Шлюз безопасности “криптомаршрутизатор” реализован в виде дополнительной рабочей станции, устанавливаемой на выходе локальной сети. В качестве программной платформы шлюза используется операционная система Windows 2000 Professional. В реализации шлюза используется часть встроенных сетевых компонентов (в частности программный маршрутизатор) операционной системы. Шлюз безопасности построен на основе разработанного сетевого драйвера, встраиваемого в подсистему сетевых драйверов NDIS. Шлюз выполняет обработку IP-трафика и совместим с локальными сетями (в данной версии только стандарта IEEE 802.3).

Создание защищенного туннеля между шлюзами безопасности или шлюзом безопасности и агентом удаленного пользователя обеспечивается за счет шифрования, имитозащиты и инкапсуляции пакетов с использованием сеансовых ключей. В процессе инкапсуляции пакетов исходный IP-заголовок заменяется на новый, обеспечивая передачу инкапсулированного пакета на шлюз безопасности или хост удаленного пользователя. Контекст туннеля безопасности содержит счетчик пакетов, обеспечивающий защиту от повторов инкапсулированных пакетов. Кроме того, шлюз безопасности может автоматически отключать “поврежденные” туннели.

Криптографическая защита в туннеле безопасности осуществляется с использованием поточного шифрования каждого пакета и вычисления имитовставки на сеансовых ключах. В качестве базовых используются алгоритмы ГОСТ 28147-89 и Х-ГОСТ. Сеансовые ключи защищаются с использованием алгоритмов Х-ГОСТ, причем каждый пакет защищается на отдельном сеансовом ключе. Криптографический контекст туннеля содержит транспортный ключ, на котором передаются заголовки инкапсулированных пакетов, включая и сеансовый ключ.

Управление шлюзом безопасности “криптомаршрутизатор” может осуществляться локально или удаленно. При удаленном управлении шлюз безопасности функционирует в автоматическом режиме. Администратор безопасности сети выполняет настройку шлюза безопасности через своего программного агента установленного на шлюзе. При подключении удаленного администратора к шлюзу безопасности вначале выполняется их взаимная идентификация. Удаленный администратор может управлять работой шлюза, настраивая правила безопасности (таблицы туннелирования и маршрутизации) и форматирования потоков данных, проходящих через шлюз. Установка транспортных ключей в криптографический контекст туннеля безопасности осуществляется удаленно. Для защиты данных управления используются многоэтапные состоятельные протоколы, базирующиеся на ГОСТ 28147-89, ГОСТ 34310-95, ГОСТ 34311-95, а также X9.42.

Заключение

Разработанный шлюз безопасности применяется для защиты потока данных между локальными сетями, обеспечивая управление доступом к ресурсам локальной сети из других сетей или с хостов удаленных пользователей, в соответствии с устанавливаемыми правилами безопасности. Однако реализованный проект требует доработки. Среди основных – реализация поддержки дополнительных протоколов канального уровня, используемых в современных локальных сетях (так как данная версия поддерживает только Ethernet). Кроме того, необходима реализация возможности сбора статистической информации и информации о состоянии туннелей со шлюзов безопасности, а также возможности быстрого восстановления всей сети шлюзов безопасности в случае сбоя.

УДК 681.3.06:519.248.681

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПОКАЗАТЕЛЕЙ СЛОЖНОСТИ И СТОЙКОСТИ ЦИФРОВЫХ ПОДПИСЕЙ ECSS И ECDSA

Станислав Збитнев, Игорь Коновалов*, Даниил Меалковский**, Андрей Поляков
*Харьковский Национальный Университет Радиоэлектроники,*АТ “ИИТ”, **ДСТСЗИ СБУ*

Анотація: **Здійснюється порівняльний аналіз RSA-подібних систем з цифровими підписами на еліптичній кривій у скінченних полях. Аналізуються основні алгоритми цифрових підписів побудованих на основі еліптичної кривої (ECDSA і ECSS). Даються рекомендації з використання цифрових підписів.**

Summary: **Produce comparative analysis RSA – similar systems with digital signatures on elliptic curves in finite fields. Analyzed main algorithms of digital signatures on elliptic curves (ECDSA and ECSS). Given recommendations on using the digital signatures.**

Ключові слова: **Эллиптическая кривая, цифровая подпись, модуль преобразований, криптоанализ, общесистемные параметры, модель угроз.**

Обеспечение целостности и наблюдаемости информации и ресурсов в различных информационных технологиях обеспечивается, прежде всего, за счет использования цифровой подписи (ЦП) [1]. На сегодня разработано, аттестовано и используется ряд аппаратов ЦП, которые базируются на преобразованиях в кольцах, полях и подполях (подгруппах).

Основными требованиями, которые предъявляются к алгоритмам ЦП, есть требования стойкости и минимизации сложности (максимизации скорости) алгоритмов вычисления ЦП.

В первых средствах, реализующих асимметричные криптоалгоритмы ЦП, применяли сравнительно небольшие длины модулей преобразования, что позволяло обеспечить требуемый уровнь стойкости. С развитием математических методов криптоанализа и ростом производительности ЭВМ стойкость стали обеспечивать за счет увеличения длины модулей преобразования.

Рост мощностей криптоаналитических систем вынуждает увеличивать длины модулей преобразований, что не позволяет формировать и проверять цифровые подписи в реальном масштабе времени. Так, к примеру, по данным RSA Labs [2] для обеспечения требуемого уровня безопасности при использовании алгоритма RSA длина модуля преобразования должна составлять не менее 1024 бита. Использование математического аппарата над группой точек эллиптической кривой позволило модифицировать существующие алгоритмы цифровой подписи, так что они обеспечивают требуемый уровень безопасности при сравнительно небольшой длине блока преобразования. Представляет практическое значение решение задачи сравнения получивших наибольшее распространения алгоритмов ЦП, реализуемых за счет преобразований в полях и подполях, а также их модификации за счет использования преобразований в группах точек ЭК.

На рис. 1 и 2 приведены графики зависимости сложности криптоанализа в зависимости от длины модуля преобразований для преобразований в простом поле и преобразований в группе точек.

$$I_{prime} = e^{\left(1.923(\log p)^{\frac{1}{3}}(\log(\log p))^{\frac{2}{3}}\right)} \qquad I_{\rho} = \sqrt{\frac{\pi n}{4}}$$

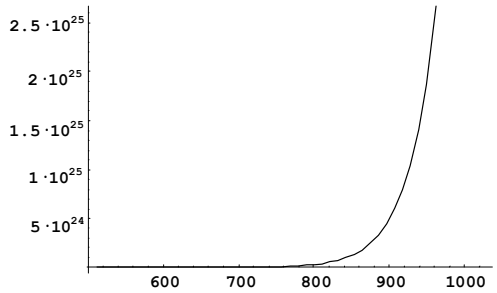

Рисунок 1 – Сложность атаки для преобразований в простом поле

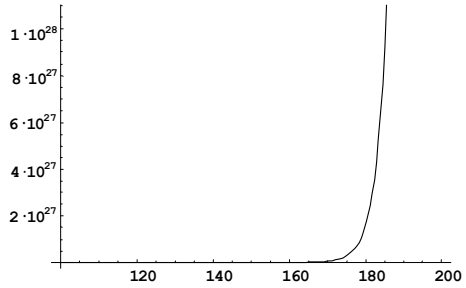

Рисунок 2 – Сложность атаки для преобразований в группе точек на эллиптической кривой

Из графиков видно, что при длине ключа $n \geq 160$ бит, криптосистемы, построенные на эллиптической кривой, имеют стойкость, сравнимую со стойкостью криптосистем, построенных в простом поле с модулем преобразования $p \geq 1024$.

Очевидно, что построенные на эллиптических кривых криптографические алгоритмы пройдут тот же эволюционный путь, что и криптосистемы RSA, Эль-Гамаля и Диффи-Хелмана, однако на сегодняшний день они обеспечивают наиболее высокую стойкость.

Целью данной статьи является проведение сравнительного анализа получивших наибольшее распространение цифровых подписей ECSS и ECDSA, реализуемых с использованием преобразований в группах точек эллиптических кривых [3].

I Сравнение ЦП ECSS и ECDSA

Алгоритм ЦП ECSS заключается в следующем:

Формирование ЦП (стороной *A*)**.**

Входные параметры: секретный ключ *d*, сообщение *M*, базовая точка *P*, порядок точки *n*,

Проверка ЦП (стороной *B*).

Входные параметры: открытый ключ Q_A

параметры ЭК *a*, *b*

Выход: цифровая подпись (r, s).

1. Преобразуем сообщение M в бинарную строку.
2. Используя хеш-функцию вычисляем $h = H(M)$.
3. Выбираем случайное целое число *k* (сеансовый ключ) в интервале [1, *n* –1].
4. Вычисляем точку $(x_k, y_k) = kP$.
5. Вычисляем $r = x_k + h \mod n$.
6. Используя личный ключ *d*, вычислим $s = k - dr \bmod n$
7. Посылаем стороне *B* сообщение *M* и подпись (r, s).

стороны *A*, сообщение *M*, цифровая подпись (r, s), базовая точка *P*, порядок точки *n*, параметры ЭК *a*, *b*

Выход: принятая подпись или отвергнутая подпись.

1. Берем открытый ключ Q_A стороны *A*.
2. Вычисляем точку $(x_k, y_k) = sP + rQ_A$.
3. Вычисляем хеш по сообщению *M*: $h = H(M)$.
4. Вычисляем $r' = x_k + h \mod n$.
5. Принимаем подпись сообщения *M* стороны *A*, если $r = r'$; иначе отвергаем подпись.

Алгоритм ЦП ECDSA состоит из следующих этапов:

Входные и выходные параметры ЦП ECDSA совпадают с ECSS.

Формирование ЦП (стороной *A*)**.**

1. Преобразуем сообщение M в бинарную строку.
2. Используя хеш-функцию, вычисляем $h = H(M)$.
3. Выбираем случайное целое число *k* (сеансовый ключ) в интервале [1, *n* –1].
4. Вычисляем точку $(x_k, y_k) = kP$.
5. Вычисляем $r = x_k \bmod n$.
6. Используя личный ключ *d*, вычислим $s = k^{-1}(h + dr) \mod n$.
7. Посылаем стороне *B* сообщение *M* и подпись (r, s).

Проверка ЦП (стороной *B*).

1. Берем открытый ключ Q_A стороны *A*.
2. Если $(r \bmod n) = 0$ отвергаем сообщение.
3. Вычисляем хеш по сообщению *M*, $h = H(M)$.
4. Вычислим $s^{-1} \bmod n$
5. Вычисляем $u = s^{-1}h \mod n$ и $v = s^{-1}r \bmod n$
6. Вычисляем точку $(x_k, y_k) = uP + vQ_A$.
7. Вычисляем $r' = x_k \bmod n$.
8. Принимаем подпись сообщения *M* стороны *A*, если $r = r'$; иначе отвергаем подпись.

Проведем сравнительный анализ ECDSA и ECSS. В случае подписи ECSS из шагов алгоритма формирования подписи имеем:

$$\begin{cases} e = r - x_k, \\ e = (k - s)\ d^{-1} - x_k, \end{cases} \quad или \quad rd = k - s \tag{2}$$

При использовании подписи ECDSA из шагов 5, 6 алгоритма формирования подписи получаем, что:

$$\begin{cases} x_k = r, \\ x_k = (sk - e)\ d^{-1}, \end{cases} \quad или \quad rd = sk - e \tag{3}$$

Из (2) следует, что криптоанализ ЦП ECSS можно свести к решению сравнения от двух неизвестных d и k (личный и сеансовый ключи). Для ECDSA криптоанализ сводится к решению неравенства с тремя неизвестными d, k и e (личный ключ, сеансовый ключ и хеш-функция от сообщения). Если сообщение не является конфиденциальным, то решение задачи сводится также к сравнению от двух неизвестных d и k (личный и сеансовый ключи).

Сравнение алгоритмов ЦП показало, что ECSS обладает меньшей вычислительной сложностью, чем ECDSA. Это достигается за счет:

1. отказа от вычисления обратного элемента в простом поле при формировании и проверке ЦП ECSS;
2. отказа от 2-х умножений в простом поле по модулю при проверке ЦП ECSS.

Из приведенного выше видно, что для криптосистем, требующих максимальной производительности,

предпочтительно использовать алгоритм ЦП ECSS, а для достижения максимальной стойкости – алгоритм ЦП ECDSA.

Однако на наш взгляд, наиболее перспективным является алгоритм ECDSA, который обеспечивает по сравнению с ECSS наибольшую стойкость. Кроме того, существует алгоритм уменьшения сложности проверки ЦП для ECDSA.

В дальнейшем используем обозначения:

m – степень расширения поля;

M – исходное сообщение;

M' – полученное сообщение;

r' и s' – параметры цифровой подписи проверяемого сообщения;

h – значение хеш-функции;

$y \bmod x$ – остаток от деления y а x;

n – порядок базовой точки на эллиптической кривой;

G – базовая точка на эллиптической кривой;

d – долговременный секретный ключ;

k – сеансовая компонента ЦП;

$\langle r, s \rangle$ – параметры ЦП сообщения *M*;

$f(x)$ – примитивный полином (трехчлен или пятичлен) степени m;

u – порядок эллиптической кривой над полем $GF(2^m)$, который равняется $\#E\left(GF(2^m)\right)$;

a, b – параметры эллиптической кривой $y^2 + xy = x^3 + ax^2 + b\left(\bmod f(x), 2\right)$;

l_h – длина хеш-функции в битах, равная одному из значений: 160, 192, 256, 512.

II Формирование цифровой подписи в группах точек ЭК

Цифровая подпись формируется в следующей последовательности. Текст сообщения, поданный в виде битовой строки, поддается обработке согласно нижеописанному алгоритму, в результате которого для исходного сообщения формируется электронная цифровая подпись (ЭЦП).

Процедура подписи сообщения содержит следующие этапы.

1. Используя соответствующий алгоритм хеширования, вычислить значение хеш-функции $h = H(M)$ от сообщения длиной l_h. Если $h \equiv 0 \bmod n$, тогда h присвоить значение $0^{l_h - 1}1$
2. Сформировать целое случайное число k в диапазоне $[1, n-1]$ и вычислить точку $(x_1, y_1) = kG\left(\bmod f(x), 2\right)$ на эллиптической кривой.
3. Выделить значение x_1', точки (x_1, y_1).
4. Назначить $r = x_1' \bmod n$. Если $r = 0$, перейти на шаг 2.
5. Используя секретный ключ d пользователя (отправителя сообщения) вычислить $s = k^{-1}(h + dr) \bmod n$. Если $s = 0$, перейти к шагу 2. Если $s \neq 0$ закончить работу алгоритма.

   Подпись для сообщения M является вектором $\langle r, s \rangle$.

Отправитель направляет адресату цифровую последовательность символов, которая состоит из двоичной последовательности текста сообщения и добавленной к нему ЭЦП. Подписанное сообщение имеет вид $\{M, \langle r, s \rangle\}$.

III Проверка подписи

Получатель должен проверить подлинность сообщения и целостность ЭЦП. Это возможно при наличии у получателя открытого ключа отправителя и общесистемных параметров $\{f(x), n, G\}$.

Процедура проверки включает следующие этапы:

1. Если значения параметров ЦП r' или s' находятся вне интервала $[1, n-1]$, то остановить процесс и выдать сообщение “подпись недействительна”.
2. Используя соответствующий алгоритм хеширования, вычислить значение хеш-функции $h' = H(M')$ от сообщения длиной l_h. Если $h' \equiv 0 \bmod n$, тогда h' присвоить значение $0^{l_h-1}1$
3. Вычислить обратный элемент $c = (s')^{-1} \bmod n$.
4. Вычислить $u_1 = h'c \pmod n$ и $u_2 = r'c \pmod n$.
5. Вычислить точку на эллиптической кривой $(x_1, y_1) = (u_1 G + u_2 Q) \pmod{f(x), 2}$. Если точка принадлежит бесконечности, то остановить выполнение алгоритма и выдать сообщение “подпись недействительна”.
6. Выделить значение x_1' точки (x_1, y_1). Вычислить значение $\nu = x_1' \bmod n$.
7. Проверить $r' = \nu$.

Если $r' = \nu$, получатель принимает решение, что полученное сообщение подписано отправителем и в процессе передачи и хранения его целостность и подлинность не нарушены. Если $r' \neq \nu$, сообщение M' недействительно или изменено.

IV Процедура построения общесистемных параметров

Общесистемные параметры эллиптической кривой над полем $GF(2^m)$ должны содержать:

1. Поле размером $q = 2^m$, которое определяет поле $GF(q)$, с примитивным полиномом $f(x)$ степени m над двоичным полем $GF(2)$;
2. Случайная или псевдослучайная Случайная Строка (СС) длиной не меньше чем 160 бит;
3. Два элемента поля a и b в поле $GF(q)$, которые определяют уравнение эллиптической кривой $E: y^2 + xy = x^3 + ax^2 + b$;
4. Два элемента поля x_G и y_G в $GF(q)$, которые определяют базовую точку $G = (x_G, y_G)$ простого порядка n (Замечание $G \neq 0$);
5. Порядок n точки G (должен быть $n > 2^{160}$ и $n > 4\sqrt{q}$);
6. Кофактор $r = u / n$.

Известны три алгоритма построения общесистемных параметров – Скуфа, через подполя и метод комплексного умножения.

IV.1 Построение общесистемных параметров, используя алгоритм Скуфа

IV.1.1 Формирование параметров a, b, G, r, n

Вход: Поле размером q, нижняя граница $r_{\min}$, тривиальный делитель $l_{\max}$.

Выход: Элементы поля a, $b \in GF(q)$, которые определяют эллиптическую кривую (ЭК) над полем $GF(q)$, точка G простого порядка $n \geq r_{\min}$ на кривой, и кофактор $r = u / n$.

1. Если необходимо построить случайную ЭК, необходимо сформировать (СС, a, b) согласно IV.1.2 и вычислить u согласно алгоритму Скуфа.

2. Проверить $b \neq 0$ для уравнения ЭК вида:

$$y^2 + xy = x^3 + ax^2 + y \pmod{f(x), 2}.$$

3. Проверить порядок ЭК u на псевдопростоту. Если u не псевдопростое, перейти на шаг 1. В противном случае, $u = rn$ где $r = l_{\max}$ – сглаженое, и $n \geq r_{\min}$ псевдопростое.

4. Проверить на MOV случай с входными данными $B \geq 32$, q, и n. Если результат “*ошибка*”, перейти

на шаг 1. Проверить кривую на анормальность, т. е. $u = q$. Если результат "*ошибка*", перейти на шаг 1.

5. Найти точку G на E порядка n в соответствии с IV.1.3.

IV.1.2 Формирование битовой строки СС

Вход: Поле размером $q = 2^m$.

Выход: Битовая строка СС длины *m*.

Битовая строка СС может формироваться генератором случайной или псевдослучайной последовательности либо хорошим псевдослучайным датчиком.

IV.1.3 Нахождение базовой точки

Вход: Простое n, позитивное число r не делимое на n, ЭК E над полем $GF\left(2^m\right)$, порядка u.

Выход: Если $u = rn$, точка G на E порядка n. Если нет, сообщение "*ошибочный порядок*".

1. Сформировать случайную точку R (не O) на E используя IV.1.4.
2. Найти $G = rR \bmod\left(f\left(x\right),2\right)$.
3. Если $G =$ O, перейти на шаг 1.
4. Найти $Q = nG \bmod\left(f\left(x\right),2\right)$.
5. Если $Q \neq$ O, сообщение "*ошибка порядка*" и остановить выполнение алгоритма. Иначе выдать точку G.

IV.1.4 Нахождение случайной точки на ЭК

1. Сформировать случайное x в $GF\left(2^m\right)$.
2. Если $x = 0$ – сформировать $\left(0, b^{2^{m-1}}\right)$ и остановить выполнение алгоритма.
3. Назначить $\alpha = \left(x^3 + ax^2 + b\right) \bmod\left(f\left(x\right),2\right)$.
4. Если $\alpha = 0$ – сформировать $\left(x,0\right)$, и остановить выполнение алгоритма.
5. Установить $\beta = \left(x^{-2}\alpha\right) \bmod\left(f\left(x\right),2\right)$.
6. Вычислить z, если $z^2 + z = \beta$ и если существует решение.
7. Если z не существует, перейти на шаг 1.
8. Сформировать случайное μ и установить $y = \left(z + \mu\right)x$.

   Выдать $\left(x, y\right)$.

IV.2 Построение общесистемных параметров, используя подполя

IV.2.1 Вычисление u через подполя

Вход: Поле $GF\left(2^m\right)$; подполе $GF\left(2^d\right)$ для малого d, которое делит *m*; нижняя и верхняя границы $r_{\min}$ и $r_{\max}$ для базовой точки.

Выход: Элементы G, a, $b \in GF\left(2^m\right)$, которые определяют ЭК E порядка $u = \#E\left(GF\left(2^m\right)\right)$, если существует; иначе, сообщение "*нет кривой*".

1. Сформировать a_0, $b_0 \in GF\left(2^d\right)$, такие что b_0 еще не использовался. Если все b_0 перебраны, выдать сообщение "*нет кривой*" и остановить выполнение алгоритма. Пусть E ЭК:

   $E : y^2 + xy = x^3 + ax^2 + b \left(\bmod f\left(x\right),2\right)$.
2. Вычислить $\omega = \#E\left(GF\left(2^d\right)\right)$.
3. Вычислить $u = \#E\left(GF\left(2^m\right)\right)$.
4. Проверить u на псевдопростоту.
5. Если u псевдопростое, установить $\lambda = 0$, $n = u$ и перейти на шаг 9.

6. Установить $u' = 2^{m+1} + 2 - u$.

7. Проверить u' на псевдопростоту.

8. Если u' псевдопростое, установить $\lambda = 1$, $n = u'$, иначе перейти на шаг 1.

9. Найти элементы a_1, $b_1 \in GF(2^m)$, которые соответствуют a_0 и b_0.

10. Если $\lambda = 0$ установить $\tau = 0$. Если $\lambda = 1$ и m четное, установить $\tau = 1$. Иначе, найти элемент $\tau \in GF(2^m)$ используя трассировку.

11. Установить $a \leftarrow a_1 + \tau$ и $b \leftarrow b_1$.

Сформировать точку G согласно с IV.1.3. Выход u, a, b, G.

IV.3 Построение общесистемных параметров, используя комплексное умножение

IV.3.1 Нахождение почти простого порядка

Вход: Поле размером *d*, простой делитель $l_{\max}$, нижняя и верхняя границы $r_{\min}$, $r_{\max}$ базовой точки.

Выход: D свободное от корня, простое r в интервале $r_{\min} \le r \le r_{\max}$, сглаженое k, такие что $u = rn$, u – порядок ЭК над $GF(2^d)$ с комплексным перемножителем *D*.

1. Сформировать $D = 7 \pmod 8$ свободное от корня, которое ранее не формировалось.

2. Вычислить H – класс группы D.

3. Установить h – количество элементов в H.

4. Если d не делится на h, перейти на шаг 1.

5. Проверить в соответствии с IV.3.2, где D дискриминант для 2^d. Если результат "*не дискриминант*" – перейти на шаг 1. Иначе результат – W.

6. Вероятностный порядок $2^d + 1 \pm W$.

7. Проверить каждый порядок на почти простоту. Если какой-либо из порядков простой, выдать (D, n, r) и остановить выполнение алгоритма.

8. Перейти на шаг 1.

IV.3.2 Проверка дискриминанта

Вход: поле размером d и $D = 7 \pmod 8$, свободное от корня.

Выход: Если D дискриминант для 2^d, нечетное W такое, что

$$2^{d+2} = W^2 + DV^2,$$

для какого либо нечетного *V*. Если нет, сообщение "*не дискриминант*".

1. Вычислить $B^2 = -D \pmod{2^{d+2}}$.

2. Пусть $A \leftarrow 2^{d+2}$ и $C \leftarrow (B^2 + D) / 2^{d+2}$.

3. Пусть $S \leftarrow \begin{pmatrix} A & B \\ B & C \end{pmatrix}$ и $U \leftarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.

4. Пока $|2B| \le A \le C$, повторять шаги.

4.1 Пусть $\delta \leftarrow \left\lfloor \frac{B}{C} + \frac{1}{2} \right\rfloor$.

4.2 Пусть $T \leftarrow \begin{pmatrix} 0 & -1 \\ 1 & \delta \end{pmatrix}$.

4.3 Поменять U на $T^{-1}U$.

4.4 Поменять S на $T^{t} S T$, где T^{t} – транспонированная матрица T.

5. Пусть X и Y будут вхождением в U. Так что,

$$U = \begin{pmatrix} X \\ Y \end{pmatrix}.$$

6. Если $A = 1$, выдать $W \leftarrow X$ и остановить выполнение алгоритма.
7. Если $A = 4$ и Y четные, выдать $W \leftarrow (4X + BY)/2$ и остановить выполнение алгоритма.

Выводы

1. Реализация ЦП в группах точек ЭК позволяет повысить стойкость криптопреобразований и одновременно уменьшить размеры общесистемных параметров и ключей. По сравнению с преобразованиями в полях с ростом длины модуля преобразований выигрыш увеличивается.
2. Проведенные нами сравнения показали, что алгоритм ECDSA обладает рядом преимуществ в области стойкости.
3. Алгоритмы ECDSA и ECSS обладают приблизительно одинаковой вычислительной сложностью прямых и обратных преобразований. Нами выявлены алгоритмы для ECDSA, позволяющие ускорить процедуры проверки подписи.
4. С учетом возможного появления новых методов и более производительных средств криптоанализа необходимо предусмотреть различные значения параметров ЭК. Например, использовать кривые с порядком базовых точек 2^{160}, 2^{192}, 2^{224}, 2^{256}, 2^{384}, 2^{512}.

31 октября 2001 года завершился первый двухлетний этап рассмотрения алгоритмов – кандидатов на стандарты в разных областях криптографии. После тщательного исследования различными криптографами мира алгоритм ЦП ECDSA прошел первый отборочный тур, а алгоритм ECSS не был представлен на рассмотрение [4].

Литература: 1. IEEE P1363 / D9 (Draft Version 9). Standard Specifications for Public Key Cryptography. Number-Theoretic Background. 1999. 2. www.rsa.com 3. X9.62 – 1999 Public Key Cryptography For The Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA). 4. crypto.nessie.org.

УДК 681.3

КОНТРОЛЬ И ВОССТАНОВЛЕНИЕ ЦЕЛОСТНОСТИ ИНФОРМАЦИИ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ

Николай Будько, Вячеслав Василенко, Михаил Короленко
Открытое акционерное общество "КП ВТИ"

Аннотация: **Предлагается использование помехоустойчивого корректирующего кода для задач контроля, контроля и восстановления целостности информационных объектов автоматизированных систем.**
Summary: **Use of a noiseproof adjusting code for tasks control, control and restoration of information integrity the objects automated systems is offered.**
Ключевые слова: **Информация, целостность, контроль, восстановление.**

I Введение

В современных условиях обеспечение высокой надежности, эффективности и технологичности автоматизированных систем (АС) возможно только при условии обеспечения высокого уровня защищенности информации, которая циркулирует в этих АС. Для этого в соответствии с законами Украины об информации и ее защите, а также в соответствии с нормативными документами Системы технической защиты информации (ТЗИ) Украины необходимо применение в АС специальных средств защиты, которые предназначаются для достижения оптимального для данной АС объединения четырех свойств защищённости информации автоматизированных систем [1, 2, 3]: конфиденциальности, целостности, доступности и наблюдаемости. В зависимости от условий применения, сложности и класса АС, а также характеристик возможных угроз вес этих функциональных свойств может изменяться, но проблема обеспечения целостности является одной из основных при разработке и внедрении любых защищенных АС.

Система ТЗИ обеспечивает целостность информации, если она сохраняется, передается ли обрабатывается достоверной, полной и защищенной от неумышленных и преднамеренных искажений.