Молдовяну П.А. , канд. тенх. наук, Молдовян Д.Н., Морозова Е.В. , канд. тенх. наук,
Пилькевич С.В., канд. тенх. наук

ПОВЫШЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ ПРОЦЕДУР КОММУТАТИВНОГО ШИФРОВАНИЯ

Ряд криптографических протоколов требует использования алгоритмов коммутативного шифрования. Алгоритм Полинга-Хеллмана реализует такое шифрование, используя операцию возведения сообщения в степень в конечном простом поле большого порядка. Рассматривается способ повышения производительности процедуры коммутативного шифрования путем реализации алгоритма Полинга-Хеллмана над конечным полем заданным в векторной форме, в котором операция умножения может быть эффективно распараллелена.

1. Введение

Для решения ряда специальных задач (построение протоколов игры в покер по телефону или передачи секретного сообщения по открытому каналу без использования процедуры распределения секретных ключей [1,2]) требуется применение алгоритмов коммутативного шифрования, обеспечивающих достаточную производительность, простоту аппаратной реализации и высокую стойкость криптографических протоколов. Этим требованиям в достаточной степени удовлетворяет алгоритм Полинга-Хеллмана [3], однако практика для ряда приложений выдвигает требования повышения производительности криптографических протоколов, основанных на коммутативном шифровании.

В настоящей работе решается задача ускорения процедур коммутативного шифрования путем применения конечных полей, заданных в векторной форме [4,5], для реализации алгоритма подобного алгоритму Полинга-Хеллмана.

2. Конечные поля с распараллеливаемой операцией умножения

В работах [4,5] предложен способ задания расширенных конечных полей, в которых операция умножения может быть эффективно распараллелена. Этот способ основан на идее представления элементов поля в векторной форме, т. е. в виде упорядоченного набора координат, значения которых лежат в некотором конечном поле $GF(p^m)$, называемом базовым полем, и задания операции умножения с помощью процедуры, в которой каждая координата результирующего вектора вычисляется независимо. Последнее определяет возможность эффективного распараллеливания операции умножения. Поля с возможностью эффективного распараллеливания операции умножения предложено называть векторными полями. Векторные конечные поля представляют интерес для реализации алгоритмов эллиптической криптографии [6] и алгоритмов электронной цифровой подписи, заданных над конечными группами невырожденных квадратных матриц [7]. Вводятся векторные конечные поля следующим образом.

Рассмотрим конечное множество m-мерных векторов вида $a\mathbf{e} + b\mathbf{i} +\ldots+ c\mathbf{v}$, где $\mathbf{e}$, $\mathbf{i}$ и $\mathbf{v}$ – формальные базисные вектора; a, b и z – целые числа, принадлежащие конечному полю $GF(p^s)$, где $s \geq 1$ – натуральное число (степень расширения базового поля), p – простое число (характеристика базового поля). Коэффициенты при базисных векторах называются координатами вектора. Вектора также записываются в виде упорядоченного набора координат, т. е. в виде $(a, b,\ldots, q) = a\mathbf{e} + b\mathbf{i} +\ldots+ q\mathbf{v}$. Выражения $a\mathbf{e}$, $b\mathbf{i}$ и $q\mathbf{v}$ обозначают вектора $(a, 0, \ldots, 0)$, $(0, b, 0,\ldots, 0)$ и $(0, \ldots, 0, q)$, соответственно, и называются компонентами вектора $(a, b,\ldots, q)$. Определим операцию сложения векторов как сложение одноименных координат: $(a, b,\ldots, q) + (a', b',\ldots, q') = (a + a', b + b', q + q')$. В последнем выражении знак + обозначает две разных операции – сложение элементов поля $GF(p^s)$ и сложение векторов, однако это не вносит неопределенности ввиду очевидности каждого конкретного варианта правильной интерпретации этого обозначения.

Умножение двух векторов определяется по правилу перемножения каждой компоненты первого вектора с каждой компонентой второго вектора, т. е. по формуле:

$$(a\mathbf{e}+b\mathbf{i}+\ldots+q\mathbf{v})\circ(x\mathbf{e}+y\mathbf{i}+\ldots+z\mathbf{v}) = ax\mathbf{e}\circ\mathbf{e}+ay\mathbf{e}\circ\mathbf{i}+\ldots+az\mathbf{e}\circ\mathbf{v}+$$
$$+bx\mathbf{i}\circ\mathbf{e}+by\mathbf{i}\circ\mathbf{i}+\ldots+bz\mathbf{i}\circ\mathbf{v}+\ldots+qx\mathbf{v}\circ\mathbf{e}+qy\mathbf{v}\circ\mathbf{i}+\ldots+qz\mathbf{v}\circ\mathbf{v}.$$

В правой части появляются произведения различных пар базисных векторов. Вместо них подставляется некоторый однокомпонентный вектор, задаваемый по так называемой таблице умножения базисных векторов (ТУБВ). Координаты таких однокомпонентных векторов, присутствующих в ТУБВ называются коэффициентами растяжения. После такой замены правая часть будет представлять собой сумму однокомпонентных векторов. После их сложения в общем случае получим результат в виде m-мерного вектора вида $a''\mathbf{e} + b''\mathbf{i} +\ldots+ c''\mathbf{v}$. Для задания векторного умножения, обладающего свойствами ассоциативности и коммутативности предложены различные варианты ТУБВ [5,7,8], с помощью которых можно задать формирование векторных конечных полей. При соответствующем выборе значений коэффициентов растяжения ε и μ для произвольных значений размерности m могут быть получены векторные конечные поля $GF((p^s)^m)$ с помощью ТУБВ общего вида, представленного таблицей 1.

Таблица 1. Общий тип распределения растягивающих коэффициентов $\varepsilon \in GF(p^s)$ и $\mu \in GF(p^s)$

$\circ$	**e**	**i**	**j**	**k**	**u**	**v**	**...**	**...**	**z**
e	**e**	**i**	**j**	**k**	**u**	**v**	**...**	**...**	**z**
i	**i**	ε**j**	ε**k**	ε**u**	ε**v**	ε…	ε…	ε**z**	εμ**e**
j	**j**	ε**k**	ε**u**	ε**v**	ε…	ε…	ε**z**	εμ**e**	μ**i**
k	**k**	ε**u**	ε**v**	ε…	ε…	ε**z**	εμ**e**	μ**i**	μ**j**
u	**u**	ε**v**	ε…	ε…	ε**z**	εμ**e**	μ**i**	μ**j**	μ**k**
v	**v**	ε…	ε…	ε**z**	εμ**e**	μ**i**	μ**j**	μ**k**	μ**u**
...	**...**	ε…	ε**z**	εμ**e**	μ**i**	μ**j**	μ**k**	μ**u**	μ**v**
...	**...**	ε**z**	εμ**e**	μ**i**	μ**j**	μ**k**	μ**u**	μ**v**	μ…
z	**z**	εμ**e**	μ**i**	μ**j**	μ**k**	μ**u**	μ**v**	μ…	μ…

При $\mu = 1$ получаем ТУБВ с распределением коэффициента растяжения ε (первый вариант), а при $\varepsilon = 1$ – ТУБВ с распределением коэффициента растяжения μ (второй вариант). Оба варианта общего распределения коэффициентов растяжения могут быть скомбинированы. Такая комбинация также задает ассоциативную и коммутативную операцию векторного умножения для произвольных значений m при любых $\varepsilon, \mu \in GF(p^s)$. Следующие два утверждения доказываю, что каждый из предложенных общих типов распределения коэффициентов растяжения реализуют задание ассоциативной операции умножения m-мерных векторов. Обозначим базисные вектора $\mathbf{e}, \mathbf{j}, \mathbf{k}, \mathbf{u}, \mathbf{v}, \ldots,$ как $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \ldots,$ соответственно.

Утверждение 1. При $\varepsilon = 1$ и при любом значении $\mu \in F$, где $s \geq 1$, для произвольных троек $\mathbf{v}_i$, $\mathbf{v}_j$ и $\mathbf{v}_k$, где $i, j, k \in \{0, 1, 2, \ldots, m-1\}$, выполняется закон ассоциативности умножения базисных векторов:

$$(\mathbf{v}_i \circ \mathbf{v}_j) \circ \mathbf{v}_k = \mathbf{v}_i \circ (\mathbf{v}_j \circ \mathbf{v}_k). \tag{1}$$

Доказательство. Легко заметить, что для любых пар значений i и j имеет место:

$$\mathbf{v}_i \circ \mathbf{v}_j = \mu^b \mathbf{v}_h, \tag{2}$$

где $h = (i + j) \bmod m$ и $b = (i + j) \operatorname{div} m$. Аналогично, для любых троек $i, j, k \in \{0,1,2,\ldots,m-1\}$:

$$\mathbf{v}_i \circ \mathbf{v}_j \circ \mathbf{v}_k = \mu^b \mathbf{v}_h, \tag{3}$$

где $h = (i + j + k) \bmod m$ и $b = (i + j + k) \operatorname{div} m$. Из (3) непосредственно следует справедливость (1). □

Обозначим базисные вектора $\mathbf{e}, \mathbf{j}, \mathbf{k}, \mathbf{u}, \mathbf{v}, \ldots, \mathbf{z}$ как $\mathbf{v}_0, \mathbf{v}_{m-1}, \mathbf{v}_{m-2}, \mathbf{v}_{m-3}, \mathbf{v}_{m-4}, \ldots, \mathbf{v}_1$ соответственно.

Утверждение 2. При $\mu = 1$ и при любом значении $\varepsilon \in F$, где $s \geq 1$, для произвольных троек $\mathbf{v}_i$, $\mathbf{v}_j$ и $\mathbf{v}_k$, где $i, j, k \in \{0, 1, 2, \ldots, m-1\}$, выполняется закон ассоциативности умножения базисных векторов:

$$(\mathbf{v}_i \circ \mathbf{v}_j) \circ \mathbf{v}_k = \mathbf{v}_i \circ (\mathbf{v}_j \circ \mathbf{v}_k). \tag{4}$$

Доказательство. Легко заметить, что для любых пар значений i и j имеет место:

$$\mathbf{v}_i \circ \mathbf{v}_j = \varepsilon^b \mathbf{v}_h, \tag{5}$$

где $h = (i + j)\bmod m$ и $b = (i + j)\operatorname{div} m$. Аналогично, для любых троек $i, j, k \in \{0,1,2,...,m-1\}$:

$$\mathbf{v}_i \circ \mathbf{v}_j \circ \mathbf{v}_k = \varepsilon^{\mathrm{b}} \mathbf{v}_h, \tag{6}$$

где $h = (i + j + k)\bmod m$ и $b = (i + j + k)\operatorname{div} m$. Из (6) непосредственно следует справедливость (1). □

Рассмотренные выше два общих типа распределения коэффициентов растяжения используются далее для определения операции векторного умножения для произвольных значений размерности векторного пространства. В соответствии с результатами [4] при μ = 1 конечное m-мерное векторное пространство над конечным полем $GF(p^s)$ представляет собой векторное поле $GF((p^s)^m)$ при выполнении следующих двух условий:

1) m делит нацело значение $p^s - 1$;
2) уравнение $x^d = \varepsilon$ не имеет решений в поле $GF(p^s)$ для всех делителей $d > 1$ числа m.

Аналогично, при ε = 1 векторное поле формируется при выборе значения μ, при котором уравнение $x^d = \mu$ не имеет решений в поле $GF(p^s)$ для всех нетривиальных делителей $d|m$. Легко задать формирование векторных полей также и в случае, когда оба растягивающих коэффициента отличны от единицы поля $GF(p^s)$. Однако с целью получения более низкой временной сложности операции векторного умножения предпочтительно выбирать значение одного из коэффициентов ε и μ, равное единицы, а для второго коэффициента выбирать значение минимальной длины, при котором обеспечивается формирование векторного поля. Как правило, для произвольных значений m, p и s легко подобрать нееденичный коэффициент растяжения размером 2-3 бита, при котором обеспечивается формирование векторного поля. Другие подходы к снижению сложности умножения в векторных полях рассмотрены в работах [7,8].

3. Примеры векторных полей.

Приводимые ниже случаи формирования векторных полей иллюстрируют векторные поля, порядок которых имеет сравнительно малый размер. Во всех примерах используется значение коэффициента растяжения μ, равное единице базового конечного поля. Аналогичные примеры могут быть сгенерированы для практически важного случая векторных полей, имеющих размер порядка от 1024 до 2048 бит.

Случай $m = 3$, $p = 2$ и $s = 16$. Определим операцию умножения трехмерных векторов $a\mathbf{e} + b\mathbf{i} + c\mathbf{j}$, где координаты a, b и c представляют собой многочлены над полем $GF(2)$, значением коэффициента растяжения $\varepsilon = \varepsilon(z) = z^3 + 1$. В качестве неприводимого многочлена возьмем двоичный многочлен $\eta(z) = z^{16} + z^{15} + z^{14} + z^{12} + z^{11} + z^{10} + z^9 + z^2 + 1$, который можно также представить в хорошо известной форме записи двоичных многочленов $\eta(z) = (11101111000000101)$ в которой двоичный многочлен представлен последовательностью его коэффициентов. Выбранные параметры задают векторное конечное поле $GF((2^{16})^3)$, генератором мультипликативной группы которого является вектор $G_\Omega = (1101)\mathbf{e} + (1001)\mathbf{i} + (110)\mathbf{j}$.

Случай $m = 5$, $p = 2$ и $s = 12$. Определим операцию умножения пятимерных векторов $a\mathbf{e} + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} + g\mathbf{u}$, где координаты a, b, c, d и g представляют собой многочлены над полем $GF(2)$, значением коэффициента растяжения $\varepsilon = \varepsilon(z) = z^2 + 1$. В качестве неприводимого многочлена возьмем двоичный многочлен $\eta(z) = z^{12} + z^{10} + z^5 + z^4 + z^2 + z + 1$, который можно также представить в хорошо известной форме записи двоичных многочленов $\eta(z) = (1010000110111)$ в которой двоичный многочлен представлен последовательностью его коэффициентов. Выбранные параметры задают векторное конечное поле $GF((2^{12})^5)$, генератором мультипликативной группы которого является вектор $G_\Omega = (1101)\mathbf{e} + (1011)\mathbf{i} + (110)\mathbf{j} + (101)\mathbf{k} + (10)\mathbf{u}$.

Случай $m = 4$, $p = 3$ и $s = 12$. Определим операцию умножения четырехмерных векторов $a\mathbf{e} + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, где координаты a, b, c и d представляют собой многочлены над простым полем $GF(3)$, значением коэффициента растяжения $\varepsilon = \varepsilon(z) = z^2 + 1$. В качестве неприводимого многочлена возьмем троичный многочлен $\eta(z) = z^{12} + 2z^{11} + z^{10} + 2z^9 + 2z^8 + z^7 + z^6 + z^5 + z^3 + 1$, который можно также представить в хорошо известной форме записи троичных многочленов $\eta(z) = (1212211101001)$, в которой троичный многочлен представлен последовательностью его коэффициентов. Выбранные параметры задают векторное конечное поле $GF((3^{12})^4)$, генератором мультипликативной группы которого является вектор $G_\Omega = (1101)\mathbf{e} + (2011)\mathbf{i} + (112)\mathbf{j} + (121)\mathbf{k}$.

Случай $m = 5$, $p = 3$ и $s = 8$. Определим операцию умножения пятимерных векторов $a\mathbf{e} + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} + h\mathbf{u}$, где координаты a, b, c, d и h представляют собой многочлены над полем $GF(3)$, значением коэффициента растяжения $\varepsilon = \varepsilon(z) = z^2 + 1$. В качестве неприводимого многочлена возьмем троичный многочлен $\eta(z) = z^8 + z^7 + 2z^6 + z^5 + z^3 + z^2 + 1$, который можно также представить в хорошо известной форме записи троичных многочленов $\eta(z) = (112101101)$, в которой троичный многочлен представлен последовательностью его коэффициентов. Выбранные параметры задают векторное конечное поле $GF((3^8)^5)$, генератором мультипликативной группы которого является вектор $G_\Omega = (1201)\mathbf{e} + (2011)\mathbf{i} + (112)\mathbf{j} + (121)\mathbf{k} + (12)\mathbf{u}$.

Случай $m = 5$, $p = 991$ и $s = 2$. Этот случай относится к пространству пятимерных векторов $a\mathbf{e} + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} + h\mathbf{u}$, где координаты a, b, c, d и h представляют собой многочлены первой или нулевой степени над полем $GF(991)$. Все возможные значения многочленов степени, меньшей чем $s = 2$ (т.е. многочленов первой и нулевой степени), образуют поле $GF(991^2)$, в котором умножение задано как умножение многочленов по модулю неприводимого многочлена второй степени ($s = 2$), например, по модулю неприводимого многочлена $\eta(z) = z^2 + 373z + 601$. Определим операцию умножения пятимерных векторов значением коэффициента растяжения $\varepsilon = \varepsilon(z) = 3z^2 + 2z$. Выбранные параметры задают векторное конечное поле $GF((991^2)^5)$, генератором мультипликативной группы которого является вектор многочленов $G_\Omega = (3z + 1)\mathbf{e} + (5z + 7)\mathbf{i} + (3z + 2)\mathbf{j} + (z + 1)\mathbf{k} + (3z + 1)\mathbf{u}$.

Случай $m = 3$, $p = 127$ и $s = 5$. Рассмотрим пространство трехмерных векторов $a\mathbf{e} + b\mathbf{i} + c\mathbf{j}$, где координаты a, b, и c представляют собой многочлены над полем $GF(127)$. Все возможные значения многочленов степени не выше четвертой образуют поле $GF(127^5)$, в котором умножение задано как умножение многочленов по модулю неприводимого многочлена, например, $\eta(z) = z^5 + 120z^4 + 16z^3 + 114z^2 + 69z + 34$. Определим операцию умножения трехмерных векторов значением коэффициента растяжения $\varepsilon = \varepsilon(z) = z^2 + 5z + 2$. Выбранные параметры задают векторное конечное поле $GF((127^5)^3)$, генератором мультипликативной группы которого является вектор $G_\Omega = (z^3 + 2z^2 + z)\mathbf{e} + (7z^3 + 5z^2 + 3z + 2)\mathbf{i} + (z^3 + 3z^2 + 5z + 1)\mathbf{j}$.

Случай $m = 5$, $p = 268675256028581$ и $s = 1$. Определим операцию умножения пятимерных векторов $a\mathbf{e} + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} + g\mathbf{u}$ значением коэффициента растяжения $\varepsilon = 3048145277787$. Такое значение ε не может быть представлено в виде 5-й степени какого-либо другого элемента базового поля $GF(p)$, поэтому формируется векторное конечное поле $GF(p^5)$, генератором мультипликативной группы которого является вектор $G_\Omega = 2\mathbf{e} + 5\mathbf{i} + 7\mathbf{j} + 11\mathbf{k} + 13\mathbf{u}$.

4. Реализация функции коммутативного шифрования

Коммутативной называется функция шифрования $\mathbf{E}_K(M)$, где M – преобразуемое сообщение и K – ключ шифрования, для которой выполняется соотношение

$$\mathbf{E}_A(\mathbf{E}_B(M)) = \mathbf{E}_B(\mathbf{E}_A(M)),$$

где A и B – различные секретные ключи, например принадлежащие абонентам A и B, соответственно. Следующий протокол, известный под названием трехпроходный протокол Шамира (см. с. 516-517 в [1]), позволяет передать секретное сообщение по открытому каналу без того, чтобы отправитель и получатель обменивались какими-либо ключами. Протокол требует использования стойкой коммутативной функции шифрования и включает следующие шаги:

1. Абонент A шифрует сообщение m, получает шифртекст $C_1 = \mathbf{E}_A(M)$ и посылает C_1 абоненту B.
2. Абонент B зашифровывает сообщение C_1 (теперь сообщение M зашифровано дважды с использованием двух различных ключей), получает шифртекст $C_2 = \mathbf{E}_B(C_1) = \mathbf{E}_B(\mathbf{E}_A(M))$ и посылает c_2 абоненту A.
3. Абонент A, используя процедуру расшифрования $\mathbf{D}$, преобразует сообщение C_2, получает шифртекст $C_3 = \mathbf{D}_A(C_2) = \mathbf{D}_A(\mathbf{E}_B(\mathbf{E}_A(M))) = \mathbf{D}_A(\mathbf{E}_A(\mathbf{E}_B(M))) = \mathbf{E}_B(M)$ и посылает c_3 абоненту B.

Получив значение C_3, абонент B без труда восстанавливает сообщение $M = \mathbf{D}_B(\mathbf{E}_B(M))$. Этот протокол вообще не требует обмена ни секретными, ни открытыми ключами. Наиболее сложной проблемой является построение шифрующих преобразований, обладающих свойством коммутативности и обеспечивающих высокую криптостойкость этого протокола. Свойство коммутативности обеспечивается процедурой шифрования, заключающейся в наложении с помощью операции XOR ($\oplus$) на сообщение M ключа, длина которого равна длине M. Пусть ключи A и B являются случайными равновероятными ключами, тогда в отдельности каждая из процедур шифрования $C_A = M \oplus A$ и $C_B = M \oplus B$ обеспечивает абсолютную стойкость криптографического преобразования. Однако такой способ шифрования неприемлем в рассматриваемом протоколе. Действительно, в этом случае на шагах 1, 2 и 3 по открытому каналу пересылаются сообщения $C_1 = M \oplus A$; $C_2 = M \oplus A \oplus B$; $C_3 = M \oplus B$, а следовательно, потенциальный нарушитель может легко вычислить $M = C_1 \oplus C_2 \oplus C_3$.

Для реализации этого протокола пригоден метод шифрования Полинга-Хеллмана, использующий операцию возведения в большую дискретную степень по модулю большого простого числа p в качестве шифрующей процедуры. При этом при зашифровании и расшифровании осуществляется возведение в различную степень. Пусть зашифрование сообщения $M < p$ состоит в возведении в степень, т. е. значение шифртекста равно $C = M^e \pmod p$, тогда для правильного расшифрования нужно найти такую степень d, что будет выполняться условие $M = C^d = M^{ed} \pmod p$. Из теории чисел известно, что последнее условие справедливо для любого $M < p$, если имеет место условие $ed = 1 \pmod{p-1}$. Также известно, что если выбрать e взаимно простым с $p-1$, то для такого e существует и с помощью расширенного алгоритма Евклида легко находится соответствующее ему обратное (по модулю $p - 1$) число d, удовлетворяющее упомянутому выше условию.

Таким образом, приходим к стойкой реализации протокола «бесключевого шифрования», включающей передачу следующих значений C_1, C_2 и C_3:

$C_1 = m^{e_A} \pmod p$, где e_A есть ключ зашифрования абонента А;

$C_2 = C_1^{e_B} = m^{e_A e_B} \pmod p$, где e_B есть ключ зашифрования абонента В;

$C_3 = C_2^{d_A} = m^{e_A e_B d_A} = m^{e_B} \pmod p$, где d_A есть ключ расшифрования абонента А.

Получив шифртекст C_3, абонент В легко расшифровывает сообщение: $M = C_3^{d_B}$. Действительно, имеем $C_3^{d_B} = M^{e_B d_B} = M \pmod p$. В данном случае по значениям C_1, C_2 и C_3 нарушитель не может восстановить передаваемое сообщение. Например, нарушитель по значениям C_2 и $C_3 = C_2^{d_A} \pmod p$ может попытаться вычислить d_A и восстановить сообщение $M = C_1^{d_A} \pmod p$. Однако для этого ему придется решить задачу дискретного логарифмирования, что является вычислительно неосуществимым при правильном выборе простого числа p (разложение числа $p - 1$ должно содержать, по крайней мере, один большой простой множитель).

Алгоритм коммутативного шифрования можно реализовать, используя операцию возведения в степень в конечных полях различного типа. В алгоритме Полинга-Хеллмана используется простое конечное поле. Но аналогичным способом легко построить алгоритм коммутативного шифрования над конечным полем многочленов $GF(p^s)$ и векторным конечным полем $GF((p^s)^m)$ или векторным полем $GF(p^m)$, причем в последнем случае при параллельной реализации вычислений может быть достигнут значительный выигрыш в производительности (примерно в m раз) при заданном размере порядка поля.

Генерация ключей зашифрования и расшифрования осуществляется с учетом используемого конечного поля. Например, в случае векторного поля $GF(p^m)$ в качестве ключа зашифрования выбирается достаточно большое число e, удовлетворяющее условию НОД$(e, p^m - 1) = 1$ и вычисляется $d = e^{-1} \pmod{p^m - 1}$.

Сложность задачи дискретного логарифмирования в конечных полях зависит от размера порядка поля, причем эта зависимость имеет примерно одинаковый вид, что дает основание сравнивать случаи реализации алгоритма над конечными полями, у которых размер порядка одинаков, как варианты, обеспечивающие примерно одинаковую стойкость. В следующем разделе рассматривается оценка производительности алгоритмов коммутативного шифрования при использовании конечных полей различного типа.

5. Сравнительная оценка производительности

Сравним сложность операции умножения в векторном поле $GF(p^m)$ со сложностью умножения в простом поле $\mathbf{Z}_{p'}$, где $|p|$ обозначает битовую длину числа p и $|p'| = m|p|$. Последнее условие задает одинаковый размер порядков сравниваемых полей. Операция умножения элементов поля $GF(p^m)$ включает m^2 операций умножения в поле $GF(p)$, причем сложность операции умножения в поле $GF(p)$ пропорциональна $|p|^2$, поэтому при прямолинейном выполнении операции умножения в поле $GF(p^m)$, представленном в векторной форме, ее сложность примерно равна сложности умножения в поле $\mathbf{Z}_{p'}$. Имеющие место в случае векторного поля операции арифметического сложения и умножения на коэффициенты растяжения не учитываются, поскольку их вклад достаточно мал при выборе коэффициентов растяжения достаточно малого размера (два-три бита).

Однако в случае векторного поля имеется возможность снижения сложности умножения следующим образом. Осуществляются обычные арифметические операции умножения соответствующих пар координат векторов-сомножителей, результаты, соответствующие одинаковым базисным векторам, суммируются и только потом выполняется операция арифметического деления полученного результата на значение p. При этом число арифметических умножений остается равным m^2, а число делений уменьшается в m раз, становясь равным m. При этом сложность операции деления возрастает за счет увеличения делимого несущественно, так как размер последнего увеличивается всего лишь в m раз, т.е. его длина возрастает на несколько битов. Это не вносит существенного увеличения сложности операции деления в случае практически значимых размеров значений координат, которые определяются размерами модуля от $|p| = 16$ до $|p| = 200$ бит и значениями размерности векторов от $m = 13$ до $m = 3$, соответственно. Поскольку сложность операции деления значительно превосходит сложность операции умножения, то сложность операции умножения элементов поля $GF(p^m)$ снижается с возрастание значения m. Менее грубая оценка может быть получена, если принять конкретную модель вычислителя (электронного устройства, реализующего вычисления). Для модели вычислителя, в котором операция арифметического деления имеет временную сложность в k раз более высокую, чем арифметическое умножение, сложность умножения в простом поле равна $W_{GF(p')} \approx c(1+k) \cdot |p'|^2 \approx c(1+k) \cdot m^2 |p|^2$, где c – некоторый коэффициент пропорциональности, а сложность умножения в векторном поле равна $W_{GF(p^m)} \approx c(m^2 + km) \cdot |p|^2$. Отсюда имеем отношение

$$\rho = \frac{W_{GF(p')}}{W_{GF(p^m)}} \approx \frac{(1+k) \cdot m^2}{m^2 + km} = \frac{(1+k) \cdot m}{m+k}.$$

При достаточно больших значениях k и малых m имеет место $\rho \approx m$, а при сравнительно малых k и больших m – $\rho \approx k$.

При программной реализации операции умножения в простом поле снижение ее временной сложности может быть достигнуто путем реализации умножения по способу Монтгомери [9] (см. с. 299-303). Сложность умножения в простом поле, выполняемого по этому способу, примерно равна сложности трех арифметических умножений. При реализации умножения по Монтгомери имеем следующие оценки сложности: $W'_{GF(p')} \approx 3c|p'|^2 \approx 3cm^2 |p|^2$, $W'_{GF(p^m)} \approx 3cm^2 |p|^2$ и $\rho' \approx 1$, т. е. в этом случае временная сложность операции умножения в обоих видах полей примерно одинакова. Однако ввиду меньшего числа операций деления при выполнении умножения в векторном поле, имеет смысл сравнить значения сложностей $W'_{GF(p')}$ и $W_{GF(p^m)}$:

$$\rho'' = \frac{W'_{GF(p')}}{W_{GF(p^m)}} \approx \frac{3c \cdot m^2 |p|^2}{c(m^2 + km) \cdot |p|^2} = \frac{3m}{m+k}.$$

При $m \geq k/2$ имеет место соотношение $1 \leq \rho'' < 3$.

Рассмотрим сложность умножения в поле $GF(p^m)$, заданном в виде конечного кольца многочленов степени $m - 1$. Операция умножения двух многочленов включает m^2 операций арифметического умножения $|p|$-битовых чисел и примерно $2m$ операций деления $2|p|$-битовых чисел на модуль p (операциями сложения пренебрегаем ввиду их низкой сложности). В результате выполнения этих операций получаем многочлен степени $2m - 2$, который далее делится на неприводимый многочлен. Наличие этой операции не допускает эффективного распараллеливания операции умножения в поле многочленов. Наиболее эффективная реализация деления на неприводимый многочлен (в котором коэффициент при старшей степени переменной равен единице) требует выполнения не менее m операций арифметического умножения $|p|$-битовых чисел и m операций деления $2|p|$-битовых чисел на модуль p. Получаем следующую оценку сложности операции умножения $W^*_{GF(p^m)}$ в конечном поле многочленов:

$$W^*_{GF(p^m)} > c\left(m^2 + m + k(2m + m)\right) | p |^2 .$$

Для отношения сложностей умножения в поле многочленов и векторном поле получаем выражение

$$\rho^* = \frac{W^*_{GF(p^m)}}{W_{GF(p^m)}} > \frac{m^2 + m(3k+1)}{m^2 + km} = \frac{m + 3k + 1}{m + k} > 1 ,$$

т.е. умножение в поле многочленов является более сложным, чем умножение в векторном поле. Способ умножения Монтгомери может быть реализован также и в случае полей многочленов [10] (см. с. (с.214-216). Однако он дает существенное снижение сложности умножения в поле многочленов, заданном по модулю неприводимого многочлена с достаточно большим числом коэффициентов размера $|p|$, когда сложность операции деления на неприводимый многочлен в 4 и более раза превышает сложность операции арифметического умножения двух многочленов. Для этого случая выполняется соотношение $\rho^* >> 1$.

Таким образом, временная сложность операции умножения в векторном поле при различных вариантах ее реализации меньше сложности умножения в конечном простом поле и в поле многочленов. Переход к векторной форме задания расширенных конечных полей дает выигрыш в вычислительной эффективности *даже в случае использования однопроцессорного вычислительного устройства*. При этом операция умножения в векторном поле $GF(p^m)$ обладает возможностью эффективного распараллеливания на m процессов, поэтому увеличивая сложность аппаратной реализации, имеется возможность сокращения времени выполнения умножения в m раз. Распараллеливание может быть применено и для реализации умножения в конечном простом поле и в поле многочленов, однако это требует существенно большего количества затрачиваемых дополнительных аппаратных ресурсов, причем требуется реализовать нестандартные схемы вычислителей. При использовании стандартных многопроцессорных вычислителей параллельная реализация умножения в простых полях и полях многочленов не дает значительного сокращения времени, затрачиваемого на выполнение умножения.

6. Выбор характеристики векторного поля

Стойкость рассмотренного выше алгоритма коммутативного шифрования определяется сложностью задачи дискретного логарифмирования в используемом конечном поле. Эта задача является вычислительно сложной, если порядок поля делится на простое число большого размера. Кроме того, следует принять во внимание то, что в качестве основания для вычисляемого логарифма может оказаться произвольный элемент поля, так как шифруемое сообщение может иметь произвольное значение (предполагается, что на шифруемое сообщение накладывается единственное ограничение – его значение не должно выходить за границы множества значений элементов поля, с использованием которого реализуется алгоритм коммутативного шифрования). То есть с некоторой вероятностью сообщение будет соответствовать элементу поля, порядок которого является малым числом. В этом случае задача дискретного логарифмирования легко решается. Хотя решение не дает возможность получения значительной информации о секретном ключе, максимально возможное снижение вероятности появления таких сообщений является целесообразным. Последнее может быть достигнуто, если для данного значения размерности векторов m и размера характеристики поля $|p|$ выбрать значение характеристики p таким образом, что число $q = \frac{p^{m-1} + p^{m-2} + \ldots + p + 1}{m}$ является простым. Это возможно, если число m является простым [4,8]. В этом случае практически все возможные сообщения как элементы поля будут иметь порядок, содержащий в качестве своего делителя большое простое число q, имеющего размер $|q| \approx (m-1)|p|$. Пренебрежимо малое число сообщений M будет иметь порядок $\omega(M) \leq m(p-1)$. Вероятность случайного выбора сообщения имеющего малый порядок $\Pr(\omega \leq m(p-1))$ может быть вычислена, используя известное положение, что число элементов конечного поля $\psi(\omega)$, имеющие порядок ω, равно функции Эйлера от ω, т. е. $\psi(\omega) = \varphi(\omega)$:

$$\Pr\left(\omega \leq m(p-1)\right) = \left(p^m - 1\right)^{-1} \sum_{d_i | m(p-1)} \varphi(d_i) = \frac{m(p-1)}{p^m - 1} = \frac{m}{p^{m-1} + p^{m-2} + \ldots + p + 1} = \frac{1}{q}.$$

Для простых значений $m \geq 3$ при порядке векторного поля, имеющем размер не менее 1024 бит, значение вероятности рассматриваемого события не превышает значения $\approx 2^{-680}$. Такой вероятностью на практике можно пренебречь.

7. Заключение

Процедуры коммутативного шифрования, используемые в ряде практически значимых криптографических протоколах, имеют временную сложность, существенно зависящую от способа реализации операции умножения в используемом конечном поле и от типа применяемого вычислительного устройства. Для заданного вычислительного устройства и размера порядка поля в случае векторных конечных полей и полей многочленов можно подобрать соотношения между характеристикой и степенью расширения поля, при которых обеспечивается значительный выигрыш в производительности. Векторные конечные поля обеспечивают возможность наиболее простой параллельной реализации операции умножения в поле, благодаря тому, что в них отсутствует операция деления по модулю большого простого числа (в простых полях) или по модулю неприводимого многочлена большого размера (в полях многочленов). При использовании метода Монтгомери для реализации операции экспоненцирования в поле устраняется необходимость выполнения операции арифметического деления, причем в случае полей многочленов появляется возможность параллельной реализации умножения. Однако при этом увеличивается число операций умножения, а распараллеливание вычислений при выполнении умножения в поле многочленов не столь эффективно как в векторном конечном поле.

Работа поддержана грантом РФФИ № 08-07-00096-а.

Литература

1. *Schneier B.* Applied Cryptography: Protocols, Algorithms and Source Code (Second Edition) // New York: John Wiley & Sons. – 1996. – 758 p.
2. *Молдовян Н.А.* Введение в криптосистемы с открытым ключом. – СПб: БХВ – Петербург, 2007.-286 с.
3. *Hellman M.E., Pohling S.C.* Exponentiation Cryptographic Apparatus and Method // U.S. Patent # 4,424,414. 3 Jan. 1984.
4. *Молдовяну П.А., Дернова Е.С., Молдовян Д.Н.* Синтез конечных расширенных полей для криптографических приложений // Вопросы защиты информации. 2008. № 3(82). С. 2-7.
5. *Молдовян Н.А.* Алгоритмы аутентификации информации в АСУ на основе структур в конечных векторных пространствах // Автоматика и телемеханика. 2008. № 12. С.163-177.
6. *Moldovyan N.A.* Acceleration of the Elliptic Cryptography with Vector Finite Fields // International Journal of Network Security. 2009. V.9. No 2. P.180-185.

7. *Доронин С.Е., Молдовяну П.А., Синев В.Е.*. Векторные конечные поля: задание умножения векторов большой четной размерности // Вопросы защиты информации. 2008. № 4(83). С.2-7.
8. *Молдовян Д.Н., Молдовяну П.А.* Задание умножения в полях векторов большой размерности // Вопросы защиты информации. 2008. № 3(82). С. 12-17.
9. *Смарт Н.* Мир программирования. Криптография. – М.: Техносфера, 2005. – 525 с.
10. *Болотов А.А., Гашков С.Б., Фролов А.Б.,Часовских А.А.* Элементарное введение в эллиптическую криптографию. Алгебраические и алгоритмические основы. М., КомКнига, 2006.- 324 с.