учебный план которой предполагалось взять за базовый для новой специальности, усилив в нем физико-техническую и инженерную компоненту и добавив блок специальных дисциплин, адаптированный к области комплексных систем защиты информации. К сожалению, начавшаяся было разработка пакета нормативных документов к специальности 7.160104 "Административный менеджмент в сфере защиты информации с ограниченным доступом" прервалась с ликвидацией Госкомсекретов Украины. На сегодняшний день подготовка специалистов по этой специальности на основе самостоятельно разработанных учебных планов ведется в Национальном авиационном университете (г. Киев), Национальном горном университете (г. Днепропетровск), Государственном университете информационно-коммуникационных технологий (г. Киев). Во всех трех вузах преимущественный акцент при подготовке специалистов заметно смещен в сторону инженерно-технической компоненты образования, что объясняется традициями и спецификой этих вузов и может рассматриваться как введение технической специализации исходной специальности 7.160104. Вопрос о подготовке специалистов непосредственно в области комплексных систем защиты информации пока остается открытым.

УДК 681.3

МАТЕМАТИЧЕСКИЕ ОСНОВАНИЯ АСИММЕТРИЧНОЙ КРИПТОГРАФИИ

Михаил Савчук
Национальный технический университет Украины "КПИ"

Анотація: **На основі матеріалів публікацій обговорюються математичні ідеї, на яких базується криптографія з відкритим ключем.**
Summary: **Mathematical ideas, on which the public key cryptography is based, are discussed on the grounds of open materials.**
Ключові слова: **Криптографія, криптосистеми з відкритим ключем, автентифікація, цифровий підпис.**

I Введение

После более чем двух тысяч лет развития криптография, благодаря идеям, изложенным Диффи и Хеллманом, а также Ривестом, Шамиром и Адлеманом в их революционных работах [1, 2], соответственно 1976 и 1978 годов, существенно изменила свой облик как с практической стороны, так и в теоретическом отношении. Новая криптография, называемая асимметричной или с открытым ключом, позволила решить такие практические задачи и создать такие криптографические протоколы, которые невозможно было осуществить средствами криптографии с секретными ключами. Две первые практически назревшие с распространением электронных средств обработки и передачи информации проблемы – распространение ключей и цифровая подпись – как раз и решались в работах [1, 2]. С теоретической точки зрения криптография превратилась в математическую дисциплину, которая с одной стороны опирается на фундаментальные математические результаты последних десятилетий (в частности, в теории сложности [3–5]) и сама ставит задачи, решение которых будет означать существенное продвижение в чисто математических областях. С другой стороны современная криптография имеет массу практических приложений в области защиты современных информационных технологий. Бурный рост научных публикаций по криптографии и количество специалистов, ею занимающихся по всему миру, говорит о том, насколько интересна и важна сегодня эта область. В настоящей статье, опираясь на публикации, приведенные, в частности, в списке литературы (прежде всего на работы [6–8]) обсуждаются основные математические идеи, лежащие в основе криптографии с открытым ключом.

II Односторонние функции

Два понятия – односторонней функции и односторонней функции с "потайным ходом", или "лазейкой" – являются центральными для всей криптографии с открытым ключом. Рассмотрим произвольные конечные множества X и Y, а также некоторую функцию $f : X \to Y$. Обозначим через $f[X]$ область значений f. Функция f называется *односторонней,* если ее значение $f(x)$ может быть легко вычислено для каждого аргумента $x \in X$, тогда как почти для всех $y \in f[X]$ нахождение хотя бы одного такого $x \in X$, что $f(x) = y$, является трудновычислимым. В этом неформальном определении нужно уточнить два момента.

Понятие “легко вычисляется” формализуется в теории сложности понятием полиномиального от длины входа алгоритма, в то время как “трудновычислимая” означает, что не существует полиномиального алгоритма вычисления и алгоритм обращения функции имеет как минимум экспоненциальную сложность. И хотя понятия полиномиальной и экспоненциальной сложности асимптотические (по отношению к длине входа), они позволяют разделить задачи на практически вычислимые или решаемые и на вычислительно нереализуемые или нерешаемые [3–5]. Решаемые задачи могут быть рассчитаны на современных ЭВМ за вполне допустимое время, в то время как для вычислений в нерешаемых задачах понадобится время, превосходящее, например, возраст вселенной. “Для почти всех” означает, что доля тех $y \in f[X]$, для которых обращение f возможно, пренебрежимо мала (понятие “для почти всех” имеет четкое математическое определение). А то, что такие y существует, показывает следующий пример. Так как $f(x)$ легко вычисляется, то нетрудно составить таблицу, найдя $y_i=f(x_i)$, $i=1,\ldots,t$, для такого t, которое допускают вычислительные ресурсы и время:

x	x_1	x_2	x_3	…..	x_t
$f(x)$	y_1	y_2	y_3	…..	y_t

Если y принадлежит таблице, то обращение функции осуществляется, как видно из таблицы, быстро. Отсюда понятно, что мощности множеств X и Y у односторонних функций должны быть очень большими.

Нынешнее состояние наших знаний пока еще не позволяет нам доказать, что односторонние функции вообще существуют, так как их существование эквивалентно решению знаменитой проблемы: равны ли классы решаемых задач P и труднорешаемых NP. И все же, несмотря ни на что, имеются кандидаты среди функций, эффективно вычислять которые можно для любого входа, хотя при этом никаких эффективных алгоритмов вычисления обратных функций до сих пор не известно.

Первым простым примером кандидата на одностороннюю функцию является целочисленное умножение. В самом деле, известно, что перемножить два любых, пусть и очень больших, числа относительно нетрудно, тогда как даже самый мощный из существующих сейчас компьютеров не в состоянии разложить на множители с помощью наилучшего имеющегося в его распоряжении алгоритма четырехсотзначное десятичное число, являющееся произведением двух примерно одинакового размера простых чисел. “Не в состоянии” здесь означает не в состоянии за “приемлемое” время (например, за время, ограниченное возрастом вселенной).

Другим очень важным примером кандидата на одностороннюю функцию является дискретное возведение в степень, или модульное экспоненцирование (с фиксированными основанием и модулем). Пусть n и a — целые числа, такие, что $1<a<n$, тогда модульным возведением в степень, или экспоненцированием (относительно основания a и модуля n), называется функция $f(x) = a^x \bmod n$, x – целое, $1<x<n$. Такую функцию можно вычислить эффективно, когда длина каждого из трех параметров a, n и x составляет несколько сотен десятичных знаков, например, с помощью так называемой схемы Горнера по возрастанию или по убыванию.

По аналогии с вещественным анализом задача вычисления функции, обратной модульному возведению в степень, известна как задача дискретного логарифмирования: даны положительные целые числа a, n и y, требуется найти такое целое x (если конечно оно существует), что $a^x \bmod n = y$. Например, $5^4 \bmod 21 = 16$. Так что 4 — это дискретный логарифм 16 с основанием 5 по модулю 21. В настоящее время вычисление больших модульных экспонент можно выполнить очень быстро даже на “персоналке”, но, тем не менее, на сегодняшний день неизвестно ни одного алгоритма вычисления дискретных логарифмов больших чисел за приемлемое время даже на самых мощных, самых быстродействующих суперкомпьютерах. При этом, хотя мы и не можем доказать, что таких эффективных алгоритмов вообще не существует, имеются веские основания предполагать, что модульное возведение в степень (с фиксированными основанием и модулем) действительно является односторонней функцией.

Очевидно, что односторонние функции не могут непосредственно использоваться в качестве криптосистем (когда сообщение m шифруется как $f(m)$), поскольку тогда даже законный получатель не смог бы определить скрытый текст. Но несмотря на это они широко используются в системах защиты информации, например, для защиты паролей. Гораздо более употребимым в криптографии является понятие односторонней функции с потайным ходом (лазейкой). Функция $f: X \to Y$ называется односторонней функцией с потайным ходом (или, что то же самое, с лазейкой), если, во-первых, не только сама f, но и функция f^{-1}, обратная ей, могут быть вычислены эффективно тем, кому известен

потайной ход, а во-вторых, даже если такой эффективный алгоритм вычисления *f* известен, то никакое, пусть самое полное, описание его работы не должно давать возможности построить эффективный алгоритм вычисления f^{-1} тому, кто не знает лазейки. Секрет, с помощью которого, тем не менее, можно эффективно вычислить функцию f^{-1}, как раз и называется потайным ходом, или лазейкой для функции *f*.

Первый кандидат на одностороннюю функцию с потайным ходом – это дискретное возведение в степень, но с фиксированной экспонентной и модулем. Пусть *m*, *n* и *x* – целые положительные числа, тогда дискретное возведение в степень (относительно экспоненты *m* и модуля *n*) есть функция $y = g(x) = x^m \bmod n$. По аналогии с вещественным анализом, операция, обратная $g(x)$, известна как дискретное извлечение корня *m*-ой степени из *y* по модулю *n*: даны целые положительные числа *m*, *n* и *y*, найти такое целое число *x* (если оно существует), что $x^m = y \bmod n$. Например, 5 — это корень 4-ой степени из 16 по модулю 21, потому что $5^4 \bmod 21 = 16$. Очевидно, что 2 также является корнем 4-ой степени из 16 по модулю 21.

Эффективный алгоритм вычисления $g(a)$ для любого основания *a* для случая, когда экспонента *m* и модуль *n* фиксированы, уже приводился. В противоположность задаче дискретного логарифмирования, тем не менее известно, что существует также и эффективный алгоритм извлечения корня *m*-ой степени из *y* по модулю *n* (или выяснения, что такого корня нет) для любого заданного *y*, но только при условии, что известно разложение *n* на простые множители. Именно по этой причине $g(x)$ и является кандидатом на одностороннюю функцию с потайным ходом, для которой *m* и *n* используются как открытая информация, тогда как разложение служит в качестве секретного потайного хода.

Важным частным случаем модульного возведения в степень является тот, при котором экспонента равна 2, а модуль – число некоторого специального вида. Если *p* и *q* – два различных больших простых числа примерно одинакового размера, и кроме того и *p* и *q* сравнимы с 3 по модулю 4, то их произведение $n = p \cdot q$ называется целым числом Блюма. Определим *Z** как множество целых чисел от 1 до *n* –1, которые не делятся ни на *p*, ни на *q*. Наконец, определим Q_n как подмножество множества *Z**, состоящее из чисел, которые являются квадратами по модулю *n*. Элементы Q_n известны как квадратичные вычеты по модулю *n*. Число элементов в *Z** равно $(p-1)(q-1)$, причем в точности четвертую их часть составляют квадратичные вычеты. Каждый квадратичный вычет допускает ровно четыре различных "квадратных корня", из которых лишь один единственный является квадратичным вычетом. Этот особый корень называется примитивным (первообразным) квадратным корнем. Имеющий криптографическое значение факт заключается в том, что способность определять примитивные квадратные корни по модулю такого числа *n* оказывается вычислительно эквивалентной умению раскладывать это *n* на множители. Иначе говоря, тот, кто знает разложение *n* на множители, может эффективно вычислять и примитивные квадратные корни по модулю *n,* тогда как такие вычисления столь же трудны, сколь и факторизация *n,* для того, кто делителей *n* не знает.

Второй кандидат на одностороннюю функцию с потайным ходом получим случайно выбирая *p* и *q* и вычисляя $n = p \cdot q$, которое открыто объявляется. После этого любой человек может эффективно возводить в квадрат по модулю *n,* но только вы сможете эффективно произвести обратные вычисления (в предположении, что факторизация трудна). Открытой информацией здесь является число *n,* а секретным потайным ходом, опять таки, служит его разложение на множители.

III Открытое распределение ключей

Одна из основных трудностей, которую всегда необходимо учитывать в больших многопользовательских криптографических системах заключается в том, что каждая пара ее пользователей, предполагающих обмениваться друг с другом конфиденциальной информацией, должна заранее выработать свой обоюдный секретный ключ. Эта так называемая проблема распространения ключей и решалась в статье Диффи и Хеллмана "Новые направления в криптографии" (1976 г.) методами, которые положили начало эпохе криптографии с открытыми ключами.

Предназначение криптосистемы с открытым распределением ключей как раз и заключается в том, чтобы позволить двум пользователям А и В выработать в результате переговоров друг с другом по несекретному каналу связи такой совместный секретный ключ, который "нарушитель" не мог бы разгадать даже после прослушивания всех этих переговоров. Причем, этого необходимо добиться даже в том случае, если А и В не обменивались заранее никакой информацией, которая не была бы известна нарушителю.

Первый кажущийся абсолютно невозможным протокол, который тем не менее достигает этой цели, был предложен Диффи и Хеллманом в 1976 году. Он основывается на задаче дискретного логарифмирования. Пусть n – некоторое большое целое число, и пусть g – другое целое, лежащее строго между 1 и $n-1$. В качестве первого шага протокола Диффи-Хеллмана **A** и **B** уславливаются об n и g посредством несекретного канала связи (в качестве альтернативы n и g могли бы быть стандартными параметрами, применяемыми всеми пользователями системы). Затем **A** выбирает некоторое большое целое число x и вычисляет $X = g^x \bmod n$. Соответственно, **B** выбирает число y и вычисляет $Y = g^y \bmod n$. После этого **A** и **B** обмениваются числами X и Y по тому же несекретному каналу связи, сохраняя в секрете x и y (**A** при этом знает только x, а **B** — только y). Наконец, **A** вычисляет $Y^x \bmod n$, а **B**, соответственно, вычисляет $X^y \bmod n$. Оба эти значения, очевидно, равны между собой, так как каждое из них равно g^{xy}. Это как раз и есть тот самый секретный ключ k, который **A** и **B** хотели совместно выработать.

Нарушитель при таком протоколе сталкивается с задачей вычисления k из пересылаемых по несекретному каналу чисел g, n, X и Y. Очевидным подходом для нарушителя было бы вычислить x из g, n и X (или по крайней мере некоторое $\hat{x}$, такое, что $g^{\hat{x}} \bmod n = X$, так как для любого подобного x всегда $Y^{\hat{x}} \bmod n = k$). Однако это в точности задача нахождения дискретного логарифма, которая считается практически невыполнимой. Кроме того, никто пока не придумал способа вычислять k эффективно из g, n, X и Y, как никто и не смог доказать, что это невозможно, или хотя бы продемонстрировать, что не существует лучшего способа сделать это, не находя по сути дела в процессе вычисления дискретного логарифма.

Выбор g и n может оказывать существенное влияние на эффективность и надежность. В том случае, когда n является простым числом, всегда существует такое g, что $g^x \bmod n$ принимает каждое из значений в промежутке от 1 до $n-1$, когда x пробегает значения из того же интервала. Подобные g, которые называются образующими элементами или генераторами циклической группы, и есть желательными. При этом надежнее выбрать n таким образом, чтобы $\frac{n-1}{2}$ также было простым.

В предположении, что n и g являются стандартными параметрами, возможна организация и использование некоторого единого для всех пользователей каталога. Каждый пользователь записывает в этот каталог свой собственный открытый ключ X, вычисленный как $g^x \bmod n$ в соответствии с личным случайно выбранным секретным ключом x. Это позволяет любым двум пользователям сформировать свой совместный секретный ключ даже до их предварительной договоренности о нем друг с другом. Основной недостаток такого общедоступного каталога состоит в том, что он не поддерживает достаточно частые изменения пользователями личных секретных ключей.

IV Криптосистемы с открытым ключом

Система открытого распределения ключей позволяет двум сторонам сформировать совместную часть некоторой распределенной секретной информации. Однако при этом ни одна из сторон не имеет никакого непосредственного влияния на то, какой окажется эта информация. Будем в дальнейшем обозначать отправителя **A**, а получателя – **B**. Если бы **A** захотел передать **B** некоторое сообщение, то за использованием системы открытого распределения ключей должно было бы последовать использование криптосистемы с секретным ключом, в которой первоначальная совместно сформированная ими информация сохранялась бы в качестве их общего секретного ключа.

Криптосистемы же с открытым ключом могут непосредственно использоваться для шифрования. Криптографические системы с открытым ключом во многом подобны криптосистемам с секретным ключом. Они состоят из пространства ключей K, из пространства открытых сообщений M_k, пространства шифртекстов C_k для каждого $k \in K$ и пары функций $E_k : M_k \to C_k$ и $D_k : C_k \to M_k$ таких, что $D_k(E_k(m)) = m$ для любого открытого текста $m \in M_k$ и $k \in K$. Так же как и в криптосистемах с секретным ключом эффективные алгоритмы для вычисления и E_k и D_k должны легко получаться для каждого ключа k, причем E_k затем эффективно зашифровывает и при неизвестном потайном коде (т. е. секретном ключе k), а эффективный алгоритм D_k может быть

найден только при известном k. Т. е. важной новой отличительной особенностью является то, что E_k должна быть односторонней функцией с потайным ходом – необходимо, чтобы было практически невозможно построить никакого эффективного алгоритма для вычисления D_k, зная описание алгоритма для вычисления E_k и не зная лазейки. В частности, это означает, что значение k не должно присутствовать в явном виде в алгоритме шифрования.

Криптографические системы с открытым ключом используются следующим образом. Каждый пользователь раз и навсегда выбирает для себя некоторый случайный секретный ключ $k \in K$. Этот ключ он использует для получения обоих алгоритмов E_k и D_k. Затем он делает публично доступным свой алгоритм шифрования E_k, возможно посредством использования некоторого справочника, но при этом хранит в строгой тайне свой алгоритм дешифрования D_k. Тогда, если один из пользователей криптосистемы захочет послать другому некоторое сообщение, то он найдет в справочнике его открытый алгоритм шифрования и использует этот алгоритм для того, чтобы зашифровать свое сообщение. В этом случае после его пересылки, используя собственный секретный ключ, только законный получатель сможет расшифровать полученный шифртекст. Заметим, что в противоположность криптосистемам с секретным ключом, если **A**, зашифровав некоторое сообщение m для **B**, сохранит шифртекст c, но потеряет соответствующий ему открытый текст (или забудет все, что в нем содержалось), то он не будет иметь никаких преимуществ перед нарушителем в раскрытии m из крптограммы c.

Также в противоположность криптосхемам с секретным ключом, если нарушитель, перехватив шифртекст, знает, для кого он был зашифрован, то он может использовать открытый алгоритм шифрования для проверки любого конкретного предположения о том, каким может быть соответствующий открытый текст. Возможность формировать шифртексты из открытых сообщений по своему выбору позволяет организовать атаки на основе выбранного шифртекста.

Криптосистемы с открытым ключом, в принципе, могут существовать лишь в том случае, если существуют не только просто односторонние функции, но и односторонние функции с потайным ходом. Поэтому точно так же, как и в случае односторонних функций пока нет доказательства существования криптосистем с открытым ключом. Были предложены несколько кандидатов таких криптосистем.

Самой первой криптографической системой с открытым ключом из тех, что были предложены в открытой литературе вообще, является криптосистема Ривеста, Шамира и Адлемана, которая известна под названием RSA. Эта криптосистема основывается на предположении, что модульное возведение в степень при фиксированных экспоненте и модуле является односторонней функцией с потайным ходом.

Пусть p и q – два больших различных простых числа, $n = p \cdot q$, а e – некоторое целое, взаимно простое с $(p-1)(q-1)$. Используя обобщенный алгоритм Евклида для нахождения наибольшего общего делителя, необходимо вычислить целое число d такое, что $e \cdot d = 1 \bmod \varphi(n)$, где функция Эйлера $\varphi(n) = (p-1)(q-1)$. Тогда для криптосистемы RSA в качестве личного (секретного) ключа может быть выбрана любая такая тройка $k = (p, q, d)$. Пусть также каждое из соответствующих пространств открытых текстов M_k и шифрованных сообщений C_k принадлежат Z_n, — множеству неотрицательных целых чисел, меньших n. Если при этом реальные сообщения окажутся слишком длинными, чтобы принадлежать Z_n то их необходимо разбить на части и зашифровать, используя известные режимы шифрования. Тогда соответствующая ключу k функция шифрования $E_k : M_k \to C_k$ определяется как $E_k(m) = m^e \bmod n$. Пара (n, e) чисел называется открытым ключом и может быть помещена в открытом справочнике. По известной теореме Эйлера $m^{ed} = m \bmod n$ для каждого целого числа m. Функция дешифрования $D_k : C_k \to M_k$ определяется как $D_k(c) = m^d \bmod n$ и для ее вычисления также может быть использован эффективный алгоритм модульного возведения в степень.

Как указано выше, E_k является кандидатом на одностороннюю функцию с потайным ходом, и хотя существует эффективный алгоритм вычисления обратной ей функции D_k при известном k, но неизвестно как получить его эффективно, задаваясь только алгоритмом вычисления E_k (т.е. только при известных n и e).

Таким образом, каждый пользователь криптосистемы RSA должен совершенно случайно и раз и навсегда выбрать для себя подходящие целые числа p, q и e и вычислить с их помощью d. После чего он должен сделать свой открытый ключ (e,n) доступным в пользовательском справочнике, но при этом сохранять d в секрете. Это дает возможность остальным пользователям зашифровывать посылаемые ему сообщения, которые только он один потом сможет расшифровать. Для того чтобы эта идея могла быть реализована на практике, решающим является удовлетворение требования, чтобы генерация больших

случайных простых чисел и вычисление d были легкоосуществимы, что действительно имеет место. Проверке чисел на простоту посвящен ряд интересных работ (см., например, И. Горбенко и В. Вервейко "Тестирование чисел на простоту: теория и практика" в настоящем сборнике). Расширенный алгоритм Евклида для вычисления d можно найти, например, в [9].

Отметим ситуацию, когда криптоаналитик, перехвативший шифртекст $c=E_k(m)$, посланный известному пользователю, знает алгоритм шифрования E_k, который используется отправителем для вычисления c. Такое предположение имеет два важных следствия. Если бы нарушитель мог в точности угадать открытый текст сообщения m, то он был бы в состоянии точно так же, как и отправитель, вычислить $E_k(m)$, а затем проверить свою догадку, сравнив полученный результат с шифртекстом c. Учитывая возможность исчерпывающего поиска, такая угроза является довольно опасной, если число возможных открытых текстов сравнительно невелико. Если же добавлять в короткие сообщения случайные биты, то эта трудность может быть до некоторой степени разрешена, однако, более приемлемым решением, несомненно, является использование вероятностного шифрования.

Более существенным для криптосистемы RSA является другое следствие из того факта, что нарушителю доступен открытый алгоритм шифрования. Действительно, он знает, что $c = m^e \bmod n$ для известных значений c, e и n (хотя и не знает m). Поэтому, если бы он мог разложить n на множители (раскрыв таким образом личный секретный ключ (p,q,d) законного получателя шифртекста c, то он мог бы получить и $\varphi(n) = (p-1)(q-1)$, после чего, применив расширенный алгоритм Евклида, вычислить d, чтобы затем найти $m = c^d \bmod n$. В настоящее время неизвестно никакого алгоритма, с помощью которого можно было бы разложить на множители четырехсотзначное десятичное число "за приемлемое время", и поэтому считается абсолютно надежным выбирать числа p и q длиной примерно в двести (десятичных) знаков. Выбирать p и q необходимо с особой тщательностью, чтобы при использовании известных алгоритмов факторизации не предоставить криптоаналитику никаких "зацепок". В частности, наибольший общий делитель чисел $p-1$ и $q-1$ должен быть небольшим, а оба они, как $p-1$, так и $q-1$, должны иметь большие простые делители, например, $(p-1)/2$ и $(q-1)/2$, которые являются простыми числами.

Даже если считать, что факторизация действительно трудна, остается неизвестным, является ли столь же трудным само раскрытие RSA. Ведь вполне вероятно, что d может быть вычислено из открытой информации e и n вообще без разложения n на множители. Но возможно также и то, что значение d (а, стало быть, и значения множителей числа n) действительно трудно вычислить практически из e и n, даже если существует какой-то иной эффективный алгоритм раскрытия m из e, n и $m^e \bmod n$. Пока никто таких алгоритмов не предложил. Система RSA до настоящего времени считается при правильном выборе параметров вполне надежной. Но из-за достаточно большой трудоемкости модульных вычислений и, следовательно, невысокой скорости передачи сообщений, криптосистема RSA используется, как правило, для передачи очень коротких сообщений, ключей или цифровой подписи. Собственно основополагающая работа, в которой была предложена система RSA, и называлась "Методы получения цифровой подписи и криптосистемы с открытыми ключами" [2]. Цифровая подпись решила проблему аутентификации источника сообщений, подтверждения целостности сообщения и возможности доказать это другим лицам.

V Аутентификация и цифровая подпись

До сих пор обсуждалось лишь положение пассивного криптоаналитика, чья цель заключалась только в прослушивании канала связи. Активный криптоаналитик идет дальше: не удовлетворяясь прослушиванием канала связи, он может также вводить свои собственные сообщения в надежде на то, что получатель при их расшифровке может поверить, что они были посланы законными отправителями. Например, финансовые сделки должны быть защищены в первую очередь от подобной фальсификации, а не быть непременно засекреченными.

Целью системы аутентификации, или иначе системы удостоверения авторства, точно так же, как и системы подтверждения целостности, является выявление указанного выше фальсификатора, криптоаналитика или случайных, непреднамеренных искажений информации и идентификаторов. Всякий раз, когда B получает сообщение, в котором утверждается, что оно было послано от A, система должна позволить ему убедиться не только в том, что это сообщение действительно исходит от **A**, но и в том, что оно не было изменено при передаче. Допускается, что фальсификатор в состоянии прослушивать столько аутентифицированных (то есть подтверждающих свою соответственную подлинность) сообщений, сколько он хочет, и его цель состоит в том, чтобы добиться именно такой подделки сообщения, которая позволит ему избежать ее обнаружения. Это подделанное сообщение может либо

полностью отличаться от уже перехваченных, либо частично отличаться от некоторых из них, либо быть послано до перехваченных сообщений. Поэтому для того, чтобы легче было обнаружить подмену, важно, чтобы каждое сообщение включало временную отметку или некий порядковый номер.

Не сразу было осознанно, что задача аутентификации – это отличная от шифрования отдельная криптографическая проблема, которая должна решаться своими методами. Теория аутентификации, во многом схожая с теорией связи в секретных системах Шеннона, была заложена в 80-х годах прошлого века Г. Дж. Симмонсом (см. [10]).

Любая аутентификационная система состоит из пространства ключей K, и для каждого $k \in K$ – из пространства сообщений M_k, пространства меток T_k и аутентификационной функции $A_k : M_k \to T_k$. При этом для любого заданного k должен легко получаться эффективный алгоритм вычисления A_k. Аутентификационная система используется следующим образом. Если **A** и **B** ожидают, что со временем они могут обмениваться друг с другом подтверждающими свою подлинность сообщениями, то они сначала должны договориться о некотором секретном ключе $k \in K$. Всякий раз, когда **A** захочет аутентифицировать некоторое сообщение $m \in M_k$ для **B** (то есть подтвердить ему свое авторство данного сообщения), он вычисляет его метку $t = A_k(m)$ и посылает ее вместе с m. Для того чтобы удостовериться в подлинности этого сообщения, **B** также вычисляет $A_k(m)$ и сравнивает его с той меткой t, которую он получил. Конечно, это не исключает возможности зашифровывать сообщение m, если требуется не только подтверждение его подлинности, но и соблюдение секретности.

Аутентификационная функция не обязана быть взаимнооднозначной и пространство меток может быть существенно меньше, чем пространство сообщений. Это важно также с практической точки зрения при передаче сообщений. Однако оно не должно быть слишком маленьким с тем, чтобы случайно выбранная метка все-таки имела пренебрежимо малую вероятность быть правильной для конкретного удачно измененного фальсификатором сообщения. Роль таких аутентификационных функций играют, в частности, хеш-функции с ключом. Как и в случае криптографии с секретным ключом, здесь также можно выделить различные уровни атаки.

Криптосистема с секретным ключом может использоваться в целях аутентификации. Идея заключается в том, чтобы зашифровывать сообщения, например, блочным криптоалгоритмом либо в режиме шифрования со сцеплением блоков, либо в режиме шифрования с обратной связью по шифртексту и разрешить использовать для аутентификационной метки последний зашифрованный таким образом блок. В этом случае, согласно определению каждого из режимов, метка будет зависеть от всего сообщения. Если же добиваться секретности и аутентификации одновременно, и если для обеспечения обеих этих функций используется одна и та же криптосистема с секретным ключом, то тогда существенно надежнее будет использовать два различных секретных ключа.

Несмотря на то, что схема аутентификации позволяет **B** достичь большой уверенности в том, что сообщение, которое он получил, исходило именно от **A**, эта схема не позволяет ему убедить кого бы то ни было еще в том, что **A** и в самом деле посылал полученное им сообщение. Таким образом, вышеописанные аутентификационные схемы могут использоваться в отношениях между двумя доверяющими друг другу сторонами, но они не способны обеспечивать улаживание возникающих между ними разногласий. Такая возможность предоставляется только благодаря существованию односторонних функций с потайным ходом и основанного на них более сильного понятия цифровой подписи.

Почему приведенная ранее схема аутентификации не обеспечивает цифровой подписи? Заметим, что если **A** аутентифицирует сообщение для **B** так, как это описано выше, то **B** было бы столь же легко получить соответствующую метку и самому. Потому как **A** может утверждать, что **B** сам создал поддельное сообщение, так и **B** заявить, что **A** прислал ему сообщение, которое тот не посылал. Но доказать что-либо другим лицам ни **A**, ни **B** не может, даже открыв секретный ключ.

Если **A** посылает **B** сообщение, подписанное своей цифровой подписью, то **B** при его получении не только сам сможет убедиться в том, что это сообщение подписано ни кем иным, как его составителем (т. е. **A**), но и будет также способен доказать в суде (если такая возможность предусмотрена законом), что именно **A**, и никто иной, подписал это сообщение. Понятие цифровой подписи было введено Диффи и Хеллманом [1]. Цифровая подпись и электронная почта сулят значительные преимущества, которые даже в принципе неосуществимы в традиционном бумажном документообороте.

Криптосхема с открытым ключом позволяет обеспечить возможность цифровой подписи. Для осуществления цифровой подписи такая схема используется следующим образом. Пусть a — некоторый

секретный ключ **A** и пусть E_a и D_a — ее функции шифрования и, соответственно, дешифрования. Тогда только **A** сможет вычислить D_a эффективно, хотя при этом каждый знает, как вычислять E_a.

Рассмотрим далее некоторый открытый текст m и положим $s = D_a(m)$. Очевидно, что любой пользователь криптосистемы может эффективно вычислить $E_a(s)$ и выяснить, что ее значением является m. Однако только **A** обладает теми знаниями, которые необходимы, чтобы получить такое s, что $E_a(s)=m$. В этом смысле s может рассматриваться как подпись самого **A** под сообщением m. Другими словами, секретный алгоритм дешифрования D_a может рассматриваться в этом случае как алгоритм, осуществляющий цифровую подпись, а открытый алгоритм шифрования E_a – как соответствующий алгоритм подтверждения этой подписи.

В криптосистемах с открытым ключом цифровая подпись может использоваться совместно с шифрованием, если требуется также соблюдать и секретность. Предположим, что b — секретный ключ **B**. Тогда, если **A** захочет послать **B** подписанное секретное сообщение m, то он использует свой секретный алгоритм подписи D_a и его открытый алгоритм шифрования E_b, чтобы получить $c = E_b(m, D_a(m))$. Если **A** пошлет **B** сообщение c по открытому каналу, то тот сможет вычислить сообщение m и подпись **A** под этим сообщением как $(m, s) = D_b(c)$, а затем проверить $m = E_a(s)$. При этом предполагается, конечно, что в заголовке сообщения явно говорится, что оно исходит от **A**, так что **B** знает, чей алгоритм подтверждения подписи применять для того, чтобы получить открытый текст. Более того, если **B** сохранит s, то он сможет доказать, что именно **A**, и никто другой, послал ему сообщение m.

Если RSA используется как для обеспечения секретности, так и для цифровой подписи, может быть предпочтительней, чтобы каждый пользователь хранил для двух разных целей две различные пары функций с потайным ходом. Тогда у каждого пользователя была бы одна запись в открытом справочнике шифрования, а другая – в открытом справочнике проверки подписей. Такое разделение целесообразно по двум причинам. Во-первых, оно позволяет избежать проблемы переразбиения на блоки, которая в противном случае возникает, если модуль отправителя оказывается больше модуля получателя. Во-вторых, криптосистема RSA является слабой относительно атаки на основе выбранного шифртекста. И такую атаку может быть труднее проводить, если процедура цифровой подписи отличается от процедуры дешифрования.

Криптосистема RSA предоставляет возможность цифровой подписи. Однако ее использование обнаруживает недостатки, сходные с теми, которые она имеет как схема шифрования с открытым ключом. В частности, неизвестно, является ли ее раскрытие таким же трудным, как и разложение на множители больших целых чисел. Даже если и в самом деле трудно для выбранного открытого текста m и какого-то открытого ключа (e, n) вычислить подпись s такую, что $m = s^e \bmod n$, то может оказаться намного проще сделать то же самое при известной, кроме этого, паре чисел $(\hat{s}, \hat{m})$, где $\hat{s}$ – подпись законного пользователя под некоторым сообщением $\hat{m}$, которое лишь немного отличается от m. Здесь могут оказаться эффективными атаки, опирающиеся на так называемую задачу о днях рождения. На практике цифровые подписи формируются после преобразования сообщения m с помощью односторонней, стойкой к коллизиям хеш-функции $h(x)$ и применения алгоритма $D_a(H)$ к хеш-образу сообщения $H = h(m)$. Это позволяет стандартизировать и значительно ускорить процесс создания и проверки подписи. Подпись сообщения имеет вид $(m, D_a(H))$. Хеш-функция $h(x)$ не является секретной и известна в том числе и проверяющим сторонам. При проверке подписи делается сравнение $h(m) = E_a(D_a(H))$. При равенстве подпись считается подлинной и сообщение не измененным. В противном случае подпись не подтверждена.

Криптография с открытыми ключами позволила осуществить такие криптографические протоколы и решать такие криптографические задачи, которые ранее невозможно было даже ставить. Так, например, это различные виды цифровой подписи (групповая, неоспоримая, слепая и т. д.), протоколы доказательства с нулевым знанием, защищенных вычислений, предъявления случайных бит и др. Для реализации протоколов с ассиметричными ключами стали применяться новые математические объекты и теории (например, эллиптические кривые), которые позволили значительно увеличить скорость вычислений. Так, в новом стандарте цифровой подписи Украины “ДСТУ 4145-2002” [11] вычисления производятся в группах точек эллиптических кривых над конечными полями, что позволило увеличить скорость формирования и проверки цифровой подписи в несколько раз. Подробно математические результаты по теории сложности алгоритмов, их применении в криптографии, обоснованию оценок стойкости криптографических систем описаны, например, в работах [4, 5, 12, 13].

VI Заключение

Криптография с открытым ключом позволила решить целый ряд практических задач защиты информации, что просто невозможно было осуществить с помощью симметричной криптографии. Ярким примером является полное техническое решение с помощью асимметрической криптографии проблемы надежной проверки соблюдения договоров о запрещении ядерных испытаний [14].

В этой статье были рассмотрены только некоторые первоначальные задачи, решаемые с помощью асимметричной криптографии. До настоящего времени постоянно предлагаются новые криптосистемы и криптографические протоколы, которые обладают некоторыми привлекательными характеристиками, а также новые или более совершенные методы криптоанализа и криптоатаки на традиционные и новейшие криптосистемы. Стойкость криптосистем с открытым ключом опирается на проблему существования односторонних функций. Хотя пока это математически не доказано, но большинство математиков считают, что классы решаемых задач P и труднорешаемых NP не равны и существуют односторонние функции и, следовательно, стойкие криптосистемы с открытым ключом. Широкое использование асимметричной криптографии по всему миру подтверждает возможность эффективного и надежного применения алгоритмов и методов криптографии с открытым ключем.

Литература: ***1.*** *Diffie, W., Hellman, M.E. New directions in cryptography. // IEEE Transactions on Information theory. – 1976. – V. IT–22. – P. 644–654.* ***2.*** *Rivest, R.L., Shamir, A., Adleman, L.M. A method for obtaining digital signatures and public-key cryptosystems. // Communications of the ACM. – 1978. – V. 21. – P. 120-126.* ***3.*** *Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. – М.: Мир, 1979. – 536 с.* ***4.*** *Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи. – М.: Мир, 1982. – 416 с.* ***5.*** *Вербіцький О. В. Вступ до криптології. – Львів: Науково-технічна література, 1998. – 248 с.* ***6.*** *Брассар Ж. Современная криптология. – М.: ”Полимед”, 1999. – 176 с.* ***7.*** *Диффи У., Хеллман М. Защищенность и имитостойкость. // ТИИЭР. – 1979. – Т. 67, № 3. – С. 71-109.* ***8.*** *Месси Дж. Л. Введение в современную криптологию. // ТИИЭР. – 1988. – Т. 76, № 5. – С. 24–42.* ***9.*** *Виноградов И. М. Основы теории чисел. – М.: Наука, 1965 – 172 с.* ***10.*** *Симмонс Г. Дж. Обзор методов аутентификации информации. // ТИИЭР. – 1988. – Т. 76, № 5. – С. 105-125* ***11.*** *Національний стандарт України. Інформаційні технології. Криптографічний захист інформації. Цифровий підпис, що ґрунтується на еліптичних кривих. Формування та перевіряння. ДСТУ 4145-2002. – К.: Державний комітет України з питань технічного регулювання та споживчої політики, 2003. – 38 с.* ***12.*** *Саломаа А. Криптография с открытым ключом. – М.: Мир, 1996. – 318 с.* ***13.*** *Кузьминов Т. В. Криптографические методы защиты информации. – Новосибирск: Наука. Сиб. предприятие РАН, 1998. – 194 с.* ***14.*** *Симмонс Г. Дж. Как обеспечить доверие к данным, используемым для проверки соблюдения договоров. // ТИИЭР. – 1988. – Т. 76, № 5. – С. 126–133.*