УДК 519.7

МЕТОД ВЫПОЛНЕНИЯ АРИФМЕТИЧЕСКИХ ОПЕРАЦИЙ НАД ЧИСЛАМИ В КОНЕЧНЫХ ПОЛЯХ ХАРАКТЕРИСТИКИ 2 И ЕГО ПРИМЕНЕНИЕ В КРИПТОГРАФИИ

Н.С. Матвеев, А.Н. Марков

Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», Минск, Беларусь

Аннотация. В статье рассматриваются методы выполнения арифметических операций в конечных полях характеристики 2 и их применение в криптографии. Описаны основные операции – сложение и умножение, – а также методы их оптимизации. Особое внимание уделено практическим аспектам реализации данных операций и областям их применения.

Ключевые слова: конечные поля; характеристика 2; арифметические операции; криптография; постквантовая криптография; алгоритм; оптимизация вычислений; безопасность данных; двоичное представление; поля Галуа.

METHOD FOR COMPUTING NUMBERS IN FINITE FIELDS OF CHARACTERISTIC 2 AND ITS APPLICATION TO CRYPTOGRAPHY

N.S. Matsveyeu, A. N. Markov

Educational Institution "Belarusian State University of Informatics and Radioelectronics", Minsk, Belarus

Abstract. The article discusses methods for performing arithmetic operations in finite fields of characteristic 2 and their application in cryptography. The main operations – addition and multiplication – are described, along with methods for their optimization. Special attention is paid to the practical aspects of implementing these operations and their areas of application.

Keywords: finite fields; characteristic 2; arithmetic operations; cryptography; post-quantum cryptography; algorithm; computation optimization; data security; binary representation; Galois fields.

Введение

Конечным полем называется конечное множество, на котором определены произвольные операции, называемые сложением, умножением, вычитанием и делением (кроме деления на 0) в соответствии с аксиомами поля [1].

Конечные поля, или поля Галуа, играют важную роль в теории чисел, алгебраических структурах и их приложениях в информатике. Поля характеристики 2 представляют особый интерес, потому что их можно легко представить в виде двоичных строк. Арифметические операции в конечных полях характеристики 2, включая сложение, умножение и вычисление обратных элементов, лежат в основе многих криптографических алгоритмов. Примеры таких алгоритмов включают схемы симметричного и асимметричного шифрования, алгоритмы генерации цифровых подписей и криптографические протоколы на основе эллиптических кривых. Оптимизация выполнения этих операций является важной задачей для повышения производительности и безопасности криптографических систем.

В данной статье рассматриваются методы выполнения арифметических операций над элементами конечных полей характеристики 2, а также их применение в криптографии.

Основная часть

Сложение в конечном поле характеристики 2. В качестве примера рассмотрим конечное поле $GF(2^5)$. Будем считать, что неприводимый над $GF(2)$ многочлен $p(x)$ степени 5 мы уже построили. Пусть $p(x)=x^5+x^3+x^2+x+1$. Каждый элемент поля имеет вид $a_4\alpha^4+a_3\alpha^3+a_2\alpha^2+a_1\alpha+a_0$, где $\alpha\in GF(2^5)$ - корень многочлена $p(x)$, все $\alpha_i\in GF(2)$. Поэтому каждый такой элемент можно закодировать двоичной строкой $a_4a_3a_2a_1a_0$ для удобства хранения в ПК. В таком случае операция сложения многочленов будет представлять собой операцию XOR над соответствующими строками.

Умножение в конечном поле характеристики 2. Пусть $a=a_4\alpha^4+a_3\alpha^3+a_2\alpha^2+a_1\alpha+a_0$ и $b=b_4\alpha^4+b_3\alpha^3+b_2\alpha^2+b_1\alpha+b_0$ - элементы поля $GF(2^5)$. Умножим первый элемент поля на второй:

$$a\cdot b=((a_4\alpha^4+a_3\alpha^3+a_2\alpha^2+a_1\alpha+a_0)\cdot(b_4\alpha^4+b_3\alpha^3+b_2\alpha^2+b_1\alpha+b_0))\bmod p(\alpha). \quad (1)$$

После раскрытия скобок и перехода к двоичным строкам получим:

$$a\cdot b=(a\cdot b_4 0000+a\cdot b_3 000+a\cdot b_2 00+a\cdot b_1 0+a\cdot b_0)\bmod 101111. \quad (2)$$

Учитывая то, что переменная b может принимать только значения 0 или 1, последнее можно переписать в следующем виде:

$$a\cdot b=((10000\cdot a)\bmod 101111)\cdot b_4+((1000\cdot a)\bmod 101111)\cdot b_3+$$
$$+((100\cdot a)\bmod 101111)\cdot b_2+((10\cdot a)\bmod 101111)\cdot b_1+a\cdot b_0. \quad (3)$$

Теперь рассмотрим произвольный элемент поля $c\in GF(2^5)$, представленный двоичной строкой. Тогда $(c\cdot 10)\bmod 101111=c$, если $c\cdot 10<100000$, и $(c\cdot 10)\bmod 101111=c+101111$, если $c\cdot 10\geq 100000$. Тогда $(c\cdot 100)\bmod 101111$ можно представить следующим образом:

$$(c\cdot 100)\bmod 101111=(((c\cdot 10)\bmod 101111)\cdot 10)\bmod 101111. \quad (4)$$

Аналогичным образом раскладываются $c \cdot 1000$, $c \cdot 10000$ и т. д.

Теперь можно построить итеративный алгоритм, на каждой итерации которого текущее значение элемента a умножается на 10 (сдвиг влево) по модулю порождающего многочлена поля, полученный результат присваивается переменной a. После этого полученное значение умножается на разряд b, соответствующий текущей итерации и добавляется к переменной r, хранящей текущий результат.

Схема описанного алгоритма представлена на рис. 1.

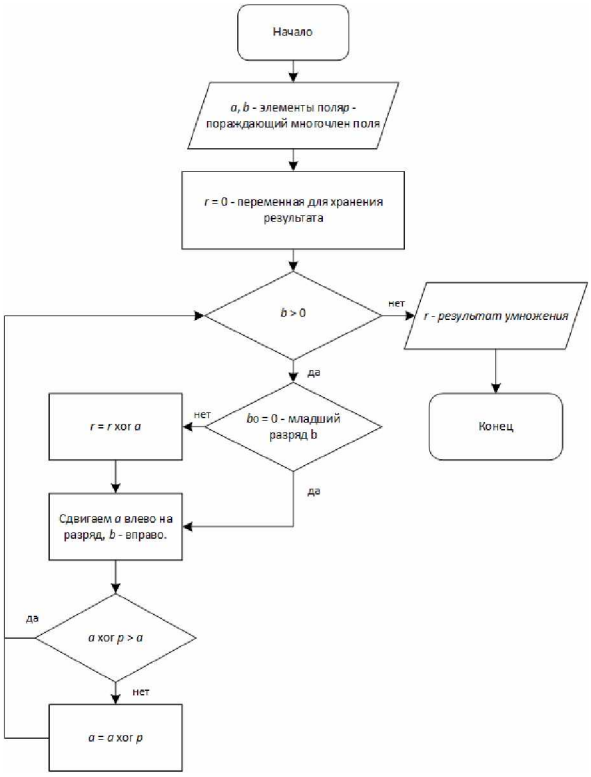

Рис. 1. Алгоритм умножения элементов конечного поля

Применение. Конечные поля применяются симметричных шифрах, таких как AES и Кузнечик, конечные поля (например, $GF(2^8)$) служат основой для построения нелинейных подстановок, которые обеспечивают высокую стойкость алгоритмов к различным атакам. Эти шифры получили признание на международном уровне и используются в стандартах, таких как ISO/IEC 18033-3:2010 для AES и ГОСТ Р 34.12-2015 для Кузнечика.

Код Рида–Соломона, построенный на основе арифметики конечных полей, нашел применение в оптических дисках, системах хранения данных, спутниковой связи и QR-кодах. Коды Гоппы, также основанные на конечных полях, используются для создания исправляющих кодов, способных обнаруживать и корректировать ошибки в каналах связи [2]. Их применение не ограничивается только коррекцией ошибок, они также интегрируются в криптографические протоколы, где необходима дополнительная защита и устойчивость системы

Конечные поля играют также ключевую роль в построении асимметричных и постквантовых криптосистем. Примером является криптосистема Мак-Элиса [3], в основе которой лежит задача декодирования случайных линейных кодов,

построенных с использованием конечных полей. Такие подходы позволяют создавать алгоритмы, устойчивые к атакам с использованием квантовых компьютеров, а также способствуют развитию новых направлений в постквантовой криптографии. Помимо этого, многие криптографические конструкции, в том числе алгоритмы на эллиптических кривых, используют конечные поля для задания математической структуры, необходимой для обеспечения безопасности и эффективности криптографических операций.

Кроме того, конечные поля находят применение в построении имитостойких схем и кодов аутентификации [4]. Здесь они обеспечивают математическую строгость, позволяя создавать оптимальные конструкции, способные обнаружить даже минимальные изменения в передаваемых данных. Это особенно важно для защиты информации от подмены или подделки. Также конечные поля используются в генераторах псевдослучайных чисел, что имеет значение для криптографических протоколов, где необходимы последовательности с хорошими статистическими свойствами. Теоретические исследования, опирающиеся на арифметику конечных полей, позволяют анализировать стойкость алгоритмов, строить доказательства их безопасности и выявлять потенциальные уязвимости.

Заключение

В данной работе рассмотрены методы выполнения базовых арифметических операций – сложения и умножения – в конечных полях характеристики 2, а также проанализированы области их применения. Конечные поля обеспечивают надежность симметричных шифров, каких как AES и Кузнечик с помощью нелинейных подстановок, эффективную коррекцию ошибок в системах хранения и передачи данных через коды Рида-Соломона и Гоппы, а также являются основой для асимметричных постквантовых криптосистем, таких как система Мак-Элиса и алгоритмы на эллиптических кривых. Дальнейшие исследования в данной области важны для повышения устойчивости информационных систем к современных угрозам.

Список использованных источников

1. Журавлёв Ю.И., Флеров Ю.Ф., Вялый М.Н. (2007) Дискретный анализ. Основы высшей алгебры. Москва, Издательство «МЗ Пресс».

2. Рацеев С.М. (2022) Элементы высшей алгебры и теории кодирования: учебное пособие для вузов. Санкт-Петербург, Издательство «Лань».

3. Dinh H., Moore C., Russell A. (2011) McEliece and Niederreiter Cryptosystems That Resist Quantum Fourier Sampling Attacks. Advances in Cryptology – CRYPTO 2011. (31), 761–779.

4. Рацеев С.М. (2022) Математические методы защиты информации: учебное пособие для вузов. Санкт-Петербург. Издательство «Лань».

References

1. Zhuravlyov Yu.I., Flerov Yu.F., Vyalyy M.N. (2007). Discrete Analysis: Fundamentals of Higher Algebra. Moscow, M3 Press Publiching House (in Russian).

2. Ratseev S.M. (2022) Elements of Higher Algebra and Coding Theory: A Textbook for Universities. Saint Petersburg, Lan' Publishing House (in Russian).

3. Dinh H., Moore C., Russell A. (2011) McEliece and Niederreiter Cryptosystems That Resist Quantum Fourier Sampling Attacks. Advances in Cryptology – CRYPTO 2011. (31), 761–779.

4. Ratseev S.M. (2022) Mathematical Methods of Information Protection: A Textbook for Universities. Saint Petersburg, Lan' Publishing House (in Russian).

Сведения об авторах

Матвеев Н.С., студент 4 курса факультета компьютерных систем и сетей, учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», nazarmatveev2003@gmail.com.
Марков А.Н., магистр технических наук, старший преподаватель кафедры информатики, учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», markov@bsuir.by.

Information about the authors

Matsveyeu N.S., fourth-year student of the Faculty of Computer Systems and Networks, Educational Institution "Belarusian State University of Informatics and Radioelectronics", nazarmatveev2003@gmail.com.
Markov A.N., Master of Technical Sciences, Senior Lecturer at the Department of Computer Science, Educational Institution "Belarusian State University of Informatics and Radioelectronics", markov@bsuir.by.