УДК

ВВЕДЕНИЕ В ЭЛЛИПТИЧЕСКУЮ КРИПТОГРАФИЮ

Патюпин М.С.[1], студент гр.250505

Белорусский государственный университет информатики и радиоэлектроники[1]
г. Минск, Республика Беларусь

Смирнова И.А. – ассистент кафедры ВМ

Аннотация. Математические свойства эллиптических кривых, алгоритм Диффи-Хеллмана его описание, числовая и программная реализация. Принцип работы алгоритма ECDSA и подбор параметров эллиптической кривой.

Ключевые слова. Алгоритм ECDSA, алгоритм Диффи-Хеллмана, эллиптические кривые, эллиптическая криптография.

Оглавление:

1. Введение

Эллиптические кривые в криптосистемах предложили использовать Нил Коблиц и Виктор Миллер еще в 1985 году, сейчас мы можем наблюдать их использование в электронной подписи Bitcoin, в сетевых протоколах SSH и TLS, в электронной подписи(Citisen Card) граждан некоторых стран (Австрия). В Беларуси был принят стандарт для решения задач связанных с цифровой подопью на основе эллиптических кривых в 2013 году[1].

1.2 Основные плюсы и минусы эллиптической криптографии

Основные плюсы эллиптической криптографии:

- Более высокая стойкость при равной трудоемкости по сравнению с обычными криптосистемами[2].
- Меньший размер ключа чем в асимметричной криптографии. Криптостойкость достигаемая в алгоритме алгоритме RSA с использованием ключа в 3072-байт, на эллиптических кривых используется с размером ключа в 256 байт[2].
- Возможность использоваться в устройствах с ограниченными вычислительными ресурсами[3].
- Сложность атак: атаки на системы, защищенные эллиптической криптографией, требуют значительного объема вычислений и времени.

Основные минусы эллиптической криптографии:

- Вероятность появления субэкспоненциальных алгоритмов решения задачи дискретного логарифмирования. При их появлении алгоритмы шифрования на эллиптических кривых будут легко решаемы[4].
- При переходе на алгоритмы шифрования основанных на эллиптических кривых велика вероятность выявления большого числа ошибок и уязвимостей, которые уже отработаны для более привычных методов шифрования.

2. Математические свойства эллиптических кривых

2.1 Определение эллиптических кривых

Для начала определим эллиптическую кривую как алгебраическую кривую, те каким-то множеством точек которые удовлетворяют следующему уравнению: (1)

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \text{ (1)}$$

где x, y – переменные, a_1, a_2, a_3, a_4, a_6 – коэффициенты. Так-же уравнение(1) можно представить как(2):

$$y^2 = x^3 + ax + b \text{ (2)}$$

где x, y – переменные, a, b– коэффициенты. Функция (2) называется функцией Вейерштрасса, не все эллиптические кривые можно представить таким уравнением, но для большинства использующихся в криптографии он корректен.

Так как график кривой параллелен оси абсцисс, чтобы найти точки, являющиеся корнями, нужно решить уравнение третьей степени (3).

$$x^3 + ax + b = 0 \text{ (3)}$$

Здесь можно использовать формулу Кардано. Дискриминант вычисляется по формуле (4)

$$D = \left(\frac{a}{3}\right)^3 + \left(\frac{b}{3}\right)^2 \text{ (4)}$$

При дискриминанте меньше нуля, уравнение (3) имеет три разных решения a, b, z; при дискриминанте равном нулю, уравнение (3) имеет три корня, a, b, c, два из которых одинаковые, при дискриминанте больше нуля, уравнение (3) имеет одно решение a и два комплексно сопряженных. Графики по результатам вычислений представлены на рисунках 1-3.

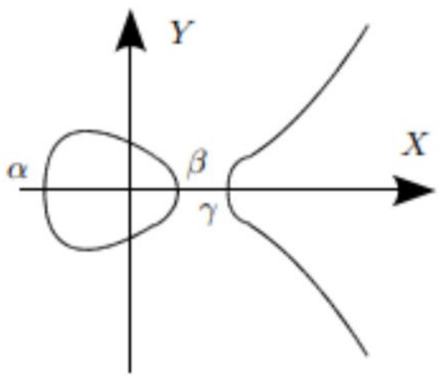

Рисунок 1 – Кривая с D < 0

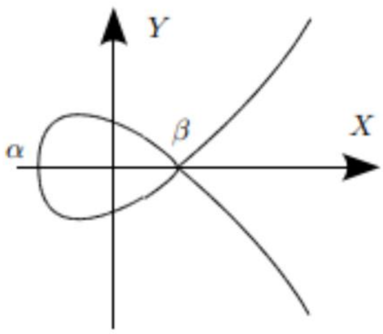

Рисунок 2 – Кривая с D = 0

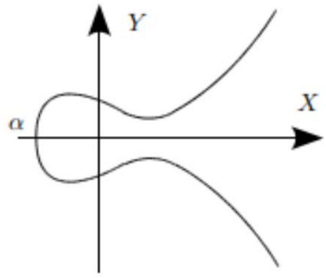

Рисунок 3 – Кривая с D > 0

Эллиптические кривые дискриминант которых не равен нулю (рисунок 1, 3) называются несингулярными, соответственно если дискриминант равен нулю, то сингулярные. Последние используются редко в связи с снижением криптостойкости алгоритмов и протоколов.

2.2 Операции над точками эллиптической кривой

2.2.1 Сложение точек

Пусть $(x_p;\ y_p)$ координаты точки P, а $(x_q;\ y_q)$ координаты точки Q, для нахождения точки R $(x_r;\ y_r)$ – суммы точек P и Q, необходимо провести прямую через эти точки P и Q, получаем пересечение прямой с кривой в точке R' и отразить эту точку относительно OX(рис. 1). То есть

$$P + Q + R' = 0\ ,\ P + Q = R \quad (5)$$

Для нахождения координат точки R найдем коэффициент α,

$$\alpha = \frac{y_q - y_p}{x_q - x_p}\ \text{, далее}\ y_r\ =\ -y_p + \alpha(x_p - x_{r'})\ ,\ x_r\ =\ \alpha^2 - x_p - x_q$$

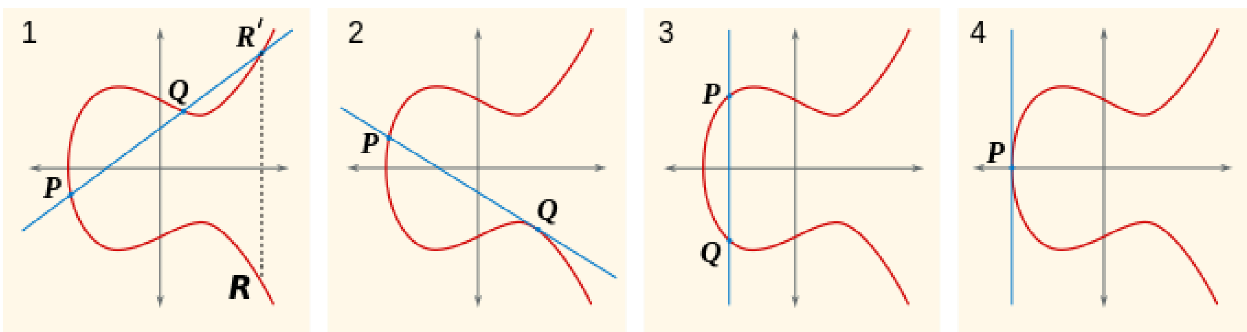

Рисунок 4.1-4.4

Для случая если прямая пересекает кривую только в двух местах и пересекает оси (рис. 4.2), то выполняется следующие уравнение $P + Q + Q = 0$ (6), соответственно $P + Q = -Q$ (7)

Третий случай если прямая пересекает кривую в двух местах и параллельна оси ординат (рис. 4.3):

$$P + Q + 0 = 0 \text{ (8)}$$

Четвертый если прямая касается кривой в одной точке (рис. 4.4):

$$P + P + 0 = 0 \text{ (9)}$$

2.2.2 Вычитание точек

Пусть $(x_p;\ y_p)$ координаты точки P, а $(x_q;\ y_q)$ координаты точки Q, $(x_r;\ -y_r)$ – координаты точки -Q. Вычитание точек, это сложение точек с обратной точкой(10).

$$R = P - Q = P + (-Q) \text{ (10)}$$

2.2.3 Умножение точки на число

Пусть P – точка на эллиптической кривой, n – любое целое число, Q = n * P – произведение точки P на число n. Для нахождения Q будем использовать алгоритм быстрого умножения.

Разберем алгоритм умножения, пусть n = 37:

1. Разложим n по степеням двойки:
   n = 37 = 32 + 4 + 1
2. Раскладываем произведение n на P:
   Q = 37*P = 32*P + 4*P + P

   Рассмотрим возможные слагаемые:
   1*P = P
   2*P = P + P
   4*P = 2*P + 2*P
   8*P = 4*P + 4*P
   16*P = 8*P + 8*P
   32*P = 16*P + 16*P, можем заметить, что для вычисления Q потребуется 7 сложений.

3. Алгоритмы на эллиптических кривых

3.1.1 Алгоритм Диффи-Хеллмана

Рассмотрим пример. Предположим, существует два абонента: Алиса и Боб. Обоим абонентам известны некоторые два числа g и p, которые не являются секретными и могут быть известны также другим заинтересованным лицам. Для того, чтобы создать неизвестный более никому секретный ключ, оба абонента генерируют случайные числа: Алиса — число a, Боб — число b. Затем Алиса вычисляет остаток от деления (11):

$$A = g^a mod\ p \text{ (11)}$$

и пересылает его Бобу, и Боб вычисляет остаток от деления (12):

$$B = g^b mod\ p \text{ (12)}$$

и передаёт Алисе. Предполагается, что злоумышленник может получить оба этих значения, но не модифицировать их. На втором этапе Алиса на основе имеющегося у неё a и полученного по сети B вычисляет значение (13):

$$B^a\ mod\ p = g^{ab}\ mod\ p \text{ (13)}$$

Боб на основе имеющегося у него b и полученного по сети A вычисляет значение (14):

$$A^b \, mod\, p = \; g^{ab} \, mod\, p \; (14)$$

Можем видеть что, у Алисы и Боба получилось одно и то же число (15):

$$K = \; g^{ab} \, mod\, p \; (15)$$

Его они могут использовать в качестве секретного ключа.

Работа алгоритма показана на рисунке.

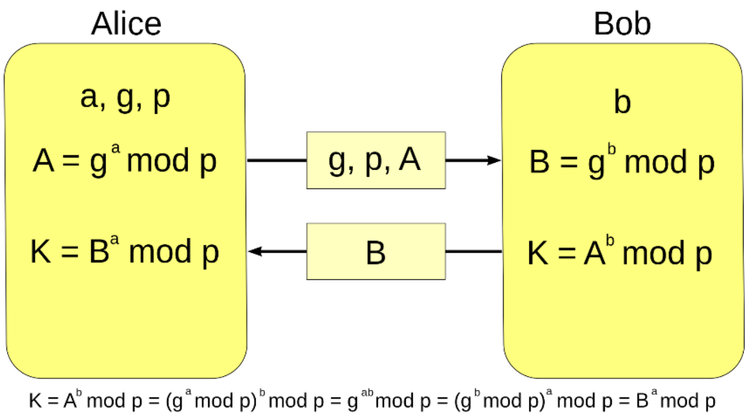

Рисунок 5 - Работа алгоритма.

При работе алгоритма каждая сторона [5]:

1. Генерирует случайное натуральное число a — закрытый ключ.
2. Совместно с удалённой стороной устанавливает открытые параметры g и p.
3. Вычисляет открытый ключ A, используя преобразование (11) над закрытым ключом.

$$A \; = \; g^a \, mod\, p$$

4. Обменивается открытыми ключами с удалённой стороной.
5. Вычисляет общий секретный ключ K (15), используя открытый ключ удаленной стороны B и свой закрытый ключ a.

$$K \; = \; B^a \, mod\, p$$

$$B^a \, mod\, p \; = \; (g^b \, mod\, p)^a \, mod\, p \; = \; g^{ab} \, mod\, p \; = \; (g^a \, mod\, p)^b \, mod\, p \; = \; A^b \, mod\, p$$

3.1.2 Числовая реализация

Пусть s = 2 - секретный ключ, g = 5 - первообразный корень по модулю p, p = 23 - открытое простое число, a = 6 - секретный ключ Алисы, A = g^a mod p = 8 - открытый ключ Алисы, b = 15 - секретный ключ Боба, B = g^b mod p = 19 - открытый ключ Боба.

Тогда, пройдя и записывая каждый шаг в алгоритме Диффи-Хеллмана, составим следящую таблицу:

Alice		Bob	
Знает	Не знает	Знает	Не знает
p = 23	b = ?	p = 23	a = ?
g = 5		g = 5	
a = 6		b = 15	
A = 5^{6} mod 23 = 8		B = 5^{15} mod 23 = 19	
B = 5^{b} mod 23 = 19		A = 5^{a} mod 23 = 8	
s = 19^{6} mod 23 = 2		s = 8^{15} mod 23 = 2	
s = 8^{b} mod 23 = 2		s = 19^{a} mod 23 = 2	
s = 19^{6} mod 23 = 8^{b} mod 23		s = 8^{15} mod 23 = 19^{a} mod 23	
s = 2		s = 2	

3.1.3 Программная реализация

Перейдем к программной реализации алгоритма. Алгоритм Диффи-Хеллмана реализован на языке C, в среде разработки CLion.

Создаем функцию для вычисления $a^m \ mod \ n$ (в нашем случае уравнения 11,12 рис. 6):

```
// Функция для вычисления `a^m mod n`
int compute(int a, int m, int n) {
    int r;
    int y = 1;

    while (m > 0) {
        r = m % 2;

        if (r == 1) {
            y = (y * a) % n;
        }
        a = a * a % n;
        m = m / 2;
    }
    return y;
}
```

Рисунок 6

Объявляем следующие значения согласно числовой реализации: g - первообразный корень по модулю p, p - открытое простое число, a - секретный ключ Алисы, A - открытый ключ Алисы, b - секретный ключ Боба, B - открытый ключ Боба. И действуем согласно алгоритму (рис.7).

```c
// Программа на C для демонстрации алгоритма Диффи-Хеллмана
int main()
{
    int p = 23;        // открытое простое число,
    int g = 5;         // первообразный корень по модулю p

    int a, b;    // `a` - секретный ключ Алисы, `b` - секретный ключ Боба.
    int A, B;    // `A` - открытый ключ Алисы, `B` - открытый ключ Боба

    a = 6;          // выбираем секретное целое число для закрытого ключа Алисы (известного только Алисе)
    A = compute(g, a, p); // Вычисление открытого ключа Алисы (Алиса отправит Бобу `A`)

    b = 15;        // выбираем секретное целое число для закрытого ключа Боба (известного только Бобу)
    B = compute(g, b, p); // Вычислить открытый ключ Боба (Боб пошлет `B` Алисе)

    // Алиса и Боб обмениваются своими открытыми ключами `A` и `B` друг с другом

    // Находим секретный ключ
    int keyA = compute(B, a, p);
    int keyB = compute(A, b, p);

    printf("Alice's secret key is %d\nBob's secret key is %d", keyA, keyB);

    return 0;
}
```

Рисунок 7.

Результат выполнения программы (секретный ключ) (рис.8).

```
Alice's secret key is 2
Bob's secret key is 2
Process finished with exit code 0
```

Рисунок 8 – Результат выполнения программы.

3.2.1 Алгоритм ECDSA (Elliptic Curve Digital Signature Algorithm)

ECDSA — алгоритм с открытым ключом, использующийся для построения и проверки электронной цифровой подписи(ЭЦП). Алгоритм начинается с выбора параметров эллиптической кривой, для облегчения этой задачи национальным институтом стандартов и технологий (NIST), был составлен список эллиптических кривых с уже известным количеством точек, которые рекомендовано использовать в схемах ЭЦП.

Кривая в стандарте описывается набором из 6 параметров D=(p,a,b,G,n,h), где

p – простое число, модуль эллиптической кривой, данное число относится к обобщенным числам Мерсенна, это означает, что его можно представить как сумму различных степеней двойки.

a, b – задают уравнение эллиптической кривой(2).

G – точка эллиптической кривой большого порядка.

n – порядок точки G;

h – параметр, называемый кофактор. Определяется отношением общего числа точек на эллиптической кривой к порядку точки G. Данное число должно быть как можно меньше[5]. Вот несколько кривых рекомендованных NIST (табл. 2, 3).

Curve P-192	
p=	6277101735386680763835789423207666416083908700390324961279
n=	6277101735386680763835789423176059013767194773182842284081
a=	-3
b=	64210519 e59c80e7 0fa7e9ab 72243049 feb8deec c146b9b1
Gx=	188da80e b03090f6 7cbf20eb 43a18800 f4ff0afd 82ff1012
Gy=	07192b95 ffc8da78 631011ed 6b24cdd5 73f977a1 1e794811
h=	1

Таблица 2 – Curve P-192

Curve P-224	
p=	26959946667150639794667015087019630673557916260026308143510066298881
n=	26959946667150639794667015087019625940457807714424391721682722368061
a=	-3
b=	b4050a85 0c04b3ab f5413256 5044b0b7 d7bfd8ba 270b3943 2355ffb4
Gx=	b70e0cbd 6bb4bf7f 321390b9 4a03c1d3 56c21122 343280d6 115c1d21
Gy=	bd376388 b5f723fb 4c22dfe6 cd4375a0 5a074764 44d58199 85007e34
h=	1

Таблица 3 – Curve-224

Точка G принадлежит эллиптической кривой. Соответственно для нее выполняется равенство(2), из которого можем вычислить y(16):

$$y = \sqrt{x^3 + ax + b} \, mod \, p \quad (16)$$

Формирование и проверка подписи

Рассмотрим алгоритм обмена ключами. Пусть пользователи A и B хотят обменяться ключами, но их трафик прослушивает злоумышленник E. Алгоритм следующий:

1. Пользователь A генерирует случайно число d_A в диапазоне [1; n-1]. Это число его закрытый ключ.
2. Затем A вычисляет $Q_A = d_A G$ и посылает координаты точки пользователю B. Q_A – открытый ключ пользователя A.
3. Пользователь B генерирует случайно число d_B в диапазоне [1; n-1]. Это число его закрытый ключ.
4. Затем B вычисляет $Q_b = d_B G$ и посылает координаты точки пользователю A. Q_B – открытый ключ пользователя B.
5. Пользователь A получает Q_b, вычисляет $R = d_A Q_B$ и считает, что x_R– это общий ключ.
6. Пользователь B получает Q_{Aa}, вычисляет $R = d_B Q_A$ и считает, что x_R– это общий ключ.

Оба пользователя получили один и тот же ключ, потому что $d_A Q_B = d_A d_B G = d_B Q_A$
Злоумышленник E видит только Q_A и Q_B. – открытые ключи пользователей.[5]

Список использованных источников:

1. *СТБ 34.101.45-2013 АЛГОРИТМЫ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ И ТРАНСПОРТА КЛЮЧА НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ*
2. *Guide to Elliptic Curve Cryptography*
3. *Шифрование данных на базе эллиптических кривых, Д.Ф. Пастухов Ю.Ф. Пастухов П.Р. Синица*
4. *Интернет-ресурс: https://habr.com/ru/articles/692842/*
5. *Интернет ресурс: http://habrahabr.ru/post/191240/*

UDC

INTRODUCTION TO ELLIPTICAL CRYPTOGRAPHY

Patsiupin M.S.

Belarusian State University of Informatics and Radioelectronics[1], Minsk, Republic of Belarus

Surname N.P. – PhD in Physics and Mathematics

Annotation. Mathematical properties of elliptic curves, the Diffie-Hellman algorithm, its description, numerical and software implementation. The principle of operation of the ECDSA algorithm and the selection of parameters of the elliptic curve.

Keywords. ECDSA algorithm, Diffie-Hellman algorithm, elliptic curves, elliptic cryptography.