О КРИПТОГРАФИИ, ОСНОВАННОЙ НА АЛГЕБРАИЧЕСКИХ ТОРАХ

Пихтилькова О.А., Казакова О.Н.
Оренбургский государственный университет, г. Оренбург

Развитие математики и информатики, компьютерных технологий, позволяют создавать новые системы шифрования-дешифрования для передачи информации. Использование таких систем в реальной жизни необходимо требует и подготовки соответствующих специалистов.

В настоящее время широкое распространение получили системы с открытым ключом на основе алгебраической геометрии: эллиптическая и торическая криптография.

Эллиптическая криптография уверенно завоевала место среди профессиональных шифросистем. За более чем 25 лет после ее появления практическая выгода от использования эллиптических кривых осознана всеми: она предлагает меньший размер ключа и более эффективные преобразования при том же уровне криптостойкости.

Торическая криптография была предложена в последнее десятилетие К. Рубиным и А. Сильвебергом [1,2]. Шифросистема, основанная на ней, получила название CEILIDH. Возможно, что торическая криптография повторит успех эллиптической.

В основе торической криптографии лежит математическое понятие – алгебраический тор. Алгебраический тор – алгебраическая группа, изоморфная над некоторым расширением основного поля прямому произведению конечного числа мультипликативных групп G_m. В теории алгебраических групп алгебраический тор играет роль, схожую с ролью торов в теории групп Ли.

Не будем останавливаться на теории алгебраических торов. Подробно она изложена в [3, 4].

Обозначим через F_{q^r} конечное поле из q^r, где q – простое. В работе рассмотрена шифросистема Эль-Гамаля для алгебраического тора $T_2(F_q)$.

Согласно лемме:

1) $T_n(F_q) \cong G_{q,n}$;

2) $\left|T_n(F_q)\right| = \Phi_n(q)$;

3) Если $h \in T_n(F_q)$ элемент простого порядка не делящего n, то тогда h не лежит в собственном подполе расширения F_{q^r} / F_q,

тор $T_n(F_q)$ имеет такую же криптостойкость, как и мультипликативная группа $F^*_{q^n}$ [2].

В качестве учебного примера эта шифросистема реализована на компьютере на алгоритмическом языке C# для q=2081. Длина передаваемого сообщения равна 8 бит.

Предположим, что целое число d является квадратичным невычетом в поле F_q. Тогда $F_{q^2} = F_q(\sqrt{d})$.

Определим отображение $\Psi : A^1(F_q) \to T_2(F_q)$ одномерного аффинного пространства в мультипликативную группу $F^*_{q^2}$.

Положим $\psi(a) = \dfrac{a+\sqrt{d}}{a-\sqrt{d}}$. Обратное отображение определяется по формуле $\rho(\beta_1 + \beta_2\sqrt{d}) = \dfrac{1+\beta_1}{\beta_2}$.

Отображения ρ и ψ являются взаимообратными отображениями множеств $A^1\{0\}$ и .

Порядок группы T_2 равен $\Phi_2(2081) = 2082 = 2 \cdot 3 \cdot 347$, где $\Phi_n(x)$ – многочлен деления круга на n частей.

Пусть α – элемент порядка $l = 347$ в T_2. Для этого обычно достаточно возвести элемент T_2 в степень $\dfrac{q^2-1}{l}$. В нашем примере это элемент $\psi(1)^{12480}$.

Абонент *А* выбирает случайное число α, $1 \le \alpha < l-1$ и вычисляет $\beta = \alpha^a$.

Абонент *Б* выбирает случайное число k, $1 \le k < l-1$.

Пусть *M* – сообщение, целое число $1 \le M < q-1$.

Абонент *Б* вычисляет $\gamma = \rho(\alpha^\kappa)$, и посылает шифротекст (γ, δ).

Открытым ключом являются (q, α, β), секретным ключом является a.

Абонент *А* расшифровывает сообщение по формуле $\rho(\psi(\delta)\psi(\gamma)^{-a})$.

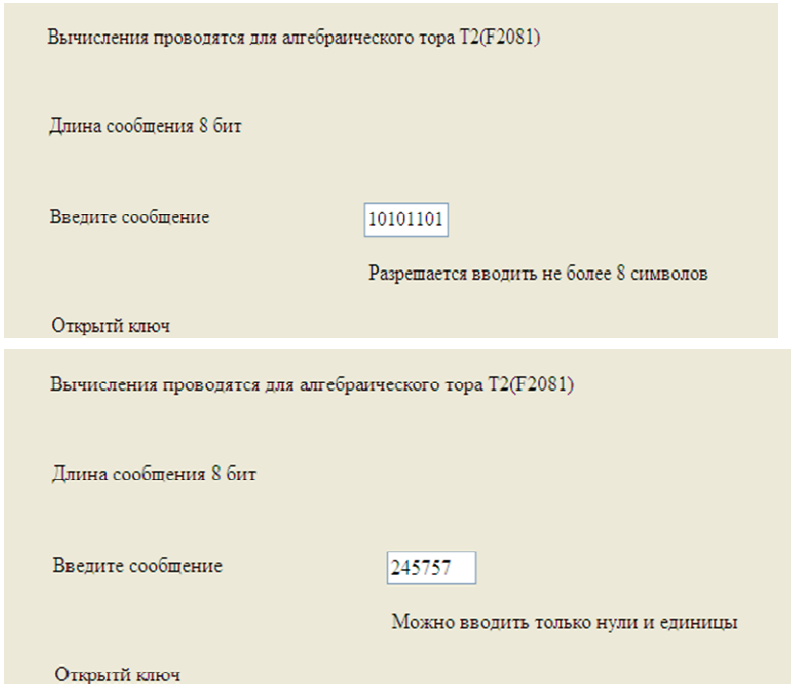

Рисунок 1. Пример выполнения программы в случае некорректного ввода информации

В работе указано, в отличие от [1], явное преобразование сообщения *M* в элемент группы T_2.

На рисунках 1 и 2 показаны примеры работы программы: длина сообщения составляет 8 бит и состоит из нулей и единиц. Программа определяет корректность вводимой информации и в случае несоответствия требованиям выводит на экран соответствующее замечание (рисунок 1). Открытый ключ состоит из трех чисел, закрытый – из одного. Также выводится передаваемый шифротекст. Как видим, отправляемое сообщение и сообщение после декодирования одинаковы (рисунок 2).

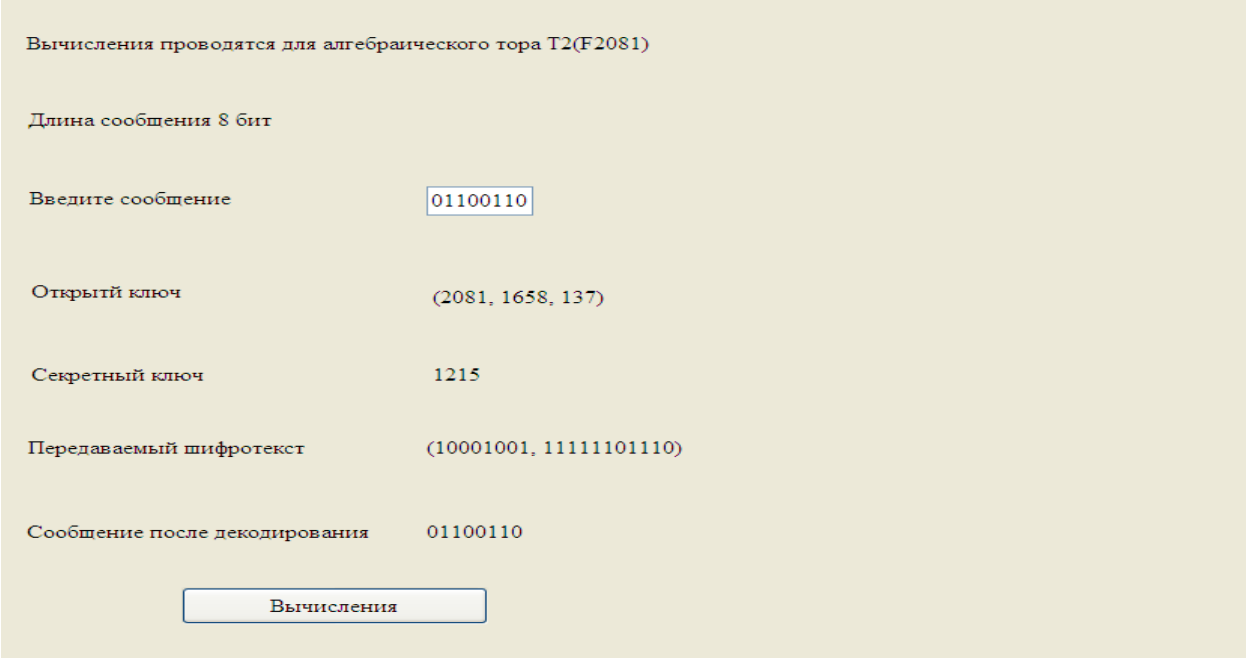

Рисунок 2. Пример выполнения программы в случае корректного ввода информации

В программе описан следующий класс, реализующий элементы поля $F_{q^2} = F_q(\sqrt{d})$.

```
public class Fq
        {
            private int a, b, q = 2081;
            public Fq(int a, int b)
            {
                this.a = a; this.b = b;
            }
            public string ToString (string format)
            {
                return Convert.ToString(a) + "+sqr(3)" + Convert.ToString(b);
            }
```

С помощью перегрузки операторов реализованы операции сложения, умножения и деления классов.

```
            public static Fq operator + (Fq x, Fq y)
            {
                int c=(x.a+y.a)%x.q;
                int d=(x.b+y.b)%x.q;
                return new Fq(c,d);
            }
            public static Fq operator *(Fq x, Fq y)
            {
                int c = (x.a * y.a + 3 * x.b * y.b + x.q * x.q) % x.q;
                int d = (x.a * y.b + x.b * y.a + x.q * x.q) % x.q;
                return new Fq(c, d);
```

```
        }
        public static Fq operator /(Fq x, Fq y)
        {
            Fq ys = new Fq(y.a, -y.b);
            int Dn = (y.a * y.a - 3 * y.b * y.b + x.q * x.q) % x.q;
            Fq Nm = x * ys;
            int c = (Nm.a * MultObr(Dn, x.q)+x.q * x.q) % x.q;
            int d = (Nm.b * MultObr(Dn, x.q)+x.q * x.q) % x.q;
            return new Fq(c, d);

        }
```

Использован стандартный алгоритм нахождения линейного представления наибольшего общего делителя, с помощью которого находится мультипликативный обратный.

```
    private static int GCD(int a, int b, out int x, out int y)
    {
        if (a == 0)
        {
            x = 0;
            y = 1;
            return b;
        }
        int x1, y1;
        int d = GCD(b % a, a, out x1, out y1);
        x = y1 - (b / a) * x1;
        y = x1;
        return d;
    }
    public static int MultObr(int a, int q)
    {
        int x, y;
        GCD(a, q, out x, out y);
        return x;
    }
```

Следующие программы реализую операторы $\rho(\beta_1+\beta_2\sqrt{d})=\frac{1+\beta_1}{\beta_2}$ и $\psi(a)=\frac{a+\sqrt{d}}{a-\sqrt{d}}$.

```
        public static int Rho(Fq x)
        {
            int y = ((1 + x.a) * MultObr(x.b, x.q)+ x.q * x.q) % x.q;
            return y;
        }
    public static Fq Psi(int a)
    {
        Fq x = new Fq(a,1);
        Fq y = new Fq(a, -1);
        Fq z = x / y;
        return z;
    }
```

Для возведения класса в степень используется стандартный алгоритм.

```
    public static Fq Pow(Fq a, int k)
    {
       Fq b = new Fq(1, 0); Fq s;
        while (k>0)
        {
            int r = 0;
            int q = k / 2; r=k % 2;
```

```
            if (r == 0)
            {
                k = q;
                s = a*a;
                a = s;
            }
            else
            {
                k = k-1;
                s = b*a;
                b = s;
            }
        }
        return b;

    }
```

Следующие операторы осуществляют кодирование-декодирование сообщения.

```
        Fq s = Psi(1);
        Fq alpha = Form1.Pow(s, 12480);
        Fq st = Form1.Pow(alpha, 347);
        int a = 5;
        int Pa = Fq.Rho(Form1.Pow(alpha, a));
        Random rnd = new Random();
        int k = rnd.Next(0,2080);
        int gamma = Fq.Rho(Form1.Pow(alpha, k));
        byte[] bytes = System.Text.Encoding.ASCII.GetBytes(textBox1.Text);
        n = bytes.Length;
        int M = 0;
        for (i = 0; i < n; i++)
        {
            if (bytes[i] == Convert.ToByte(48)) M = 2 * M;
            if (bytes[i] == Convert.ToByte(49)) M = 2 * M+1;
        };
        Fq MFq = Psi(M);
        int delta = Fq.Rho(MFq*Form1.Pow(Psi(Pa), k));
        Fq DMFq = Psi(delta) * Form1.Pow(Psi(gamma), 347 - a);
        int DM = Fq.Rho(DMFq);
```

Список литературы

1. Rubin K., Silverberg A. Torus-based cryptography // Advances of Cryptology.- 2003.- P. 349-365.

2. Rubin K., Silverberg A. Compression in finite fields and Torus-based cryptography // Siam J. Comp.- 2008.- V. 37(5).- P. 1401-1428.

3. Воскресенский, В.Е. Алгебраические торы / В.Е. Воскресенский.- М.: Наука, 1977.- 223 с.

Воскресенский, В.Е. Бирациональная геометрия линейных алгебраических групп / В. Е. Воскресенский. - М. : МЦНМО, 2009. - 404 с.