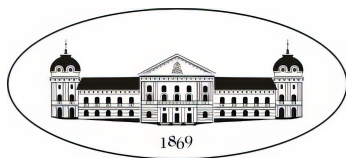


БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ
ИНСТИТУТ ПО МАТЕМАТИКА И ИНФОРМАТИКА



**ЕФЕКТИВНИ АЛГОРИТМИ С ПРИЛОЖЕНИЕ
В КРИПТОГРАФИЯТА С ПУБЛИЧЕН КЛЮЧ
И ТЕОРИЯ НА КОДИРАНЕТО**

Мирослав Марков

Научен ръководител: доц. Юри Борисов

ДИСЕРТАЦИЯ

за присъждане на образователна и научна степен „доктор“

Професионално направление: 4.6 „Информатика и компютърни науки“

Научна специалност: „Информатика“

София, 2024г.

Декларация

Заявявам, че с изключение на случаите, в които изрично се прави препратка към работата на други автори, съдържанието на тази дисертация е оригинално и не е било представяно изцяло или отчасти за разглеждане за придобиване на каквато и да е друга степен или квалификация в този или друг университет, висше училище или научна организация. Резултатите са получени при проведени от мен научни изследвания в сътрудничество и с подкрепата и съдействието на научния ми ръководител. Необходимите за изложението резултати, получени, описани и/или публикувани от други учени, са надлежно и подробно цитирани в библиографията.

Мирослав Марков
София, 2024г.

Благодарности

Бих искал да изразя най-дълбоката си признателност към моя научен ръководител доц. Юри Борисов за неговата неизменна подкрепа, насоки и търпение по време на процеса на създаване на тази дисертация. Получената обратна връзка и конструктивни забележки оказаха дълбоко влияние върху качеството и оригиналността на моите изследвания.

Изразявам своята признателност към ръководството и служителите на Института по математика и информатика (ИМИ) на Българската академия на науките (БАН), които създадоха благоприятна среда за провеждането на моите изследвания. Оказаната институционална подкрепа и предоставения достъп до огромни изчислителни ресурси изиграха съществена роля за успешното завършване на този труд. Изследването, довело до постигането на част от резултатите, е извършено с помощта на инфраструктурата, закупена по Националната пътна карта за научна инфраструктура, финансирана от Министерството на образованието и науката на Република България (договор № Д01-168/28.07.2022г.).

Бих искал да изразя своята благодарност и към колегите си от секция „Математически основи на информатиката“, към ИМИ-БАН. Общуването с тях и техните ценни предложения и конструктивна критика несъмнено подобриха цялостната стойност на моето изследване.

Накрая бих искал да изразя своята признателност към моето семейство за тяхната любов, търпение и безрезервна вяра в мен. Тяхната подкрепа беше безценна по време на процеса на създаване на тази дисертация.

Съдържание

Списък на таблиците	ix
Списък с абrevиатури и съкращения	xi
Резюме	1
Увод	3
Преброяване на точките върху елиптическа крива	3
Теглово разпределение на двоичните кодове на Рид-Малер	7
1 Метод за преброяване на точките върху елиптически криви от вида	
$y^2 = x^3 + a \pmod{p}$, $a \neq 0$	9
1.1 Въведение	9
1.2 Встъпителни бележки	11
1.3 Описание на метода	16
1.3.1 Явна формула за реда на елиптическата крива $E_a \in \mathcal{E}_p$	
редуциран по модул p	17
1.3.2 Изчислителни аспекти на преброяването на точките в \mathcal{E}_p	
когато p е голямо просто число	19
1.3.3 Алгоритми	20
1.4 Примери	23
1.4.1 Пример с модула на елиптическата крива secp256k1	23
1.4.2 Пример с модул просто число, намерено от нас	24
1.4.3 Сравнение на ефективността спрямо алгоритъма SEA	25
1.5 Заключение	27
2 Метод за преброяване на точките върху елиптически криви от вида	
$y^2 = x^3 + ax \pmod{p}$, $a \neq 0$	29

2.1	Въведение	29
2.2	Встъпителни бележки	30
2.3	Описание на подхода	31
2.3.1	Явна формула за реда на елиптичната крива $D_a \in \mathcal{D}_p$, редуциран по модул p	31
2.3.2	Спектърът на $\#D_a$, когато a варира в \mathbb{Z}_p^*	32
2.3.3	Изчислителни аспекти на преброяването на точките в \mathcal{D}_p , когато p е голямо просто число	33
2.4	Пример	34
2.5	Заклучение	35
3	Теглово разпределение на двоичния код на Рид-Малер $\mathcal{R}(4, 9)$	37
3.1	Основни понятия	37
3.2	Опит за намиране на тегловото разпределение чрез теоремата на Gleason	49
3.3	Описание на метода на Sarwate и прецизирането му	53
3.3.1	Теоретично представяне	53
3.3.2	Пресмятане на $\mathcal{W}[z; \mathcal{R}(4, 9)]$	55
3.3.3	Оценяване на изчислителните разходи	57
3.3.4	Използвани компютърни ресурси	59
3.4	Заклучение	59
	Библиография	61
	Приложение А: $\mathcal{R}(4, 9)$ таблици	67
	Приложение Б	71
	Изнесени доклади по темата на дисертацията	73
	Публикации по темата на дисертацията	75

Списък на таблиците

1	Еквивалентен размер на ключа (битове)	4
1.1	Прости числа p_i , $i = 1, \dots, 10$	26
1.2	Сравнение на ефективността	27
3.1	Таблицата на истинност на f	38
A.1	Размери на разбиванията $\Delta(e)$ и $\Delta'(e)$	67
A.2	Съответствие между $\mathcal{E}'(4, 7)$ и $\mathcal{E}(4, 7)$	68
A.3	Теглово разпределение на $[512, 256, 32]$ Рид-Малер кода	69

Списък с абривиатури и съкращения

Акроними / Абривиатури

ACCT	Algebraic and Combinatorial Coding Theory
\mathcal{ALR}	Absolute Least Residue
ANF	Algebraic Normal Form
APR-CL	Adleman-Pomerance-Rumely - Cohen-Lenstra
CPU	Central Processing Unit
\mathcal{CR}_p	Cubic Residue (mod p)
ECC	Elliptic Curve Cryptography
ECDLP	Elliptic Curve Discrete Logarithm Problem
ECDSA	Elliptic Curve Digital Signature Algorithm
GA	General Affine group
GL	General Linear group
GMP-ECM	GNU Multiple Precision Arithmetic Library - Elliptic Curve Method
GMP	GNU Multiple Precision Arithmetic Library
IoT	Internet of Things
IWSDA	International Workshop on Signal Design and its Applications in Communications
\mathcal{LNR}	Least Non-negative Residue

NIST	National Institute of Standards and Technology
PARI/GP	Computer Algebra System for fast computations in Number Theory
\mathcal{QR}_p	Quadratic Residue (mod p)
RAM	Random-access Memory
RSA	Rivest–Shamir–Adleman
SAGE	System for Algebra and Geometry Experimentation
SEA	Schoof-Elkies-Atkin
SEC	Standards for Efficient Cryptography
WCC	International Workshop on Coding and Cryptography

Резюме

Целта на настоящото изследване е да разработи ефективни алгоритмични методи за решаването на две специфични задачи: „Преброяване на точките върху елиптични криви“ и „Пресмятане на тегловото разпределение на двоични Рид-Малер кодове“. Тези алгоритмични методи намират приложение в областта на криптографията с публичен ключ и теорията на кодирането.

Конкретните **задачи** за постигането на тази цел са:

1. Да се изведе явна формула за реда на крива от семейството

$$\mathcal{E}_p = \{E_a : y^2 = x^3 + a \pmod{p}, a \neq 0\},$$

редуциран по модул p .

2. Да се разработи ефективен алгоритъм за едновременно пресмятане на шестте възможни реда, свързани с \mathcal{E}_p , при фиксирано $p \equiv 1 \pmod{6}$.
3. Да се сравни ефективността на алгоритъма SEA [18] спрямо предложения от нас, и двата работещи върху случая $p \equiv 1 \pmod{6}$.
4. Да се изведе явна формула за реда на крива от семейството

$$\mathcal{D}_p = \{D_a : y^2 = x^3 + ax \pmod{p}, a \neq 0\},$$

редуциран по модул p .

5. Да се разработи ефективен алгоритъм за едновременно пресмятане на четирите възможни реда, свързани с \mathcal{D}_p , при фиксирано $p \equiv 1 \pmod{4}$.
6. Да се разработи ефективен алгоритъм за пресмятане на тегловото разпределение на двоичния Рид-Малер код $\mathcal{R}(4, 9)$.

Глава 1 описва подход за определяне броя на точките върху елиптични криви от семейството $\mathcal{E}_p = \{E_a : y^2 = x^3 + a \pmod{p}, a \neq 0\}$, където p е просто число по-голямо от 3. Същността на този подход се състои в комбиниране на добре познатата граница на Хасе с явна формула за броя на точките, редуциран по модул p . Това позволява да се разработи ефективен метод за пресмятане на шестте мощности (виж Следствие 1.3.4) свързани със семейството \mathcal{E}_p . Сложността на този метод е $\tilde{O}(\log^2 p)$, което подобрява най-доброто известно досега алгоритмично решение [43] с почти един порядък. (Да отбележим, че съгласно дефиницията си сложност $\tilde{O}(n)$ е кратко означение за $O(n \log n)$ [13], т.е. премахнати са членовете с по-нисък ред, които не влияят съществено на основния член на израза.)

В **Глава 2** е представен метод за пресмятане на редовете на елиптични криви от семейството $\{D_a : y^2 = x^3 + ax \pmod{p}, a \neq 0\}$, където p е просто число. Този метод комбинира затворена формула за редовете на кривите, редуцирани по модул p , с добре известната граница на Хасе. Освен своята ефективност (алгоритъмът изисква най-много $\tilde{O}(\log^2 p)$ битови операции), подходът позволява да се определи и спектърът на редовете на кривите, когато p е фиксирано.

В **Глава 3** е пресметнато тегловото разпределение на кода на Рид-Малер $\mathcal{R}(4, 9)$ с дължина 512 от 4-ти ред. Предложеният от нас метод комбинира подхода, описан в докторската дисертация на D. V. Sarwate [46] от 1973 г., с резултатите за класификацията на Булевите функции според афинната им еквивалентност, публикувани през 2008г. в [34] и 2023г. в [19]. По-конкретно за решаването на тази задача, поставена например в книгата на MacWilliams и Sloane [36] от 1977г., е приложен подход, базиран на класификацията на Булевите форми от четвърта степен на осем променливи [34], получена от Ph. Langevin и G. Leander през 2008г., както и на скорошните резултати относно класификацията на фактор-пространството $\mathcal{R}(4, 7)/\mathcal{R}(2, 7)$ [19], която дължим на V. Gillot и Ph. Langevin.

В **Приложение А** е предоставено тегловото разпределение на кода на Рид-Малер $\mathcal{R}(4, 9)$, както и някои таблици с междинни резултати, получени в процеса на намирането му.

В **Приложение Б** е предоставено описание на методите за компресиране и кодиране на данните, получени във фазата на предварителните изчисления в Глава 3, както и примери, които илюстрират използването на съответните методи.

Предоставени са списъци на публикациите и изнесените доклади, свързани с темата на дисертацията.

Предоставен е списък на използваната литература.

Увод

Преброяване на точките върху елиптична крива

Глави 1 и 2 на дисертацията са посветени на описанието на два ефективни алгоритъма за пресмятане броя на точките върху елиптични криви от семействата

$$\mathcal{E}_p = \{E_a : y^2 = x^3 + a \pmod{p}, a \neq 0\}$$

и

$$\mathcal{D}_p = \{D_a : y^2 = x^3 + ax \pmod{p}, a \neq 0\},$$

където p е просто число.

Елиптичните криви над крайни полета са основен компонент в съвременната криптография, характеризиращи се с хармонична комбинация от алгебрична елегантност и изчислителна ефективност. Те осигуряват стабилна основа за разработването на сигурни системи за криптиране с публичен ключ, които са устойчиви срещу различни видове криптографски атаки, включително и специално адаптирани. За по-задълбочено разбиране на криптографския им потенциал вижте [54] (вж. също така пионерските трудове на V. Miller и N. Koblitz от 80-те години [41, 31]). Основно предимство на криптографията с използване на елиптични криви (ECC), в сравнение с традиционната публична криптография (RSA), е че ECC може да осигури същото ниво на защита, но с по-къси ключове. Например, алгоритъмът за цифров подпис, базиран на елиптична крива (ECDSA), предлага еквивалентна криптографска сила при много по-къси ключове в сравнение с RSA (виж Таблица 1). Това е много полезно за ефективността и производителността на криптографските системи и особено за мобилни устройства и интернет на нещата (IoT), които обикновено имат ограничена изчислителна мощност.

Таблица 1 Еквивалентен размер на ключа (битове)

ECDSA	RSA
160	1024
224	2048
256	3072
384	7680
512	15360

Ключов аспект при проектирането на една система за криптиране с помощта на елиптични криви е изучаването на броя на точките върху избраната крива, тъй като той играе централна роля при определяне на сигурността, размера на ключа и производителността на криптографската система. В контекста на елиптичните криви над прости полета (напр. \mathbb{F}_p), броят на точките върху кривата (наречен ред на кривата) е основна характеристика, която може да служи като критичен параметър, влияещ на избора на останалите параметри на системата за криптиране с публичен ключ. Нека разгледаме всяка една от тези характеристики поотделно:

1. **Брой на точките:** За елиптична крива E над просто поле \mathbb{F}_p , с уравнение $y^2 = x^3 + ax + b \pmod{p}$, броят на точките (означен като N) е фундаментална характеристика, представляваща общия брой точки (включително безкрайната точка), които удовлетворяват уравнението на кривата. Намирането на броя на точките върху кривата е задача от съществено значение за целите на криптографията. Добре известно е, че за да се избегне успешно прилагане на известните атаки срещу криптографска система, базирана на криптография с елиптични криви, броят на точките върху използваната крива трябва да има поне един много голям прост делител. По-специално, когато редът е голямо просто число, тогава се използва целият капацитет на кривата за постигане на максимално ниво на сигурност.

2. **Сигурност и Размер на ключа:** Сигурността на криптографията с елиптични криви се основава на трудността за намиране на дискретния логаритъм на произволен елемент от елиптичната крива спрямо публично известната базова точка, т.е. намирането при произволно избрана точка P на скалара $k : P = kG$, където G е базовата точка на кривата. Тази задача често се нарича „задача за дискретния логаритъм върху елиптична крива“ или „Elliptic Curve Discrete Logarithm Problem“ (ECDLP). Сигурността на системата е пряко свързана с размера на групата, образувана от точките върху кривата, който се определя от броя на точките N , удовлетворяващи уравнението на кривата

$E : y^2 = x^3 + ax + b \pmod{p}$). По-голямо N обикновено предполага по-висока сигурност, тъй като увеличава изчислителната сложност за решаване на ECDLP чрез пълно изчерпване. Криптографите често избират елиптични криви от голям прост ред, за да осигурят високо ниво на сигурност. Размерът на ключа се определя от дължината в битове на N , като по-дългите ключове (както вече споменахме) осигуряват по-силни гаранции за сигурност. В заключение, при определянето на параметри на ЕСС-система е важно да се избират криви от достатъчно голям прост ред, за да е устойчива тя на атаки, прилагащи ρ -алгоритъма на Pollard и други подобни (вж. напр. [57]).

3. Производителност: Броят на точките върху елиптична крива влияе на производителността на криптографските операции. Кривите с по-голям брой точки изискват повече изчислителни ресурси и време за генериране на ключове, както и за шифриране и дешифриране. За разлика от тях кривите с по-малък брой точки позволяват по-бързи изчисления и са подходящи за приложения, изискващи работа в реално време или с големи масиви от данни, но разбира се това е за сметка на сигурността, както вече споменахме. Затова предварителното познаване на съответните сценарии за използване позволява постигане на баланс между сигурност и производителност чрез избор на криви с подходящ брой точки.

4. Сигурност срещу Производителност: При избора на елиптични криви за публична криптография се търси компромис между сигурността, размера на ключа и производителността. Кривите с изключително голям брой точки могат да предложат висока сигурност за сметка на по-ниска производителност и по-големи ключове. Обратно, кривите с малък брой точки могат да предложат по-висока производителност, но са по-уязвими при атаки. Криптографите трябва внимателно да оценят и направят необходимите компромиси за да изберат елиптични криви, които отговарят на желаното ниво на сигурност, като същевременно поддържат приемлива производителност.

5. Стандартизация: Стандартизационни органи като National Institute of Standards and Technology (NIST) играят решаваща роля при определянето на насоки за избор на елиптични криви за публична криптография. Тези организации предоставят препоръки относно параметрите на кривата, включително броя на точките, които отговарят на специфични изисквания за сигурност. NIST публикува стандарти за криптография с елиптични криви, включително специфични елиптични криви с дефиниран брой точки, които се считат за сигурни при различни нива на сигурност. Тези стандартизирани криви

преминават стриктно оценяване, за да се гарантира, че предлагат добър баланс между сигурност, размер на ключа и производителност и са подходящи за широко използване в приложения за криптография.

В резюме, броят на точките върху елиптична крива над просто поле е критичен параметър, който влияе върху сигурността, размера на ключа и производителността на криптосистемите. Криптографите трябва много внимателно да вземат предвид последствията, които могат да са следствие от броя на точките върху избраните криви, за да се постигне сигурно и ефективно криптографско решение.

Теглово разпределение на двоичните кодове на Рид-Малер

Глава 3 на дисертацията е посветена на пресмятане тегловото разпределение на двоичния код на Рид-Малер $\mathcal{R}(4, 9)$ с дължина 512 от 4-ти ред.

Добре известно е, че тегловото разпределение на линеен код позволява да се определят вероятностите за грешка и за неуспех при декодиране, когато съответният код се използва за предаване на информация по зашумен канал и декодерът прилага алгоритъм за декодиране, базиран на принципа за максимална правдоподобност.

За основните понятия от теория на кодирането се позоваваме на [36]. Кодовете, които разглеждаме тук, са двоични, т.е. над азбуката $\mathbb{F}_2 = \{0, 1\}$.

Двоичните кодове на Рид-Малер представляват фундаментален клас кодове за корекция на грешки, които намират приложение в безжичните комуникации. Техните възможности за коригиране на множество произволни грешки позволяват надеждно предаване на данни през огромни разстояния, където влошаването на сигнала поради шум и смущения е неизбежно, и това им отрежда значителна роля в комуникационните системи свързани с далечния космос [39]. Кодовете на Рид-Малер се наричат така според имената на техните изобретатели: David E. Muller, който ги открива през 50-те години на миналия век [42], и Irving S. Reed, който през 1954г. предлага първия ефективен алгоритъм за декодирането им [45], базиран на логически схеми с мажоритарна логика. Въпреки че са известни толкова отдавна, има малко общи резултати относно тегловата им структура, т.е., тегловите разпределения са известни само за:

- кодовете от 1^{6u} и 2^{pu} ред [51] (1970);
- произволен ред, когато теглото е по-малко от $2d$ [27] (1970), разширено по-късно ([28], 1976) за тегла по-малки от $2.5d$, където d е минималното тегло.

В едно по-ранно изследване [55] са представени резултати за спектрите на теглата на някои кодове от трети ред, а в съвсем скорошната работа [9] се разглеждат спектрите на цели семейства от двоични Рид-Малер кодове. Някои частични резултати относно тегловите разпределения на кодовете на Рид-Малер от трети и четвърти ред са получени в [46], [28], [53] и [52]. За повече информация

по този въпрос, касаещ двоичните кодове от този вид с конкретни дължина и ред, на читателя се препоръчва да се обърне към [50].

Тегловият спектър на кода на Рид-Малер от четвърти ред с дължина 512, $\mathcal{R}(4, 9)$, е намерен в [9], където е представен като числен пример, който демонстрира разработения там метод. Доколкото ни е известно, има съвсем малко опити да се определи точното теглово разпределение на този код, който е един от изброените най-малки кодове на Рид-Малер, чиито теглови разпределения са неизвестни към 1977 година (виж, [36, р. 447]). Конкретно, в заключителните бележки на докторската си дисертация [46], D. V. Sarwate обсъжда приложимостта на методите, описани от него, към кодовете на Рид-Малер с дължини по-големи от 256. Според неговата оценка има твърде много класове на еквивалентност, състоящи се от съседни класове от желания тип и съобразно това той прави заключението, че намирането на тегловото разпределение на $\mathcal{R}(4, 9)$ е недостижимо чрез тези методи. Друг обещаващ подход за атакуване на разглеждания проблем се състои в използването на факта, че се занимаваме с двойночетен двоичен самодуален код и общата форма на номераторите на теглата на такива кодове е известна от работата на A. M. Gleason (виж, например, [36, Ch.19]). Въпреки че този втори подход се е доказал в случая с по-къси кодове от този вид и изисква скромни изчислителни усилия, за успешното му прилагане се нуждаем от повече съществена информация за $\mathcal{R}(4, 9)$ от тази, представена в [27] (виж, [11, Ch. 11] за подробности). Не е ясно също така дали разработеният в [53] инструментариум би могъл да бъде приложен успешно към $\mathcal{R}(4, 9)$, тъй като по това време (около 1996г.) класификацията на Булевите форми от четвърта степен на осем променливи не е била налична.

Глава 1

Метод за преброяване на точките върху елиптични криви от вида

$$y^2 = x^3 + a \pmod{p}, \quad a \neq 0$$

Изследването, което се съдържа в тази глава, се основава на работата, представена от нас в статията със заглавие „*An Approach for Computing the Number of Points on Elliptic Curve $y^2 = x^3 + a \pmod{p}$ via Explicit Formula for That Number Modulo p* “ [P1] \cong [3], както и в статията със заглавие „*An Efficient Approach to Point-Counting on Elliptic Curves from a Prominent Family over the Prime Field \mathbb{F}_p* “ [P3] \cong [4].

1.1 Въведение

В [47] е представен ефективен детерминистичен алгоритъм за пресмятане реда (т.е. броя на точките) на дадена елиптична крива от общ тип, чиято сложност е най-много константа по $\log^8 q$ побитови операции, където q е реда на използваното крайно поле. В тази глава обаче интерес представлява цялото семейство от криви

$$\mathcal{E}_p = \{E_a : y^2 = x^3 + a \pmod{p}, a \neq 0\},$$

съдържащо $p - 1$ члена. Ето защо прилагането на алгоритъма на Schoof [47] за намиране редовете на кривите в \mathcal{E}_p е под въпрос, когато p е голямо (въпреки че е осъществимо, вземайки предвид съществуването само на шест еднакво вероятни възможности (вж. Следствие 1.3.4) и задачата за *събирането на купони* от елементарната теория на вероятностите [14]). Подобно съждение е валидно и към

10 $\tilde{O}(\log^2 p)$ метод за преброяване на точките върху $y^2 = x^3 + a \pmod{p}$, $a \neq 0$

стохастичното подобрене на алгоритъма на Schoof, т.е. SEA алгоритъма [47] с очаквано време за изпълнение, евристично, $\tilde{O}(\log^4 p)$.

В тази глава се разглежда задачата за определяне на реда $\#E_a$ на кривата $E_a \in \mathcal{E}_p$ в термините на параметъра a и модула p . Успешно решаване на поставената задача е възможно с помощта на алгоритмичния подход описан в [43], който отнема $O(\log^3 p)$ побитови операции. Съществува обаче и още по-добър подход, предложен от нас, който е обект на изучаване в главата.

Има две основни разлики между подхода, следван в [43], и описания тук:

- Munuera и Tena предлагат да се използва стохастичен алгоритъм от общ тип [44] за намиране на квадратния корен на произволен квадратичен остатък по модул p , с цел да се пресметне $\sqrt{-3}$, където $p \equiv 1 \pmod{3}$. По такъв начин техният алгоритъм става със сложност $O(\log^3 p)$, докато предложеният тук подобрява сложността до $\tilde{O}(\log^2 p)$, дължащо се на използвания ефективен целеви метод за пресмятане на тази специфична стойност;
- Авторите на [43] намират решения на диофантовото уравнение $F(X, Y) = X^2 + XY + Y^2 = 3p$, докато ние постигаме целта си чрез решаване на уравнението $X^2 + 3Y^2 = p$. Ще отбележим само, че и двете последни задачи се решават чрез подходящи прилагания на алгоритъма на Евклид за p и $\sqrt{-3} \pmod{p}$, така че и двете отнемат $O(\log^2 p)$ побитови операции съгласно оценките за сложност на този алгоритъм (вж. напр. [30] или [56]).

Въз основа на изложените две разлики може да се заключи, че нашият подход е по-добър от този в [43] почти на порядък, въпреки че също е от стохастичен тип.

За аналитично решаване на разглежданата задача препращаме читателя към [29], където са получени конкретни формули за реда на кривата $E_a \in \mathcal{E}_p$ по отношение на представянето на простото число p във вида $p = X^2 + Y^2 - XY$ за някакви цели числа X и Y . Тези формули обаче разглеждат множество отделни случаи, а изчислителната ефективност далеч не е сред основните цели на автора (вж. по-подробно в [29], Теорема 1). Някои частни случаи на тази задача са дадени и като упражнения в [26, Гл. 8, Упр. 15 и 27].

Накрая си струва да се отбележи, че резултатите, получени с помощта на подхода следван в тази глава, са изчерпателни и компактни, въпреки че използват някои отдавна установени факти от теорията на квадратичните разбивания на

прости числа. Да отбележим, още, че този подход беше предложен за първи път от нас в [P1] \cong [3], но там неговата ефективност бе демонстрирана само за случая $p \equiv 7 \pmod{12}$, докато в тази глава идеята е допълнително доразвита, усъвършенствана и разработена в пълна цялост.

Глава 1 е организирана, както следва. В Раздел 1.2 се предоставят някои предварителни знания. Раздел 1.3 излага подхода към проблема, включително подобрените оценки за изчислителната сложност при големи p . Раздел 1.4 предоставя пример със специално конструирано просто число за модул и също така обсъжда резултатите от експеримент с програма за сравняване производителността на предложената алгоритмична техника спрямо тази на алгоритъма SEA в разглеждания сценарий. В последния раздел са направени някои заключения.

1.2 Встъпителни бележки

Нека p е просто число, по-голямо от 3, и нека \mathbb{Z}_p е пръстенът от остатъци по модул p , който може също така да бъде идентифициран с крайното поле \mathbb{F}_p . Разглеждаме семейството от елиптични криви, дефинирани като

$$\mathcal{E}_p = \{E_a : y^2 = x^3 + a \pmod{p}, a \in \mathbb{Z}_p^*\},$$

където $\mathbb{Z}_p^* = \mathbb{Z}_p \setminus \{0\}$ е мултипликативната група на пръстена \mathbb{Z}_p . Целта е да се намери подходящ метод, даващ затворени алгебрични формули за пресмятане на реда на общия член на това семейство, кривата E_a , означаван като $\#E_a$.

Цялото число z се нарича квадратичен остатък по модул p , ако съществува цяло x , такова че $z = x^2 \pmod{p}$. В противен случай z се нарича квадратичен неостатък по модул p . По подобен начин се дефинира остатък (неостатък) от порядък $d > 2$ по модул p , а именно, когато съществува (не съществува) цяло число x , такова че да е изпълнено $z = x^d \pmod{p}$. Множеството на всички остатъци от порядък d лежащи в \mathbb{Z}_p^* образуват подгрупа на \mathbb{Z}_p^* . Ще означаваме подгрупите от квадратичните и кубичните остатъци ($d = 2, 3$) по модул p съответно с \mathcal{QR}_p и \mathcal{CR}_p .

Дефиниция 1.2.1 *Символ на Лежандър:* За дадено нечетно просто число p и цяло число a , символът на Лежандър $\left(\frac{a}{p}\right)$ се дефинира по следния начин:

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{ако } a \text{ е квадратичен остатък по модул } p \text{ и } a \not\equiv 0 \pmod{p}, \\ -1 & \text{ако } a \text{ е квадратичен неостатък по модул } p, \\ 0 & \text{ако } a \equiv 0 \pmod{p}. \end{cases}$$

Теорема 1.2.2 Броят N на точките върху елиптичната крива $y^2 \equiv x^3 + ax + b \pmod{p}$ е

$$N = p + 1 + \sum_{x=0}^{p-1} \left(\frac{x^3 + ax + b}{p} \right),$$

където $\left(\frac{r}{p}\right)$ е символът на Лежандър.

Доказателство: Когато x се променя между 0 и $p-1$, броят на y , за които е удовлетворено сравнението $y^2 \equiv x^3 + ax + b \pmod{p}$, е 0, 1 или 2, в зависимост от това дали стойността на израза $x^3 + ax + b$ е квадратичен неостатък, е сравнима с 0, или е квадратичен остатък, всичките по модул p . Като вземем предвид и точката в безкрайността, получаваме

$$\begin{aligned} N &= 1 + \sum_{x=0}^{p-1} \left[1 + \left(\frac{x^3 + ax + b}{p} \right) \right] = \\ &= p + 1 + \sum_{x=0}^{p-1} \left(\frac{x^3 + ax + b}{p} \right), \end{aligned}$$

с което доказателството е завършено. \square

Дефиниция 1.2.3 *Най-малък неотрицателен остатък (\mathcal{LNR})* на цяло число z по модул друго (нечетно) цяло число m : Ако $z \equiv a \pmod{m}$ и $0 \leq a \leq m-1$, тогава a се нарича най-малък неотрицателен остатък на z по модул m .

Дефиниция 1.2.4 *Най-малък по абсолютна стойност остатък (\mathcal{ACR})* (минимален остатък) на цяло число z по модул друго (нечетно) цяло число m : Под най-малък по абсолютна стойност остатък на $z \pmod{m}$ имаме предвид този остатък на z , който лежи между $-\frac{1}{2}m$ и $\frac{1}{2}m$. Той е положителен или отрицателен в зависимост от това дали най-малкият неотрицателен остатък на $z \pmod{m}$ лежи съответно между 0 и $\frac{1}{2}m$ или между $\frac{1}{2}m$ и m .

Означенията „ \equiv “ за сравнимост по модул p и „ $=$ “ в \mathbb{Z}_p ще бъдат използвани взаимозаменяемо, в зависимост от контекста.

Следващият факт е непосредствено разширение на известния критерий на Ойлер от елементарната теория на числата (вж., например, [21] гл. 7.5).

Твърдение 1.2.5 *Ако d дели $p-1$, тогава мономот $\mathbf{m}(z) = z^{\frac{p-1}{d}}$ приема точно d различни стойности в \mathbb{Z}_p^* , всяка от които $\frac{p-1}{d}$ пъти. Тези стойности са d -тите корени на единицата в \mathbb{Z}_p^* . В частност, $\mathbf{m}(z)$ е равно на 1 тогава и само тогава, когато z е остатък от порядък d по модул p .*

За пълнота представяме доказателство на това твърдение.

Доказателство: Да припомним, \mathbb{Z}_p^* е циклична група, изоморфна на адитивната група на \mathbb{Z}_{p-1} , и нека g бъде нейният генериращ елемент, т.е.

$$\mathbb{Z}_p^* = \{g^0 = 1, g, g^2, \dots, g^{p-2}\}.$$

За фиксирано $r : 0 \leq r < d$, нека

$$I_r = \{i \in \mathbb{Z}_{p-1} : i \equiv r \pmod{d}\}$$

и

$$C_r = \{g^i : i \in I_r\}.$$

Множествата C_r , $0 \leq r < d$, образуват разбиване на \mathbb{Z}_p^* , като всяко от тях е с мощност $\frac{p-1}{d}$. (Последното твърдение е частен случай на теоремата на Лагранж, за подгрупата на остатъците от порядък d в \mathbb{Z}_p^* .) Освен това е лесно да се покаже, че $z^{\frac{p-1}{d}}$ е инвариантно върху C_r за фиксирано r . Наистина, като вземем предвид, че редът на \mathbb{Z}_p^* е $p-1$, за всяко $\zeta \in C_r$ имаме: $\zeta^{\frac{p-1}{d}} = c^r$, където константата c е равна на $g^{\frac{p-1}{d}}$. Сега, нека r_1, r_2 са два различни остатъка по модул $d : r_1 < r_2$. Да допуснем, че $c^{r_1} = c^{r_2}$, т.е.

$$c^{r_2-r_1} = (g^{\frac{p-1}{d}})^{(r_2-r_1)} = 1,$$

което води до противоречие с предположението, че g е генериращ елемент на \mathbb{Z}_p^* , тъй като

$$\frac{p-1}{d}(r_2-r_1) < \frac{p-1}{d} \times d = p-1.$$

14 $\tilde{O}(\log^2 p)$ метод за преброяване на точките върху $y^2 = x^3 + a \pmod{p}$, $a \neq 0$

Следователно, $c^{r_1} \neq c^{r_2}$ за всяко $r_1 \neq r_2$, което означава, че мономът $z^{\frac{p-1}{d}}$ приема различни стойности върху подмножествата C_r , $0 \leq r < d$. Накрая, за да докажем, че това са d -тите корени на единицата в \mathbb{Z}_p^* , просто забелязваме, че:

$$(c^r)^d = ((g^{\frac{p-1}{d}})^r)^d = ((g^{\frac{p-1}{d}})^d)^r = (g^{p-1})^r = 1,$$

с което доказателството е завършено. \square

Добре известно е, че $-3 \in \mathcal{QR}_p$ тогава и само тогава, когато $p \equiv 1 \pmod{3}$. (Едно кратко доказателство на този факт може да се получи, като се използва например законът за квадратичната реципрочност.) Да отбележим също, че в този случай $\sqrt{-3} \pmod{p}$ приема две стойности, които се различават с точност до знак.

Следващото твърдение е от съществено значение за ефективността на нашия подход и показва как да намерим поне един от исканите квадратни корени.

Твърдение 1.2.6 *Нека z е кубичен неостатък по модул p , където $p \equiv 1 \pmod{3}$. Тогава $2z^{\frac{p-1}{3}} + 1$ е равно на един от квадратните корени $\sqrt{-3} \pmod{p}$.*

Доказателство: Съгласно Твърдение 1.2.5, предположението $z \notin \mathcal{CR}_p$ води до извода, че $z' = z^{\frac{p-1}{3}}$ е трети корен на единицата в \mathbb{Z}_p^* , различен от 1. Така z' удовлетворява квадратното уравнение $Z^2 + Z + 1 = 0$ в полето \mathbb{F}_p , т.е. $z' = \frac{-1 \pm \sqrt{-3}}{2}$, което е еквивалентно на $\pm\sqrt{-3} = 2z' + 1$. \square

Забележка 1.2.7 *От Твърдение 1.2.5 (при $d = 3$) лесно може да се заключи, че когато $p \equiv 1 \pmod{3}$, броят на елементите на множеството от кубични неостатъци по модул p е равен на $\frac{2}{3}(p-1)$. От последното може да се направи извода, че случайно избран елемент от \mathbb{Z}_p^* е кубичен неостатък с вероятност от $2/3$. Следователно, при наличието на висококачествен генератор на случайни цели числа в интервала $[2, p-1]$, кубичен неостатък може да бъде намерен чрез средно 1.5 опита. По този начин, квадратните корени на -3 по модул p могат да бъдат ефективно определени с помощта на Твърдение 1.2.6.*

Следващото твърдение е факт от математическия фолклор и също се използва в нашия подход към поставената задача.

Твърдение 1.2.8 *За нечетно просто число p , нека*

$$S_k(p) = 1^k + 2^k + \dots + (p-1)^k,$$

където $k = 0, 1, \dots$. Тогава е валидно:

$$S_k(p) \pmod{p} = \begin{cases} 0, & \text{ако } k \not\equiv 0 \pmod{p-1} \\ -1, & \text{в противен случай.} \end{cases}$$

Доказателство: Използваме факта, че \mathbb{Z}_p^* е циклична група. Нека g е нейният пораждащ елемент, тоест за всяко $z \in \mathbb{Z}_p^*$, съществува $i : 0 \leq i \leq p-2$, такова че $z = g^i$. Това означава, че

$$S_k(p) = \sum_{z=1}^{p-1} z^k \equiv \sum_{i=0}^{p-2} (g^i)^k \pmod{p}.$$

Полагайки $u = g^k$ виждаме, че

$$S_k(p) \pmod{p} = \sum_{i=0}^{p-2} u^i.$$

Следователно трябва да се разгледат следните два случая:

- ако $k \not\equiv 0 \pmod{p-1}$, тъй като редът на \mathbb{Z}_p^* е $p-1$, следва че $u = g^k \neq 1$, което води до $S_k(p) \pmod{p} = (u^{p-1} - 1)/(u - 1) = 0$;
- в противен случай $u = 1$, и следва, че $S_k(p) \equiv p-1 \pmod{p} = -1$.

С това доказателството е завършено. \square

Няма явна формула за броя на точките върху елиптична крива над \mathbb{Z}_p от общ тип. Най-подходящият и добре известен резултат в тази посока е следната граница (вижте, напр., [57] гл. 4).

Теорема 1.2.9 (Хасе) *Броят на точките N върху елиптична крива над \mathbb{Z}_p удовлетворява неравенството*

$$|(N-1) - p| \leq 2\sqrt{p}.$$

В края на този раздел, припомняме един необходим факт от теорията на квадратичните разбивания на прости числа. Това е отдавна известен резултат, който дължим на Якоби (1827), и по-късно доразвит от Щерн (1832) (вижте, [17] том III, стр. 55 за исторически подробности).

Твърдение 1.2.10 Ако p е просто число от вида $p = 6k + 1$, такова че $p = X^2 + 3Y^2$, тогава

$$\pm 2X = \frac{(2k+1) \dots (3k)}{k!} \pmod{p},$$

където знакът е такъв, че $\pm X \equiv 1 \pmod{3}$.

Ще използваме Твърдение 1.2.10 и Равенство 1.9, за да пресметнем биномния коефициент по модул p в Израз 1.7, като вземем подходящото X от решенията на квадратното диофантово уравнение $X^2 + 3Y^2 = p$.

1.3 Описание на метода

Следното твърдение помага недвусмислено да се намери броя N на точките върху дадена елиптична крива, при условие че може да се пресметне минималният по абсолютна стойност остатък на $(N - 1)$ по модул p , означен като $\mathcal{ALR}(N - 1, p)$.

Твърдение 1.3.1 В означенията на Теорема 1.2.9, за просто число $p \geq 17$, е валидно:

$$N = \mathcal{ALR}(N - 1, p) + p + 1.$$

Доказателство: Лесно се доказва, че ако $p \geq 17$, то $2\sqrt{p} < \frac{p}{2}$. Така от теоремата на Хасе се получава, че

$$|(N - 1) - p| < \frac{p}{2},$$

което означава, че

$$\mathcal{ALR}(N - 1, p) = (N - 1) - p.$$

С това доказателството е завършено. □

Между \mathcal{ALR} и \mathcal{LNR} има очевидна връзка, която се използва на определен етап в пресмятанията:

Забележка 1.3.2 Ако се пресметне най-малкият неотрицателен остатък R на цяло число z при деление на нечетно m , то може лесно да се получи:

$$\mathcal{ALR}(z, m) = \begin{cases} R, & \text{ако } R < \frac{m}{2} \\ R - m, & \text{в противен случай.} \end{cases}$$

1.3.1 Явна формула за реда на елиптичната крива $E_a \in \mathcal{E}_p$ редуциран по модул p

Като изключим точката в безкрайността и вземем предвид дефиницията на символа на Лежандър, за мощността $\#E_a - 1 = N'$ на множеството от „реални“ точки, лежащи върху кривата $E_a \in \mathcal{E}_p$, се получава следният израз:

$$N' = \sum_{x=0}^{p-1} \left[1 + \left(\frac{x^3 + a}{p} \right) \right] = p + \sum_{x=0}^{p-1} \left(\frac{x^3 + a}{p} \right). \quad (1.1)$$

След това, като редуцираме Равенство 1.1 по модул p и използваме критерия на Ойлер, получаваме:

$$N' \equiv \sum_{x=0}^{p-1} \left(\frac{x^3 + a}{p} \right) \equiv \left[\left(\frac{a}{p} \right) + h(a, p) \right] \pmod{p}, \quad (1.2)$$

където с $h(a, p)$ е означена сумата $\sum_{x=1}^{p-1} (x^3 + a)^{\frac{p-1}{2}}$.

По-нататък, като развием биномите и променим реда на сумиране, получаваме:

$$h(a, p) = \sum_{x=1}^{p-1} \sum_{i=0}^{\frac{p-1}{2}} \binom{\frac{p-1}{2}}{i} x^{3(\frac{p-1}{2}-i)} a^i =$$

$$\sum_{x=1}^{p-1} x^{3\frac{p-1}{2}} + \binom{\frac{p-1}{2}}{1} a \sum_{x=1}^{p-1} x^{3\frac{p-3}{2}} + \dots + \binom{\frac{p-1}{2}}{\frac{p-3}{2}} a^{\frac{p-3}{2}} \sum_{x=1}^{p-1} x^3 + a^{\frac{p-1}{2}} \sum_{x=1}^{p-1} 1 \quad (1.3)$$

Тъй като последното събираемо по-горе е равно на $a^{\frac{p-1}{2}}(p-1) \equiv -\left(\frac{a}{p}\right) \pmod{p}$, Сравнение 1.2 се опростява до

$$N' \equiv H(a, p) \pmod{p}, \quad (1.4)$$

където изразът за $H(a, p)$ се получава от (1.3) чрез премахване на това последно събираемо. И така, за броя на точките върху елиптична крива, без точката в безкрайността, се получава следната конгруенция:

$$\#E_a - 1 \equiv H(a, p) \pmod{p}, \quad (1.5)$$

където

$$H(a, p) = \sum_{i=0}^{\frac{p-3}{2}} \binom{\frac{p-1}{2}}{i} a^i S_{3i}(p), \quad (1.6)$$

с $l = \frac{p-1}{2} - i$ и суми $S_{3l}(p)$, дефинирани в Твърдение 1.2.8.

След това, при оценката на $H(a, p) \pmod{p}$ с помощта на Твърдение 1.2.8, се забелязва, че участващите степени са кратни само на 3 и лежат в интервала $[3, 3\frac{p-1}{2}]$. Така че има два различни случая, които трябва да се разгледат:

- $p \equiv 5 \pmod{6}$

В този случай от Твърдение 1.2.8 следва, че всички събираеми от дясната страна на Равенство 1.6 са равни на 0 по модул p . Така че $H(a, p) \equiv 0 \pmod{p}$ и в резултат за всяко a е в сила: $\#E_a = p + 1$. Действително това е добре известен факт (вижте, напр., [26] Гл. 18, Пример 1).

- $p \equiv 1 \pmod{6}$

В този съществен случай лесно може да се види, че $H(a, p)$ съдържа точно едно ненулево събираемо по модул p , а именно това, което се получава при $i = \frac{p-1}{6}$. Така че е валидно следното:

$$\begin{aligned} H(a, p) &\equiv \binom{\frac{p-1}{2}}{\frac{p-1}{6}} a^{\frac{p-1}{6}} S_{p-1}(p) \equiv \\ &= \binom{\frac{p-1}{2}}{\frac{p-1}{6}} a^{\frac{p-1}{6}} \pmod{p}. \end{aligned} \quad (1.7)$$

Накрая, заедно с Твърдение 1.3.1, това води до следното:

Теорема 1.3.3 *За просто число $p \geq 19$, такова че $p \equiv 1 \pmod{6}$, важи:*

$$\#E_a = \mathcal{R}(a, p) + p + 1, \quad (1.8)$$

където с $\mathcal{R}(a, p)$ е означен най-малкият по абсолютна стойност остатък на Израз 1.7.

Непосредствено следствие (с изключение на тривиалните случаи $p = 7, 13$) от Твърдение 1.2.5 при $d = 6$ и Теорема 1.3.3 е следващото.

Следствие 1.3.4 Ако p е просто число $\equiv 1 \pmod{6}$, то редът на кривите от \mathcal{E}_p приема точно шест различни стойности, всяка една от които $\frac{p-1}{6}$ пъти в съответствие с шестите корени на единицата в \mathbb{Z}_p^* : $\pm 1, \pm \zeta, \pm(\zeta + \sqrt{-3})$, където $\zeta = \frac{-1-\sqrt{-3}}{2}$.

Забележка 1.3.5 Въпреки че твърдението в Следствие 1.3.4 е известно под една или друга форма (вижте, например, [5]), изглежда, че равномерното разпределение на реда на кривите не е широко обсъждано в литературата.

1.3.2 Изчислителни аспекти на преброяването на точките в \mathcal{E}_p когато p е голямо просто число

Съобразно развитата в предния параграф техника ключова част от пресмятанията е тази на $\left(\frac{\frac{p-1}{2}}{\frac{p-1}{6}}\right) \pmod{p}$. Ефективно решение за това пресмятане може да се намери, като се забележи, че когато p е от вида $6k + 1$, то е в сила:

$$\left(\frac{\frac{p-1}{2}}{\frac{p-1}{6}}\right) = \frac{(2k+1) \dots (3k)}{k!}. \quad (1.9)$$

Твърдение 1.2.10 позволява модулното пресмятане на интересувания ни биномен коефициент да бъде извършено чрез вземане на подходящото X от решенията на квадратното диофантово уравнение $X^2 + 3Y^2 = p$. Едно такова решение може да се намери чрез прилагане на метод, сходен с този, който е описан в [58]. А именно, последният се състои от следните две стъпки:

Стъпка 1. Намиране на квадратен корен от -3 в \mathbb{Z}_p^* ;

Стъпка 2. Намиране на X , прилагайки (частично) алгоритъма на Евклид за p и вече намерения $\sqrt{-3} \in \mathbb{Z}_p^*$.

Както следва от Твърдение 1.2.6, *Стъпка 1* може да бъде изпълнена, ако предварително се знае кубичен неостатък по модул p . Ако за дадено p такъв неостатък не е наличен, той може да бъде намерен след средно 1.5 опита, следвайки Забележка 1.2.7. При всеки опит със случайно избрано цяло число $z \in [2, p-1]$ се пресмята в \mathbb{Z}_p елемента $z' = z^{\frac{p-1}{3}}$ и се проверява дали $z' \neq 1$. Ако това се случи, тогава $2z' + 1$ е един от възможните необходими $\sqrt{-3} \pmod{p}$. Така че взимайки предвид сложността на едно умножение (или повдигане в квадрат) (вижте, например, [22, 12]), очакваното количество работа в *Стъпка 1* е

евристично $\tilde{O}(\log^2 p)$. Освен това *Стъпка 2* е със сложност $O(\log^2 p)$ (вижте например [56] Теорема 3.13 за подробности).

Забележка 1.3.6 Ако $p \equiv 7 \pmod{12}$ лесно може да се види, че $\sqrt{\zeta} = \zeta^{\frac{p+1}{4}}$, където ζ е квадратичен остатък $/(\zeta^{\frac{p+1}{4}})^2 = \zeta^{1+\frac{p-1}{2}} = \zeta \cdot \zeta^{\frac{p-1}{2}} = \zeta/$. Това означава, че има ефективен детерминистичен начин за намиране на квадратен корен в съответното просто поле, а именно като се пресметне $\zeta^{\frac{p+1}{4}}$. В частност той е приложим и за $\zeta = -3$. (вижте [P1] \cong [3]).

Освен това както лесно се вижда от Следствие 1.3.4, шестте възможни различни стойности на втория множител в Израз 1.7, а именно $a^{\frac{p-1}{6}}$, могат да се изразят линейно чрез вече намерения $\sqrt{-3}$.

В заключение, разгледаните съображения доказват валидността на следната теорема:

Теорема 1.3.7 *Общата изчислителна сложност за едновременно намиране на шестте реда, свързани със семейството \mathcal{E}_p , чрез предложената алгоритмична техника, е $\tilde{O}(\log^2 p)$.*

1.3.3 Алгоритми

Сорс кодовете на представените алгоритми са качени в GitHub на адрес: <https://github.com/madmarks/PhD-Thesis-codes>

Алгоритъм 1 представя описаният по-горе подход за пресмятане на един от квадратните корени $\sqrt{-3} \in \mathbb{Z}_p^*$ в случаите, когато $p \equiv 7 \pmod{12}$ (детерминистичен подход), както и когато $p \not\equiv 7 \pmod{12}$ (вероятностен подход):

Algorithm 1: SqrtMinus3(p) - пресмятане на един от квадратните корени $\sqrt{-3} \in \mathbb{Z}_p^*$, $p \equiv 1 \pmod{6}$

Input: $p \equiv 1 \pmod{6}$

Output: един от квадратните корени $\sqrt{-3} \in \mathbb{Z}_p^*$

```

1 if  $p \equiv 7 \pmod{12}$  then
    | /* детерминистичен подход */
2 | return  $(-3)^{\frac{p+1}{4}}$  ;          /*  $\sqrt{-3} \pmod{p}$  (Забележка 1.3.6) */
3 else
    | /* вероятностен подход */
4 | do
5 | |  $z := \text{RandomInt}(2, p-1)$ ;
6 | |  $z' := z^{\frac{p-1}{3}}$ ;
7 | while  $z' \equiv 1 \pmod{p}$ ;
    | /*  $z \notin \mathcal{CR}_p$  */
8 | return  $2z' + 1$  ;          /*  $\sqrt{-3} \pmod{p}$  (Твърдение 1.2.6) */
9 end

```

Алгоритъм 2 представя описаният по-горе подход за едновременно пресмятане на шестте възможни реда, свързани със семейството елиптични криви

$$\mathcal{E}_p = \{E_a : y^2 = x^3 + a \pmod{p}, a \neq 0\},$$

редуцирани по модул p , при фиксирано $p \equiv 1 \pmod{6}$.

Algorithm 2: Едновременно пресмятане на шестте възможни реда, свързани с \mathcal{E}_p , при фиксирано $p \equiv 1 \pmod{6}$

Input: $p \equiv 1 \pmod{6}$

Result: шестте възможни реда, свързани със семейството криви \mathcal{E}_p

```

1  z := SqrtMinus3(p) ;                               /*  $\sqrt{-3} \pmod{p}$  */
2  X,Y := SolveDiophantine( $X^2 + 3Y^2 = p$ );
3  if  $X \not\equiv 1 \pmod{3}$  then
4  |   X := -X ;    /* избираме знака пред X, така че  $X \equiv 1 \pmod{3}$  */
5  end
6  binom_coeff := 2X ;                                  /* пресмятаме  $\left(\frac{p-1}{\frac{p-1}{6}}\right) \pmod{p}$  */

   /* пресмятаме стойностите на множителя  $a^{\frac{p-1}{6}}$  от Израз 1.7,          */
   /* т.е. шестите корени на единицата в  $\mathbb{Z}_p^*$  (Твърдение 1.2.5):          */
   /*  $\pm 1, \pm \zeta_1, \pm \zeta_2$ , където  $\zeta_1 = \frac{-1-\sqrt{-3}}{2}$  и  $\zeta_2 = \zeta_1 + \sqrt{-3}$  */

7   $\mu[1] := +1$ ;
8   $\mu[2] := -1$ ;
9   $\mu[3] := \frac{1+z}{2}$  ;                               /*  $\frac{1+\sqrt{-3}}{2}$  */
10  $\mu[4] := -\mu[3]$  ;                                  /*  $\frac{-1-\sqrt{-3}}{2}$  */
11  $\mu[5] := \frac{1-z}{2}$  ;                               /*  $\frac{1-\sqrt{-3}}{2}$  */
12  $\mu[6] := -\mu[5]$  ;                                  /*  $\frac{-1+\sqrt{-3}}{2}$  */

13 for i in [1,6] do
14 |    $\mathcal{R}[i] := -\text{binom\_coeff} \cdot \mu[i]$  ;          /* пресмятаме Израз 1.7 */
15 |   if  $\mathcal{R}[i] < \frac{p}{2}$  then
16 |   |    $\mathcal{ALR}[i] := \mathcal{R}[i]$ ;
17 |   else
18 |   |    $\mathcal{ALR}[i] := \mathcal{R}[i] - p$ ;
19 |   end
20 end
21 for i in [1,6] do
22 |    $\#E[i] := \mathcal{ALR}[i] + p + 1$  ;                /* пресмятаме шестте реда, */
                                                    /* свързани с фамилията  $\mathcal{E}_p$  */
23 end

```

1.4 Примери

1.4.1 Пример с модула на елиптичната крива secp256k1

За да илюстрираме нашия детерминистичен подход (виж Забележка 1.3.6), представяме пример, в който p е модула на елиптичната крива secp256k1. Както е добре известно, тази крива е дефинирана в „Standards for Efficient Cryptography“ (SEC) [10] и:

$$p = 2^{256} - 2^{32} - 2^9 - 2^8 - 2^7 - 2^6 - 2^4 - 1.$$

Отбелязваме, че $p \equiv 7 \pmod{12}$.

(Числовите данни, представени по-долу, са в десетична бройна система.)

Последователно извършваме следните стъпки:

- пресмятаме квадратен корен от -3 по модул p чрез $(-3)^{\frac{p+1}{4}} \pmod{p}$:

4602937940656409685400179041082242364498080236264115595900560044423621507154;

- решаваме диофантовото уравнение $X^2 + 3Y^2 = p$ и след това пресмятаме $\left(\frac{\frac{p-1}{2}}{\frac{p-1}{6}}\right) \pmod{p} = \pm 2X$, където знакът е такъв, че $\pm X \equiv 1 \pmod{3}$:

671331852483699643819086596696745227420;

- пресмятаме $\zeta_1 = \frac{-1-\sqrt{-3}}{2}$ and $\zeta_2 = \zeta_1 + \sqrt{-3}$:

60197513588986302554485582024885075108884032450952339817679072026166228089408,
55594575648329892869085402983802832744385952214688224221778511981742606582254;

- Пресмятаме стойностите на Израз 1.7, като използваме ± 1 , $\pm \zeta_1$ и $\pm \zeta_2$ в ролята на множителя $a^{\frac{p-1}{6}}$. След това, използваме Теорема 1.3.3, за да намерим шестте реда, свързани с фамилията \mathcal{E}_p :

115792089237316195423570985008687907852598652813156864395638497411212089444244,
115792089237316195423570985008687907853702405052206223696310004874299507848991,
115792089237316195423570985008687907853508896131558604026424249738214906721757,
115792089237316195423570985008687907853941316518124263683276670604605579899084,
115792089237316195423570985008687907852837564279074904382605163141518161494337,
115792089237316195423570985008687907853031073199722524052490918277602762621571.

24 $\tilde{O}(\log^2 p)$ метод за преброяване на точките върху $y^2 = x^3 + a \pmod{p}$, $a \neq 0$

- Разлагаме на множители получените редове, използвайки метода GMP – ECM, имплементиран в SAGE (SageMath). Техните най-големи прости делители са съответно:

5669387787833452836421905244327672652059,
1013176677300131846900870239606035638738100997248092069256697437031,
211853322379233867315890044223858703031485253961775684523,
1992751017769525324118900703535975744264170999967,
115792089237316195423570985008687907852837564279074904382605163141518161494337,
41245443549316649091297836755593555342121.

Накрая си заслужава да се отбележи, че известната стойност за елиптичната крива на Bitcoin: $\#E_7$ се потвърждава (предпоследната по-горе), както и това, че тя е единственото просто число измежду пресметнатите редове.

1.4.2 Пример с модул просто число, намерено от нас

Представеният тук пример илюстрира нашия *вероятностен* подход. За модул е избрано следното просто число

$$p = 2^{256} + 2^{56} + 2^{44} + 1,$$

което е сравнимо с 1 по модул 12.

Забележка 1.4.1 Това просто число е намерено чрез случайно търсене и използване на теста за простота APR-CL, имплементиран в SAGE (SageMath).

(Числовите данни, представени по-долу, са в шестнадесетична бройна система.)

Последователно изпълняваме следните стъпки:

- Пресмятаме $\pm\sqrt{-3} \pmod{p}$:

- генерираме случайно число $z \in [2, p-1]$:

D03CE6183277A6719AD52F362919187D79CA6A50D000975EC4903782F08DAAEB

- пресмятаме $z' = z^{\frac{p-1}{3}}$:

1AF6F28A07E19B1F1FCCB53D1A166E14092BC3AF668514D3D2ABE80CBD6C0A16

- $z' \neq 1$, и така $2z' + 1$ е корен квадратен от -3 в \mathbb{F}_p^* :

35EDE5140FC3363E3F996A7A342CDC281257875ECD0A29A7A557D0197AD8142D;

- Намираме X чрез прилагане на алгоритъма на Евклид за p и намерения $\sqrt{-3} \in \mathbb{Z}_p^*$, и пресмятаме $\left(\frac{\frac{p-1}{2}}{6}\right) \pmod{p} = 2X$ (вижте Равенство 1.9 и Твърдение 1.2.10):

[illegible]

- Пресмятаме $\zeta_1 = \frac{-1-\sqrt{-3}}{2}$ и $\zeta_2 = \zeta_1 + \sqrt{-3}$:

E5090D75F81E64E0E0334AC2E5E991EBF6D43C50997AEB2C2E5427F34293F5EA,
1AF6F28A07E19B1F1FCCB53D1A166E14092BC3AF668514D3D2ABE80CBD6C0A16;

- Пресмятаме стойностите на Израз 1.7, като използваме $\pm 1, \pm \zeta_1$ и $\pm \zeta_2$ в ролята на множителя $a^{\frac{p-1}{6}}$. Накрая намираме шестте реда, свързани с фамилията \mathcal{E}_p :

[illegible]

Подлагайки горните числа на теста за простота $\text{APR} - \text{CL}$, откриваме наличието на два прости реда (отбелязани със звезда), които съответстват на броя на точките върху кривите E_{31} и E_{11} .

1.4.3 Сравнение на ефективността спрямо алгоритъма SEA

Напомниме, че теоретичната оценка за сложността на алгоритъма SEA е $\tilde{O}(\log^4 p)$, докато нашата алгоритмична процедура е със сложност $\tilde{O}(\log^2 p)$ (вижте Теорема 1.3.7).

При проведените програмни експерименти за сравняване ефективността на алгоритъма SEA спрямо предложения, и двата работещи върху разгледания случай на задачата, е използван стандартен лаптоп с процесор Intel Core i7-6820HQ при 2.7 GHz (четири ядра). Реализацията на предложения алгоритъм е написана на

26 $\tilde{O}(\log^2 p)$ метод за преброяване на точките върху $y^2 = x^3 + a \pmod{p}$, $a \neq 0$

Python, докато тази на SEA е високо оптимизиран код на C от системата за компютърна алгебра PARI/GP, предназначена специално за бързи изчисления в теорията на числата.

В Таблица 1.1 е даден списък на 257-битовите прости числа, използвани като входни данни за експериментите.

Таблица 1.1 Прости числа p_i , $i = 1, \dots, 10$

p_i	Prime Number (HEX)
p_1	1744AA82FB357A0A99A571EABF8E72B860517859044F993E2606ECA7BC6CB169
p_2	1032FAF22DC31F3E339E3F0CAC8BF44F21B383D3A687A41326A4CC77EAC31D881
p_3	19C7E604E23D3DEF8A371353FD8EFA4C9F7503083CD2FCE2EA7FEF1120EC3B3E9
p_4	1750F9C8F1490EEDC1B05F0CA012ED4B42925C588AA5FFCC285F84E802EA71C65
p_5	161D8802C08AC9AB133B20100B50C4CF1710A7BEDBA3292B56567D996DE3CEF4D
p_6	1BF6DA0DA929F9784E07C6835AD78389B06CBD5FB776F9F2371AC79B7C7FC1B6D
p_7	1946A87890B83A015439E75B2BA2C20C9D742E7A85B592815A5D6C11DDACD4695
p_8	1819AA8747CF5595260B5A3D7FF8E800DD365E21E26DEBC306F7E48B12C2E2A29
p_9	18864DC62E42429367F6826C5F2AAF1401875EA94E1DA3D70DB1BB7D049F90525
p_{10}	1304670800156954405D850ABD3086D0E8AC7B898E4CC9F18000CF2B9087DBD15

Както се вижда от Таблица 1.2, нашият метод е между 20 и 67 пъти по-бърз, въпреки че реализацията не е оптимизирана.

Таблица 1.2 Сравнение на ефективността

Test №	Prime p_i	SEA Execution Time (ms)	Our Method Execution Time (ms)
1	p_1	829.7	12.4
2	p_2	251.8	12.3
3	p_3	636.4	12.2
4	p_4	430.9	11.5
5	p_5	436.8	11.1
6	p_6	284.7	12.3
7	p_7	355.4	10.9
8	p_8	558.0	12.2
9	p_9	398.1	11.1
10	p_{10}	393.2	11.1

При проведен експеримент със случайно 857-битово просто число, SEA изчислява реда за 22,5 секунди, докато разработената от нас програма извършва работата за 33,1 милисекунди, т.е. почти 680 пъти по-бързо. Това показва, че методът е много по-ефективен от алгоритъма SEA за големи прости числа, да кажем, над 800 бита.

1.5 Заключение

В съвременната литература са описани формули за пресмятане на реда на елиптични криви над крайни полета (вижте например [57, 49, 29], и т.н.). В тази глава е изведена явна формула за реда на крива от семейството

$$\mathcal{E}_p = \{E_a : y^2 = x^3 + a \pmod{p}, a \neq 0\},$$

редуциран по модул p . Освен това описаният подход позволява определяне на спектъра от редове при фиксирано $p \equiv 1 \pmod{6}$, както и предказва съответния известен факт в допълващия случай $p \equiv 5 \pmod{6}$. Освен това въз основа на класическите резултати за квадратични разбивания на прости числа е описан ефективен алгоритъм (със сложност $\tilde{O}(\log^2 p)$) за едновременно пресмятане на

шестте реда, свързани с \mathcal{E}_p . Експерименталните резултати потвърждават теоретичните оценки за ефективност, с очаквани леки отклонения поради все още неоптимизираната реализация. Тази техника подобрява с почти един порядък най-доброто известно досега алгоритмично решение, предложено от Мунуега и Тена в [43] през 1993г., позволявайки при същите разходи да се постигнат стойности на параметъра p , характерни за ECC системи с по-висока сигурност. Тя е особено полезна, когато се търсят (например случайно) елиптични криви от прост ред, принадлежащи към семейства от разглеждания тип, когато модулът p варира.

Резултатите, получени в тази глава, са докладвани на „2019 Ninth International Workshop on Signal Design and its Applications in Communications (IWSDA), Dongguan, China, 20-24 October 2019“ [T1], както и на „Национален семинар по теория на кодирането ”Професор Стефан Додунев“, Чифлика, България, 21-24 Ноември 2019“ [T2].

Глава 2

Метод за преброяване на точките върху елиптични криви от вида

$$y^2 = x^3 + ax \pmod{p}, \quad a \neq 0$$

Изследването, което се съдържа в тази глава, се основава на работата, представена от нас в статията със заглавие „*Point-Counting on Elliptic Curves Belonging to One Prominent Family: Revisited*“ [P2] \cong [37].

2.1 Въведение

Съществува ефективен алгоритъм (SEA), който изчислява реда на дадена елиптична крива от общ тип (вижте например [47]). В тази глава обаче интерес представлява цялото семейството от криви

$$\mathcal{D}_p = \{D_a : y^2 = x^3 + ax \pmod{p}, \quad a \neq 0\}$$

с мощност $p - 1$, и директното прилагане на споменатия алгоритъм, за намиране редовете на всички криви в \mathcal{D}_p , не е осъществимо при големи стойности на p . Затова са необходими затворени алгебрични формули, по отношение на параметрите a и p , които решават интересувашата ни задача.

Аналитично решение на задачата е представено в [57, Раздел 4.4]. Въпреки това, Теорема 4.23 от [57], която обобщава резултатите, оставя празнота относно знака на участващата следа на Фробениус за някои стойности на параметъра k . Вероятно уточнението на знака е оставено като упражнение за читателите

на учебника [57], тъй като коментарът на автора е „това е много по-деликатен проблем и го пропускаме“.

Едно алгоритмично решение, за което научихме неотдавна, е дадено в [43], а именно за случая на елиптични криви с j -инвариант 1728. Но както авторите на [43] са отбелязали, техният алгоритъм изисква $O(\log^3 p)$ битови операции, докато нашият подход подобрява сложността до $\tilde{O}(\log^2 p)$.

Също така следва да се отбележи, че подходът, следван в настоящата глава, е подобен на решението на същата задача, относно семейството от елиптични криви

$$\mathcal{E}_p = \{E_a : y^2 = x^3 + ax \pmod{p} \mid a \neq 0\}$$

$[P1] \cong [3]$, и работи еднакво успешно и в двата случая, тъй като получените резултати са изчерпателни и компактни, въпреки че използват някои отдавна установени факти от теорията на квадратичните разбивания на прости числа. Поради тази причина главата е по-кратка.

2.2 Встъпителни бележки

Целта ни е да опишем подход за определяне на реда на обща крива D_a от семейството \mathcal{D}_p в зависимост от a и p .

За удобство на читателите припомняме едно основно свойство на символа на Лежандър, изложено тук като лема.

Лема 2.2.1 *За всяко цяло число z е в сила: $\left(\frac{z}{p}\right) \equiv z^{\frac{p-1}{2}} \pmod{p}$.*

Лема 2.2.1 е общоизвестна като критерий на Ойлер за определяне дали цялото число z е квадратичен остатък по модул p .

Ще припомним също един по-малко известен отдавнашен, но полезен факт, дължащ се на К.Ф. Гаус (вижте, [16, том II, стр. 234] за исторически подробности).

Твърдение 2.2.2 (Наблюдение на Гаус) *Ако простото число $p = 4k + 1$ се представи във вида $X^2 + Y^2$, където X е нечетно, а Y е четно, то $\pm X$ е равно на най-малкия по абсолютна стойност остатък (т.е., \mathcal{ALR}) по модул p на*

$$\frac{1}{2} \frac{(k+1) \dots (2k)}{k!},$$

и този остатък е положителен или отрицателен в зависимост от това дали положителната стойност на X е от вида $4m + 1$ или $4m + 3$.

Скорошна дискусия относно това наблюдение на Гаус може да бъде намерена в [35, стр. 192, 202] (вижте също [26, Гл. 8, Упр. 26]).

2.3 Описание на подхода

Най-напред ще напомним, че Твърдение 1.3.1 позволява недвусмислено да се намери броя N на точките върху дадена елиптична крива, при условие че може да се пресметне минималният по абсолютна стойност остатък на $(N - 1)$ по модул p , означаващ като $\mathcal{ACR}(N - 1, p)$.

2.3.1 Явна формула за реда на елиптичната крива $D_a \in \mathcal{D}_p$, редуциран по модул p

Нека, за удобство, $N' = \#D_a - 1$ за фиксирана крива $D_a \in \mathcal{D}_p$. Действайки по сходен начин, както в Раздел 1.3.1, лесно получаваме:

$$N' \equiv h(a, p) \pmod{p}, \quad (2.1)$$

където

$$h(a, p) = \sum_{i=0}^{\frac{p-1}{2}} \binom{\frac{p-1}{2}}{i} a^i S_{k(i)}(p). \quad (2.2)$$

Тук са използвани означенията $S_{k(i)}(p)$ за сумите от степените на целите положителни числа, въведени в Твърдение 1.2.8, където $k(i) = 3^{\frac{p-1}{2}} - 2i$.

По-нататък оценяваме $h(a, p) \pmod{p}$, като използваме Твърдение 1.2.8 и забелязваме, че въвлечените степени са само нечетните (или само четните) цели числа от интервала $[\frac{p-1}{2}, 3^{\frac{p-1}{2}}]$. Следователно има два различни случая, които трябва да бъдат разгледани:

- $p \equiv 3 \pmod{4}$

В този случай няма индекс i , за който $k(i) = p - 1$, и от Твърдение 1.2.8 следва, че всички събираеми в дясната страна на уравнение (2.2), редуцирани по модул p , са равни на 0. Така, че $h(a, p) \equiv 0 \pmod{p}$, и Сравнение (2.1) заедно с границата на Хасе водят до $N' = p$. Следователно за всяко a е

изпълнено $\#D_a = p + 1$, което е добре известен факт (вижте, например, [26, Гл. 18.4, Теорема 5]);

- $p \equiv 1 \pmod{4}$

В този съществен случай е лесно да се види, че изразът (2.2) съдържа точно едно ненулево събираемо, а именно когато $i = \frac{p-1}{4}$. Следователно според Твърдение 1.2.8 е валидно:

$$h(a, p) \equiv \left(\frac{\frac{p-1}{2}}{\frac{p-1}{4}}\right) a^{\frac{p-1}{4}} S_{p-1}(p) \equiv -\left(\frac{\frac{p-1}{2}}{\frac{p-1}{4}}\right) a^{\frac{p-1}{4}} \pmod{p},$$

или според Сравнение (2.1) еквивалентното:

$$N' \equiv -\left(\frac{\frac{p-1}{2}}{\frac{p-1}{4}}\right) a^{\frac{p-1}{4}} \pmod{p}. \quad (2.3)$$

Накрая, от Сравнение (2.3), Твърдение 1.3.1 и забележката след него, директно извеждаме основния резултат на тази глава.

Теорема 2.3.1 *За просто число $p \geq 17$, такова че $p \equiv 1 \pmod{4}$, важи:*

$$\#D_a = \begin{cases} p + 1 + \mathcal{R}(a, p), & \text{ако } \mathcal{R}(a, p) < \frac{p}{2} \\ 1 + \mathcal{R}(a, p), & \text{в противен случай,} \end{cases} \quad (2.4)$$

където $\mathcal{R}(a, p)$ представлява най-малкия неотрицателен остатък на дясната страна на Сравнение (2.3).

2.3.2 Спектърът на $\#D_a$, когато a варира в \mathbb{Z}_p^*

Следното твърдение характеризира по-прецизно възможните редове на криви в съществения случай $p \equiv 1 \pmod{4}$.

Твърдение 2.3.2 *Ако $p \equiv 1 \pmod{4}$ е просто число ≥ 17 , то редът на кривите от \mathcal{D}_p приема точно четири различни стойности, всяка една от които $\frac{p-1}{4}$ пъти, в съответствие с четвъртите корени на единицата в \mathbb{Z}_p^* : $\pm 1, \pm \sqrt{-1}$.*

Доказателство: Сравнение (2.3) показва, че за фиксирано p редът на крива $D_a \in \mathcal{D}_p$ се определя от втория множител $a^{\frac{p-1}{4}}$. Съгласно Твърдение 1.2.5

последният приема стойности, съответстващи на четирите различни четвърти корени на единицата, когато a варира в \mathbb{Z}_p^* . \square

Забележка 2.3.3 *Като непосредствено следствие от Теорема 4.23 в [57], може да се изведе, че редът на всяка елиптична крива от семейството \mathcal{D}_p винаги е четно число, което не може да се види директно от Теорема 2.3.1.*

2.3.3 Изчислителни аспекти на преброяването на точките в \mathcal{D}_p , когато p е голямо просто число

По-долу е описан ефективен подход за пресмятане на редовете на кривите в \mathcal{D}_p , когато p е голямо просто число.

Ключов момент в този изчислителен процес е пресмятането на $\left(\frac{\frac{p-1}{2}}{\frac{p-1}{4}}\right) \pmod{p}$. Тази задача може да бъде решена с помощта на Твърдение 2.2.2, като забележим, че ако p е от вида $p = 4k + 1$, то тогава е валидно:

$$\left(\frac{\frac{p-1}{2}}{\frac{p-1}{4}}\right) = \frac{(k+1) \dots (2k)}{k!}.$$

Следователно, това твърдение позволява модулното пресмятане на интересувания ни биномен коефициент да се извърши чрез решаване на квадратното диофантово уравнение $X^2 + Y^2 = p$. Решението може да бъде намерено чрез метод, представен в [7], който е значително опростен в сравнение с този, открит по-рано от Сера и Ермит [48, 23]. Накратко, методът се състои от следните две стъпки:

- *Стъпка 1.* Намиране на квадратен корен от -1 в \mathbb{Z}_p^* ;
- *Стъпка 2.* Намиране на нечетното X чрез прилагане на алгоритъма на Евклид за p и вече намерения $\sqrt{-1} \in \mathbb{Z}_p^*$.

Стъпка 1 може да бъде изпълнена ефективно, ако е известен предварително квадратичен неостатък по модул p (за подробности как да се избере или намери такъв неостатък, вижте например [7, *Забележка, стр.1012*]).

Тук ще дискутираме един друг общ подход това да се осъществи. А именно, въз основа на Твърдение 1.2.5 (Лема 2.2.1), квадратен корен от -1 по модул p може да бъде намерен след средно два опита, при условие че е наличен висококачествен генератор на случайни цели числа в интервала $I_p = [2, p-1]$. Всеки един от тези опити се състои от пресмятане, за произволно избрано $\mathcal{R} \in I_p$, на елемента

$\mathcal{R}' = \mathcal{R}^{\frac{p-1}{4}}$, както и проверка дали последният е различен от ± 1 . Ако това се случи, то \mathcal{R}' е търсеният $\sqrt{-1}$. (Читателят се препраща към [59, Гл. 10] или към ранната работа [1] за допълнително обосноваване на този вероятностен подход.) Грубо казано, работата в *Стъпка 1* е пропорционална на $\log p \log^2 p$. Да напомним също, че *Стъпка 2* има горна граница на изчислителната сложност $O(\log^2 p)$, което вече беше споменато в **Глава 1** (вижте например [56, Теорема 3.13]).

Имайки предвид Твърдение 2.3.2, за да намерим едновременно четирите реда, свързани със семейството \mathcal{D}_p , може да следваме следния подход. След като пресметнем $2X$, умножаваме резултата по вече намерения $\sqrt{-1}$ и след това вземаме противоположните стойности по модул p . В заключение, общата сложност на задачата за определяне на интересующите ни редове е $\tilde{O}(\log^2 p)$.

2.4 Пример

Ще илюстрираме описания по-горе подход с пример, използвайки простото число

$$p = 2^{224} - 2^{96} + 1.$$

Да отбележим, че това просто число се използва за модул в параметрите на елиптичната крива `secp224r1` (спецификация на NIST за 224-битова елиптична крива).

Числовите данни, представени тук, са в десетична бройна система. Последователно се изпълняват следните стъпки:

- Пресмятаме $\sqrt{-1} \pmod{p}$, като прилагаме вероятностния подход, описан в предишния раздел:

- намираме случайно число \mathcal{R} :

3014128213946592536941101147799016574912189298231161892015573

- пресмятаме $\mathcal{R}' = \mathcal{R}^{\frac{p-1}{4}}$:

3338362603553219996874421406887633712040719456283732096017030791656

- тъй като $\mathcal{R}' \not\equiv \pm 1 \pmod{p}$, то \mathcal{R}' е $\sqrt{-1} \pmod{p}$.

- Решаваме диофантовото уравнение $X^2 + Y^2 = p$, вземаме нечетната положителна стойност на X и пресмятаме $\left(\frac{p-1}{4}\right) \pmod{p} = 2X$:

5789010730181395098356382620729282

- Пресмятаме стойностите на израза (2.3), използвайки ± 1 и $\pm\sqrt{-1}$ в ролята на множителя $a^{\frac{p-1}{4}}$. Накрая намираме четирите реда, свързани с фамилията \mathcal{D}_p :

26959946667150639794667015087019624884547186078631209787127445569600,
 26959946667150639794667015087019636462568646441421406499892687028164,
 26959946667150639794667015087019622052238908448795226442971178695282,
 26959946667150639794667015087019639294876924071257389844048953902482

Струва си да се отбележи, че редът на последната крива е четно полупросто число, т.е. е равен на два пъти просто число.

2.5 Заключение

Преразглеждайки задачата за пресмятане реда на елиптична крива от семейството \mathcal{D}_p , намерихме опростена явна формула, когато този ред се редуцира по модул p . Въпросната формула, в комбинация със знаменитата граница на Хасе, решава задачата изчерпателно и кратко. Освен това, въз основа на класическите резултати за квадратичните разбивания на прости числа, описваме ефективна техника, със сложност най-много $\tilde{O}(\log^2 p)$, за едновременно пресмятане на четирите възможни редове на кривите от $\mathcal{D}_p, p \equiv 1 \pmod{4}$. Следователно предложеният подход подобрява най-доброто известно досега решение почти на порядък.

Резултатите, получени в тази глава, са докладвани на „17th International Workshop on Algebraic and Combinatorial Coding Theory (ACCT), online, Bulgaria, 11-17 October 2020“ [T3].

Глава 3

Теглово разпределение на двоичния код на Рид-Малер $\mathcal{R}(4, 9)$

Изследването, което се съдържа в тази глава, се основава на работата, представена от нас в статията със заглавие „*Weight Distribution of the Binary Reed-Muller Code $R(4, 9)$* “ [P4] \cong [38].

3.1 Основни понятия

Нека $\mathbb{F}_2 = \{0, 1\}$ е крайно поле с два елемента, а \mathbb{F}_2^n е n -мерното векторно пространство над \mathbb{F}_2 . Булева функция f на n променливи е изображение на \mathbb{F}_2^n в \mathbb{F}_2 . Нека с B_n да означим множеството от всички булеви функции на n променливи. Да отбележим, че мощността му е $|B_n| = 2^{2^n}$. Булевите функции имат различни представяния, някои от които канонични (вектор), а други не (формули, схеми от функционални елементи, хиперкуб).

Всяка булева функция на n променливи може да бъде представена като множество от 2^n наредени двойки, първият елемент на които е n -битов вектор, т.е. елемент на \mathbb{F}_2^n , а вторият, булева стойност (0 или 1).

Ще дадем един пример за булева функция на 3 променливи:

$$f = \{[(0, 0, 0), 1], [(0, 0, 1), 1], \\ [(0, 1, 0), 1], [(0, 1, 1), 0], \\ [(1, 0, 0), 1], [(1, 0, 1), 0], \\ [(1, 1, 0), 0], [(1, 1, 1), 1]\}.$$

Можем да опишем функцията f много по-прегледно с таблица, наречена *таблица на истинност*:

Таблица 3.1 Таблицата на истинност на f

x_1	x_2	x_3	f
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Тъй като входните вектори са подредени отгоре надолу лексикографски, не е необходимо да ги пишем. За да определим функцията е достатъчно да използваме само колоната от нейните стойности.

Таблицата на истинност на булева функция f може да бъде представена чрез вектор, чиито координати са стойностите на функцията за всеки n -битов вход $T_f = \{f(0, \dots, 0), f(0, \dots, 0, 1), \dots, f(1, \dots, 1)\}$, допускайки имплицитно, че входните вектори са подредени лексикографски. Това е *каноничното* представяне на булева функция. Броят на единиците в последния стълб на таблицата на истинност T_f се нарича *Хемингово тегло* на f , означено с $wt(f)$. *Хеминговото разстояние* между две булеви функции $f, g \in B_n$ се означава с $d(f, g)$ и е равно на $wt(f + g)$. Това означава, че *Хеминговото разстояние* е мощността на множеството $\{x \in \mathbb{F}_2^n \mid f(x) \neq g(x)\}$.

Булевите функции се представят също така уникално и чрез полиноми на много променливи, наречени *алгебрична нормална форма* (ANF):

$$f(x_1, \dots, x_n) = \sum_{u \in \mathbb{F}_2^n} a_u x^u, \quad (3.1)$$

където $x^u = x_1^{u_1} x_2^{u_2} \dots x_n^{u_n}$ е *моном*, съставен от променливите за които $u_i = 1$ и $a_u \in \mathbb{F}_2$. Това представяне се нарича също така *полином на Жегалкин* на f .

Дефиниция 3.1.1 *Степен на булева функция, означена с d_f , е степента на монома с най-висока степен в ANF представянето.*

За по-задълбочени познания относно булевите функции и тяхното приложение в теория на кодирането и криптографията, насочваме читателя към [8] и [15].

Дефиниция 3.1.2 *Тегло (по Хеминг) на двоичен вектор $\underline{x} = (x_1, x_2, \dots, x_n) \in \mathbb{F}_2^n$, означаващо с $wt(\underline{x})$, се нарича броят на ненулевите координати на вектора, т.е. на координатите, равни на 1:*

$$wt(\underline{x}) = |\{i \mid x_i = 1\}|.$$

Дефиниция 3.1.3 *Разстояние (по Хеминг) между два двоични вектора $\underline{x} = (x_1, x_2, \dots, x_n)$ и $\underline{y} = (y_1, y_2, \dots, y_n)$ от \mathbb{F}_2^n , означаващо с $d(\underline{x}, \underline{y})$, се нарича броят на координатите, в които те се различават:*

$$d(\underline{x}, \underline{y}) = |\{i \mid x_i \neq y_i\}|.$$

Дефиниция 3.1.4 *Под сума на векторите $\underline{x} = (x_1, x_2, \dots, x_n)$ и $\underline{y} = (y_1, y_2, \dots, y_n)$ от \mathbb{F}_2^n , означаваща с $\underline{x} + \underline{y}$, ще разбираме вектора*

$$\underline{x} + \underline{y} = (x_1 + y_1, x_2 + y_2, \dots, x_n + y_n),$$

принадлежащ на \mathbb{F}_2^n .

Дефиниция 3.1.5 *Под скалярно произведение на векторите $\underline{x} = (x_1, x_2, \dots, x_n)$ и $\underline{y} = (y_1, y_2, \dots, y_n)$ от \mathbb{F}_2^n , означаващо с $(\underline{x}, \underline{y})$ или $\underline{x} \cdot \underline{y}$, ще разбираме елемента*

$$(\underline{x}, \underline{y}) = \underline{x} \cdot \underline{y} = x_1 y_1 + x_2 y_2 + \dots + x_n y_n,$$

принадлежащ на \mathbb{F}_2 .

Два вектора са *ортогонални*, ако скалярното им произведение е равно на 0.

Дефиниция 3.1.6 *Под умножение на вектор $\underline{x} = (x_1, x_2, \dots, x_n)$ от \mathbb{F}_2^n със скалар $a \in \mathbb{F}_2$, означаващо с $a\underline{x}$, ще разбираме вектора*

$$a\underline{x} = (ax_1, ax_2, \dots, ax_n),$$

принадлежащ на \mathbb{F}_2^n .

Теорема 3.1.7 (Критерий за подпространство) Множеството $C \subseteq \mathbb{F}_2^n$ е подпространство на \mathbb{F}_2^n тогава и само тогава, когато са изпълнени следните две условия:

- (1) $\forall \underline{x}, \underline{y} \in C \Rightarrow \underline{x} + \underline{y} \in C$
- (2) $\forall a \in \mathbb{F}_2$ и $\forall \underline{x} \in C \Rightarrow a\underline{x} \in C$

Дефиниция 3.1.8 Ако $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_k$ са вектори от \mathbb{F}_2^n и a_1, a_2, \dots, a_k са скалари от \mathbb{F}_2 , то векторът

$$a_1\underline{x}_1 + a_2\underline{x}_2 + \dots + a_k\underline{x}_k \in \mathbb{F}_2^n$$

се нарича линейна комбинация на $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_k$ с координати a_1, a_2, \dots, a_k .

Теорема 3.1.9 Ако $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_k$ са вектори от \mathbb{F}_2^n , то множеството от всички линейни комбинации на $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_k$ е подпространство на \mathbb{F}_2^n .

Подпространството от Теорема 3.1.9 се нарича линейна обвивка на $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_k$.

Дефиниция 3.1.10 Векторите $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_k \in \mathbb{F}_2^n$ се наричат линейно зависими, ако съществуват скалари $a_1, a_2, \dots, a_k \in \mathbb{F}_2$, поне единият от които е различен от 0, за които

$$a_1\underline{x}_1 + a_2\underline{x}_2 + \dots + a_k\underline{x}_k = 0$$

Ако $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_k$ не са линейно зависими, то те се наричат линейно независими.

Дефиниция 3.1.11 Нека C е подпространство на \mathbb{F}_2^n . Множеството от вектори $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_k \in C$ се нарича пораждащо за C , ако всеки вектор на C може да се представи като линейна комбинация на $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_k$.

Дефиниция 3.1.12 Базис на подпространството $C \subseteq \mathbb{F}_2^n$ наричаме всяко пораждащо множество за C , което се състои от линейно независими вектори.

Теорема 3.1.13 Ако $C \subseteq \mathbb{F}_2^n$ и $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_k \in C$ е базис на C , то:

- (1) всеки вектор от C може да се представи еднозначно като линейна комбинация на базисните вектори $\underline{x}_1, \underline{x}_2, \dots, \underline{x}_k$;
- (2) C съдържа точно 2^k вектора.

От Теорема 3.1.13 следва, че ако $C \subseteq \mathbb{F}_2^n$ е подпространство с 2^k елемента, то всеки базис на C съдържа точно k линейно независими вектора.

Дефиниция 3.1.14 Броят на векторите в който да е базис на подпространството $C \subseteq \mathbb{F}_2^n$ се нарича размерност на C и се означава с $\dim(C)$.

Следователно $|C| = 2^k \iff \dim(C) = k$ и, в частност, $\dim(\mathbb{F}_2^n) = n$, тъй като $|\mathbb{F}_2^n| = 2^n$.

Нека $V = \mathbb{F}_2^n$ е n -мерното векторно пространство над крайното поле $\mathbb{F}_2 = \{0, 1\}$. Всяко непразно подмножество \mathbf{C} на V ще наричаме двоичен код с дължина n . Векторите от \mathbf{C} се наричат *кодими думи*. В частност, всяко k -мерно линейно подпространство на V се нарича двоичен линейен код с дължина n и размерност k и се означава като $[n, k]$ -код.

Като източник за част от следващите дефиниции е използван [6].

Дефиниция 3.1.15 Пораждаща матрица на линейен код \mathbf{C} с дължина n и размерност k наричаме всяка матрица с k реда и n стълба с елементи от полето \mathbb{F}_2 , чиито редове образуват базис на \mathbf{C} .

Дефиниция 3.1.16 Дуален код \mathbf{C}^\perp на линейния код \mathbf{C} наричаме ортогоналното допълнение на \mathbf{C} , т.е.

$$\mathbf{C}^\perp = \{\underline{y} \in \mathbb{F}_2^n \mid (\underline{x}, \underline{y}) = 0, \forall \underline{x} \in \mathbf{C}\}.$$

Да отбележим, че е в сила равенството $\dim(\mathbf{C}) + \dim(\mathbf{C}^\perp) = n$ (Теорема за ранга и дефекта).

Дефиниция 3.1.17 Наричаме линейния код \mathbf{C} самоортогонален, ако $\mathbf{C} \subseteq \mathbf{C}^\perp$, и самодуален, когато $\mathbf{C} = \mathbf{C}^\perp$.

Дефиниция 3.1.18 Проверочна матрица H на кода \mathbf{C} наричаме всяка пораждаща матрица на дуалния му код \mathbf{C}^\perp .

Дефиниция 3.1.19 Нека \mathbf{C} е линейен $[n, k]$ код над \mathbb{F}_2 и $\underline{x} \in \mathbb{F}_2^n$ е произволен вектор. Тогава множеството

$$\underline{x} + \mathbf{C} = \{\underline{x} + \underline{c} \mid \underline{c} \in \mathbf{C}\}$$

се нарича съседен клас на \mathbf{C} .

Дефиниция 3.1.20 *Лидер на съседен клас наричаме вектор с минимално тегло в съседния клас, т.е. всеки вектор $\underline{y} \in \underline{x} + \mathbf{C}$ може да е лидер на съседния клас, ако $wt(\underline{y}) = \min\{wt(\underline{x} + \underline{c}) \mid \underline{c} \in \mathbf{C}\}$.*

Теорема 3.1.21 (Теорема на Лагранж) *Нека \mathbf{C} е линеен $[n, k]$ код над \mathbb{F}_2 . Тогава за съседните класове на \mathbf{C} са в сила следните твърдения:*

- (1) *Всеки вектор на \mathbb{F}_2^n принадлежи на съседен клас на \mathbf{C} ;*
- (2) *Всеки съседен клас съдържа точно $2^k = |\mathbf{C}|$ вектора;*
- (3) *Два съседни класа или съвпадат или въобще не се пресичат.*

Дефиниция 3.1.22 *Теглово разпределение на кода \mathbf{C} с дължина n е векторът $W(\mathbf{C}) = (W_0, \dots, W_n)$, където W_i е броят на кодовите думи с тегло по Хеминг i .*

Дефиниция 3.1.23 *Спектър на кода \mathbf{C} с теглово разпределение $W(\mathbf{C}) = (W_0, \dots, W_n)$ е множеството от индекси $\{i : 0 \leq i \leq n, W_i > 0\}$.*

Дефиниция 3.1.24 *Номератор на теглата в кода \mathbf{C} , притежаващ теглово разпределение $W(\mathbf{C}) = (W_0, \dots, W_n)$, се дефинира като следния полином (с цели неотрицателни коефициенти) на променливата z :*

$$\mathcal{W}[z; \mathbf{C}] = \sum_{i=0}^n W_i z^i.$$

Понякога за номератор на теглата се използва и следната дефиниция:

Дефиниция 3.1.25 *Номератор на теглата в кода \mathbf{C} , притежаващ теглово разпределение $W(\mathbf{C}) = (W_0, \dots, W_n)$, се дефинира като следния полином (с цели неотрицателни коефициенти) на две променливи:*

$$\mathcal{W}[x, y; \mathbf{C}] = \sum_{i=0}^n W_i x^{n-i} y^i.$$

Добре известно е, че за коригиращата способност на линеен код от особена важност е най-малкото ненулево тегло на кодови думи, което се нарича минимално тегло на кода. Когато двоичен линеен $[n, k]$ -код има минимално тегло d , той се означава като $[n, k, d]$ -код.

Тук се интересуваме от следната $(\underline{u}, \underline{u} + \underline{v})$ конструкция за получаване на нов код от известни кодове [36, Ch.2.9].

Дефиниция 3.1.26 При дадени $[n, k_1, d_1]$ -код \mathbf{C}_1 и $[n, k_2, d_2]$ -код \mathbf{C}_2 , с еднакви дължини n на кодовите думи, можем да конструираме нов код \mathbf{C}_3 с дължина $2n$, състоящ се от всички вектори

$$(\underline{u}, \underline{u} + \underline{v}), \quad \underline{u} \in \mathbf{C}_1, \quad \underline{v} \in \mathbf{C}_2.$$

Теорема 3.1.27 ([36, Ch.2.9]) \mathbf{C}_3 е $[2n, k_1 k_2, d = \min\{2d_1, d_2\}]$ -код.

Задачата, която изучаваме в тази глава, касае една много популярна фамилия от кодове, така наречените кодове на Рид-Малер, чиято дефиниция е следната:

Дефиниция 3.1.28 Нека r и m са цели числа и $0 \leq r \leq m$. Двоичен код на Рид-Малер от ред r с дължина $n = 2^m$, е множеството от всички двоични вектори $\underline{\mathbf{f}}$ с дължина n , които са таблици на истинност на булевите функции $f(\mathbf{x})$, $\mathbf{x} = (x_1, \dots, x_m)$, с алгебрични нормални форми от степен най-много r . За този код ще използваме означението $\mathcal{R}(r, m)$.

Оттук нататък двоичен вектор $\underline{\mathbf{f}}$ с дължина 2^m ще бъде идентифициран със съответната Булева функция f на m променливи, т.е. $\underline{\mathbf{f}}$ е таблицата на истинност на f . Означенията f и $\underline{\mathbf{f}}$ ще се използват взаимозаменяемо, в зависимост от контекста. Ще използваме означението $\underline{\mathbf{f}}_n$ вместо $\underline{\mathbf{f}}$, когато искаме да наблегнем на дължината на вектора.

При двоични променливи е изпълнено $x_i^2 = x_i$ и всеки моном от степен $r : 0 \leq r \leq m$ на m променливи е резултат от произведението на r различни променливи, следователно има точно $\binom{m}{r}$ монома от степен r . Под действието на операцията събиране тези мономи генерират група, която означаваме с $\mathcal{H}^{(r)}(m)$ - групата на хомогенните полиноми от степен r на m променливи. Идентитетът на групата е 0 (нулевият полином) и го разглеждаме като полином от произволна степен. Той е единственият елемент, който е общ за $\mathcal{H}^{(i)}(m)$ и $\mathcal{H}^{(j)}(m)$, когато $i \neq j$. Освен това е лесно да се забележи, че $\mathcal{H}^{(r)}(m)$ съдържа $\mathcal{H}^{(r)}(m-1)$ като подгрупа.

Добре известен е следният факт:

Твърдение 3.1.29 За всяко m и всички r , за които $0 \leq r \leq m$, двоичният код на Рид-Малер $\mathcal{R}(r, m)$ е линеен $[n, k, d]$ код с:

- дължина $n = 2^m$
- размерност $k = \sum_{i=0}^r \binom{m}{i}$

- *минимално тегло* $d = 2^{m-r}$;

Дуалният код на $\mathcal{R}(r, m)$ е $\mathcal{R}(m - r - 1, m)$. В частност за всяко $s \geq 1$ кодът $\mathcal{R}(s, 2s + 1)$ е самодуален код.

Кодове на Рид-Малер с дължина 2^{m+2} могат да бъдат получени лесно от такива с дължина 2^{m+1} , като използваме $(\underline{u}, \underline{u} + \underline{v})$ конструкцията.

Теорема 3.1.30 ([36, Ch.13.3])

$$\mathcal{R}(r + 2, m + 2) = \{(\underline{u}, \underline{u} + \underline{v}) : \underline{u} \in \mathcal{R}(r + 2, m + 1), \underline{v} \in \mathcal{R}(r + 1, m + 1)\}.$$

Съществува еквивалентно твърдение в терминологията на пораждащите матрици. Нека $G(r, m)$ е пораждаща матрица на $\mathcal{R}(r, m)$. Тогава Теорема 3.1.30 може да бъде записана във вида:

$$G(r + 2, m + 2) = \begin{pmatrix} G(r + 2, m + 1) & G(r + 2, m + 1) \\ 0 & G(r + 1, m + 1) \end{pmatrix}.$$

Действително кодова дума, генерирана от тази матрица, има формата $(\underline{u}, \underline{u} + \underline{v})$, където $\underline{u} \in \mathcal{R}(r + 2, m + 1)$, $\underline{v} \in \mathcal{R}(r + 1, m + 1)$.

Ще използваме и два факта, публикувани за първи път в [46] през 1973г., които са формулирани в следващите две теореми. (За пълнота на изложението и удобство на читателите излагаме техните доказателства.)

Теорема 3.1.31 ([46, 5.12]) *За $0 \leq r \leq m$, е в сила следното:*

$$\mathcal{W}[z; \mathcal{R}(r + 2, m + 2)] = \sum_{\underline{p} \in \mathcal{H}^{(r+2)}(m+1)} \mathcal{W}^2[z; \underline{p} + \mathcal{R}(r + 1, m + 1)].$$

Доказателство: Нека \underline{p} представлява вектор, съответстващ на полином в $\mathcal{H}^{(r+2)}(m + 1)$, а \underline{q} и \underline{q}' са вектори в $\mathcal{R}(r + 1, m + 1)$. Можем да запишем всеки вектор в $\mathcal{R}(r + 2, m + 1)$ като $\underline{p} + \underline{q}$ и използвайки Теорема 3.1.30 да изразим $\mathcal{R}(r + 2, m + 2)$ чрез $\mathcal{R}(r + 2, m + 1)$ и $\mathcal{R}(r + 1, m + 1)$:

$$\mathcal{R}(r + 2, m + 2) = \{(\underline{p} + \underline{q}, \underline{p} + \underline{q} + \underline{q}'); \forall \underline{p}, \forall \underline{q}, \forall \underline{q}'\}.$$

След това дефинираме множество $S_{\underline{p}, \underline{q}}$ като множеството от вектори $(\underline{p} + \underline{q}, \underline{p} + \underline{q} + \underline{q}')$, където \underline{p} и \underline{q} са фиксирани, а \underline{q}' приема всяка една от допустимите стойности.

Тогава е лесно да се види, че

$$\mathcal{R}(r+2, m+2) = \bigcup_{\forall \underline{p}} \bigcup_{\forall \underline{q}} S_{\underline{p}, \underline{q}}.$$

Номераторът на теглата на $S_{\underline{p}, \underline{q}}$ може да бъде намерен по следния начин. Тъй като

$$S_{\underline{p}, \underline{q}} = \{(\underline{p} + \underline{q}, \underline{p} + \underline{q} + \underline{q}'), \forall \underline{q}'\}.$$

то лявата половина на векторите от $S_{\underline{p}, \underline{q}}$ е константа и има едно и също тегло, а именно $wt(\underline{p} + \underline{q})$. Освен това при фиксирано \underline{q} векторите $\underline{q} + \underline{q}'$, $\forall \underline{q}'$ образуват цялото $\mathcal{R}(r+1, m+1)$, когато \underline{q}' пробягва $\mathcal{R}(r+1, m+1)$. Следователно множеството от вектори $\underline{p} + \underline{q} + \underline{q}'$ образува множеството $\underline{p} + \mathcal{R}(r+1, m+1)$, т.е. съответния съседен клас в $\mathcal{R}(r+2, m+1)$. Следователно, имаме:

$$\mathcal{W}[z; S_{\underline{p}, \underline{q}}] = z^{wt(\underline{p} + \underline{q})} \cdot \mathcal{W}[z; \underline{p} + \mathcal{R}(r+1, m+1)].$$

Сумирайки по всички \underline{q} , получаваме

$$\begin{aligned} \mathcal{W}[z; \bigcup_{\forall \underline{q}} S_{\underline{p}, \underline{q}}] &= \mathcal{W}[z; \underline{p} + \mathcal{R}(r+1, m+1)] \cdot \sum_{\forall \underline{q}} z^{wt(\underline{p} + \underline{q})} \\ &= \mathcal{W}[z; \underline{p} + \mathcal{R}(r+1, m+1)] \cdot \mathcal{W}[z; \underline{p} + \mathcal{R}(r+1, m+1)] \\ &= \mathcal{W}^2[z; \underline{p} + \mathcal{R}(r+1, m+1)]. \end{aligned}$$

Сумирайки последното по всички \underline{p} (да припомним, че \underline{p} е вектор съответстващ на полином в $\mathcal{H}^{(r+2)}(m+1)$), получаваме

$$\mathcal{W}[z; \mathcal{R}(r+2, m+2)] = \sum_{\forall \underline{p} \in \mathcal{H}^{(r+2)}(m+1)} \mathcal{W}^2[z; \underline{p} + \mathcal{R}(r+1, m+1)],$$

с което доказателството е завършено. \square

Теорема 3.1.32 ([46, 5.13]) Нека $p = e + f \cdot x_{m+1} \in \mathcal{H}^{(r+2)}(m+1)$ за дадени $e \in \mathcal{H}^{(r+2)}(m)$ и $f \in \mathcal{H}^{(r+1)}(m)$. Тогава номераторът на теглата в съседния клас $\mathcal{C}(p) = \underline{p} + \mathcal{R}(r+1, m+1)$ е равен на:

$$(*) \quad \mathcal{W}[z; \underline{p} + \mathcal{R}(r+1, m+1)] = \sum_{g \in \mathcal{H}^{(r+1)}(m)} \mathcal{W}[z; \underline{e} + \underline{g} + \mathcal{R}(r, m)] \cdot \mathcal{W}[z; \underline{e} + \underline{f} + \underline{g} + \mathcal{R}(r, m)].$$

Доказателство: Нека \underline{e} е векторът, съответстващ на полинома $e \in \mathcal{H}^{(r+2)}(m)$ и \underline{f} и \underline{g} съответстват на полиноми в $f, g \in \mathcal{H}^{(r+1)}(m)$, а \underline{g}' и \underline{h} са вектори от $\mathcal{R}(r, m)$. Можем да запишем всеки вектор в $\mathcal{R}(r+1, m)$ като $\underline{g} + \underline{g}'$, и използвайки Теорема 3.1.30 да изразим $\mathcal{R}(r+1, m+1)$ чрез $\mathcal{R}(r+1, m)$ и $\mathcal{R}(r, m)$. Така получаваме:

$$(\underline{e}, \underline{e} + \underline{f}) + \mathcal{R}(r+1, m+1) = \{(\underline{e} + \underline{g} + \underline{g}', \underline{e} + \underline{f} + \underline{g} + \underline{g}' + \underline{h}); \forall \underline{e}, \forall \underline{f}, \forall \underline{g}, \forall \underline{g}', \forall \underline{h}\}.$$

По-нататък, да дефинираме множеството $S_{\underline{e}, \underline{f}, \underline{g}, \underline{g}'}$ като множеството от вектори $(\underline{e} + \underline{g} + \underline{g}', \underline{e} + \underline{f} + \underline{g} + \underline{g}' + \underline{h})$ където \underline{e} , \underline{f} , \underline{g} и \underline{g}' са фиксирани, а \underline{h} приема всяка една от допустимите стойности. Тогава е лесно да се види, че

$$\underline{p} + \mathcal{R}(r+1, m+1) = \bigcup_{\forall \underline{g}} \bigcup_{\forall \underline{g}'} S_{\underline{e}, \underline{f}, \underline{g}, \underline{g}'}$$

Номераторът на теглата на $S_{\underline{e}, \underline{f}, \underline{g}, \underline{g}'}$ може да бъде намерен по следния начин. В множеството от вектори

$$S_{\underline{e}, \underline{f}, \underline{g}, \underline{g}'} = \{(\underline{e} + \underline{g} + \underline{g}', \underline{e} + \underline{f} + \underline{g} + \underline{g}' + \underline{h}); \forall \underline{h}\}$$

забелязваме, че лявата половина има едно и също тегло, а именно $wt(\underline{e} + \underline{g} + \underline{g}')$. Тъй като \underline{g}' и \underline{h} принадлежат на $\mathcal{R}(r, m)$, при фиксирано \underline{g}' , векторите $\underline{g}' + \underline{h}$, $\forall \underline{h}$ образуват самото множество $\mathcal{R}(r, m)$. Следователно имаме

$$\mathcal{W}[z; S_{\underline{e}, \underline{f}, \underline{g}, \underline{g}'}] = z^{wt(\underline{e} + \underline{g} + \underline{g}')} \cdot \mathcal{W}[z; \underline{e} + \underline{f} + \underline{g} + \mathcal{R}(r, m)].$$

Сумирайки по всички \underline{g}' , получаваме

$$\begin{aligned} \mathcal{W}[z; \bigcup_{\forall \underline{g}'} S_{\underline{e}, \underline{f}, \underline{g}, \underline{g}'}] &= \left(\sum_{\forall \underline{g}'} z^{wt(\underline{e} + \underline{g} + \underline{g}')} \right) \cdot \mathcal{W}[z; \underline{e} + \underline{f} + \underline{g} + \mathcal{R}(r, m)] \\ &= \mathcal{W}[z; \underline{e} + \underline{g} + \mathcal{R}(r, m)] \cdot \mathcal{W}[z; \underline{e} + \underline{f} + \underline{g} + \mathcal{R}(r, m)]. \end{aligned}$$

Накрая, сумирайки по всички \underline{g} , получаваме желанния резултат

$$\mathcal{W}[z; \underline{p} + \mathcal{R}(r+1, m+1)] = \sum_{\underline{g} \in \mathcal{H}^{(r+1)}(m)} \mathcal{W}[z; \underline{e} + \underline{g} + \mathcal{R}(r, m)] \cdot \mathcal{W}[z; \underline{e} + \underline{f} + \underline{g} + \mathcal{R}(r, m)],$$

с което доказателството е завършено. \square

Следват дефинициите на *Общата афинна група* $GA(m)$ и нейната подгрупа - *Общата линейна група* $GL(m, 2)$ [36, Ch.13.9].

Нека $A = (a_{ij})$ е обратима $m \times m$ двоична матрица, а \underline{b} е двоичен вектор с дължина m . Трансформацията

$$T : \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix} \rightarrow A \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix} + \underline{b} \quad (3.2)$$

е пермутация на множеството от 2^m вектора с дължина m , която изпраща $\underline{0}$ във вектора \underline{b} .

Можем също така да разглеждаме трансформацията T като пермутираща булеви функции:

$$T : f(v_1, \dots, v_m) \rightarrow f(\sum a_{1j}v_j + b_1, \dots, \sum a_{mj}v_j + b_m). \quad (3.3)$$

Множеството от всички такива трансформации T образува група, като операцията на групата е композиция. Редът на тази група се намира по следния начин. Първата колона на A може да бъде избрана по $2^m - 1$ начина, втората по $2^m - 2$, третата по $2^m - 4$, и така нататък. Освен това има 2^m възможности за избор на \underline{b} . Тази група, наречена *Обща афинна група* и означавана с $GA(m)$, има ред

$$|GA(m)| = 2^m(2^m - 1)(2^m - 2)(2^m - 2^2) \dots (2^m - 2^{m-1}). \quad (3.4)$$

Подгрупата на $GA(m)$, състояща се от всички трансформации

$$T : \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix} \rightarrow A \begin{pmatrix} v_1 \\ \vdots \\ v_m \end{pmatrix}, \quad (3.5)$$

(т.е. за които $\underline{b} = \underline{0}$) се нарича *Обща линейна група* и се означава с $GL(m, 2)$. Нейният ред е

$$|GL(m, 2)| = (2^m - 1)(2^m - 2)(2^m - 2^2) \dots (2^m - 2^{m-1}). \quad (3.6)$$

Действието на $A \in GA(m)$ върху булева функция $f(\mathbf{x})$ означаваме с $f \circ A$, т.е., $f \circ A = f(A(\mathbf{x}))$. Друга необходима дефиниция е тази за афинна еквивалентност на два съседни класа на код на Рид-Малер:

Дефиниция 3.1.33 *Съседните класове C_1 и C_2 на $\mathcal{R}(r, m)$, с представители $f_1 \in C_1$ и $f_2 \in C_2$ съответно, се наричат афинно еквивалентни, ако съществува трансформация $A \in GA(m)$, такава че $f_1 \circ A = f_2$.*

Ако съседните класове C_1 и C_2 са афинно еквивалентни, т.е. съществува въпросната афинна трансформация A , то под нейното действие съседният клас C_1 се изобразява в съседния клас C_2 , тъй като A изобразява биективно $\mathcal{R}(r, m)$ в себе си.

По-нататък ще използваме широко следното добре известно свойство (вижте например [24]):

Свойство \mathcal{P} . *Номераторите на теглата на два афинно еквивалентни съседни класа на един код на Рид-Малер са идентични.*

Класификацията по афинна еквивалентност на съседните класове на кодовете на Рид-Малер е полезна при изучаването на важни кодово-теоретични и криптографски свойства на Булевите функции, които съставят тези класове. В статията [34] е представена стратегия за пресмятане на пълната класификация на Булевите форми от четвърта степен на осем променливи, т.е. класификацията на фактор-пространството $\mathcal{R}(4, 8)/\mathcal{R}(3, 8)$ под действието на $GL(8, 2)$. На това място само като екстракт от този резултат, отбелязваме, че Булевите форми от четвърта степен на осем променливи могат да бъдат класифицирани в 999 класа на линейна еквивалентност, изброени в [33]. От информацията, представена в [33], ще използваме само представителите и размера на орбитата на всеки клас. Наскоро интересът към тази тема бе подновен от [19], където (освен всичко друго) е предоставена класификация по афинна еквивалентност на фактор-пространството $\mathcal{R}(4, 7)/\mathcal{R}(2, 7)$. Авторите на [19] и [34] скицират някои приложения на своите резултати, свързани с радиусите на покритие на някои РМ кодове, както и с Булевите бент функции. В Раздел 3.3 ще опишем още едно приложение на горепосочените резултати, а именно пресмятането на тегловото разпределение на $\mathcal{R}(4, 9)$.

3.2 Опит за намиране на тегловото разпределение чрез теоремата на Gleason

От теоремата на McEliece за делимостта на теглата на кодовите думи в $\mathcal{R}(r, m)$ (вж., напр., [36, Гл. 15, Теорема 12]) следва:

Следствие 3.2.1 *Всички тегла в $\mathcal{R}(r, m)$ са кратни на*

$$2^{\lfloor (m-1)/r \rfloor} = 2^{\lceil m/r \rceil - 1}.$$

В случая на $\mathcal{R}(4, 9)$ всички тегла са кратни на 4, т.е. $\mathcal{R}(4, 9)$ е двойночетен код.

Нека с W_w означим броя на кодовите думи с тегло w в $\mathcal{R}(r, m)$. Добре известно е (вж., напр., [36, Гл. 13, Теорема 9(b)]), че е в сила:

Твърдение 3.2.2 *Броят на кодовите думи с минимално тегло в $\mathcal{R}(r, m)$ е*

$$W_{2^{m-r}} = 2^r \prod_{i=0}^{m-r-1} \frac{2^{m-i} - 1}{2^{m-r-i} - 1}.$$

От [27, Теорема 2] за тегловото разпределение на $\mathcal{R}(r, m)$, в интервала $d = 2^{m-r} < w < 2d$, е в сила следното:

Твърдение 3.2.3 *За $r \geq 2$ и $2^{m-r} < w < 2^{m-r-1}$, нека $\alpha = \min(m-r, r)$ и $\beta = \frac{1}{2}(m-r+2)$. Тогава*

(i) $W_w = 0$, освен ако $w(\mu) = 2^{m-r+1} - 2^{m-r+1-\mu}$ за $\mu : 1 \leq \mu \leq \max(\alpha, \beta)$.

(ii) Ако $\mu = 2$ или $\max(\alpha, 2) < \mu \leq \beta$, то

$$W_{w(\mu)} = 2^{r+\mu^2+\mu-2} \frac{\prod_{i=0}^{r+2\mu-3} (2^{m-i} - 1)}{\prod_{i=0}^{r-3} (2^{r-2-i} - 1) \prod_{i=0}^{\mu-1} (4^{i+1} - 1)}. \quad (3.7)$$

(iii) Ако $\max(\beta, 2) < \mu \leq \alpha$, то

$$W_{w(\mu)} = 2^{r+\mu^2+\mu-1} \frac{\prod_{i=0}^{r+\mu-1} (2^{m-i} - 1)}{\prod_{i=0}^{r-\mu-1} (2^{r-\mu-i} - 1) \left[\prod_{i=0}^{\mu-1} (2^{\mu-i} - 1) \right]^2}. \quad (3.8)$$

(iv) Ако $3 \leq \mu \leq \min(\alpha, \beta)$, то $W_{w(\mu)}$ е равно на сумата от (3.7) и (3.8).

Ще използваме Твърдение 3.2.3, за да намерим всички тегла (включително нулевите) на $\mathcal{R}(4, 9)$ в интервала $[36, 60]$.

Твърдение 3.2.3 е разширено за тегла $w : 2d \leq w < 2.5d$ в [28]. Ще използваме това разширение, за да намерим теглата на $\mathcal{R}(4, 9)$ в интервала $[64, 76]$.

$\mathcal{R}(4, 9)$ е двойночетен двоичен самодуален код и общата форма на номераторите на теглата на такива кодове е известна от теоремата на А. М. Gleason [20, 2]. Ще представим тази теорема във форма, която касае само разглеждания код.

Теорема 3.2.4 *Нека \mathbf{C} е двоичен самодуален линеен код, на който всяко тегло е кратно на 4 (двойночетен код). Тогава номераторът на теглата на кода \mathbf{C} е сума от произведенията на следните два полинома*

$$f_1(x, y) = x^8 + 14x^4y^4 + y^8$$

и

$$f_2(x, y) = x^4y^4(x^4 - y^4)^4,$$

т.е.

$$\mathcal{W}(x, y; \mathbf{C}) = \sum_{i,j} K_{ij} f_1^i(x, y) f_2^j(x, y),$$

където i, j са неотрицателни цели числа, K_{ij} са някакви константи, и $n = 8i + 24j$ е дължината на кода.

Според Следствие 3.2.1 всички ненулеви тегла на Рид-Малер кода $\mathcal{R}(4, 9)$ се делят на 4. В опит да намерим тегловото му разпределение, последователно извършваме следните действия:

- (1) Пресмятаме броя на кодовите думи с минимално тегло, използвайки Твърдение 3.2.2

$$W_{32} = 52955952$$

- (2) С помощта на Твърдение 3.2.3 от [27], намираме следващата част от тегловото разпределение:

$$W_{36} = W_{40} = W_{44} = W_{52} = 0$$

$$\begin{aligned}
W_{48} &= 919315326720, \\
W_{56} &= 271767121346560, \\
W_{60} &= 860689275027456.
\end{aligned}$$

- (3) Използвайки разширената версия на Твърдение 3.2.3 от [28], намираме следващата част от тегловото разпределение:

$$\begin{aligned}
W_{64} &= 89163020044002040, \\
W_{68} &= 1777323352931696640, \\
W_{72} &= 64959328938397057024, \\
W_{76} &= 2094952122987829002240.
\end{aligned}$$

- (4) Използвайки теоремата на Gleason (Теорема 3.2.4) и вече намерените 20 тегла, се опитваме да намерим тегловото разпределение на $\mathcal{R}(4, 9)$.

- (i) Трябва да решим системата

$$\sum_{i=0,4|i}^n W_i x^i y^{n-i} = \sum_{i,j} K_{ij} f_1^i(x, y) f_2^j(x, y),$$

където $n = 512 = 8i + 24j$ и за (i, j) има 22 възможни комбинации.

- (ii) Тъй като имаме 22 уравнения, а знаем стойностите само на 20 от теглата, избираме за параметри теглата W_{80} и W_{84} . Означаваме коефициентите K_{ij} като b_0, b_1, \dots, b_{21} .

- (iii) Решаваме системата

$$\sum_{i=0,4|i}^{512} W_i x^i y^{n-i} = \sum_{i=0}^{21} b_i f_1^{64-3i}(x, y) f_2^i(x, y)$$

и за търсените коефициенти b_0, b_1, \dots, b_{21} получаваме

$$\begin{aligned}
b_0 &= 1, \\
b_1 &= -896, \\
b_2 &= 366400, \\
b_3 &= -90591488, \\
b_4 &= 15125312416, \\
b_5 &= -1804090983680, \\
b_6 &= 158633285104000, \\
b_7 &= -10462864561216000, \\
b_8 &= 521844917396785920, \\
b_9 &= -19703740374727094272, \\
b_{10} &= 560488125281758654464, \\
b_{11} &= -11885355149550800420864, \\
b_{12} &= 184790978198497210204160, \\
b_{13} &= -2057511108351137030602752, \\
b_{14} &= 15886111690015316194099200, \\
b_{15} &= -81428598375642719259197440, \\
b_{16} &= 261053953183414744116101120, \\
b_{17} &= -481398485705822986116792320, \\
b_{18} &= 451352262788439851856297984, \\
b_{19} &= -176727086611150405833326592, \\
b_{20} &= W_{80} + 20053805254648300758129920, \\
b_{21} &= W_{84} + 24W_{80} - 6064979573531823991900160.
\end{aligned}$$

От получения резултат става ясно, че за да намерим тегловото разпределение на $\mathcal{R}(4, 9)$, използвайки теоремата на Gleason, се нуждаем от повече съществена информация от тази, представена в [27] и [28] (вижте [11, Ch. 11] за подробности).

3.3 Описание на метода на Sarwate и прецизирането му

3.3.1 Теоретично представяне

Ще опишем стратегия, която прави възможно пресмятането на $\mathcal{W}[z; \mathcal{R}(4, 9)]$. Нека $n(r, m)$ е броят на орбитите на които фактор-пространството

$$\mathcal{R}^*(r, m) = \mathcal{R}(r, m) / \mathcal{R}(r - 1, m),$$

се разбива под действието на $GL(m, 2)$, когато $1 \leq r \leq m$. Освен това приемаме, че е фиксирано произволно номериране на тези орбити (класове на линейна еквивалентност).

Следното следствие от Теорема 3.1.31 ни дава възможност да я използваме ефективно.

Следствие 3.3.1 *Нека $p_i \in \mathcal{H}^{(r+2)}(m+1)$ е представител на i -тия клас на линейна еквивалентност в $\mathcal{R}^*(r+2, m+1)$ с размер L_i . Тогава е вярно следното:*

$$\mathcal{W}[z; \mathcal{R}(r+2, m+2)] = \sum_{i=1}^{n(r+2, m+1)} L_i \mathcal{W}^2[z; \underline{p}_i + \mathcal{R}(r+1, m+1)]. \quad (3.9)$$

Доказателство: Твърдението е непосредствен резултат от Теорема 3.1.31 и Свойство \mathcal{P} . \square

Това следствие намалява броя на необходимите пресмятания на номератори на теглата до $n(r+2, m+1)$, което е значително по-малко от директния израз

$$|\mathcal{H}^{(r+2)}(m+1)| = 2^{\binom{m+1}{r+2}}$$

в Теорема 3.1.31. Например, както вече беше споменато, $n(4, 8) = 999 \approx 2^{10}$, което е 2^{60} пъти по-малко в сравнение с $|\mathcal{H}^{(4)}(8)| = 2^{70}$.

Забележка 3.3.2 *Следствие 3.3.1 се използва неявно в [46] за намиране номераторите на теглата на по-късите кодове на Рид-Малер.*

Ще формулираме още едно твърдение, което позволява допълнително намаляване на пресмятанията.

Следствие 3.3.3 За дадено $e \in \mathcal{H}^{(r+2)}(m)$, нека $\mathcal{H}^{(r+1)}(m)$ бъде разбито на блокове (подмножества) $G_i, 1 \leq i \leq s$, такива че, винаги когато $g \in G_i$, номераторът на теглата $\mathcal{W}[z; \underline{e} + \underline{g} + \mathcal{R}(r, m)]$ е (различен) константен полином $w_i(z)$. Тогава е вярно следното:

- (a) номераторът на теглата на съседния клас $\mathcal{C}(p) = \underline{p} + \mathcal{R}(r+1, m+1)$, $p = e + f \cdot x_{m+1}$ за фиксирано $f \in \mathcal{H}^{(r+1)}(m)$, се изразява чрез

$$\sum_{i=1}^s w_i(z) \left(\sum_{g \in G_i} \mathcal{W}[z; \underline{e} + \underline{g} + \underline{f} + \mathcal{R}(r, m)] \right).$$

- (b) броят на умноженията на полиноми за пресмятане на посочения номератор на теглата е равен на s , т.е. на броят на различните номератори на тегла $\mathcal{W}[z; \underline{e} + \underline{g} + \mathcal{R}(r, m)]$, $g \in \mathcal{H}^{(r+1)}(m)$, докато този на събиранията на полиноми е $2^{\binom{m}{r+1}} - s$.

Доказателство: Пренареждането на събираемите в (*) от Теорема 3.1.32 и изваждането извън скобите на общите множители $w_i(z)$ доказва (a). Твърдението (b) е пряко следствие от (a). \square

Класификацията по афинна еквивалентност на $\mathcal{R}(r+2, m)/\mathcal{R}(r, m)$ позволява да се реализира използването на Следствие 3.3.3. За да се види това, ще припомним следната дефиниция:

Дефиниция 3.3.4 Подгрупата $St(e)$ на $GA(m)$, която фиксира $e \in \mathcal{H}^{(r+2)}(m)$, се нарича стабилизатор на e в $GA(m)$, т.е., за всяко $A \in St(e)$ е в сила: $e \circ A \in \underline{e} + \mathcal{R}(r+1, m)$.

За дадено $e \in \mathcal{H}^{(r+2)}(m)$ стабилизаторът $St(e)$ разбива съседните класове от вида $\underline{e} + \underline{g} + \mathcal{R}(r, m)$, където $g \in \mathcal{H}^{(r+1)}(m)$, в непресичащи се орбити. Да означим това разбиване с $\Delta(e)$. Освен това Свойство \mathcal{P} влече, че когато g се движи по орбита от $\Delta(e)$, номераторът на теглата $\mathcal{W}[z; \underline{e} + \underline{g} + \mathcal{R}(r, m)]$ се запазва. Последното позволява да се състави ефективно разбиването $\Delta'(e) = \{G_i, 1 \leq i \leq s\}$ на $\mathcal{H}^{(r+1)}(m)$ (вижте Следствие 3.3.3) чрез обединяване на тези орбити, които имат идентични номератори на теглата (като последните се пресмятат предварително върху избрани представители на орбитите).

3.3.2 Пресмятане на $\mathcal{W}[z; \mathcal{R}(4, 9)]$

Изчислителната работа се разделя на две основни фази: предварително изчисление и същинско изчисление.

Целта на предварителното изчисление е да предостави инструменти за ефективно пресмятане на израза $(*)$ в Теорема 3.1.32 за конкретни e и f и се извършва, следвайки Следствие 3.3.3 и последващите съображения от предишния подраздел.

Нека $\mathcal{E}(4, 7)$ бъде множеството от представителите на дванадесетте класа на линейна еквивалентност на $\mathcal{R}^*(4, 7)$, дадени в [32]. За фиксирано $e \in \mathcal{E}(4, 7)$, предварителното изчисление включва следните три задачи:

- $\mathcal{T}1$: Съставяне и съхранение на орбитите на разбиването $\Delta(e)$;
- $\mathcal{T}2$: Пресмятане на номераторите на теглата на съседните класове $\underline{e} + \underline{g} + \mathcal{R}(2, 7)$, когато g варира върху множеството от представители на орбитите на $\Delta(e)$;
- $\mathcal{T}3$: Сливане на орбитите с идентични номератори на теглата, за да се получи „по-грубото“ разбиване $\Delta'(e)$.

(От самото начало на предварителното изчисление подреждането на данните е такова, че позволява за дадено $f \in \mathcal{H}^{(3)}(7)$ да се намира индексът на блока в $\Delta(e)$, съдържащ f , с цел директен достъп до общия номератор на теглата.)

За всички $e \in \mathcal{E}(4, 7)$, в Таблица А.1 на Приложение А, са представени размерите на разбиванията $\Delta(e)$ и $\Delta'(e)$, съответно.

Забележка 3.3.5 *Заслужава да се отбележи, че:*

- задача $\mathcal{T}1$ се извършва ефективно на базата на така наречения „orbit algorithm“ [25], използвайки множеството от генератори на стабилизатора $\mathcal{St}(e)$, предоставени от [32];
- задача $\mathcal{T}2$ може да бъде извършена едновременно за всички представители на орбитите, чрез генериране на всички кодовите думи на $\mathcal{R}(2, 7)$ с помощта на код на Грей.

Следвайки стратегията, описана в Раздел 3.3.1, представяме алгоритъм за пресмятане на номератора на теглата $\mathcal{W}[z; C(p)]$ на съседния клас $C(p) = \underline{p} + \mathcal{R}(3, 8)$, където $p = e + f \cdot x_8$ за фиксирано $e \in \mathcal{E}(4, 7)$ и дадено $f \in \mathcal{H}^{(3)}(7)$. Да обърнем

внимание, че алгоритъмът може да бъде имплементиран като подпрограма. Да отбележим също така, че общият номератор на теглата $w_i(z)$, съответстващ на блока G_i в $\Delta'(e)$, вече е пресметнат в задача $\mathcal{T}2$ от предварителните изчисления, където $1 \leq i \leq |\Delta'(e)| = s(e)$ и с $s(e)$ е означен размерът на $\Delta'(e)$, т.е. броят на блоковете, които се съдържат в $\Delta'(e)$.

Algorithm 3: Връща номератора на теглата $\mathcal{W}[z; C(p)]$, където $p = e + f \cdot x_8$ при фиксирано e и дадено $f \in \mathcal{H}^{(3)}(7)$

```

1 U[z] := 0;
2 for i in [1, s(e)] do
3   UU(z) := 0;
4   for g in G[i] do
5     j := FindBlock(g+f);
6     UU(z) := UU(z) + w[j](z);
7   U(z) := U(z) + w[i](z) * UU(z);
8 W[z; C(p)] := U(z);
```

При същинското изчисление прилагаме уравнение (3.9), предполагайки, че е наличен списък \mathcal{S} от двойки: (представител p_i , размер на орбита L_i) за i -тия клас O_i , $1 \leq i \leq 999$, от класификацията на $\mathcal{R}^*(4, 8)$. Без ограничение на общността, можем да предположим, че всеки p_i е от вида $e + f_i x_8$ за някое $e \in \mathcal{E}(4, 7)$ и $f_i \in \mathcal{H}^{(3)}(7)$, така че множеството от класове е естествено разбито на подмножества $\mathcal{O}(e)$ с мощност $\mu(e)$, $e \in \mathcal{E}(4, 7)$. (Стойностите $\mu(e)$ са дадени в първата колона на Таблица А.2 от Приложение А.) Очевидно, имаме $\sum_{e \in \mathcal{E}(4, 7)} \mu(e) = 999$. По-долу е представен алгоритъм за пресмятане на сумата в уравнение (3.9) и съответно номератора на теглата $\mathcal{W}[z; \mathcal{R}(4, 9)]$. (Обръщаме внимание, че се извиква подпрограмата $\mathcal{W}[z; C(p)]$.)

Algorithm 4: Пресмятане на $\mathcal{W}[z; \mathcal{R}(4, 9)]$

```

1 V(z) := 0;
2 for e in  $\mathcal{E}(4, 7)$  do
3   for j in [1, n(e)] do
4     p := Representative( $\mathcal{O}(e)[j]$ );
5     L := Size( $\mathcal{O}(e)[j]$ );
6     V(z) := V(z) + L *  $\mathcal{W}^2[z; C(p)]$ ;
7  $\mathcal{W}[z; \mathcal{R}(4, 9)] = V(z)$ ;
```

Данните, представени в [33], съдържат необходимата информация за формирането на списък \mathcal{S}' от вид, подобен на \mathcal{S} . Въпреки подобие то представителите p'_i в \mathcal{S}' в него са от вида $e' + f'_i \cdot x_8$, и новите e' образуват различно множество от представители на дванадесетте класа на $\mathcal{R}^*(4, 7)$, да речем $\mathcal{E}'(4, 7)$. Линеината еквивалентност на някои от елементите на $\mathcal{E}(4, 7)$ и $\mathcal{E}'(4, 7)$ е очевидна при визуална проверка. За останалите определяме двойките линеинно еквивалентни, като пресмятаме инвариантите на техните дуални (за подробности, препращаме читателите към [24, pp. 115-117]). Намереното съответствие е представено в редовете на **Таблица А.2**, където $\bar{\mathcal{E}}(4, 7)$ и $\bar{\mathcal{E}}'(4, 7)$ са множествата, състоящи се от дуалните форми на тези в $\mathcal{E}(4, 7)$ и $\mathcal{E}'(4, 7)$, съответно. По-нататък, за да намерим за всяка една от определените двойки (e', e) неособена (7×7) матрица \mathbf{A} (т.е. линеината трансформация на преход) със свойството $e' \circ \mathbf{A} \in \underline{e} + \mathcal{R}(3, 7)$, написахме проста програма на C, която генерира на „случаен“ принцип такава неособена квадратна матрица и след това проверява дали е удовлетворено наложеното условие. Тази техника е достатъчно ефективна поради относително големите размери на стабилизаторите (вижте [34, **Таблица А.2**]) и програмата свърши успешно своята работата за разумно време. (За подобна техника за изследване на афинната еквивалентност на Булеви функции вижте [40].) Получените резултати са представени в последната колона на **Таблица А.2** от **Приложение А**. Накрая, прилагайки линеините трансформации на преход върху съответните f'_i , $1 \leq i \leq 999$, (разбира се, като се игнорират членовете от степен по-малка от 3), получаваме списък от представители, който е необходим за прилагане на **Алгоритъм 4**. Тегловото разпределение, което се получава, е представено в **Таблица А.3** на **Приложение А**.

Забележка 3.3.6 *Функциите $FindBlock(\dots)$, $Representative(\dots)$ и $Size(\dots)$ имат имена, които сами обясняват тяхното предназначение.*

3.3.3 Оценяване на изчислителните разходи

Следвайки [25] и [32], оценяваме, че изчислителната цена на задача $\mathcal{T}1$ е

$$|\mathcal{H}^{(3)}(7)| \times \sum_{e \in \mathcal{E}(4, 7)} |Sg(e)| = 2^{35} \times 26 \approx 2^{39.7}$$

афинни трансформации, където с $Sg(e)$ е означено множеството от генераторите на стабилизатора $St(e)$. Изчислителната сложност на задача $\mathcal{T}2$ е пропорционална на произведението $68443 \times 2^{29} \approx 2^{45.06}$, като първият множител тук е броят на класовете на $\mathcal{R}(4, 7)/\mathcal{R}(2, 7)$, а вторият е обемът на $\mathcal{R}(2, 7)$. Задача $\mathcal{T}3$ може да бъде изпълнена чрез прилагане на техника за сортиране. Резюмирайки, може да кажем, че предварителните изчисления в случая $r = 2$ и $m = 7$ се извършват ефективно. Освен това компресираното съхранение на орбитите и подредба на данните в RAM изисква най-много 124 GB памет. Също така, както вече беше споменато в предишния подраздел, намирането на линейните трансформации на преходите, изброени в **Таблица А.2**, не изисква много изчислителни усилия.

Множеството от класове на линейни еквивалентност на $\mathcal{R}^*(4, 8)$ е естествено разбито на подмножества с мощности $\mu(e)$ за фиксирано $e \in \mathcal{E}(4, 7)$ и различни $f \in \mathcal{H}^{(3)}(7)$ (вижте [33]):

$$\bar{\mu} = (3, 2, 21, 15, 89, 56, 10, 7, 502, 1, 1, 292)$$

При действителното изчисление, за всяко $e \in \mathcal{E}(4, 7)$, **Алгоритъм 3** изисква $|\Delta'(e)|$ умножения и около 2^{35} събираня на полиноми от степен 128 с неотрицателни цели коефициенти, докато **Алгоритъм 4** изисква

$$\sum_{e \in \mathcal{E}(4, 7)} (\mu(e) \times |\Delta'(e)|) = 1\,827\,252 \approx 2^{20.8}$$

умножения и около

$$n(4, 8) \times 2^{\binom{7}{3}} = 999 \times 2^{35} \approx 2^{45}$$

събираня на полиноми от същия вид; също така 999 повдигания на квадрат на полиноми от степен 256, както и някои допълнителни операции с пренебрежимо ниска цена.

Забележка 3.3.7 *Директното прилагане на Теорема 3.1.32 (базирано на оригиналното разбиване $\Delta(e)$) ще изисква около 6 пъти повече умножения на полиноми от 128-ма степен отколкото са изпълнени в действителност.*

Забележка 3.3.8 *Накрая, имаме две забележки относно имплементацията:*

- *За да се справим с ограниченията на RAM паметта (128 GB в нашия случай), използваме подходящите за тази цел Delta компресия и VByte кодиране на данните;*

- *Използваме 256-битовите CPU регистри, което гарантира, че аритметичните операции се извършват ефективно и отстранява необходимостта от допълнително оценяване на броя на процесорните операции.*

Техниките за компресиране, които използваме тук, са съществени за нашето компютърно подпомогнато решение и поради това ще бъдат илюстрирани в **Приложение Б**.

3.3.4 Използвани компютърни ресурси

При пресмятане на тегловото разпределение на двоичния код на Рид-Малер $\mathcal{R}(4, 9)$, са използвани следните ресурси:

- *Avitohol*
 - 150 x HP Cluster Platform SL250S GEN8 servers
 - * 2 x Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz 8 core
Total Cores: 16
RAM: 64 GB
 - * 2 x Intel Xeon Phi 7120P
- *Center of Excellence for Informatics and Information and Communication Technology*
 - 40 x Fujitsu Primergy RX 2540 M4 servers
 - * 2 x Intel(R) Xeon(R) Gold 5118 CPU @ 2.30GHz 24 core
Total Cores: 48
RAM: 128 GB
 - * NVIDIA Tesla V100 32GB

3.4 Заключение

Благодарение на скорошния напредък в класификацията на Булевите функции [19],[34] и използването на съвременни високопроизводителни компютри бе получено решение на задачата за намиране тегловото разпределение на двоичния

код на Рид-Малер $\mathcal{R}(4, 9)$. Въпреки това трябва да се признае, че едва ли е възможно този подход да бъде приложен за по-дълги кодове на Рид-Малер, тъй като изчислителните разходи нарастват неимоверно с увеличаване на кодовата дължина.

Резултатите, получени в тази глава, са докладвани на „*Национален семинар по теория на кодирането "Професор Стефан Додунеков"*, Арбанаси, България, 10-13 Ноември 2022“ [T4], на „*High-Performance Computing for Mathematics and Applications*, Sofia, Bulgaria, 28 June 2023“ [T5], на „*Национален семинар по теория на кодирането "Професор Стефан Додунеков"*, Хисаря, България, 2-5 Ноември 2023“ [T6], както и на „*The Thirteenth International Workshop on Coding and Cryptography (WCC)*, Perugia, Italy, 17-21 June 2024“ [T7].

Библиография

- [1] Aladov, N. S. On the distribution of quadratic residues and nonresidues modulo a prime number p in the sequence $1, 2, \dots, p - 1$. (in Russian). *Matematicheskii Sbornik* **18**, pp. 61–75 (1896).
- [2] Berlekamp, E., MacWilliams, F. & Sloane, N. Gleason’s theorem on self-dual codes. *IEEE Transactions on Information Theory* **18**, pp. 409–414. DOI: 10.1109/TIT.1972.1054817 (1972).
- [3] Borissov, Y. & Markov, M. *An Approach for Computing the Number of Points on Elliptic Curve $y^2 = x^3 + a \pmod{p}$ via Explicit Formula for That Number Modulo p* in: *2019 Ninth International Workshop on Signal Design and its Applications in Communications (IWSDA)* (2019), pp. 1–5. DOI: 10.1109/IWSDA46143.2019.8966127.
- [4] Borissov, Y. & Markov, M. An Efficient Approach to Point-Counting on Elliptic Curves from a Prominent Family over the Prime Field \mathbb{F}_p . *Mathematics* **9**. DOI: 10.3390/math9121431 (2021).
- [5] Bos, J. W. *и др. Elliptic curve cryptography in practice* in: *Financial Cryptography and Data Security: 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers 18* (2014), pp. 157–175.
- [6] Boyvalenkov, P. *Кодиране, комбинаторика и криптография* (ЮЗУ „Неофит Рилски“, 2003).
- [7] Brillhart, J. Note on Representing a Prime as a Sum of Two Squares. *Mathematics of Computation* **26**, pp. 1011–1013 (1972).
- [8] Carlet, C. *Boolean Functions for Cryptography and Coding Theory* DOI: 10.1017/9781108606806 (Nov 2020).
- [9] Carlet, C. & Solé, P. The weight spectrum of two families of Reed-Muller codes. *Discrete Mathematics* **346**, pp. 113568. DOI: <https://doi.org/10.1016/j.disc.2023.113568> (2023).

- [10] Certicom. Sec 2: Recommended Elliptic Curve Domain Parameters. *Certicom Res., Mississauga, ON, Canada, Tech. Rep. SEC2-Ver-2.0* (2010).
- [11] Charpin, P., Pless, V. & Huffman, W. Open problems on cyclic codes. *Handbook of coding theory* **1**, pp. 963–1063 (1998).
- [12] Cohen, H. *A course in computational algebraic number theory* (Springer Science & Business Media, 2013).
- [13] Cormen, T., Leiserson, C., Rivest, R. & Stein, C. *Introduction to Algorithms, fourth edition* (MIT Press, 2022).
- [14] Croucher, J. S. Collecting Coupons-A Mathematical Approach. *Australian Senior Mathematics Journal* **20**, pp. 31–35 (2006).
- [15] Cusick, T. W. & Stănică, P. *Cryptographic Boolean Functions and Applications* (Academic Press/Elsevier, 2009).
- [16] Dickson, L. E. *History of the Theory of Numbers: Quadratic and Higher Forms* (The New Era Printing Company, Lancaster, PA., 1919).
- [17] Dickson, L. E. *History of the Theory of Numbers: Quadratic and Higher Forms* (The Lord Baltimore Press, Baltimore, MD., 1923).
- [18] Galin, B. *Schoof-Elkies-Atkin Algorithm* Department of Mathematics, Stanford University. <https://api.semanticscholar.org/CorpusID:131768553>.
- [19] Gillot, V. & Langevin, P. Classification of some cosets of the Reed-Muller code. *Cryptography and Communications* **15**, pp. 1129–1137. DOI: 10.1007/s12095-023-00652-4 (2023).
- [20] Gleason, A. M. *Weight polynomials of self-dual codes and the MacWilliams identities* in: *Actes Congres International des Mathematiciens* **3** (1970), pp. 211–215.
- [21] Hardy, G. H., Wright, E. M., Heath-Brown, D. R. & Silverman, J. H. *An Introduction to the Theory of Numbers* (Oxford University Press, 2008).
- [22] Harvey, D. & Hoeven, J. v. d. Integer multiplication in time $O(n \log n)$. *Annals of Mathematics* **193**, pp. 563–617. DOI: 10.4007/annals.2021.193.2.4 (2021).
- [23] Hermite, C. (in French). *Note sur un théorème relatif aux nombres entiers* in: *Œuvres de Charles Hermite* pp. 264–264 (Cambridge University Press, 2009).
- [24] Hou, X.-d. $GL(m, 2)$ acting on $R(r, m)/R(r - 1, m)$. *Discrete Mathematics* **149**, pp. 99–122. DOI: [https://doi.org/10.1016/0012-365X\(94\)00342-G](https://doi.org/10.1016/0012-365X(94)00342-G) (1996).

-
- [25] Hulpke, A. *Computing with group orbits* available at <https://www.math.colostate.edu/~hulpke/talks/polyhedralpost.pdf>. 2006.
 - [26] Ireland, K. & Rosen, M. I. *A Classical Introduction to Modern Number Theory* (Springer Science & Business Media, 1990).
 - [27] Kasami, T. & Tokura, N. On the weight structure of Reed-Muller codes. *IEEE Transactions on Information Theory* **16**, pp. 752–759. DOI: 10.1109/TIT.1970.1054545 (1970).
 - [28] Kasami, T., Tokura, N. & Azumi, S. On the weight enumeration of weights less than $2.5d$ of Reed-Muller codes. *Information and Control* **30**, pp. 380–395. DOI: [https://doi.org/10.1016/S0019-9958\(76\)90355-7](https://doi.org/10.1016/S0019-9958(76)90355-7) (1976).
 - [29] Kirlar, B. B. On the elliptic curves $y^2 = x^3 - c$ with embedding degree one. *Journal of Computational and Applied Mathematics* **235**. Congressional Contributions to Computational and Applied Mathematics: ICCAM2009, pp. 4724–4728. DOI: <https://doi.org/10.1016/j.cam.2010.08.020> (2011).
 - [30] Knuth, D. E. *The art of computer programming, volume 2: Seminumerical Algorithms* (Addison-Wesley Longman Publishing Co., Inc., 1997).
 - [31] Koblitz, N. Elliptic curve cryptosystems. *Mathematics of computation* **48**, pp. 203–209 (1987).
 - [32] Langevin, P. *Classification of $RM(4, 7)/RM(2, 7)$* available at <https://langevin.univ-tln.fr/project/rm742/rm742.html>. 2012.
 - [33] Langevin, P. *Classification of Boolean Quartic Forms in eight Variables*, Output of the numerical experiment available at <https://langevin.univ-tln.fr/project/quartics/index.html>. 2007.
 - [34] Langevin, P. & Leander, G. *Classification of Boolean Quartic Forms in eight Variables* in: *Boolean Functions in Cryptology and Information Security* (eds. Preneel, B. & Logachev, O. A.) **18** (IOS Press, 2008), pp. 139–147.
 - [35] Lemmermeyer, F. *Reciprocity Laws: from Euler to Eisenstein* (Springer Monographs in Mathematics, Springer-Verlag, Berlin, 2000).
 - [36] MacWilliams, F. & Sloane, N. *The Theory of Error-correcting Codes* (North-Holland Publishing Company, 1977).
 - [37] Markov, M. & Borissov, Y. *Point-Counting on Elliptic Curves Belonging to One Prominent Family: Revisited* in: *2020 Algebraic and Combinatorial Coding Theory (ACCT)* (2020), pp. 106–109. DOI: 10.1109/ACCT51235.2020.9383390.

- [38] Markov, M. & Borissov, Y. *Weight Distribution of the Binary Reed-Muller Code $R(4,9)$* in: *2024 The Thirteenth International Workshop on Coding and Cryptography (WCC)* (2024), pp. 288–298.
- [39] Massey, J. L. *Deep-space communications and coding: A marriage made in heaven* in: *Advanced Methods for Satellite and Deep Space Communications* (ed. Hagenauer, J.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 1992), pp. 1–17.
- [40] Meng, Q.-S., Zhang, H.-G., Yang, M. & Wang, Z.-Y. Analysis of Affinely Equivalent Boolean Functions. *Science in China Series F: Information Sciences* **50**, pp. 299–306. DOI: 10.1007/s11432-007-0030-9 (2007).
- [41] Miller, V. S. *Use of elliptic curves in cryptography* in: *Conference on the theory and application of cryptographic techniques* (1985), pp. 417–426.
- [42] Muller, D. E. Application of Boolean algebra to switching circuit design and to error detection. *Transactions of the I.R.E. Professional Group on Electronic Computers* **EC-3**, pp. 6–12. DOI: 10.1109/IREPGELC.1954.6499441 (1954).
- [43] Munuera, C. & Tena, J. G. An algorithm to compute the number of points on elliptic curves of j -invariant 0 or 1728 over a finite field. *Rendiconti del Circolo Matematico di Palermo* **42**, pp. 106–116 (1993).
- [44] Peralta, R. A simple and fast probabilistic algorithm for computing square roots modulo a prime number (Corresp.) *IEEE Transactions on Information Theory* **32**, pp. 846–847 (1986).
- [45] Reed, I. A class of multiple-error-correcting codes and the decoding scheme. *Transactions of the IRE Professional Group on Information Theory* **4**, pp. 38–49. DOI: 10.1109/TIT.1954.1057465 (1954).
- [46] Sarwate, D. V. *Weight enumeration of Reed-Muller codes and cosets* Advisors: E. R. Berlekamp and J. D. Ullman. PhD thesis (Princeton University, Princeton, NJ, Aug 1973).
- [47] Schoof, R. Counting points on elliptic curves over finite fields. *Journal de théorie des nombres de Bordeaux* **7**, pp. 219–254 (1995).
- [48] Serret, J.-A. Sur un théorème relatif aux nombres entiers. (in French). *Journal de Mathématiques Pures et Appliquées 1e série*, **13**, pp. 12–14 (1848).
- [49] Silverman, J. H. *The Arithmetic of Elliptic Curves* (Springer, 2009).
- [50] Sloane, N. *The On-Line Encyclopedia of Integer Sequences* available at https://oeis.org/wiki/List_of_weight_distributions. 2024.

-
- [51] Sloane, N. & Berlekamp, E. Weight enumerator for second-order Reed-Muller codes. *IEEE Transactions on Information Theory* **16**, pp. 745–751. DOI: 10.1109/TIT.1970.1054553 (1970).
 - [52] Sugino, M., Ienaga, Y., Tokura, N. & Kasami, T. Weight distribution of (128, 64) Reed-Muller code (Corresp.) *IEEE Transactions on Information Theory* **17**, pp. 627–628. DOI: 10.1109/TIT.1971.1054678 (1971).
 - [53] Sugita, T., Kasami, T. & Fujiwara, T. The weight distribution of the third-order Reed-Muller code of length 512. *IEEE Transactions on Information Theory* **42**, pp. 1622–1625. DOI: 10.1109/18.532911 (1996).
 - [54] van Tilborg, H. C. A. Elliptic curve cryptosystems; too good to be true? *Nieuw Archief Voor Wiskunde* **5**, pp. 220–225 (2001).
 - [55] van Tilborg, H. C. A. *Weights in the third-order Reed-Muller codes* in: *Technical Report 32-1526* **4** (NASA Jet Propulsion Laboratory, California Institute of Technology, 1971).
 - [56] von zur Gathen, J. & Gerhard, J. *Modern Computer Algebra, Third Edition* DOI: <https://doi.org/10.1017/CBO9781139856065> (Cambridge University Press, 2013).
 - [57] Washington, L. C. *Elliptic Curves: Number Theory and Cryptography* (CRC press, 2008).
 - [58] Wilker, P. An Efficient Algorithmic Solution of the Diophantine Equation $u^2 + 5v^2 = m$. *Mathematics of Computation* **35**, pp. 1347–1352 (1980).
 - [59] Wright, S. *Quadratic Residues and Non-Residues: Selected Topics* 2016. arXiv: 1408.0235 [math.NT].

Приложение А

$\mathcal{R}(4, 9)$ таблици

Таблица А.1 Размери на разбиванията $\Delta(e)$ и $\Delta'(e)$

$e \in \mathcal{E}(4, 7)$: ANF съгласно ([32])	$ \Delta(e) $	$ \Delta'(e) $
0	12	12
4567	63	52
1235+1345+1356+1456+2346+2356+2456	130	112
2367+4567	289	182
1237+4567	480	306
1257+1367+4567	730	395
1237+1247+1357+2367+4567	204	157
1236+1257+1345+1467+2347+2456+3567	1098	675
1236+1356+1567+2357+2467+2567+3456	1340	811
1367+2345+2356+3456+4567	6449	2170
1234+1237+1267+1567+2345+3456+4567	23988	3377
1236+1367+1567+2345+3456+3457+3467	33660	4636

Таблица А.2 Соответствие между $\mathcal{E}'(4, 7)$ и $\mathcal{E}(4, 7)$

Разпределение на $\mu(e)$	$\mathcal{E}'(4, 7)$	$\mathcal{E}(4, 7)$	Линейна трансформация на прехода
3	0	0	[100000 0100000 0010000 0001000 0000100 0000010 0000001]
2	123	123	[100000 0100000 0010000 0001000 0000100 0000010 0000001]
21	127+136+145	137+147+157+237+247+267+467	[0011001 0011110 0100110 1011000 1111010 1001100 0001100]
15	125+134	123+145	[1000000 0100000 0001000 0000100 0010000 0000010 0000001]
89	126+345	123+456	[1000000 0100000 0001000 0000100 0000010 0010000 0000001]
56	126+135+234	123+245+346	[0100000 0010000 0001000 0000010 0000100 1000000 0000001]
10	135+146+235+236+245	123+145+246+356+456	[1000000 0000010 0001000 0010000 0000100 0100000 0000001]
7	127+136+145+234	124+137+156+235+267+346+457	[0110001 1011001 0110011 0111010 1100101 0010111 1001011]
502	125+134+135+167+247+357	127+134+135+146+234+247+457	[0001000 0010000 0000001 0000100 0100000 0000010 1100110]
1	123+247+356	123+127+147+167+245	[0010000 0110011 1010000 0001110 0000001 0010011 0000100]
1	147+156+237+246+345	123+127+167+234+345+456+567	[0101010 1001010 1001001 1111111 0011000 0100010 1001011]
292	127+146+236+345	125+126+127+167+234+245+457	[0100111 0001110 0110110 1011000 0000010 0000100 0010110]

Таблица А.3 Теглово разпределение на [512,256,32] Рид-Малер кода

Тегло		Брой на кодовите думи
0	512	1
32	480	52955952
48	464	919315326720
56	456	271767121346560
60	452	860689275027456
64	448	89163020044002040
68	444	1777323352931696640
72	440	64959328938397057024
76	436	2094952122987829002240
80	432	86129855718211879936768
84	428	3718387228743293604986880
88	424	216407674400647746861465600
92	420	15958945395035022932054114304
96	416	1570964763114053055495174389136
100	412	207755244457303752035637154283520
104	408	34164336816436357675455725024378880
108	404	5992987676360073735151889707696128000
112	400	983217921810034263357552475089021004288
116	396	140881159168600922710983130625456163782656
120	392	17178463264607761296016540993629780705771520
124	388	1770270551281316280504947079180771901717872640
128	384	154198773988541804525321284585063483246993999900
132	380	11380437366712812474455950864177326068447989202944
136	376	713793445298874211607839796879716106185715280216064
140	372	38161660034401312989486264769054124765959796671119360
144	368	1744077996406613042017016863461234839306732612077058560
148	364	68320936493023612641136928149296775084064365913214812160
152	360	229974420480046580245331663759578382910891280202806751744
156	356	66674424868716978552789375387240003239187186349775851094016
160	352	1668559700964160587350805664583122924498928358151715733007408
164	348	36117082274027891545154187373048131661136552390031364702863360
168	344	677483598989547107793615101247739514269621184741356041461104640
172	340	11032441933713096201663286389373184730113421621201515757397082112
176	336	156225095497619813307679231937780861426835567156776476525084177664
180	332	1926667532217097161576702991776654344250440175688196887457279508480
184	328	20723534026876536792281002394151796205045793736436788802938336133120
188	324	194671442741837852939975553363771856234841259238404365556287065292800
192	320	1599044990181340998819270766161596605692512085057170791477694075282632
196	316	11498415685246302189888474222781442491860129957714864173250891967627264
200	312	72459467570743603819378812718772497540870770484626494838959726267809792
204	308	400549932263936554220342987258224499780564121712827465674395223861493760
208	304	1944071611978423909059426198144849863064608675044397429548995177751732480
212	300	8291211853278378544436157221213736835450108801042695204524353086973542400
216	296	31095502600701130763682713427899390240950550846409105550583369693522427904
220	292	102622652435510219354959437959897900434480615845926142166854426192158654464
224	288	298206281302110726623000750445450132512881810629607123478473554095237810960
228	284	763396919631666688676755106996803883003881847438728311891109384630797598720
232	280	1722452776176219896357452486934573175804665343735169479919087899582551687168
236	276	3426750460257305904470547641506642175867699465315478403354123631366508642304
240	272	6013163599489683999312799935491777179772724247998877953378442920501417933824
244	268	9309551320248854051332692772889245412495562988894547412532818045057116405760
248	264	12718986044129514620716674156341900030463015021774940408815989741288144568320
252	260	15336997499945305387056357527918950456934399969250231086077675815418680311808
256		16324199909251682000435577287934368523097397692548071777837483832108326674502

Приложение Б

- *Delta компресия:* представлява съхраняване на **сортирани** цели числа във формата на разлики (делти) между тях. Нека разгледаме следната последователност:

260, 261, 265, 270, 280, 295, 301, 312, 320, 324

Можем да съхраним тези 10 стойности като 2-байтови цели числа, използвайки общо 20 байта, но вместо това от всяко число ще извадим предходното и ще кодираме тази разлика:

1, 4, 5, 10, 15, 6, 11, 8, 4

Накрая, съхраняваме началната стойност 260 като 2-байтово цяло число, а останалите 9 отмествания съхраняваме като 1-байтови цели числа, използвайки общо 11 и спестявайки 9 байта.

- *VByte кодиране:* 'байт-ориентирана схема', при която първо опитваме да съхраним дадено цяло число в един байт. Ако не може, използваме два байта и така нататък. Във всеки байт старшият бит е контролен а останалите 7 бита са информационни. Контролният бит се задава на 0, когато това е последният байт с който се кодира числото, в противен случай е 1. Да разгледаме няколко примера:

6 →

0	0	0	0	0	1	1	0
---	---	---	---	---	---	---	---

127 →

0	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---

128 →

0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Изнесени доклади по темата на дисертацията

- [T1] Borissov, Y. & Markov, M. *An Approach for Computing the Number of Points on Elliptic Curve $y^2 = x^3 + a \pmod{p}$ via Explicit Formula for That Number Modulo p* 2019 Ninth International Workshop on Signal Design and its Applications in Communications (IWSDA), Dongguan, China, 20-24 October 2019.
- [T2] Borissov, Y. & Markov, M. *An Approach for Computing the Number of Points on Elliptic Curve $y^2 = x^3 + a \pmod{p}$ via Exact Formula for That Number Modulo p* Национален семинар по теория на кодирането "Професор Стефан Додунеков", Чифлика, България, 21-24 Ноември 2019.
- [T3] Markov, M. & Borissov, Y. *Computing the Number of Points on Elliptic Curves Belonging to One Popular Family: Revisited* 17th International Workshop on Algebraic and Combinatorial Coding Theory (ACCT), online, Bulgaria, 11-17 October 2020.
- [T4] Markov, M. & Borissov, Y. *On the weight distribution of the binary Reed-Muller code $RM(4, 9)$* Национален семинар по теория на кодирането "Професор Стефан Додунеков", Арбанаси, България, 10-13 Ноември 2022.
- [T5] Markov, M. & Borissov, Y. *Our experience with NCHDC resources, gained in finding the weight distribution of the binary Reed-Muller code $\mathcal{R}(4, 9)$* High-Performance Computing for Mathematics and Applications, Sofia, Bulgaria, 28 June 2023.
- [T6] Markov, M. & Borissov, Y. *On an Approach for Computing the Weight Distribution of a Binary Reed-Muller Code II* Национален семинар по теория на кодирането "Професор Стефан Додунеков", Хисаря, България, 2-5 Ноември 2023.

- [T7] Markov, M. & Borissov, Y. *Weight Distribution of the Binary Reed-Muller Code $\mathcal{R}(4, 9)$* The Thirteenth International Workshop on Coding and Cryptography (WCC), Perugia, Italy, 17-21 June 2024.

Публикации по темата на дисертацията

- [P1] Borissov, Y. & Markov, M. *An Approach for Computing the Number of Points on Elliptic Curve $y^2 = x^3 + a \pmod{p}$ via Explicit Formula for That Number Modulo p* in: *2019 Ninth International Workshop on Signal Design and its Applications in Communications (IWSDA)* (2019), pp. 1–5. <https://ieeexplore.ieee.org/document/8966127>.
- [P2] Markov, M. & Borissov, Y. *Point-Counting on Elliptic Curves Belonging to One Prominent Family: Revisited* in: *2020 Algebraic and Combinatorial Coding Theory (ACCT)* (2020), pp. 106–109.
- [P3] Borissov, Y. & Markov, M. An Efficient Approach to Point-Counting on Elliptic Curves from a Prominent Family over the Prime Field \mathbb{F}_p . *Mathematics* **9**. <https://www.mdpi.com/2227-7390/9/12/1431> (2021).
- [P4] Markov, M. & Borissov, Y. *Weight Distribution of the Binary Reed-Muller Code $R(4,9)$* in: *2024 The Thirteenth International Workshop on Coding and Cryptography (WCC)* (2024), pp. 288–298. https://wcc2024.sites.dmi.unipg.it/WCC_proceedings.pdf.