МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра компьютерной алгебры и теории чисел

Криптосистемы с открытым ключом и подписи

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студентки 2 курса 227 группы

направление 02.04.01 — Математика и компьютерные науки

механико-математического факультета

Жумагалиева Айтаса Толегеновича

Научный руководитель
доцент, к. ф.-м.н., доцент Е.В. Сецинская

Зав. кафедрой
зав. каф., к.ф.-м.н., доцент А.М. Водолазов

Саратов 2019

Введение. Целью магистерской работы является изучение алгоритмов криптосистемы с открытым ключом и подписи. Начало асимметричным шифрам было положено в работе «Новые направления в современной криптографии» Уитфилда Диффи и Мартина Хеллмана, опубликованной в 1976 году. Находясь под влиянием работы Ральфа Меркле (Ralph Merkle) о распространении открытого ключа, они предложили метод получения секретных ключей, используя открытый канал. Этот метод экспоненциального обмена ключей, который стал известен, как обмен ключами Диффи-Хеллмана, был первым опубликованным практичным методом для установления разделения секретного ключа между заверенными пользователями канала. В 2002 году Хеллман предложил называть данный алгоритм «Диффи — Хеллмана — Меркле», признавая вклад Меркле в изобретение криптографии с открытым ключом. Эта же схема была разработана Малькольмом Вильямсоном в 1970-х, но держалась в секрете до 1997 года. Метод Меркле по распространению открытого ключа был изобретён в 1974 году и опубликован в 1978, его также называют загадкой Меркле. В 1977 году учёными Рональдом Райвестом (Ronald Linn Rivest), Ади Шамиром (Adi Shamir) и Леонардом Адлеманом (Leonard Adleman) из Массачусетского Технологического Института (MIT) был разработан алгоритм шифрования, основанный на проблеме о разложении на множители. Система была названа по первым буквам их фамилий. Эта же система была изобретена Клиффордом Коксом (Clifford Cocks) в 1973 году, работавшим в центре правительственной связи (GCHQ). Но эта работа хранилась лишь во внутренних документах центра, поэтому о её существовании было не известно до 1977 года. В основу известных асимметричных криптосистем кладётся одна из сложных математических проблем, которая позволяет строить односторонние функции и функции-лазейки. Например, криптосистемы Меркля — Хеллмана и Хора — Ривеста опираются на так называемую задачу об укладке рюкзака. Напомним, что в симметричной криптографии каждая из переписывающихся сторон должна иметь копию общего секретного ключа, что создает сложнейшую проблему управления ключами. В криптосистемах, о которых пойдет речь в этой курсовой работе, используются два ключа: открытый и секретный. Открытый ключ может быть опубликован в справочнике наряду с именем пользователя. В ре-

зультате любой желающий может зашифровать с его помощью свое письмо и послать закрытую информацию владельцу соответствующего секретного ключа. Расшифровать посланное сообщение сможет только тот, у кого есть секретный ключ. Более точно, имеют место преобразования:

сообщение + ОТКРЫТЫЙ КЛЮЧ АЛИСЫ = ШИФРОТЕКСТ ШИФРОТЕКСТ + секретный ключ Алисы = сообщение.

Таким образом, каждый может послать Алисе секретную информацию, воспользовавшись ее открытым ключом. Но только Алиса в состоянии расшифровать сообщение, поскольку лишь у нее есть соответствующий секретный ключ. Причина работоспособности таких криптосистем заключается в односторонней математической связи, существующей между двумя ключами, при которой информация об открытом ключе никак не помогает восстановить секретный, но владение секретным ключом обеспечивает возможность расшифровывать сообщения, зашифрованные открытым. На первый взгляд такая связь кажется странной, и для ее понимания требуется определенное время и умственные усилия. Идея криптографии с открытым ключом впервые появилась в 1976 г. в революционной работе Диффи и Хеллмана «Новые направления в криптографии». Но только год спустя была опубликована первая (и наиболее успешная) криптосистема с открытым ключом, а именно, RSA. В предыдущем абзаце описана «официальная» история возникновения криптографии с открытым ключом. Однако в конце 1990-ых годов появилась неофициальная версия освещения событий. Оказалось, что в 1969 году, более чем за пять лет до публикации основополагающей работы Диффи и Хеллмана, Джеймс Эллис, работающий на центр связи Британского правительства GCHQ, открыл концепцию криптографии с открытым ключом (или несекретное шифрование, как он ее называл) как средство решения проблемы распределения ключей. Впрочем, Эллис, как и Диффи с Хеллманом, не смог разработать в деталях соответствующую криптосистему.

Основная часть. Основная часть данной работы состоит из 3 разделов, а именно: 1. Алгоритмы шифрования с открытым ключом; 2. Цифровые подписи; 3. Реализация операций в **RSA** и **DSA**. Наиболее важная односторонняя функция, используемая в криптографии с открытым ключом, — это разложение на множители или факторизация целых чисел. Под разложением на

множители целого числа понимают его представление в виде произведения простых делителей, например,

$10 = 2 * 5$,

$60 = 2^2 * 3 * 5$,

$2^{113} - 1 = 3391 * 23279 * 65993 * 1868569 * 1066818132868207$.

Определение множителей является очень трудоемкой вычислительной операцией. Для оценки сложности алгоритма, раскладывающего целое число N на простые множители, часто используют функцию

$$L_N(\alpha, \beta) = \exp\left((\beta + o(1))(\ln N)^{\alpha}(\ln \ln N)^{1-\alpha}\right).$$

Обратим внимание на то, что если алгоритм, раскладывающий на множители целое число, имеет сложность $O(L_N(0, \beta))$, то ему требуется полиномиальное время на работу (напомним, что размер задачи на входе — $\ln N$). Однако при сложности алгоритма $O(L_N(1, \beta))$, ему для работы потребуется уже экспоненциальное время. Таким образом, скорость роста функции $L_N(\alpha, \beta)$ при $0 < \alpha < 1$ лежит между полиномиальной и экспоненциальной. Поэтому про алгоритм со сложностью $O(L_N(\alpha, \beta))$ при $0 < \alpha < 1$ говорят, что он требует суб-экспоненциального времени. Заметим, что умножение, т.е. процесс обратный к разложению на множители, — очень простая операция, требующая времени меньше, чем $O(L_N(0, 2))$

Лемма 1.1. В любой конечной абелевой группе G задача Диффи-Хеллмана не сложнее проблемы дискретного логарифмирования.

Лемма 1.2. Проблема выбора Диффи-Хеллмана в любой конечной абелевой группе G не сложнее задачи Диффи-Хеллмана.

Лемма 1.3. Задачи разложения на множители и вычисления квадратного корня полиномиально эквивалентны.

Лемма 1.4. Задача RSA не сложнее проблемы факторизации.

Существует гипотеза, подтвержденная некоторыми косвенными соображениями, что задача RSA на самом деле легче проблемы факторизации, т. е. эти задачи не эквивалентны [7]. В настоящее время проверка этой гипотезы — один из главных открытых вопросов криптологии. Алгоритм RSA, первый из алгоритмов шифрования с открытым ключом, достойно выдержал

испытание временем. Этот алгоритм основывается на задаче RSA, с которой мы познакомились в предыдущем параграфе. Как вы помните, она сводится к поиску простых делителей больших натуральных чисел. Так что можно утверждать, что криптостойкость алгоритма RSA базируется на сложности проблемы факторизации, хотя и не в полной мере, поскольку задачу RSA можно решать не прибегал к разложению на множители.

Простейший алгоритм шифрования, основывающийся на дискретном логарифмировании, — это криптосистема Эль-Гамаль. Сейчас опишем шифрование в системе Эль-Гамаль, использующее конечные поля. Существует аналогичная система эллиптических кривых. В отличие от RSA, в алгоритме Эль-Гамаль существуют некоторые открытые параметры, которые могут быть использованы большим числом пользователей. Они называются параметрами домена и выглядят следующим образом: P — «большое простое число», т. е. число, насчитывающее около 1024 битов, такое, что P — 1 делится на другое, «среднее простое число» Q, лежащее неподалеку от 2^{160}. G — элемент мультипликативной группы поля F^*p, порядок которой, делится на Q, причем

$$G^{(P-1)/Q}(mod\ P) \neq 1.$$

Все параметры домена, т. е. P, Q и G, выбираются таким образом, чтобы элемент

$$G^{(P-1)/Q}$$

был образующей абелевой группы A порядка Q, Информация об этой группе открыта и используется большим числом пользователей. После выбора параметров домена определяют открытый и секретный ключи. Секретным ключом может априори быть любое натуральное число x, а открытый ключ получается по следующей формуле:

$$H = G^x(mod\ P).$$

Лемма 1.5. Если задача Диффи - Хеллмана трудноразрешима, то система Эль-Гамаль защищена против атак с выбором открытого текста, где защи-

щенность означает, что нападающий не может восстановить открытый текст по перехваченной шифрограмме за разумное время.

Есть еще одна криптосистема, принадлежащая Рабину, которая основывается на трудной проблеме факторизации больших целых чисел. Более точно, она связана с трудностью извлечения квадратного корня по модулю составного числа $N = \rho * q$. Напомним, что эти задачи эквивалентны, т. е. зная простые делители числа N, мы можем извлекать квадратные корни по модулю N, умея извлекать квадратные корни по модулю N, в состоянии разложить N на простые множители.

Потому такую систему можно считать в некотором отношении более криптостойкой, чем RSA. Процесс шифрования в алгоритме Рабина происходит намного быстрее, чем практически в любой другой криптосистеме с открытым ключом. Однако, несмотря на эти преимущества, система Рабина используется все же реже, чем RSA. Тем не менее, система Рабина важна как с исторической точки зрения, так и в качестве наглядного примера криптосистемы, основные идеи которой используются в протоколах более высокого уровня. Выберем разные простые числа, удовлетворяющие условию:

$$\rho = q = 3(mod\ 4).$$

Одна из возможностей пресечь атаку на протокол Диффи - Хеллмана, описанную в конце предыдущего параграфа, заключается в визировании сообщений. В этом случае обе стороны доподлинно знают, с кем ведут обмен сообщениями. Подписи — важнейший момент в криптографии с открытым ключом. Они были предложены Диффи и Хеллманом все в той же статье 1976 года, но первая практическая разработка подписи принадлежит Ривесту, Шамиру и Адле-ману. Движущая пружина подписей в криптографии с открытым ключом изображена на следующей схеме:

СООБЩЕНИЕ + секретный ключ Алисы = ПОДПИСЬ,

СООБЩЕНИЕ + ПОДПИСЬ + ОТКРЫТЫЙ КЛЮЧ АЛИСЫ = ДА/НЕТ.

Эта схема называется схемой подписи с преломлением, так как подпись добавляется в конец сообщения перед его передачей. Полученное сообщение

подается на вход процедуры проверки подписи. Другой вариант — схема подписи с восстановлением сообщения, когда сообщение восстанавливается на выходе процедуры проверки подписи:

СООБЩЕНИЕ + секретный ключ Алисы = ПОДПИСЬ, ПОДПИСЬ + ОТКРЫТЫЙ КЛЮЧ АЛИСЫ = ДА/НЕТ + СООБЩЕНИЕ.

Важным моментом здесь является то, что только Алиса может подписать свое сообщение, поскольку только она имеет доступ к своему секретному ключу. С другой стороны, ее подпись может быть проверена любым желающим, поскольку для этого нужен лишь открытый ключ Алисы.

Главная проблема состоит в том, каким открытым ключам можно доверять. Как удостовериться, что тот или иной открытый ключ принадлежит конкретному лицу? Вы можете считать, что данный ключ использует Алиса, в то время как он имеет отношение к Еве. Поэтому Ева может подписывать чеки или нечто подобное, а вы будете считать, что все это исходит от Алисы. Кажется мы столкнулись с той же проблемой управления ключами, что имела место в симметричных системах, хотя акцент теперь падает не на сохранение ключей в тайне, а на проверку их подлинности. Более формально, схема цифровой подписи состоит из таких двух преобразований: секретное преобразование подписи s, открытое преобразование проверки v.

Криптографическая хэш-функция h — это функция, определенная на битовых строках произвольной длины со значениями в строках битов фиксированной длины. Ее значение часто называют хэги-кодом или хэш-значением, В информатике тоже используются своего рода хэш-функции, но важное отличие криптографических хэш-функций от стандартных состоит в том, что первые должны быть односторонними. Другими словами, должно быть невозможно в вычислительном отношении по элементу Y из множества значений хэш-функции подобрать такой x из области определения, при котором $h(x) = Y$. Другая характеризация односторонних хэш-функций — сказать о них, что они защищены от восстановления прообразов. Применение криптографических хэш-функций позволяет создать схему подписи RSA без восстановления сообщения, что намного эффективней для длинных сообщений.

Предположим, нам дано длинное сообщение M для визирования. Сначала вычисляется $h(M)$ и потом применяется преобразование подписи RSA к хэш-

значению h(M), т.е. подпись получается как

$$S = h(M)^d (mod\ N).$$

Наконец, подпись и само сообщение передаются вместе в виде пары (M,S). Проверка пары (M,S) состоит из трех этапов: «Шифрование» S с помощью шифрующей экспоненты RSA для получения $H^{'}$:

$$H^{'} = S^E (mod\ N).$$

Вычисление h(M) по M. Проверка равенства $H^{'}$ — h(M), Если оно верно, то подпись законна. В противном случае — незаконна.

Лемма 2.1. Свойство защищенности от восстановления прообразов сильнее защищенности от повторений и вторых прообразов.

Лемма 2.2. Защищенность от вторых прообразов сильнее защищенности от повторений.

Множество вариантов схем подписей основывается на дискретных логарифмах. С практической точки зрения интересен алгоритм, носящий название подписи Шнорра. Расскажем об этом алгоритме в его оригинальном виде, оставив читателю разбираться самостоятельно с его обобщением на эллиптические кривые.

Пусть A — открытая конечная абелева группа, порожденная элементом G простого порядка Q. Ключевая пара в алгоритме подписи Шнорра совпадает с такой же парой в DSA, а именно, секретным ключом служит целое число x из интервала (0,Q), а открытый ключ определяется формулой Y = G^x. Чтобы подписать сообщение, в алгоритме Шнорра поступают следующим образом: Выбирают эфемерный ключ k из промежутка (0,Q). Вычисляют соответствующий открытый ключ R — G^k. Находят $E = h(M||R)$. Обратите внимание на то, что значение хэш-функции зависит как от сообщения, так и от эфемерного открытого ключа. Вычисляют $S = k + x * E(mod\ Q)$.

Полученная таким образом пара (E,S) является искомой подписью. Проверка подписи довольно проста: вычисляют $R = G^S Y^{-E}$ и h(M||R), Подпись корректна, если верно равенство E=h(M||R). Для примера выберем следую-

щие параметры домена:

$$Q = 101,\ P = 607 \text{ и } G = 601.$$

Чтобы зафиксировать ключевую пару, положим x=3 и

$$Y = G^x (mod\ P) = 391.$$

Затем генерируем эфемерный ключ k=65 и вычисляем

$$R = G^k (mod\ p) = 223.$$

Теперь находим хэш-значение $E = h(M||R)(mod\ Q)$. Допустим, что при этом получилось E=93, Тогда вторая компонента подписи выглядит как

$$S = k + x * E(mod\ Q) = 65 + 3 * 93(mod\ 101) = 41.$$

Может оказаться, что подпись на коротком сообщении будет более длинной, чем его основное содержание. Напомним, что RSA можно использовать и как схему подписи с приложением, и как схему подписи с восстановлением сообщения. Пока ни один из описанных нами алгоритмов подписи, основанный на дискретном логарифмировании, нельзя использовать в виде схемы с восстановлением сообщения. Приведем пример схемы подписи с восстановлением сообщения, называемый алгоритмом подписи Ниберга-Руппеля, основанный на вычислении логарифмов в открытой конечной абелевой группе A.

Все схемы подписи с восстановлением сообщения используют открытую функцию избыточности f. Функция f преобразует фактическое сообщение в данные, которые затем подписываются. Это действие напоминает то, что делает хэш-функция в схемах подписи с приложением. Однако в отличие от хэш-функции функция избыточности должна быть легко обратимой. Для ускорения процесса возведения в степень в алгоритме RSA применяется множество приемов, специально разработанных для этой криптосистемы. Выбор ускоряющего метода зависит от того, выполняем ли мы процедуру шифрования (проверки подлинности подписи) с открытым ключом или обращаемся

к процессу расшифрования (подписывания сообщения) с секретным ключом. В случае расшифрования или подписывания сообщения в алгоритме RSA показатель вычисляемой степени будет общим 1000-битовым числом. Следовательно, нам нужен какой-нибудь путь сокращения числа операций. К счастью, знаем секретный ключ,а значит и разложение модуля алгоритма на множители: N = ρq. При расшифровывании сообщения приходится вычислить

$$m = C^d (mod\ N).$$

Как уже было отмечено часто применяются достаточно небольшие открытые (шифрующие) экспоненты, например, E = 3,17 или 65 537. Причина, по которой выбираются такие значения, состоит в том, что эти числа имеют небольшой вес Хем-минга, фактически наименьший из возможных для открытых RSA-ключей, а именно, 2. Поэтому двоичный метод или любой другой разумный алгоритм потенцирования будет использовать лишь одно общее умножение и к возведений в квадрат, где k — число двоичных знаков открытой экспоненты. Например [17],

$$M^3 = M^2 * M, \qquad M^{17} = M^{16} * M = (((M^2)^2)^2)^2 * M.$$

Напомним, что при проверке подлинности подписи в алгоритме DSA нам необходимо вычислить

$$R = G^A Y^B.$$

Это можно сделать, поочередно возводя в степень сначала G, потом Y, а затем перемножая полученные значения. Однако зачастую проще возводить в степень эти элементы одновременно. Существует множество методов, реализующих эту идею, которые используют различные варианты оконной техники или что-либо похожее. Но все они по сути эксплуатируют один трюк, называемый приемом Шамира. Сначала заполняется поисковая таблица

$$G_i = G^{i_0} Y^{i_1},$$

где $i = (i_0, i_1)$ — двоичное представление числа i = 0,1,2,3. Затем заполняется массив показателей, исходя из данных A и B. Его размерность — $2 \times t$, где t —

максимум из числа знаков A и B. Строки массива — двоичное представление показателей A и B. Через $I_j(j = 1, ..., t)$ обозначаются числа, чье двоичное представление задается столбцами этого массива. Наконец, параметру r присваивают единичное значение и вычисляют

$$r = r^2 * G_{I_j} \text{ для } j \text{ от } 1 \text{ до } t.$$

Один из методов ускорения умножения носит название умножения Карацубы. Предположим, нам нужно перемножить два n-битовых числа x и y. Запишем их в виде

$$x = x_0 + 2^{n/2}x_1, \qquad y = y_0 + 2^{n/2}y_1,$$

где $0 \leqslant x_0, x_1, y_0, y_1 < 2^{n/2}$. Умножение происходит с помощью вычислений:

$$A = x_0 * y_0,$$

$$B = (x_0 + x_1) * (y_0 + y_1),$$

$$C = x_1 * y_1.$$

Ответ получается в результате преобразований:

$$C2^n + (B - A - C)2^{n/2} + A = z_1y_1 2^n + (x_1y_0 + x_0y_1)2^{n/2} + x_0y_0 =$$

$$= (x_0 + 2^{n/2}x_1) * (y_0 + 2^{n/2}y_1) = x * y.$$

После знакомства с умножением перейдем к самой сложной арифметической операции — делению. Оно необходимо, в частности, для определения остатков от деления двух целых чисел — основной задачи арифметики остатков, а значит и RSA. Итак, для данных двух целых x иу чисел мы хотим найти такие q и r, что x=qy+r, причем, как обычно, $0 \leqslant r < y$. Напомним, что такое деление называется делением с остатком или евклидовым делением. Представим делимые числа в малословном формате

$$x = (x_0, ..., x_n) \text{ и } y = (y_0, ..., y_n),$$

где основание разложения равно $b = 2^{\omega}$, и обозначим через $t << v$ сдвиг целого числа t влево на v слов, т.е. результат умножения t на b^v. При этих соглашениях алгоритм на псевдокоде выглядит так: (см. в приложении)

Заключение. Алгоритмы криптосистемы с открытым ключом можно использовать, как самостоятельные средства для защиты передаваемой и хранимой информации, как средства распределения ключей. Обычно с помощью алгоритмов криптосистем с открытым ключом распределяют ключи, малые по объёму. А саму передачу больших информационных потоков осуществляют с помощью других алгоритмов, как средства аутентификации пользователей. Преимущество асимметричных шифров перед симметричными шифрами состоит в отсутствии необходимости предварительной передачи секретного ключа по надёжному каналу. В симметричной криптографии ключ держится в секрете для обеих сторон, а в асимметричной криптосистеме только один секретный. При симметричном шифровании необходимо обновлять ключ после каждого факта передачи, тогда как в асимметричных криптосистемах пару (E,D) можно не менять значительное время. В больших сетях число ключей в асимметричной криптосистеме значительно меньше, чем в симметричной.

Недостатки: Преимущество алгоритма симметричного шифрования над несимметричным заключается в том, что в первый относительно легко внести изменения. Хотя сообщения надежно шифруются, но «засвечиваются» получатель и отправитель самим фактом пересылки шифрованного сообщения.