МОДЕРНІЗАЦІЯ АЛГОРИТМІВ ХЕШУВАННЯ ПРИ ЗАХИСТІ ІНФОРМАЦІЇ

Рисований Олександр Миколайович,
професор, к. т. н., НТУ «ХПІ»
rysov81524@gmail.com

У даному дослідженні розглядаються можливі шляхи до удосконалення алгоритмів хешування при захисті інформації. Головний аспект цієї проблеми – якісні показники алгоритмів не повинні бути гіршими від початкових значень. Найпростішим прийомом удосконалення відомих алгоритмів є шлях зменшення часу отримання хеш без зміни інших критеріїв оцінювання. Це досягається за рахунок використання інструкцій паралельної обробки даних набору AVX2. Наступною проблемою, яка вирішується у роботі є перевірка колізій при отриманні хеш та відповідність якісних показників початковим.

This study considers possible ways to improve hashing algorithms for information protection. The main aspect of this problem is that the quality indicators of the algorithms should not be worse than the initial values. The simplest way to improve known algorithms is to reduce the time to obtain a hash without changing other evaluation criteria. This is achieved by using the parallel data processing instructions of the AVX2 set. The next problem that is solved in the work is checking for collisions when obtaining a hash and the correspondence of the quality indicators to the initial ones.

Актуальність отримання хеш-функцій при захисті інформації визначається безпекою зберігання даних, важливістю застосування цифрового підпису, гарантією цілісності даних, захистом від шкідливого програмного забезпечення. Зростання обчислювальної потужності дозволяє зловмисникам швидше аналізувати великі масиви даних та отримувати доступ до важливої інформації. В даний час існує дуже велика кількість різних алгоритмів одержання хеш, починаючи від простих некриптографічних та закінчуючи складними криптографічними алгоритмами. Практично застосовуються ті алгоритми, які мають основні властивості: стійкості до колізій, рівномірності розподілу хеш і швидкості обчислень.

У роботі проаналізовано шляхи вдосконалення алгоритмів хешування за їх основними властивостями та отримано практичні приклади.

Всі шляхи вдосконалення алгоритмів хешування засновані на головному аспекті цієї проблеми – якісні показники алгоритмів не повинні бути гіршими від початкових значень. Найпростішим прийомом удосконалення відомих алгоритмів є шлях зменшення часу отримання хеш без зміни інших критеріїв оцінювання.

Наступною проблемою, яка вирішується в роботі є перевірка колізій при отриманні хеш та відповідність якісних показників первісним.

Модернізація алгоритмів хешування – це актуальне завдання, особливо з урахуванням зростання обсягів даних, вимог до безпеки та продуктивності (що особливо важливо у контексті середовища програмування MASM64 та низькорівневих реалізацій).

Найпростішим критерієм поліпшення властивостей хеш-функцій є час отримання хеш. Найбільш наочним шляхом є покращення алгоритму обробки даних. Але це – шлях розробника алгоритму. А при зміні алгоритму виходить зовсім інший алгоритм, що не допустимо.

Очевидним шляхом є застосування команд (інструкцій) паралельної обробки даних. Для цього можна використовувати першу масову технологію паралельної обробки на мікропроцесорах – це технологія Intel® MMX (січень 1997 р.). Вона застосовується для збільшення рівня паралелізму під час обробки цілих даних. Її обмеженням є розрядність – всього 8 регістрів по 64 бітів та тип даних – лише цілі числа. Але останній показник для отримання хеш не має значення, тому що хеш це ціле число.

Інструкції SSE використовують регістри XMM0-XMM15 (1999 р.) і мають розрядність 128 бітів [1-2]. Цілі (скалярні) числа позначаються суфіксом SS і мають мінімальну розрядність 32 біти. У розрядній сітці в 128 бітів вміщуються лише 4 32-розрядні скалярні числа. Але і поділ основного потоку на 4 дозволяє отримати хеш 128 бітів. А цього показника на сьогоднішній день вистачає майже всім відомим алгоритмам.

Інструкції AVX використовують регістри YMM0-YMM15 (2008 р.) і розрядність в 256 бітів. Можна використовувати регістри ZMM0-ZMM31 (2017 р.) та інструкції AVX512, але ці регістри використовуються тільки у високопродуктивних ядрах, а в малих (енергоефективних) ядрах вони відсутні. Крім того, компілятори обробляють на високопродуктивних ядрах 512-розрядні регістри, але у налагоджувачах така розрядність поки не введена – видно лише молодші 256-розрядні частини. Наприклад, так відбувається у налагоджувачі рівня ядра з назвою x64Dbg. Не видно і старші регістри ZMM16-ZMM31, хоча у них формується результат.

Таким чином, зараз без обмеження архітектури можна використовувати інструкції AVX2 розрядністю 256 бітів і 16 регістрів YMM. Наприклад, як показано в блоці коду для алгоритму XXHash64:

```
; --- завантажити v1..v4 в ymm0 (4 x 64-bit)
   vmovdqu ymm0, ymmword ptr vstate
; --- broadcast констант
   vpbroadcastq ymm6, qword ptr [P1] ; 64-розрядне в 256
   vpbroadcastq ymm7, qword ptr [P2]
```

Використання інструкцій AVX512, зважаючи на розглянуті особливості, представляється не цілком прийнятним для всіх архітектур мікропроцесорів.

Всі ці технології з різною розрядністю паралельної обробки цілих чисел можна використовувати в одній програмі при формуванні остаточного хеш.

Наступним широко застосовуваним сучасним прийомом прискорення отримання хеш є складання хеша з кількох частин для зменшення часу отримання результату. Причому всі потоки можна обробляти паралельно. Але кількість потоків має відповідати теоретичним можливостям конкретного алгоритму. Наприклад, якщо алгоритм отримує за теорією лише 32-розрядний хеш, процес отримання результату максимум може складатися з 4-х потоків. Але й отриманий

результуючий хеш отримуватиме результат, що обов'язково не відповідає теоретичному послідовному алгоритму. І тут можна говорити, що отримано модифікований результат. І таких модифікацій зараз дуже багато. Найголовніше при такій модифікації – кількість колізій не повинна бути значно більшою, ніж у класичному методі. Їх необхідно перевіряти ще раз. Але при застосуванні початкових теоретичних констант та основних теоретичних пунктів алгоритму кількість колізій не повинна істотно відрізнятись від теоретичного алгоритму.

Наступним прийомом, застосовуваним при модернізації є заміна логічних команд зсуву (SHL, SHR, SAL, SAR) логічними командами циклічного зсуву (ROL, ROR). Це дозволяє не втратити біти, які виштовхуються з розрядної сітки при отриманні результуючого хеш.

Після модернізації алгоритму проводять дослідження отриманого алгоритму. Для цього проводиться побудова гістограми розподілу, статистичний тест на рівномірність Chi-Square тест (χ^2), перевірка бітової рівномірності, розраховується автокореляція, колізії та повторюваність. Крім того, проводиться дискретне перетворення Фур'є, тестU01 "Crush/BigCrush", а також генерація розподілу хеш.

В результаті проведених досліджень основних відомих алгоритмів хешування можна зробити висновок, що не можна поліпшити якийсь один критерій без аналізу інших показників. Цей шлях – це система з кількох рівнянь, у кожному з яких беруть участь інші показники. Найдоступнішим шляхом модернізації є шлях розпаралелювання процесу отримання хеш з використанням сучасних наборів інструкцій.

Після отриманих перетворень необхідно провести розрахунки розподілу колізій щодо відхилення отриманих характеристик від теоретичних.