4-бөлім **Раздел 4** **Section 4**

Қолданылмалы математика **Прикладная математика** **Applied Mathematics**

МРНТИ 27.17.27; 27.41.41 DOI: https://doi.org/10.26577/JMMCS.2020.v107.i3.08

У.К. Турусбекова[1*], **А.С. Тургинбаева**[2]
[1]Казахский университет экономики, финансов и международной торговли, г. Нур-Султан, Казахстан
[2]Евразийский национальный университет имени Л.Н. Гумилева, г. Нур-Султан, Казахстан
*e-mail: umut.t@mail.ru

ХЕШИРОВАНИЕ НА ОСНОВЕ МНОГОЧЛЕНОВ

В современной криптографии широко используются различные хеш-функции. Хеш-функции - это простые для вычисления функции сжатия, которые принимают входные данные переменной длины и преобразуют их в выходные данные фиксированной длины. Они используются в качестве компактных представлений или цифровых отпечатков пальцев для обеспечения целостности сообщения. Основная проблема использования хеш-функций заключается в том, что существование необратимых функций, исключающих возможность столкновений, не доказано. Кроме того, не существует универсальных методов хеширования, и их следует выбирать в зависимости от области их применения. Особую роль играют теоретико-сложностные проблемы, а именно алгебраическая теория чисел. Одной из таких проблем является поиск неприводимых многочленов заданной степени над конечным полем, которые можно использовать для поиска хеш-кодов сообщений. Актуальность исследования неприводимых полиномов над простыми и расширенными полями Галуа обусловлена их разнообразным применением в различных областях науки и техники. Неприводимые многочлены нашли свое применение в различных областях математики, информационной техники и защите информации. Использование свойств неприводимых многочленов позволяет максимизировать эффективную компьютерную реализацию арифметики в конечных полях, что имеет особое значение для криптографии и теории кодирования. Поиск неприводимых многочленов является сложной для вычисления задачей, особенно над полями большой размерности. Процедура нахождения неприводимых многочленов требует эффективных алгоритмов и больших вычислительных ресурсов, как в случае нахождения простых чисел, что является основной проблемой для построения эффективных алгоритмов хеширования на их основе. В представленной статье описан метод построения хеш-функций, основанный на вычислении остатка от деления на неприводимый многочлен. Кроме того, рассмотрена проблема поиска неприводимых многочленов. Выполнено компьютерное моделирование хеш-функций с использованием неприводимых многочленов над конечными полями. Представлены результаты использования различных неприводимых многочленов и их анализ. Результаты статьи могут быть использованы в криптографических приложениях и теории кодирования.

Ключевые слова: неприводимый многочлен, Хеш-функция, конечное поле, избыточный циклический код, столкновение.

Ұ.Қ. Тұрысбекова[1*], А.С. Тургинбаева[2]
[1]Қазақ экономика, қаржы және халықаралық сауда университеті,
Нұр-Сұлтан қ., Қазақстан
[2]Л.Н. Гумилев атындағы Еуразия ұлттық университеті, Нұр-Сұлтан қ., Қазақстан
*e-mail: umut.t@mail.ru

Көпмүшеліктер негізінде хештеу

Қазіргі заманғы криптографияда әр түрлі хеш-функциялары кеңінен қолданылады. Хэш - функциялар - бұл өзгермелі ұзындықтағы кіріс қорын қабылдап, оларды тұрақты ұзындықтағы шығыс қорына түрлендіретін, есептеуге оңай сығымдау функциялары. Олар хабарламаның тұтастығын қамтамасыз ету үшін ықшам көріністер немесе сандық саусақ іздері ретінде қолданылады. Хеш-функцияларын қолданудағы негізгі мәселе соқтығысулар мүмкіндігін жоққа шығаратын қайтымсыз функциялардың болуының дәлелденбеуі болып табылады. Сонымен қатар, хештеудің әмбебап әдістері жоқ және оларды қолдану саласына қарай таңдаған жөн. Ерекше рөлді теориялық-күрделілік проблемалары, атап айтқанда алгебралық сандар теориясы атқарады. Осындай проблемалардың бірі ақырлы өрісте дәрежесі берілген келтірілмейтін көпмүшеліктерді іздеу болып табылады, оларды хабарламалардың хеш-кодтарын іздеуде қолдануға болады. Қарапайым және кеңейтілген Галуа өрістерінде келтірілмейтін көпмүшеліктерді зерттеудің өзектілігі олардың ғылым мен техниканың әр түрлі салаларында түрлі қолданылуымен байланысты. Келтірілмейтін көпмүшеліктер математиканың, ақпараттық технологияның және ақпараттық қауіпсіздіктің әр түрлі салаларында қолданыс тапты. Келтірілмейтін көпмүшеліктердің қасиеттерін қолдану арифметиканың ақырлы өрістерде компьютерлік тиімді іске асырылуын арттыруға мүмкіндік береді, ал бұл, өз кезегінде, криптография мен кодтау теориясы үшін ерекше маңызды. Келтірілмейтін көпмүшеліктерді табу есептеу үшін, әсіресе өлшемі үлкен өрістер үшін күрделі мәселе болып табылады. Келтірілмейтін көпмүшеліктерді іздеу процедурасы жай сандар жағдайындағы сияқты тиімді алгоритмдер мен үлкен есептеу қорларын қажет етеді, ал бұл, өз кезегінде, олардың негізінде тиімді хештеу алгоритмдерін құру үшін негізгі мәселелердің бірі болып табылады. ұсынылған мақалада хеш-функцияларды құрудың келтірілмейтін көпмүшелікке бөлгендегі қалдықты есептеуге негізделген әдісі сипатталған. Сонымен қатар, келтірілмейтін көпмүшеліктерді іздеу мәселесі қарастырылады. Ақырлы өрістерде келтірілмейтін көпмүшеліктерді қолдана отырып, хеш-функцияларды компьютерлік модельдеу жүргізілді. Әр түрлі келтірілмейтін көпмүшеліктерді қолдану нәтижелері және оларды талдау келтірілген. Мақаланың нәтижелерін криптографиялық қосымшалар мен кодтау теориясында қолдануға болады.

Түйін сөздер: келтірілмейтін көпмүшелік, хеш-функция, ақырлы өріс, артық циклдік код, соқтығысу.

U.K. Turusbekova[1*], A.S. Turginbayeva[2]
[1]Kazakh University of Economics, Finance and International Trade,
Nur-Sultan, Kazakhstan
[2]L.N. Gumilyov Eurasian National University,Nur-Sultan, Kazakhstan
*e-mail: umut.t@mail.ru

Hashing with polynomials

Hash functions are easy-to-compute compression functions that take a variable-length input and convert it to a fixed-length output. Hash functions are used as compact representations, or digital fingerprints, of data and to provide message integrity. In modern cryptography, various hash functions are widely used. The main problem with using hash functions is that the existence of irreversible functions that exclude the possibility of collisions has not been proven. In addition, there are no universal hashing methods, and they should be selected depending on their area of application. A special role is played by complexity-theoretical problems, namely, algebraic number theory. One of these problems is the search for irreducible polynomials of a given degree over a finite field, which can be used to search for message hash codes. The relevance of the study of irreducible polynomials over simple and extended Galois fields is due to their diverse

application in various fields of science and technology. Irreducible polynomials have found their application in various fields of mathematics, information technology and information security. Using the properties of irreducible polynomials allows you to maximize the effective computer implementation of arithmetic in finite fields, which is of particular importance for cryptography and coding theory. Finding irreducible polynomials is difficult to compute, especially over large fields. The procedure for finding irreducible polynomials requires efficient algorithms and large computational resources, as in the case of finding prime numbers, which is the main problem for constructing effective hashing algorithms based on them. This article describes a method for constructing hash functions based on calculating the remainder of a division by irreducible polynomials. In addition, the problem of searching for irreducible polynomials is considered. Computer modeling of hash functions using irreducible polynomials over finite fields has been performed. The results of using various irreducible polynomials and their analysis are presented. The results of the article can be used in cryptographic applications and coding theory.

Key words: irreducible polynomial, Hash function, finite field, redundant cyclic code, collision.

1 Введение

Системы информационных технологий требуют наличия эффективных инструментов, которые позволили бы значительно сократить объём памяти, необходимый для хранения и передачи больших объёмов данных, доступных ограниченному числу пользователей, и для проверки их целостности. Это также связано с тем, что финансовые операции с денежными средствами и хранение личных данных пользователей осуществляются в Интернете. Для таких целей широко используются хеш-функции. Они преобразуют исходные данные произвольной длины в последовательность фиксированной длины, называемую хеш-кодом или сверткой сообщения.

Хеш-функции целесообразно применять к ценным конфиденциальным данным, доступ к которым могут получить только определенные лица. Такие данные чаще всего представлены в виде текста или последовательности символов. Следует отметить, что при незначительных изменениях во входных данных результат хеш-функции должен полностью измениться, то есть иметь лавинный эффект, чтобы гарантировать, что данные не могут быть фальсифицированы незначительными изменениями. Эта хеш-функция также позволяет использовать их в следующих случаях: для поиска дубликатов в наборе данных; построение ассоциативных массивов; расчет контрольных сумм для последующего выявления и исправления ошибок, возникших при передаче или хранении данных; разработка электронной цифровой подписи; для сохранения паролей в базах данных.

Разработка качественной хеш-функции является сложной задачей. При разработке алгоритмов хеширования следует учитывать уязвимость хеш-функций. Степени набора входных последовательностей и множества всех возможных значений хеш-функции можно найти в любом соотношении. Как правило, набор входных данных имеет большую размерность, чем количество всех возможных значений функции, что приводит к преобразованию различных сообщений в один хеш. Такой случай называется «столкновением» (или «коллизией») и является одним из важных факторов, который учитывается при построении алгоритмов хеширования в криптографических системах [1], а также во многих структурах данных, таких как хеш-таблицы [2].

На данном этапе развития теории хеширования до сих пор нет четкого определения понятия хеш-функции и точных требований к их построению. Общие требования,

которым должны соответствовать хеш-функции, это – необратимость (невозможно создать алгоритм с полиномиальной вычислительной сложностью, который восстанавливает исходные данные в реальном времени), устойчивость к столкновениям, высокая скорость вычислений и наличие лавинного эффекта (с небольшим изменением во входных данных результат должен существенно отличаться). В зависимости от приложения к хеш-функции предъявляются дополнительные требования, такие как сложность вычислений, длина свертки и криптографическая стабильность.

Основная проблема использования хеш-функций заключается в том, что существование необратимых функций, исключающих возможность столкновений, не доказано. Кроме того, не существует универсальных методов хеширования, и их следует выбирать в зависимости от области их применения. На практике используются функции, для которых теоретическая вероятность столкновений близка к нулю, но с появлением более мощных вычислительных устройств поиск столкновений может оказаться не такой сложной задачей. По этой причине существующие алгоритмы требуют постоянного улучшения. Особую роль играют теоретико-сложностные проблемы, а именно алгебраическая теория чисел [3]. Одной из таких проблем является поиск неприводимых многочленов заданной степени над полем k_p или $Gk(p)$, которые можно использовать для поиска хеш-кодов сообщений. В статье рассматривается проблема поиска неприводимых многочленов, а также метод хеширования, основанный на вычислении остатка от деления на неприводимый многочлен.

2 Обзор литературы

В 1976 году Диффи и Хеллман впервые подчеркнули необходимость построения однонаправленной функции как составной части схемы цифровой подписи [4]. Этот год можно считать отправной точкой развития хеш-функций. Хеш-функции используются в качестве строительного блока во многих приложениях. Некоторые хэш-функции, используемые в настоящее время, оказались уязвимыми. В работе [5] автор утверждает, что их замены должны основываться на математической теории. В работе [6] исследованы потенциальные математические принципы и структуры, которые могут обеспечить основу для криптографических хеш-функций, а также представить простую и эффективно вычисляемую хеш-функцию, основанную на неассоциативной операции с многочленами над конечным полем характеристики 2. Общий обзор хеш-функций приведен в работе [7]. В работе [8] в хронологическом порядке развития приведены основные принципы построения алгоритмов хеширования. Отметим работу [9], в которой предложен способ усложнения поиска коллизий хеш-функций методом рандомизации входных данных для функции сжатия. Такой способ позволяет замаскировать коллизии в функции сжатия. Способ позиционируется авторами как отдельный режим работы криптосистемы хеширования без изменения самой ее конструкции. Может быть полезен в цифровых подписях для предотвращения сценария атаки нахождения второго прообраза.

Для изучения методов хеширования, основанных на использование деления по модулю неприводимого многочлена, мы развиваем идеи работы [3]. Кроме того, рассматриваем проблему поиска неприводимых многочленов и их анализ.

3 Материал и методы

3.1 Неприводимые многочлены над конечными полями

Такие разделы алгебры, как теория конечных полей и теория многочленов над конечными полями, все больше влияют на построение различных систем защиты информации, кодирования и декодирования информации. В частности, появились алгоритмы для циклических избыточных кодов [10], которые используют многочлены над полями k_p. Циклические избыточные коды могут использоваться в качестве хеш-функций для обнаружения ошибок и проверки целостности данных.

Поскольку конечное поле является множеством с конечным числом элементов, операции сложения, вычитания, умножения и деления могут выполняться в соответствии с аксиомами поля [11]. Так как конечные поля являются замкнутыми относительно вышеупомянутых операций, то для любых двух элементов поля $a, b \in k_p$, при выполнении любой из операций, результатом является элемент $c \in k_p$, принадлежащий этому полю. Следует иметь в виду, что все вычисления в конечных полях производятся по модулю p, который является характеристикой конечного поля и является простым числом.

Простейшим примером конечного поля является кольцо классов вычетов $Z/(p)$ по модулю простого числа p, которое можно отождествить с полем Галуа $k_p = Gk(p)$ порядка p [6]. Согласно теореме о существовании и единственности конечных полей для любого простого числа p и натурального числа n существует конечное поле из p^n элементов. Чтобы построить поле k_{p^n}, необходимо найти многочлен $S(x)$ степени n, неприводимый над полем k_p. Такое поле представлено многочленами над k_p степенью не выше $n-1$.

В компьютерной криптографии многочлены, особенно неприводимые многочлены, играют значительную роль в последние два десятилетия. Напомним, что *неприводимый многочлен* – это многочлен, который не разлагается на нетривиальные многочлены и является аналогом простых чисел в натуральном ряду. Особенностью неприводимых многочленов является то, что, будучи неприводимым в одной области, многочлен оказывается приводимым в другой области, что нашло применение в теории кодирования и системах защиты информации.

Поиск неприводимых многочленов является сложной для вычисления задачей, особенно над полями большой размерности. Процедура нахождения неприводимых многочленов требует эффективных алгоритмов и больших вычислительных ресурсов, как в случае нахождения простых чисел [12], что является основной проблемой для построения эффективных алгоритмов хеширования на их основе. На данный момент нет эффективных алгоритмов поиска неприводимых многочленов, есть только критерии неприводимости и методы проверки неприводимости. Поиск осуществляется путем изучения мульти-тел и проверки каждого из них на неприводимость. Для проверки многочлена $S(x)$ степени $n \geq 2$ на неприводимость над полем характеристики p существует следующий алгоритм [13], [14]:

1. Инициализируется начальное значение многочлена $G_0(x) = x$.
2. Рассчитывается следующее значение $G_1(x) = G_0(x)^p mod S(x)$.
3. Рассчитывается наибольший общий делитель (НОД) многочленов $S(x)$ и $(G_1(x) - x)$. Если НОД не равен единице, то этот многочлен приводим. В противном случае,

следующее значение рассчитывается по формуле повторения

$$G_i(x) = G_{i-1}(x)^p mod S(x),$$

где $i = \overline{1, \lfloor n/2 \rfloor}$, $\lfloor \rfloor$ – операция взятия целой части числа.

4. Если НОД $S(x)$ и каждого $(G_i(x) - x)$ равен единице, то многочлен $S(x)$ – неприводим.

Недостатком такого алгоритма является низкая скорость вычислений для достаточно больших значений, поскольку на каждом шаге выполняется операция увеличения и нахождения НОД.

Для вычислений в конечных полях используется полиномиальная арифметика. Сложение в поле k_{p^n} соответствует обычному сложению многочленов по модулю p. Умножение выполняется в два этапа – сначала как простое умножение многочленов, а затем вычисляется остаток от деления на неприводимый многочлен, с помощью которого строится поле k_{p^n}. Например, поля одной и той же размерности могут быть построены по-разному, в зависимости от выбора неприводимого многочлена. Они одного порядка и изоморфны друг другу. Это следует из того факта, что для характеристики поля имеется несколько неприводимых многочленов степени n. Примеры неприводимых многочленов для поля k_2 приведены ниже [15]:

$n = 2$ $\quad x^2 + x + 1;$

$n = 3$ $\quad x^3 + x^2 + 1, \;\; x^3 + x + 1;$

$n = 4$ $\quad x^4 + x^3 + x^2 + x + 1, \;\; x^4 + x^3 + 1, \;\; x^4 + x + 1;$

$n = 5$ $\quad x^5 + x^2 + 1, \;\; x^5 + x^3 + x^2 + x + 1, \;\; x^5 + x^4 + x^3 + x + 1,$

$x^5 + x^4 + x^3 + x^2 + 1, \;\; x^5 + x^4 + x^2 + x + 1.$

3.2 Моделирование хеш-функций на основе неприводимых многочленов

В последние два десятилетия значительную роль в компьютерной криптографии играют многочлены, особенно неприводимые многочлены. Одним из возможных способов построения хеш-функции является использование деления по модулю неприводимого многочлена [3]. Для эффективности компьютерной реализации удобно использовать вычисления в полях k_{2^r}. Это позволяет производить расчеты по данным в виде последовательности битов. Поиск оставшейся части деления осуществляется с использованием побитовых сдвигов и разделительной дизъюнкции.

Для хеширования данные кодируются некоторым выбранным способом в последовательности $a_1, a_2, \ldots, a_m$ нулей и единиц, соответствующих определенному многочлену $A(x)$, а хеш-код $h(a_1, a_2, \ldots, a_m)$ представляет собой последовательность битов, полученных делением на неприводимый многочлен $S(x)$, и вычисляется по следующим формулам:

$$B(x) = A(x) mod S(x) \tag{1}$$

$$h(a_1, a_2, \ldots, a_m) = b_n b_{n-1} \ldots b_1 b_0 \quad (2)$$

В формуле (2) $b_n b_{n-1} \ldots b_1 b_0$ – это коэффициенты многочлена $B(x)$, полученные как остаток от деления многочлена $A(x) = a_1 x^{m-1} + a_2 x^{m-2} + \cdots + a_{m-1} x + a_m$ на многочлен $S(x) = s_n x^n + s_{n-1} x^{n-1} + \cdots + s_1 x + s_0$ степени n.

Такая функция устойчива к восстановлению исходных данных, поскольку, даже зная размерность поля и используемый неприводимый многочлен, трудно расшифровать данные, особенно для больших степеней неприводимого многочлена. Неприводимые многочлены следует выбирать на основе области действия хеш-функций, так как длина свертки равна степени многочлена. Итак, для применения в системах защиты информации на данный момент оптимальная длина составляет не менее 128 бит и не более 512 бит. Использование многочлена достаточной размерности играет существенную роль. Если выбран многочлен степени l, то множество всех возможных значений, которые может принять свертка функции, равно 2^l. Например, при использовании неприводимого многочлена $S(x) = x^4 + x + 1$ количество всех возможных пакетов будет 16, и поиск сообщений с одинаковыми свертками не составит труда.

4 Результаты и обсуждение

Компьютерное моделирование хеш-функций проводилось на основе неприводимых многочленов степени 32 с разным количеством одночленов. В результате анализ эффективности хеширования проводится с использованием каждого из многочленов. Важным фактором при выборе неприводимого многочлена является количество в нем одночленов. Для более высокой скорости вычислений желательно найти многочлены с минимальным количеством одночленов. Для сравнения результатов хеширования были выбраны многочлены с 5, 12 и 18 одночленами. Скорости вычислений для сверток входных данных различной длины и использования различных неприводимых многочленов одинаковой степени приведены в таблице 1. Неприводимые многочлены были записаны в двоичном представлении, представляющем собой последовательность коэффициентов при одночленах. Коэффициент наибольшей степени здесь не учитывается. Например, для многочлена $S(x) = x^4 + x + 1$ будет верным равенство $x^4 = x + 1$, так как операция вычитания аналогична добавлению по модулю 2. Напишем многочлен в виде $x^4 = 0 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x^1 + 1 \cdot x^0$, поэтому в двоичном представлении многочлен будет записан как 0011, а количество бит для его записи равно степени многочлена.

На первый взгляд разница в скорости вычислений невелика, но, при обработке объемов данных от 1 МБ или более, разница между вычислениями может составлять один час или более. Этот метод хеширования эффективен для применения небольших объемов данных в пределах нескольких килобайт. Таблица 2 показывает результаты хеширования неприводимыми многочленами из таблицы 1 для произвольной 64-битной строки и для той же строки с небольшими изменениями для проверки лавинного эффекта и свойства смешения.

Функции, реализованные на основе рассматриваемых многочленов, обладают свойством перемешивания. Это означает, что нет никакой связи между сверткой и исходными данными. Поскольку при незначительном изменении исходных данных результат

Таблица 1 – Скорость расчета свертки с использованием неприводимых многочленов степени 32

№	Неприводимые многочлены	Длина последовательности, бит		
		64	256	512
1	00000000010000000000000000000111	1,1	1,8	2,2
2	10000001010000010100000110101011	1,9	2,1	2,5
3	01110100000110111000110011010111	2,1	3,6	4,9

Таблица 2 – Результаты применения хеш-функций на основе неприводимых многочленов степени 32

Исходная битовая последовательность	Неприводимые многочлены		
	1	2	3
0110001100111001	11010011	01000110	01010111
0001110000011111	11101101	00001100	11011111
1110000011001000	01000000	01101101	00101111
0110110100011111	00001110	01100110	01001100
0110001100111001	11010000	01010010	11110010
0010110000011111	11101101	00000101	00010011
1110000011001000	11010000	00100010	11101010
0110110100011111	00101010	10110110	01001010

хеширования должен значительно измениться, в исходной битовой последовательности 19-й и 20-й биты были изменены для проверки соответствия этому свойству. На основании результатов, приведенных в таблице 2, наилучшим лавинным эффектом обладают функции, основанные на многочленах с большим количеством одночленов. Стоит отметить, что, несмотря на небольшую степень приведенных выше многочленов, хеш-функции на их основе имеют некоторое сопротивление столкновениям. При сортировке пачек, полученных обработкой данных в объеме 1000, 5000, 10000 и 30000, коллизий обнаружено не было, хотя это не может гарантировать их отсутствие на больших объемах.

Рассматриваемый метод хеширования подходит для битовых последовательностей, которые могут быть представлены многочленом более высокой степени, чем степень выбранного неприводимого многочлена. Меньшие последовательности должны быть дополнены функцией. В большинстве существующих алгоритмов хеширования добавление к требуемой длине выполняется путем добавления к последовательности одного бита и битов нулей. Кроме того, желательно добавить в последовательность ее первоначальную длину, что уменьшит вероятность столкновения после добавления. Использование остатка от деления на неприводимый многочлен может служить отдельной хеш-функцией и используется в сочетании с другими алгоритмами для улучшения определенных свойств. Также хеш, найденный этим методом, может быть использован в качестве криптографической соли.

5 Заключение

Теория конечных полей может быть использована для построения хеш-функций, но наряду с ее применением возникают проблемы, которые требуют отдельного исследования для дальнейших решений. Одной из таких проблем является нахождение неприводимых многочленов с определенными свойствами. Для полей, характеризующих простые числа большой разрядности, задача поиска неприводимых многочленов определенных степеней значительно сложнее и требует больших вычислительных затрат.

Было показано, что целесообразно использовать неприводимые многочлены достаточно больших степеней. Многочлены, состоящие из небольшого числа одночленов, позволяют находить свертки для меньшего числа операций. Однако многочлены с большим числом одночленов улучшают лавинный эффект хеш-функции. Оба имеют одинаковое сопротивление столкновениям. Неприводимый многочлен должен быть выбран на основе требуемых свойств хеш-функции. Чтобы усилить криптографическую стойкость и улучшить лавинный эффект, необходимо выбрать неприводимые многочлены степени 128 и выше с максимально возможным количеством одночленов. В случаях, когда хеш-функция используется в системах, требующих высокой скорости вычислений, рекомендуется использовать неприводимые многочлены с минимальным количеством одночленов.

Основным недостатком хеш-функций, основанных на неприводимых многочленах, является низкая скорость вычислений для больших объемов данных. Помимо расширений рассматриваемого поля, для увеличения устойчивости к столкновениям необходимо учитывать поля больших характеристик, что является задачей для дальнейшего решения. Это позволит хешировать данные в элементы из большего поля, но следует учитывать, что это усложнит компьютерные операции на компьютере.

Список литературы

[1] Шнайер Б. Прикладная криптография: протоколы, алгоритмы, исходные тексты на Си / пер. с англ.; под ред. Н. Дубновой. // Изд. 2-е.- М.: Диалектика. - 2003. – 610 с.

[2] Sedgewick R. Algorithms in C++, Parts 1-4 // Fundamentals, Data Structure, Sorting, Searching. - 3rd ed.- 1988.-752 p.

[3] Хомич Э.А. Неприводимые многочлены над конечными полями и связь с криптографией // Academic Publicistics.- 2017.-№3.- С.19-22.

[4] Whitfield Diffiee, Martin E. Hellman. New directions in cryptography // IEEE Trans. on Information Theory, Vol. IT-22, No. 6.- 1976. - P.644-654.

[5] Landau S. Find Me a Hash // Notices Amer. Math. Soc. – 2006. - V.53. – P. 330-332.

[6] Shpilrain V. Hashing with Polynomials // Information Security and Cryptology – ICISC 2006. - LNCS 4296. - Springer, 2006. - P. 22-28. DOI: https://doi.org/10.1007/11927587_4

[7] Menezes A., P. van Oorschot, Vanstone S. Handbook of Applied Cryptography// CRC Press. - 1997.

[8] Авезова Я.Э. Современные подходы к построению хеш-функций на примере финалистов конкурса SHA-3 // Вопросы кибербезопасности.- 2015.- №3(11).- С.60-67.

[9] Shai Halevi, Hugo Krawczyk. Strengthening Digital Signatures via Randomized Hashing // Advances in Cryptology - CRYPTO - LNCS 4117. - Springer, 2006. - P.41-59. DOI: https://doi.org/10.1007/11818175_3

[10] Henry S. Warren, Jr. Hacker's Delight. - 3rd ed.// Addison Wesley. - 2013.- 816 p.

[11] Лидл Р., Нидеррайтер Х. Конечные поля. - в 2 т. // пер. с англ.; под ред. В.И. Нечаева. // М.: Мир. - 1988.- Т.1.- 430 с.

[12] Turusbekova U.K., Azieva G.T. Investigation of irreducible normal polynomials special type over a field of characteristic 2 // Вестник КазНПУ им. Абая, серия "Физико-математические науки2019.- №3(67).-С.122-127.

[13] Crandall R. E., Pomerance C. B. Prime Numbers: A Computational Perspective. // New York: Springer-Verlag. - 2005.- 597 p.

[14] Горбенко И.Д., Штанько И.А. Функции хеширования. Понятия, требования, классификация, свойства и применение // Радиоэлектроника и информатика. - 1998. - №1.-С.64-69.

[15] Sankhanil Dey, Amlan Chakrabarti, Ranjan Ghosh. 4-bit crypto S-boxes: Generation with irreducible polynomials over Galois field GF(24) and cryptanalysis // International Journal of Tomography and Simulation. – 2019.- Vol. 32, no. 3.- P.46-60.

References

[1] Shcneier B., *Applied Cryptography, Second Edition: Protocols, Algorthms, and Source Code in C* , (1995): 784.

[2] Sedgewick R., *Algorithms in C++ , Parts 1-4: Fundamentals, Data Structure, Sorting, Searching* - 3rd ed. (1988): 752.

[3] Khomich E.A., "Neprivodimyye mnogochleny nad konechnymi polyami i svyaz' s kriptografiyey [Irreducible polynomials over finite fields and connection with cryptography]" , *Academic Publicistics*, Vol.3 (2017): 19-22.

[4] Whitfield Diffiee and Martin E. Hellman, "New directions in cryptography" , *IEEE Trans. on Information Theory* Vol. IT-22, no. 6.(1976):644-654.

[5] Landau S., "Find Me a Hash" , *Notices Amer. Math. Soc.*, vol.53 (2006): 330-332.

[6] Shpilrain V., "Hashing with Polynomials" , *Information Security and Cryptology – ICISC 2006* LNCS 4296 (2006): 22-28. Springer, 2006. DOI: https://doi.org/10.1007/11927587_4

[7] Menezes A., P. van Oorschot and S Vanstone, "Handbook of Applied Cryptography" , *CRC Press* (1997).

[8] Avezova YA.E., "Sovremennyye podkhody k postroyeniyu khesh-funktsiy na primere finalistov konkursa SHA-3 [Modern approaches to the construction of hash functions using the example of the finalists of the SHA-3 contest]" , *Voprosy kiberbezopasnosti*, no. 3(11) (2015): 60-67.

[9] Shai Halevi and Hugo Krawczyk, "Strengthening Digital Signatures via Randomized Hashing" , *Advances in Cryptology-CRYPTO 2006,* LNCS 4117 (2006): 41-59. Springer, 2006. DOI: https://doi.org/10.1007/11818175_3

[10] Henry S. and Warren, Jr., *Hacker's Delight* , - 3rd ed., Addison Wesley (2013): 816.

[11] Lidl R. and Niderrayter H., *Finite Fields* , Cambridge: Cambridge University Press (2000): 768.

[12] Turusbekova U.K. and Azieva G.T., "Investigation of irreducible normal polynomials special type over a field of characteristic 2" , *Vestnik KazNPU im. Abaya, seriya "Fiziko-matematicheskiye nauki"* no 3(67) (2019): 122-127.

[13] Crandall R. E. and Pomerance C. B., *Prime Numbers: A Computational Perspective* , New York: Springer-Verlag (2005): 597.

[14] Gorbenko I.D. and Shtan'ko I.A., "Funktsii kheshirovaniya. Ponyatiya, trebovaniya, klassifikatsiya, svoystva i primeneniye [Hash Functions Concepts, requirements, classification, properties and application]" , *Radioelektronika i informatika*, no. 1 (1998): 64-69.

[15] Sankhanil Dey, Amlan Chakrabarti and Ranjan Ghosh, "4-bit crypto S-boxes: Generation with irreducible polynomials over Galois field GF(24) and cryptanalysis" , *International Journal of Tomography and Simulation*, vol. 32, no. 3 (2019): 46-60.