УДК 004.4 (043.2)

Самофалов К.Г., чл.-кор. НАН Украины,
Марковский А.П., к.т.н.,
Шаршаков А.С.

СПОСОБ УСКОРЕННОЙ РЕАЛИЗАЦИИ ЭКСПОНЕНЦИРОВАНИЯ НА ПОЛЯХ ГАЛУА В СИСТЕМАХ ЗАЩИТЫ ИНФОРМАЦИИ

Национальный технический университет Украины "КПИ"

Предложен новый подход к повышению производительности систем защиты информации, основанных на арифметике конечных полей. Основываясь на представлении, экспоненты в форме произведения сумм разработан новый алгоритм экспоненцирования на полях Галуа, использующий распараллеливание и предвычисления. Показано, что предложенный подход позволяет увеличить скорость программной реализации экспоненцирования на полях Галуа в 3 раза и повысить производительность аппаратной реализации на 2 порядка.

Введение

Арифметические операции, выполняемые на полях Галуа, занимают важное место в современных информационных технологиях. В частности, они лежат в основе большинства методов обнаружения и коррекции ошибок передачи и хранения данных, широко используются при кодовом уплотнении передачи информации, в системах измерения и регистрации данных.

Особо важную роль играют вычисления на полях Галуа в современных криптографических механизмах защиты информации: они используются в алгоритме симметричного шифрования *Rijndael*, ставшем победителем всемирного конкурса *AES*, а также в механизмах ассиметричного шифрования и цифровой подписи на основе эллиптических кривых [1]. В криптографии на основе эллиптических кривых базовой операцией является экспоненцирование на полях Галуа, выполняемое над числами большой разрядности (512 – 1024 бит), превышающей длину машинного слова современных процессоров.

Существенной сложностью вычислительной реализации арифметики на полях Галуа является неприспособленность к ней архитектуры обычных процессорных средств, ориентированных на двоичную арифметику. Это обстоятельство диктует необходимость разработке специальных сложных программных средств, что резко замедляет выполнение арифметических операций на полях Галуа. Вместе с тем, анализ динамики развития прикладных задач, в которых активно используется арифметика на полях Галуа, показывает, что большая их часть выполняется в реальном времени и требует быстрой реализации соответствующих вычислений. Другой важной проблемой использования арифметики на полях Галуа на современном этапе развития информационных технологий является рост разрядности обрабатываемых чисел.

Это требует разработки новых методов организации вычислений на конечных полях в первую очередь, это касается наиболее трудоемких вычислительных операций, таких как экспоненцирование.

Таким образом, научная задача разработки новой организации экспоненцирования на конечных полях, выполняемого над числами большой разрядности, обеспечивающей карди-нальное уменьшение временной сложности вычисления экспоненты, является актуальной и практически важной для современного этапа развития информационных технологий.

Анализ способов экспоненцирования на полях Галуа

Операция экспоненцирования A^E на поле Галуа, задаваемым образующим его неразложимым полиномом $Q(x)$ степени m предполагает, что все его компоненты представляют собой m-разрядные двоичные коды: $A=\{a_0, a_1,...,a_{m-1}\}$, $\forall j \in \{0,...,m-$

1}: $a_j \in \{0,1\}$ и $E=\{e_0,e_1,...,e_{m-1}\}$, $e_j \in \{0,1\}$, которым соответствуют полиномиальные представления:

$P(A) = = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3 + ... + a_{m-1} \cdot x^{(m-1)}$ и $P(E) = = e_0 + e_1 \cdot x + e_2 \cdot x^2 + e_3 \cdot x^3 + ... + e_{m-1} \cdot x^{(m-1)}$.

К настоящему времени предложен ряд способов выполнения операции экспоненцирования на полях Галуа [2-4]. Их анализ показывает, что в качестве основного резерва повышения производительности их авторы рассматривают возможность распараллеливания выполнения базовой для экспоненцирования операции умножения на полях Галуа.

Сама процедура модулярного экспоненцирования A^E на полях Галуа, подобно обычному модулярному экспоненцированию, сводится к последовательному выполнению m циклов, в каждом из которых осуществляется операция возведения в квадрат полученного на предшествующем цикле результата и, дополнительно, в зависимости от текущего бита экспоненты E, – операция умножения. Исходя из порядка, в котором анализируются разряды экспоненты E, существует две разновидности модулярного экспоненцирования: справа налево и слева направо. На практике чаще применяется анализ разрядов экспоненты E начиная со старших разрядов, т.е. слева направо. Структурно данный способ организации вычисления экспоненты A^E на полях Галуа можно представлен на рис. 1.

Формально этот способ можно представить с помощью нотаций языка программирования C++:

1. $R = 1$;
2. for ($j = s$-1; j >=0; j--)
2.1. for ($l = k$-1; l >=0; l--)
{
2.1.1. $R = R \cdot R$;
2.1.2. if ($e_{j,l} = 1$) $R = R \cdot A$;
}

Результат $R=A^E$.

Проиллюстрируем работу базового способа экспоненцирования на полях Галуа следующим примером. Пусть разрядность обрабатываемых чисел равна 12:

A = 1000 1100 1101_2, E = 1000 1000 0010_2, а M = 1 0000 0101 0011_2.

Это соответствует образующему полиному

$Q(x) = x^{12} + x^6 + x^4 + x + 1)$.

Ход выполнения вычислений по базовому способу продемонстрирован в таблице 1.

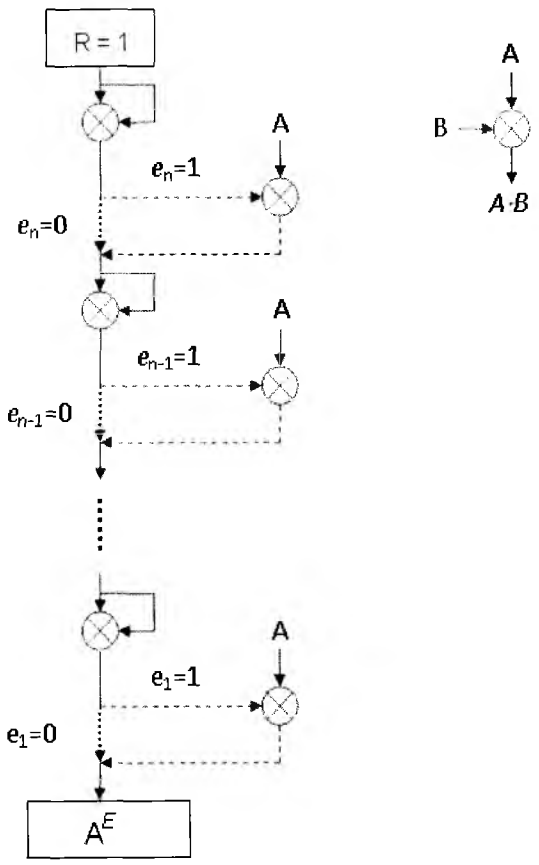

Рис. 1. Структура операции экспоненцирования

Очевидно, что среднее и максимальное число операций умножения на полях Галуа составляет соответственно $1.5 \cdot m$ и $2 \cdot m$. Каждая операция умножения [4] на поле Галуа включает m циклов, на каждом из которых выполняется: сдвиг множимого, если при этом $(m+1)$-й равен единице, то осуществляется прибавление к сдвинутому множимому кода Q, анализ значения текущего разряда множителя и, если он равен единице – прибавление сдвинутого множимо к сумме частичных произведений. Таким образом, операция

включает, в среднем, одну операцию сдвига и одну операцию суммирования, так, что вычислительная сложность умножения на поле Галуа составляет $O(2 \cdot m)$. Поскольку операции анализа разрядов множителя могут выполняться параллельно, то временная сложность умножения на поле Галуа составляет $O(\log_2 m)$.

Табл. 1. Пример вычисления экспоненты на поле Галуа по базовому алгоритму

j	l	$e_{j,l}$	операция			
2	3	1	R =	0000	0000	0001
			R = R • R =	0000	0000	0001
			R = R • A =	1000	1100	1101
2	2	0	R = R • R =	1001	0011	0010
2	1	0	R = R • A =	1000	0100	1000
2	0	0	R = R • R =	1000	0110	1111
1	3	1	R = R • R =	1100	0111	1010
			R = R • A =	1101	0100	1101
1	2	0	R = R • R =	1111	0110	0001
1	1	0	R = R • R =	1010	1001	0010
1	0	0	R = R • R =	1100	0110	0111
0	3	0	R = R • R =	1111	0011	1010
0	2	0	R = R • R =	1110	1000	0100
1	1	1	R = R • R =	1111	1010	1100
			R = R • A =	0011	0110	0100
0	0	0	R = R • R =	1010	1110	0000

Временная сложность экспоненцирования на полях Галуа, при последовательном анализе разрядов экспоненты $O(3 \cdot m^2)$, а временная сложность: $O(2 \cdot m \cdot \log_2 m)$.

Проведенный анализ свидетельствует о том, что основным фактором, ограничивающим время выполнения экспоненцирования на полях Галуа, является последовательный характер анализа разрядов кода экспоненты. Следовательно, для ускорения выполнения экспоненцирования необходимо исследовать возможности организации параллельной обработки разрядов экспоненты.

Способ экспоненцирования с использованием предвычислений

Теоретической основой предлагаемого способа экспоненцирования на полях Галуа является следующая теорема.

Теорема. Если код n экспоненты является степенью 2 ($n=2^l$), где l-целое, то есть, в полиномиальном представлении содержит одну компоненту $P(n) = x^{2^l}$, то результат экспонен - цирования $A = a_0 + a_1 \cdot 2^n + a_2 \cdot 2^{2 \cdot n} + a_3 \cdot 2^{3 \cdot n} + \ldots + a_{m-1} \cdot 2^{(m-1) \cdot n}$ на поле Галуа может быть представлен в виде суммы:

$$P(A^n) = a_0 + a_1 \cdot x^n + a_2 \cdot x^{2 \cdot n} + a_3 \cdot x^{3 \cdot n} + \ldots + a_{m-1} \cdot x^{(m-1) \cdot n} \quad (1)$$

Доказательство. Для доказательства используется метод математической индукции. При $n=1$ $P(A^1) = P(A) = a_0 + a_1 \cdot x + \ldots + a_{m-2} \cdot x^{m-2} + + a_{m-1} \cdot x^{m-1}$.

Предположим, что (1) выполняется для некоторого $n>1$ такого, что $n=2^l$: $P(A^n) = a_0 + + a_1 \cdot x^n + a_2 \cdot x^{2 \cdot n} + a_3 \cdot x^{3 \cdot n} + \ldots + a_{m-1} \cdot x^{(m-1) \cdot n}$.

Покажем, что в этом случае (1) выполняется и для следующего значения $2 \cdot n = 2^{l+1}$:

$P(A^{2 \cdot n}) = a_0 + + a_1 \cdot x^{2 \cdot n} + a_2 \cdot x^{2 \cdot 2 \cdot n} + a_3 \cdot x^{3 \cdot 2 \cdot n} + \ldots + a_{m-1} \cdot x^{(m-1) \cdot 2 \cdot n}$.

Для преобразования выполним умножение с учетом свойств сложения и умножения на полях Галуа:

$P(A^n \cdot A^n) = (a_0 + a_1 \cdot x^n + a_2 \cdot x^{2 \cdot n} + + a_3 \cdot x^{3 \cdot n} + \ldots + a_{m-1} \cdot x^{(m-1) \cdot n}) \cdot (a_0 + a_1 \cdot x^n + a_2 \cdot x^{2 \cdot n} + + a_3 \cdot x^{3 \cdot n} + \ldots + a_{m-1} \cdot x^{(m-1) \cdot n}) = a_0 + x^n \cdot (a_0 \cdot a_1 + a_1 \cdot a_0) + + x^{2 \cdot n} \cdot (a_0 \cdot a_2 + a_2 \cdot a_0 + a_1 \cdot a_1) + x^{3 \cdot n} \cdot (a_0 \cdot a_3 + a_3 \cdot a_0 + + a_1 \cdot a_2 + a_2 \cdot a_1) + \ldots + x^{((m-1) \cdot 2 - 1) \cdot n} \cdot (a_{m-2} \cdot a_{m-1} + + a_{m-1} \cdot a_{m-2}) + x^{(m-1) \cdot 2 \cdot n} \cdot a_{m-1} \cdot a_{m-1} = a_0 + a_1 \cdot x^{2 \cdot n} + + a_2 \cdot x^{2 \cdot 2 \cdot n} + a_3 \cdot x^{3 \cdot 2 \cdot n} + \ldots + a_{m-1} \cdot x^{(m-1) \cdot 2 \cdot n}$.

Что и требовалось доказать.

Применяя рассмотреную теорему к задаче экспоненцирования на полях Галуа, можно сформулировать способ, основанный на применении предвычислений. В системах защиты информации, использующих несимметричную криптографию на основе эллиптических кривых, экспонента E и образующий поле полином являются открытым ключом и меняются редко. Это позволяет считать код экспоненты E практически постоянным и использовать предвычисления для уменьшения вычислительной сложности экспоненцирования.

Идея предлагаемого способа экспоненцирования A^E на полях Галуа, где $E= e_0 + e_1 \cdot 2 + ... + + e_{m-2} \cdot 2^{m-2} + e_{m-1} \cdot 2^{m-1}$, состоит в том, что A^E вычисляется в виде произведения компонент, каждый из которых представляет собой экспоненту A с показателем, являющимся степенью 2:

$$A^E = A^{e_0} \cdot A^{e_1 \cdot 2} \cdot ... \cdot A^{e_{m-2} \cdot 2^{m-2}} \cdot A^{e_{m-1} \cdot 2^{m-1}} \quad (2)$$

Каждая i-тая из компонент (2), в соответствии с доказанной выше теоремой, вычисляет-ся в виде:

$$A^{e_i \cdot 2^i} = a_0 + a_1 \cdot 2^i + a_2 \cdot 2^{2 \cdot i} + \\ + a_3 \cdot 2^{3 \cdot i} + ... + a_{m-1} \cdot 2^{(m-1) \cdot i} \quad (3)$$

Значения 2^i , $2^{2 \cdot i}$, $2^{3 \cdot i}$,..., $2^{(m-1) \cdot i}$ (в полиномиальном представлении x^i, $x^{2 \cdot i}$, $x^{3 \cdot i}$,..., $x^{(m-1) \cdot i}$) на поле Галуа, заданном образующим полиномом, предлагается вычислять заранее и сохранять в табличной памяти: $T_i[1]=2^i$, $T_i[2]= 2^{2 \cdot i}$,...,$T_j[m-1]= =2^{(m-1) \cdot i}$. С учетом этого вычисление i-той компоненты (3) предлагается организовать в виде:

$$A^{e_i \cdot 2^i} = a_0 + a_1 \cdot T_i[1] + a_2 \cdot T_i[2] + \\ + ... + a_{m-1} \cdot T_i[m-1] \quad (4)$$

С учетом (4) экспоненцирование A^E на поле Галуа и при фиксированном значении E предлагается реализовать в виде:

$$A^E = \prod_{i=0}^{m-1} (a_0 + a_1 \cdot T_i[1] + a_2 \cdot T_i[2] + \\ + ... + a_{m-1} \cdot T_i[m-1])^{e_i}. \quad (5)$$

Достоинством реализации экспоненты A^E на полях Галуа в виде (5) является возможность независимого вычисления каждой из ее мультипликативных компонент.

Структурно предлагаемый способ организации вычисления экспоненты A^E на полях Галуа представлен на рис.2.

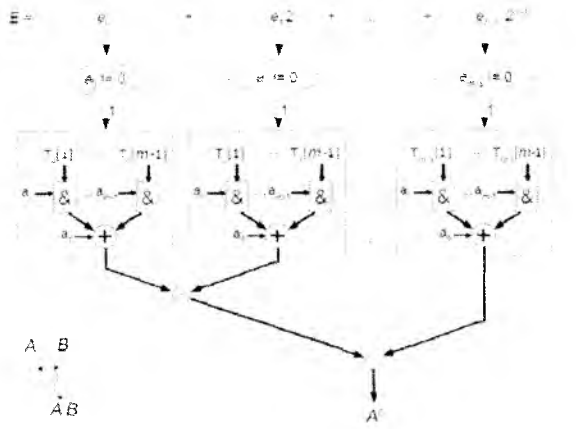

Рис. 2. Организация вычисления экспонентов

Предлагаемый способ предполагает осуществление предвычислений с сохранением результатов в табличной памяти.

При заполнении табличной памяти осуществляется вычисление m^2 значений вида $T_i[j] = 2^{j \cdot 2^i} \bmod M$, где $i, j = \overline{0, m-1}$. Таким образом, если двоичный код экспоненты E содержит $l < m$ единиц, то в табличной памяти реализуется хранение $l \cdot m$ m-разрядных чисел, так, что максимальный объем памяти составляет m^3 бит, а средний объем - $m^3/2$ бит.

При выполнении операции экспоненцирования на поле Галуа обработка каждого из двоичных разрядов $e_0,e_1,...,e_{m-1}$ кода экспоненты E осуществляется независимо.

Предложенный способ модулярного экспоненцирования на полях Галуа иллюстрируется следующим примером. Пусть разрядность обрабатываемых чисел равна 12.

$$A = 1000\ 1100\ 1101_2,$$
$$E = 1000\ 1000\ 0010_2,$$
$$M = 1\ 0000\ 0101\ 0011_2$$

(что соответствует образующему полиному $Q(x) = x^{12} + x^6 + x^4 + x + + 1$). Если считать, что код E, являющийся частью открытого ключа криптосистемы, постоянен, то для применения предложенного способа экспоненцирования на полях Галуа формируются три таблицы, соответствующие единичным битам кода экспоненты E.

Результаты сформированных в результате предвычислений таблиц T_1, T_7 и T_{11}, соответствующих единичным битам

кода экспоненты E приведены в таблице 2.

Табл. 2. Пример таблиц превычислений

i	T_1[i]	T_7[i]	T_{11}[i]
0	000000000001	000000000001	000000000001
1	000000000100	000100001011	000001001101
2	000000010000	010101110101	000000000010
3	000001000000	000110001011	000010011010
4	000100000000	001001101101	000000000100
5	010000000000	001100000110	000100110100
6	000001010011	010000111001	000000001000
7	000101001100	000001000011	001001101000
8	010100110000	000010010001	000000010000
9	010010010011	011011110000	010011010000
10	001000011111	000110110111	000000100000
11	100001111100	111100111100	100110100000

Поскольку единичными разрядами A являются a_0, a_2, a_3, a_6, a_7 и a_{11}, то, в соответствии с (4) :

$$A^{e_1 \cdot 2} = T_1[0]+T_1[2]+T_1[3]+T_1[6]+T_1[7]+T_1[11] =$$
$$= 000000000001_2 + 000000010000_2 +$$
$$+ 000001000000_2 + 000001010011_2 +$$
$$+ 000101001100_2 + 100001111100_2 =$$
$$= 001001100102$$

$$A^{e_7 \cdot 2^7} = T_7[0]+T_7[2]+T_7[3]+T_7[6]+T_7[7]+T_7[11] =$$
$$= 000000000001_2 + 010101110101_2 +$$
$$+ 000110001011_2 + 010000111001_2 +$$
$$+ 000001000011_2 + 111100111100_2 =$$
$$= 111110111001_2.$$

$$A^{e_{11} \cdot 2^{11}} = T_{11}[0]+T_{11}[2]+T_{11}[3]+T_{11}[6]+T_{11}[7]+T_{11}[11] =$$
$$= 000000000001_2 + 000000000010_2 +$$
$$+ 000010011010_2 + 000000001000_2 +$$
$$+ 001001101000_2 + 100110100000_2 =$$
$$= 101101011001_2.$$

В соответствии с (5):

$$A^E = (100100110010_2 \cdot 111110111001_2) \cdot$$
$$\cdot 101101011001_2 = 001000100101_2 \cdot$$
$$\cdot 101101011001_2 = 010111100000_2$$

Видно, что результат получен тот же, что и при вычислении по базовому способу.

Анализ базовой формулы (5) для вычисления экспоненты на полях Галуа показывает, что реализация предложенного способа требует $(l-1)$ операций умножения и, в среднем, $\frac{l \cdot m}{2}$ операций сложения. Принимая вычислительную сложность умножения на полях Галуа равной $O(2 \cdot m)$, вычислительная сложность экспоненцирования на поле Галуа по предложенному способу составляет $O(2 \cdot l \cdot m)$. С учетом того, что, в среднем, $l=m/2$, вычисление экспоненты по предложенному способу имеет вычислительная сложность $O(m^2)$, что в 3 раза меньше по сравнению с известными способами. Так, в рамках приведенного выше примера число умножений равно l-1=3, в то время, как экспоненцирование по известному способу требует 18-ти умножений. Уменьшение вычислительной сложности достигается за счет использования предвычислений для постоянного кода экспоненты E, что имеет место для криптографических применений.

Основным достоинством предложенного метода является возможность широкого распараллеливания. Действительно, анализ выражения (5) показывает, что каждый из l сомножителей может вычисляться одновременно, причем суммирование его компонент также может осуществляться параллельно. Временная сложность в этом случае определяется количеством операций умножения на критическом пути – $\log_2 l$. Учитывая, что временная сложность умножения - $O(\log_2 m)$, а среднее значение единичных битов экспоненты $l=m/2$, временная сложность экспоненцирования на полях Галуа предложенным способом составляет $O(\log_2{}^2 m)$. По сравнению с известными способами экспоненцирования на полях Галуа разработанный способ позволяет уменьшить временную сложность в h раз, где h определяется формулой:

$$h = \frac{2 \cdot m \cdot \log_2 m}{\log_2^2 m} = \frac{2 \cdot m}{\log_2 m}. \qquad (6)$$

Учитывая, большую разрядность m чисел, используемых в системах защиты информации при очевидной тенденции к ее дальнейшему росту, эффективность предложенного способа в плане ускоре-

ния вычисления экспоненты при аппаратной реализации представляется достаточно значительной. Например, для наиболее часто встречаемого в настоящее время значения m=1024 предложенный способ обеспечивает выигрыш во временной сложности по сравнению с известными более чем в 200 раз.

Выводы

Предложен новый способ экспоненцирования на полях Галуа, позволяющий существенно уменьшить время выполнения этой операции программными и аппаратными средствами. Основным источником уменьшения вычислительной сложности является использование результатов предвычислений, выполняемых один раз для постоянного кода экспоненты. Уменьшение временной сложности экспоненцирования достигнуто за счет новой организации вычислений, допускающей широкое распараллеливание.

В результате, предложенный способ позволяет в 3 раза ускорить экспоненцирование при программной реализации и ориентировочно на 2 порядка при аппаратной реализации. Разработанный способ ориентирован на применение в системах криптографической защиты информации с открытым ключом.

Список литературы

1. Menezes A.J., Blake I.F., Gao S., Mullin R.C., Vanstone S.A., Yactobian T. Application os Finite Fields // N.Y. Knuwer Academic Published. – 1993. – 387 p.

2. Стефанская В.А., Мухаммад Мефлех Алиса Абабне, Левчун Д.Ю. К проблеме повышения эффективности аппаратной реализации мультипликативных операций на полях Галуа // Вісник Національного технічного університету України "КПІ". Інформатика, управління та обчислювальна техніка – К.: "ВЕК++". – 2005. – № 43. – С. 104 – 112.

3. Popovici E.M., Fitzpatrick P. Algorithm and Architecture for a Galois Field Multiplicative Arithmetic Processor. // IEEE Transaction on Information theory. Vol. 49. - № 12. – 2003. – P. 3303 – 3307.

4. Wu H., Hasan M.A., Blake I.F., Gao S. Finite field multiplier using redundant representation.// IEEE Trans. Computers, Vol.51, № 51. – 2002. – P. 1306 – 1316.