УДК 004.223.2

СЛОЖЕНИЕ ДВУХ ЭЛЛИПТИЧЕСКИХ ТОЧЕК И ГРУППОВЫЕ ОПЕРАЦИИ. СУЖЕНИЕ МНОЖЕСТВА ЭЛЛИПТИЧЕСКИХ ТОЧЕК НА МНОЖЕСТВО ОСТАТКОВ ЭЛЛИПТИЧЕСКОЙ КРИВОЙ. ГРАФИЧЕСКОЕ ТЕСТИРОВАНИЕ С ИНТЕРФЕЙСОМ.

П.Р. СИНИЦА
(Представлено: канд. физ.-мат. наук., доц. Д.Ф. ПАСТУХОВ)

В статье представлена определение сложения двух эллиптических точек и групповые операции. Сужение множества рациональных бесконечного числа эллиптические точек на конечное целочисленное множество остатков эллиптические кривой. Применение циклической абелевой группы. Программная реализация с графическим интерфейсом.

Ключевые слова: *эллиптическая кривая, шифрование, дешифрование, групповые операции, множество остатков, абелева группа, интерфейс.*

Введение. Эллиптическая криптография – раздел криптографии, который изучает асимметричные криптосистемы, основанные на эллиптических кривых над конечными полями.

Шифрование данных методом эллиптических кривых преследует цели выработать метод быстрого и эффективного шифрования на базе эллиптической криптографии и в то же время повысить устойчивость шифрования (стойкость шифра) и целостность передаваемой информации в процессе протоколе обмена.

Роль основной криптографической операции выполняет операция скалярного умножения точки на эллиптической кривой на данное целое число, определяемое через операции сложения и удвоения точек эллиптической кривой. Последние, в свою очередь, выполняются на основе операции сложения, умножения и инвертирования в конечном поле, над которыми рассматривается кривая. Особый интерес к криптографии эллиптических кривых обусловлен теми преимуществами, которые дают ее применение в беспроводных коммуникациях – высокое быстродействие и небольшая длина ключа.

Сложение двух эллиптических точек и групповые операции. Сужение множества. Циклическую группу образуют из множества точек эллиптической кривой, связанных геометрической групповой структурой, дополняют полевой целочисленной структурой по модулю простого числа *p*:

$$y^2 = x^3 + ax + b(\mathrm{mod}\ p) \tag{1}$$

В конечном итоге мы пользуемся формулами (3), (4) и (6), получая последовательно все точки эллиптической кривой циклической абелевой группы.

Формулы (1), (2), (3), (4) получены в первой части статьи:

$$\begin{cases} (\dfrac{y_2 - y_1}{x_2 - x_1}) \equiv (y_2 - y_1) \bmod p * (x_2 - x_1)^{-1} \bmod p, (x_2 - x_1)(x_2 - x_1)^{-1} \equiv 1 \bmod p \\ \dfrac{3x_1^2 + a}{2y_1} \equiv (3x_1^2 + a) \bmod p * (2y_1)^{-1} \bmod p, (2y_1) * (2y_1)^{-1} \equiv 1 \bmod p \end{cases} \tag{2}$$

Краткое описание алгоритма построения последовательности точек:

1. Находим обратный элемент к $2y_1$, либо к $x_2 - x_1$.
2. Находим числа $k_1 = (y_2 - y_1) \bmod p * (x_2 - x_1)^{-1} \bmod p,$ либо $k_1 = (3x_1^2 + a), \bmod p * (2y_1)^{-1} \bmod p.$
3. Находим числа

$$\begin{cases} x = (k_1^2 - x_1 - x_2) \bmod p \\ y = (-y_1 + k_1(2x_1 + x_2 - k_1^2)) \bmod p \end{cases} \tag{3}$$

Либо по формулам:

$$\begin{cases} x = (k_1^2 - 2x_1) \bmod p \\ y = (-y_1 + k_1(3x_1 - k_1^2)) \bmod p \end{cases} \tag{4}$$

Если говорить вкратце, то формулу шифрования и дешифрования можно записать в виде:

(*kG, Pm + k * Pb*) (шифрование) $\rightarrow$ *Pm + k * nb * G – nb * k * G = Pm* (дешифрование), где *nb* – закрытый ключ абонента *b*, а *Pb* открытый ключ абонента *b*.

Программная реализация с графическим интерфейсом. В качестве интерфейса программы ввода данных можно рассмотреть, например, следующий интерфейс с вводом длины сообщения в символах *nn*, параметры эллиптической кривой *a*, *b*, *p*, открытый ключ (*kx*, *ky*), случайное число *k*.

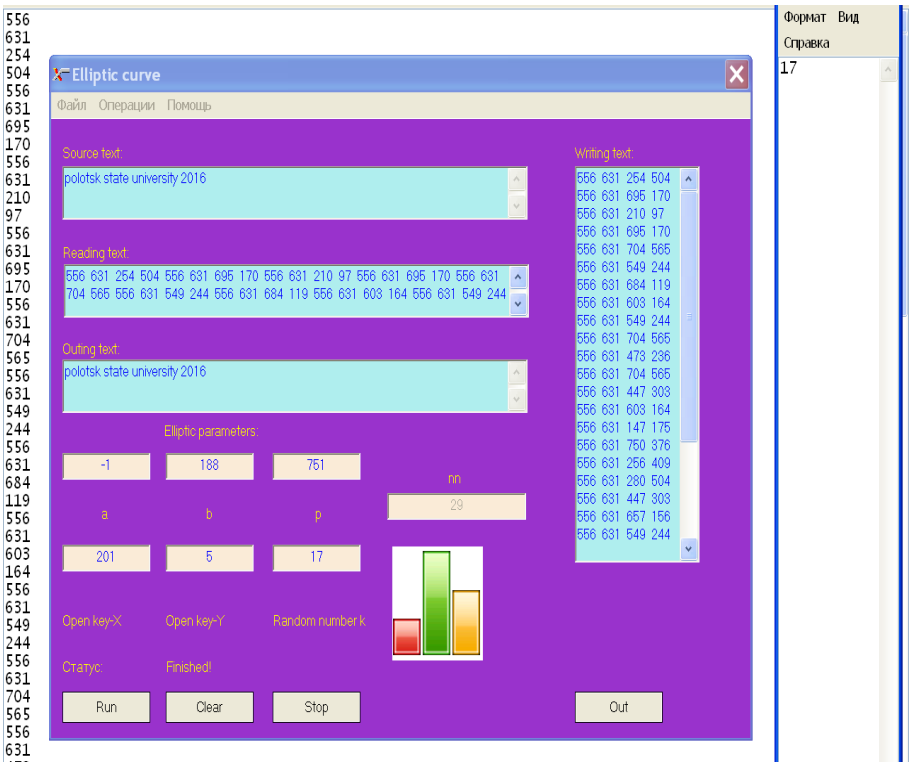

Рисунок 1. – Шифрование интерфейсом исходного текста

Интерфейс включает следующие окна: окна ввода чисел *a*, *b*, *p*, окна ввода открытого ключ Open Key – X, Open Key – Y, окно ввода случайного числа Random number *k*.

Функциональные элементы запуска, очистки и остановки интерфейса выхода из программы Run, Clear, Stop, Out.

Окно ввода первичной информации Source text, куда вводится исходный текст с помощью клавиатуры и установки курсора в окно ввода.

Окно шифрованного текста Writing text непосредственно перед записью шифра в текстовый файл *balka1.txt*.

Длина записи исходного текста в окне nn не является активной, т.е. в это окно ничего не заносится. Длина строки вводимого текста nn автоматически изменяется в данном окне при удалении или добавлении каждого нового символа.

Окно прочитанного шифра Reading text из файла 1.txt должно совпадать с шифром в окне Writing text.

Окно расшифрованного текста Outing text должно выводить текст, посимвольно совпадающий с исходным текстом.

Данный интерфейс не только шифрует исходный текст, но и является программой – контролем на всех этапах шифрования и дешифрования и создания текстового файла для шифра.

Интерфейс содержит также функциональные окна с падающим списком опций, с помощью которых также можно шифровать исходный текст (например, операции – кодировать, закрыть; файл – просмотреть; помощь – о программе).

На рисунке 1 показана введенная фраза с латинским шрифтом Полоцкий государственный университет 2019: «polotsk state university 2019» с длиной строки *nn* = 29. Результат ввода текста, параметры эллиптической кривой *a* = –1, *b* = 188, *p* = 751, открытый ключ (*kx*, *ky*) = (201, 5).

На рисунке 1 мы видим результат шифрования в поле Writing text. Каждому исходному символу текста соответствует четыре целые координаты двух точек эллиптической кривой, расположенных в одной строке. Для удобства ввода и построчного считывания тот же шифр в один столбец записывается в текстовый файл *1.txt*.

На рисунке 2 представлен интерфейс дешифрования, входными параметрами для которого являются данные из файла *1.txt* и *2.txt*.

Кроме того, интерфейс программы содержит пассивный логический оператор «Статус», который в случае успешного запуска возвращает состояние Finished. В случае ошибки данный логический оператор указывает состояние (род произошедшей ошибки).

Справа на рисунке 2 указано случайное число $k = 17$, записанное программой в текстовый файл *2.txt*. Оно совпадает с числом, введенным в интерфейс в окно Random number *k*. Однако по каналу связи в открытом виде число *k* не передается, так как оно скрыто в шифре вида ($k * G$, $Pm + k * Pb$). Число *k* в текстовый файл выводится только на этапе тестирования программы.

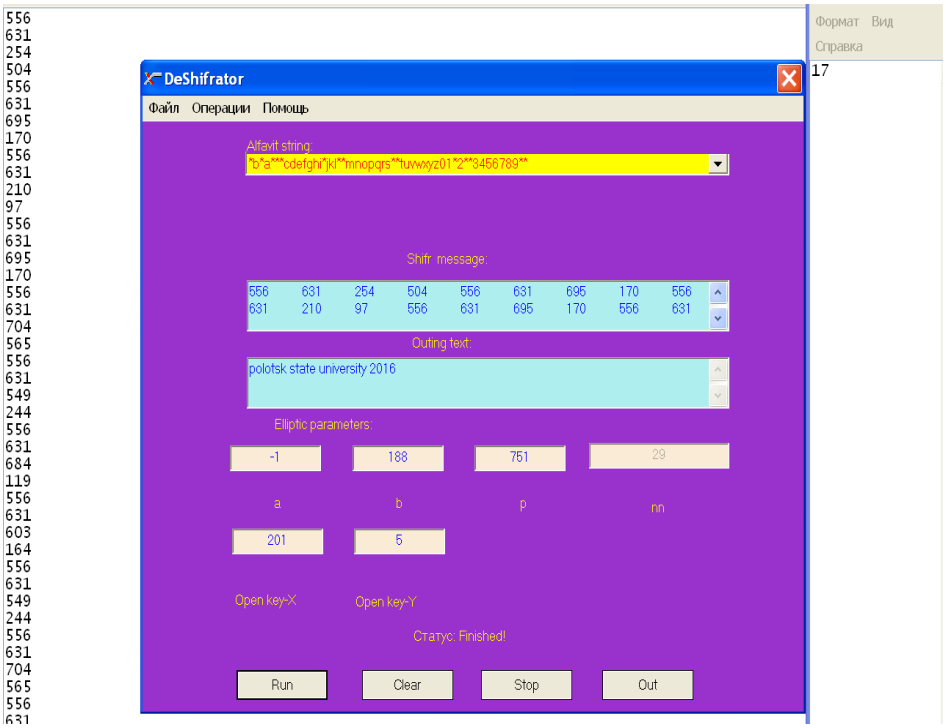

Рисунок 2. – Интерфейс дешифрования

Стоит отметить, что интерфейс дешифрования имеет 2 возможности:

1) загружать шифр из текстового файла и дешифровать отдельно;
2) вставлять шифр в окно шифра в режиме онлайн и дешифровать его в окно дешифрованного текста.

Заключение. Правильным подходом при разработке веб-интерфейса является использование современных технологий, которые позволяют решать свои задачи во всех браузерах. Это, в первую очередь, экономит время при разработке интерфейса, а также ресурсы на обработку данных. Это является очень актуальным, при большом количестве информации, которая подлежит постоянному изменению.

В данной статье был разработан дружелюбный интерфейс пользователя для шифрования и дешифрования данных на основе эллиптических кривых. Так же описано удобство использования.

ЛИТЕРАТУРА

1. Пастухов Д.Ф., Пастухов Ю.Ф., Синица П.Р., Шифрование данных на базе эллиптических кривых: учебно-методическое пособие для студентов спец. 1-98 01 01; Электронная библиотека Полоцкого государственного университета. – Полоцк, 2016. – С. 1–183.
2. Neal Koblitz. Random Curves: Journeys of a Mathematician, – Spinger, 2009. – С. 312 –313. – 402 с. – ISBN 9783540740780.
3. Koblitz N.A. Course in namber theory and cryptography. – USA: Spinger – Verlag?1994. – 235 с.
4. Ишмухаметов Ш.Т. Методы факторизации натуральных чисел. – Казань: КФУ,2011. – 1990 с.
5. Koblitz N.A. Course in namber theory and cryptography. – USA: Spinger – Verlag?1994. – 235 с.