УДК 378.6:519.83

Студ. К.С. Марчук
Науч. рук. доц. И.К. Асмыкович
(кафедра высшей математики)

ИСПОЛЬЗОВАНИЕ ТЕОРИИ ГРУПП ТОЧЕК НА ЭЛЛИПТИЧЕСКОЙ КРИВОЙ ДЛЯ СОЗДАНИЯ ЭЛЕКТРОННОЙ ПОДПИСИ

Долгое время люди копили информацию. Сначала это были рисунки на стенах, после – печатный или письменный текст, а в наше время мы имеем огромное количество информации в электронном виде. Ни для кого не секрет, что большая часть информации имеет ценность, особенно для юридических лиц. В нашем примере, информация нуждается в проверке принадлежности – подписи.

ECDSA (Elliptic Curve Digital Signature Algorithm) – криптографический алгоритм с открытым ключом для создания цифровой подписи, определённый в группе точек эллиптической кривой [1]. Подпись создается секретно, но может быть публично проверена. Это означает, что только один субъект может создать подпись сообщения, но любой может проверить её корректность.

Цель работы: на конкретных примерах подробно рассмотреть алгоритм создания электронной подписи и способов ее публичной проверки.

До того, как ECC стала популярной, почти все алгоритмы с открытым ключом основывались на RSA, DSA и DH [2], альтернативных криптосистемах на основе модулярной арифметики. Они попрежнему популярны, и часто используются вместе с ECC. Однако основы ECC всё ещё являются для большинства людей загадкой.

Секретные ключи и открытые ключи. К примеру, в Биткойне [3] используется вариант эллиптической криптографии **secp256k1**. Покажем, откуда берутся секретные и открытые ключи и как они связаны друг с другом. В ECDSA секретный ключ – это случайное целое число между 1 и значением порядка – количеством элементов группы. Открытый же ключ получается из секретного при помощи операции скалярного умножения базовой точки на значение секретного ключа. В виде уравнения:

Открытый ключ = секретный ключ * базовая точка.

Это показывает, что максимально возможное количество секретных ключей - **конечно**, и равно порядку. Так может, они когда-то закончатся? Вряд ли, потому что порядок – это **действительно большое** число.

Вычисление открытого ключа разбивается на ряд операций удвоения и сложения точек, начиная с базовой точки [1]. Сложение точек p + q определяется покомпонентно следующим образом:

$$c = (q_y - p_y) / (q_x - p_x), \quad r_x = c^2 - p_x - q_x, \quad r_y = c\,(p_x - r_x) - p_y.$$

А операция «удвоения» точки p выглядит следующим образом:

$$c = (3p_x^2 + a) / 2p_y, \quad r_x = c^2 - 2p_x, r_y = c\,(p_x - r_x) - p_y.$$

Объем вычислений на реальном примере был бы невероятно сложным, но мы можем попробовать на примере с небольшими числами, чтобы увидеть, как это работает.

Итак, возьмем уравнение кривой Биткойна, будем использовать маленькие числа. Уравнение кривой: $y^2 = x^3 + 7$. Модуль: 67. Базовая точка: (16, 4). Порядок: 79.

Требуется выбрать секретный ключ из 79 доступных. Возьмём 3. Найдём к нему публичный ключ. Нам для этого потребуется операция удвоения и сложения:

$$c = (3 * 16^2 + 0) / (2 * 4) \bmod 67 \rightarrow c = 768 / 8 \bmod 67 \rightarrow 31 / 8 \bmod 67.$$

Но как нам выполнить операцию деления в контексте конечного поля, где результат должен быть целочисленным? Для этого мы должны умножить 31 на величину, обратную к 8. Это должно быть такое число, при умножении на которое, мы получаем остаток 1 по заданному модулю, т. е. $8^{-1} = 42$.

Далее все довольно просто:

$$c = 31 * 42 \bmod 67 = 29, r_x = (29^2 – 2 * 16) \bmod 67 = 809 \bmod 67 = 5,$$

$$r_y = (29 * (16 – 5) – 4) \bmod 67 = 315 \bmod 67 = 47.$$

Далее сложим эту точку с базовой p = (16, 4), q = (5, 47):

$$c = (47 – 4)/(5 – 16) \bmod 67 = 43/56 \bmod 67 = 43 * 6 \bmod 67 = 57,$$

$$r_x = 3249 – 16 – 5 \bmod 67 = 12,$$

$$r_y = 57 * (16 – 12) – 4 \bmod 67 = 224 \bmod 67 = 23.$$

Итак, мы определили, что для секретного ключа 3 публичным ключом будет **точка** (12, 23).

Получить публичный ключ, имея секретный в разы проще, чем получить секретный, имея публичный. Хотя получить секретный ключ из публичного **теоретически** и возможно, это является вычислительно невозможным из-за огромных чисел-параметров, используемых в эллиптической криптографии.

Как и секретный ключ, публичный ключ тоже представляется в виде шестнадцатеричного числа. Однако только что мы получили точку, а не число. Так вот, публичный ключ – это просто записанные вместе два 256-битных числа, являющиеся его x и y координатами.

Подпись данных секретным ключом.Теперь у нас есть секретный и публичный ключ, которые мы можем использовать для подписи данных. Сами данные могут иметь любую длину. Для начала данные хэшируются, чтобы получить уникальное число, содержащее такое же количество битов (256), как и порядок кривой. Упрощая нашу задачу, мы пропустим шаг хеширования и подпишем данные z. Обозначим через G базовую точку, через n – порядок, а d – закрытый ключ. Алгоритм создания подписи выглядит следующим образом:

1. Выбрать некоторое целое k в пределах от 1 до n-1.
2. Рассчитать точку $(x, y) = k * G$, используя скалярное умножение.
3. Найти $r = x \bmod n$. Если $r = 0$, вернуться к шагу 1.
4. Найти $s = (z + r * D) / k \bmod n$. Если $S = 0$, вернуться к шагу 1.
5. Пара (r, s) является искомой подписью.

Напомним, вместо деления числителя на знаменатель, мы числитель умножаем на обратную знаменателю величину. На шаге 1 важно, чтобы k не повторялось в разных подписях, и чтобы его не могла угадать третья сторона [2]. То есть k должен быть либо случайным, либо создан детерминированным процессом, который хранится в тайне. Иначе третья сторона получает возможность найти секретный ключ, начиная с шага 4, так как s, z, r, k и n всем известны.

Давайте выберем в качестве наших данных число 21 и подпишем его секретным ключом 2.

$$z = 17 \text{ (данные)}, n = 79 \text{ (порядок)}, G = (16, 4) \text{ (базовая точка)},$$
$$d = 3 \text{ (секретный ключ)}.$$

1. Выберем случайное число:

$$k = rand(1, n - 1) \rightarrow \quad k = rand(1, 79 - 1) \rightarrow k = 2.$$

2. Рассчитаем точку. Это делается таким же образом, как ранее при вычислении публичного ключа – для краткости опустим подробную арифметику сложения и удвоения.

$$(x, y) = 2G = (16, 4) + (16, 4) = (5, 47) \rightarrow x = 5, y = 47.$$

3. Находим *r*:$r = x mod n \rightarrow r = 5 \bmod 79 = 5$.

4. Находим *s*:

$$s = (z + r * d) / k \bmod n \rightarrow s = (21 + 5 * 3) / 2 \bmod 79 = 36 / 2 \bmod 79 = 18 \bmod 79 = 18.$$

Обратите внимание, что выше мы смогли разделить на 2, так как результат был целым числом. Если искать обратную величину, результат будет таким же 36 * 40 mod 79 = 1440 mod 79 = 18.

5. Теперь наша подпись – это пара $(r, s) = (5, 18)$. Как и секретные и публичные ключи, подпись обычно представляется в виде шестнадцатеричной строки.

Проверка подписи публичным ключом. Теперь у нас есть данные и подпись. Третья сторона, которая знает наш публичный ключ, может получить наши данные и подпись, и убедиться, что именно мы являемся отправителями. Обозначив наш открытый ключ Q, шаги для проверки подписи будут следующими:

1. Убедиться, что r и s находятся в диапазоне от 1 до n-1.
2. Рассчитать $w = s^{-1} \bmod n$.
3. Рассчитать $u = (z * w) \bmod n$.
4. Рассчитать $v = (r * w) \bmod n$.
5. Рассчитать точку $(x, y) = uG + vQ$.
6. Убедиться, что $r = x \bmod n$. Если это не так, то подпись недействительна.

На сторонних ресурсах можно найти довольно большое доказательство того, что данный алгоритм проверки работает. Но мы просто прогоним алгоритм самостоятельно и покажем, что он работает. Напомним, наши переменные:

$$z = 21 \text{ (данные)}, (r, s) = (5, 18) \text{ (подпись)}, n = 79 \text{ (порядок)},$$
$$G = (16, 4) \text{ (базовая точка)}, Q = (12, 23) \text{ (публичный ключ)}.$$

1. Убедимся, что r и s находятся в диапазоне от 1 до n-1.

$$r: 1 <= 5 < 79, \qquad s: 1 <= 18 < 79.$$

2. Рассчитаем w:

$$w = s^{-1} \bmod n \rightarrow w = 18^{-1} \bmod 79 = 22.$$

3. Рассчитаем u:

$$u = (z * w) \bmod n \rightarrow u = (21 * 22) \bmod 79 = 462 \bmod 79 = 67.$$

4. Рассчитаем v:

$$v = (r * w) \bmod n \rightarrow v = (5 * 22) \bmod 79 = 110 \bmod 79 = 31.$$

5. Рассчитаем точку (x, y):

$$(x, y) = uG + vQ.$$

Разберем операции удвоения и сложения в uG и vQ отдельно.

uG = 67G = 2(2(16G)+G)+G = 2(2(2(2(2(2G))))+G)+G

Таким разложением мы заменяем 67 сложений точек шестью удвоениями и двумя сложениями. Данная группировка делает наши вычисления намного проще.

uG = 2(2(2(2(2((5,47)))))+G)+G = 2(2(2(2(49,2)))+G)+G =
2(2(2(58,22))+G)+G = 2(2(34,7)+G)+G = 2(2(34,7)+G)+G =
2((55,17)+G)+G = 2(11,20)+G = (14,65)+(16,4) = (46,40).

И теперь, все то же для *vQ*:

vQ = 31Q = 2(2(2(2Q+Q)+Q)+Q)+Q = 2(2(2((30,41) +Q)+Q)+Q)+Q =
2(2(2(26,30)+Q)+Q)+Q = 2(2((64,39)+ Q)+Q)+Q = 2(2(24,30)+Q)+Q =
2((23,39)+Q)+Q = 2(2,22)+Q = (52,7)+(12,23) = (38,41)

Посчитав, складываем их вместе:

$$(x, y) = uG + vQ \rightarrow (x, y) = (46, 40) + (38, 41) = (5, 47).$$

Сразу видно, что больше всего работы приходится на 5 шаг.

6. Наконец, убедимся, что r = x mod n

$$r = x \bmod n \rightarrow 5 \bmod 79 = 5.$$

Таким образом, мы рассмотрели достаточно новый, сложный и надёжный алгоритм для создания и проверки цифровой подписи, ныне используемый повсеместно.

ЛИТЕРАТУРА

1. Острик В. В., Цфасман М. А. Алгебраическая геометрия и теория чисел: рациональные и эллиптические кривые. – М.: МЦНМО, 2001. – С. 48.

2. Доступно о криптографии на эллиптических кривых [Электронный ресурс]. – Режим доступа: https://habr.com/ru/post/335906/

3. Википедия, свободная энциклопедия [Электронный ресурс]. – Режим доступа: https://www.wikipedia.org

4. Биткойн [Электронный ресурс]. – Режим доступа: https://ru.wikipedia.org/wiki/Биткойн

5. «Что такое биткойн?» [Электронный ресурс]. – Режим доступа: https://coinspot.io/beginners/chto-takoe-bitcoin/