УДК 681.3.06:519.248.681

И.Д. ГОРБЕНКО,д-р.техн.наук, Д.С.БАЛАГУРА

СХЕМЫ НАПРАВЛЕННОГО ШИФРОВАНИЯ В ГРУППАХ ТОЧЕК НА ЭЛЛИПТИЧЕСКИХ КРИВЫХ

Введение

При информационном взаимодействии часто возникает задача зашифрования информации таким образом, чтобы расшифровать её мог только получатель. Эту задачу успешно решают схемы направленного шифрования, суть которых состоит в том, что информация шифруется на открытом ключе получателя либо на ключе, полученном с помощью открытого ключа, а расшифровывается на секретном ключе получателя, либо на ключе, полученном с помощью секретного ключа. Сейчас известно несколько схем направленного шифрования. Основными проблемами в этом классе криптографических преобразований является невысокая скорость таких преобразований, а также уменьшающаяся стойкость вследствие развития математических методов и средств криптоанализа. Целью настоящей статьи является рассмотрение состояния вопроса в области направленного шифрования, обоснование возможности реализации направленного шифрования в группах точек на эллиптических кривых [1] и разработка концептуальных вопросов их реализации и оценки стойкости.

1. Анализ применяемых алгоритмов направленного шифрования

Вначале рассмотрим реализацию **RSA направленного шифрования**[2]. При использовании этого алгоритма получатель должен знать общесистемные параметры P_j и Q_j, причем P_j и Q_j должны быть "сильными" простыми числами. Этим повышается стойкость алгоритма. Используя эти числа, получатель должен выполнить следующие действия:

- сформировать открытый ключ E_k такой, что $1 \le E_k \le \varphi(N_j)$, где $N_j = P_j \cdot Q_j$, а $\varphi(N_j)$ - функция Эйлера, и $(E_k, \varphi(N_j)) = 1$;
- вычислить личный ключ D_k как обратный элемент к E_k в кольце, причём D_k вычисляется из соотношения $E_k \cdot D_k \equiv 1 (\mathrm{mod}\ \varphi(N))$;
- передать отправителю открытый ключ получателя E_k и модуль преобразования N_j, обеспечивая их целостность и подлинность.

Зашифрование выполняется по следующей схеме: всё сообщение делится на блоки, длина которых равна длине модуля преобразований $M = M_1 \| M_2 \| ... \| M_n$, затем выполняется зашифрование каждого блока сообщения по формуле $C_j = M_i^{E_k} (\mathrm{mod}\ N)$.

Расшифрование каждого блока выполняется с использованием секретного ключа как $M_i' = C_i^{D_k} (\mathrm{mod}\ N)$ $M_i' = C_i^{D_k} (\mathrm{mod}\ N)$, а затем все блоки объединяются в сообщение.

Считается [3], что стойкость метода базируется на сложности факторизации модуля преобразований N.

Метод обладает двумя существенными недостатками:

1) в связи с разработкой новых методов и средств криптоанализа сложность факторизации модуля становится субэкспоненциальной [3], например: $I = \exp\left(\delta (\ln N)^{\nu} (\ln \ln N)^{1-\nu}\right)$, где (δ, ν) - параметры используемого метода факторизации;
2) для повышения стойкости алгоритма необходимо увеличивать длину модуля преобразований, что приводит к повышению сложности прямого и обратного преобразований.

Разновидностью направленного шифрования RSA являются комбинированная схема направленного шифрования и схема RSA-OAEP [4].

Комбинированная схема направленного шифрования разработана с целью ускорения процедур зашифрования/расшифрования. Суть изменений заключается в том, что информация шифруется при помощи какого-либо симметричного алгоритма, а направленно шифруется лишь ключ симметричного шифрования и некоторая служебная информация.

Схема RSA-OAEP разработана как замена для стандартного направленного шифрования RSA. В схеме была введена специальная процедура предварительного шифрования данных на случайном ключе. Введенная процедура сводит на нет возможность атаки с выбранным *криптотекстом*.

Вторым алгоритмом направленного шифрования является **алгоритм, использующий схему Диффи-Хеллмана** [5]. Алгоритм позволяет вырабатывать сеансовые ключи динамически, то есть непосредственно перед началом передачи данных. Для согласования ключей используется схема Диффи-Хеллмана [5]. При использовании данной схемы каждый из абонентов должен обладать сертифицированными общесистемными параметрами домена: большим простым числом P_j (модуль вычислений) и первообразным корнем θ_V. Генерация общего секрета выполняется по формулам $Y_A = \theta_V^{X_A} \left(\mathrm{mod}\ P_j\right)$, $K_{BA} = Y_A^{X_B} = \theta_V^{X_A X_B} \left(\mathrm{mod}\ P_j\right)$, где Y_A - открытый ключ, K_{AB} - общий секрет. Абонент В выполняет те же действия.

После согласования ключей выполняется шифрование либо при помощи симметричного алгоритма шифрования, либо с помощью шифрующего устройства, выполняется разворачивание ключа до необходимой длины, а затем осуществляется направленное шифрование.

Стойкость метода базируется на сложности решения дискретного логарифма $X_A = \log_{\theta_\gamma} Y_B (\mathrm{mod}\ P)$ [4].

Но этот метод также обладает недостатком, так как в связи с разработкой новых методов криптоанализа сложность решения дискретного логарифмического уравнения уже носит субэкспоненциальный характер.

При использовании **алгоритма Эль-Гамаля** [6] отправитель и получатель должны знать открытые параметры p и g и открытый ключ получателя Y, где p-большое простое число, число g входит в диапазон $1 < g < p-1$ и имеет в мультипликативной группе Z_p^* большой порядок. В идеальном варианте g - первообразный элемент по модулю p. После этого получатель должен выполнить следующие действия:

- выбрать себе произвольное число k в диапазоне от 1 до $p-1$ (секретный ключ);
- вычислить открытый ключ как $h = g^k \bmod p$;
- передать отправителю открытый ключ получателя Y, обеспечивая его целостность и подлинность.

Зашифрование выполняется по схеме: все сообщение M делится на блоки M_i таким образом, что $M_i \in Z_p^*$, после чего каждый блок шифруют следующим образом:

Выбирают случайное число r такое, что $1 \le r \le p-1$;

Вычисляют пару $C = (c_1, c_2)$, где $c_1 = g^r \bmod p$, $c_2 = MY^r \bmod p$;

Пара $C = (c_1, c_2)$ передаётся получателю.

Расшифрование выполняется получателем по правилу $D(C) = c_2 \cdot \left(c_1^{\,k}\right)^{-1} \bmod p$. Получив все пары $C = (c_1, c_2)$ и расшифровав их, получатель собирает сообщение M.

Сложность системы базируется на сложности определения ключа k, произвольной составляющей r и информации M при знании открытых параметров p, g и ключа Y. Сложность криптоанализа определяется сложностью решения дискретного логарифма $r = \log_g Y \bmod p$, сложностью решения дискретного логарифма $r = \log_g c_1 \bmod p$ и определения значения сообщения M из соотношения $c_2 = MY^r \bmod p$.

Этот метод также обладает недостатком, так как в связи с разработкой новых методов криптоанализа сложность решения дискретного логарифмического уравнения уже носит субэкспоненциальный характер. Для увеличения криптостойкости системы необходимо увеличивать модуль преобразований. Кроме того, при зашифровании для каждого блока необходимо выполнять два возведения в степень при большом модуле преобразований.

Указанные недостатки в значительной мере могут быть устранены за счёт реализации направленного шифрования в группах точек эллиптических кривых.

2. Простая схема шифрования с использованием аппарата эллиптических кривых [1]

Необходимыми условиями для использования простой схемы направленного шифрования с использованием аппарата ЭК является набор параметров ЭК q, a, b, G, n и h, а также хэш-функция вместе с функцией генерирования ключей, где $q = p$ или $q = 2^m$ - порядок поля, a и b – коэффициенты ЭК, Q – базовая точка, n – порядок базовой точки на ЭК, h – кофактор.

2.1. Алгоритм зашифрования данных (рис.1). Входные данные: строка шифруемых данных M длиной l_M. Открытый ключ Q получателя зашифрованных данных, строка дополнительных данных $д\_д$, используемая совместно отправителем и получателем (необязательно). Открытый ключ Q должен соответствовать параметрам ЭК и быть подлинным. Для выполнения зашифрования необходимо использовать примитив Диффи-Хеллмана (ДХ), примитив генерирования ключей (ГК) и функции генерирования ключей на основе хеш-функции, например, SHA-1 или SHA-2 [7].

Зашифрование битовой строки M выполняется следующим образом:

1. Сгенерировать динамическую (сеансовую) пару ключей (d_e, Q_e), соответствующую параметрам ЭК, используя примитив ГК, где d_e - личный ключ, Q_e - открытый ключ.
2. Используя примитив ДХ, выработать из d_e и Q_e общий секрет $z \in F_q$.
3. Используя общий секрет Z и строку $д\_д$ (необязательно), сгенерировать ключ зашифрования K^3.
4. Используя ключ зашифрования, зашифровать открытые данные M: $C_i = M_i \oplus K_i^3$, где C – зашифрованные данные.

Выходные данные: битовая строка $QE||C$, где QE – битовая строка, сформированная из ключа Q_e

2.2. Алгоритм расшифрования (рис.1). Входные данные: битовая строка из ключа и зашифрованных данных $QE||C$, личный ключ d, принадлежащий получателю зашифрованных данных C, битовая строка $д\_д$. Личный ключ d должен быть сгенерирован с помощью примитива ГК. При выполнении преобразования расшифрования используется модуль подтверждения подлинности открытого ключа, модуль ДХ и функции генерирования ключей. Непосредственное формирование гаммы зашифрования осуществляется в блоке KDF по алгоритму, базирующемуся на преобразовании общего секрета, дополнительных данных и системных данных с помощью многократного применения однонаправленной безключевой хэш-функции SHA-1. Количество вызовов функции зависит от необходимой длины ключевой по-

следовательности. При использовании алгоритма на базе SHA-1 псевдослучайная последовательность представляет собой последовательность символов, формирующихся как

$$\Gamma_i = SHA(Z \| д\_д \| с\_д \| count), \quad (1)$$

где Γ_i - 160-битная последовательность развёрнутого ключа,

SHA-1 – функция хэширования,

Z – общий секрет абонентов,

д_д дополнительные данные,

с_д системные данные,

count– счётчик, который изменяется при каждом вызове хэш-функции SHA-1.

Расшифрование выполняется согласно схеме (рис.1)

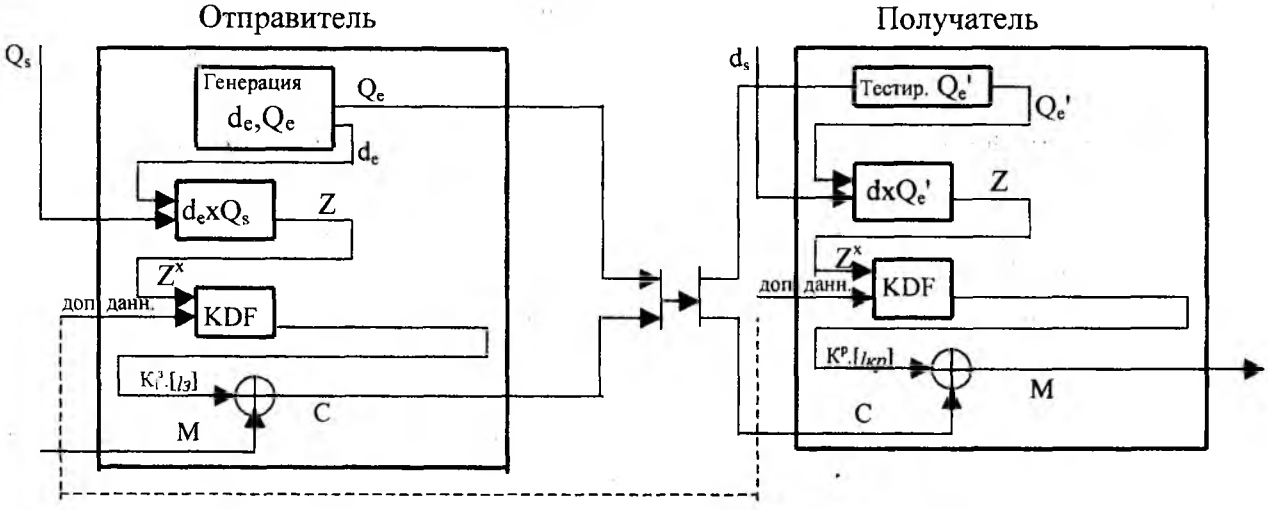

Рис. 1

Проведенный анализ показал, что более высокий уровень стойкости может быть обеспечен при использовании усиленной СНШ.

3. Усиленная схема направленного шифрования с использованием эллиптических кривых[1]

При использовании усиленной схемы СНШ в группах точек на ЭК корреспонденты должны обладать теми же знаниями, что и при использовании простой СНШ. Кроме того, объекты, использующие схему, должны согласовать схему вычисления кода аутентификации MAC.

3.1. Алгоритм зашифрования (рис.2). Входные данные такие же как и в простой схеме, только возможно использование двух строк дополнительных данных *дд1* и *дд2*.

Зашифрование битовой строки *M* выполняется следующим образом:

1. Генерируется сеансовая пара ключей (d_e, Q_e), с использованием параметров ЭК.
2. Используя примитив ДХ, выработать из d_e и Q_e общий секрет $z \in F_q$.
3. Из *Z* формируются ключевые данные КД=К³||*мак_ключ* длиной $l_{кз} + l_{мак}$, где *мак_ключ* –ключ зашифрования MAC.
4. Зашифровываются открытые данные как $C_i = M_i \oplus K_i^3$.
5. Вычисляется контрольная сумма тег *мак_тэг* для битовой строки: *мак_тэг* =*C* ||[$дд_2$] используя *мак_ключ* и MAC -схему.

Выходные данные: битовая строка *QE*||*C*||*мак_тэг*.

3.2. Алгоритм расшифрования (рис.2). Входные данные: битовая строка *QE*||C'||*мак_тэг'*, личный ключ *d*, принадлежащий получателю, битовые строки $дд_1$ и $дд_2$. При выполнении алгоритма используются те же модули, что и в простой схеме.

Расшифрование битовой строки *QE*||C'||*мак_тэг'* выполняется согласно рис.2

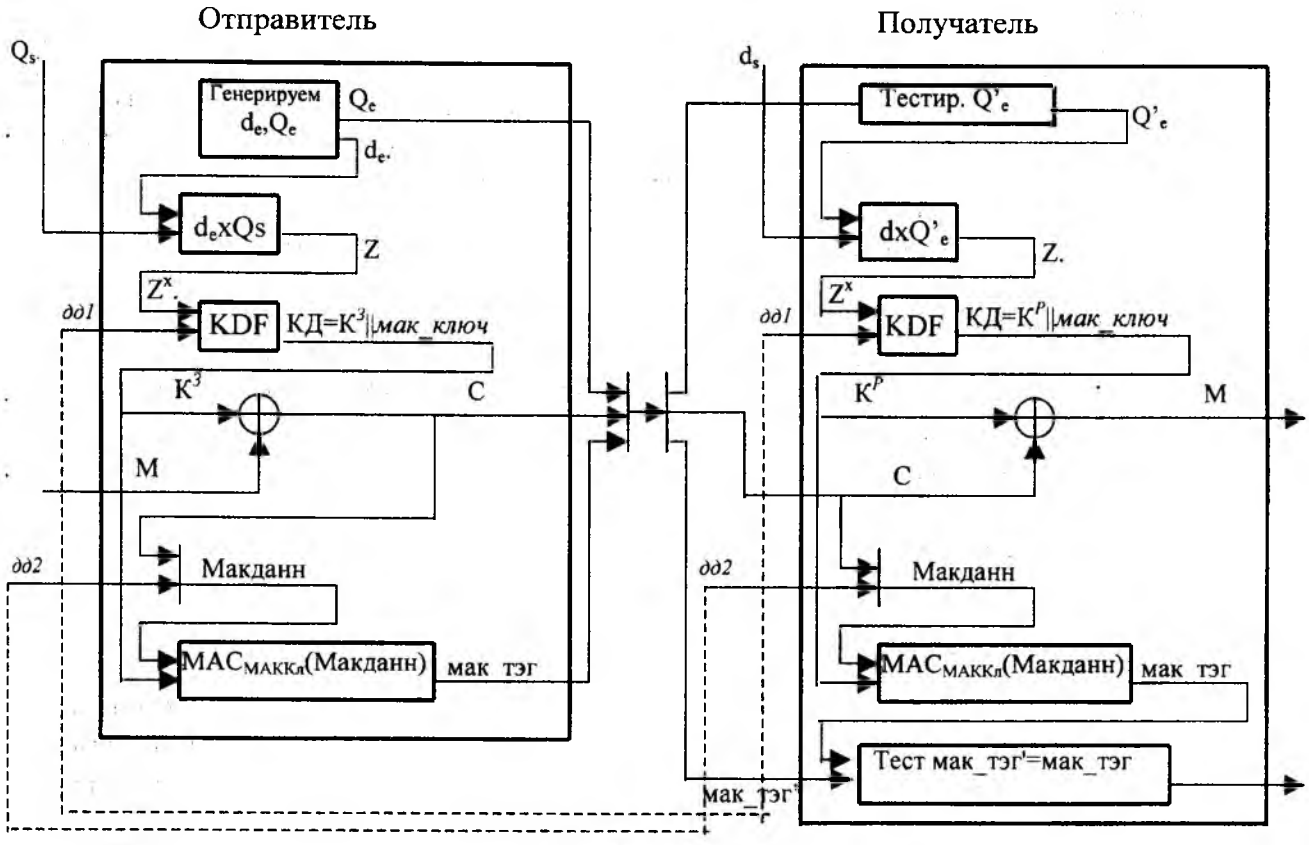

Рис. 2

4. Анализ схем, условия их применения

Рассмотрим основные угрозы, которые предотвращают эти схемы.

Простая схема направленного шифрования [1]. Основная угроза, которую предотвращает эта схема – нарушения конфиденциальности информации, то есть схема реализует услугу конфиденциальности. Услуга реализуется за счет поточного симметричного шифрования информации на развёрнутом сеансовом ключе.

Аутентификация получателя. Эта услуга заложена в саму суть схем направленного шифрования, так как расшифровать сообщение может только определённый абонент.

Дополнительная услуга, предоставляемая простой схемой – аутентификация отправителя. Данная услуга реализуется в том случае, если дополнительные данные известны только двум абонентам. В этом случае, верно расшифровав информацию, получатель может быть уверен, что она послана определённым абонентом, иначе, если получатель неверно расшифровал информацию, то либо при передаче информации была допущена ошибка, либо соединение было атаковано злоумышленником, либо отправитель сообщения не тот, за кого себя выдаёт.

Основные атаки – повторение сообщений, модификация сообщений, удаление пакетов. Разработчики схемы не предусматривали защиту от этих атак, поэтому защита от них полностью лежит на протоколах более низкого уровня.

Усиленная схема направленного шифрования [1]. Основные угрозы, предотвращаемые этой схемой – нарушение конфиденциальности информации и нарушение целостности зашифрованной информации и ключа. Кроме того, схема гарантирует аутентификацию получателя. Услуга конфиденциальности реализуется при помощи поточного симметричного шифрования информации на развёрнутом сеансовом ключе. Услуга целостности обеспечива-

ется при помощи ключевой MAC – схемы. Аналогично простой схеме может быть введена аутентификация отправителя.

Как показывает анализ, простая схема может применяться, в том случае, если канал связи целостный, т.е. надёжно защищён на системном либо канальном уровнях. В случае, если целостность канала связи не гарантируется, необходимо использовать усиленную СНШ, т.к. она защищает данные от модификации или подмены. Но, учитывая небольшую разницу в необходимых вычислительных ресурсах на выполнение каждой из схем, в приложениях рекомендуется использовать усиленную СНШ.

По сравнению со схемами RSA и Диффи-Хеллмана схема на эллиптических кривых обладает рядом преимуществ.

Математический аппарат групп точек на эллиптических кривых над полем Галуа обеспечивает значительно более высокую стойкость. Стойкость в зависимости от метода криптоанализа определяется через сложность, измеряемую в необходимом числе операций сложения на эллиптической кривой [8]:

- Метод λ-Полларда. Сложность криптоанализа $I_\lambda = 2\sqrt{n}$.
- Метод ρ-Полларда. Сложность криптоанализа $I_\rho = \sqrt{\frac{\pi n}{2}}$.
- Метод ρ-Полларда оптимальный. Сложность криптоанализа $I_\rho = \sqrt{\frac{\pi n}{4}}$.

В связи с тем, что при использовании групп точек на эллиптической кривой стойкость к криптоанализу достаточно высока, появляется возможность использовать модули преобразования меньших размеров, чем при преобразованиях в полях и кольцах. Так сложность факторизации 1024-битного модуля RSA приблизительно равна сложности криптоанализа ЭК с модулем преобразований 160 бит. Для эффективной защиты [8] сейчас вполне достаточно размера модуля от 2^{256} и более. Данные по стойкости приведены в табл. 1. Указанное позволяет ускорить процесс вычислений (криптографических преобразований). Кроме того, для вычислений может использоваться проективное представление (базис), применение которого позволяет ускорить вычисления.

Таблица 1

Длина модуля преобразований	Сложность факторизации модуля RSA	Сложность криптоанализа ЭК
192 бита	$2^{40,41} \approx 10^{12,32}$	$2^{95,82} \approx 10^{29,21}$
256 битов	$2^{40,56} \approx 10^{14,5}$	$2^{127,82} \approx 10^{39}$
512 битов	$2^{65,15} \approx 10^{19,86}$	$2^{255,82} \approx 10^{78}$
1024 бита	$2^{88,47} \approx 10^{27}$	$2^{511,82} \approx 10^{156}$

Кроме рассмотрения стойкости определения общего секрета необходимо произвести и анализ непосредственно шифрующей последовательности, вырабатываемой функцией KDF. В стандарте X9.63 предлагается алгоритм KDF, базирующийся на преобразовании общего секрета, дополнительных данных и системных данных с помощью многократного применения однонаправленной безключевой хэш-функции SHA-1, для хэширования можно использовать и появившийся алгоритм SHA-2 [7]. Количество вызовов функции зависит от необходимой длины ключевой последовательности. При использовании алгоритма на базе SHA-1 псевдослучайная последовательность представляет собой последовательность символов, формирующихся по закону (1).

Однако, мы не смогли найти результатов исследования таких последовательностей и доказательства их свойств (статистические характеристики, статистическая безопасность,

структурные свойства, период повторения). Поэтому предложенный алгоритм формирования псевдослучайной последовательности требует дополнительного исследования, так как если функция развёртывания ключа обладает небольшим периодом повторения либо плохими статистическими свойствами, то использование этой функции может значительно снизить стойкость всей схемы.

Исследование стандартного алгоритма развёртывания ключа требуется ещё и потому, что кроме стандартного алгоритма на основании хэш-функции могут быть использованы алгоритмы, построенные на основе алгоритмов блочного симметричного шифрования в режимах выработки псевдослучайной последовательности Γ_i, алгоритмы, построенные на основе линейных рекуррентных регистров, а также алгоритмы, базирующиеся на комбинации симметричных алгоритмов и ЛРР, хэш-функции и ЛРР. Для этого можно использовать:

- примитив, основанный на алгоритме ГОСТ 28147-89 в режиме выработки гаммы шифрующей (Γ_i);
- примитив, основанный на алгоритме RIJNDAEL в режиме выработки Γ_i;
- примитив, основанный на линейном рекуррентном регистре;
- примитив, основанный на комбинации ЛРР и ГОСТ 28147-89 в режиме выработки Γ_i;
- примитив, основанный на комбинации ЛРР и RIJNDAEL в режиме выработки Γ_i;
- примитив, основанный на комбинации ЛРР и стандартного примитива на SHA-1.

Перечисленные выше примитивы были протестированы на соответствие всем необходимым требованиям.

Полученные результаты показывают, что применение алгоритма НШ в группах точек на ЭК, когда Γ_i формируется с использованием соотношения $\Gamma_i = SHA(Z \| \partial\_\partial \| c\_\partial \| count)$ обладает следующими особенностями:

1) Реализация Γ_i зависит как от общего секрета, так и дополнительных данных, которые могут вводить взаимодействующие объекты по своему усмотрению.

2) Обеспечивается достаточно высокая скорость формирования Γ_i.

3) Период повторения зависит от составляющих, участвующих в формировании входных параметров функции генерации ключа и является случайным.

4) Если в качестве дополнительных данных использовать последовательность со строгим периодом повторения и нумерацию блоков осуществлять с использованием возрастающей функции, то можно надеяться, что период Γ_i будет не меньше периода повторения дополнительной последовательности. Для того, чтобы период повторения дополнительной последовательности был гарантированно высоким, необходимо использовать выборки из ЛРР гарантировано большого периода повторения.

5) Такая последовательность обладает хорошей статистической безопасностью (независимостью), что объясняется с одной стороны использованием конкатенации общего секрета, дополнительных и системных данных, а с другой стороны использованием однонаправленной хэш-функции.

Рассмотрим сложность преобразования в группах точек на ЭК. Сравнение операции скалярного умножения на ЭК с операцией модульного возведения в степень показывает, что производительность операции скалярного умножения на ЭК с модулем 160 битов приблизительно на треть выше, чем производительность модульного возведения в степень с соответствующей по сложности криптоанализа длиной модуля. И с ростом модуля преобразования выигрыш в производительности существенно возрастает (для 256 битов уже более чем в 3 раза).

Значения времени выполнения полных процедур зашифрования и расшифрования (в качестве функции развёртывания ключа используется SHA-1) на компьютере с процессором К6-233 приведены в табл. 2.

Таблица 2

M		191	191	409	409
L_k		190	190	408	408
$L_д$		100байт	100кбайт	100байт	100кбайт
T, с	Обычная схема	0,40	0,43	2,96	3
	Усиленная схема	0,44	0,47	3	3,03

При рассмотрении данных таблицы необходимо учесть, что большинство времени вы полнения процедур уходит на генерацию общего секрета и ключа. Конечно, время, затрачи ваемое на выполнение зашифрования, при модуле преобразования 409 бит достаточно высо ко, но при этом необходимо учитывать, что увеличение размера шифруемого текста не при ведет к значительному увеличению суммарного времени шифрования (время, затрачиваемо на зашифрование 100 байт на 0,03 секунды меньше, чем время, затрачиваемое на зашифро вание 100кбайт). Кроме того, модуль преобразования 409 битов является слишком большим При нынешнем развитии средств вычислительной техники и математическом аппарате крип тоанализа при шифровании не требуется использования модулей такой длины. Заметим, что операции вычисления ключевой пары d_e и Q_e, а также вычисления общего секрета Z мож но производить либо предварительно и затем хранить их соответствующим образом, либо возлагать на специальный сопроцессор. Более того, процедура генерации d_e и Q_e може быть вынесена за пределы самого алгоритма шифрования, а также может выполняться пред варительно. В этом случае скорость шифрования может быть увеличена на несколько поряд ков. Кроме того, операции вычисления ключа и зашифрования могут быть распараллелены Так же многое зависит от производительности алгоритма выработки ключевой последова тельности. В качестве алгоритма может использоваться любой алгоритм симметричного шифрования, работающий в режиме обратной связи по шифротексту, и даже ЛРР с общим секретом в качестве начальной установки.

5. Возможности модификации схем с целью улучшения их защищённости

Как мы выяснили, наиболее предпочтительными для направленного шифрования с точ ки зрения защищённости являются схемы на эллиптических кривых. Но и они не гарантиру ют защиты от всех возможных угроз. Как сказано выше, рекомендуется использовать уси ленную схему направленного шифрования на ЭК, но она не обеспечивает такой важной ус луги, как аутентификация отправителя.

Для реализации услуги аутентификации отправителя может быть использован механизм цифровой подписи. Так как все схемы НШ базируются на несимметричной криптографии, то внедрение цифровой подписи не представляет никакой сложности. Правда, применение циф ровой подписи может повлечь за собой значительное увеличение объёмов ключевой инфор мации, но этого можно избежать. Направленное шифрование базируется на наличии у субъ екта – получателя секретного и открытого ключей, это же условие является необходимым для реализации механизма цифровой подписи. В сети в большинстве случаев существуе возможность связаться каждому субъекту с каждым, это значит, что каждый субъект облада ет парами открытый – личный ключи, следовательно, не возникает необходимость введени новых ключевых данных, так как ключи из одной пары могут использоваться и для направ ленного шифрования, и для цифровой подписи. Что допускается согласно стандарту ISO 11166. Информацией, подписываемой цифровой подписью, может являться MAC – код схемах использующих общий секрет, либо зашифрованный блок в схемах с поблочной пере дачей данных.

Перечислим достоинства и недостатки состоятельных протоколов НШ по сравнению с стандартными протоколами НШ. Основным недостатком введения в схемы НШ механизмов выполняющих вычисление MAC – кода и цифровой подписи, является увеличение времен

вычислений. Например, в схеме НШ на основе RSA для обеспечения целостности и аутентификации информации, вычисления MAC – кода и цифровой подписи необходимо выполнять для каждого блока. Это увеличивает временные затраты на обработку одного блока на время вычисления MAC – кода и время вычисления цифровой подписи, то есть более чем в два раза. Для схем Д-Х и Д-Х на ЭК увеличение времени на обработку данных не столь значительно по сравнению с классическими алгоритмами НШ, так как в этих схемах вычисление MAC – кода и цифровой подписи выполняется один раз. Вторым недостатком является необходимость увеличения количества ключей в случае, если система требует использования разных ключей для шифрования и выработки цифровой подписи.

К достоинствам состоятельных протоколов НШ можно отнести защищённость от атак всех типов (в том числе и от атаки man-in-the-middle, если открытые ключи абонентов распространяются в виде сертификатов).

Анализируя основные свойства состоятельных протоколов НШ, можно сделать вывод, что их использование является более предпочтительным, чем использование обычных схем НШ.

Заключение

Следует ожидать, что в ближайшее время в качестве направленного шифра будут использоваться алгоритмы (схемы), реализованные в группах точек на эллиптической кривой. Их использование позволяет с одной стороны обеспечить необходимый уровень стойкости, с другой уменьшить сложность, т. е. повысить скорость преобразований. Эти схемы могут применяться в банковских системах, а также в Интернет.

Список литературы: 1. X9.63-199x Public Key Cryptography For The Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography. 2. ISO 11166 – 1995 Banking – Key management by means of asymmetric algorithms. Part 2: Approved algorithms using the RSA cryptosystem. 3. *И.Д. Горбенко, П.В. Колесников* Оценка стойкости RSA систем, в которых открытые ключи или параметры являются личными// Радиотехника: Всеукр. межвед. научн.-техн. сб. 2001г. вып. 119. 4. *Jakob Jonsson, Burt Kaliski:* RSA-OAEP Encryption scheme. B: Primitive specification and supportingdocumentation, 2000, by. 5. X9.42 - 1998, Public Key Cryptography for The Financial Service Industry : Agreement of Symmetric Keys on Using Diffie-Hellman and MQV Algorithms. 6. *Вербіцький О.В.* Вступ до криптології. Львів: ВНТЛб. 1998р.7.E. Biham and A. Shamir. Differential cryptoanalysis of FEAL and N-Hash. In Advances in Cryptology - Eurocrypt '91, pages 1-16, 1991.8. *И.Д. Горбенко, С.И. Збитнев, А.А. Поляков* Криптоанализ криптографических преобразований в группах точек эллиптических кривых методом Полларда// Радиотехника: Всеукр. межвед. научн.-техн. сб. №119. 2001г.

Харьковский национальный университет радиоэлектроники

Поступила в редколлегию 25.04.2002