Практическая работа №11. **Алгоритм обмена ключами Диффи–Хеллмана**

11.1. Краткие теоретические сведения

В 1976 году американцы Уитфилд Диффи и Мартин Хеллман (Diffi W., Hellman M.) в статье "Новые направления в криптографии" предложили новый принцип построения криптосистем, не требующий не только передачи ключа принимающему сообщение, но даже сохранения в тайне метода шифрования. Этот метод получил название “метод экспоненциального ключевого обмена” и является первой криптосистемой с открытым ключом. Метод запатентован, но срок действия патента истек 29 апреля 1997 года.

Итак, в 1976 г. Диффи и Хеллман опубликовали статью, которая ознаменовала собой рождении асимметричной криптографии и привела к росту числа открытых исследований в области криптографии. Она содержала ошеломляющий результат: возможно построение практически стойких секретных систем, которые не требуют передачи секретного ключа.

Диффи и Хеллман ввели понятие односторонней функции с потайным ходом. Под *односторонней функцией* f понимается функция f(x), которая легко вычислима для любого значения аргумента x из области определения, однако для данного y из области ее значений вычислительно сложно нахождение значения аргумента x, для которого f(x) = y. Применение таких функций для защиты входа в вычислительную систему путем одностороннего преобразования паролей было известно. Но как применить одностороннюю функцию в криптографических системах, когда даже законный получатель не сможет выполнить дешифрования? Для шифрования была предложена односторонняя функция с потайным ходом (секретом).

Под односторонней функцией с потайным ходом понимается семейство обратимых функций f_z с параметром z , таких, что для данного z можно найти алгоритмы E_z и D_z , позволяющие легко вычислить значение $f_z(x)$ для всех x из области определения, а также вычислить значение $f_z^{-1}(y)$ для всех y из области значений, однако практически для всех значений параметра z и практически для всех значений y из области значений f_z нахождение $f_z^{-1}(y)$ вычислительно неосуществимо даже при известном E_z.

В качестве односторонней функции Диффи и Хеллман предложили функцию дискретного возведения в степень $f(x) = g^{\wedge x} \pmod n$, где x – целое число, $1 \le x \le n - 1$, n – k-битовое простое число.

Причем выбирается такое число $g < n$, степени которого по модулю n представляют собой упорядоченное множество чисел $\{g^1, g^2, \dots , g^{n-1}\}$, являющееся некоторой перестановкой чисел {1, 2, … , n-1}. (Такое число g называется первообразным корнем по модулю n.)

Даже для очень больших модулей n (например, при k = 1024 бит) для данного x легко вычислить значение этой функции. Процедура вычисления этой функции называется дискретным возведением в степень. Для выполнения этой процедуры достаточно выполнение около $2\log_2 n$ операций умножения k-битовых чисел (или $\log_2 n$ умножений и $\log_2 n$ делений 2k-битовых чисел на k-битовые). Процедура дискретного возведения в степень основана на предварительном вычислений значений (по модулю n).

Обратной к функции дискретного возведения в степень является функция $f^{-1}(y)$, которая ставит в соответствие заданному значению y такое значение x, для которого выполняется условие $g^{\wedge x} = y \pmod n$. Задача нахождения такого x называется задачей дискретного логарифмирования (нахождения дискретных логарифмов). Дискретные логарифмы сложно вычисляются, когда число n-1 содержит один большой простой множитель, например, когда оно представимо в виде $n-1 = 2n'$, где n' - простое число.

При этом условии трудоемкость задачи нахождения дискретного логарифма равна примерно $\sqrt{n}$ умножений по модулю n. Решение такой задачи является вычислительно неосуществимым при больших значениях k (например, при $k \ge 512$), а следовательно, при указанных условиях, накладываемых на выбор чисел n и g , функция дискретного возведения в степень является односторонней.

Методом открытого распространения ключей Диффи–Хеллмана называется следующий способ использования дискретного возведения в степень для обмена секретными ключами между пользователями сети с применением только открытых сообщений. Выбирается большое простое число n и соответствующий ему первообразный корень $g < n$. (Для обеспечения стойкости рассматриваемой системы открытого шифрования на число n накладывается следующее условие: разложение числа n-1 на множители должно содержать по крайней мере один большой простой множитель; размер числа n должен быть не менее 512 бит.)

Механизм распределения секретных ключей по открытому каналу состоит в следующем. Каждый абонент выбирает случайный секретный ключ x и вырабатывает открытый ключ y, соответствующий выбранному секретному ключу, в соответствии с формулой $y = g^x \pmod n$.

Для любого значения x легко вычислить y, однако при размере числа n, равном 512 бит и более, вычислительно неосуществимо выполнение дискретного логарифмирования, а следовательно и определение числа x, для которого значение $g^x \bmod n$ равно заданному значению y.

Все абоненты размещают свои открытые ключи в общедоступном справочнике. Данный справочник должен быть заверен специально созданным доверительным центром, чтобы исключить возможные нападения путем подмены открытых ключей или навязывания ложных открытых ключей. Если два абонента Боб и Алиса хотят установить секретную связь, то они поступают следующим образом.

Общий секретный ключ может использоваться абонентами для шифрования сеансовых секретных ключей, а последние – для шифрования сообщений с использованием симметричных методов шифрования. Решение задачи дискретного логарифмирования существует, но оно вычислительно неосуществимо. Таким образом, стойкость метода Диффи – Хеллмана основана на сложности дискретного логарифмирования.

В симметричных криптосистемах существуют две принципиальные проблемы:

• распределение секретных ключей по защищенному каналу;

• аутентификация секретного ключа.

Под аутентификацией понимается проведение процедуры, которая позволяет удостовериться получателю, что секретный ключ принадлежит законному отправителю (например центру распределения ключей). Система открытого распределения ключей решает первую проблему, т.е. она позволяет обойтись без защищенного канала для распределения секретных ключей. Однако она не устраняет необходимость аутентификации.

11.2 Алгоритм Диффи–Хеллмана

Рассмотрим основные положения метода.

Пусть абоненты A и B условились организовать секретную переписку между собой используя секретный ключ сгенерированный при помощи алгоритма Диффи–Хеллмана. Для этого они вместе выбирают два достаточно больших простых числа n и q так, чтобы q было примитивным элементом в GF(n). Эти два числа необязательно хранить в секрете. Абоненты A и B могут передать эти числа по открытому каналу связи. Затем абоненты реализуют следующий алгоритм.

1. A выбирает случайное большое целое число α, вычисляет $x = q^{\alpha} \bmod n$ и посылает B число x.
2. B выбирает случайное большое целое число β, вычисляет $y = q^{\beta} \bmod n$ и посылает A число y.
3. A вычисляет $k_1 = y^{\alpha} \bmod n$.
4. B вычисляет $k_2 = x^{\beta} \bmod n$.

В итоге A и B получили такие числа, что $k_1 = k_2 = q^{\alpha\beta} \bmod n$. Никто из злоумышленников, имеющих доступ к этому открытому каналу не может определить эти значения, так как им известны n, q, x и y, но неизвестны α и β. Для получения α и β необходимо вычислить дискретный логарифм, что является трудной в вычислительном плане задачей. Таким образом, величина $k = k_1 = k_2$ может являться секретным ключом, который A и B вычислили независимо. В данном алгоритме выбор n и q существенно влияет на криптостойкость.

Пример 1. Пусть абоненты A и B условились организовать секретную переписку между собой используя секретный ключ сгенерированный при помощи алгоритма Диффи–Хеллмана. Тогда они вместе выбирают числа n = 67 и q = 11. Затем

1. A выбирает случайное целое число α = 47, вычисляет $x = q^{\alpha} \bmod n = 11^{47} \bmod 67 = 2$ и посылает B число 2.

2. B выбирает случайное целое число β = 51 вычисляет $y = q^{\beta} \bmod n = 11^{51} \bmod 67 = 3$ и посылает A число 3.

3. A вычисляет $k_1 = y^{\alpha} \bmod n = 3^{47} \bmod 67 = 27$

4. B вычисляет $k_2 = x^{\beta} \bmod n = 2^{51} \bmod 67 = 27$

В итоге A и B получили секретный ключ $k = k_1 = k_2 = 27$.

Пример 2. А теперь рассмотрим похожий пример, но с большими числами, а именно n=
1749800579826409539498001781694097092282535544714569949140616485127962399359500738578810541618443087 7 и q=
4980057982640953976500178169262709228253554471452369503406164941279623993595307385078105416180853461.

1. A выбирает случайное целое число α=
1749800579826409539498001781694097092282535544714569949140616485127962399359500738578810541618443136 7 вычисляет $x = q^{\alpha} \bmod n =$
4980057982640953976500178169262709228253554471452369503406164941279623993595307385078105

$16180853461^{1749800579826409539498001781694097092282535544714569949140616485127962399359500738578810541618443136 7}$ mod
1749800579826409539498001781694097092282535544714569949140616485127962399359500738578810541618443087 7 =
1707898771420490189036182320364824629870888864570628383436341394008976949807175044603463218465795703 6 и посылает B число x .

2. B выбирает случайное целое число β=
4374501449566023848745004454235242730706338861786424872851541212819905998398751846447026354046454419 вычисляет y = q^{β} mod n =
4980057982640953976500178169262709228253554471452369503406164941279623993595307385078105416180853461^{4374501449566023848745004454235242730706338861786424872851541212819905998398751846447026354046454419} mod
1749800579826409539498001781694097092282535544714569949140616485127962399359500738578810541618443087 7 = 9446043995490484971892239232298678418450409052905625061872522661677845494058935332443487281531990777 0562506187252266167784549405893533244348728153199077 7 и посылает A число y.

3. A вычисляет $k_1 = y^{\alpha}$ mod n =
9446043995490484971892239232298678418450409052905625061872522661677845494058935332443487281531990777$^{1749800579826409539498001781694097092282535544714569949140616485127962399359500738578810541618443136 7}$ mod
1749800579826409539498001781694097092282535544
7145699491406164851279623993595007385788105416184430877 =
1203419001139627736370826683447563462740343725027845634563745588933246634313030852752683 2915254594034

4. B вычисляет $k_2 = x^{\beta}$ mod n =
1707898771420490189036182320364824629870888864570628383436341394008976949807175044603463218465795703 6^{4374501449566023848745004454235242730706338861786424872851541212819905998398751846447026354046454419} mod
1749800579826409539498001781694097092282535544714569949140616485127962399359500738578810541618443087 7 =
1203419001139627736370826683447563462740343725027845634563745588933246634313030852752683 2915254594034.

В итоге A и B получили секретный ключ $k = k_1 = k_2$ =
1203419001139627736370826683447563462740343725027845634563745588933246634313030852752683 2915254594034.

Без дополнительных мер безопасности (введения сертификатов открытых ключей), рассмотренный метод ключевого обмена уязвим с точки зрения атаки, известной под названием “человек посередине” (main in the midlle attact).

Предположим, что злоумышленник C может не только подслушивать сообщения между A и B, но также изменять, удалять и создавать новые ложные сообщения. Тогда C может выдавать себя за A, что сообщающего B, и за B, что сообщающего A. Атака состоит в следующем:

1. A посылает B свой открытый ключ. C перехватывает его и посылает B свой собственный открытый ключ.

2. B посылает A свой открытый ключ. C перехватывает его и посылает A свой собственный открытый ключ.

3. Когда A посылает сообщения B, зашифрованное на его открытом ключе, C перехватывает его. Т.к. сообщение в действительности зашифровано на открытом ключе C, он расшифровывает его, снова зашифровывает его на открытом ключе B и посылает B.

4. Когда B посылает сообщения A, зашифрованное на его открытом ключе, C перехватывает его. Так как сообщение в действительности зашифровано на открытом ключе C, он расшифровывает его, снова зашифровывает его на открытом ключе A и посылает A.

Атака возможна, даже если открытые ключи A и B и хранятся в БД. Злоумышленник C может перехватить запрос A к БД и подменить открытый ключ. Данная атака очень эффективна. Открытые ключи должны проходить сертификацию, чтобы предотвратить подобные атаки, связанные с подменой ключей и должны регулярно меняться.

11.3. ***Задание*** на практическую работу

1) Разработать программу, реализующую алгоритм обмена ключами Диффи–Хеллмана. Программа должна уметь работать с текстом произвольной длины.

Замечание. На «отлично» необходимо чтобы программа выполняла шифрование данных как с файла, так и с текстового окна программы. На «хорошо» – программа должна выполнять шифрование только с файла. На «удовлетворительно» – программа должна выполнять шифрование только с текстового окна.

2) С помощью соответствующего разработанного Вами программного продукта зашифровать сообщение, представляющее собой первые буквы своих фамилии, имени и отчеств.

Выполнить ту же операцию вручную. Проверить себя, расшифровав сообщение с помощью соответствующего программного продукта.

Практическая работа №12. **Шифросистема RSA** или **КРИПТОСИСТЕМА RSA**.

12.1. Краткие теоретические сведения

В 1978 г. появилась работа [Rivest, R. A Method for obtaining digital signatures and public keyCryptosystems / R. Rivest, A. Shamir, L. Adleman // Communications of the ACM. February. 1978], в которой Рон Райвест (Ron Rivest), Ади Шамир (Adi Shamir) и Лен Адлеман (Len Adleman) предложили алгоритм с открытым ключом. Схема Райвеста–Шамира–Адлемана (RSA) получила широкое распространение. Шифросистема RSA, названая по первым буквам фамилий авторов (R. **R**ivest, A. **S**hamir, L. **A**dleman), находит применение в различных криптографических протоколах и на сегодняшний день.

В июне 2003 года в Сан-Диего, Калифорния, состоялось очередное вручение Тьюринговской премии, учрежденной Ассоциацией вычислительной техники (Association for Computing Machinery). Эта премия названа именем Алана Тьюринга (1912-1954), английского математика, заложившего основы компьютерных наук и внесшего решающий вклад в раскрытие германского шифра «Энигма» в годы Второй мировой войны. Она присуждается с 1966 года специалистам в области компьютерных наук, создавшим теоретические и технические предпосылки для новых, этапных, достижений в области информационных технологий. Лауреатами 2002 года стали Рональд Ривест, Ади Шамир и Леонард Адлмен. В 1977-78 годах, работая в Массачусетском технологическом институте, они создали шифр, названный RSA, который произвел переворот в криптографии и открыл новый период в сфере защиты информации. В настоящее время RSA – самый распространенный метод шифрования, используемый в компьютерных сетях. В этом шифре осуществлена другая казавшаяся несбыточной мечта криптографов: возможность защищенной связи без передачи секретного ключа.

Шифросистема RSA основана на вычислительно сложной задаче факторизации больших целых чисел (разложение на простые множители) и может применяться как для шифрования, так и для электронно-цифровой подписи (ЭЦП) [2].

После некоторых необходимых предварительных сведений дадим краткое описание шифра RSA.

Напомним, что натуральное число, большее 1, называется *простым*, если оно делится только на 1 и на себя. Первые десять простых чисел: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31. Простых чисел бесконечно много, они распределены по натуральному ряду вне какой-либо известной закономерности. Числа, не являющиеся простыми, называются *составными*.

Всякое составное число единственным образом можно представить в виде произведения степеней простых чисел. Например, 12=2^2·3, 45=3^2·5, 105=3·5·7 и т.д. Существующие алгоритмы позволяют персональному компьютеру за несколько секунд проверить, является ли простым предъявленное число, имеющее порядка 180 цифр (уровень современной практической криптографии). В то же время задача разложения на множители столь же больших составных чисел лежит далеко за пределами современных технологических возможностей.

Два натуральных числа a и b называются *взаимно простыми*, если у них нет общих делителей, т.е. таких натуральных чисел, на которые делились бы и a, и b. Так, $50=2\cdot5^2$ и $63=3^2\cdot7$ являются взаимно простыми числами, а $36=2^2\cdot3^2$ и $105=3\cdot5\cdot7$ – нет: у них имеется общий делитель 3.

Рассмотрим шифр. Пусть имеется компьютерная сеть, абоненты которой хотят обмениваться информацией, не предназначенной для непредусмотренных пользователей. Абонент A выбирает два больших (примерно по 100 цифр) простых числа p и q, находит их произведение $n=p\cdot q$ и подбирает целое число e в интервале от 2 до (p-1) (q-1), взаимно простое с p-1 и с q-1. Затем он опубликовывает пару (n,e), это его *открытый* ключ, он применяется для шифрования сообщений.

Предположим, что другой абонент B желает отправить для A секретное сообщение. Он переводит открытый текст в числовую форму m (например, заменяя a на 01, b – на 02, … , z – на 26, а пробел между словами – на 00). Если полученное число m превышает n, его можно разбить на последовательные части, каждая меньше n, так что для простоты пусть $m < n$.

Далее B вычисляет $c=(m^e) \bmod n$. Это криптограмма, которую он и посылает абоненту A. Для того чтобы ее прочитать, A уже заготовил свой закрытый ключ – число d, удовлетворяющее двум требованиям: $1< d < n$ и $e\times d \equiv 1 \pmod{(p-1)\cdot(q-1)}$. Из теории известно, что такое число существует и притом только одно. Далее A вычисляет $(c^d) \bmod n$ и (математическая теорема) получает m.

Возьмем для примера p=3, q=11. Тогда n = p·q = 3·11 =33, (p-1) · (q-1) = 2·10 = 20. Выберем e=7. Открытым ключом является пара чисел (33, 7).

Теперь нужно «изготовить» закрытый ключ (ключ расшифрования), т.е. найти число d такое, что e·d≡1 (mod 20). Очевидно, что d=3, так как 7·3 = 21 mod 20 =1. Предположим, что нужно зашифровать m=2. Тогда c = (m^e) mod n = 2^7 mod 33 = 128 mod 33 = 29. Итак, криптограммой сообщения m=2 является c=29. Дешифрование: (c^d)mod n=(29^3) mod 33 = $(-4)^3$ mod 33=(-64)mod 33=(-31)mod 33=2=m.

Задание.

При p=3, q=11, e=7 зашифруйте сообщение m=3, сообшение m=4.

В условиях предыдущей задачи расшифруйте криптограмму c=5.

12.2 **Алгоритм RSA**

Вычисление ключей

Важным моментом в криптоалгоритме RSA является создание пары ключей: *открытого* и *закрытого*.

Для алгоритма RSA *этап создания ключей* состоит из следующих операций:

1) выбираются два различных случайных простых числа p и q.

2) вычисляется их произведение n = p · q (называемое модулем RSA);

Замечание. Под простым числом будем понимать такое число, которое делится только на 1 и на само себя. Взаимно простыми числами будем называть такие числа, которые не имеют ни одного общего делителя, кроме единицы.

3) вычисляется функция Эйлера φ(n) = (p - 1) · (q - 1);

Замечание. Функция Эйлера показывает количество целых положительных чисел от 1 до n, которые взаимно просты с n.

4) выбирается целое число e (открытая экспонента), взаимно простое со значением функции φ(n), удовлетворяющее неравенству 1 < e < φ(n).

При реальном шифровании длина e выбирается приблизительно равной L/3, где L – длина n. Можно взять e равным произвольному простому числу, меньшему n и отличному от p и q, проверив при этом условие того, что φ(n) не делится на e ($\varphi(N) \bmod e \neq 0$).

5) вычисляется число d (секретная экспонента), удовлетворяющее условию d · e = 1 mod φ(n);

Замечание. Вычисляется d расширенным алгоритмом Евклида таким образом, что ($e \cdot d - 1$) делится на ($p - 1$) · ($q - 1$). Основным равенством, используемым для вычисления НОД чисел A и B, является условие: НОД(A, B) = НОД(B, A mod B).

6) числа e, n публикуются, то есть являются открытым ключом RSA;

7) числа d, n являются закрытым ключом RSA.

№ п/п	Описание операции	Пример
1	Выбираются два простых числа[1] **p** и **q**.	p=7, q=13
2	Вычисляется произведение **n** = p * q.	n=91
3	Вычисляется **функция Эйлера[2] φ(n)**.	φ(n)=(7-1)(13-1)= 91-7-13+1 = 72
4	Выбирается открытый ключ **e**, как произвольное число (0<e<n), взаимно простое[3] с результатом функции Эйлера (e ⊥ φ(n)).	e=5
5	Вычисляется секретный ключ **d**, как обратное число[4] к **e** по модулю **φ(n)**, из соотношения (d*e) mod φ(n) = 1.	(d*5) mod 72 = 1, d = 29
6	Публикуются открытый ключ (**e**, **n**) в специальном хранилище, где исключается возможность его подмены (**общедоступном сертифицированном справочнике**).	

***Шифрование** RSA* с помощью пары чисел производится следующим образом:

1. Отправитель разбивает своё сообщение M на блоки m_i. Значение $m_i < n$, поэтому длина блока m_i в битах не больше $k = [\log_2(n)]$ бит, где квадратные скобки обозначают, взятие целой части от дробного числа. Например, если $n = 21$, то максимальная длина блока $k = [\log_2(21)] = [4.39\ldots] = 4$ бита.
2. Подобный блок может быть интерпретирован как число из диапазона $(0; 2^k - 1)$.
Для каждого такого числа m_i вычисляется выражение (c_i – зашифрованное сообщение):
$c_i = ((m_i)^e) \bmod n$.
Заметим: Необходимо добавлять нулевые биты слева в двоичное представление блока до размера $k = [\log_2(n)]$ бит.

Итак, Алгоритм *шифрования* RSA:1) каждый символ исходного открытого текста преобразуется в число с использованием стандартной кодировки, то есть формируется сообщение, обозначим его x; 2) с использованием открытого ключа (открытой экспоненты) вычисляется передаваемый шифротекст y(x) = x^^e mod n (см. Приложение).

Алгоритм ***расшифрования*** RSA: Чтобы получить открытый текст, необходимо каждый блок дешифровать отдельно: $m_i = ((c_i)^d) \bmod n$, те:

1) с использованием закрытого ключа (секретной экспоненты) вычисляется x'= $y(x)^d$ mod m.

2) полученное значение x' преобразуется в текст с помощью стандартной кодировки.

Стойкость шифра RSA обосновывается следующими соображениями. Для того чтобы прочитать криптограмму c, нужно знать закрытый ключ d. Поскольку числа e и n=p·q известны, для нахождения d достаточно найти произведение (p-1)(q-1), так как e·d≡1 (mod (p-1) · (q-1)), Таким образом, все сводится к определению множителей p и q числа n. Как уже было отмечено выше, задача разложения на множители для больших составных чисел в настоящее время вычислительно не разрешима.

Все шифры, которые рассматривались до настоящего момента, обладают тем свойством, что для шифрования и дешифрования в них применяется один и тот же секретный ключ. Поэтому такие шифры называют *симметричными*. Шифр RSA этим свойством не обладает, процедуры шифрование и дешифрование в нем осуществляются на разных ключах. Подобные шифры называются *асимметричными*.

Для коротких сообщений шифр RSA почти идеален, но при передаче информации большого объема он сильно уступает по скорости симметричным алгоритмам шифрования. Так, самые быстрые микросхемы для RSA имеют пропускную способность около 65 Кбит/с, в то время, как скорость реализации, например AES, достигает 70 Мбит/с. Поэтому в коммуникационных сетях с большой нагрузкой рекомендуется применять RSA вместе с AES (по протоколу «цифровой конверт»): абонент A, желая установить защищенную связь с абонентом B, посылает ему по открытому каналу секретный AES-ключ K, зашифрованный по методу RSA; абонент B расшифровывает полученную криптограмму, используя свой закрытый RSA-ключ, и теперь может приступить к скоростному обмену информацией с A, применяя шифрование по методу AES на ключе K.

Пример 2:
Выберем два простых числа: $p = 7$, $q = 17$.
Вычислим $n = p \cdot q = 7 \cdot 17 = 119$.
Вычислим $\varphi(n) = (p - 1) \cdot (q - 1) = 96$.

Выберем e так, чтобы e было взаимнопростым с $\varphi(n) = 96$ и меньше, чем $\varphi(n)$: $e = 5$.
Определим d так, чтобы $d \cdot e \equiv 1 \bmod 96$ и $d < 96$, $d = 77$, так как $77 \cdot 5 = 385 = 4 \cdot 96 + 1$.
Результирующие ключи: открытый {5, 119} и закрытый ключ {77, 119}.
Например, требуется зашифровать сообщение $M = 19$: $19^5 = 66 \pmod{119}$, те $C = = 66$.
Для дешифрования вычисляется $66^{77} \pmod{119} = 19$.

Пример 3. Авторы RSA при описании принципов функционирования своей системы выбрали в качестве исходного текста фразу «ITS ALL GREEC TO ME» («Для меня все это совершенно непонятно»). Для того чтобы преобразовать этот текст в одно большое число, авторы закодировали пробел между словами 0, букву A – 1, букву B – 2, букву C – 3, ... , букву Z – 26. На представление каждого символа выделили пять двоичных разрядов.

В результате приведенной фразе стало соответствовать число m = 09201900011212000718050511002015001305.

Для шифрования авторы выбрали e = 9007 и n = 114381625757888867669235779976146612010218296721242362562651842935706935245733897830597123563958705058989075147599290026879543541.

После зашифрования получили число c = m^e mod n = 199935131497805100452317122740260647423204017058391463103703717406259716089489275043992096267258267501289355446135382376974802 6.

Число n было произведением 64-значного и 65-значного простых чисел p и q, выбранных случайным образом.

Пример 4. Зашифруем сообщение “CAB”.
1. Выберем p=3 и q=11.
2. Определим n=3*11=33.
3. Найдем φ(n)=(p-1)(q-1)=20.
4. Выберем в качестве d, число взаимно простое с φ(n)=20, например, d = 3.
Взаимно простые числа делятся только на 1 и на само себя. 5.
Выберем число e. В качестве такого числа может быть взято любое число, для которого удовлетворяется соотношение (e×3) (mod 20) = 1, например, e=7.
6. Представим шифруемое сообщение как последовательность целых чисел с помощью отображения: A→1, B→2, C→3. Тогда сообщение принимает вид (3, 1, 2). Зашифруем сообщение с помощью ключа {7,33}.
$ШТ_1 = (3^7) \pmod{33} = 2187 \pmod{33} = 9$,
$ШТ_2 = (1^7) \pmod{33} = 1 \pmod{33} = 1$,
$ШТ_3 = (2^7) \pmod{33} = 128 \pmod{33} = 29$.
7. Расшифруем полученное зашифрованное сообщение (9, 1, 29) на основе закрытого ключа {3,33}:
$ИТ_1 = (9^3) \pmod{33} = 729 \pmod{33} = 3$,
$ИТ_2 = (1^3) \pmod{33} = 1 \pmod{33} = 1$,
$ИТ_3 = (2^{93}) \pmod{33} = 24389 \pmod{33} = 2$.
Здесь ШТ – шифротекст, ИТ – исходный текст.

Итак, в реальных системах алгоритм RSA реализуется следующим образом: каждый пользователь выбирает два больших простых числа, и в соответствии с описанным выше алгоритмом выбирает два простых числа e и d. Как результат умножения первых двух чисел (p и q) устанавливается n. {e, n} образует открытый ключ, а {d, n} - закрытый (хотя можно взять и наоборот).

Открытый ключ публикуется и доступен каждому, кто желает послать владельцу ключа сообщение, которое зашифровывается указанным алгоритмом. После шифрования, сообщение невозможно раскрыть с помощью открытого ключа. Владелец же закрытого ключа без труда может расшифровать принятое сообщение.

Общий вид криптосистемы RSA приведен на рис. 5.1.

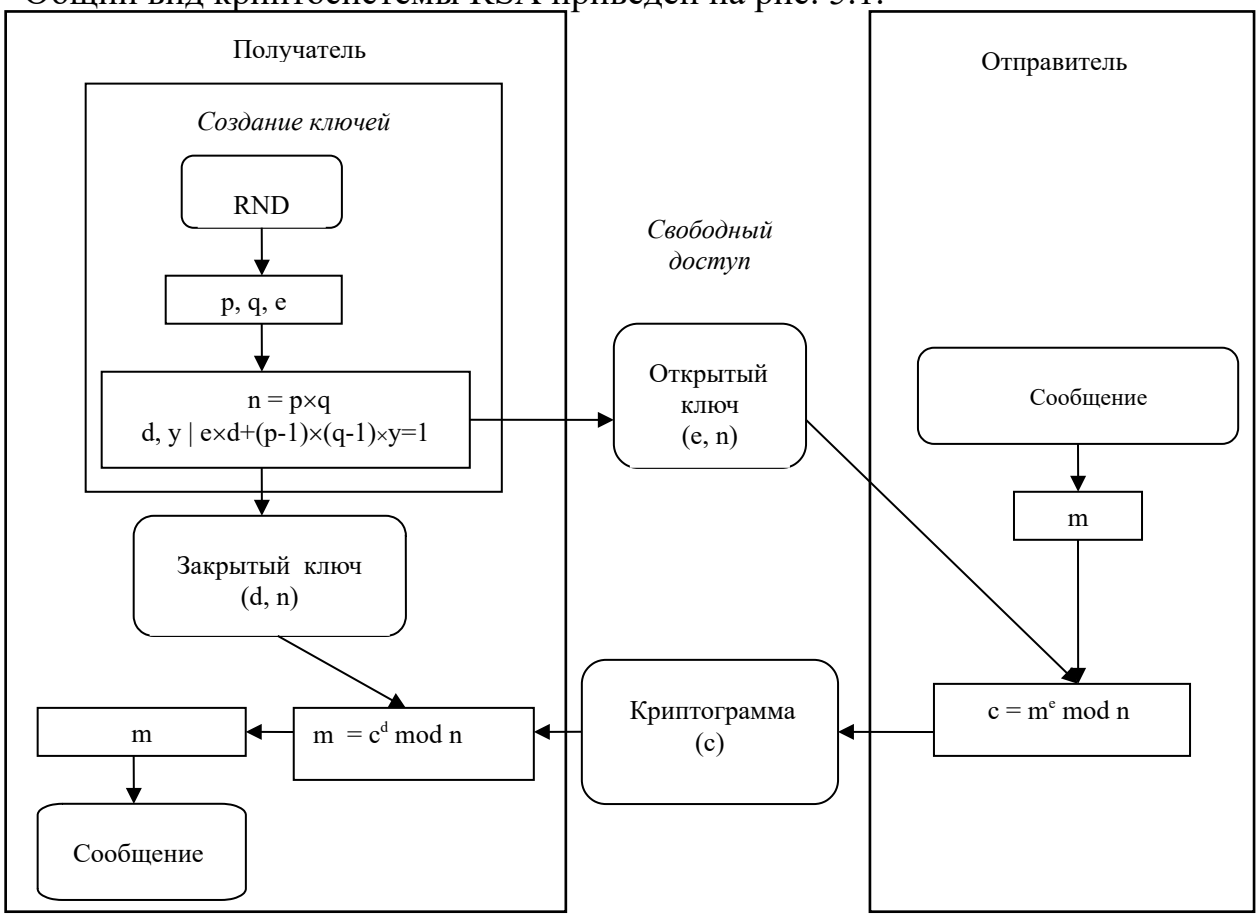

Рис. 3.2. Криптосистема RSA

12.3. ***Задания*** на практическую работу

1) Используя заданные в соответствии с вариантом значения p, q и закрытого ключа K_c вычислить открытый ключ K_o при помощи расширенного алгоритма Евклида и выполнить шифрование по алгоритму RSA открытым ключом (K_o) своей фамилии, имени и отчества. Для представления букв в числовой форме использовать следующее соответствие: ‘А’ – 2, ‘Б’ – 3, ‘В’ – 4, ..., ‘Ё’ – 8, ..., ‘Я’ – 34. 2.3. Выполнить проверку правильности расшифрования полученных зашифрованных данных при помощи закрытого ключа K_c. Проверить себя можно будет впоследствии, расшифровав сообщение с помощью разработанного Вами программного продукта.

Варианты заданий

1. p=5, q=7, Kc=11.	2. p=17, q=5, Kc=7.	3. p=11, q=5, Kc=13.	4. p=7, q=11, Kc=19.
5. p=7, q=17, Kc=5.	6. p=3, q=17, Kc=23.	7. p=13, q=3, Kc=11.	8. p=5, q=13, Kc=19.
9. p=7, q=13, Kc=17.	10. p=3, q=19, Kc=7.	11. p=5, q=7, Kc=23.	12. p=17, q=5, Kc=19.
13. p=11, q=5, Kc=17.	14. p=7, q=11, Kc=13.	15. p=7, q=17, Kc=11.	16. p=3, q=17, Kc=13.
17. p=13, q=3, Kc=17.	18. p=5, q=13, Kc=11.	19. p=7, q=13, Kc=19.	20. p=3, q=19, Kc=13.
21. p=5, q=7, Kc=19.	22. p=17, q=5, Kc=23.	23. p=11, q=5, Kc=19.	24. p=7, q=11, Kc=17.
25. p=7, q=17, Kc=23.	26. p=3, q=17, Kc=11.	27. p=13, q=3, Kc=19.	28. p=5, q=13, Kc=17.
29. p=7, q=13, Kc=23	30. p=3, q=19, Kc=11.		

2) Разработать программу для шифрования/дешифрования по алгоритму RSA текстов. Программа должна уметь работать с текстом произвольной длины.

Замечание. На «отлично» необходимо чтобы программа выполняла шифрование данных как с файла, так и с текстового окна программы. На «хорошо» – программа должна выполнять шифрование только с файла. На «удовлетворительно» – программа должна выполнять шифрование только с текстового окна.

3) Сообщение из текстового файла зашифровать и расшифровать с помощью разработанного Вами программного продукта с использованием чисел p и q из табл. 5.1 и с использованием функции автоматической генерации ключей (на выбор).

Таблица 5.1. Пары простых чисел

1. p=13, q=17	2. p=23, q=13	3. p=19, q=11	4. p=17, q=23	5. p=19, q=13
6. p=11, q=29	7. p=19, q=23	8. p=11, q=23	9. p=11, q=17	10. p=13, q=29
11. p=17, q=19	12. p=7, q=23	13. p=7, q=29	14. p=7, q=19	15. p=7, q=31
16. p=7, q=37	17. p=5, q=31	18. p=5, q=37	19. p=5, q=29	20. p=29, q=11
21. p=23, q=19	22. p=17, q=13	23. p=19, q=17	24. p=13, q=23	25. p=23, q=11
26. p=29, q=13	27. p=23, q=7	28. p=31, q=7	29. p=29, q=17	30. p=23, q=29

Примечания.

Способы расчета функции Эйлера

Случай	Формула	Пример (число / расчетная формула / список взаимно простых чисел)
n простое число	$\varphi(n) = n - 1$	n = 7 $\varphi(7) = 7 - 1 = 6$ {1, 2, 3, 4, 5, 6}
n = p q произведение двух простых чисел	$\varphi(n) = \varphi(p)\,\varphi(q) =$ $= (p - 1)(q - 1) = n - p - q + 1$ (за исключением случая p = q = 2)	n = 15 = 3 * 5 $\varphi(15) = \varphi(3)\,\varphi(5) = (3 - 1)(5 - 1) = 15 - 3 - 5 + 1 = 8$ {1, 2, 4, 7, 8, 11, 13, 14}
$n = p^q$ простое число в степени	$\varphi(n) = p^q - p^{q-1}$	n = 9 = 3^2 $\varphi(9) = 3^2 - 3^{2-1} = 9 - 3 = 6$ {1, 2, 4, 5, 7, 8}
$n = p_1^{q1}\,p_2^{q1} \dots p_k^{qk}$ разложение числа согласно основной теоремы арифметики (общий случай)	$\varphi(n) = \varphi(p_1^{q1})\,\varphi(p_2^{q2}) \dots \varphi(p_k^{qk}) =$ $= p_1^{q1}\left(1-\frac{1}{p_1}\right) p_2^{q2}\left(1-\frac{1}{p_2}\right) \dots p_k^{qk}\left(1-\frac{1}{p_k}\right) =$ $= n\left(1-\frac{1}{p_1}\right)\left(1-\frac{1}{p_2}\right) \dots \left(1-\frac{1}{p_k}\right)$	n = 84 = $2^2\,3^1\,7^1$ $\varphi(84) = 84\left(1-\frac{1}{2}\right)\left(1-\frac{1}{3}\right)\left(1-\frac{1}{7}\right) = 24$ {1, 5, 11, 13, 17, 19, 23, 25, 29, 31, 37, 41, 43, 47, 53, 55, 59, 61, 65, 67, 71, 73, 79, 83}

АУТЕНТИФИКАЦИЯ. ЭЛЕКТРОННАЯ ЦИФРОВАЯ ПОДПИСЬ.

Идентификация – это назначение объекту системы уникальной условной метки, которая позволяет однозначно определить этот объект. Под аутентификацией понимается проверка подлинности объекта, предъявившего данный идентификатор. Аутентификация основана на информации, которая может быть известна только истинному пользователю системы.

Пусть в коммуникационной сети, снабженной системой шифрования RSA, абонент A желает распространить открытое сообщение m и подтвердить свое авторство. Всем пользователям сети доступен открытый ключ абонента A – пара чисел (n,e). Кроме того, A держит в секрете свой закрытый ключ d – единственное число, вместе с e и $n=p\cdot q$ удовлетворяющее сравнению $e\cdot d\equiv 1 (\mathrm{mod}\ (p-1)\cdot(q-1))$. Для осуществления своей задачи A представляет m в числовом виде, пусть окажется $m < n$, и вычисляет $s=(m^d) \mathrm{mod}\ n$ – это его цифровая подпись. Затем он рассылает по сети пару чисел (m, s). Абонент B, прочитав m и желая убедиться в том, что приславший сообщение на самом деле тот, за кого он себя выдает, извлекает из RSA-справочника сети принадлежащий A открытый ключ (n, e) и находит с его помощью число $(s^e) \mathrm{mod}\ n$. Если полученное число совпадает с m, проверяющий убеждается в том, что целостность исходного сообщения не нарушена, т.е. в процессе передачи оно не было изменено, и что приславший это сообщение знает закрытый ключ, связанный с открытым ключом абонента A, т.е. это и есть A.

ПРИЛОЖЕНИЕ для ПЗ Алгоритм RSA

Алгоритм возведения в степень по модулю натурального числа.

Для выполнения шифрования по методу RSA приходится выполнять возведение в большую степень различных чисел, а результат приводить по модулю числа N, являющегося параметром метода и имеющего длину 512 и более бит. Уже для небольших ***a*** и ***e*** вычислить значение $c = a^e \bmod N$ (1)

выполняя сначала возведение в степень, а потом вычисляя остаток от деления a^e на N, становится невозможным. Между тем, если применить алгоритм, описанный в этом разделе, можно вычислять выражения (1), для достаточно больших чисел a, e, N, оставаясь в рамках обычных операций с целыми числами, заложенных в компьютере.

Алгоритм быстрого возведения в степень основывается на идее замены прямого вычисления возведения в степень a^e последовательными операциями умножения на a и возведения в квадрат. Для этого представим степень e число в двоичном исчислении

$$e = t_0 t_1 ... t_k \qquad (2)$$

где любое t_i для $0 \le i \le k$ принадлежит $\{0,1\}$, $t_0 = 1$. Зная вектор разрядов $(t_0, t_1 ... t_k)$, можно вычислить число *e*, применяя последовательные вычисления:

$$e_0 = t_0 = 1,\ e_{i+1} = 2 \cdot e_i + t_{i+1},\ i = 0, 1, \ldots, k-1 \qquad (3)$$

Например, если *e* = 13, то в двоичном представлении $e = 1101_2$, и 13 можно представить как результат вычисления $e_0 = 1,\ e_1 = 2 \cdot e_0 + t_1 = 2 + 1 = 3,\ e_2 = 2 \cdot e_1 + t_2 = 6 + 0 = 6$,

$e_3 = 2 \cdot e_2 + t_3 = 12 + 1 = 13$.

Последнее число и есть *e*.

Используя формулы (3), можно процедуру возведения в степень по модулю натурального числа N, записать в виде последовательности итераций:

$c_0 = a, c_{i+1} = c_i^{\,2} \cdot a^{t_{i+1}} \bmod N,$ (4)

где $i = 0, 1, \ldots, k-1$. Множитель $a^{t_{i+1}}$ в зависимости от t_{i+1} принимает либо значение *a,* если $t_{i+1} = 1$, либо 1, если $t_{i+1} = 0$. Результат вычислений можно свести в таблицу

$t_0 = 1$	t_1	t_2	...	t_k
$c_0 = a$	c_1	c_2	...	c_k

Пример. Вычислить $c = 5^{13} \bmod 19$.

Решение. Переведем степень e=13 в двоичный вид. Для этого заполним следующую таблицу:

Таблица 1. Перевод десятичного числа e к двоичному представлению

e div 2	13	6	3	1
e mod 2	1	0	1	1

Искомое двоичное разложение числа *e* будет во второй строке таблицы, записанное в обратном порядке справа налево.

Далее, составим таблицу вычисления c, заполняя следующую таблицу:

Таблица 2. Возведение a=5 в степень e=13 по модулю 19

e	1	1	0	1
c	5	11	7	10

В первой строке запишем цифры двоичное разложения числа 13. В первую ячейку второй строки поместим основание *a*=5. Далее каждое следующее значение *c* будем вычислять по формуле:

$$c_{i+1} = \begin{cases} c_i^{\,2} \cdot a \bmod N, \textit{если } e_{i+1} = 1 \\ c_i^{\,2} \bmod N, \textit{если } e_{i+1} = 0 \end{cases}$$

Например,

$$c_2 = 5^2 \cdot 5 \bmod 19 = 125 \bmod 19 = 11$$

$$c_3 = 11^2 \bmod 19 = 121 \bmod 19 = 7$$

$$c_4 = 7^2 \cdot 5 \bmod 19 = 245 \bmod 19 = 17$$

Практическая работа №13. **Вероятностное шифрование. Шифросистема Эль-Гамаля**

Вероятностное шифрование является разновидностью криптосистем с открытым ключом (авторы – Шафи Гольдвассер (Shafi Goldwasser) и Сильвио Микали (Silvio Micali)). Данный вид шифрования относят к допускающим неоднозначное вскрытие. Основной целью вероятностного шифрования является устранение утечки информации в криптографии с открытым ключом. Поскольку криптоаналитик всегда может зашифровать случайные сообщения открытым ключом, он может получить некоторую информацию. При условии, что у него есть шифртекст C ($C = E_{k1}(T)$) и он пытается получить открытый текст T, он может выбрать случайное сообщение T' и зашифровать: $C' = E_{k1}(T')$. Если C' = C, то он угадал правильный открытый текст. В противном случае он делает следующую попытку.

Вероятностное шифрование такую попытку атаки шифртекста делает бессмысленной. Другими словами, при шифровании с помощью одного и того же открытого ключа можно получить разные шифртексты, которые при расшифровке дают один и тот же открытый текст.

$C_1 = E_{k1}(T)$, $C_2 = E_{k1}(T)$, $C_3 = E_{k1}(T)$, ..., $C_N = E_{k1}(T)$, (9.4)

$T = D_{k2}(C_1) = D_{k2}(C_2) = D_{k2}(C_3) = ... = D_{k2}(C_N)$. (9.5)

В результате, даже если у криптоаналитика имеется шифртекст Ci и он угадает T, то в результате операции шифрования получиться $C_j = E_{k1}(T)$. Вероятность того, что $C_i = C_j$ крайне низка. Таким образом, криптоаналитик даже не узнает, была ли правильной его догадка относительно T или нет.

Ниже рассматриваются два алгоритма вероятностного шифрования:

- алгоритм шифрования Эль-Гамаля;
- алгоритм на основе эллиптических кривых.

Практическая работа №13. **Шифросистема Эль-Гамаля**

6.1. Краткие теоретические сведения

Асимметричная схема была предложена Тахером Эль-Гамалем (El Gamal) в 1984 г (Taher ElGamal, «A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms», IEEE Transactions on Information Theory, v. IT-31, n. 4, 1985, pp. 469–472). Он усовершенствовал систему Диффи-Хеллмана и получил два алгоритма, которые использовались для шифрования и обеспечения аутентификации. Стойкость данного алгоритма базируется на сложности решения задачи *дискретного логарифмирования*.

Суть задачи заключается в следующем. Имеется уравнение $g^x \bmod p = y$. Требуется по известным g, y и p найти целое неотрицательное число x (дискретный логарифм).

Асимметричная схема Эль Гамаль использует операцию возведения в степень по модулю простого числа. При этом трудноразрешимой задачей для злоумышленника является отыскание не числа, которое возведено в степень, а то, в какую степень возведено известное число. Эта задача носит название проблемы дискретного логарифма.

Алгоритм *формирования ключей* системы Эль-Гамаля:

1) для всей группы абонентов выбирается некоторое большое *простое* число **p** и число **g** из условия, что различные степени числа **g** суть различные числа по модулю **p**. Простое число **p** выбирается таким, чтобы выполнялось равенство $\mathbf{p=2 \cdot q+1}$, где **q** - тоже простое число. Тогда в качестве числа g можно использовать любое число, удовлетворяющее неравенствам $\mathbf{1 < g < p-1}$ и $\mathbf{(g^q) \bmod p <> 1}$;

2) числа **p** и **g** публикуются (передаются всем абонентам);

3) каждый абонент выбирает секретное число **x**, удовлетворяющее неравенству $\mathbf{1 < x < p-1}$ и вычисляет открытое число $\mathbf{y = g \bmod p}$;

4) получен открытый ключ (**p**, **g**, **y**) и закрытый ключ **x**.

№ п/п	Описание операции	Пример
1	Выбирается простое число **p**.	p=37
2	Выбирается число **g**, являющееся первообразным корнем по модулю **p** и меньшее **p**.	g=2
3	Выбираются произвольное число **x**, меньшее **p**.	x=5
4	Вычисляется **y** = g^x mod p	y = 2^5 mod 37 = 32 mod 37 = 32
5	Открытый ключ - **y, g** и **p**. Причем g и p можно сделать общими для группы пользователей. Закрытый ключ - **x**.	

Алгоритм *шифрования* Эль-Гамаля:

1) исходный открытый текст преобразуется в число с использованием стандартной кодировки, то есть формируется сообщение, обозначим его M, при этом M<p;

2) выбирается сессионный ключ k, удовлетворяющий неравенству 1 < k <p -2;

3) вычисляется пара чисел, **r=(g^k) mod p** и **e=(M·$y^к$) mod p** являющихся шифротекстом.

Алгоритм *расшифрования* Эль-Гамаля:

1) вычисляется **M' = (e · $r^{(p-1-x)}$) mod p**.

2) полученное значение M’ преобразуется в текст с помощью стандартной кодировки.

Пример 1. В качестве простого числа, порождающего циклическую группу, выберем p = 11, за образующий элемент примем число a = 7 (при возведении 7 в степень 1, 2, 3 и т. д. по модулю 11 последовательно проходят все 10 значений [7, 5, 2, 3, 10, 4, 6, 9, 8, 1]). Секретным ключом x выберем 6, параметр b принимает значение b = (a^x mod p) = (7^6 mod 11) = 4. В целом ключ принимает вид (a = 7, p = 11, b = 4).

Предположим, что некий абонент хочет передать сообщение. Он выбрал случайное число, не превосходящее p, например, y = 9. В начало шифрограммы помещается число (a^y mod p) = 7^9 mod 11) = 8. Кроме того, на основе y и открытого ключа отправитель вычисляет k = b^y mod p = 4^9 mod 11 = 3. Выбрав значение 3 или какие-либо его биты в качестве симметричного ключа, отправитель шифрует передаваемые данные и стирает величины 9 и 3 со своих накопителей.

Получатель по приходу пакета для вычисления k = (a^y mod p)^x mod p возводит число 8 из заголовка шифрограммы в степень секретного ключа и получает k = 8^6 mod 11 = 3 – то же самое значение, которое использовал отправитель, шифруя собственно данные.

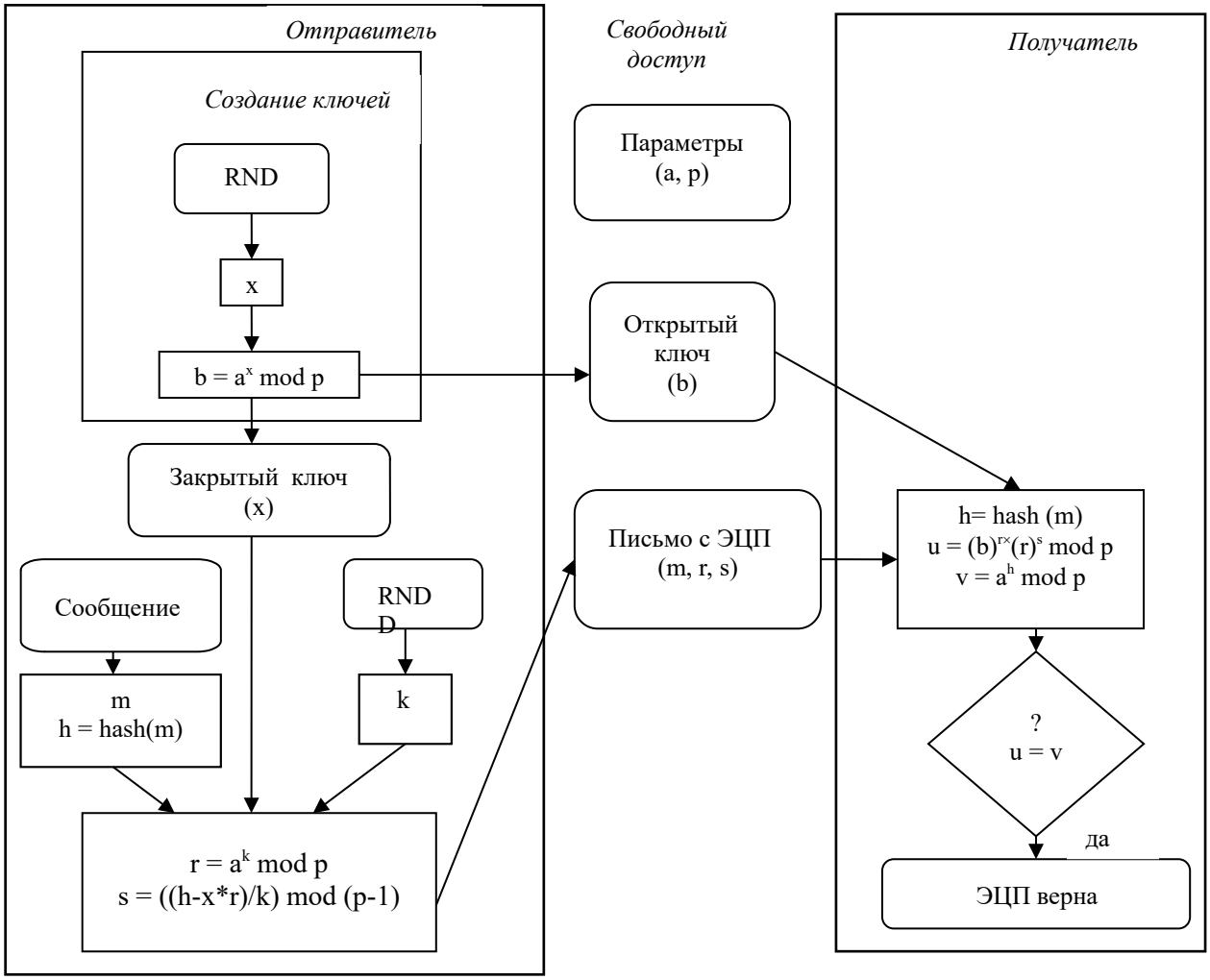

Рис. 5.3. ЭЦП Эль Гамаля

13.2. ***Задание*** на практическую работу

1) Разработать программу для шифрования/дешифрования по шифросистеме Эль-Гамаля текстов. Программа должна уметь работать с текстом произвольной длины.

Замечание. На «отлично» необходимо чтобы программа выполняла шифрование данных как с файла, так и с текстового окна программы. На «хорошо» – программа должна выполнять шифрование только с файла. На «удовлетворительно» – программа должна выполнять шифрование только с текстового окна.

2) С помощью соответствующего разработанного Вами программного продукта зашифровать сообщение, представляющее собой первые буквы своих фамилии, имени и отчества, используя ключи длиной 32, 64 и 256 бит (используется автоматическая генерация ключей).

3) Выполнить ту же операцию вручную для произвольного небольшого значения p. Проверить себя, расшифровав сообщение с помощью соответствующего программного продукта.

13.3. ***Контрольные*** вопросы

1. Какая процедура является более производительной – асимметричное шифрование/ дешифрование или симметричное шифрование/дешифрование?
2. К какому типу криптоалгоритма (с точки зрения его устойчивости к взлому) и почему относится алгоритм RSA?
3. Какая трудноразрешимая математическая задача лежит в основе стойкости алгоритма RSA?
4. Какая трудноразрешимая математическая задача лежит в основе стойкости алгоритма Эль- Гамаль?
5. В чем заключается проблема дискретного логарифма?
6. В чем заключаются проблемы разложения больших чисел на простые множители и вычисления корней алгебраических уравнений?

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА:

1. С.Г.Баричев, В.В.Гончаров, Р.Е.Серов. Основы современной криптографии – Москва, Горячая линия – Телеком, 2001
2. А.В.Беляев. Методы и средства защиты информации (курс лекций). [Электронная версия] http://www.citforum.ru/internet/infsecure/index.shtml
3. А. А. Болотов, С. Б. Гашков, А. Б. Фролов, А. А. Часовских. Алгоритмические основы эллиптической криптографии. Учебное пособие – М.: Изд-во МЭИ. 2000 г., 100 с.
4. Т.Илонен. Введение в криптографию (Ylonen Tatu. Introduction to Cryptography), [Электронная версия] http://www.ssl.stu.neva.ru/psw/crypto/intro.html
5. Ш.Т. Ишмухаметов. Технологии защиты информации в сети – Казань, 2008, 91 с. [Электронная версия] http://depositfiles.com/files/e9zxcqos9
6. Н.Коблиц. Теория чисел и криптография – М.:, ТВР, 2001 [Электронная версия] http://gabro.ge/biblio/0708/0081/file/Cryptography/Koblic_-_Teoriya_Chisel_i_Cryptografiya.rar
7. О.Р. Лапонина. Криптографические основы безопасности, курс Интернет-университета [Электронная версия] http://www.intuit.ru/department/security/networksec
8. Р.Лидл, Г.Нидеррайтер. Конечные поля, в 2 т., пер.с англ. – М.: Мир, 1998, 438 с.
9. А.А. Молдовян, Н.А. Молдовян, Б.Я. Советов. Криптография – М., Лань, 2001
10. А.А. Молдовян, Н.А. Молдовян, Введение в криптосистемы с открытым ключом – БХВ-Петербург, 2005, с. 286 [Электронная версия] http://cyberdoc.nnm.ru/vvedenie_v_kriptosistemy_s_otkrytym_klyuchom
11. А.Г.Ростовцев. Алгебраические основы криптографии – СПб, Мир и Семья, 2000.
12. А.Г.Ростовцев, Е.Б.Маховенко. Теоретическая криптография – СПб.: АНО, ПО “Профессионал”, 2005, [Электронная версия] http://bookpedia.ru/index.php?newsid=1265
13. Г.Семенов. Цифровая подпись. Эллиптические кривые. [Электронная версия] http://www.morepc.ru/security/crypt/os200207010.html?print
14. В.Столлингс. Основы защиты сетей. Приложения и стандарты – М.: Вильямс, 2002, 429 с.
15. Брюс Шнайер. Прикладная криптография, 2-е издание: протоколы, алгоритмы и исходные тексты на языке С, [Электронная версия] http://www.ssl.stu.neva.ru/psw/crypto/appl_rus/appl_cryp.htm
16. Dr. Michael Ganley, Thales eSecurity Ltd. Метод эллиптических кривых, [Электронная версия] http://www.racal.ru/rsp/eliptic_curve_cryptography.htm
17. В.М.Фомичев. Дискретная математика и криптология – Диалог-МИФИ, 2003, 399 с.
18. Сайт Криптографический ликбез [Электронная версия] http://www.ssl.stu.neva.ru/psw/crypto.html