УДК 681.3.06:519.248.681

И.Д. ГОРБЕНКО, д-р техн. наук, С.И. ЗБИТНЕВ, А.А. ПОЛЯКОВ

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЦП В ГРУППАХ ТОЧЕК ЭЛЛИПТИЧЕСКИХ КРИВЫХ

Введение

В современных автоматизированных системах управления, компьютерных системах и сетях, различных информационных технологиях и системах информационных технологий предъявляются жесткие требования к обеспечению целостности, наблюдаемости и достоверности информации на всех этапах их жизненного цикла. При этом под информацией мы будем понимать совокупность всех данных и программ, которые используются в системе или технологии, независимо от их логического или физического представления. Под информацией будем понимать также и сообщения, циркулирующие в соответствующих системах или технологиях. Опыт применения и проведенные исследования показали, что эти жесткие требования, особенно по реализации функции причастности, могут быть обеспечены только за счет применения цифровой подписи. Цифровая подпись (ЦП), по сути, представляет собой добавленные к информации данные, вычисленные посредством криптографического преобразования защищаемой информации и параметров, наличие которых позволяет удостовериться в целостности информации и подлинности ее источника, а также обеспечить защиту от подлога со стороны получателя.

По существу цифровая подпись представляет собой цифровой эквивалент подписи (штампа, печати, водяного знака и т.д.), наличие которой в сообщении, данных или программе позволяет с высокой вероятность определить источник (источники) этого сообщения или данных и юридически доказать, что с указанной допустимой вероятностью $P_д$ только он мог сформировать эту подпись, но подделать ее в течении заданного времени, при ограниченных ресурсах, злоумышленник может с вероятностью, не превышающей заданной величины $P_з$. Причем ЦП в таком применении вычисляется на основе защищаемой информации с использованием личного (конфиденциального) ключа конкретного субъекта или объекта, являющегося ее источником или отправителем. Проверка целостности и подлинности производится с использованием открытого ключа, причем знание открытого ключа не позволяет подделать ЦП с вероятностью, превышающей $P_з$.

Проведенный анализ показал [1-3], что криптографические преобразования типа ЦП должны удовлетворять ряду требований, основными из них являются следующие:

1) алгоритмы выработки и проверки ЦП должны быть открытыми, т.е. несекретными;
2) алгоритмы выработки и проверки ЦП должны обладать не выше чем полиномиальной сложностью;
3) алгоритм нахождения конфиденциального ключа и/или подделки подписи должен обладать не ниже, чем экспоненциальной, т.е. практически не реализуемой сложностью;
4) ЦП должна обладать чувствительностью к любым изменениям подписанных данных, т.е. обнаруживать нарушения целостности;
5) вероятность появления двух одинаковых подписей в разных сообщениях не должна превышать допустимого значения;
6) вычислительная сложность выработки и проверки ЦП должна быть минимизирована и иметь близкие по величине значения;

7) обеспечивать защиту от подменны, подделки и имитации полной ЦП с требуемой вероятностью;
8) цифровые подписи, полученные для одной и той же информации в разное время и на различных устройствах должны отличаться с большой вероятностью;
9) ключи выработки должны быть конфиденциальными, а ключи проверки ЦП – открытыми;
10) ЦП должна обладать максимальной стойкостью к обнаружению любых изменений, подделок и нарушений;
11) должна существовать возможность принять ЦП с различными уровнями стойкости и сложностью выработки и проверки ЦП;
12) возможность программной, аппаратно-программной и аппаратной реализации ЦП с примерно одинаковой сложностью;
13) возможность использования ЦП как с одинаковыми общесистемными параметрами в сети, так и индивидуальными для отдельной части объектов (субъектов);
14) возможность многоуровневой выработки и проверки ЦП одной и той же информации с использованием различных ключей и при необходимости различных общесистемных параметров.
15) используемая ЦП должна позволять проведение следственных экспериментов с целью обеспечения судебного разбирательства и арбитража.
16) должна существовать возможность хранения ЦП как вместе с защищаемой информацией, так и отдельно от неё.

В конце 20-го века протоколы цифровой подписи получили широкое распространение в силу увеличение компьютеризации бумагооборота. Летом 2000 года президентом США Биллом Клинтоном был подписан и вступил в действие с 1 октября указ «Electronic Signatures in Global and National Commerce Act», приравнивающий в коммерческих документах электронную подпись к чернильной (более того и сам этот указ стал первым документом, подписанным электронной подписью)[4]. Европейский Союз издал распоряжение, согласно которому цифровая подпись в скором времени будет иметь силу во всех странах Союза. Над инициативами в данной области, взаимодействуя друг с другом, работают многие азиатские государства, причём в некоторых из них электронная подпись уже закреплена законодательно. В Российской федерации опубликован и доступен по Internet Проект федерального закона «Об электронной цифровой подписи».Правительство Сингапура объявило, что с 2008 г. электронные деньги в этой стране станут легальной валютой, имеющей хождение наравне с наличными, при этом осуществлять расчёты с любыми торговыми организациями можно будет с помощью карманного компьютера или сотового телефона.

Параллельно с развитием криптографических систем, еще боле интенсивно развиваются математические методы и криптоаналитические системы, что влечет за собой повышение требований к стойкости криптосистем, в частном случае к электронной цифровой подписи.

1. Классификация известных ЦП

Абсолютное большинство разрабатываемых и используемых в мире ЦП базируется на использовании несимметричных криптографических преобразований, выполняемых в кольцах [1], полях Галуа [2] и группах точек эллиптических кривых [3]. К ЦП, реализованных в кольцах, необходимо отнести RSA подобные алгоритмы [6], к преобразованиям в полях Галуа – алгоритмы Диффи-Хеллмана [7] и Ель-Гамаля [8]. Опыт применения и проведения исследований ЦП, базирующихся на преобразованиях в кольцах и полях, показали, что они практически исчерпали себя и в ближайшее время не будут обеспечивать требуемой стойкости. Одним из способов решения поставленной задачи является увеличение длины ключа ныне действующих цифровых подписей: *RSA* и *DSA*. Однако увеличение длины ключа повышает требование этих криптосистем к вычислительным возможностям ЭВМ, что не всегда является приемлемым (не все организации в состоянии поменять весь парк компьютеров для

осуществления приемлемого уровня быстродействия обновленных алгоритмов электронной подписи). Для разрешения этого противоречия разработаны и начали внедряться новые или модифицированные криптографические преобразования, выполняемые в группах точек эллиптических кривых. Появилось значительное число методов, на их основе разработаны стандарты и проекты стандартов. Поэтому представляет интерес задача их изучения, выявления особенностей и возможностей, анализа стойкости и сложность выполнения ЦП. Основой при сравнительном анализе конечно же должны быть требования 1-16, приведенные во введении. Вместе с тем, при выполнении сравнительного анализа необходимо задаться видами атак и типами угроз ЦП соответствующей информации [6-9].

На наш взгляд основными видами атак на ЦП являются следующие криптоаналитические атаки на ЦП [9]:

1) *Атака на основе известного открытого ключа* (key-only attack). Самая слабая из атак, практически всегда доступная криптоаналитику (злоумышленнику). Она может выполняться при априорной определенности криптоаналитика относительно реализации ЦП, знании общесистемных параметров, а также действующих открытых ключах.
2) *Атака на основе известных подписанных сообщений* (known-message attack). Для этой атаки полагается, что в распоряжении криптоаналитика имеется некоторое число пар $(m,\langle r,s\rangle)$ подписанных сообщений m, при этом он не может выбирать сообщение m. Кроме этого криптоаналитик знает систему и параметры ЦП.
3) *Простая атака с выбором подписанных сообщений* (generic chosen-message attack). В этом случае криптоаналитик имеет возможность выбрать некоторое количество подписанных сообщений, знает общесистемные параметры и имеет доступ до открытых ключей после выбора подписанных сообщений.
4) *Направленная атака с выбором сообщения* (direct chosen-message attack). Криптоаналитик знает общесистемные параметры, может по своему усмотрению выбирать открытый ключ и после этого выбирать подписанные сообщения.
5) *Адаптивная атака с выбором подписанного сообщения* (adaptive chosen-message attack). При осуществлении атаки криптоаналитик может выбирать открытый ключ, а также подписанное сообщение. При этом выбор следующего подписанного сообщения он может делать на основе знания допустимой подписи предыдущего выбранного сообщения.

Проведенный анализ показал, что каждая атака направлена на достижение определенной цели. С учетом этого можно выделить следующие виды угроз, в порядке возрастания опасности, для всех схем электронной цифровых подписей [9]:

1) *Экзистенциональная подделка* (existential forgery). Угроза заключается в создании криптоаналитиком для какого-нибудь, возможно бессмысленного сообщения m', отличающегося от перехваченного, реальной (правильной) ЦП.
2) *Селективная подделка* (selective forgery). Представляет угрозу создания для заранее выбранного сообщения m правильной ЦП.
3) *Универсальная подделка* (universal forgery). Эта угроза заключается в нахождении криптоаналитиком алгоритма формирования подписи, функционально эквивалентного действительному алгоритму ЦП, что позволяет создать или модифицировать истинные подписанные сообщения.
4) *Полное раскрытие* (total break). При этой угрозе криптоаналитик может вычислить секретный ключ, возможно отличный от d, но соответствующий открытому ключу Q. Это позволяет криптоаналитику формировать цифровые подписи для любых сообщений и в дальнейшем навязывать эти сообщения корреспондентам.

2. Сравнительный анализ основных ЦП

Проведенный анализ показывает, что наиболее надежными являются схемы ЦП, стойкие против самой слабой из угроз на основе самой сильной из атак, т.е. против экзистенциональ-

ной подделки на основе атаки с выбором подписанных сообщений. Показано [10], что схемы цифровой подписи стойкие против такой атаки существуют только тогда, когда существует коллизионностойкая односторонняя функция.

Кроме доказательств теоретической стойкости цифровой подписи, одним из основных факторов является проверка реально стойкой ЦП временем. В проекте стандарта IEEE, X9-62 предложен вариант цифровой подписи, являющейся модификацией применяющегося стандарта X9.30 (DSA) и получивший название *ECDSA*. В нем в качестве математического аппарата выбрана группа точек эллиптической кривой над простым полем. Использование математического аппарата эллиптических групп позволяет уменьшить длину ключа, что в свою очередь позволяет повысить скорость алгоритма формирования и проверки подписи. Самое же главное это надежда на то, что по мере развития математических методов и производительности криптоаналитических систем криптографические преобразования в группах точек эллиптических кривых будут более устойчивыми к криптоанализу, чем преобразования в кольцах [1] и полях [2]. Кроме того, при разработке преобразований в группах точек эллиптических кривых появилась возможность учесть требование реализации цифровой подписи с различными длинами. Естественно, что попытки удовлетворить противоречивым требованиям к цифровым подписям привели к потоку разработки различных модификаций криптопреобразований, в том числе цифровых подписей.

Другой важнейшей особенностью при анализе и сравнении подписей является применение в ЦП функции хэширования. Дело в том, что для практической реализации состоятельного протокола аутентификации необходимо получить интерактивный случайный запрос от проверяющего [11]. Шамир предложил способ преобразования протокола аутентификации в схему цифровой подписи посредством замены случайного запроса проверяющего на хэш-функцию подписывающего сообщения. В этом случае вместо обращения к проверяющему (он же получатель сообщения), доказывающий (он же подписывающий) вычисляет хэш-функцию от сообщения M – $H(M)$ и использует его в качестве запроса. В результате строится практически однопроходный протокол и обеспечивается групповая передача подписанных сообщений от одного источника многим получателям. В определенном смысле подпись Шнора является классической. Сущность ее в следующем.

1. Подписывающий формирует случайное число $k \in (1,2,...,q-1)$ и вычисляет

$$r = a^k \pmod p,$$

где a – первообразный элемент поля; p – простое число. Параметры используются в качестве первой компоненты цифровой подписи.

Подписывающий вычисляет случайный запрос в виде значения хэш-функции от параметров и подписываемого сообщения M

$$e = H(r, M).$$

2. Подписывающий вычисляет вторую компоненту ЦП

$$s = (x \cdot e + k) \bmod q,$$

где x – долговременный секретный ключ, и формирует сообщение M с подписью (e, s).

3. Получатель обрабатывает подписанное сообщение вида $(M';(e',s'))$. Он вычисляет

$$r' = a^{s'} y^{e'} \pmod p$$

и проверяет выполняется ли равенство

$$e' = H(r', M).$$

Если равенство выполняется, то подпись принимается, а информация M считается целостной и подлинной.

Стойкость схемы Шнора в значительной степени зависит от свойств функции H. Если криптоаналитик может осуществлять коллизии специального вида, т.е. по заданной паре (r, M) находить другую информацию (сообщение) M', $M' \neq M$, такую что

$$H(r, M) = H(r', M'),$$

то он может по крайней мере осуществлять экзистенциальную подделку подписи. Для этого достаточно получить M и подпись (e, s) для него, а также найти коллизию указанного вида. Тогда пара (e, s) будет подписью также и для сообщения М. Фактически подпись Шнора и другие подписи такого класса будут стойкими, если хэш-функция ведет себя как случайная величина (для различных сообщений).

Алгоритмы ЦП Шнора и ECSS [11,12] в группах точек эллиптической кривой приведен ниже.

<table>
<tr><td colspan="2">Алгоритм Шнора на эллиптической кривой</td></tr>
<tr><td>Формирование цифровой подписи</td><td>Проверка цифровой подписи</td></tr>
<tr><td>Вход: секретный ключ d, открытый ключ $Q = -d \times G$, общесистемные параметры.
Выход: ЦП $\langle r, s \rangle$ для сообщения M.</td><td>Вход: открытый ключ Q, общесистемные параметры, ЦП $\langle r', s' \rangle$, для сообщения M'.
Выход: Подпись действительна или нет.</td></tr>
<tr><td>1. Выбираем $k \in \{1, \ldots, n-1\}$;
2. $k \times G = (x, y)$;
3. $r = h(x, e) \bmod n$;
4. $s = k^{-1}(dr + e) \bmod n$.</td><td>1. $(x, y) = s' \times G + r' \times Q$;
2. $v = \pi(x, y) \bmod n$;
3. $r' \stackrel{?}{=} v$.</td></tr>
<tr><td colspan="2">ECSS</td></tr>
<tr><td>Вход: секретный ключ d, открытый ключ $Q = dG$, общесистемные параметры.
Выход: ЦП $\langle r, s \rangle$ для сообщения M.</td><td>Вход: открытый ключ Q, общесистемные параметры, ЦП $\langle r', s' \rangle$, для сообщения M'.
Выход: Подпись действительна или нет.</td></tr>
<tr><td>1. $e = h(M)$;
2. Выбираем $k \in \{1, \ldots, n-1\}$;
3. $k \times G = (x, y)$;
4. $r = (x + e) \bmod n$;
5. $s = (k - dr) \bmod n$.</td><td>1. $e' = h(M')$;
2. $(x, y) = r' \times G + s' \times Q$;
3. $v = \pi(x, y) \bmod n$;
4. $r' \stackrel{?}{=} v$.</td></tr>
</table>

Здесь общесистемные параметры – параметры эллиптической кривой a и b, порядок эллиптической кривой $u = \#E(F_q)$, кофактор h, базовая точка G и порядок базовой точки $n = u/h$.

В проекте стандарта ISO/IEC CD 15946-2 включено четыре алгоритма ЦП в группах точек эллиптической кривой. Ниже приведены ЦП вошедшие в этот проект (кроме *ECSS*, так как он описан выше). Рассмотрим их более подробно.

<table>
<tr><td colspan="2">Алгоритм ECDSA</td></tr>
<tr><td>Формирование цифровой подписи</td><td>Проверка цифровой подписи</td></tr>
<tr><td>Вход: секретный ключ d, $Q = d \times G$, общесистемные параметры.
Выход: ЦП $\langle r, s \rangle$ для сообщения M.</td><td>Вход: открытый ключ Q, общесистемные параметры, ЦП $\langle r', s' \rangle$, для сообщения M'.
Выход: Подпись действительна или нет.</td></tr>
<tr><td>1. $e = h(M)$;
2. Выбираем $k \in \{1,...,n-1\}$;
3. $k \times G = (x, y)$;
4. $r = \pi(x, y) \bmod n$;
5. $s = k^{-1}(dr + e) \bmod n$.</td><td>1. $e' = h(M')$;
2. $w = (s')^{-1} \bmod n$;
3. $u_1 = e'w \bmod n$ и $u_2 = r'w \bmod n$;
4. $(x, y) = u_1 \times G + u_2 \times Q$;
5. $v = \pi(x, y) \bmod n$;
6. $r' \stackrel{?}{=} v$.</td></tr>
<tr><td colspan="2">Алгоритм $EC-GDSA$</td></tr>
<tr><td>Вход: секретный ключ d, открытый ключ $Q = d^{-1} \times G$, общесистемные параметры.
Выход: ЦП $\langle r, s \rangle$ для сообщения M.</td><td>Вход: открытый ключ Q, общесистемные параметры, ЦП $\langle r', s' \rangle$, для сообщения M'.
Выход: Подпись действительна или нет.</td></tr>
<tr><td>1. $e = h(M)$;
2. Выбираем $k \in \{1,...,n-1\}$;
3. $k \times G = (x, y)$;
4. $r = \pi(x, y) \bmod n$;
5. $s = (kr - e)d \bmod n$.</td><td>1. $e' = h(M')$;
2. $w = (r')^{-1} \bmod n$;
3. $u_1 = e'w \bmod n$ и $u_2 = s'w \bmod n$;
4. $(x, y) = u_1 \times G + u_2 \times Q$;
5. $v = \pi(x, y) \bmod n$;
6. $r' \stackrel{?}{=} v$.</td></tr>
</table>

Алгоритм $EC-KCDSA$	
Вход: **секретный ключ** d, **открытый ключ** $Q=d^{-1}\times G$, **общесистемные параметры.** Выход: **ЦП** $\langle r,s\rangle$ **для сообщения** M.	Вход: **открытый ключ** Q, **общесистемные параметры, ЦП** $\langle r',s'\rangle$, **для сообщения** M'. Выход: **Подпись действительна или нет.**
1. **Выбираем** $k\in\{1,...,n-1\}$; 2. $k\times G=(x,y)$ **и** $c=x\|\|y$; 3. $r=H(c)\bmod n$; 4. $h=H(Z_A\|\|M)$; 5. $e=r\oplus h\bmod n$; 6. $s=(k-e)d\bmod n$.	1. $h'=H(Z_A\|\|M')$; 2. $e'=r'\oplus h'\bmod n$; 3. $(x,y)=e'\times G+s'\times Q$; 4. $v=H(x\|\|y)\bmod n$; 5. $r'\overset{?}{=}v$.

Здесь $\|\|$ – конкатенация двух строк; $\oplus$ – сложение по модулю 2; π – функция выделение $x \bmod n$. В случае $EC-KCDSA$ в общесистемные параметры входит дополнительная информация Z_A.

В рамках конкурса NESSIE [13] участвует семь алгоритмов цифровой подписи и одна из них $ECDSA$. По предварительному результату отбора $ECDSA$ вышла в следующий этап.

Рассмотрим стойкость всех цифровых подписей к ранее перечисленным угрозам.

Экзистенциональная подделка. Этот вид угрозы возникает при использовании хэш-функции. В связи с тем, что хэш-функция производит отображение $m\in M$ на $h\in H$, где множество $H\subset M$, возможны коллизии, при которых для $h=H(m)$, $h'=H(m')$ и $h=h'$, $m\neq m'$. Для защиты от экзистенциальной подделки на хэш-функцию накладывается требование отсутствия полиномиального алгоритма создания коллизий.

Обычно при доказательстве стойкости цифровой подписи предполагается, что хэш-функция является случайным черным ящиком (оракулом), на вход которого поступают случайные запросы $m_0,m_1,m_2...$, а на выходе формируется случайные ответы $h_0,h_1,h_2....$ Все запросы и ответы оракул запоминает, и если на вход поступает $m_i=m_j$ и $i\neq j$, то он выдает ранее вычисленный ответ.

На практике хэш-функция должна удовлетворять, по крайней мере, следующим требованиям [14]:

a) не выше чем полиномиальная сложность вычисления, что позволяет эффективно вычислить значения h;
b) однонаправленность, при которой обеспечивается односторонность преобразований. Сущность этого свойства заключается в невозможности вычисления сообщения m по известному h (например, имеет не ниже чем экспоненциальную сложность);
c) защищенность от коллизий, при которой практически невозможно найти m_1 и m_2 такие, что $H(m_1)=H(m_2)$, так как нахождение m_1 и m_2 носит не ниже чем экспоненциальную сложность.

Рассмотрим необходимость выполнения этих условий на примере $ECDSA$. Если хэш-функция обратима, то криптоаналитик B может осуществить эффективную атаку на ЦП

следующим образом. Он выбирает случайное число l и вычисляет параметр цифровой подписи $r=\pi(Q+lG)$. Далее B принимает, что $s=r$, и вычисляет $e=r\cdot l(\bmod n)$. Если B может найти сообщение m такое что $e=H(m)$, тогда $\langle r,s\rangle$ действительная ЦП.

Если используемая хэш-функция не обеспечивает защиту от коллизий, то B может найти $H(m_1)=H(m_2)$, где m_1 действительное, заранее подписанное сообщение легальным пользователем. Затем он присоединяет ЦП $\langle r,s\rangle$ для сообщения m_1 к сообщению m_2 и отсылает сообщение $\langle m_2,\langle r,s\rangle\rangle$. Получатель при проверке ЦП не обнаружит подделки, и ему будет навязано ложное сообщение m_2.

Селективная подделка. Для подписи заранее выбранного сообщения m при неизвестном ключе d необходимо сформировать для сообщения m подпись $\langle r,s\rangle$ так, чтобы проверка на целостность и подлинность этого сообщения m давала положительный результат. Рассмотрим алгоритм подделки подписи для m.

1) Формируется или выбирается $k^c\in(1,2,...,n-1)$.
2) Вычисляем $r^c=\pi(k\times G)$.
3) Выбираем или подбираем $s^c\in\{1,...,n-1\}$.
4) Посылаем ложное сообщения M^c с подписью $\langle r^c,s^c\rangle$.

Получатель при приеме проверяет целостность и подлинность сообщения $\langle M^c,\langle r^c,s^c\rangle\rangle$. Для этого он выполняет следующее.

1. Вычисляет значение хэш-функции $e'=h(M^c)$.
2. Вычисляет значение параметров $w=(s^c)^{-1}(\bmod n)$, $u_1=e'w(\bmod n)$ и $u_2=r^cw(\bmod n)$.
3. Находит точку эллиптической кривой $(x,y)=u_1\times G+u_2\times Q$.
4. Преобразует точку эллиптической кривой $v=\pi(x,y)\bmod n$.
5. Сравнивает $r^c\overset{?}{=}v$.

Проверка на 5-м шаге будет выполнена только в том случае, если $s^c=(k^c)^{-1}(dr^c+e)\bmod n$. Анализ этого выражения показывает, что вероятность правильного выбора s^c однозначно определяется вероятностью подбора или угадывания ключа d и составляет для ECDSS очень малую величину. Аналогичной стойкостью против селективной подделки обладают подписи ECSS, ECCDS, ECKCDSA и Шнора.

Полное раскрытие. По современным понятиям стойкость всех приведенных алгоритмов ЦП основана на сложности решения дискретного логарифма в мультипликативной группе точек эллиптической кривой. Для нахождения секретного ключа необходимо решить относительно d сравнение

$$Q=d\times G, \tag{1}$$

(в случаях *ECDSA* и *ECSS*), сравнение

$$Q=d^{-1}\times G, \tag{2}$$

(в случаях $EC-GDSA$ и $EC-KCDSA$), и сравнение

$$Q=-d\times G, \tag{3}$$

в случае алгоритма Шнора.

Рассмотрим возможность нахождения d по перехваченным подписанным сообщениям. Пусть перехвачено i подписанных сообщений. Решая для ECDSS сравнение

$$s = k^{-1}(dr + e) \bmod n,$$

относительно d получим

$$d = {(ks - e)}/{r} \pmod n.$$

Для i сообщений получим i сравнений с $i+1$ неизвестными (4), т.е. $k_1, k_2, ..., k_i$ и d. По аналогии из сравнения $s = (k - dr) \bmod n$ для ECSS имеем (5), т.е. получим тоже i сравнений с $i+1$ неизвестными $k_1, k_2, ..., k_i$ и d.

$$\begin{cases} d = {(k_1 s_1 - e_1)}/{r_1} \bmod n, \\ \vdots \\ d = {(k_i s_i - e_i)}/{r_i} \bmod n. \end{cases} \qquad \begin{cases} d = {(k_1 - s_1)}/{r_1} \bmod n, \\ \vdots \\ d = {(k_i - s_i)}/{r_i} \bmod n. \end{cases} \qquad (4, 5)$$

Для алгоритма Шнора используя сравнение $s = (dr + k) \bmod n$, также получаем i сравнений с $i+1$ неизвестными:

$$\begin{cases} d = {(s_1 - k_1)}/{r_1} \bmod n, \\ \dots \\ d = {(s_i - k_i)}/{r_i} \bmod n. \end{cases} \qquad (6)$$

Аналогично используя алгоритмы ECCDSA и ECKCDSA, можно получить соответственно системами сравнений i-го порядка с $i+1$ неизвестными:

$$\begin{cases} d = {s_1}/{(r_1 k_1 - e_1)} \bmod n, \\ \dots \\ d = {s_i}/{(r_i k_i - e_i)} \bmod n. \end{cases} \qquad \begin{cases} d = {s_1}/{(k_1 - e_1)} \bmod n, \\ \dots \\ d = {s_i}/{(k_i - e_i)} \bmod n. \end{cases} \qquad (7, 8)$$

Таким образом, для полного раскрытия, т.е. определенная секретного ключа по i полученным ЦП на ключе d, необходимо решать систему i-го порядка с $i+1$ неизвестными.

В случае, если сообщение M является зашифрованным, то неизвестными являются значения хэш-функций $e_1, e_2, ..., e_i$. В результате для EC–KCDSA и ECDSA получим систему уравнений с $2i+1$ неизвестными, поэтому шифрование подписанных сообщений позволяет существенно повысить стойкость. Но этим свойством не обладают алгоритмы Шнора и ECSS.

Отличительной особенностью ЦП EC-KCDSA является введение дополнительно параметра Z_A. В качестве дополнительного параметра могут использоваться время, идентификаторы, пароли, псевдослучайные данные и др. Причем вследствие стойкости *EC-KCDSA* к экзистенциальной атаке подделать дополнительную информацию для устаревшей ЦП практически не возможно.

3. Характеристики известных хэш-функций

Считаем необходимым привести достаточно полный перечень разработанных и используемых в настоящее время функций хэширования [14].

Таблица 1

Название функции	Класс	Длина значения, (бит)
Whirlpool	Однонаправленная хэш-функция	512
SHA-2	Однонаправленная хэш-функция	512 (256, 384)
SHA-2	Однонаправленная хэш-функция	256 (384, 512)
ГОСТ 34311-95	Однонаправленная хэш-функция	256
HAVAL	Однонаправленная хэш-функция	256 (128, 160,192)
Tiger	Однонаправленная хэш-функция	192
SHA-1	Однонаправленная хэш-функция	160
RIPEMD	Однонаправленная хэш-функция	160
MD4	Однонаправленная хэш-функция	128
MD5	Однонаправленная хэш-функция	128

Указанные хэш-функции могут быть использованы при реализации ЦП. Выбор той или иной ЦП зависит от требований по стойкости и конкретных значений параметров ЦП, прежде всего модуля (порядка базовой точки) G. Наличие односторонних коллизионностойких хэш-функций со значениями длин 128, 160, 192, 256, 384, 512 дает возможность строить с такими же длинами ее составляющих r и s.

4. Анализ стойкости ЦП

Проведенный анализ показывает [3-9], что для каждой из расмотренных ЦП существует возможность правильного выбора или подбора ключей k_i, d, d^{-1}, но вероятности этих событий чрезвычайно малы, поэтому обычно их не рассматривают. По сути, очень малая вероятность этих событий обеспечивается выбором величины модуля n. Поэтому атаки типа "грубая сила" маловероятны и не могут быть реализованы для подделки ЦП или формирования ЦП для ложных сообщений.

На наш взгляд важнейшим фактором оценки стойкости ЦП является ее устойчивость против появления со временем эффективных криптоаналитических атак или обнаружения лазеек. Алгоритм DSA уже боле 10 лет выдержал испытание временем, поэтому следует ожидать, что и его модификации, реализованные в группах точек эллиптических кривых, будут наследовать реальную криптостойкость.

Рассмотрим более подробно криптографические атаки, основанные на решении дискретного логарифмического уравнения (9)-(11) в группах точек эллиптических кривых. Задача формируется следующим образом.

Криптоаналитику известен вид ключевых уравнений

$$Q = d \times G\left(\mathrm{mod}\, f(x), p\right),\ Q = d^{-1} \times G\left(\mathrm{mod}\, f(x), p\right) \qquad (9, 10)$$

или

$$Q = -d \times G\left(\mathrm{mod}\, f(x), p\right), \qquad (11)$$

где Q – открытый ключ, а G – базовая точка порядка n. Необходимо вычислить или найти личный ключ d.

Наиболее простым методом поиска d есть атака «грубая сила». В этом случае криптоаналитик вычисляет Q_i изменяя $d = 1, 2, 3, \ldots, n-1$, пока не получим значение Q_j открытого ключа. Возможность выполнения этой атаки существует всегда. Выбирая величину n, эту атаку можно сделать чрезвычайно маловероятной.

- Алгоритм Полинга-Хеллмана. Впервые предложен в [15]. В его основе лежит решение задачи факторизации порядка n базовой точки G. Алгоритм понижает сложность нахождения ключа за счет решения дискретного логарифма по каждому из модулей, полученных при факторизации $n-1$. При нахождении ключа используется китайская теорема об остатках. Для защиты от этой атаки необходимо использовать базовую точку с простым порядком n и большим простым делителем порядка $n-1$.
- Алгоритм «маленьких» и «больших» шагов [16]. В отличие от «грубой силы», этот алгоритм обеспечивает компромисс между быстродействием и используемой памятью. Он требует хранения порядка $\sqrt{n}$ точек и времени выполнения в худшем случае $\sqrt{n}$ шагов. При $n \geq 2^{160}$ и соответственно $\sqrt{n} \geq 2^{80}$ этот метод практически не реализуем.
- Алгоритм ρ-Полларда. Алгоритм описан в [17] и представляет собой вероятностную версию алгоритма «маленьких» и «больших» шагов. Он требует почти таких же вычислений, но значительно меньше объема памяти для хранения промежуточных результатов.

В работе [18-19] показано, как ускорить алгоритм ρ-Полларда в $\sqrt{2}$ раз.

- Распределенный алгоритм ρ-Полларда. В работе [20] предложен метод распределения алгоритма на r процессоров, что позволяет увеличить скорости в $\sqrt{\pi n}/(2r)$ раз.
- Алгоритм λ-Полларда [18]. Этот метод, как и ρ-метод, является вероятностным и рассмотрен в [18]. Как и ρ-Поллард λ-Поллард может быть распределен на r процессоров. Метод быстрее ρ-метода, если искомый логарифм находиться в подинтервале $[0 \ldots 0{,}39n]$.
- Составной логарифм. В [21] показано, как при известном разложении дискретного логарифма в группах точек эллиптической кривой E с базовой точкой G можно ускорить метод ρ-Полларда с использованием тех же эллиптической кривой и базовой точки. Более точно, если в первом случае поиск дискретного логарифма занимает t времени, то во втором случае понадобится только $\left(\sqrt{2}-1\right)t \approx 0.41t$. Имея решение двух случаев, ожидаемое время в третьем случае составляет $\left(\sqrt{3}-\sqrt{2}\right)t \approx 0.32t$. Строя последовательность известных разложений для заданной эллиптической кривой и базовой точки, можно уменьшить сложность дискретного логарифма в последующих шагах, т.е. для нахождения последующих ключей при тех же общесистемных параметрах.

Во избежании атаки данного типа, необходимо выбирать параметры эллиптической кривой, чтобы первый шаг был невыполним.

- Суперсингулярные эллиптические кривые. В работах [22-24] показана возможность понижения сложности дискретного логарифма на эллиптической кривой E, определенной над F_q, до сложности дискретного логарифма в мультипликативной группе в расширении F_{q^k}, где $k \geq 1$, для которого может быть применен алгоритм квадратичного решета. Такая атака возможна при существовании малых k. Для проверки устойчивости кривой к этой атаке необходимо проверить, что бы n не делилось на $q^k - 1$.
- Аномальные кривые над простым полем. Эллиптическая кривая E над полем F_p называется аномальной, если порядок $\#E(F_p) = p$. В работах [25-27] приведены эффективные алгоритмы разложения дискретного логарифма для аномальных кривых. Атака не применима к кривым, у которых порядок $\#E(F_p) \neq p$.

- Кривые, определенные над малым полем. Предполагается, что E определена над F_{2^e}. В [18,28] показано, как метод ρ-Полларда на эллиптической кривой $E\left(F_{2^{ed}}\right)$ в этом случае может быть ускорен в $\sqrt{d}$ раз.

Из проведенного анализа, а также из [17-27] следует, что наилучшим алгоритмом является распараллеленный алгоритм ρ-Полларда. На данный момент максимальная длина "взломанной" эллиптической кривой по данным [30] составляет 109 битов. Для решения этой задачи использовался метод ρ-Полларда и около 50000 компьютеров Pentium Pro 200Mhz на протяжении 2 месяцев.

Важной характеристикой различных ЦП есть сложность выработки и проверки ЦП. Наибольшей сложностью обладают алгоритмы *EC-KCDSA* и Шнора. Это связанно с возможностью выноса процедуры хэширования в алгоритмах *ECDSA*, *EC-GDSA* и *ECSS* и невозможностью в *EC-KCDSA* и Шнора.

Для ускорения алгоритмов можно произвести *некоторую* модификацию. Так в алгоритме Шнора можно заменить $e=h(M\|x)$ на $e=h(h(M)\|x)$, в $EC-KCDSA$ заменить $H=h(Z_A\|M)$ на $H=(Z_A\|h(M))$, что позволит понизить зависимость производительности ЦП от хэш-функции.

Вторым фактором, влияющим на производительность, – есть нахождения при выработке и проверке подписи нахождения обратного элемента. В алгоритме EC-GDSA удалось избежать этой сложной операции, подписи за счет особого способа вычисления открытого ключа $Q=d^{-1}G$. Применение такого способа в EC-KCDSA позволяет уменьшить сложность выработки и проверки ЦП.

Наименее сложным и следовательно наиболее быстрым может быть алгоритм ЦП Шнора, если вычисление хэш-значения вынести за пределы алгоритма ЦП. Если это не сделать, то минимальная вычислительная сложность обеспечивается в алгоритме ECSS.

Во всех алгоритмах ЦП существует возможность существенного повышения производительности за счет выноса самой сложной операции скалярного умножения за пределы алгоритма ЦП. Это можно сделать за счет предварительного формирования таблиц (k_i,k_iG) пар. Одна из пар может быть выбрана случайно по необходимости. Однако такие таблицы должны формироваться, храниться и использовать как секретные, аналогичные личным ключам.

Алгоритм решения сравнения на основе ρ-Полларда, примеры нахождения личного ключа и результаты оценки сложности криптоанализа приведены в [31].

5. Проблемные вопросы и перспективы развития преобразований в группах точек эллиптической кривой

Ряд источников [32, 33] показывает, что возможным направлением усовершенствования методов и средств криптографических преобразований может быть применение гиперэллиптических кривых [33] и эллиптических кривых над оптимальным расширенным полем [32].

Гиперэллиптической кривой С степени $g\geq 1$ над полем F_q называется кривая вида

$$y^2+h(x)y=f(x), \tag{12}$$

где $h(x)$ – полином степени $\leq g$ (может быть $h(x)=0$); $f(x)$ – полином степени $2g+1$.

Пример: $y^2=x^5+1$.

При $g=1$ получаем частный случай – обычную эллиптическую кривую.

Множество рациональных точек на гиперэллиптической кривой не формирует группу, поэтому в качестве элементов группы выбраны так называемые делители [33]:

$$D = \sum_{P \in C} m_p P,$$

где m_p – целые числа.

Групповая операция обладает свойствами:

$$\sum_{P \in C} m_p P + \sum_{P \in C} n_p P = \sum_{P \in C} (m_p + n_p) P .$$

Достоинства эллиптических кривых:

1) порядок поля F_q может быть малым (50-80 бит);

2) большее множество возможных кривых.

Недостаток – большая вычислительная сложность групповой операции.

Одна из проблем, возникающих при реализации эллиптических кривых над полем $GF\left(2^n\right)$, – большая вычислительная сложность операций в этом поле. Одно из предложений – использование поля $GF\left(\left(2^n\right)^m\right)$. Однако показано, что такие кривые являются нестойкими. Поэтому предлагается применять поле $GF\left(p^m\right)$, где p – простое число вида $2^n \pm c$. При этом m выбирается таким образом, чтобы существовал неприводимый полином $P(x) = x^m - w$, $w \in GF(p)$.

Существует 2 типа OEF:

- тип $1 - c = 1$;
- тип $2 - w = 2$.

Примеры OEF: $GF\left(\left(2^{31}-1\right)^6\right)$, $GF\left(\left(2^8-17\right)^{17}\right)$, $GF\left(\left(2^{13}-1\right)^{13}\right)$.

Заключение

В настоящее время вместо криптографических преобразований в кольцах и полях начали применять преобразования в группах точек эллиптической кривы. Разработаны и рекомендованы к применению ЦП Шнора, *ECSS*, *EC-DSA*, *EC-GDSA* и *EC-KCDSA*. Основными преимуществами ЦП, реализованных за счет преобразований в группах точек эллиптической кривых, является возможность уменьшения длин ключей при одновременном уменьшении сложности прямых и обратных преобразований.

Все рассмотренные алгоритмы ЦП обладают практически одинаковой стойкостью. Алгоритм *EC-KCDSA* обеспечивает защиту от навязывания ранее переданных подписанных сообщений, что достигается за счет применения дополнительной информации – например, времени формирования ЦП.

Наибольшая минимизация сложности прямого и обратного преобразований может быть достигнута с использованием алгоритмов Шнора или *ECSS*. Дальнейшее уменьшение сложности преобразований может быть достигнуто за счет использования проективного базиса.

Использование нескольких значений n порядка базовой точки эллиптической кривой позволят обеспечивать различные уровни стойкости. В качестве базовых порядков и точек эллиптических кривых можно выбрать значения $n = 196$, 224, 256, 320, 384, и 512. Одним из факторов выбора этих значений n есть наличие статических алгоритмов вычисления хэш-функции.

Список литературы: 1. IEEE P1363 / D9 (Draft Version 9). Standard Specifications for Public Key Cryptography. Number-Theoretic Background. 1999 2. X9.62 – 1999 Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA) 3. X9.42 - 1998, Public Key Cryptography for The Financial Service Industry : Agreement of Symmetric Keys on Using Diffie-Hellman and MQV Algorithms 4. По материалам Chip Magazine 5. *A. Menezes, P. van Oorschot, S. Vanstone* Handbook of Applied Cryptography. CRC Press, 1997. 6. PKCS#1: RSA Encryption Standard. Version 2.0. RSA Data Security Inc. 1997 7. *Diffie W., Hellman M.E.* New Direction in Cryptography / IEEE Trans. Inf. Theory., Nov. 1976, IT-22, 644-654 8. *ElGamal T.A.* Public-Key Cryptosystem and Signature Scheme Based on Discrete Logarithm // Proc. Of CRYPTO'84 Lecture Notes in Comp. Sci. Springer-Verlag, 1985, V. 196, 10-18 9. *Иванов М.А.* Криптографические методы защиты информации в компьютерных системах и сетях. М.: КУДИЦ-ОБРАЗ, 2001. 368с. 10. *Pointcheval D., Stern J.* "Security proofs for signature scheme", Eurocrypt'96, Springer-Verlag, 387-398 11. *Beaver D., Feigenbaum J., Shoup V.* "Hiding instance in zero-knowledge proof system", Crypto'90, Springer-Verlag, 326-338 12. Введение в криптографию. / Под общей редакцией *Ященко В.В.*. М.:МЦНМО-ЧеРо, 2000. 268с. 13. *Don Johnson , Alfred Menezes, Scott Vanstone* "The Elliptic Curve Digital Signature Algorithm (ECDSA)", Certicom Research, Canada 14. . E. Biham. On the Applicability of Differential Cryptoanalysis to Hash Functions. In E.I.S.S Workshop on Cryptographic Hash Functions, pages 25-27, March 1992. 15. *D. Pointcheval, J. Stern* "Security proofs for signature schemes", Advancer in Cryptology – Eurocrypt '96, Lecture Notes in Computer Science, 1070 (1993), Springer-Verlag, 387-398 16. *S. Pohlig and M. Hellman* "An improved algorithm for computing logarithm over $GF(p)$ and its cryptographic significance", IEEE Translation on Information Theory, 24 (1978) 106-110. 17. *J. Pollard* "Monte Carlo methods for index computation mod p " , Mathematics of Computation, 32 (1978), 918-924 18. *R. Gallant , R. Lambert, S. Vanstone* "Improving the parallelized Pollard lambda search on binary anomalous curves " to appear in Mathematics of Computation. 19. *M. Wiener, R. Zuccherato* "Faster attacks on elliptic curve cryptosystem", Selected Areas in Cryptography, Lecture Notes in Computer Science, 1556 (1999), Springer-Verlag 252-266 20. *P. van Oorschot, M. Wiener* "Parallel collision search with cryptanalytic application" Journal of Cryptography, 12 (1999), 1-28 21. *R. Silverman and J. Stapleton*, Contribution to ANSI X9F1 working group, 1997. 22. *A. Menezes*, Elliptic Curve Public Key Cryptosystems, Kluwer Academic Publishers, Boston, 1993 23.*A.Menezes, T. Okamoto and S. Vanstone*, "Reducing elliptic curve logarithms to logarithms in a finite field", IEEE Transactions on Information Theory, 39 (1993), 1639-1646. 24. *G. Frey and H. Riick*, "A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves", Mathematics of Computation, 62 (1994), 865-874. 25.I.Semaev, "Evaluation of discrete logarithms in a group of p-torsion points of an elliptic curve in characteristic p", Mathematics of Computation, 67 (1998), 353-356. 26. N. Smart, "The discrete logarithm problem on elliptic curves of trace one", Journal of Cryptology, 12 (1999), 193-196. 27. T. Satoh and K. Araki, "Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves", Commentarii Mathematici Universitatis Sancti Pauli, 47 (1998),81-92. 28. M. Wiener and R. Zuccherato, "Faster attacks on elliptic curve cryptosystems", Selected Areas in Cryptography, Lecture Notes in Computer Science, 1556 (1999), Springer-Verlag, 190-200. 29. A. Escott, J. Sager, A. Selkirk and D. Tsapakidis, "Attacking elliptic curve cryptosystems using the parallel Pollard rho method", CryptoBytes, The Technical Newsletter of RSA Laborato- ries, volume 4, number 2, Winter 1999, 15-19. Also available at http://www.rsasecurity.com 30.*Michael J. Wiener, Robert J. Zuccherato* Faster Attacks on Elliptic Curve Cryptosystems, 1998, 8 с.31. *Горбенко И.Д., Збитнев С.И., Поляков А.А.* Криптоанализ криптографических преобразований в группах точек эллиптических кривых методом Поларда // Радиотехника: Всеукр. межвед. науч.-техн. сб. 2001. Вып. 119. С.43-50 32. *N. Kobliz* "Hyperelliptic cryptosystem", Journal of Cryptology, 1 (1989), 139-150. 33. *M. Petersen* "Hyperelliptic cryptosystem", Technical Report, University of Aarhus, Denmark, 1994

Харьковский национальный университет радиоэлектроники

Поступила в редколлегию 22.04.2002