PKEET-enc(epk, m): Алгоритм зашифрования, принимающий на вход открытый ключ epk и сообщение m и возвращающий соответствующий шифртекст c.

   Шифртекст $C = (C^{(1)}, C^{(2)}, C^{(3)}, C^{(4)}, C^{(5)})$ вычисляется следующим образом: $u, v \in \boldsymbol{Z}_p$ выбираются случайно,

   $C^{(1)} = g^u$,

   $C^{(2)} = g^v$,

   $C^{(3)} = H_1(g^{ux}) \oplus m || u$,

   $C^{(4)} = g^{H_2(g^{vy})}$,

   $C^{(5)} = H_3(C^{(1)} || C^{(2)} || C^{(3)} || C^{(4)} || m || u)$.

3. PKEET-dec(esk, c): Алгоритм расшифрования, принимающий закрытый ключ esk и шифртекст c и возвращающий соответствующее сообщение m. Шаги:

   1) Вычисляется $m' || u' = C^{(3)} \oplus H_1((C^{(1)})^x$.
   2) Проверяются условия:
      a. $C^{(1)} = g^{u'}$
      b. $C^{(5)} = H_3(C^{(1)} || C^{(2)} || C^{(3)} || C^{(4)} || m' || u')$

   В случае их выполнения m' – расшифрованное сообщение.

4. PKEET-aut(esk): Принимает на вход закрытый ключ esk и возвращает лазейку tk, используемую для тестов на равенство.

   Для пользователя, предъявившего закрытый ключ *esk*, возвращает токен $T_i = y_i$.

5. PKEET-com(C_i, C_j, T_i, T_j): Принимает на вход два шифртекста C_i, C_j и токены T_i и T_j и возвращает true, в случае если они зашифровывают одно и то же нераскрываемое сообщение, и false – иначе.

   Алгоритм возвращает 1 в случае, если $C_i^{(4)} g^{-H_2((C_i^{(2)})^{T_i}} = C_j^{(4)} g^{-H_2((C_j^{(2)})^{T_j}}$, и 0 иначе.

2.3 Применение отзыва со связыванием к схеме кольцевой подписи

Для организации механизма отзыва представленная в разделе 1.3 схема кольцевой подписи должна быть дополнена механизмом обеспечения свойства связности. Для реализации этого свойства каждому участнику кольца выдается сертификат $cert$. Подпись сообщения дополняется элементом T, представляющим собой данный сертификат, зашифрованный на ключе epk. Затем при проверке сообщения проверяющий сначала проверяет корректность подписи, а затем обращается к центру отзыва, который, владея лазейкой tk, сравнивает зашифрованный сертификат из подписи с зашифрованными сертификатами из списка отзыва с помощью теста на равенство и отправляет проверяющему ответ, содержится ли данный участник в списке отзыва.

Однако в данном методе существует следующая проблема безопасности: автор подписи (нарушитель) может зашифровать и отправить не свой сертификат и таким образом пройти проверку при сравнении подписи со списком отзыва. Для решения данной проблемы предлагается воспользоваться одним из следующих подходов.

Первый подход заключается в том, что центр отзыва вместо списка отозванных подписей будет хранить список подписей активных участников группы (белый список). Тогда при проверке подписи сертификат автора должен быть связан с одним из сертификатов из списка. В случае, если право подписи пользователя было отозвано, его подпись удаляется из списка.

Преимуществом данного подхода является то, что он не требует дополнительных вычислений от автора подписи или проверяющего, основной нагрузкой является сравнение зашифрованных сертификатов, осуществляемое сервером. Однако для многих систем такой подход не является эффективным, так как количество активных участников зачастую в разы больше, чем количество участников, чье право подписи было отозвано. Таким образом, сервер будет вынужден сравнивать проверяемый сертификат с большим числом сертификатов, состоящих в белом списке.

В подобных системах возможно применение другого подхода. Для того, чтобы реализовать схему проверки с помощью списка отозванных подписей, необходимо дополнить подпись неинтерактивным доказательством с нулевым разглашением. Доказательство с нулевым разглашением позволяет одному участнику (доказывающей стороне) доказать другому участнику (проверяющей стороне) истинность утверждения, не раскрывая сущности доказательства. Неинтерактивное доказательство позволяет осуществить эту процедуру без взаимодействия сторон. Для этого может быть использован эвристический метод Фиата–Шамира, обозначаемый SoK (Signatures of knowledge) [24].

Тогда подпись участника должна быть дополнена доказательством π:

$$\pi \leftarrow SoK\{(x_i, cert): cert = \mathrm{Sign}(sk, f(x_i)) \wedge T = \mathrm{Enc}(epk, cert)\}.$$

Этот подход, в отличие от предыдущего, требует от автора подписи и проверяющего дополнительных вычислений для создания и проверки доказательства, однако позволяет эффективнее производить проверку центром отзыва за счет меньшего размера черного списка сертификатов в сравнении с белым списком.

Можно сделать вывод, что в зависимости от параметров реализации системы и ее свойств может быть выбран один из предложенных методов. Например, для систем с конечными устройствами пользователей, обладающими малой вычислительной мощностью, нецелесообразно возлагать на эти устройства дополнительные вычислительные затраты, эффективнее будет воспользоваться подходом, основанным на белом списке.

Таким образом, протокол кольцевой подписи на решетках с механизмом отзыва со связыванием состоит из следующих алгоритмов:

1) Алгоритм Gen.
    a. Для каждого участника группы генерируются ключи подписи (spk, ssk) с помощью алгоритма Ring-gen схемы кольцевой подписи.

b. Для каждого участника генерируются ключи шифрования (epk, esk) с помощью алгоритма PKEET-gen шифрования с открытым ключом с тестами на равенство.
c. Каждому участнику выдается сертификат $cert$.
d. Центру отзыва выдается ключ связывания tk, который является лазейкой сгенерированной алгоритмом PKEET-aut(esk).
e. В случае реализации схемы с белым списком формируется белый список, состоящий из сертификатов участников группы $\{cert_1, \dots, cert_l\}$, зашифрованных с помощью алгоритма PKEET-enc. В случае реализации схемы с черным списком, на данном этапе он остается пустым.

2) Алгоритм Sign.
   a. Для подписания сообщения m участник i сначала формирует подпись на ключе подписи ssk_i и открытых ключах участников группы $L = \{spk_1, \dots, spk_l\}$ с помощью алгоритма Ring-sign: $S = \text{Ring-sign}(m, ssk_i, L)$.
   b. Затем автор подписи зашифровывает свой сертификат $cert_i$ на открытом ключе шифрования epk_i с помощью алгоритма PKEET-enc: $T = \text{PKEET-enc}(epk, cert)$.

      Итоговая подпись представляет собой пару: (S, T).

3) Алгоритм Verify.
   a. Получатель проверяет подпись S сообщения m с помощью алгоритма проверки Ring-verify. Если алгоритм вернул значение 1, значит подпись корректна и в пункте b проверяется, не был ли автор подписи исключен из группы. Если алгоритм вернул 0, то подпись признается неверной и проверка завершается.
   b. После проверки корректности подписи проверяющий обращается к центру отзыва. Центр отзыва проверяет, не был ли автор исключен из списка участников.

Обладая лазейкой tk, центр отзыва сравнивает зашифрованный сертификат T со всеми зашифрованными сертификатами в списке отзыва с помощью теста на равенство PKEET-Com(T, T', tk). В случае реализации с белым списком, если сертификат оказался связанным с одним из сертификатов в списке (алгоритм PKEET-Com вернул 1), то центр отзыва возвращает проверяющему значение 1, если же ни одна подпись из списка не соответствует проверяемой – 0.

В случае черного списка наоборот: если алгоритм PKEET-Com признает, что сертификат связан хотя бы с одним сертификатом из списка, значит, пользователь был исключен из группы.

4) Алгоритм Revoke.
   a. В случае белого списка для отзыва права подписи у участника необходимо исключить его зашифрованный сертификат из списка отзыва. Поиск необходимого сертификата происходит путем сравнения сертификат T участника со всеми сертификатами из белого списка с помощью теста на равенства PKEET-Com(T, T', tk).
   b. В случае черного списка зашифрованный сертификат T просто добавляется в список отзыва.

На рис. 2.2 представлена схема взаимодействия субъектов описанной кольцевой подписи: участников группы, проверяющих и центра отзыва, указаны открытые и закрытые ключи подписи и шифрования, которыми они владеют, и алгоритмы формирования и проверки подписи, которыми они оперируют, а также данные, которыми они друг с другом обмениваются.

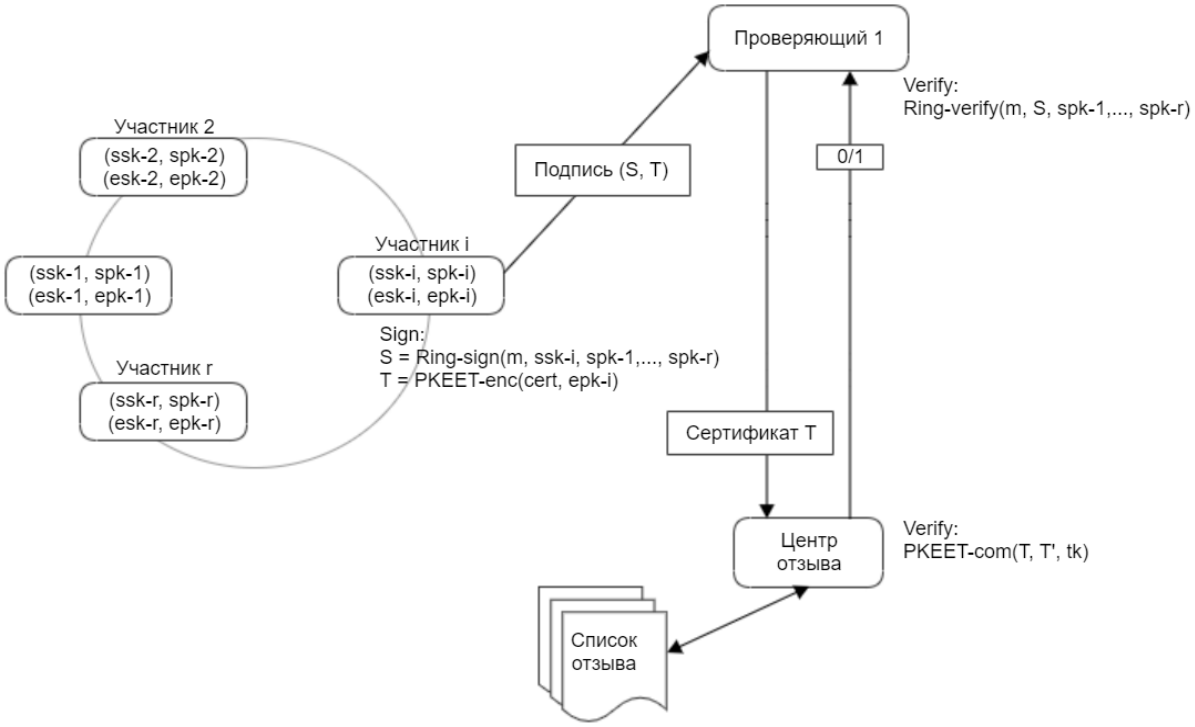

Рисунок 2.2 – Схема взаимодействия участников в разработанной схеме

2.4 Повышение безопасности и эффективности механизма отзыва

Так как данный механизм отзыва основывается на участии в процессе проверки подписи онлайн-сервера, он может быть подвержен атакам по сети со стороны злоумышленников. В случае компрометации центра отзыва злоумышленник получит возможность сравнивать любые две подписи и определять, были ли они сформированы одним и тем же пользователем.

Для обеспечения безопасности в случае компрометации центра отзыва предлагается применить метод разделения ключа связывания между несколькими центрами связывания. (t, n)-пороговая схема разделения секрета, предложенная А. Шамиром в работе [25], позволяет распределить секрет s между n участниками таким образом, чтобы для восстановления секрета s необходимо было взаимодействие как минимум t участников.

Тогда для сравнения подписей центр отзыва должен будет связаться минимум с t центрами связывания, чтобы воспользоваться ключом связывания tk. Для реализации данной идеи предлагается использовать пороговое шифрование с открытым ключом с тестами на равенство (Threshold AoN-PKEET), что позво-

лит достичь снижения требуемого уровня доверия к центру отзыва и повысить надежность схемы.

(t, n)-пороговая схема А. Шамира позволяет разделить некоторый секрет D (в нашем случае – ключ связывания) на n долей $D_1, \dots, D_n$ таким образом, чтобы выполнялись следующие свойства:

- знания любого набора из t или более долей D_i достаточно для вычисления D;
- знания любого набора из $(t-1)$ или менее долей D_i не позволяет раскрыть никакую информацию о значении D.

Данная (t, n)-пороговая схема разделения ключа основывается на задаче интерполяции полиномов. Для заданных t точек $(x_1, y_n), \dots, (x_t, y_t)$ существует ровно один полином $q(x)$ степени $t-1$ такой, что $q(x_i) = y_i$ для $1 \leq i \leq t$.

Для разделения секрета D на доли D_i, выбирается случайный полином степени $t-1$: $q(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1}$, в котором $a_0 = D$ и вычисляются: $D_1 = q(1), \dots, D_i = q(i), \dots, D_n = q(n)$.

При наличии любого набора из t значений D_i можно найти коэффициенты $q(x)$ с помощью метода интерполяции, после чего оценить $D = q(0)$. При этом $t-1$ значений недостаточно для вычисления D.

Предлагаемая к использованию пороговая схема T-AoN-PKEET отличается от классической схемы AoN-PKEET следующими алгоритмами:

- DKAut(tk, t, n): принимает на вход лазейку tk, порог t и общее число разделения n и возвращает разделение $(tk_i), 1 < i < n$ такое, что для восстановления tk требуется взаимодействие минимум t сущностей.
- TShare(T, T', tk_i): принимает на вход два шифртекста T и , T' и долю лазейки tk_i и возвращает соответствующие доли C и C' для проведения теста на равенство.
- TSCom$(\{C_i, C'_i\}_{t \leq i \leq n})$: принимает на вход наборы долей $\{C_i, C'_i\}_{t \leq i \leq n}$двух шифртекстов и возвращает true, в случае если они

зашифровывают одно и то же нераскрываемое сообщение, и false – иначе.

Тогда проверка подписи центром отзыва с учетом применения механизма разделения лазейки и хранения токенов в списке отзыва проводится в соответствии со следующим алгоритмом.

CheckStatus(RL, L, σ): Принимая на вход список отзыва RL, набор центров связывания L и подпись σ, алгоритм определяет статус участника, являющегося автором подписи. Для этого он связывается с t центрами связывания $L_i \in L$ с помощью алгоритма TShare, получая от каждого соответствующие доли C_i токена проверяемого участника, после чего применяет к данному набору долей алгоритм TSCom, чтобы восстановить итоговый токен участника. Если данный токен состоит в списке отзыва, алгоритм возвращает true, иначе – false. Взаимодействие участников схемы при проверке подписи представлено на рис. 2.3.

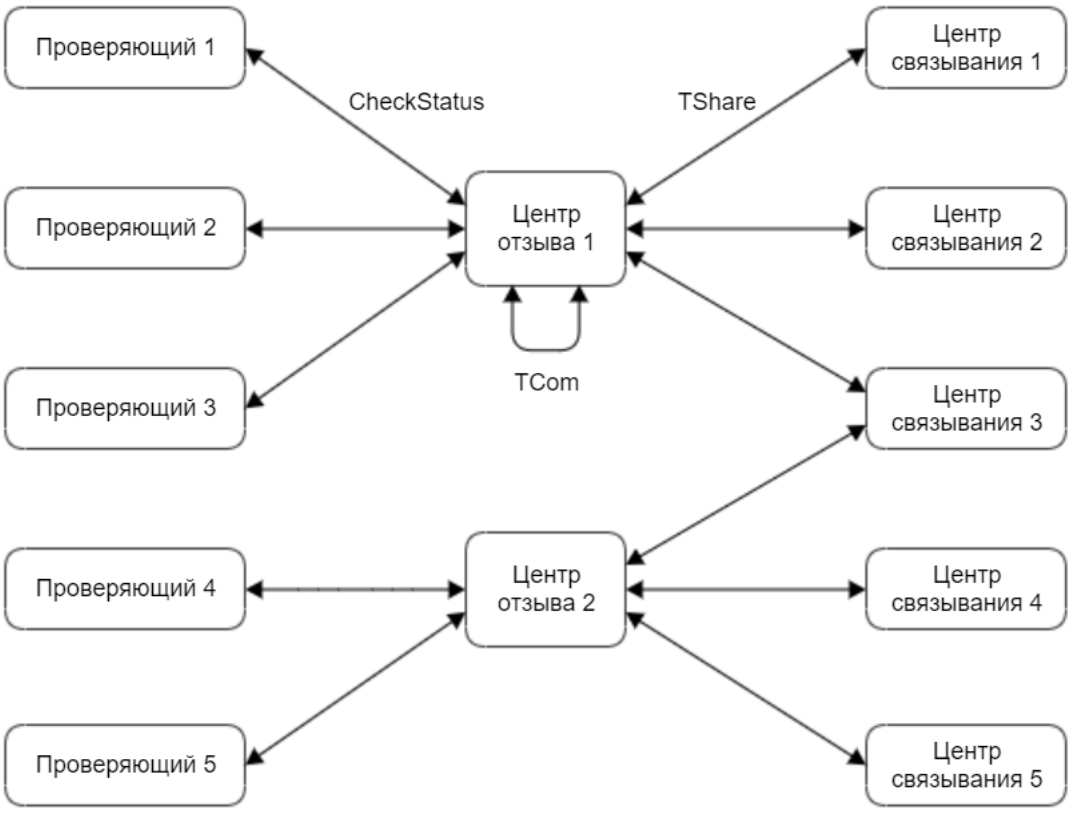

Рисунок 2.3 – Схема с разделением ключа связывания

Такой метод разделения ключа связывания между несколькими центрами связывания позволяет требовать меньшего уровня доверия к центру отзыва, чем в случае схемы с единым сервером.

Другим недостатком системы является то, что время проверок, производимых центром отзыва, линейно зависит от размера списка отзыва, то есть имеет сложность O(R), где R – число пользователей в списке.

Чтобы время проверки не зависело от размера списка, процедура сравнения двух подписей может быть заменена процедурой формирования токена. То есть алгоритм Com(T, T', tk) модифицируется таким образом, чтобы вместо проведения теста на равенство для двух шифртекстов он возвращал значение, вычисляемое из заданного шифртекста T и лазейки tk, то есть токен участника t←Com(T, ⊥, tk).

Тогда в списке отзыва вместо подписей будут храниться такие токены в виде хеш-таблицы, что при проверке за время O(1) можно будет определить, принадлежит ли проверяемый токен списку. При этом сам токен не раскрывает никакой информации об авторе подписи. При отзыве права подписи участник остается анонимным, так как для вычисления токена требуется только его подпись и ключ связывания.

2.5 Выводы

В данной главе рассмотрены варианты реализации отзыва права подписи в кольцевой схеме, произведено сравнение механизмов отзыва и выбран предпочтительный – механизм отзыва со связыванием. Предложено два варианта применения выбранного метода: с белым и черным списками пользователей. В результате разработана схема кольцевой подписи на решетках с отзывом со связыванием, использующая шифрование с открытым ключом с тестами на равенство для реализации свойства связываемости.

В следующей главе будет приведена программная реализация и оценка разработанной схемы.

3. РЕАЛИЗАЦИЯ СХЕМЫ КОЛЬЦЕВОЙ ПОДПИСИ НА РЕШЕТКАХ С ОТЗЫВОМ СО СВЯЗЫВАНИЕМ

Для оценки влияния механизма отзыва на эффективность схемы было решено осуществить программную реализацию разработанной схемы подписи с отзывом со связыванием, а также схемы, использующей базовый подход, – перевыпуск ключей пользователей при отзыве права подписи у одного из участников. Необходимо протестировать и замерить время работы всех реализованных алгоритмов схемы подписи.

3.1 Программная реализация схемы

В разрабатываемой системе можно выделить два основных компонента:

- протокол кольцевой подписи на решетках;
- протокол шифрования с открытым ключом с тестами на равенство.

В связи с тем, что реализуемые алгоритмы предполагают работу с такими математическими структурами, как решетки, было решено использовать для разработки программы продукт, содержащий встроенные модули для удобной работы с ними. В качестве такого продукта был выбран Sage [26] – математическое программное обеспечение с открытыми исходными кодами для исследовательской работы в таких областях, как алгебра, геометрия, теория чисел, криптография и других. Sage позволяет создавать скрипты на языке Python, которые затем обрабатываются на локальном сервере Sage.

В ходе выполнения практической части работы была написана программа на языке Python [27], реализующая алгоритмы протокола кольцевой подписи на решетках с отзывом со связыванием. В приложении приведен исходный код программы с комментариями.

В качестве хеш-функции, используемой при генерации и проверки подписи, а также при зашифровании и расшифровании сертификата, была использована хеш-функция SHA с длиной хеш-значения 512 бит, определенная в модуле hashlib.

Для получения случайной матрицы при генерации ключей схемы подписи на решетках была использована функция Sage random_matrix(), которая позволяет задавать размерность матрицы и множество значений ее элементов. Для задания множества была использована функция Zmod(q), с помощью которой можно задать множество классов вычетов по модулю q.

При генерации подписи для выборки векторов y в соответствии с дискретным нормальным распределением был использован модуль sage.stats.distributions.discrete_gaussian_lattice, в котором определена функция DiscreteGaussianDistributionLatticeSampler, позволяющая задать множество значений выборки (в данной работе – $\mathbf{Z}_m$) и среднеквадратичное отклонение.

Для генерации ключей и сертификатов пользователей использовалась функция random.randint, определенная в модуле random и позволяющая задавать диапазон генерируемого целого числа.

Для вычисления мультипликативно обратного элемента в алгоритме сравнения двух зашифрованных сообщений была использована функция Sage inverse_mod.

В ходе выполнения работы были разработаны две системы. Первая – описанная в разделе 1.3 схема подписи на решетках, которая была дополнена алгоритмом отзыва права подписи, осуществляемого путем повторной генерации (перевыпуска) открытых и закрытых ключей участников группы за исключением удаленного. Вторая система является реализацией описанной в разделе 2.3 схема подписи на решетках с отзывом со связыванием, в качестве списка отзыва в которой используется белый список участников группы.

Структура разработанной программы соответствует описанной структуре схемы подписи. Алгоритмы генерации ключей, формирования и проверки подписи и отзыва права подписи реализованы в функциях setup(), sign(), verify(), revoke(). Каждая из этих функций вызывает функции, реализующие соответствующие алгоритмы схемы подписи и шифрования.

В функции setup() генерируются открытые (RS_vk_T, RS_vk_A) и закрытые (RS_sk) ключи подписи для каждого участника с помощью функции

RS_key_gen(). С помощью функции PKEET_key_gen происходит генерация открытых (PKEET_p) и закрытых (PKEET_sk) ключей шифрования пользователей. Затем генерируются сертификаты пользователей (CERT) и с помощью функции PKEET_enc происходит их зашифрование. Зашифрованные сертификаты пользователей составляют белый список группы (WL), в котором также указывается, является ли пользователь активным участником.

Функция sign() для сообщения msg вызывает функцию формирования кольцевой подписи RS_sign, передавая ей открытые ключи всех участников группы (RS_vk_A) и закрытый ключ автора подписи (RS_sk[j]). Затем вызывается функция PKEET_enc, зашифровывающая сертификат автора подписи (CERT[j]) на его открытом ключе шифрования (PKEET_pk[j]). Полученная подпись (z, c) и зашифрованный сертификат (c2, c4) объединяются в кортеж и возвращаются в качестве результата функции.

В функция verify() происходит проверка корректности подписи (z, c) сообщения msg с помощью функции RS_verify, принимающей на вход также открытые ключи участников группы. Затем проверяется, не исключен ли автор подписи из белого списка участников. Для этого зашифрованный сертификат (c2, c4) передается функции PKEET_com, которая сравнивает его с каждым сертификатом из списка. Если сертификат окажется связан с одним из сертификатов в списке, автор подписи признается активным. Функция возвращает 1 в случае, если оба этапа проверки дают положительный результат.

Функция revoke() осуществляет поиск зашифрованного сертификата пользователя, которого необходимо исключить, в белом списке с помощью функции PKEET_com, проверяющей, связаны ли два сертификата, после чего удаляет его из списка.

Для измерения времени исполнения программы была использована функция time.time из модуля time, возвращающая время в секундах, прошедшее с начала эпохи Unix.

3.2 Тестирование и оценка разработанной системы

Для проведения тестирования разработанной системы были использованы параметры в соответствии с рекомендациями в работе Любашевского [3], на которой основана реализуемая в системе схема кольцевой подписи:

- модуль $q \geq 3$ – простое число;
- размерность матрицы: $n \in \mathbf{Z} \geq 64, m \approx 64 + n \log q \,/\, \log 3$;
- параметры нормального распределения: среднеквадратичное отклонение $\sigma \approx 36\, k \sqrt{m}$.

Тестирование разработанной программы проводилось для n = 64, q = 61, m = 239, k = 80.

Параметры p (порядок мультипликативной группы G) и g (образующая группы G) были сгенерированы в функции PKEET_param_gen(). Для поиска образующей группы был использован следующий критерий:

Элемент a является образующей группы тогда и только тогда, когда $a^{(p-1)/q} \not\equiv 1 (mod\ p)$ для всех простых q – делителей $p - 1$.

Таким образом были сгенерированы параметры:

p = 13419005495551248730641302049631477087692535813 01,

g = 14922578501688351340704660452576883153761508920129784878538191291561205808667455064901451426550236353695837369547299448900196564136048670818751273704430 86.

Количество участников кольца изменялось от 2 до 2000. Для тестирования производился последовательный запуск скриптов реализующих генерацию ключей, подписывание сообщения, проверки подписи и отзыва права подписи для базовой схемы с перевыпуском ключей, и для разработанной схемы с отзывом со связыванием.

В табл. 3.1–3.4 представлены результаты замеров времени работы перечисленных алгоритмов в зависимости от числа участников. Для каждого значения числа участников было произведено по пять тестовых запусков, в таблицы были внесены средние значения.

Таблица 3.1 – Сравнение времени генерации ключей

Число участников группы	Время генерации ключей в схеме с перевыпуском, с	Время генерации ключей в разработанной схеме
2	0,129	0,129
10	0,663	0,692
50	3,265	3,461
100	6,676	6,970
500	37,085	39,705
1000	74,302	77,008
1500	108,410	111,089
2000	168,285	173,511

Таблица 3.2 – Сравнение времени формирования подписи

Число участников группы	Время формирования подписи в схеме с перевыпуском, с	Время формирования подписи в разработанной схеме, с
2	1,038	1,009
10	1,909	1,886
50	3,554	3,613
100	5,428	5,684
500	22,265	23,736
1000	37,704	38,737
1500	59,769	61,540
2000	77,252	79,894

Таблица 3.3 – Сравнение времени проверки подписи

Число участников группы	Время проверки подписи в схеме с перевыпуском, с	Время проверки подписи в разработанной схеме, с	
		со стороны проверяющего	со стороны центра отзыва
2	0,007	0,007	0,001
10	0,024	0,022	0,004
50	0,089	0,093	0,019
100	0,187	0,179	0,035
500	0,971	0,986	0,173
1000	2,041	2,182	0,443
1500	2,763	2,833	0,537
2000	3,860	3,893	0,742

Таблица 3.4 – Сравнение времени отзыва права подписи

Число участников группы	Время отзыва права подписи в схеме с перевыпуском, с	Время отзыва права подписи в разработанной схеме, с
2	0,129	0,001
10	0,680	0,004
50	3,208	0,019
100	7,811	0,037
500	36,419	0,188
1000	75,296	0,380
1500	129,579	0,542
2000	161,415	0,883

На рис. 3.1–3.4 представлены графики зависимости времени работы алгоритмов от числа участников группы, построенные по соответствующим значениям из таблиц.

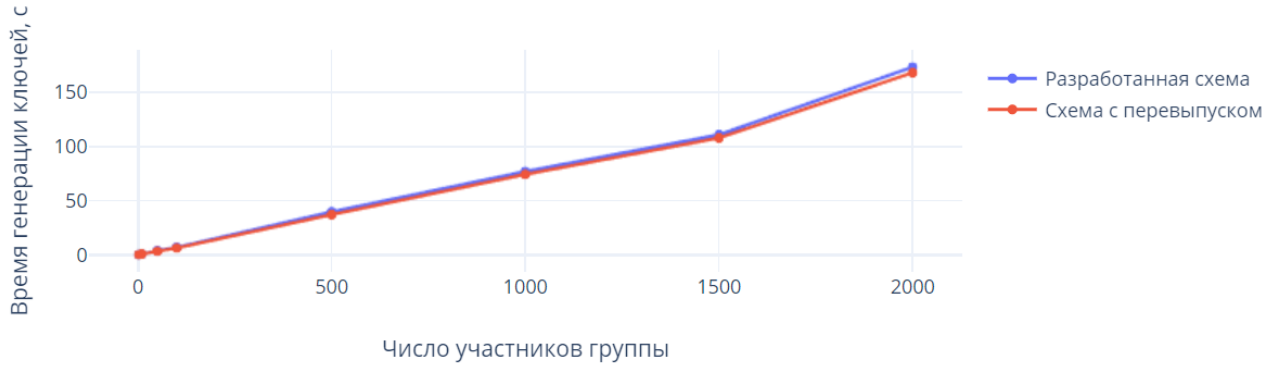

Рисунок 3.1 – Графики зависимости времени генерации ключей от числа участников

Рисунок 3.2 – Графики зависимости времени формирования подписи от числа участников

Рисунок 3.3 – Графики зависимости времени проверки подписи от числа участников

Рисунок 3.4 – Графики зависимости времени отзыва права подписи от числа участников

Полученные экспериментальные данные подтверждают теоретические оценки влияния механизма отзыва со связыванием на эффективность схемы кольцевой подписи.

Наибольшее влияние проявляется на этапе генерации ключей (рис 3.1), так как время выполнения этого алгоритма зависит от размера группы: для каждого из них выпускаются ключи шифрования и сертификаты. Влияние, оказываемое на формирование подписи, является незначительным, так как на этом этапе к базовому алгоритму добавляется только зашифрование сертификата, размер которого (а соответственно и время зашифрования) не зависит от размера группы.

При проверке подписи со стороны проверяющего не требуется никаких дополнительных вычислений, следовательно, время проверки не изменилось в

сравнении с базовой подписью, но добавились вычисления со стороны центра отзыва, заключающиеся в осуществлении тестов на равенство. Наибольшая разница в эффективности двух схем проявляется в алгоритме отзыва права подписи. Перевыпуск ключей требует значительно больших вычислительных затрат, чем удаление пользователя из списка отзыва.

3.3 Выводы

Таким образом, в данной главе осуществлена программная реализация разработанной в выпускной квалификационной работе схемы кольцевой подписи на решетках с отзывом со связыванием. Произведена оценка влияния выбранного механизма отзыва на время выполнения алгоритмов генерации ключей, формирования и проверки подписи и отзыва права подписи. Полученные экспериментальные результаты соответствуют теоретическим оценкам, описанным в разделе 2.1.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы решена задача организации отзыва права подписи в схеме кольцевой подписи на решетках.

В рамках работы проведен анализ свойств кольцевых подписей, выбрана базовая схема кольцевой подписи на решетках, не имеющая механизма отзыва права подписи у участников. Осуществлен обзор существующих методов организации отзыва в групповых подписях, их сравнение по критериям влияния на эффективность схемы кольцевой подписи, определяемую объемом вычислений, требуемых от каждого субъекта схемы, и размером подписи. В результате сравнения выбран механизм отзыва, основанный на свойстве контролируемой связываемости, позволяющий производить отзыв права подписи без дополнительных вычислений для участников группы за счет добавления в схему центра отзыва, который является онлайн-сервером для проверяющих. Влияние выбранного механизма на размер подписи и на объем вычислений, производимых при формировании и проверке подписи, не зависит от числа участников группы.

Для применения данного механизма выбранная схема кольцевой подписи была дополнена алгоритмом сравнения подписей с помощью протокола AoN-PKEET, реализующего схему шифрования с открытым ключом с тестами на равенство. Предложены два варианта применения механизма в зависимости от характеристик системы – с черным и белым списками. Список отзыва хранится центром отзыва и содержит сертификаты пользователей, зашифрованные с помощью протокола AoN-PKEET. При проверке подписи зашифрованный сертификат, являющийся одним из элементов подписи, сравнивается со всеми подписями в списке отзыва с помощью тестов на равенство. Для отзыва права подписи у участника группы его зашифрованный сертификат исключается из списка отзыва (в случае белого списка) или добавляется в него (в случае черного списка). Описаны способы повышения безопасности и эффективности схемы за счет разделения ключа связывания, которым владеет центр отзыва, между не-

сколькими центрами связывания, а также за счет хранения списка отзыва в виде хеш-таблицы.

Предложенная схема реализована программно на языке Python. Произведено тестирование системы с замерами времени исполнения алгоритмов генерации ключей, формирования и проверки подписи, а также отзыва права подписи у участника в зависимости от количества участников группы. Полученные результаты подтвердили теоретические оценки, сделанные при выборе механизма отзыва. По полученным результатам сделаны выводы о том, что влияние механизма отзыва на алгоритмы формирования и проверки подписи незначительно, так как объем дополнительных вычислений не зависит от размера группы, при этом время выполнения алгоритма отзыва права подписи значительно ниже, чем у других рассмотренных механизмов, так как не требует дополнительных вычислений от пользователей схемы.

Таким образом, разработанная схема кольцевой подписи на решетках обеспечивает незамедлительное удаление участников группы без необходимости синхронизации данных между пользователями, позволяет производить проверку подписи за время, не зависящее от списка отзыва, и обеспечивает безопасность работы центра отзыва.

Результаты работы могут быть использованы в системах контроля доступа к ресурсу, для которых важно иметь возможность незамедлительного ограничения доступа определенного субъекта к ресурсу, при этом отсутствует необходимость уникальной идентификации субъектов в пределах некоторого круга.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Rivest R. L. How to leak a secret // International Conference on the Theory and Application of Cryptology and Information Security. – Springer, Berlin, Heidelberg, 2001. – P. 552-565.

2. Gentry C. Trapdoors for hard lattices and new cryptographic constructions // Proceedings of the fortieth annual ACM symposium on Theory of computing. – ACM, 2008. – P. 197-206.

3. Lyubashevsky V. Lattice signatures without trapdoors // Annual International Conference on the Theory and Applications of Cryptographic Techniques. – Springer, Berlin, Heidelberg, 2012. – P. 738-755.

4. Brakerski Z., Kalai Y. T. A Framework for Efficient Signatures, Ring Signatures and Identity Based Encryption in the Standard Model // IACR Cryptology ePrint Archive. – 2010. – Vol. 2010. – P. 86.

5. Cayrel P. L. et al. A lattice-based threshold ring signature scheme // International Conference on Cryptology and Information Security in Latin America. – Springer, Berlin, Heidelberg, 2010. – P. 255-272.

6. Wang C., Wang H. A new ring signature scheme from NTRU lattice // 2012 Fourth International Conference on Computational and Information Sciences. – IEEE, 2012. – P. 353-356.

7. Wang S. Lattice-based ring signature scheme under the random oracle model // International Journal of High Performance Computing and Networking. – 2018. – Vol. 11. – №. 4. – P. 332-341.

8. Ateniese G. Quasi-efficient revocation of group signatures // International Conference on Financial Cryptography. – Springer, Berlin, Heidelberg, 2002. – P. 183-197.

9. Bresson E., Stern J. Efficient revocation in group signatures // International Workshop on Public Key Cryptography. – Springer, Berlin, Heidelberg, 2001. – P. 190-206.

10. Boneh D., Shacham H. Group signatures with verifier-local revocation // Proceedings of the 11th ACM conference on Computer and communications security. – ACM, 2004. – P. 168-177.

11. Camenisch J., Lysyanskaya A. Dynamic accumulators and application to efficient revocation of anonymous credentials // Annual International Cryptology Conference. – Springer, Berlin, Heidelberg, 2002. – P. 61-76.

12. Slamanig D. Linking-based revocation for group signatures: a pragmatic approach for efficient revocation checks // International Conference on Cryptology in Malaysia. – Springer, Cham, 2016. – P. 364-388.

13. Александрова Е. Б., Рехвиашвили И. Ш. Организация отзыва для схемы кольцевой подписи // Проблемы информационной безопасности. Компьютерные системы. – 2019. – №. 2. – P. 80-85.

14. Ростовцев А. Г., Маховенко Е. Б. Теоретическая криптография – СПб.: АНО НПО «Профессионал, 2005. – 479 с.

15. Micciancio D. Lattice-based cryptography // Encyclopedia of Cryptography and Security. – 2011. – P. 713-715.

16. Nguyen P. Q., Stern J. The two faces of lattices in cryptology // International Cryptography and Lattices Conference. – Springer, Berlin, Heidelberg, 2001. – P. 146-180.

17. Boneh D., Boyen X., Shacham H. Short group signatures // Annual international cryptology conference. – Springer, Berlin, Heidelberg, 2004. – P. 41-55.

18. Brickell E., Li J. Enhanced privacy ID: A direct anonymous attestation scheme with enhanced revocation capabilities // Proceedings of the 2007 ACM workshop on Privacy in electronic society. – 2007. – P. 21-30.

19. Au M. H. Dynamic universal accumulators for DDH groups and their application to attribute-based anonymous credential systems // Cryptographers Track at the RSA Conference. – Springer, Berlin, Heidelberg, 2009. – P. 295-308.

20. Nakanishi T., Funabiki N. A short verifier-local revocation group signature scheme with backward unlinkability // IEICE transactions on fundamentals of elec-

tronics, communications and computer sciences. – 2007. – Vol. 90. – №. 9. – P. 1793-1802.

21. Zhou S., Lin D. Shorter verifier-local revocation group signatures from bilinear maps // International Conference on Cryptology and Network Security. – Springer, Berlin, Heidelberg, 2006. – P. 126-143.

22. Blazy O. Non-interactive plaintext (in-) equality proofs and group signatures with verifiable controllable linkability // Cryptographers' Track at the RSA Conference. – Springer, Cham, 2016. – P. 127-143.

23. Tang Q. Public key encryption supporting plaintext equality test and user-specified authorization // Security and Communication Networks. – 2012. – Vol. 5. – №. 12. – P. 1351-1362.

24. Fiat A., Shamir A. How to prove yourself: Practical solutions to identification and signature problems // Conference on the Theory and Application of Cryptographic Techniques. – Springer, Berlin, Heidelberg, 1986. – P. 186-194.

25. Shamir A. How to share a secret // Communications of the ACM. – 1979. – Vol. 22. – №. 11. – P. 612-613.

26. Sage Reference Manual v9.0 [Электронный ресурс]. URL: https://doc.sagemath.org/html/en/reference/index.html. – (дата обращения: 15.11.2019).

27. Python 2.7.17 Documentation [Электронный ресурс]. URL: https://docs.python.org/2/index.html. – (дата обращения: 15.11.2019).

Приложение. Исходный код разработанной программы

```
from sage.all import *
from sage.stats.distributions.discrete_gaussian_lattice import Discrete-
GaussianDistributionLatticeSampler
import random
import hashlib
import time


# число участников кольца
l = 500
# параметры кольцевой подписи
n = 64
m = 239
q = 61
k = 80
M = 2.7
# параметры нормального распределения
sd = 300 # среднеквадратичное отклонение
eta = 1.1
D = DiscreteGaussianDistributionLatticeSampler(ZZ**m, sd)

# параметры шифрования
p = 134190054955512487306413020496314770876925358130 1 # порядок мульти-
пликативной группы G
g                                                                                =
149225785016883513407046604525768831537615089201297848785381912915612058086674550649014514265502363536958373695472994489001965641360486708187512737044308 6# образующая группы G

# хеш-функция, используемая при шифровании
def H2(x):
    h = int(hashlib.sha512(str(x)).hexdigest(), 16) % p
    return h

# генерация параметров шифрования
def PKEET_param_gen():
    q1 = 3*p+1
    h = random.randint(1, p)
    g = pow(h, (q1-1/p), q1)
    return 1

# генерация ключей шифрования
# Y[] - закрытые ключи участников кольца
# G_Y[] - открытые ключи участников кольца
def PKEET_key_gen():
    Y = []
    G_Y = []
    i = 0
    while i < l:
        y = random.randint(1, p)
        g_y = pow(g, y, p)
        Y.append(y)
        G_Y.append(g_y)
        i = i + 1
```

```
    return (Y, G_Y)

# зашифрование сообщения msg на открытом ключе g_y
# шифртекст – (C2, C4)
def PKEET_enc(msg, g_y):
    v = random.randint(1, p)
    C2 = pow(g, v, p)
    C4 = pow(g, H2(pow(g_y, v, p)) + msg, p)
    return (C2, C4)

# тестирование на равенство шифртекстов (Ci2, Ci4) и (Cj2, Cj4)
# Ti, Tj – ключ связывания
def PKEET_com(Ci2, Ci4, Cj2, Cj4, Ti, Tj):
    equal = 0
    gi = pow(g, H2(pow(Ci2, Ti, p)), p)
    gi_inv = inverse_mod(gi,p)
    gj = pow(g, H2(pow(Cj2, Tj, p)), p)
    gj_inv = inverse_mod(gj,p)
    if ((Ci4 * gi_inv) % p  == (Cj4 * gj_inv) % p) :
        equal = 1
    return equal

# белый список
# каждая запись состоит из следующих элементов:
# (c2, c4) – зашифрованный сертификат
# T – ключ связывания
# 0/1 – участник активен или неактивен
Wight_List = []
RS_sk = [] # закрытые ключи схемы подписи
# открытые ключи схемы подписи (RS_vk_T, RS_vk_A)
RS_vk_T = 0
RS_vk_A = []
PKEET_sk = [] # закрытые ключи схемы шифрования
PKEET_pk = [] # открытые ключи схемы шифрования
CERT = [] # сертификаты пользователей

# генерация сертификатов
def CERT_gen():
    cert = []
    i = 0
    while i < l:
        cert.append(random.randint(1, p))
        i = i + 1
    return cert

# создание белого списка, состоящего из зашифрованных сертификатов
def WL_gen():
    WL = []
    i = 0
    while i < l:
        c2, c4 = PKEET_enc(CERT[i], PKEET_pk[i])
        el = [c2, c4, PKEET_sk[i], 1]
        WL.append(el)
        i = i + 1
    return WL

# хеш-функция, используемая при формировании подписи
def H(x):
```

```
    h = int(hashlib.sha512(str(x)).hexdigest(), 16)
    out = vector([0]*k, Zmod(q))
    for i in range(0, k) :
        out[i] = h%3
        h /= 3
        i += 1
    return out

# генерация ключей подписи
# A[] – открытые ключи участников кольца
# S[] – закрытые ключи участников кольца
def RS_key_gen():
    T = random_matrix(Zmod(q), n, k)
    S = []
    A = []
    i = 0
    while i < l:
        S.append(matrix(Zmod(q), m, k, lambda i, j: choice(range(0, 3))))
        A.append(S[i].solve_left(T))
        i = i + 1
    sk = S
    vk = (A, T)
    return (sk, vk)

# генерация подписи
# A[] - открытые ключи участников кольца
# Sj - секретный ключ автора подписи
# (z,c) - подпись
def RS_sign(msg, A, Sj, j):
    count = 0
    while True:
        count+=1
        if count == 1e3: # ограничение на количество запусков
            return (0, 0)

        y = []
        z = []
        Ay = 0
        i = 0
        while i < l:
            y.append(vector(D()))
            Ay = Ay + A[i] * y[i].change_ring(Zmod(q))
            i = i + 1
        c = H((Ay, A, msg))
        i = 0
        while i < l:
            if i != j:
                z.append(y[i])
            else:
                Sc = Sj.change_ring(ZZ)*c.change_ring(ZZ)
                z.append(Sc + y[i])
            i = i + 1
        pxe = float(-2*z[j]*Sc + (Sc.norm())**2)
        if random.random() < exp(pxe/(2*(sd**2)))/M:
            return (z,c)

# проверка подписи
# (z,c) – подпись, A[] - открытые ключи участников кольца
```

```
def RS_verify(msg, z, c, A, T):
    i = 0
    ok_norm = 1
    Az = 0
    while i < l:
        if z[i].norm() > eta*sd*sqrt(m):
            ok_norm = 0
        Az = Az + A[i] * z[i]
        i = i + 1
    if ok_norm == 1 and c == H((Az-T*c, A, msg)):
        return 1
    return 0

# генерация ключей подписи и шифрования, сертификатов
# формирование белого списка
def setup():
    global RS_sk
    global RS_vk_T
    global RS_vk_A
    global PKEET_sk
    global PKEET_pk
    global CERT
    global Wight_List
    start = time.time()
    RS_sk, RS_vk = RS_key_gen() # ключи подписи пользователей
    RS_vk_A = RS_vk[0]
    RS_vk_T = RS_vk[1]
    PKEET_sk, PKEET_pk = PKEET_key_gen() # ключи шифрования пользователей
    CERT = CERT_gen()
    Wight_List = WL_gen()
    end = time.time()
    print "Время генерации: ", (end-start)
    return 0

# формирование подписи сообщения msg, состоящей из подписи (z, c)
# и зашифрованного сертификата (c2, c4)
def sign(msg, j):
    global RS_sk
    global RS_vk_A
    global CERT
    global PKEET_pk
    start = time.time()
    z, c = RS_sign(msg, RS_vk_A, RS_sk[j], j)
    c2, c4 = PKEET_enc(CERT[j], PKEET_pk[j])
    cc = (z, c, c2, c4)
    end = time.time()
    print "Время подписывания: ", (end-start)
    return cc

# проверка корректности подписи (z, c) сообщения msg
# и проверка наличия сертификата в списке отзыва
def verify(msg, z, c, c2, c4, j):
    global RS_vk_A
    global RS_vk_T
    global PKEET_sk
    start = time.time()
    ver_sign = 1
    if (RS_verify(msg, z, c, RS_vk_A, RS_vk_T) != 1):
```

```
        ver_sign = 0
    end = time.time()
    print "Время проверки проверяющим: ", (end-start)
    #print "ver_sign", ver_sign
    start = time.time()
    ver_cert = 0
    for i in Wight_List:
        if ((PKEET_com(c2, c4, i[0], i[1], PKEET_sk[j], i[2]) == 1) and
(i[3] == 1)) :
            ver_cert = 1
    end = time.time()
    print "Время проверки центром отзыва: ", (end-start)
    #print "ver_cert", ver_cert
    return ver_sign*ver_cert

# исключение сертификата (c2, c4) из списка отзыва
def revoke(c2, c4, j):
    start = time.time()
    global Wight_List
    global PKEET_sk
    for i in Wight_List:
        if (PKEET_com(c2, c4, i[0], i[1], PKEET_sk[j], i[2]) == 1) : i[3]
= 0
    end = time.time()
    print "Время отзыва: ", (end-start)
    return 0

# реализация базовой схемы с перевыпуском ключей при отзыве права подписи
def basic():
    start = time.time()
    sk, vk = RS_key_gen()
    end = time.time()
    print "Время генерации: ", (end-start)

    start = time.time()
    z, c = RS_sign("new world", vk[0], sk[0], 0)
    end = time.time()
    print "Время подписывания: ", (end-start)

    start = time.time()
    RS_verify("new world", z, c, vk[0], vk[1])
    end = time.time()
    print "Время проверки: ", (end-start)

    start = time.time()
    sk, vk = RS_key_gen()
    end = time.time()
    print "Время отзыва: ", (end-start)

    return 0

setup() # генерация ключей
z, c, c2, c4 = sign("new world", 1) # формирование подписи
verify ("new world", z, c, c2, c4, 1) # проверка подписи
revoke(c2, c4, 1) # отзыв права подписи
print verify ("new world", z, c, c2, c4, 4) # повторная проверка подписи

basic() # схема с перевыпуском ключей
```