Мы надеемся, что все это обеспечит эффективное применение ДСТУ 4145-2002 в технологиях защиты информации.

Литература: ***1.*** *Miller V. S. Use of Elliptic Curves in Cryptography// Advances in Cryptology – Crypto'85. – LNCS 218. – 1986. – p. 417 – 426.* ***2.*** *N. Koblitz, "Elliptic Curve Cryptosystems", Mathematics of Computation, 48 (1987), 203-209.* ***3.*** *R. Schoof, "Elliptic Curves over Finite Fields and the Computation of Square Roots mod p", Mathematics of Computation, 44 (1985), 483–494.* ***4.*** *K. Nyberg and R. Rueppel, "A New Signature Scheme Based on the DSA Giving Message Recovery", 1st ACM Conference on Computer and Communication Security, 58–61, ACM Press, 1993.*

УДК 621.391

ПРИНЦИПЫ ПОСТРОЕНИЯ КРИПТОГРАФИЧЕСКИХ АЛГОРИТМОВ НА ЭЛЛИПТИЧЕСКИХ КРИВЫХ

Анатолий Кочубинский
МП "Дина"

Анотація: **Наведено принципи побудови криптографічних алгоритмів на еліптичних кривих, використані в стандарті ДСТУ 4145-2002, які забезпечують високу криптографічну стійкість цифрового підпису.**
Summary: **Design principles of elliptic curve cryptographic algorithms are explained. It is shown that implementation of these principles in the National digital signature standard DSTU 4145-2002 guarantees a high cryptographic strength of the digital signature.**
Ключові слова: **Захист інформації, автентифікація, цифровий підпис, скінченні поля, еліптичні криві, криптографічна стійкість, дискретне логарифмування.**

С 1 июля 2003 г. вступает в действие новый стандарт вычисления и проверки цифровой подписи ДСТУ 4145-2002. Новый стандарт имеет ряд преимуществ по сравнению с теми средствами аутентификации данных и их происхождения, которые используются сейчас в нашей стране. Стандарт ДСТУ 4145-2002 основан на новых теоретических принципах и обеспечивает очень высокую стойкость при сравнительно небольшом размере параметров цифровой подписи, включая размер ключей. Стандарт гибок в отношении выбора параметров безопасности и функции хеширования и легко адаптируется к конкретным условиям использования. Новый стандарт достаточно просто реализуется в программном и аппаратном виде. Он отличается от действующего унификацией представления входных и выходных данных при сохранении большой свободы в реализации основных алгебраических алгоритмов, составляющих стандарт. Важная особенность стандарта – нормирование датчика случайных чисел.

Основная причина создания и введения нового стандарта – невозможность с помощью существующих средств обеспечить уровень стойкости цифровой подписи, отвечающий современным требованиям. В табл. 1 приведена стойкость алгоритма ДСТУ 4145-2002 для трех значений степени m основного поля, отвечающих минимальному уровню стойкости $m = 163$, стандартному уровню стойкости с точки зрения современных требований $m = 257$ и очень высокому уровню стойкости $m = 509$. Высокая стойкость нового стандарта достигается при весьма скромных размерах параметров алгоритма, что дает возможность разрабатывать достаточно эффективные программные и аппаратные его реализации. Второй и третий столбцы этой таблицы показывают, что эквивалентные по стойкости реализации стандарта ГОСТ 34.310-95 и алгоритма RSA достаточно сложно создать для стандартного уровня стойкости и практически невозможно для высокого уровня стойкости.

Таблица 1 – Сравнительная стойкость основных алгоритмов цифровой подписи

Степень поля m	Стойкость	ГОСТ 34.310 (эквивалент)	RSA (эквивалент)
163	10^{24}	1024	3072
257	10^{38}	3250	9750
509	10^{78}	15500	46500

Мы хотим показать, что отраженное в табл. 1 соотношение не является случайным или временным, связанным, скажем, с недостаточной изученностью криптографических преобразований на эллиптических кривых, а определяется фундаментальными математическими свойствами эллиптических кривых.

Стойкость основного криптографического преобразования, используемого при вычислении цифровой подписи на эллиптической кривой, определяется сложностью решения задачи дискретного логарифмирования в циклической подгруппе $\langle P \rangle$ большого простого порядка *n* группы точек эллиптической кривой, порожденной базовой точкой эллиптической кривой *P*, т. е. сложностью решения уравнения

$$Q = kP, Q \in \langle P \rangle$$

относительно *k*, *k* – целое число, $1 < k < n$. Поясним, что сложность решения задачи, задаваемой входной последовательностью длиной *t* битов, определяется как число битовых операций $L(t)$, которые необходимо выполнить для получения решения. Если функция $L(t)$ представляет собой многочлен, то такая задача имеет полиномиальную сложность и считается простой. В качестве примеров таких задач можно привести задачу возведения целого числа в степень по модулю целого числа, задачу вычисления наибольшего общего делителя двух целых чисел, задачу доказательства простоты целого числа или задачу подсчета числа точек, лежащих на эллиптической кривой, заданной над конечным полем. Если функция $L(t)$ имеет вид $L(t) = e^{\lambda t}$, где λ — постоянная, то говорят, что задача имеет экспоненциальную сложность. Такие задачи считаются очень сложными и представляют наибольший интерес для криптографии, использующей асимметричные алгоритмы. В теории сложности рассматриваются функции $L(t)$, имеющие промежуточную скорость роста. Эти функции зависят от трех параметров и имеют вид $L(t, v, \lambda) = \exp\left(\lambda t^{v} (\log t)^{1-v}\right)$, где $0 \le v \le 1, \quad \lambda > 0$. При $v = 0$ $L(t,0,\lambda) = t^{\lambda}$ получаем полиномиальную сложность, при $v = 1$ $L(t,1,\lambda) = e^{\lambda t}$ имеем экспоненциальную сложность. Если же $0<v<1$, то эта промежуточная сложность называется субэкспоненциальной. В этом случае длина входной последовательности для задач дискретного логарифмирования – это длина двоичного представления числа n, т. е. $t = \lceil \log n \rceil$. Поэтому применительно к задаче дискретного логарифмирования экспоненциальная сложность имеет порядок роста n^{λ}, а субэкспоненциальная сложность имеет порядок $\exp\left(\lambda (\log n)^{v} (\log\log n)^{1-v}\right)$. Очевидно, что чем меньше v, тем проще в вычислительном отношении задача и, значит, практически она может быть решена для больших значений n.

В произвольной конечной циклической группе (и, следовательно, группе $\langle P \rangle$) задачу дискретного логарифмирования можно решить за $\sqrt{n}$ операций. В 1968 г. Д. Шэнкс [1] предложил метод определения порядка группы классов мнимого квадратичного поля. Этот метод затем нашел очень широкое применение в самых разных задачах теории чисел и алгебры, в частности, с помощью этого метода можно решить и задачу дискретного логарифмирования в произвольной конечной циклической группе. Это метод заключается в составлении двух списков размером $t = \left\lfloor \sqrt{n} \right\rfloor + 1$ каждый. Первый список состоит из пар $\{(i, itP),\ i = 0, \ldots, t-1\}$ и отсортирован по второй компоненте. Второй список состоит из пар вида $\{(j, Q + jP),\ j = 0, \ldots, t-1\}$ и тоже отсортирован по второй компоненте. Такая упорядоченность списков позволяет легко найти две пары с равными вторыми компонентами, т. е. пары (i, itP) и $(j, Q + jP)$, такие что $itP = Q + jP$. Тогда $itP=(k+j)P$, следовательно, $k = it - j$ по модулю n.

Приведем пример. Эллиптическая кривая

$$Y^2 + XY = X^3 + 0006$$

над полем $GF(2^{16})$ имеет порядок 65552. Возьмем подгруппу простого порядка $n = 241$, порожденную базовой точкой P= (F7A2,5B11), и решим в этой подгруппе задачу дискретного логарифмирования

$$Q = kP,\ Q = (\text{D486,A5C2}).$$

Для составления таблиц в данном случае достаточно положить $t = 16$.

Таблица 2 – Метод Д. Шэнкса

Таблица 1	Таблица 2
i = 0 *O*	j = 0 P(D486,A5C2)
i = 1 P(2E20,B3CF)	j = 1 P(BB09,E3A1)
i = 2 P(8315,DA32)	j = 2 P(3830,37E5)
i = 3 P(CEB2,9073)	j = 3 P(3300,CE2D)
i = 4 P(B051,3C7B)	j = 4 P(EBBC,83EF)
i = 5 P(6DE6,0EBC)	j = 5 P(F3CD,72E9)
i = 6 P(991C,6497)	**j = 6 P(6DE6,0EBC)**
i = 7 P(0A15,68C6)	j = 7 P(3053,7337)
i = 8 P(5DAF,B6F2)	j = 8 P(6185,EEAA)
i = 9 P(74CA,B856)	j = 9 P(6C9A,946D)
i = 10 P(3053,4364)	j = 10 P(B748,BB26)
i = 11 P(8D19,5C06)	j = 11 P(7DFB,10B8)
i = 12 P(FDBD,288C)	j = 12 P(2D7C,C910)
i = 13 P(3510,60C7)	j = 13 P(C5AD,281D)
i = 14 P(9335,7BF7)	j = 14 P(8026,86BE)
i = 15 P(F7A2,ACB3)	j = 15 P(A04A,9573)

Совпадение в двух таблицах находим при $i = 5$ и $j = 6$. Это означает, что $k = (it - j) \bmod 241$, т. е. $k = 5*16 - 6 = 74$. Таким образом, $Q = 74P$.

Для реализации этого метода требуется достаточно много памяти для хранения таблиц. Дж. Поллард [2] предложил два метода поиска аналогичного совпадения с помощью случайных блужданий, для реализации которых память не нужна. В этих методах сначала определяется случайное блуждание в группе $\langle P \rangle$. Для этого выбирается небольшой набор точек группы вида $M_i = a_i P + b_i Q, i = 1, \ldots, s$, s – параметр, зависящий от реализации (обычно, 20 – 30), a_i, b_i – случайные целые числа, и определяется отображение f группы $\langle P \rangle$ в множество целых чисел $\{1, 2, \ldots, s\}$. Если теперь взять произвольную начальную точку G_0 группы $\langle P \rangle$, то отображение $G_{i-1} \to G_i = G_{i-1} + M_{f(G_{I-1})}, i \geq 1$, определяет случайное блуждание в группе $\langle P \rangle$. Поскольку группа $\langle P \rangle$ – конечная, рано или поздно две точки этого случайного блуждания совпадут, т. е. для некоторых индексов i и l $x_i P + \bar{x}_i Q = x_l P + \bar{x}_l Q$. Отсюда сразу следует, что $(x_i - x_l)P = (\bar{x}_l - \bar{x}_i)Q$, т. е. $k = (x_i - x_l)(\bar{x}_l - \bar{x}_i)^{-1} \bmod n$, если только $\text{НОД}(\bar{x}_l - \bar{x}_i, n) = 1$. Давно известен эффективный алгоритм Флойда поиска циклов в случайных блужданиях [3], а из теории случайных блужданий известно, что среднее время до такого совпадения имеет порядок $\sqrt{n}$ (см., например, [4]). Этот метод называется ρ -методом Полларда.

Приведем пример использования этого метода. Решим задачу дискретного логарифмирования $Q=kP$ в циклической группе <*P*> из предыдущего примера, P = (F7A2,ACB3), Q = (AB67,BBBC). Пусть s = 3 и $M_0 = 2P$, $M_1 = P+Q$ и $M_2 = 3P + 4Q$. Отображения f определим следующим образом: если R – точка эллиптической кривой с координатами (x, y), то двоичное представление координаты x преобразуем в целое число и приведем его по модулю 3, полученное число примем в качестве значения $f(R)$. Возьмем в качестве начальной точки случайного блуждания точку P и построим случайное блуждание (см. табл. 3).

Таблица 3 – ρ -метод Полларда

0	1P+ 0Q = (F7A2,ACB3)
1	2P+ 1Q = (140E,4294)
2	3P+ 2Q = (3F08,1299)
3	6P+ 6Q = (E29D,21FE)
4	9P+ 10Q = (F3CD,72E9)
5	10P+ 11Q = (E1A1,AC60)

Продолжение Таблицы 3

6	13P+ 15Q = (8F14,3AC8)
7	14P 16Q = (9B2D,8F56)
8	**17P +20Q = (6185,8F2F)**
9	20P+ 24Q = (7EEA,7D9B)
10	22P+ 24Q = (331B,AF17)
11	24P+ 24Q = (BF42,E78E)
12	27P+ 28Q = (CDD1,EF16)
13	29P+ 28Q = (51D9,7C01)
14	30P+ 29Q = (B051,8C2A)
15	33P+ 33Q = (6DE6,635A)
16	**35P+ 33Q = (6185,8F2F)**

Из таблицы находим, что совпадают точки из строк 8 и 16, т. е. $17P + 20Q = 35P + 33Q$, откуда следует, что $-18P = 13Q = 13kP$ или $223 = 13k \bmod 241$, поэтому $k = 184$.

Второй метод Полларда называется λ -методом. В этом методе параллельно строятся два случайных блуждания G_i и H_i с разными начальными состояниями. Снова ищется совпадение точек, только на этот раз точки относятся к разным блужданиям. Когда совпадение найдено, решение задачи дискретного логарифмирования находится тем же способом, что и в ρ -методе. Сложность λ -метода тоже равна $\sqrt{n}$. При очень большом сходстве двух методов λ -метод обладает серьезными преимуществами над ρ -методом. Во-первых, он хорошо распараллеливается [5] в больших распределенных вычислительных системах типа Интернета, поскольку в отличие от ρ -метода не требует постоянного контакта с сервером. Во-вторых, на самом деле сложность λ -метода равна квадратному корню из длины интервала, который содержит решение задачи дискретного логарифмирования. Это означает, что если известно, что решение этой задачи не распределено равномерно во всем интервале от 1 до $n - 1$, то его можно найти существенно быстрее. Именно этот метод был использован в апреле 2000 г. для решения задачи дискретного логарифмирования в группе точек эллиптической кривой

$$y^2 + xy = x^3 + x^2 + 1$$

над полем $GF(2^{109})$, порядок которой равен 324518553658426701487448656461467 (108 бит), в рамках организованного группой французских специалистов международного проекта. Задача была решена за 4 месяца с помощью 9500 компьютеров с использованием ресурсов Интернета. Заметим, что выполненного объема вычислений хватило бы для решения 50 задач факторизации 512-битовых чисел. Для решения аналогичной задачи в поле $GF(2^{163})$ с использованием той же вычислительной техники и точно такого же алгоритма потребовалось бы примерно 40 000 000 лет. Этот пример отлично иллюстрирует разницу между алгоритмами экспоненциальной и субэкспоненциальной сложности.

Приведем пример использования этого метода решения задачи дискретного логарифмирования. Решим задачу дискретного логарифмирования $Q = kP$ в той же циклической группе. Для этого используем описанный выше способ построения случайного блуждания. На этот раз мы строим два случайных блуждания. Пусть начальное значение первого случайного блуждания есть $7P$, а начальное состояние второго блуждания есть $2Q$. Полученные случайные блуждания приведены в табл. 4.

Таблица 4 – λ -метод Полларда

0	7P+ 0Q = (9127,9A0D)	0P+ 2Q =(43CA,7345)
1	8P +1Q = (9B68,064D)	3P+ 6Q =(51D9,7C01)
2	9P +2Q = (CEB2,5EC1)	4P+ 7Q =(F885,661D)
3	11P+ 2Q = (8B6F,5B8C)	6P+ 7Q =(CDD2,5745)
4	12P +3Q = (8B6F,D0E3)	7P+ 8Q =(EBBC,83EF)
5	13P+ 4Q = (AE08,52E6)	9P+ 8Q = (3830,37E5)
6	**16P+8Q = (8319,7DB7)**	12P +12Q =(E7AA,6CB6)
7	18P +8Q = (BC7A,0F8F)	15P+ 16Q = (2D7C,E46C)
8	19P + 9Q = (D486,7144)	16P +17Q = (E4B2,1859)
9	20P +10Q = (225C,5E17)	17P +18Q = (3612,2FCB)
10	22P + 10Q = (F3D0,0934)	19P+18Q = (0A15,68C6)

Продолжение Таблицы 4

11	23P +11Q = (43B8,9EAB)	20P +9Q = (4F9E,9341)
12	26P +15Q = (B56D,3CF2)	22P +9Q = (257E,9D45)
13	29P+ 19Q = (297B,A073)	**23P +20Q = (8319,7DB7)**

Из таблицы находим, что совпадают точка 6 из левого столбца и точка 13 из правого столбца, т. е. $16P+8Q = 23P + 20Q$, откуда следует, что $-7P = 12\ Q = 12kP$ или $234 = 12k \mod 241$. Поэтому $k = 140$ есть решение задачи дискретного логарифмирования.

Описанная группа методов дискретного логарифмирования имеет экспоненциальную сложность с параметром $\lambda = 0.5$. Сложность этих методов определяет стойкость любых криптографических преобразований, определенных в любой конечной циклической группе, в том числе, и в группе точек эллиптической кривой над конечным полем. Таким образом, величина $\sqrt{n}$ дает верхнюю оценку стойкости всех таких преобразований. Единственный способ противостоять этим методам состоит в увеличении размера циклической группы n. По современным представлениям минимальный допустимый уровень стойкости составляет 10^{24}. Поэтому в стандарте ДСТУ 4145-2002 и определено минимальное допустимое значение степени расширения основного поля $m = 163$.

Хорошо известно, что для многих конечных циклических групп существуют более быстрые алгоритмы решения задачи дискретного логарифмирования, имеющие субэкспоненцальную сложность. Покажем на примере, как строятся алгоритмы этого типа. Решим задачу дискретного логарифмирования $347280 \equiv 3^x \mod p$, $p = 835253$ – простое число, 3 – примитивный корень из 1 по модулю этого простого числа. Выберем множество небольших простых чисел {2, 3, 5, 7, 11, 13, 17, 19}. Произведение этих чисел равно $9699690 > p$, поэтому можно ожидать, что случайное число, не превышающее p, с достаточно большой вероятностью разложится на простые множители из этого набора. Этот набор небольших простых чисел называется базой разложения. Первый шаг метода состоит в вычислении дискретных логарифмов чисел из базы разложения. Для этого мы возводим число 3 в случайную степень, не превышающую $p - 1$, и пытаемся разложить полученное число на простые множители из базы разложения. После некоторого числа попыток получаем такие соотношения:

$3^{356951} = 91 = 7 \cdot 13;$
$3^{386299} = 85 = 5 \cdot 17;$
$3^{357490} = 24 = 2^3 \cdot 3;$
$3^{426086} = 99 = 3^2 \cdot 11;$
$3^{54684} = 15 = 3 \cdot 5;$
$3^{69942} = 68 = 2^2 \cdot 17;$
$3^{162209} = 26 = 2 \cdot 13.$

Если теперь вычислить дискретный логарифм от обеих частей полученных соотношений, то мы получим уравнения вида

$356951 = \log(7)+\log(13) \mod p - 1;$
$386299 = \log(5)+\log(17) \mod p - 1;$
$357490 = 3\log(2)+1 \mod p - 1;$
$426086 = 2+\log(11) \mod p - 1;$
$54684 = 1+\log(5) \mod p - 1;$
$569942 = 2\log(2)+\log(17) \mod p - 1;$
$162909 = \log(2)+\log(13) \mod p - 1.$

Таким образом, получено 7 независимых уравнений, которые связывают 7 неизвестных дискретных логарифмов чисел из базы разложения (логарифм $\log(3) = 1$ известен). Из этой системы линейных уравнений находим: $\log(2) = 119163$; $\log(3) = 1$; $\log(5) = 54683$; $\log(7) = 313205$; $\log(11) = 426084$; $\log(13) = 43746$; $\log(17) = 331616$; $\log(19) = 749481$. После выполнения этой предварительной работы можно решить любую задачу дискретного логарифмирования $y = 3^x \mod p$ по данному модулю p следующим образом. Вычисляем выражение $y3^k \mod p$ для случайных $k<p$ до тех пор, пока полученное число не разложится по выбранной базе разложения. Как только это случится, легко вычислить дискретный логарифм x. В нашем примере довольно быстро находим, что $347280 \cdot 3^{118732} = 5^4 \mod p$, $x+118732 = 4\log(5) = 4 \cdot 54683 = 218732 \mod p - 1$, поэтому $x = 218732 - 118732 = 100000$.

Анализ этого алгоритма в общем случае показывает, что он имеет субэкспоненциальную сложность с параметром $\nu = 0{,}5$. Успех применения этого метода основан на том, что существует много небольших простых

чисел и поэтому достаточно велика вероятность того, что выбранное наугад целое число полностью разложится на простые множители из сравнительно небольшой базы разложения. Вторая особенность целых чисел, способствующая успеху метода, – линейный рост размера чисел при выполнении операций над ними, например, квадрат числа в два раза длиннее исходного числа. Сделаем еще одно важное замечание. Мы привыкли отождествлять элементы простого конечного поля $GF(n)$ с целыми числами. На самом деле в этом конечном поле нет понятия простого числа и поэтому нет разложения элементов конечного поля на простые числа. В описанном выше алгоритме мы фактически сначала переходим в кольцо целых чисел, где уже есть простые числа и понятие разложения на простые множители, выполняем в этом кольце разложение на множители, а затем возвращаемся в исходное конечное поле, где и решаем систему линейных уравнений. Оказывается, что эта конструкция обобщается на другие циклические группы. Всякий раз, когда исходную циклическую группу можно отобразить в алгебраическую структуру, в которой есть понятие простого элемента и понятие разложения произвольного элемента на простые элементы, можно надеяться на существование субэкспоненциального алгоритма. Для этого надо, чтобы в этой алгебраической структуре было много небольших в том или ином смысле простых элементов и чтобы размер элементов медленно увеличивался при выполнении операций над ними. Перечислим несколько групп, для которых описанный подход дает алгоритм субэкспоненциальной сложности.

Мультипликативная группа простого конечного поля отображается в кольцо целых чисел, содержащее большое число простых чисел, размер простого числа – его абсолютная величина.

Мультипликативная группа конечного поля характеристики 2 отображается в кольцо многочленов над полем $GF(2)$, содержащее большое число неприводимых многочленов, размер неприводимого многочлена – его степень.

Группа классов мнимого квадратичного поля отображается в группу бинарных квадратичных форм, содержащую большое число простых бинарных квадратичных форм, размер простого элемента – норма соответствующей бинарной квадратичной формы.

Якобиан гиперэллиптической кривой большого рода отображается в группу дивизоров, содержащую большое число простых дивизоров, размер дивизора – его степень.

Во всех этих случаях существует алгоритм дискретного логарифмирования субэкспоненциальной сложности с параметром $v = 0{,}5$. Для простого конечного поля привлечение дополнительных соотношений из специально подобранного поля алгебраических чисел позволяет уменьшить значение параметра v до $1/3$. Этот алгоритм называется алгоритмом решета в числовых полях. Он был создан Дж. Поллардом [6] для решения задачи разложения целых чисел на простые множители и затем был перенесен на задачу дискретного логарифмирования [7]. Оценки из второго и третьего столбцов табл. 1 отвечают именно этому методу. Алгоритм дискретного логарифмирования в поле характеристики 2 с таким же значением параметра v был создан Д. Копперсмитом еще раньше [8]. Субэкспоненциальный алгоритм со значением параметра $v = 1/2$ для якобианов гиперэллиптических кривых большого рода описан в [9].

В приведенном выше списке нет эллиптических кривых и это не случайно. Формально описанный выше метод можно применить к эллиптическим кривым. Один из возможных подходов состоит в отображении исходной циклической группы в группу точек эллиптической кривой, определенной над полем рациональных чисел. По теореме Морделла – Вейля эта группа точек является конечно порожденной, т. е. эта группа либо конечная, либо порождена конечным числом точек. Это число называется рангом эллиптической кривой. Таким образом, в случае эллиптических кривых ситуация резко изменяется, у нас может быть только конечное число аналогов простых элементов, причем весьма небольшое, до сих пор известны эллиптические кривые с максимальным рангом 23. Если эта группа окажется конечной, то по теореме Мазура ее максимальный простой порядок равен 7, т. е. в этом случае метод вообще неприменим. Если эта группа бесконечная, то в подавляющем числе случаев она порождена одной или двумя точками. Пусть эта группа порождена одной точкой $\widetilde{R}$. Тогда исходная задача дискретного логарифмирования сводится к задаче дискретного логарифмирования на эллиптической кривой над полем рациональных чисел

$$\widetilde{Q} = k\widetilde{P},\ \widetilde{Q} = m\widetilde{R},\ \widetilde{P} = l\widetilde{R},\ km\widetilde{R} = l\widetilde{R},$$

т. е. должно выполняться соотношение

$$km - l \equiv 0 \bmod n.$$

Вторая особенность эллиптической кривой над полем рациональных чисел состоит в том, что при выполнении операций размер точек быстро растет, например, точка $m\widetilde{R}$ в m^2 раз длиннее точки $\widetilde{R}$. В качестве меры длины точек эллиптических кривых над полем рациональных чисел используется высота Нерона-Тейта [10]. Поэтому сложность выполнения операций над точками вносит существенный вклад в общую сложность алгоритма и для успеха алгоритма нужно, чтобы параметры m и l были малыми. Вероятность того, что

соотношение $km - l \equiv 0 \bmod n$ будет выполнено при малых *m* и *l* имеет порядок $1/n$, а общая сложность алгоритма равна *n*. В общем случае было исследовано много вариантов отображения циклической группы эллиптической кривой над конечным полем в разные алгебраические и геометрические структуры [11–13, 17, 19], но все эти попытки только подтвердили эту оценку. Основное препятствие для создания субэкспоненциальных алгоритмов для эллиптических кривых – конечное число образующих в группе точек эллиптической кривой над полем рациональных чисел и быстрый рост размера точек при выполнении операций в таких группах. Этот факт Н. Коблиц [17] назвал золотым щитом, оберегающим эллиптическую криптографию.

Хотя в самой группе точек эллиптической кривой нет субэкспоненциальных алгоритмов дискретного логарифмирования и маловероятно их появление в будущем, всегда есть возможность сведения исходной задачи дискретного логарифмирования к аналогичной задаче в других группах, где субэкспоненциальные алгоритмы существуют. При определенных условиях это дает возможность получить субэкспоненциальный алгоритм для исходной задачи.

Первое такое сведение предложено в 1963 г. А. Менезес, Т. Окамото и С. Вэнстон [14]. С помощью спаривания Вейля они свели исходную задачу над полем $GF\left(2^m\right)$ к задаче дискретного логарифмирования в мультипликативной группе некоторого расширения $GF\left(2^{km}\right)$ исходного поля. Делается это следующим образом.

Точки эллиптической кривой порядка *n* образуют подгруппу $E[n]$, которая называется группой кручения эллиптической кривой. Если *n* – нечетное простое число, то порядок группы кручения равен n^2. Криптографические алгоритмы строятся в циклической группе простого порядка *n*, эта циклическая группа является подгруппой группы кручения. Остальные точки группы кручения не принадлежат основному полю. Поскольку группа кручения – конечная, то существует расширение $GF\left(2^{ml}\right)$ исходного поля, в котором лежат все точки группы кручения. Мультипликативная группа этого расширения имеет порядок $2^{ml} - 1$, поэтому вложение группы кручения в мультипликативную группу этого расширения возможно только в том случае, если *n* делит $2^{ml} - 1$. Оказывается, что это условие и достаточно [16]. Этот факт лежит в основе метода [14] сведения задачи дискретного логарифмирования в группе точек эллиптической кривой к задаче дискретного логарифмирования в мультипликативной группе расширения исходного конечного поля, в которой, как известно, существуют алгоритмы субэкспоненциальной сложности. Фактически это сведение выполняется с помощью спаривания Вейля. Спаривание Вейля – это отображение вида

$$w: E[n] \times E[n] \to \mu_n,$$

где μ_n – группа корней степени *n* из единицы, которая при приведенном выше условии является подгруппой мультипликативной группы конечного поля $GF\left(2^{ml}\right)$. Если $Q = kP$ и *T* – точка из $E[n]$, не принадлежащая <*P*>, и если $w(P,T) = \alpha \in \mu_n$, $w(Q,T) = \beta \in \mu_n$, то $\beta = \alpha^k$. Таким образом, задача дискретного логарифмирования $Q = kP$ в циклической группе <*P*> сводится к задаче дискретного логарифмирования $\beta = \alpha^k$ в конечном поле $GF\left(2^{ml}\right)$.

Если степень расширения *l* мала, то для исходной задачи дискретного логарифмирования существует субэкспоненциальный алгоритм. Например, в случае крайне привлекательных с вычислительной точки зрения суперсингулярных кривых $l \le 6$, поэтому пришлось отказаться от применения таких кривых в криптографии. В стандарте разрешены эллиптические кривые, порядок базовых точек которых удовлетворяет условию $2^{lm} \ne 1 \bmod n$ для $1 \le l \le 32$. Это условие (условие Менезеса-Окамото-Вэнстона) легко проверяется и гарантирует экспоненциальную сложность решения задачи дискретного логарифмирования. Это условие можно назвать локальным в том смысле, что над любым полем существует много кривых, на которых задача дискретного логарифмирования не сводится к субэкспоненциальному случаю и проверка несводимости проводится индивидуально для каждой конкретной кривой. Аналогичное сведение выполняется и с помощью спаривания Тейта [15]. Условие несводимости остается прежним.

Приведем пример. Снова возьмем эллиптическую кривую $Y^2 + XY = X^3 + 0006$ над полем GF(2^{16}), ее порядок равен 65552, и возьмем циклическую подгруппу простого порядка $n = 241$. Если эту кривую рассматривать над полем *GF*(2^{48}), то она будет иметь порядок 281474973764912. Подгруппа точек порядка $n = 241$ имеет порядок $241^2 = 58081$. Легко проверить, что $2^{48} - 1$ делится на 241, а $2^{16} - 1$ не делится на 241, поэтому циклическая подгруппа <*P*> порядка 241 вкладывается в мультипликативную группу конечного поля

$GF(2^{48})$. Решим задачу дискретного логарифмирования $Q = kP$, P = (F7A2,5B11), Q = (EBBC,6853). Точка T = (4EBABC2CB649,6A1C71974D82) имеет порядок 241 и не лежит в циклической группе <P>. Вычисление спаривания Вейля дает $w(P,T)$ = 02403A2472C5, $w(Q,T)$ = 60CB16613A0E. Таким образом, исходная задача сводится к задаче дискретного логарифмирования

$$60CB16613A0E = 02403A2472C5^k$$

в поле $GF(2^{48})$; ее решение k = 163 дает решение исходной задачи.

В 2000 г. было показано [18], что задача дискретного логарифмирования в группе точек эллиптической кривой, заданной над конечным полем $GF(2^m)$, с помощью так называемого спуска Вейля сводится к задаче дискретного логарифмирования в якобиане некоторой гиперэллиптической кривой рода $g = 2^{k-1}$ или рода $2^{k-1} - 1$, заданной над тем же конечным полем или его подполем, если степень расширения поля – составное число. Параметр k – некоторое число, зависящее от степени расширения основного поля и свободного члена уравнения эллиптической кривой. Если степень расширения поля m – составное число, то k мало. Если, кроме того, $k \neq 1$, то алгоритм дискретного логарифмирования [21] в якобиане гиперэллиптической кривой, род которой $g > 1$ отвечает этому значению k, оказывается при g > 3 эффективнее по сравнению с описанными выше алгоритмами экспоненциальной сложности. Если m – простое число, то для всех эллиптических кривых над полем $GF(2^m)$ параметр k больше порядка числа 2 по модулю m. Для всех простых чисел из диапазона (163, 600) $k \geq 17$, т. е. $g > 65535$. Поскольку алгоритм дискретного логарифмирования [17] субэкспоненциален по 2^{mg}, то при таких больших значениях рода g его сложность превышает сложность экспоненциальных алгоритмов, описанных выше. Следовательно, в этом случае субэкспоненциальное сведение становится невозможным. Если $k = 1$, то род $g = 2^{k-1} - 1$ и гиперэллиптическая кривая на самом деле является эллиптической, т. е. в этом случае субэкспоненциального алгоритма не существует. Этому условию удовлетворяют так называемые эллиптические кривые Коблица, т. е. эллиптические кривые вида $y^2 + xy = x^3 + 1$ или $y^2 + xy = x^3 + x^2 + 1$. Таких кривых в диапазоне основных полей, разрешенных стандартом, всего 11. Следовательно, все поля, степень которых – составное число, не пригодны для криптографических применений. Поэтому стандартом ДСТУ 4145-2002 разрешены только конечные поля, степень которых – простое число.

Таким образом, криптографическое преобразование, использованное в стандарте ДСТУ 4145-2002, обладает следующими свойствами, обеспечивающими его высокую стойкость.

Порядок n циклической группы эллиптической кривой – простое число – для исключения применения метода Полига-Хеллмана [20].

Порядок n циклической группы эллиптической кривой больше 2^{163}, что исключает применение методов экспоненциальной сложности с учетом возможности их распараллеливания.

Порядок n циклической группы эллиптической кривой удовлетворяет условию Менезеса-Окамото-Вэнстона для исключения сведения задачи дискретного логарифмирования в этой циклической группе к задаче дискретного логарифмирования в мультипликативной группе расширения исходного поля степени менее 32.

Поле определения эллиптической кривой (основное поле) имеет простую степень – для исключения сведения задачи дискретного логарифмирования в циклической группе этой кривой к задаче дискретного логарифмирования в якобиане некоторой гиперэллиптической кривой.

Высокая стойкость основного криптографического преобразования, используемого в стандарте, является необходимым условием стойкости цифровой подписи в целом, однако практическая стойкость цифровой подписи зависит от целого ряда факторов, которые только частично учтены в стандарте. В процессе вычисления цифровой подписи обязательно используются две криптографические функции, стойкость которых столь же важна, как и стойкость основного криптографического преобразования. Это функция хеширования и датчик случайных последовательностей.

Функция хеширования преобразует исходное сообщение в блок данных стандартного размера L_H. Именно к этому блоку данных применяется основное криптографическое преобразование цифровой подписи. Стойкость даже идеально спроектированной функции хеширования не превышает $2^{L_H/2}$, поэтому если функция хеширования с данным параметром L_H используется совместно с алгоритмом цифровой подписи ДСТУ 4145-2002, то должно выполняться неравенство $L_H \geq m$, где m – степень используемого основного поля. Стандарт разрешает использовать только нормированные функции хеширования или функции хеширования, рекомендованные в установленном порядке. В настоящее время единственная такая функция хеширования – это функция ГОСТ 34.311-95. Поскольку у этой функции хеширования $L_H = 256$, то при вычислении цифровой подписи нет смысла применять поля с $m > 257$.

Стойкость цифровой подписи существенно зависит от качества датчика случайных последовательностей. Выше уже говорилось, что если можно угадать интервал, в котором находится секретный разовый параметр

цифровой подписи, то сложность решения соответствующей задачи дискретного логарифмирования может резко уменьшиться. Известно также, что если можно угадать последний байт или два байта секретного разового параметра, то по сравнительно небольшому набору известных подписей (30–40 подписей) можно определить личный ключ цифровой подписи с помощью известного алгоритма L^3. Поэтому стандарт разрешает использовать только нормированные датчики случайных последовательностей или датчики, рекомендованные в установленном порядке. Поскольку пока нет стандарта, определяющего датчик случайных последовательностей, то следует использовать алгоритм формирования случайных последовательностей, описанный в приложении А стандарта.

Стойкость цифровой подписи гарантируется только в том, если все параметры алгоритма подписи и сама подпись вычислены в строгом соответствии стандарту. Поэтому правильность всех вычислений должна контролироваться явно или косвенно. Стандарт устанавливает правила проверки почти всех вычислений. Единственный случай, когда явная проверка невозможна – это вычисление предподписи. Поэтому в этом случае используется косвенная проверка, а именно, при нарушении условия, установленного стандартом, вычисленная подпись будет недействительной. Заметим, что стандарт ГОСТ 34.310-95 этим свойством не обладает.

В реальных условиях применения любой алгоритм вычисления и проверки цифровой подписи подвержен многочисленным угрозам, связанным с манипуляциями открытыми и личными ключами цифровой подписи. Противостоять этим угрозам алгоритмическими методами почти невозможно. Поэтому практическое использование цифровой подписи должно поддерживаться юридически и организационно. Важнейшими компонентами такой поддержки являются сертификация открытых ключей цифровой подписи с обязательной идентификацией их владельцев и использование только сертифицированных программных и аппаратных средств вычисления и проверки цифровой подписи, функционирующих в контролируемой операционной среде.

Литература: ***1.*** *D. Shanks. Class number, a theory of factorization et genera. Proc. Symp. In Pure Math., 20, AMS, Providence, RI, 1969, p. 415–440.* ***2.*** *J. M. Pollard. Monte Carlo Methods for Index Computations (mod p). Math. Comp., 32, 1978, p. 918–924.* ***3.*** *Д. Кнут. Искусство программирования для ЭВМ. Т. 2. Получисленные алгоритмы. М., Мир, 1977, с. 724.* ***4.*** *В. Ф. Колчин, Б. А. Севастьянов, В. П. Чистяков. Случайные размещения. М., Наука, 1976, с. 224.* ***5.*** *P. C. van Oorschot, M. J. Wiener. Parallel Collision Search with Cryptanalytic Applications. J. Crypto., 12, 1999, p. 1–28.* ***6.*** *A. K .Lenstra and H. W. Lenstra, "The Development of the Number Field Sieve", Springer, 1993.* ***7.*** *D. M. Gordon, "Discrete Logarithms in GF (p) Using the Number Field Sieve", SIAM J. Comput., 6 (1993), 124–138.* ***8.*** *D. Coppersmith, "Fast Evaluation of Logarithms in Finite Fields of Characteristic 2", IEEE Transactions on Information Theory, 30 (1984), 587–594.* ***9.*** *L. M. Adleman, J. DeMarrais, M.D. Huang. A Subexponential Algorithm for Discrete Logarithms over the Rational Subgroup of the Jacobians over Finite Fields. ANTS-1 (LNCS 877), Springer, 1994, p. 28–40.* ***10.*** *.Silverman J. The Arithmetic of Elliptic Curves. –New York: Springer, 1986. – p. 400.* ***11.*** *Silverman J., Suzuki. Elliptic curve discrete logarithms and the index calculus. Advances in Cryptology, Asiacrypt 98, (LNCS 1514), Springer 1998, p.110–125.* ***12.*** *Silverman J. The xedni-calculus and the elliptic curve discrete logarithm problem. Preprint 1998.* ***13.*** *Jacobson M., Koblitz N., Silverman J., Stein A., Teske. Analysis of the xedni-calculus attack. Preprint 1999.* ***14.*** *A. Menezes, T. Okamoto and S. Vanstone, "Reducing Elliptic Curve Logarithms to a Finite Field", IEEE Trans. Info. Theory, 39 (1993), 1639–1646. 15. G. Frey, H. G. Rűck. A Remark Concerning m-Divisibility and the Discrete Logarithm Problem in the Divisor Class Group of Curves. Math. Comp., 62, 1994, p. 865–874.* ***16.*** *R. Balasubramanian and N. Koblitz, "The Improbability that an Elliptic Curve has Subexponential Discrete Log Problem under the Menezes-Okamoto-Vanstone Algorithm", J. of Cryptology, 11 (1998), 2, 141–145.* ***17.*** *Koblitz N. Miracles of the height function – a golden shield protecting ECC. ECC-2000, Essen* ***18.*** *Gaudry P., Hess F., Smart N. Constructive and destructive facets of Weil descent on elliptic curves. Preprint, 2000.* ***19.*** *Miller V. S. Use of Elliptic Curves in Cryptography// Advances in Cryptology – Crypto'85. – LNCS 218. – 1986. – p. 417 – 426.* ***20.*** *G. C. Pohlig, M. Hellman. An Improved Algorithm for Computing Logarithms over GF(p) and its Cryptographic Significance. IEEE Trans. Info. Theory, 24, 1978, p. 106–110.* ***21.*** *P. Gaudry. An Algorithm for Solving the Discrete Log Problem on Hyperelliptic Curves. Eurocrypt 2000 (LNCS 1807), Springer, 2000, p. 19–34.*