УДК 004.942

к.т.н., Боярінова Ю.Є., студент Надзуга В.С.

**Національний технічний університет України
«Київський політехнічний інститут»**

УЗАГАЛЬНЕНИЙ ПІДХІД ДО ВИКОРИСТАННЯ ГІПЕРКОМПЛЕКСНИХ ЧИСЛОВИХ СИСТЕМ (НА ПРИКЛАДІ ЗАДАЧІ РОЗДІЛЕННЯ СЕКРЕТУ МЕТОДОМ ПОЛІНОМІВ ЛАГРАНЖА)

Abstract

Julya E. Boiarinova, assistent, PhD; Nadzuga Vladyslav, student
Generelized approach of using hypercomplex numerical system in case solving secret division by Lagrange polynomials

This paper concerns the common approach to algorithms which use operations with hypercomplex numbers. The classical Lagrange polynomial algorithm is studied and discussed. It is defined what operations hypercomplex number class should maintain to be used in common Lagrange polynomial algorithm. The ways for further research are proposed as well.

Вступ

Для вирішення деяких класів задач необхідно подання даних за допомогою більш складних структур, ніж дійсні числа.

Комплексні числа дозволили значно спростити алгоритми обробки інформації та математичні моделі, зробити більш надійним їх функціонування і відкрили можливості вирішення нових задач. Значно більші можливості відкриваються при використанні таких методів подання інформації, як подання за допомогою різних гіперкомплексних числових систем(ГЧС).

Використання методів подання інформації вимагає, перш за все, дослідження питання множинності таких систем, їх структурних, операційних і функціональних властивостей [1]. Вивчення таких питань показало, що на базі таких форм подання інформації можливо побудувати програмно-алгоритмічны системи, які дозволять проводити моделювання практичних задач з підвищеними характеристиками по ефективності їх вирішення.

Постановка задачі

За допомогою використання системи гіперкомплексних чисел може значно покращити криптостійкістні характеристики класичних криптографічних схем вирішення задачі розділення секрету[2-5]. Існує необмежена кількість класів гіперкомплексних числових систем. Вони відрізняються за своїми властивостями. Тому доцільно буде провести дослідження алгоритмів з використанням ГЧС.

Зараз існує певна кількість програмних засобів які дозволяють виконуати дії з класами гіперкомплексних чисел такі як MathCad, Maple, Mathematica.

Існують реалізації дій з ГЧС наприклад в бібліотеці C++ “Boost”, ГЧС в цій бібліотеці має шаблонну реалізацію кватерніонів, при чому базовий елемент кватерніонів є шаблонним, тобто може бути як дійсним числом, так вектором, так і іншим гіперкомплексним або комплексним числом. Але моделювання задачі розділення секрету з гіперкомплексним поданням даних потребує розробки програмного комплексу.

Класична схема розділення секрету методом інтерполяційних поліномів Лагранжа

Розглянемо схему розділення секрету з використанням інтерполяційних поліномів Лагранжа:

Щоб сформулювати вимоги до даних з якими буде працювати схема, продемонструємо схему роботи алгоритму для дійсних чисел.

Нехай потрібно розділити секрет $M \in$ N між n сторонами таким чином, щоб будь-які $k \leq n$ учасників могли б відновити секрет (тобто потрібно реалізувати (k, n) порогову схему).

Виберемо деякий просте число p > M. Це число можна відкрито сказати всім учасникам. Воно задає кінцеве поле розміру p. Над цим полем побудуємо многочлен k-1 ступеня (тобто випадково виберемо всі коефіцієнти многочлена, крім M – вільного члена):

$$F(x) = (a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + a_{k-3}x^{k-3} + \cdots + a_1 x + M)\ mod\ p$$

У цьому многочлені M – це секрет, а решта коефіцієнтів $a_{k-1}, a_{k-2}, a_{k-3}, \ldots, a_1$ – деякі випадкові числа, які потрібно буде «забути» після того, як процедура поділу секрету буде завершена.

Тепер обчислюємо координати різних n точок:

$$k_1 = F(1) = (a_{k-1}1^{k-1} + a_{k-2}1^{k-2} + a_{k-3}1^{k-3} + \cdots + a_1 1 + M)\ mod\ p$$
$$k_2 = F(2) = (a_{k-1}2^{k-1} + a_{k-2}2^{k-2} + a_{k-3}2^{k-3} + \cdots + a_1 2 + M)\ mod\ p$$
$$k_3 = F(3) = (a_{k-1}3^{k-1} + a_{k-2}3^{k-2} + a_{k-3}3^{k-3} + \cdots + a_1 3 + M)\ mod\ p$$

$k_4 = F(4) = (a_{k-1}4^{k-1} + a_{k-2}4^{k-2} + a_{k-3}4^{k-3} + \cdots + a_1 4 + M)\ mod\ p$

Аргументи (номери секретів) не обов'язково повинні йти по порядку, головне – щоб усі вони були різні за модулем p.

Після цього секрети (разом з їх номером, числом p і ступенем многочлена) роздаються учасникам схеми.

Тепер k будь-яких учасників, знаючи координати різних точок многочлена, зможуть відновити многочлен і всі його коефіцієнти, включаючи останній з них дійсне число M – розділений секрет.

Особливістю схеми (як і всіх, що використовуються на практиці) є те, що навіть k - 1 сторін, які зібралися разом, не зможуть знайти секрет навіть методом повного перебору всіх можливих варіантів.

Прямолінійне відновлення коефіцієнтів многочлена через рішення системи рівнянь можна замінити на обчислення інтерполяційного многочлена Лагранжа (звідси одна з назв методу). Формула многочлена буде виглядати наступним чином:

$$F(x) = \sum_i l_i(x) y_i\ mod\ p$$

$$l_i(x) = \prod_{i \neq j} \frac{x - x_j}{x_i - x_j}\ mod\ p$$

де (x_i, y_i) —- координати точок многочлена. Всі операції виконуються також в скінченному полі p.

Вимоги до класу гіперкомплексних чисел для вирішення задачі розділення секрету методом інтерполяційних поліномів Лагранжа

Отже сформулюємо список вимог до гіперкомплексних чисел з якими будемо оперувати. Секрет визначений в множині гіперкомплексних чисел, як і частки, які видані кожній стороні. Модуль p також гіперкомплексний. Згідно вимог алгоритму число p повинно бути простим і більшим за секрет M. Спробуємо термін простоти перенести із множини дійсних чисел на множину гіперкомплексних чисел.

Нехай абстрактний клас абстрактного гіперкомплексного має ім’я CGiperDigit. Цей класс буде інтерфейсом для усіх функцій, які повинні бути присутніми для заміни дійсних чисел на гіперкомплексні.

class CGiperDigit.

Звичайно повинна бути підтримка базових операцій:

```
virtual const CGiperDigit & CGiperDigit::operator +=(const CGiperDigit &);
virtual const CGiperDigit & CGiperDigit::operator *=(const CGiperDigit &);
virtual const CGiperDigit & CGiperDigit::operator /=(const CGiperDigit &);//
```

Поки буде цікавити цілочислене ділення.

Ці операції мають свою загальну реалізацію але можуть мати і свої оптимізовані спеціалізовані реалізації.

Оскільки число p має бути більшим за секрет, повинна бути визначена операція порівняння.

virtual CGiperDigit& CGiperDigit::operator %=(const CGiperDigit &) = 0;

Не менш очевидно, що клас має підтримувати операцію взяття по модулю:

virtual bool CGiperDigit::operator <(const CGiperDigit &) = 0;

virtual CGiperDigit & CGiperDigit::operator %=(const CGiperDigit &) = 0;

Для деяких класів гіперкомплексних числових систем знаходження простого числа p більшого за задане може бути оптимізоване. Тому можна вести не чисто віртуальну функцію знаходження цього віртуального гіперкомплексного числа.

Висновки

Узагальнений підхід може спростити використання ГЧС і дозволить скоротити час переходу алгоритму від використання дійсних до використання гіперкомплексних чисел. Якщо для певної задачі є можливість використовувати різні класи ГЧС, то досить легко буде провести порівняльну характеристику роботи цього алгоритму. Створення програмного середовища, що дозолить використовувати гіперкомплексні числа так само легко як і дійсні, зможе стимулювати використання гіперкомплексних чисел в науці та комерційних проектах.

У майбутньому можна ввести додаткові програмні абстракції, які будуть об’єднувати групи класів гіперкомплексних чисел. Це дозволить використовувати ці групи з певними класами алгоритмів.

Література

1. Синьков М.В. Бояринова Ю.Є. Калиновсий Я.А. Конечномерные гиперкомплексные числовые системы. Основы теории. Применения. — К.:ИПРИ НАН Украины, 2010. 30-37 с, 52-53 с, 68-69 с.
2. Шнайер Б. Прикладная криптография./ Шнайер Б. –М.: Триумф,1995. –816с.
3. Zhao J. A practical verifiable multi-secret sharing scheme/ Zhao J., Zhang J., Zhao R.// Computer Standards & Interfaces – 2007–P.138-141
4. Baldoni M.W. Elementary Number Theory, Cryptography and Codes/ Baldoni M.W., Ciliberto C., Piacentini Cattaneo G.M.// Springer, 2008 - 522 p.
5. Синьков М.В.Развитие задачи разделения секрета / Синьков М.В., Бояринова Ю.Е., Калиновский Я.А., Трубников П.В.// Реєстрація, зберігання і обробка даних. — 2003. — Т. 5,№ 4. — С. 90–96