КРИПТОЛОГИЯ

вчера, сегодня, завтра

В. Олейник

-Это действительно в высшей степени любопытный рисунок, - сказал Холмс. - С первого взгляда его можно принять за детскую шалость. Кто, казалось бы, кроме детей, мог нарисовать этих крошечных танцующих человечков?

Артур Конан Дойл.

53##+305))6*;4826)4#.)4#.);806*,48+8||60)),85;;]8*;:#*8+83(88)5*+;46(;
88*96*?;8)*#(;485);5*+2:*#(;4956*2(5*=4)8||8*;4069285);)6+8)4##;1(#
9;4881;8:81;48+85;4)485+528806*81(#9;48;(88;4(#?34;48)4#;161;:188#?;

Эдгар Аллан По.

Несколько слов о терминологии.

Термин "криптология" происходит от греческих корней, означающих "тайный" и "слово", и используется для обозначения всей области секретной связи. Криптография довольно четко делится на две части: *криптографию* (шифрование) и *криптоанализ*.

Криптография - совокупность методов и средств семантического преобразования информации в целях обеспечения секретности ее содержания. Другими словами, криптограф пытается найти методы обеспечения секретности и (или) аутентичности (подлинности) сообщений. Криптоаналитик пытается выполнить обратную задачу, раскрывая шифр или, подделывая кодированные сигналы таким образом, чтобы они были приняты как подлинные.

Исходное сообщение, к которому криптограф применяет свое искусство, называется открытым текстом сообщения, или просто *открытым текстом,* а результат его работы - шифрованным текстом сообщения - *шифртекстом,* или *криптограммой.* Для управления процессом шифрования криптограф всегда использует *секретный ключ.* Часто (но не всегда) он передает этот секретный ключ каким-либо надежным способом (например, в "дипломате", пристегнутом наручниками к руке курьера) человеку (или машине), которому он собирается позднее послать криптограмму, составленную с использованием этого ключа.

Почти общепринятое допущение в криптографии состоит в том, что криптоаналитик противника имеет полный текст криптограммы. Кроме того, криптограф почти всегда руководствуется правилом, впервые сформулированным голландцем Керкхоффом (1835-1903): стойкость шифра должна определяться только секретностью ключа. Иными словами, *правило Керкхоффа* состоит в том, что весь механизм шифрования, кроме значения секретного ключа, известен криптоаналитику противника. Если криптограф принимает только эти два допущения, то он разрабатывает систему, стойкую при анализе *на основе только шифрованного текста.* Если к тому же криптограф допускает, что криптоаналитик

противника сможет достать (тем или иным способом) несколько отрывков открытого текста и соответствующего ему шифрованного текста, то разрабатывается система, стойкая при анализе *на основе открытого текста.*

Криптограф может даже допустить, что криптоаналитик противника способен ввести свой открытый текст и получить правильную криптограмму, образованную с использованием секретного ключа (анализ *на основе выбранного открытого текста*), или предположить, что криптоаналитик противника может подставить фиктивные "криптограммы" и получить текст, в который они превращаются при расшифровывании (анализ *на основе выбранного шифртекста*), или допустить обе эти возможности (анализ *на основе выбранного текста*).

Разработчики большинства современных шифров обеспечивают их стойкость к анализу на основе выбранного открытого текста даже в том случае, когда предполагается, что криптоаналитик противника сможет прибегнуть к анализу на основе шифртекста.

Краткая история развития криптологии.

Весь период с древних времен до 1949г. можно по праву назвать *эрой донаучной криптологии.* Это не означает, конечно, что история криптологии тех времен сегодня лишена всякого интереса; скорее, ею занимались тогда почти исключительно как искусством, а не как наукой.

Более 2000 лет назад Юлий Цезарь писал Цицерону и другим друзьям в Риме, используя шифр, в котором каждая буква открытого текста заменялась третьей по счету (циклически) буквой латинского алфавита [1]. Таким образом, из открытого текста CAESAR получался шифртекст FDHVDU. Сегодня мы описали бы шифр Цезаря уравнением

$$Y = X \oplus Z, \qquad (1)$$

где X - буква открытого текста (A=0, B=1, ..., Z=25), Z - секретный ключ (в качестве которого Юлий Цезарь всегда выбирал число 3, а Цезарь Август - 4), Y - буква шифртекста, а $\oplus$ обозначает сложение по модулю 26 ($23 \oplus 3 = 0$; $23 \oplus 4 = 1$, и т.д.).

У нас нет исторических свидетельств о том, что Брут раскрыл шифр Цезаря, но сегодня простой школьник, немного знающий латынь и простейшие приемы криптоанализа, мастерски описанные Эдгаром По в рассказе "Золотой жук", без труда раскроет этот шифр, имея лишь несколько предложений шифрованного текста.

И действительно в течение почти двух тысячелетий после Цезаря криптоаналитики имели явное превосходство над криптографами. Наконец в 1926 г. Г.С. Вернам, инженер Американской телефонной и телеграфной компании, опубликовал замечательный шифр, предназначенный для использования с двоичным кодом Бодо. Шифр Вернама подобен шифру Цезаря: он описывается уравнением (1), в котором теперь X,Y и Z принимают значения из двоичного алфавита [0, 1], а $\oplus$ обозначает сложение по модулю 2 ($0 \oplus 0=0$, $0 \oplus 1=1$, $1 \oplus 1=0$). Новая идея, выдвинутая Вернамом, состояла в том, чтобы использовать ключ только один раз; при этом каждый бит шифруется с использованием нового случайного бита ключа. Это требует передачи по секретному каналу ключа, объем которого равен объему шифруемого затем текста, однако это дает, как мы увидим далее, действительно нераскрываемый шифр. Вернам и в самом деле считал свой шифр нераскрываемым и

знал, что это его свойство теряется при повторном использовании битов ключа, но не представил никаких доказательств этого.

Мы называем период до 1949 г. эрой донаучной криптологии, поскольку достижения тех времен основаны на интуиции и "вере", не подкрепленных доказательствами.

Публикация в 1949 г. статьи К. Э. Шеннона "Теория связи в секретных системах" [2] возвестила начало новой *эры научной криптологии с секретными ключами*. Шеннон не только доказал невозможность раскрытия случайного шифра Вернама, но и установил четкие границы для объема секретного ключа, передаваемого по защищенному каналу предполагаемому получателю.

По причинам, которые станут ясны в дальнейшем, статья Шеннона, опубликованная с 1949 г., не вызвала резкого роста числа работ по криптологии, к какому привела его же статья по теории информации, опубликованная в 1948 г. Настоящий взрыв произошел с появлением в 1976 г. статьи У.Диффи и М.Е.Хеллмана "Новые направления в криптографии" [3]. Диффи и Хеллман впервые показали, что секретная связь возможна без всякой передачи секретного ключа между отправителем и получателем, открыв таким образом бурную *эпоху криптографии с открытыми ключами*, продолжающуюся и сегодня.

<u>Шифры с секретными ключами</u>.

Криптосистемой с секретными ключами называют систему, соответствующую схеме, показанной на рис. 1. Важная часть такой системы - "защищенный канал", по которому секретный ключ $\mathbf{Z} = [Z_1, Z_2, ..., Z_k]$, порожденный в источнике ключа и защищенный от "любопытных глаз" криптоаналитика, передается предполагаемому получателю. Для того чтобы

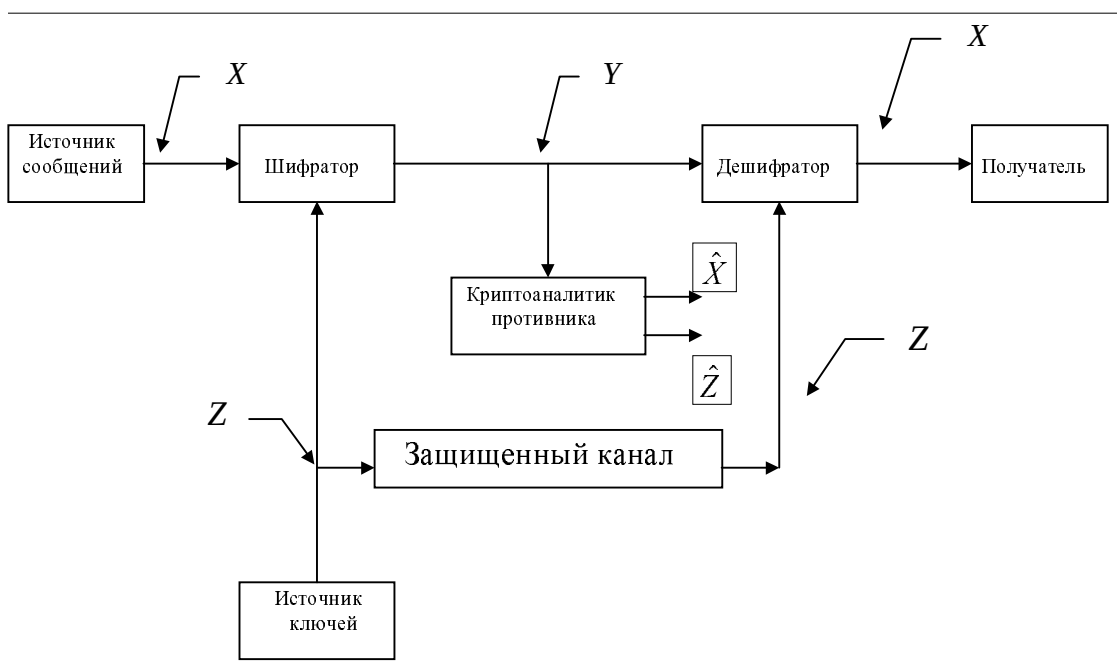

Рис. 1. Схема криптосистемы с секретными ключами.

подчеркнуть факт использования одного и того же ключа в шифраторе источника и дешифраторе получателя сообщений, криптосистемы с секретными ключами называют также *одноключевыми*, или *симметричными, системами.*

k знаков ключа - это символы некоторого конечного алфавита, в качестве которого мы будем часто использовать двоичный алфавит $\{0,1\}$. *Источник сообщений* порождает открытый текст $\mathbf{X} = [X_1, X_2, ..., X_m]$. Шифратор образует криптограмму $\mathbf{Y} = [Y_1, Y_2, ..., Y_n]$, как функцию **X** и **Z**. Это преобразование будем записывать в виде

$$Y = E_Z(X) \qquad (2)$$

для того чтобы подчеркнуть, что криптограмма **Y** является функцией одного лишь открытого текста **X**, конкретный вид которой определяется секретным ключом **Z.** Как следует из рис. 1, дешифратор способен также выполнить обратное преобразование. Таким образом, запись

$$X = D_Z(Y) \qquad (3)$$

выражает тот факт, что открытый текст **X** является функцией криптограммы **Y,** конкретный вид которой определяется одним лишь секретным ключом **Z.** Криптоаналитик противника видит только криптограмму **Y** и образует оценку $\hat{X}$ открытого текста **X** и (или) оценку $\hat{Z}$ секретного ключа **Z.**

Впервые схема рис.1 была приведена в статье К.Шеннона 1949г.[2] и до настоящего времени является актуальной для систем с секретными ключами. Здесь важно понимать, что **X, Y** и **Z** - случайные величины. Вполне понятно, что статистические свойства открытого текста **X** определяются источником сообщений; однако статистические свойства секретного ключа **Z** находятся под контролем криптографа. Как следует из рис. 1, мы всегда будем полагать, что **X** и **Z** статистически независимы.

Теоретическая и практическая стойкость.

Шеннон рассматривал вопрос о стойкости криптографических систем с двух совершенно разных точек зрения. Сначала он поставил вопрос о *теоретической стойкости*: "Насколько надежна некоторая система, если криптоаналитик противника не ограничен временем и обладает всеми необходимыми средствами для анализа криптограмм?"([2], с.360). Решение вопроса о теоретической стойкости, как мы увидим, вносит ясность в криптографию, но приводит к пессимистическому выводу: объем секретного ключа для построения теоретически стойкого шифра недопустимо велик для большинства применений. Поэтому Шеннон рассмотрел также и вопрос о *практической стойкости*: надежна ли система, если криптоаналитик располагает ограниченным временем и вычислительными возможностями для анализа перехваченных криптограмм? Системы с открытыми ключами, которые будут рассмотрены немного позже, должны обладать практической стойкостью, но не могут обеспечить теоретическую стойкость.

Совершенная секретность.

Первое допущение Шеннона в вопросе о теоретической стойкости заключается в том, что секретный ключ используется только один раз, т.е. после зашифрования **M** знаков открытого текста **X** нужно заменить секретный ключ **Z.** Второе допущение состоит в том, что криптоаналитику доступна только криптограмма **Y**, и поэтому он может предпринять лишь анализ на основе шифртекста. *Совершенная секретность,* по определению Шеннона, означает, что открытый текст **X** и криптограмма **Y** статистически независимы, т.е. **P(X=*x* | Y=*y*)** = **P(X=*x*)** для всех возможных открытых текстов $\mathbf{x} = [x_1, x_2, \ldots, x_M]$ и криптограмм ***y*.** Другими словами, криптоаналитик не может улучшить оценку открытого текста **X** на основе знания криптограммы **Y** по сравнению с оценкой при неизвестной криптограмме независимо от того, каким временем и вычислительными возможностями он располагает для анализа криптограммы. При такой точной математической постановке задачи Шеннону удалось показать, что "совершенно секретные" системы существуют.

Рассмотрим шифр, в котором символы открытого и шифрованного текста, а также ключа принимают значения из алфавита {0, 1, ..., L-1} объема L, а длина K ключа и длина N криптограммы совпадают с длиной M открытого текста, т.е. K=N=M. Предположим, что ключ выбирается *совершенно случайным образом*, т.е. $P(\mathbf{Z} = \mathbf{z}) = L^{-M}$ для всех L^M возможных значений **z** секретного ключа, а операция зашифрования определяется выражением:

$$Y_i = X_i \oplus Z_i, \qquad i=1, 2, \ldots, M, \qquad (4)$$

где $\oplus$ обозначает сложение по модулю L. Поскольку для всех возможных значений x_i из X_i и y_i из Y_i соответствует единственное значение z_i, такое что $Z_i = z_i$ удовлетворяет выражению (4), то, следовательно, $P(\mathbf{Y} = \mathbf{y} | \mathbf{X} = \mathbf{x}) = L^{-M}$ для каждого ***y*** и каждого ***x***, независимо от статистических свойств ***X***. Таким образом, ***X*** и ***Y*** статистически независимы, и такая *система Вернама по модулю L* (в терминах Шеннона) обеспечивает совершенную секретность.

Система Вернама широко известна как шифр-блокнот, употреблявшийся разведчиками некоторых стран во время и после второй мировой войны. Разведчикам давался блокнот со случайным секретным ключом и говорилось, что он может быть использован для зашифровывания только одного сообщения. В криптографических кругах, видимо, были убеждены в невозможности раскрытия такого шифра, однако впервые теоретическое доказательство этого факта было дано Шенноном.

Следует отметить, что шифр Вернама обеспечивает совершенную секретность независимо от статистических свойств открытого текста **X**. Действительно, мы покажем, что в этом шифре используется ключ наименьшего возможного объема (среди всех шифров, обеспечивающих совершенную секретность) при любых статистических свойствах открытого текста, это очень полезное свойство, так как любая зависимость системы шифрования от статистической природы источника сообщений, как правило, нежелательна. Однако требование одного знака секретного ключа для каждого знака текста делает шифр Вернама непригодным для большинства применений, за исключением, разумеется, тех немногих, где наиболее

важна секретность, а объем текста невелик (например, в линии правительственной связи Москва-Вашингтон).

Требования к ключам в совершенно секретной системе.

Для дальнейшего изложения вопросов теоретической стойкости нам необходимо будет использовать некоторые свойства "неопределенности" (или "энтропии") - основной количественной оценкой в теории информации Шеннона [4]. Неопределенность - это математическое ожидание взятого со знаком минус логарифма соответствующего распределения вероятностей. Так, H(**X**|**Y**) (читается "неопределенность **X** при известном **Y**") есть математическое ожидание логарифма величины P(**X**=x|**Y**=**y**), т.е.

$$H(\mathbf{X}|\mathbf{Y}) = \sum_x \sum_y P(X = x, Y = y)[-\log P(X = x|Y = y)],$$

где суммы вычисляются по всем возможным значениям **x** и **y** случайных величин **X** и **Y**. Неопределенности подчиняются таким естественным правилам, как H(**X**|**Y**)=H(**X**)+H(**Y**|**X**), и мы будем пользоваться ими без дальнейших обоснований. (Доказательства этих "очевидных" свойств неопределенности можно найти в статье [4] или во вводных главах любого учебника по теории информации.)

Уравнения (2) и (3) можно соответственно переписать в терминах неопределенностей в виде

$$H(\mathbf{Y}|\mathbf{X},\mathbf{Z}) = 0, \qquad (5)$$

$$H(\mathbf{X}|\mathbf{Y},\mathbf{Z}) = 0, \qquad (6)$$

поскольку, например, H(**X**|**Y**, **Z**) равно 0 тогда и только тогда, когда **Y** и **Z** вместе однозначно определяют **X**. Определение совершенной секретности можно представить в виде

$$H(\mathbf{X}|\mathbf{Y}) = H(\mathbf{X}), \qquad (7)$$

поскольку это равенство выполнено тогда и только тогда, когда **X** и **Y** статистически независимы.

Для криптосистемы с секретными ключами имеем

$$H(\mathbf{X}|\mathbf{Y}) \le H(\mathbf{X},\mathbf{Z}|\mathbf{Y}) = H(\mathbf{Z}|\mathbf{Y}) + H(\mathbf{X}|\mathbf{Y},\mathbf{Z}) = H(\mathbf{Z}|\mathbf{Y}) \le H(\mathbf{Z}). \qquad (8)$$

Здесь мы использовали равенство (6) и тот факт, что уменьшение объема известных сведений может лишь увеличить неопределенность. Если система обеспечивает совершенную секретность, то из (7) и (8) следует, что

$$H(\mathbf{Z}) \ge H(\mathbf{X}). \qquad (9)$$

Неравенство (8) - это *граница Шеннона для совершенно секретных систем: неопределенность секретного ключа должна быть не меньше неопределенности шифруемого им текста.* Если K знаков ключа выбраны из алфавита объема L_z, то

$$H(\mathbf{Z}) \le \log(L_z^K) = K \log L_z \qquad (10)$$

где равенство выполняется тогда и только тогда, когда ключ совершенно случаен. Аналогично в

$$H(\mathbf{X}) \le M \log L_x \qquad (11)$$

(где L_x - объем алфавита открытого текста) равенство выполняется тогда и только тогда, когда открытый текст полностью случаен. Таким образом, если $L_x = L_z$ (как в шифре Вернама) и открытый текст полностью случаен, граница Шеннона (9) с учетом (10) и равенства в (11) дает

$$K \ge M \qquad (12)$$

Это означает, что ключ не должен быть короче открытого текста; нижняя граница достигается в шифре Вернама, когда длина ключа равна длине открытого текста.

Криптография с открытыми ключами.

Криптоалгоритмы с открытыми ключами известны также, как асимметричные криптоалгоритмы. Эти алгоритмы используют два ключа: один ключ используется для шифрования сообщения, а другой для его дешифрования. Ключи так математически связаны, что данные зашифрованные одним могут быть расшифрованы только другим (ему парным). Каждый пользователь имеет два ключа: *открытый ключ* и *секретный ключ*. Все пользователи помещают свои открытые ключи в открытый справочник. Поскольку открытый и секретный ключ связаны математически друг с другом, любой пользователь зашифровав открытым ключом сообщение может быть уверен, что расшифровать его сможет только обладатель секретного (парного ему) ключа. Здесь предполагается, что секретные ключи пользователей хранятся надежно и процесс их генерации выполняет каждый пользователь самостоятельно.

Хорошо известной системой с открытыми ключами является система RSA (авторы Rivest, Shamir и Adleman). Рассмотрим процесс обмена секретными сообщениями между пользователями A и B, использующими систему с открытыми ключами. Прежде всего оба пользователя создают свою пару ключей:

E_A - открытый ключ A

D_A - секретный ключ A

и

E_B - открытый ключ B

D_B - секретный ключ B

Открытые ключи E_A и E_B помещаются в открытый справочник. Далее, если A желает передать секретное сообщение M пользователю B он находит его

открытый ключ в справочнике и применяет его к тексту M. Поскольку, ключи по сути своей представляют собой некоторую функцию, то мы можем записать

$$Y = E_B(M),$$

где Y - результирующее зашифрованное сообщение M. Оно отправляется пользователю B. Для дешифрации (получения исходного сообщения M) пользователь применяет свой секретный ключ D_B к сообщению Y:

$$M = D_B(Y)$$

Пользователь B для передачи секретного сообщения M пользователю A поступает аналогично, но для шифрования использует открытый ключ пользователя A - E_A.

Система RSA.

Криптосистема RSA представляет собой блочный шифр, в котором открытый текст и шифртекст представляют собой целые числа от 0 до N-1 при некотором N. В основе системы RSA лежит функция возведения в степень в модульной арифметике, причем арифметика выполняется над составными числами.

Зная открытый текст M, модуль N и показатель степени e, можно вычислить $M^e \bmod N$. Функция возведения в степень является односторонней функцией с точки зрения извлечения как корней, так и логарифмов. При некоторых значениях N, M и e обратить эту функцию может оказаться очень сложно.

В системе RSA используется тот факт, что нахождение больших простых чисел (например, 200-значных) не требует сложных вычислений, а вот разложение произведения двух таких чисел оказывается вычислительно невозможным. Для того чтобы создать свой секретный и открытый ключи, пользователь A по случайному закону выбирает два больших простых числа P и Q, перемножив которые, он получает *двухсоставной* модуль N. В качестве своего открытого ключа он выбирает модуль N и специально выбранный показатель степени e, а в качестве своего секретного ключа - числа P и Q.

Любой человек, знающий N, способен выполнить процедуру зашифрования, основанную на возведении в степень по модулю N. Но лишь только пользователь A, которому известны числа P и Q, способен выполнить обратную процедуру, т.е. расшифрование.

Используя числа P и Q, пользователь A может вычислить значение функции Эйлера $\phi(N)$, показывающее количество положительных целых чисел от 1 до N, которые взаимно просты с N. При $N=PQ$

$$\phi(N) = (P-1)(Q-1)$$

Величина $\phi(N)$ играет большую роль в теореме Эйлера, которая говорит, что если наибольший общий делитель НОД(x,N), то

$$x^{\phi(N)} \equiv 1 (\bmod N)$$

или в несколько более общей форме

$$x^{k\phi(N)+1} \equiv x(\mathrm{mod}\, N)$$

Зная $\phi(N)$, пользователь А может вычислить [7] такое число d, что

$$e \times d \equiv 1(\mathrm{mod}\, \phi(N))$$

или, что тоже самое,

$$e \times d = k \times \phi(N) + 1$$

Если криптограмму $M^e \bmod N$ возвести в степень d, то в результате получится открытый текст M, так как

$$(M^e)^d = M^{ed} = M^{k\phi(N)+1} = M(\mathrm{mod}\, N)$$

Алгоритм открытого распространения ключей Диффи-Хеллмана.

В основе алгоритма лежит экспоненциальный ключевой обмен, который в свою очередь основан на простоте вычисления показательной функции в конечном поле Галуа $GF(q)$ состоящим из q элементов [q - простое число, (числа $\{0, 1, \ldots, q - 1\}$ по модулю q)], по сравнению с трудностями вычисления логарифмов в том же поле. Если

$$Y = \alpha^X \bmod q, \qquad 1 < X < q - 1,$$

где α - фиксированный примитивный элемент поля $GF(q)$ (т.е. степени α дают все ненулевые элементы 1, 2, …, q - 1 поля $GF(q)$), то X называется логарифмом Y по основанию α над полем $GF(q)$:

$$X = \log_\alpha Y \qquad \text{над } GF(q), \qquad 1 < X < q - 1.$$

Имея X, легко вычислить Y. Если использовать повторяющее возведение в квадрат, то потребуется самое большее $2 \times \log_2 q$ операций умножения. Например,

$$\alpha^{37} = \alpha^{32+4+1} = \left(\left(\left(\left(\alpha^2\right)^2\right)^2\right)^2\right)^2 \times \left(\alpha^2\right)^2 \times \alpha$$

С другой стороны, вычисление X из Y представляет обычно гораздо более сложную задачу [8; 9; 10].

Если q выбрано правильно, то извлечение логарифмов по модулю q потребует проведения предварительных вычислений, по объему пропорциональных

$$L(q) = e^{\sqrt{\ln q \times \ln \ln q}}$$

после осуществления которых отдельные логарифмы могут быть вычислены достаточно просто. С помощью функции $L(q)$ оценивается также время, требуемое для разложения на множители составного числа аналогичной величины.

Для того чтобы осуществить связь, пользователь А выбирает случайное число X_A равновероятно из целых чисел 1, 2, ..., q - 1. Это число X_A пользователь А держит в секрете, а пользователю В он посылает

$$Y_A = \alpha^{X_A} \bmod q .$$

Аналогично пользователь В выбирает случайное число X_B и посылает пользователю А соответствующее Y_B. Проделав это, пользователи А и В могут вычислить

$$K_{AB} = \alpha^{X_A X_B} \bmod q$$

и использовать это число в качестве ключа. Для того чтобы вычислить K_{AB}, пользователь А возводит Y_B - число, полученное от пользователя В, - в степень X_A:

$$K_{AB} = Y_B^{X_A} \bmod q = \left(\alpha^{X_B}\right)^{X_A} \bmod q = \alpha^{X_B X_A} = \alpha^{X_A X_B} \bmod q$$

пользователь В получает K_{AB} аналогичным образом:

$$K_{AB} = Y_A^{X_B} \bmod q .$$

Никто, кроме пользователей А и В, не знает ни X_A, ни X_B. Поэтому любой другой человек вынужден вычислять K_{AB}, имея только Y_A и Y_B. Эквивалентность этой проблемы проблеме вычисления дискретного логарифма есть главный нерешенный вопрос в криптографии с открытым ключом. До сих пор более легкого решения, чем взятие логарифма не выявлено.

Если q простое число длиной 1000 бит, то для вычисления Y_A из X_A потребуется порядка 2000 умножений 1000-битных чисел. С другой стороны, на данный момент, вычисление логарифмов над полем $GF(q)$ потребует более 2^{100} (или примерно 10^{30}) операций.

Арифметика экспоненциального ключевого обмена не ограничивается простыми полями. Ее можно также осуществлять в полях Галуа с 2^n элементами или в кольцах по модулю произведения простых чисел [11, 12].

Криптографические протоколы.

Как показали теоретические и практические исследования простого наличия стойких криптоалгоритмов недостаточно для решения предписанных им задач. Криптоалгоритм должен использоваться в рамках набора правил или процедур, называемого *протоколом,* который гарантирует, что данный алгоритм действительно будет обеспечивать безопасность или аутентификацию, требуемые в

системе. Поэтому разработка системы для обеспечения секретности или целостности данных в действительности опирается на две области исследований: разработку стойких криптоалгоритмов и разработку надежных протоколов.

Известен ряд протоколов, в которых наиболее стойкая система RSA, теряет свои высокие свойства стойкости. Например:
несостоятельность протокола с общим абсолютным значением, несостоятельность протокола с малым значением показателя степени [5], несостоятельность протокола, использующего Китайскую теорему об остатках для ускорения вычисления модульной экспоненты [6].

Трехэтапный протокол Шамира.

Один из наиболее интересных криптографических протоколов, описанный А. Шамиром в неопубликованной работе, показывает, что секретность можно обеспечить, не распространяя предварительно ни секретных, ни открытых ключей. В этом протоколе предполагается, что между двумя пользователями существует средство связи (такое, как "бесшовная" волоконно-оптическая линия или заслуживающий доверия, но очень любопытный почтальон), которое не позволяет осуществлять имитацию или подмену сообщения, однако дает противнику возможность читать все сообщения, проходящие по каналу связи. Кроме того, используется криптосистема с секретными ключами, а шифрующая функция $E_Z(\bullet)$ *коммутативна*, поэтому для любого открытого текста x и ключей z_1 и z_2

$$E_{z_2}\left(E_{z_1}(x)\right) = E_{z_1}\left(E_{z_2}(x)\right) \qquad (13)$$

т.е. результат двукратного шифрования не зависит от того, в каком порядке используются ключи z_1 и z_2. Таким свойством обладают многие шифры. Например, под это определение подходит шифр с ключом однократного применения (4), так как $\left(x \oplus z_1\right) \oplus z_2 = \left(x \oplus z_2\right) \oplus z_1$, где сложение битов производится по модулю 2.

Рассмотрим протокол как описывает его Шамир:

1) Пользователи А и В случайно выбирают личные секретные ключи Z_A и Z_B.

2) Когда пользователь А желает послать секретное сообщение X пользователю В, он зашифровывает X с использованием ключа Z_A и посылает полученную криптограмму $Y_1 = E_{Z_A}(X)$ по открытому и защищенному от имитации и подмены каналу пользователю В.

3) Пользователь В, приняв Y_1, считает Y_1 открытым текстом и шифрует его с использованием своего ключа Z_B. Он посылает полученную криптограмму $Y_2 = E_{Z_B}(Y_1) = E_{Z_B}(E_{Z_A}(X))$ по открытому и защищенному от имитации и подмены каналу пользователю А.

4) Пользователь А, приняв Y_2, расшифровывает Y_2 с помощью своего ключа Z_A. В соответствии с коммутативным свойством (13) это снимает предыдущее зашифрование с использованием Z_A, в результате чего получается $Y_3 = E_{Z_B}(X)$. Далее пользователь А посылает Y_3 по открытому и защищенному от имитации и подмены каналу пользователю В.

5) Пользователь B, приняв Y_3, расшифровывает Y_3 с использованием своего ключа Z_B и получает секретное сообщение X, переданное A.

Какой шифр с секретными ключами можно использовать в этом протоколе? Пригоден ли в этом случае шифр с ключом однократного применения, обеспечивающий совершенную секретность? При его использовании три криптограммы принимают вид

$$\begin{aligned} Y_1 &= X \oplus Z_A, \\ Y_2 &= X \oplus Z_A \oplus Z_B, \\ Y_3 &= X \oplus Z_B \end{aligned} \qquad (14)$$

Криптоаналитик наблюдает все три криптограммы и поэтому может вычислить

$$Y_1 \oplus Y_2 \oplus Y_3 = X,$$

где использован тот факт, что две одинаковые величины при сложении по модулю 2 дают 0. Итак, при использовании шифра с ключом однократного применения трехэтапный протокол совершенно ненадежен! Причиной этого, как видно из (14), является то свойство протокола, что каждый шифр используется в нем "полтора раза", тогда как рассмотренный шифр надежен лишь при однократном применении.

Существует ли шифр, обеспечивающий секретность при его использовании в протоколе Шамира? Видимо, да. Пусть p - большое простое число, и p - 1 имеет большой простой множитель (что делает решение задачи нахождения дискретного логарифма по модулю p вычислительно сложным). Случайно выберем положительное целое число e, меньшее p-1 и такое, что НОД(e,p-1)=1, а также d-положительное целое число, меньшее p-1 и такое что

$$de = 1 (\mathrm{mod}\ p-1) \qquad (15)$$

Пусть z=(d,e) - секретный ключ, а функция зашифрования и расшифрования имеют вид

$$\begin{aligned} y &= E_Z(x) = x^e (\mathrm{mod}\ p), \\ x &= D_Z(y) = y^d (\mathrm{mod}\ p), \end{aligned} \qquad (16)$$

где x и y- целые числа, меньшие p. (Соотношение $y^d = x^{de} = x (\mathrm{mod}\ p)$ легко следует из теоремы Ферма и равенства, эквивалентного (15): de=Q(p-1)+1 для некоторого Q.) Как следует из (16),

$$(x^{e_1})^{e_2} = x^{e_1 e_2} = x^{e_2 e_1} (\mathrm{mod}\ p)$$

и поэтому этот шифр коммутативен. При его использовании в трехэтапном протоколе три криптограммы принимают вид

$$y_1 = x^{e_A} (\mathrm{mod}\ p),$$

$$y_2 = x^{e_A e_B} (\mathrm{mod}\ p), \qquad (17)$$
$$y_3 = x^{e_B} (\mathrm{mod}\ p).$$

Не трудно видеть, что для компрометации протокола теперь противнику необходимо найти e_B, что равносильно решению задачи дискретных логарифмов (по модулю p).

Защита электронных транзакций. Дайджест сообщения. Цифровая подпись. Электронный конверт.

При передаче электронных транзакций необходимо решать по крайней мере две задачи: защиту от несанкционированного доступа (шифрование данных транзакции) и аутентификацию данных. Аутентификация гарантирует целостность данных и авторство транзакции. Обычно эти задачи решаются в комплексе с помощью применения соответствующих криптоалгоритмов.

Дайджест сообщения.

Для формировании цифровой подписи документа обычно создается так называемый *дайджест сообщения* (message digest), который представляет собой *свёртку* исходного сообщения с помощью специальной хеш-функции. Длина дайджеста с одной стороны намного меньше чем возможные исходные сообщения, а с другой стороны такова, что полный перебор возможных значений является практически невыполнимым. Например длина дайджеста порождаемого алгоритмами Ривеста - MD2, MD4 и MD5 равняется 128 битам, а алгоритмом SHA - 160 бит.

Хеш-функции, создающие дайджест сообщений, обладают свойством односторонности. Это означает, что если мы имеем дайджест

$$h = H(T)$$

где H- хеш-функция, T- исходный текст, то восстановить аргумент T, если известно значение h с вычислительной точки зрения практически невозможно. Хеш - функция должна удовлетворять следующим условиям:

1) исходный текст может быть произвольной длины;

2) само значение $H(T)$ имеет фиксированную длину;

3) значение функции $H(T)$ легко вычисляется для любого аргумента;

4) восстановить аргумент по значению с вычислительной точки зрения практически невозможно;

5) функция $H(T)$ - однозначна;

Из этого определения следует, что для любой хеш-функции есть тексты-близнецы, имеющие одинаковые значения хеш-функций, так как мощность множества аргументов неограниченно больше мощности множества значений. Поэтому немаловажным свойством хеш-функций является её хорошее стохастическое *перемешивание*.

Цифровая подпись.

В конце любого письма или документа исполнитель или ответственное лицо обычно ставит свою подпись. Это обычно преследует две цели. Во-первых, получатель имеет возможность убедиться в истинности письма, сличив подпись с имеющимся у него образцом. Во-вторых, личная подпись является юридическим гарантом авторства документа. Последний аспект особенно важен при заключении разного рода сделок, составлении доверенностей, обязательств и т.д.

Для подписи электронных документов используется так называемая *цифровая подпись*.

Примером простой реализации цифровой подписи является *нотариальный протокол*. Рассмотрим такой протокол. Он предназначен для того, чтобы сообщение заверялось объектом А таким способом, который позволял бы другим проверять, что данное сообщение на самом деле было заверено А. Поскольку в таком случае А ведет себя по аналогии с нотариусом, то данный протокол называется *нотариальным протоколом*[5].

Чтобы установить нотариальный протокол, А должен выбрать параметры криптосистемы RSA: простые числа p и q и показатели степени зашифрования и расшифрования e и d, удовлетворяющие условию $ed=1 \bmod \phi(n)$, где $n=pq$. Значение n, так же как и открытый показатель степени e, делаются несекретными, тогда как d и разложение n на множители хранится А в секрете.

Для того чтобы заверить документ M, нотариус использует секретный показатель степени для вычисления сигнатуры

$$S = M^d \bmod n$$

Затем сигнатура вносится в документ, так же как при скреплении нотариальной печатью бумажного документа. Любой человек может воспользоваться открытой информацией для проверки того, что

$$S^e = M \bmod n$$

Поскольку только А знает значение d, используемое для получения S, для данного протокола считается доказанным, что только А может вычислить сигнатуру S.

Однако, как указывается в работах [13, 14], протокол можно использовать для получения поддельной сигнатуры на документе. Правда, для этого необходимо выполнить определенные условия, а именно: необходимо легально подписать специально заготовленное сообщение. Все попытки нарушения такого протокола основаны на том факте, что RSA использует функцию, которая сохраняет мультипликативную структуру входа

$$\left(XM\right)^d = X^d M^d \bmod n$$

Это означает, что любая модификация протокола с целью предупреждения таких попыток нарушения должна будет лишить фальсификаторов возможности использования такой мультипликативной структуры.

Поэтому для получения действительно стойкой цифровой подписи рекомендуется использовать более сложные протоколы и алгоритмы.

Цифровая подпись на основе алгоритма RSA.

Пусть пользователи A и B обладают своими RSA ключами - D_A, E_A и D_B, E_B соответственно, причем ключи D являются секретными, а ключи E - открытыми. Если A желает отправить подписанное сообщение M пользователю B он выполняет следующие действия:

1) вычисляет дайджест сообщения с помощью одной из хеш-функций (например MD5)

$$h = MD5(M)$$

2) с помощью своего секретного ключа вычисляет значение

$$Y = D_A(h)$$

3) далее он вычисляет с помощью открытого ключа пользователя B

$$S = E_B(Y)$$

Полученное значение S является цифровой подписью документа M и передается вместе с ним пользователю B.

Для проверки подписи пользователь B выполняет следующие действия:

1) аналогично, как и пользователь A вычисляет дайджест сообщения с помощью одной из хеш-функций (например MD5)

$$h = MD5(M)$$

2) с помощью своего секретного ключа вычисляет

$$Y = D_B(S)$$

3) с помощью открытого ключа пользователя A вычисляет

$$h' = E_A(Y)$$

4) сравнивает значения h и h'; если они совпадают, то подпись принимается (считается подлинной); иначе подпись отвергается.

Не трудно видеть, что данная схема позволяет защищаться от нескольких видов нарушений.

Пользователь A не может отказаться от своего сообщения, если он признает, что секретный ключ известен только ему.

Нарушитель без знания секретного ключа не может ни сформировать, ни сделать осмысленное изменение сообщения, передаваемого по линии связи.

Данная схема позволяет также при решении многих конфликтных ситуаций обходится без посредников.

Стандарт цифровой подписи DSS (Digital Signature Standard)

Этот стандарт определяет алгоритм цифровой подписи DSA (Digital Signature Algorithm), который может использоваться для генерации цифровой подписи. Цифровые подписи используются для выявления неавторизованных модификаций данных и аутентификации идентичности пользователя генерирующего подпись. В дополнение, получатель подписанных данных может использовать цифровую подпись в доказательстве третьей стороне того, что подпись действительно сгенерирована стороной, подписавшей данные. Это обеспечивает невозможность отречения от подписи лица подписавшего данные.

Пояснение: Этот стандарт определяет Digital Signature Algorithm (DSA) предназначенный для приложений требующих цифровых подписей. DSA цифровая подпись представляет собой пару больших чисел, представляемых в компьютере в виде двоичных цифр. Цифровая подпись вычисляется на основе набора правил (т.е. алгоритма DSA) и набора параметров, которые могут быть использованы для проверки идентичности подлинника и целостности данных. DSA включает генерацию подписи и ее проверку. Генерация использует секретный (private) ключ для получения цифровой подписи. Проверка подписи использует открытый (public) ключ, который соответствует секретному ключу, использованному при генерации подписи, но не равный ему.

Каждый пользователь обладает парой ключей: секретным и открытым. Предполагается, что открытые ключи известны всем членам группы пользователей, либо вообще доступны всем. Секретные ключи должны знать только их создатели (владельцы). Любой может проверить подпись пользователя с помощью применения его открытого ключа. Генерация подписи может быть осуществлена только владельцем секретного ключа.

При генерации подписи для получения сжатой версии данных, называемой сверткой сообщения (message digest) используется хеш-функция. Свертка сообщения подписывается. Цифровая подпись отсылается получателю вместе с подписанными данными (часто называемыми сообщением). Получатель сообщения и подписи проверяет подпись, используя открытый ключ отправителя. В процессе проверки подписи должна использоваться аналогичная хеш-функция, что и при подписывании.

Применимость: Этот стандарт применим для всех государственных подразделений и агентств для защиты и аутентификации информации. Этим стандартом должны руководствоваться все государственные подразделения оперирующие цифровой подписью на основе систем с открытыми ключами (public-key based systems).

Поощряется принятие и применение этого стандарта частными и коммерческими организациями.

Применения: DSA аутентифицирует целостность подписанных данных и идентичность подписывающего. DSA может также применяться в доказательстве третьей стороне того, что данные были подписаны тем, кто сгенерировал подпись. DSA предназначен для использования в электронной почте, передачи электронных платежей, обмена данными, распространения программного обеспечения, хранения данных и других применений, которые требуют доверия к целостности и аутентификации подлинности.

Исполнение: DSA может иметь программное, микропрограммное и аппаратное исполнение.

Ограничения: Стойкость системы цифровой подписи в значительной степени зависит от защищенности пользовательских секретных ключей. Поэтому пользователи должны хорошо защищать свои секретные ключи от неавторизованного доступа к ним.

Реализация DSS (Выполнена в компании DEKART S.R.L.)

DSS (Digital Signature Standard) - стандарт электронной подписи предложен американским национальным институтом стандартов и технологий (American National Institute of Standards and Technology (NIST)) - по рекомендации национального агентства безопасности (National Security Agency (NSA)).

В основе DSS лежит алгоритм *цифровой сигнатуры* (свертки сообщения) SHA (Secure Hash Algorithm) и Digital Signature Algorithm (DSA) в основе, которого лежит криптосистема Эль-Гамаля. Алгоритм SHA используется для выработки цифровой сигнатуры сообщения, которая в свою очередь подписывается с помощью алгоритма Эль-Гамаля и присоединяется к сообщению.

Стойкость системы в целом основана на сложности нахождения логарифмов в конечных полях (полях Галуа).

Рассмотрим положения и математические методы, которые лежат в основе системы. Прежде всего необходимо отметить, что алгоритм Эль-Гамаля представляет собой криптосистему с *открытым ключом.* Это означает, что ряд параметров (ключей) системы являются открытыми (public), но в тоже время она предполагает наличие секретного ключа (private), который должен надлежащим образом храниться и применяться.

Таким образом параметры системы разделяются на три группы:

- общие (common) параметры;
- секретный (private) ключ;
- открытый (public) ключ.

Общие параметры необходимы для функционирования системы в целом. Секретный ключ используется для формирования электронной подписи сообщения. Открытый ключ применяется для проверки подлинности электронной подписи. Таким образом подписать сообщение может только тот кто владеет секретным ключом, а проверить подлинность подписи может любой, кто этого пожелает (при наличии у него общих параметров и открытого ключа).

Общие параметры системы представляют тройку $<p,q,g>$, где p - простое число (модуль) удовлетворяет неравенству

$$2^{511} < p < 2^{512}. \quad (1)$$

q- простой делитель p - 1 и удовлетворяет неравенству

$$2^{159} < q < 2^{160}. \quad (2)$$

g - генератор, удовлетворяющий условию

$$g = h^{\frac{p-1}{q}} \bmod p, \quad (3)$$

где h- любое целое из $0 < h < p$ и такое, что

$$h^{\frac{p-1}{q}} \bmod p > 1 \quad (4)$$

Эти значения делаются общими, т.е. объявляются всем участникам, обменивающимися подписанными сообщениями.

Секретный ключ x случайно выбирается каждым пользователем из диапазона $\lfloor 1, q \rfloor$ и держится в секрете. В нашей реализации DSS для этого он зашифровывается алгоритмом DES с помощью ключа (password) пользователя и в таком виде хранится в файле user_ident.SEC.

Открытый ключ y вычисляется следующим образом

$$y = g^x \bmod p \tag{5}$$

и в нашей реализации хранится в файле PUBLIC.KEY.

Другие параметры: m - сообщение подлежащее подписыванию и передаче. k- случайное число такое, что $0 < k < q$. H- односторонняя хеш-функция.

Параметр k также является секретным и меняется от одной подписи к другой.

Криптографическая стойкость системы зависит от размера параметров p и q (в нашем случае это 512 и 160 бит соответственно). Причем со стороны q может быть возможна только *силовая атака*, т.е. полный перебор и в нашем случае криптостойкость по параметру q равна 2^{160}. Со стороны параметра p нападение равносильно нахождению логарифма в поле Галуа $GF(2^{512})$ и извлечение логарифмов по модулю p потребует проведения предварительных вычислений, по объему пропорциональных

$$L(p) = e^{\sqrt{\ln p \cdot \ln \ln p}}, \tag{6}$$

после осуществления которых отдельные логарифмы могут быть вычислены достаточно просто, но объем этих предварительных вычислений, как видно из выражения (5) настолько большой, что это не представляется возможным.

И, наконец, для того чтобы получатель мог доказать подлинность сообщения третьему лицу к сообщениям приписываются так называемые *цифровые сигнатуры*. Цифровая сигнатура - это массив данных, зависящий как от идентификатора отправителя, так и содержания сообщения. В DSS для формирования цифровой сигнатуры используется алгоритм SHA (Secure Hash Algorithm) и длина этой сигнатуры равна 160 бит.

Генерация цифровой подписи.

Процесс вычисления цифровой подписи для сообщения m состоит из следующих этапов:

1. Вычисление цифровой сигнатуры на основе алгоритма *SHA* (здесь мы заменили обозначение H на SHA)

$$h = SHA(m) \tag{7}$$

2. Выбор случайного числа k из диапазона $\lfloor 1, q \rfloor$ и вычисление

$$r = (g^k \bmod p) \bmod q \tag{8}$$

3. Вычисление

$$s = (k^{-1} \cdot (h + xr)) \bmod q, \tag{9}$$

где k^{-1} мультипликативная инверсия k по модулю q; т.е., $(k^{-1} \cdot k) \bmod q = 1$ и $0 < k^{-1} < q$.

Значения r,s являются цифровой подписью сообщения m и передаются вместе с ним (в нашей реализации это файл с расширением .DSS и именем таким же, как и у файла содержащего подписываемое сообщение).

Проверка цифровой подписи.

Процесс проверки электронной подписи производится следующим образом:

пусть m', r' и s' полученная версия значений m, r и s, соответственно и пусть y будет открытым ключом пользователя передавшего подписанное сообщение. Для проверки подписи получатель, прежде всего должен проверить что $0 < r' < q$ и $0 < s' < q$; если хотя бы одно из этих условий нарушено подпись отвергается. Далее:

1. Вычисляются значения

$$w = s'^{-1} \bmod q \tag{10}$$

$$u1 = (H(m')w) \bmod q \tag{11}$$

$$u2 = ((r')w) \bmod q \tag{12}$$

$$v = ((g^{u1} \cdot y^{u2}) \bmod p) \bmod q \tag{13}$$

2. Выполняется проверка равенства

$$v = r' \tag{14}$$

Если равенство (14) выполняется то делается вывод, что подпись верная и сообщение аутентично. В противном случае сообщение должно быть отвергнуто, поскольку в этом случае цифровая подпись не соответствует сообщению.

Электронный конверт.

Как правило реализации симметричных алгоритмов шифрования (DES, IDEA и т.п.) являются достаточно быстрыми в отличие от реализаций алгоритмов с открытыми ключами (несимметричных). Поэтому при передаче сообщений (транзакций) есть смысл использовать их совместно: симметричные для шифрования сообщения (транзакции), а несимметричные для шифрования секретного ключа (симметричного алгоритма) и подписи сообщения (транзакции). Такая схема (Рис.2.) использует электронный документ, который получил название *электронный конверт*.

Исходное сообщение зашифровывается с помощью симметричного алгоритма и секретного ключа (случайного), затем секретный ключ также зашифровывается с помощью RSA и открытого ключа корреспондента, для которого предназначено сообщение.

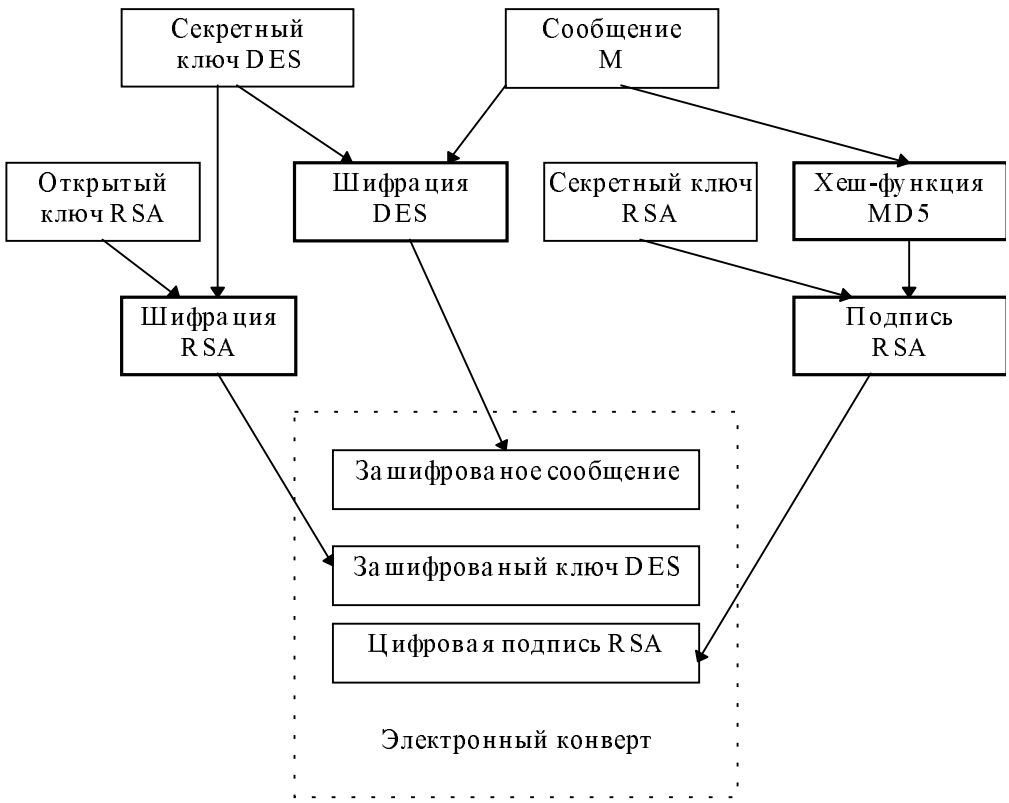

Рис.2. Структурная схема формирования электронного конверта.

Также исходное сообщение подписывается с помощью RSA и секретного ключа отправителя. Зашифрованное сообщение, зашифрованный ключ симметричного алгоритма и цифровая подпись документа помещаются в один электронный конверт (документ) и отправляются тому для которого он предназначен.

Пользователь, получивший электронный конверт, расшифровывает секретный ключ симметричного алгоритма с помощью своего секретного ключа RSA. Далее, с помощью этого ключа он расшифровывает само сообщение, вычисляет хеш-функцию и проверяет цифровую подпись с помощью открытого ключа отправителя электронного конверта.

В качестве алгоритмов шифрования и цифровой подписи рекомендуется использовать хорошо себя зарекомендовавшие алгоритмы DES, IDEA, RSA, DSA.

Аутентификация данных на картах. Статическая и динамическая аутентификация.

Наиболее важным аспектом применения смарт-технологий является процесс аутентификация данных. Аутентификация предназначена для подтверждение подлинности карт, терминального оборудования, а также транзакций.

Различают два типа аутентификации: статическую и динамическую. Динамическая аутентификация является более криптостойкой в отличие от статической, но по сравнению с последней требует более сложных протоколов и более "умных" смарт-карт.

Статическая аутентификация.

Для обеспечения статической аутентификации применяется цифровая подпись. Рассмотрим пример статической аутентификации на основе алгоритма RSA.

Эмитент создает RSA ключи: E_I - секретный; D_I - открытый. Секретный ключ хранится в секрете, а открытый ключ помещается в платежные терминалы. В процессе персонализации карт на них записываются постоянные данные о владельце, сороке окончания работы карты, номер карты и др. Из этих данных путем конкатенации отдельных значений создается значение X. Далее вычисляется цифровая подпись $S = E_I(X)$, которая и является аутентификационным значение данной карты. Это значение также помещается на карту.

При обслуживании карты в платежной точке терминал, получив с карты значения X и S, вычисляет $X' = D_I(S)$ и выполняет проверку равенства X=X'. Если равенство выполняется то считается, что карта является аутентичной и она принимается к обслуживанию. Если же равенство не выполняется то карта отвергается, поскольку она является неправильной.

Достоинством такой аутентификации является ее простота, а недостатком сложность создания много-эмитентных платежных систем, поскольку открытые ключи **всех эмитентов** должны "знать" **все платежные терминалы.** Даже если это можно обеспечить, то при появлении нового эмитента возникают значительные трудности в рассылке его открытого ключа всем терминалам. Кроме этого ключ эмитента в этой схеме не сертифицирован и это несколько её ослабляет.

Другая схема статической аутентификации (Рис.3.) устраняет последний недостаток, но несколько усложняет протокол.

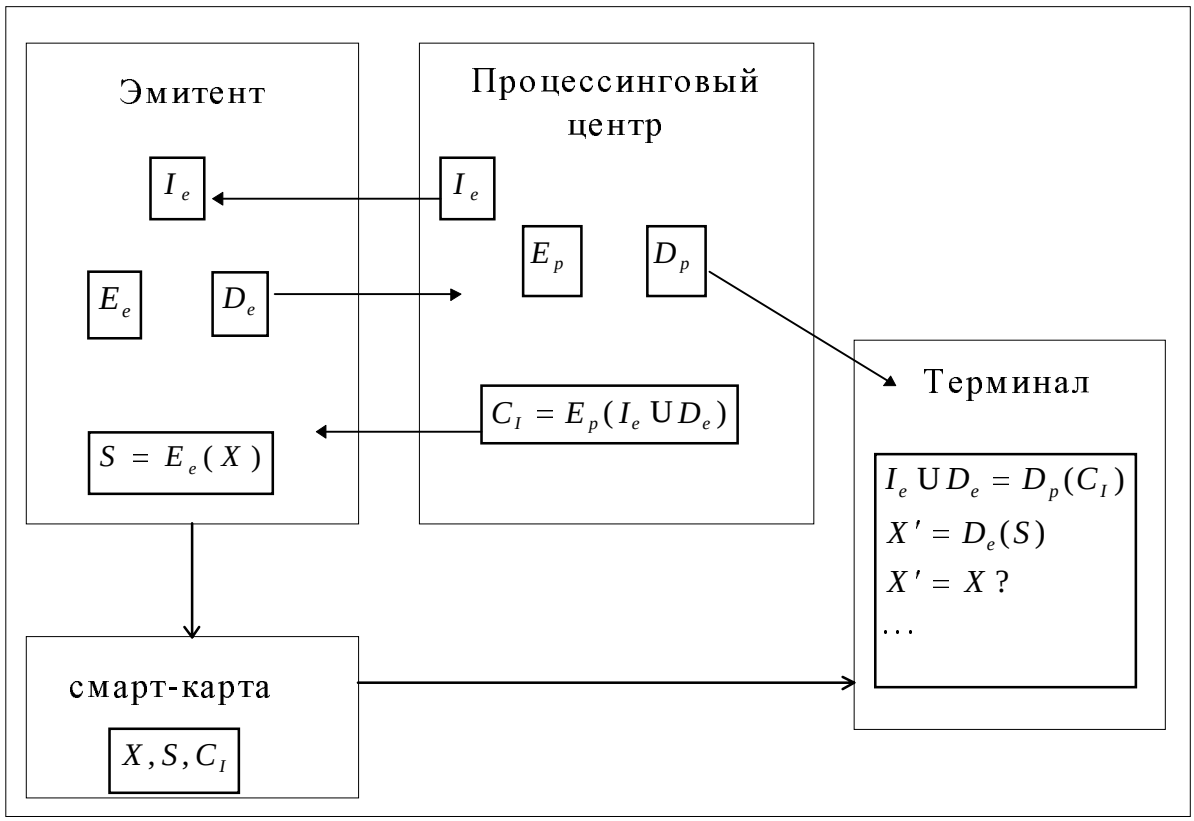

Рис.3. Статическая аутентификация с сертификацией открытого ключа эмитента.

Данная схема аутентификации позволяет создавать много-эмитентные платежные системы. Каждый эмитент создает свою пару ключей RSA (E_e, D_e). Открытые ключи D_e передаются в процессинговый центр для сертификации. Процессинговый центр выделяет каждому эмитенту идентификатор I_e, который конкатенируется с открытым ключом соответствующего эмитента, а затем ключ и идентификатор подписываются секретным ключом процессингового центра $C_I = E_p(I_e \cup D_e)$. Полученный сертификат передается эмитенту. При персонализации карт эмитент подписывает данные карты своим секретным ключом $S = E_e(X)$ и полученную цифровую подпись S вместе с сертификатом своего открытого ключа помещает на карту.

Процессинговый центр снабжает все терминалы своим открытым ключом.

При обслуживании карты терминал с помощью открытого ключа процессингового центра получает идентификатор эмитента и его открытый ключ $(I_e \cup D_e) = D_p(C_I)$. Затем с помощью открытого ключа эмитента он вычисляет $X' = D_e(S)$. Если $X'=X$, то терминал принимает карту к обслуживанию иначе он ее отвергает.

В данной схеме каждая из сторон (эмитент, процессинговый центр, терминал) знает только открытые чужие ключи. Секретные ключи известны только их

владельцам. Это придает особую силу этой схеме и она на настоящий момент является наиболее предпочтительной.

Динамическая аутентификация.

Динамическая аутентификация увеличивает стойкость протоколов обмена данными между терминалом и картой.

Она предполагает, что карта наделена достаточно высоким интеллектом. В частности, карта должна уметь выполнять шифрование с помощью алгоритмов DES либо RSA.

На Рис.4. и Рис.5. изображены схемы так называемых внешней и внутренней аутентификации. В этих схемах в протоколе аутентификации участвуют две карты: карта пользователя и так называемая мастер карта. Мастер карта постоянно находится в терминале и может быть физически недоступна из вне.

В этой карте хранится мастер-ключ, который недоступен из внешнего мира и попадает на нее при персонализации. Обычно внешняя и внутренняя аутентификации имеют различные мастер- ключи.

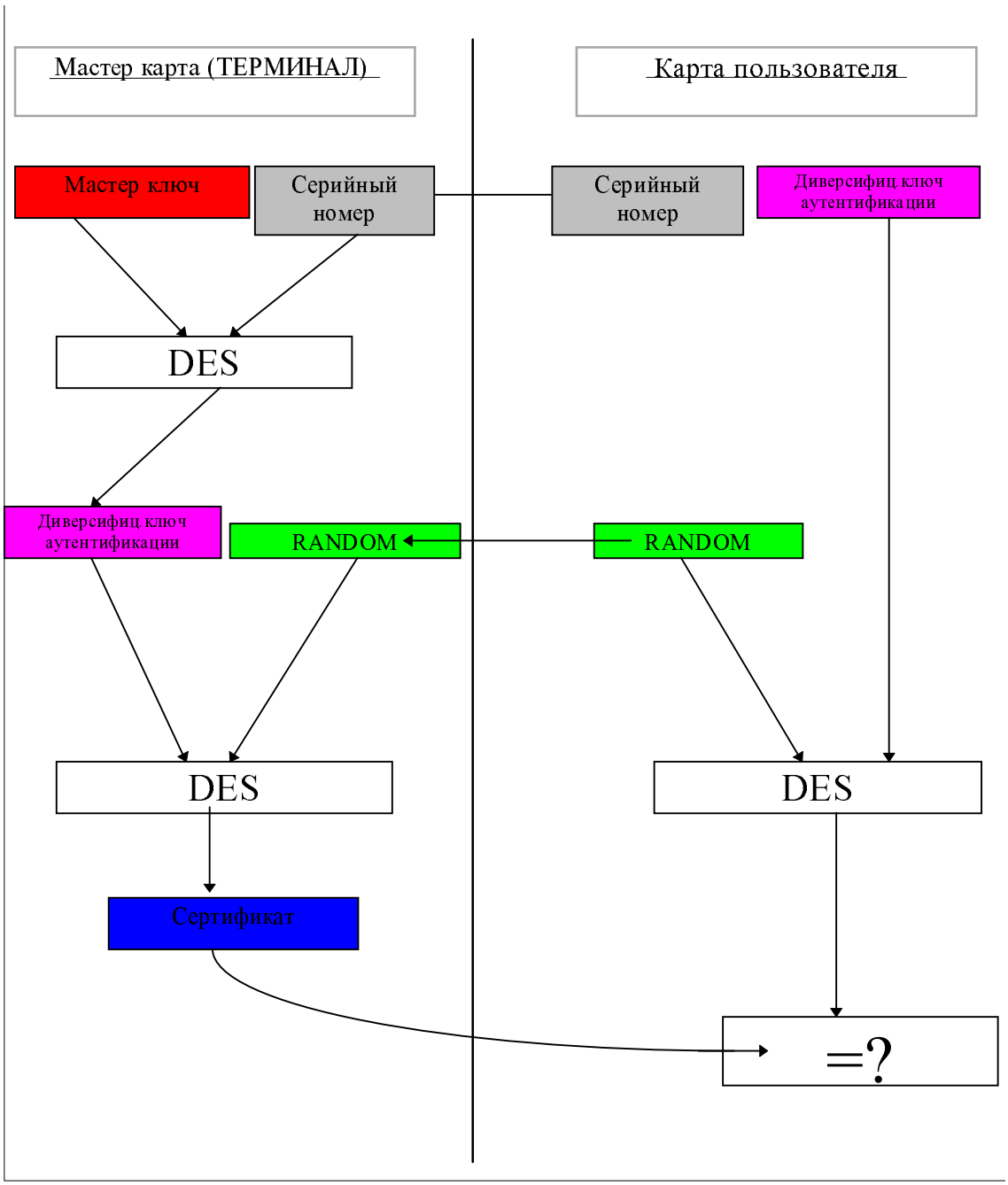

Рис.4. Внешняя аутентификация. Карточка пользователя проверяет правильный ли терминал.

В приведенных алгоритмах динамической аутентификации в обмене сообщениями между картой пользователя и терминалом присутствует параметр RANDOM, который представляет собой случайные данные. Это позволяет сделать неэффективной атаку на протокол, основанную на прослушивании линии, поскольку в каждом сеансе это значение будет другим.

Данные протоколы предполагают также, что карта пользователя "умеет" шифровать с помощью алгоритма DES.

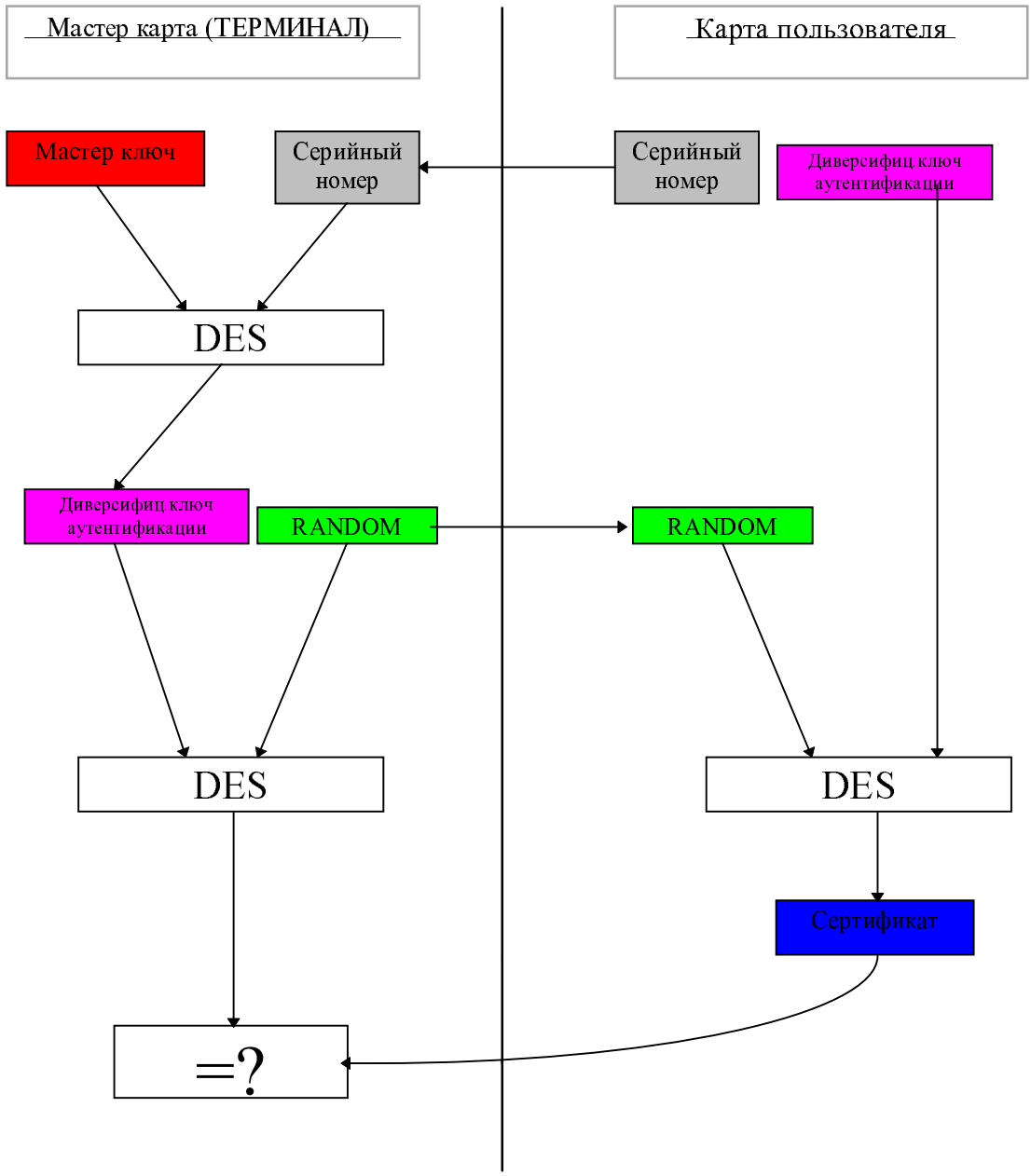

Рис.5. Внутренняя аутентификация. Терминал проверяет правильна ли карта пользователя.

Карты, работающие на основе приведенных протоколов (Рис.4, Рис.5) можно отнести к картам средней сложности (среди асинхронных). Карты высшей сложности "умеют" выполнять шифрование на основе алгоритма RSA и имеют повышенный объем памяти.

Схема динамической аутентификации на основе карт с RSA приведена на Рис.6.

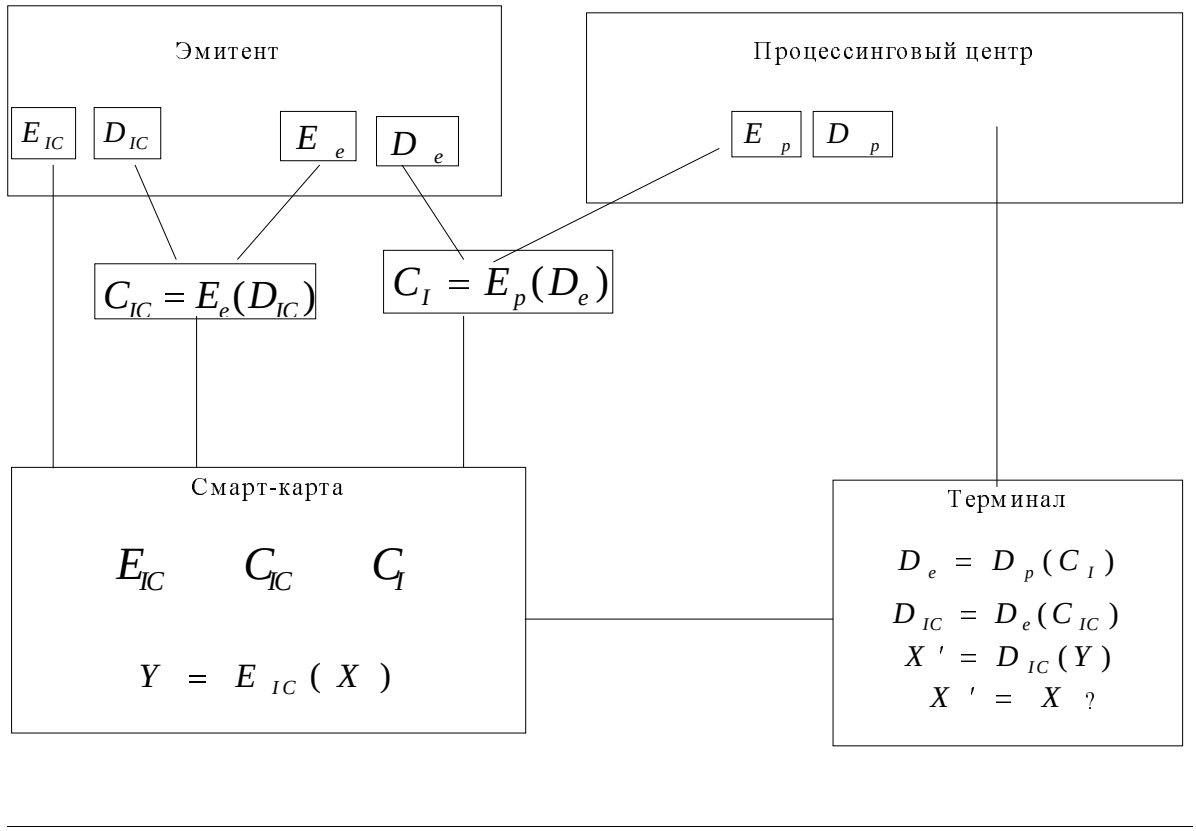

Рис.6. Схема динамической аутентификации на основе карт с RSA алгоритмом.

Проблемы криптографических протоколов. Последние достижения в криптоанализе.

Как показывают теоретические исследования и практика самым слабым местом в любой системе, использующей криптографию являются протоколы. Даже самые стойкие криптоалгоритмы при их неправильной реализации или неправильном использовании могут быть скомпрометированы.

Например, как уже отмечалось выше, что при определенных условиях существуют методы подделки подписи в нотариальном протоколе.

Известны также методы [6] компрометации алгоритма RSA, в реализации которого для ускорения вычислений модульной экспоненты используется Китайская теорема об остатках.

Нами было обнаружено, что в алгоритме криптографического преобразования данных ГОСТ 28147-89 имеются короткие циклы [15] и поэтому он не может быть применен в отдельных протоколах. Например, в режиме обратной связи по выходу (OFB).

Это говорит о том, что хорошим криптоалгоритмам должны соответствовать не менее хорошие протоколы и их системные реализации [16].

Что касается самих криптоалгоритмов то, например для DES существуют оценки стоимости *силовой атаки,* основанной на применении самых современных чипов ASIC (Application-Specific Integrated Circuits), которые следующие

Кто атакует	Бюджет	40 бит	56 бит	Надежный
Отдел корпорации	$300,000	18 с	3 ч	60 бит
Большая компания	$10,000,000	0.005 с	6 мин	70 бит
Федеральное Агентство	$300,000,000	0.0002 с	12 с	75 бит

Как видно из таблицы: достаточно увеличить длину ключа DES до 75 бит и он становится абсолютно надежным.

Такое увеличение длины ключа обычно и делается в смарт-картах путем использования DES в режимах CBC (сцепление блоков шифра) и EDE (зшифровать-расшифровать-зашифровать).

Для «взлома» ключей системы RSA как известно необходимо разложить модуль N на простые множители. Из-за большой популярности этой системы в настоящее время в эту области криптоанализа работает очень много ученых и необходимо отметить не без успеха. Например А.Ленстра и М.Манассе из «Bellcore» подключив 1600 компьютеров в Internet разложили 129 значный модуль за восемь месяцев. Стоимость подобного проекта по оценкам M.J.B.Robshaw из RSA Laboratories [18] в 1997 году будет составлять порядка $1,000,000.

Поэтому EMV рекомендует использовать RSA ключи с длиной
верхние границы, которой приводятся в следующей таблице

Принадлежность ключа	Максимальная длина
Аутентификация (модуль)	248 байт
Ключ эмитента (модуль)	247 байт
Ключ карты (модуль)	128 байт

С точки зрения практической стойкости такие длины ключей обеспечиают надлежащую стойкость вплоть до 2005 года [18, 19].

Литература.

1. D. Kahn, The Codebreakers, The Story of Secret Writing, abridget ed. New York, NY: Signet, 1973.
2. Шеннон К. Э. Теория связи в секретных системах. В кн.: Шеннон К. Э. Работы по теории информации и кибернетике. М.: ИЛ, 1963, с.243-332ю.
3. W.Diffie and M.E.Hellman, "New directions in cryptography," IEEE Trans. Informat. Theory, vol. 1 T-22, pp.644-654, Nov. 1976.
4. Шеннон К.Э. Математическая теория связи. В кн.: Шеннон К.Э. *Работы по теории информации и кибернетике*. М.: ИЛ, 1963, с.243-332.
5. Judy H. Moore, "Protocol Failures in Cryptosystems," IEEE Trans. Informat. Theory, vol. 5 T-76, pp.594-602, May 1988.
6. Marc Joye and Jean-Jacques Quisquater, "Attacks on systems using Chinese remaindering." Internet: www.dice.ucl.ac.be/crypto/ techreports.html
7. Кнут Д. Искусство программирования для ЭВМ, т.2, М.: Мир, 1977.
8. R. Silver, "The computation of indices modulo P," Mitre Corporation, Working Paper WP-07062, p.3, May 7, 1964.
9. S. C. Pohlig and M.E.Hellman, "An improved algorithm for computing logarithms in GF(p) and its cryptographic significance," IEEE Trans. Informat. Theory, vol. IT-24, pp.106-110, Jan. 1978.
10. D. Coppersmith, A. M. Odlyzko, and R. Schroeppel, "Discrete logarithms in GF(p)," Algorithmica, vol.1, pp. 1-16, 1986.
11. Z. Shmuely, "Composite Diffie-Hellman public-key generating systems are hard to break," Computer Science Departmant, Technion, Haifa, Israel, Technical Rep. 356, Feb. 1985.
12. K. S. McCurley, "A key distribution system equivalent to factoring, "Department of Mathematics, University of Southern California, June 3, 1987.
13. G. I. Davida, "Chosen signature cryptanalysis of the RSA (MIT) public key cryptosystem," Tech. Rep. TR-82-2, Dept. of Electrical Engineering and Computer Science, Univ. of Wisconsin, Milwaukee, WI, Oct. 1982.
14. D. E. Denning, "Digital signatures with RSA and other public-key cryptosystems," in Comm. of the ACM, vol.27, pp. 388-392, Apr. 1984.
15. В. Олейник, "Циклы в алгоритме криптографического преобразования данных ГОСТ 28147-89", (в печати: Сборник трудов Молдавского отделения МАИ).
16. В. Олейник, "К вопросу о криптостойкости смарт-технологий", Банки и Финансы, Инф. аген.: "ИНФОТАГ", №12 (20), декабрь 1996, с.79-83.
17. А. Л. Чмора, "Безопасность в W3", инф.-мет. журнал Защита информации "Конфидент", №4, июль-август 1996, с.21-37.
18. M.J.Robshaw, «Security Estimates for 512-bit RSA», RSA Laboratories, June 29, 1995.
19. Andrew M. Odlyzko, «The future of integer factorization», AT&T Bell Laboratories, July 11, 1995.