Криптографические алгоритмы с открытым ключом и их использование

Лекция 11

Володько Ольга Станиславовна

Алгоритм RSA

Основные сведения

Алгоритм шифрования с открытым ключом **RSA** был предложен одним из первых в конце 70-х годов XX века. Его название составлено из первых букв фамилий авторов: Р.Райвеста (R.Rivest), А.Шамира (A.Shamir) и Л. Адлемана (L.Adleman). Алгоритм RSA является, наверно, наиболее популярным и широко применяемым *асимметричным алгоритмом* в криптографических системах.

Алгоритм основан на использовании того факта, что задача разложения большого числа на простые сомножители является трудной.

Алгоритм RSA

Криптографическая система RSA базируется на следующих двух фактах из теории чисел:

- задача проверки числа на простоту является сравнительно легкой;
- задача разложения чисел вида n = pq (p и q — простые числа); на множители является очень трудной, если мы знаем только n, а p и q — большие числа (это так называемая задача факторизации).

Алгоритм RSA представляет собой блочный алгоритм шифрования, где зашифрованные и незашифрованные данные должны быть представлены в виде целых чисел между 0 и n -1 для некоторого n.

Шифрование

Итак, рассмотрим сам алгоритм.

Пусть абонент А хочет передать зашифрованное сообщение абоненту Б.

В этом случае абонент Б должен подготовить пару (открытый ключ; закрытый ключ) и отправить свой открытый ключ пользователю А.

Первым этапом является генерация открытого и закрытого ключей.

Для этого вначале выбираются два больших простых числа `P` и `Q`.

Затем вычисляется произведение `N`:

$$N = PQ.$$

После этого определяется вспомогательное число `f`:

$$f = (P - l)(Q - 1).$$

Шифрование

Затем случайным образом выбирается число d < f и взаимно простое с f.

Далее необходимо найти число e, такое, что

$$ed \bmod f = 1.$$

Числа d и N будут открытым ключом пользователя, а значение e – закрытым ключом.
Таким образом, на этом этапе у пользователя должна быть информация, указанная в следующей таблице:

	Открытый ключ	Закрытый ключ
Пользователь системы	N, d	e

Шифрование

Так как пользователь Б хочет получить зашифрованное сообщение от пользователя А, значит пользователь Б должен отправить свой открытый ключ (d, N) пользователю А.

Числа P и Q больше не нужны, однако их нельзя никому сообщать; лучше всего их вообще забыть.

На этом этап подготовки ключей закончен и можно использовать основной протокол RSA для шифрования данных.

Шифрование

Второй этап – шифрование данных. Если абонент А хочет передать некоторые данные абоненту Б, он должен представить свое сообщение в цифровом виде и разбить его на блоки m_1, m_2, m_3, … , где $m_i < N$. Зашифрованное сообщение будет состоять из блоков c_i.

Абонент А шифрует каждый блок своего сообщения по формуле

$$c_i = m_i^d \bmod N$$

используя открытые параметры пользователя Б, и пересылает зашифрованное сообщение C=(c1, c2, c3, …) по открытой линии.

Шифрование

Абонент Б, получивший зашифрованное сообщение, расшифровывает все блоки полученного сообщения по формуле

$$m_i = c^e \bmod N$$

Все расшифрованные блоки будут точно такими же, как и исходящие от пользователя А.

Злоумышленник, перехватывающий все сообщения и знающий всю открытую информацию, не сможет найти исходное сообщение при больших значениях P и Q.

Пример вычислений по алгоритму

Пусть пользователь А хочет передать пользователю Б сообщение. В этом случае вначале пользователь Б должен подготовить открытый и закрытый ключи. Пусть им выбраны, например, следующие параметры:

P = 3, Q = 11, N = 3x11 = 33.

Тогда `f = (P - 1)(Q - 1) = (3-1)(11-1) = 20`.

Затем пользователь Б выбирает любое число d, не имеющее общих делителей с f (это необходимо для того, чтобы зашифрованное сообщение можно было потом однозначно восстановить).

Пример вычислений по алгоритму

Пусть `d = 13`. Это число будет одним из компонентов открытого ключа.

Далее необходимо найти число `e`, которое можно будет использовать в качестве закрытого ключа для расшифрования сообщения. Значение `e` должно удовлетворять соотношению

$$ed \bmod f = 1.$$

Для малых значений f число e можно найти подбором. В нашем случае подходит e=17. (Проверяем: 13*17 mod 20 = 221 mod 20 = 1.)

Пример вычислений по алгоритму

Теперь пользователь Б должен запомнить свой закрытый ключ 17, отправить открытый ключ (13, 33) пользователю А и уничтожить числа P = 3 и Q = 11.

Пользователь А, получивший открытый ключ (13, 33), увидев, что N=33, разбивает исходное сообщение на три блока, причем значение каждого меньше N. Например, пусть имеется три блока m_1=8, m_2=27, m_3,=5. Затем пользователь А шифрует каждый блок:

$c_1=8^{13}$ m26 od 33 = 17 $c_2 = 27^{13}$ mod 33 = 15 $c_3 = 5^{13}$ mod 33 =

Зашифрованное сообщение, состоящее из трех блоков (17, 15, 26), передается пользователю Б, который, используя свой закрытый ключ e = 17 и N=33, расшифровывает сообщение:

$$m_1 = 17^{17} \bmod 33 = 8 \quad m_2 = 15^{17} \bmod 33 = 27 \quad m_3 = 26^{17} \bmod 33 = 5$$

Таким образом, абонент Б расшифровал сообщение от абонента А.

Вопросы практического использования алгоритма RSA

На протяжении многих лет алгоритм RSA активно используется как в виде самостоятельных криптографических продуктов, так и в качестве встроенных средств в популярных приложениях.

Открытое шифрование на базе алгоритма RSA применяется в популярном пакете шифрования PGP, операционной системе Windows, различных Интернет-браузерах, банковских компьютерных системах.

Кроме того, различные международные стандарты шифрования с открытым ключом и формирования цифровой подписи используют RSA в качестве основного алгоритма.

Вопросы практического использования алгоритма RSA

Для обеспечения высокой надежности шифрования необходимо, чтобы выступающее в качестве модуля число N было очень большим – несколько сотен или тысяч бит.

Только в этом случае будет практически невозможно по *открытым параметрам* определить закрытый ключ.

Так, известно, что в конце 1995 года удалось практически реализовать раскрытие шифра RSA для 500-значного модуля.

Для этого с помощью сети Интернет было задействовано более тысячи компьютеров.

Вопросы практического использования алгоритма RSA

Сами авторы RSA рекомендовали использовать следующие размеры модуля N: 768 бит - для частных лиц; 1024 бит - для *коммерческой информации*; 2048 бит - для особо секретной информации.

С момента получения их рекомендаций прошло какое-то время, поэтому современные пользователи должны делать поправки в сторону увеличения размера ключей.

Однако, чем больше размер ключей, тем медленнее работает система. Поэтому увеличивать размер ключа без необходимости не имеет смысла.

Вопросы практического использования алгоритма RSA

С размером ключей связан и другой аспект реализации RSA - *вычислительный*. При использовании алгоритма вычисления необходимы как при создании ключей, так и при шифровании/расшифровании, при этом, чем больше размер ключей, тем труднее производить расчеты.

Для работы с громадными числами приходится использовать аппарат *длинной арифметики*. Числа, состоящие из многих сотен бит, не умещаются в регистры большинства микропроцессоров и их приходится обрабатывать по частям.

Вопросы практического использования алгоритма RSA

При этом как шифрование, так и расшифрование включают возведение большого целого числа в целую степень по модулю N. При прямых расчетах промежуточные значения были бы невообразимыми.

Чтобы упростить процесс вычислений используют специальные алгоритмы для работы с большими числами, основанные на свойствах модульной арифметики, а также оптимизацию при возведении в степень.

Вопросы практического использования алгоритма RSA

Алгоритм RSA реализуется как программным, так и аппаратным путем. Многие мировые фирмы выпускают специализированные микросхемы, производящие шифрование алгоритмом RSA. Программные реализации значительные медленнее, чем аппаратные.

К достоинствам программного шифрования RSA относится возможность гибкой настройки параметров, возможность интеграции в различные программные пакеты.

В целом, и программная, и аппаратная реализации RSA требуют для выполнения примерно в тысячи раз большего времени по сравнению с *симметричными алгоритмами*, например ГОСТ 28147-89.

Вопросы практического использования алгоритма RSA

Алгоритм RSA может использоваться для формирования электронной цифровой подписи, а также и для обмена ключами.

Возможность применения алгоритма RSA для получения электронной подписи связана с тем, что секретный и открытый ключи в этой системе равноправны.

Каждый из ключей, d или e, могут использоваться как для шифрования, так и для расшифрования. Это свойство выполняется не во всех криптосистемах с открытым ключом.

Вопросы для проверки

- Для каких целей может применяться алгоритм RSA?
- Опишите процесс шифрования с использованием алгоритма RSA.

Упражнения для проверки

- Пусть пользователь А хочет передать пользователю Б сообщение m=10, зашифрованное с помощью алгоритма RSA. Пользователь Б имеет следующие параметры: P=7, Q=11, d=47. Опишите процесс передачи сообщения m пользователю Б.
- Пользователю системы RSA с параметрами N = 33, d = 3 передано зашифрованное сообщение c = 13. Расшифруйте это сообщение, взломав систему RSA пользователя.

Алгоритм Диффи-Хеллмана

Первая публикация данного алгоритма появилась в 70-х годах XX века в статье Диффи и Хеллмана, в которой вводились основные понятия криптографии с открытым ключом.

Алгоритм Диффи-Хеллмана не применяется для шифрования сообщений или формирования электронной подписи. Его назначение – в распределении ключей.

Он позволяет двум или более пользователям обменяться без посредников ключом, который может быть использован затем для симметричного шифрования.

Это была первая криптосистема, которая позволяла защищать информацию без использования секретных ключей, передаваемых по защищенным каналам.

Схема открытого распределения ключей, предложенная Диффи и Хеллманом, произвела настоящую революцию в мире шифрования, так как снимала основную проблему классической криптографии – проблему распределения ключей.

Алгоритм Диффи-Хеллмана

Алгоритм основан на трудности вычислений *дискретных логарифмов*. Попробуем разобраться, что это такое.

В этом алгоритме, как и во многих других алгоритмах с открытым ключом, вычисления производятся по модулю некоторого большого простого числа P.

Вначале специальным образом подбирается некоторое натуральное число A, меньшее P.

Если мы хотим зашифровать значение X, то вычисляем

$$Y = A^X \bmod P.$$

Алгоритм Диффи-Хеллмана

Причем, имея X, вычислить Y легко.

Обратная задача вычисления X из Y является достаточно сложной.

Экспонента X как раз и называется *дискретным логарифмом Y*.

Таким образом, зная о сложности вычисления *дискретного логарифма*, число Y можно открыто передавать по любому каналу связи, так как при большом модуле P исходное значение X подобрать будет практически невозможно.

На этом математическом факте основан *алгоритм Диффи-Хеллмана* для формирования ключа.

Формирование общего ключа

Пусть два пользователя, которых условно назовем пользователь 1 и пользователь 2, желают сформировать общий ключ для алгоритма симметричного шифрования.

Вначале они должны выбрать большое простое число P и некоторое специальное число A, $1 < A < P-1$, такое, что все числа из интервала [1, 2, …, P-1] могут быть представлены как различные степени A mod P. Эти числа должны быть известны всем абонентам системы и могут выбираться открыто. Это будут так называемые *общие параметры*.

Формирование общего ключа

Затем первый пользователь выбирает число X_1 (X_1<P), которое желательно формировать с помощью датчика случайных чисел. Это будет закрытый ключ первого пользователя, и он должен держаться в секрете. На основе закрытого ключа пользователь 1 вычисляет число

$$Y_1 = A^{X_1} \, mod \, P$$

которое он посылает второму абоненту.
Аналогично поступает и второй пользователь, генерируя X2 и вычисляя

$$Y_2 = A^{X_2} \, mod \, P$$

Это значение пользователь 2 отправляет первому пользователю.

Формирование общего ключа

После этого у пользователей должна быть информация, указанная в следующей таблице:

	Общие параметры	Открытый ключ	Закрытый ключ
Пользователь 1	P, A	Y_1	X_1
Пользователь 2		Y_2	X_2

Из чисел Y1 и Y2, а также своих закрытых ключей каждый из абонентов может сформировать общий секретный ключ Z для сеанса симметричного шифрования. Вот как это должен сделать первый пользователь:

$$Z = (Y_2)^{X_1} \, mod \, P$$

Формирование общего ключа

Никто другой кроме пользователя 1 этого сделать не может, так как число X_1 секретно. Второй пользователь может получить то же самое число Z, используя свой закрытый ключ и открытый ключ своего абонента следующим образом:

$$Z = (Y_1)^{X_2} \, mod \, P$$

Если весь протокол формирования общего секретного ключа выполнен верно, значения Z у одного и второго абонента должны получиться одинаковыми. Причем, что самое важное, противник, не зная секретных чисел X1 и X2, не сможет вычислить число Z. Не зная X1 и X2, злоумышленник может попытаться вычислить Z, используя только передаваемые открыто P, A, Y1 и Y2.

Формирование общего ключа

Безопасность формирования общего ключа в алгоритме Диффи-Хеллмана вытекает из того факта, что, хотя относительно легко вычислить экспоненты по модулю простого числа, очень трудно вычислить дискретные логарифмы. Для больших простых чисел размером сотни и тысячи бит задача считается неразрешимой, так как требует колоссальных затрат вычислительных ресурсов.

Пользователи 1 и 2 могут использовать значение Z в качестве секретного ключа для шифрования и расшифрования данных. Таким же образом любая пара абонентов может вычислить секретный ключ, известный только им.

Пример вычислений по алгоритму

Пусть два абонента, желающие обмениваться через Интернет зашифрованными сообщениями, решили сформировать секретный ключ для очередного сеанса связи. Пусть они имеют следующие общие параметры:

$P = 11, A = 7.$

Каждый абонент выбирает секретное число X и вычисляет соответствующее ему открытое число Y. Пусть выбраны

$X_1 = 3, X_2 = 9.$

Вычисляем

$Y_1 = 7^3 \bmod 11 = 2,$
$Y_2 = 7^9 \bmod 11 = 8.$

Пример вычислений по алгоритму

Затем пользователи обмениваются открытыми ключами Y_1 и Y_2. После этого каждый из пользователей может вычислить общий секретный ключ:

пользователь 1: $Z = 8^3 \bmod 11 = 6$.
пользователь 2: $Z = 2^9 \bmod 11 = 6$.

Теперь они имеют общий ключ 6, который не передавался по каналу связи.

Вопросы практического использования алгоритма Диффи-Хеллмана

Для того, чтобы *алгоритм Диффи-Хеллмана* работал правильно, то есть оба пользователя, участвующих в протоколе, получали одно и то же число Z, необходимо правильным образом выбрать число A, используемое в вычислениях. Число A должно обладать следующим свойством: все числа вида

$$A \bmod P,\ A^2 \bmod P,\ A^3 \bmod P,\ \ldots,\ A^{P-1} \bmod P$$

должны быть различными и состоять из целых положительных значений в диапазоне от 1 до P-1 с некоторыми перестановками.

Вопросы практического использования алгоритма Диффи-Хеллмана

Только в этом случае для любого целого Y < P и значения A можно найти единственную экспоненту X, такую, что

$$Y = A^X \bmod P, \text{ где } 0 <= X <= (P - 1)$$

При произвольно заданном P задача выбора параметра A может оказаться трудной задачей, связанной с разложением на простые множители числа P-1. На практике можно использовать следующий подход, рекомендуемый специалистами. Простое число P выбирается таким, чтобы выполнялось равенство P = 2q + l, где q — также простое число.

Тогда в качестве A можно взять любое число, для которого справедливы неравенства

$$1<A<P-1 \text{ и } A^q \bmod P \neq 1$$

На подбор подходящих параметров A и P необходимо некоторое время, однако это обычно не критично для системы связи и не замедляет ее работу. Эти параметры являются общими для целой группы пользователей. Они обычно выбираются один раз при создании сообщества пользователей, желающих использовать протокол Диффи-Хеллмана, и не меняются в процессе работы. А вот значения закрытых ключей рекомендуется каждый раз менять и выбирать их с помощью генераторов псевдослучайных чисел.

Вопросы практического использования алгоритма Диффи-Хеллмана

Следует заметить, что данный алгоритм, как и все *алгоритмы асимметричного шифрования*, уязвим для атак типа "man-in-the-middle" ("человек в середине").

Если противник имеет возможность не только перехватывать сообщения, но и заменять их другими, он может перехватить открытые ключи участников, создать свою пару открытого и закрытого ключа и послать каждому из участников свой открытый ключ.

После этого каждый участник вычислит ключ, который будет общим с противником, а не с другим участником.

Вопросы для проверки

- Для каких целей может применяться *алгоритм Диффи-Хеллмана*?
- Опишите последовательность действий при использовании *алгоритма Диффи-Хеллмана*.

Упражнения для проверки

- Вычислите закрытые ключи Y_1, Y_2 и общий ключ Z для системы Диффи-Хеллмана с параметрами A=3, P=7, X_1=3, X_2=6.

Алгоритм Эль-Гамаля

Асимметричный алгоритм, предложенный в 1985 году Эль-Гамалем (T. ElGamal), универсален. Он может быть использован для решения всех трех основных задач: для шифрования данных, для формирования цифровой подписи и для согласования общего ключа.

Кроме того, возможны модификации алгоритма для схем проверки пароля, доказательства идентичности сообщения и другие варианты.

Безопасность этого алгоритма, так же как и *алгоритма Диффи-Хеллмана*, основана на трудности вычисления *дискретных логарифмов*.

Этот алгоритм фактически использует схему Диффи-Хеллмана, чтобы сформировать общий секретный ключ для абонентов, передающих друг другу сообщение, и затем сообщение шифруется путем умножения его на этот ключ.

Алгоритм Эль-Гамаля

И в случае шифрования, и в случае формирования цифровой подписи каждому пользователю необходимо сгенерировать пару ключей.

Для этого, так же как и в схеме Диффи-Хеллмана, выбираются некоторое большое простое число P и число A, такие, что различные степени A представляют собой различные числа по модулю P.

Числа P и A могут передаваться в открытом виде и быть общими для всех абонентов сети.

Алгоритм Эль-Гамаля

Затем каждый абонент группы выбирает свое секретное число X_i, $1 < X_i < P-1$, и вычисляет соответствующее ему открытое число

$$Y_i : Y_i = A^{X_i} \, mod \, P$$

Таким образом, каждый пользователь может сгенерировать закрытый ключ Xi и открытый ключ Yi.
Информация о необходимых параметрах системы сведена в следующую таблицу.

Информация о необходимых параметрах системы сведена в следующую таблицу.

	Общие параметры	Открытый ключ	Закрытый ключ
Пользователь 1	P, A	Y_1	X_1
	…		
Пользователь i		Y_i	X_i

Шифрование

Теперь рассмотрим, каким образом производится шифрование данных. Сообщение, предназначенное для шифрования, должно быть представлено в виде одного числа или набора чисел, каждое из которых меньше P. Пусть пользователь 1 хочет передать пользователю 2 сообщение m. В этом случае последовательность действий следующая.

1. Первый пользователь выбирает случайное число k, взаимно простое с P-1, и вычисляет числа

$$r = A^k \, mod \, P, \qquad e = m \times Y_2^k \, mod \, P$$

где Y2 – открытый ключ пользователя 2. Число k держится в секрете.

Шифрование

2. Пара чисел (r, e), являющаяся шифротекстом, передается второму пользователю.
3. Второй пользователь, получив (r,e), для расшифрования сообщения вычисляет

$$m = e \times r^{P-1-X_2} \, mod \, P$$

где X2 – закрытый ключ пользователя 2. В результате он получает исходное сообщение m.

Шифрование

Если злоумышленник узнает или перехватит P, A, Y_2, r, e, то он не сможет по ним раскрыть m. Это связано с тем, что противник не знает параметр k, выбранный первым пользователем для шифрования сообщения m.

Вычислить каким-либо образом число k практически невозможно, так как это задача дискретного логарифмирования.

Следовательно, злоумышленник не может вычислить и значение m, так как m было умножено на неизвестное ему число.

Противник также не может воспроизвести действия законного получателя сообщения (второго абонента), так как ему не известен закрытый ключ X_2 (вычисление X_2 на основании Y_2 — также задача дискретного логарифмирования).

Шифрование

По аналогичному алгоритму может производиться и согласование ключа, используемого для симметричного шифрования больших объемов данных.

Более того, алгоритм Эль-Гамаля на практике целесообразно использовать именно для согласования общего *ключа сессии*, а не прямого шифрования больших сообщений.

Это связано с тем, что в алгоритме используются операции возведения в степень и умножения по большому модулю.

Так же как и в алгоритмах RSA и Диффи-Хеллмана, операции производятся над большими, состоящими из нескольких сотен или тысяч бит, числами. Поэтому шифрование больших сообщений производится крайне медленно.

Пример шифрования

Пусть два абонента, обменивающиеся через Интернет зашифрованными сообщениями, имеют следующие общие параметры:

$P = 11, A = 7.$

Кроме того, пользователи 1 и 2 имеют пары закрытых и открытых ключей

Пользователь 1: закрытый ключ $X_1 = 3$, открытый ключ $Y_1 = 7^3 \bmod 11 = 2$,

Пользователь 2: закрытый ключ $X_2 = 9$, открытый ключ $Y_2 = 7^9 \bmod 11 = 8$.

Первый абонент желает передать второму сообщение. Для этого первый абонент запрашивает из центра распределения ключей открытый ключ второго абонента Y2 = 8. Теперь он может зашифровать свое сообщение, которое в числовом виде пусть имеет значение m=9.

Пример шифрования

Первый абонент выбирает случайно число k, например k = 7.
Число k должно быть взаимно простым с P-1. Значение k = 7 не имеет общих делителей с P-1=10, значит, оно нам подходит. Первый абонент шифрует свое сообщение по формулам:

$r = A^k \bmod P = 7^7 \bmod 11 = 6$

$e = m * Y_2^k \bmod P = 9 * 8^7 \bmod 11 = 7$

Пара чисел (6, 7) будет представлять собой шифротекст и передается второму пользователю. Второй пользователь, получив (6,7) и используя свой закрытый ключ X2 = 9 для расшифрования сообщения, вычисляет

$$m = e \times r^{P-1-X_2} mod P = 7 \times 6^{11-1-9} mod 11 = 7 \times 6^1 mod 11 = 9$$

В результате он действительно получает исходное сообщение m.

Вопросы для проверки

- Для каких целей может применяться алгоритм Эль-Гамаля?.
- Опишите последовательность действий при использовании алгоритма Эль-Гамаля.

Упражнения для проверки

В системе связи, применяющей шифр Эль-Гамаля, пользователь 1 желает передать сообщение m пользователю 2. Найдите недостающие параметры при следующих заданных параметрах P = 19, A = 2, X_2 = 3, k = 5, m = 10.

Спасибо за внимание!