Załącznik 2 (wersja polska)

Autoreferat

1. **Imię i nazwisko:** Maciej Grześkowiak.

2. **Posiadane dyplomy, stopnie naukowe/artystyczne — z podaniem nazwy, miejsca i roku ich uzyskania oraz tytułu rozprawy doktorskiej**.

   magister informatyki (1999) Wydział Matematyki i Informatyki, Uniwersytet im. Adama Mickiewicza, Poznań.

   doktor nauk matematycznych w zakresie informatyki (2004) Wydział Matematyki i Informatyki, Uniwersytet im. Adama Mickiewicza, Poznań, rozprawa „Analiza algorytmów generowania kluczy w systemie XTR", promotor: prof. dr hab. J. Kaczorowski.

3. **Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych**.

   adiunkt od 2004 roku. Wydział Matematyki i Informatyki, Uniwersytet im. Adama Mickiewicza w Poznaniu.

4. **Wskazanie osiągnięcia wynikającego z art. 16 ust. 2 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz. U. 2016r. poz. 882 ze zm. w Dz. U. z 2016 r. poz. 1311.):**

   (a) **tytuł osiągnięcia naukowego:**
   Algorytmiczne metody generowania liczb pierwszych w systemach kryptograficznych.

   (b) **(autor/autorzy, tytuł/tytuły publikacji, rok wydania, nazwa wydawnictwa)**

   [H1] Maciej Grześkowiak. Algorithm for generating primes p and q such that q divides $p^4 \pm p^3 + p^2 \pm p + 1$. *Fundam. Inform.*, 114(3-4):287–299, 2012.

   [H2] Maciej Grześkowiak. Algorithms for relatively cyclotomic primes. *Fund. Inform.*, 125(2):161–181, 2013.

   [H3] Maciej Grześkowiak. Algorithms for pairing-friendly primes. *Pairing-Based Cryptography 2013, Lecture Notes in Computer Science*, 8365:215–288, 2014. (to appear).

   [H4] Maciej Grześkowiak. An algorithmic construction of finite elliptic curves of order divisible by a large prime. *Fund. Inform.*, 136(4):331–343, 2015.

   [H5] Maciej Grześkowiak. Pairing-friendly primes for abelian varieties. *Fund. Inform.*, 149(4):385–400, 2016.

   [H6] Maciej Grześkowiak. Explicit bound for the prime ideal theorem in residue classes. *Number-Theoretic Methods in Cryptology 2017, Lecture Notes in Computer Science*, 10737:48–68, 2018.

   (c) **omówienie celu naukowego ww. prac i osiągniętych wyników.**

   Wprowadzenie

   Celem naukowym badań proponowanych w postępowaniu habilitacyjnym jest znajdowanie algorytmicznych rozwiązań następującego problemu z obliczeniowej teorii liczb, który w ogólnej postaci można zdefiniować następująco. Niech G będzie skończoną grupą abelową oraz niech H będzie podgrupą właściwą grupy G. Oznaczny przez $(G : H)$ indeks podgrupy H w grupie G.

Problem Ogólny. *Dla ustalonej rodziny skończonych grup abelowych $\mathcal{F}$ skonstruować $G \in \mathcal{F}$ oraz jej podgrupę właściwą H o możliwie małym indeksie $(G : H)$. Zaproponowana metoda powinna działać w czasie wielomianowym ze względu na liczbę bitów rzędu grupy G.*

Dla potrzeb kryptologicznych ważne rodziny $\mathcal{F}$ to multiplikatywne grupy ciała skończonego, grupy puntów krzywych eliptycznych nad ciałami skończonymi oraz jakobiany krzywych hipereliptycznych nad ciałami skończonymi. Kostrukcja grup H oraz G, które spełniają założenia Problemu Ogólnego, jest niezbędna podczas implementacji asymetrycznych protokołów kryptograficznych. Wielu autorów projektowało algorytmy generujące grupy H i G, które spełniają powyższe założenia. Często jednak, analiza złożoności obliczeniowej proponowanych metod była przeprowadzona przy założeniu prawdziwości pewnych hipotez, które spotykamy w teorii liczb lub wykorzystywała argumenty heurystyczne.

W celu zilustrowania problemów, które można napotkać podczas projektowania i analizy algorytmów znajdujących rozwiązania Problemu Ogólnego rozważmy następujący przykład. Niech G będzie grupą multiplikatywną ciała skończonego $\mathbb{F}_p$, gdzie p jest liczbą pierwszą. Wtedy Problem Ogólny możemy zapisać w postaci.

Problem 1. *Znajdź dwie liczby pierwsze p oraz q takie, że*

$$q < p, \qquad q \mid p-1, \qquad \frac{p-1}{q} = s,$$

gdzie s jest małe lub $s = rt$, gdzie r jest dużą (znaną) liczbą pierwszą oraz t jest małe. Zaproponowana metoda powinna działać w czasie wielomianowym ze względu na liczbę bitów p oraz q.

Podczas konstruowania algorytmów, które znajdują liczby pierwsze q oraz p spełniające założenia Problemu 1, należy uwzględnić dwa dodatkowe warunki związane bezpośrednio z rzędem wielkości q oraz p. Pierwszy z nich odpowiada za bezpieczeństwo kryptosystemu opartego na grupie $\mathbb{F}_p^*$, a drugi dotyczy aspektów implementacyjnych oraz praktycznych.

Bezpieczeństwo protokołów zbudowanych w oparciu o $\mathbb{F}_p^*$ jest związane jest z następującym problemem obliczeniowym w $\mathbb{F}_p^*$. Niech $g, y \in \mathbb{F}_p^*$ będą dane. Znajdź, o ile istnieje $x \in \mathbb{N}$, $1 \leq x < p$ takie, że $g^x \equiv y \pmod{p}$. Liczbę x nazywamy logarytmem dyskretnym z y przy podstawie g w grupie $\mathbb{F}_p^*$. Jednym z warunków bezpieczeństwa protokółu kryptograficznego opartego na $\mathbb{F}_p^*$ jest z trudność obliczania logarytmu dyskretnego w $\mathbb{F}_p^*$. Dla $n \in \mathbb{N}$ zdefiniujmy funkcję

$$L_n(\alpha, c) = \exp((c + o(1))(\log n)^{\alpha}(\log\log n)^{1-\alpha})),$$

gdzie $a, c \in \mathbb{R}$, $0 \leq a \leq 1$, $c > 0$ [17]. Logarytm dyskretny w $\mathbb{F}_p^*$ może być wyznaczony metodą obliczania indeksów i wywaga wykonania $L_p(\frac{1}{2}, \sqrt{2})$ operacji arytmetycznych w $\mathbb{F}_p^*$ [53]. Metoda sita ciał liczbowych jest efektywniejsza i wymaga wykonania $L_p(\frac{1}{3}, 3^{2/3})$ operacji w $\mathbb{F}_p^*$ [33]. Biorąc pod uwagę złożoność obliczeniową powyższych algorytmów, protokoły krytograficzne, których bezpieczeństwo zależy od trudności obliczenia logarytmu dykretnego w $\mathbb{F}_p^*$, powinny być implementowane w grupie, dla której $p \geq 2^{1024}$. Niech H będzie podgrupą $\mathbb{F}_p^*$ rzędu q, gdzie q jest liczbą pierwszą. Istnieje wiele algorytmicznych metod znajdowania logarytmu dysktetnego w grupie H. Najefektywniejsze z nich wymagają wykonania $O(\sqrt{q})$ operacji arytmetycznych w $\mathbb{F}_p^*$ [51], [52], [66]. Dlatego, aby protokoł kryptogarficzny oparty na $\mathbb{F}_p^*$ był trudny do złamania należy generować podgrupę H rzędu $q \geq 2^{160}$.

Aspekty praktyczne związane z implementacją poszczególnych protokołów kryptologicznych dotyczą postaci generownych liczb pierwszych q oraz p występujących w Problemie 1. Na

przykład, często wymaga się znalezienia losowego elementu dużego rzędu w $\mathbb{F}_p^*$, w szczególności elementów rzędu q lub $p-1$. Algorytm znajdujący losowy generator w $\mathbb{F}_p^*$ wymaga znajomości wszystkich czynników pierwszych $p-1$ [47]. Z powyższego wynika, że aby efektywnie znaleźć generator podczas implementacji, liczba $(p-1)/q$ powinna być mała lub być iloczynem znanych liczb pierwszych. Biorąc pod uwagę wspomniany czynnik praktyczny, projektowany algorytm powinien generować dwie duże liczby pierwsze p oraz q w Problemie 1 spełniające $p = 2q+1$. Liczby pierwsze tej postaci zapewniają wysoki poziom bezpieczeństwa protokołu opartego na $\mathbb{F}_p^*$ i szybki algorytm znajdowania elementów dużego rzędu w tej grupie.

Z drugiej strony, analiza złożoności obliczeniowej algorytmu znajdującego dwie liczby pierwsze q, p takie, że $p = 2q+1$ jest trudna. Nie wiadomo, czy istnieje nieskończenie wiele liczb pierwszych powyższej postaci. Znana jest jedynie następująca hipoteza [65].

Hipoteza 1.

$$\sharp\{q \le x : 2q+1 - \textit{liczba pierwsza}\} \sim c\frac{x}{(\log x)^2},$$

gdzie

$$c = 2\prod_{p>2}\frac{p(p-2)}{(p-1)^2} \approx 1.32032.$$

Zakładając prawdziwość Hipotezy 1 można łatwo skonstruować algorytm probabilistyczny, który rozwiązuje Problem 1 w czasie wielomianowym ze względu na liczbę bitów p [65].

Warto wspomnieć o kolejnych czynnikach, które można uwzględniać podczas projektowania algorytmów znajdujących parametry protokołów kryptograficznych. Często są to ograniczenia związane z mocą obliczeniową, pojemnością magazynową konkretnych urządzeń cyfrowych oraz z wielkością i liczbą wysyłanych pakietów. Wymagane jest, aby stosowane rozwiązania obliczeniowe i transportowe były efektywne w praktyce. Zatem w Problemie 1 warunek mówiący o tym, że s jest możliwie jak najmniejsze, możemy również interpretować jako możliwie maksymalną *kompresję* generowanych parametrów q oraz p ze względu na efektywność i odporność na znane metody kryptoanalizy.

Niech $n = P_r(a)$ oznacza, że n jest liczbą naturalną, która jest iloczynem co najwyżej r czynników pierwszych q_i takich, że $q_i \ge n^a$, $a > 0$, dla $i = 1, \ldots, r$. W 2013 roku von zur Gathen oraz Shparlinski zaproponowali algorytm [72], który znajduje rozwiązanie Problemu 1, gdy $s = 2$ oraz $p - 1 = 2P_2(a)$. Algorytm działa w czasie wielomianowym dla dostatecznie dużego p, a analiza jego złożoności obliczeniowej wykorzystuje rezultat uzyskany przez Heatha-Browna [36].

Lemat 1. *Niech $k = 1, 2, 3$ oraz $K = 2^k$. Niech $u, v \in \mathbb{N}$, $(u, v) = 1$ będą takie, że $K \mid u-1$, $16 \mid v$ oraz $(\frac{u-1}{K}, v) = 1$. Wtedy istnieje $a \in (\frac{1}{4}, \frac{1}{2}]$, $a = a(k, u, v)$ taka, że*

$$\sharp\left\{p \le x : p \equiv u \pmod v, \quad \frac{p-1}{K} = P_2(a)\right\} \ge c\frac{x}{(\log x)^2},$$

gdzie $c > 0$, $c = c(a)$.

Nie można jednak stwierdzić na podstawie udowodnionego rezultatu autorów [72] czy rząd wielkości powyższych liczb pierwszych, dla których algorytm działa w czasie wielomianowym, jest zgodny z oczekiwaniami praktycznymi. Odpowiedź na to pytanie wymagałaby między innymi wyznaczenia numerycznej wartości stałej c z Lematu 1.

Warto dodać, że podczas procesu standaryzacji protokołów kryptograficznych często wybiera i ustala się tylko jedną (ewentualnie kilka) grupę $\mathbb{F}_p^*$, na której opiera się protokół.

Liczba p jest wybrana w ten sposób, aby implementowana redukcja i arytmetyka modulo p była efektywna. W ogólności, wykorzystywanie w systemie kryptograficznym ustalonej grupy $\mathbb{F}_p^*$ nie wpływa negatywnie na bezpieczeństwo protokołu, którego bezpieczeństwo zależy od trudności obliczania logarytmu dykretnego w $\mathbb{F}_p^*$. Jednak, nie można wykluczyć, że w ten sposób zbudowany protokół nie będzie narażony na specjalny - dedykowany tylko grupie $\mathbb{F}_p^*$ - atak. Z drugiej strony, służby wojskowe nie budują własnych systemów kryptograficznych w oparciu o standaryzowane struktury algebraiczne, lecz generują losowe - często tajne - parametry. Z tego powodu praca [72] dostarcza fundamentów teoretycznych do stosowanych (jawnych oraz tajnych) algorytmów praktycznych, co w konsekwencji może odbijać się na liczbie cytowań tej pracy wśród praktyków kryptologii.

Autor nimniejszego autoreferatu przyjął założenie, że konstruowane przez niego algorytmy muszą nie tylko działać efektywnie w praktyce, ale analiza ich złożoności obliczeniowej powinna dowodzić, że działają one w czasie wielomianowym ze względu na liczbę bitów danych wejściowych, bez zakładania dodatkowych hipotez i przytaczania argumentów heurystycznych. Autor, w swoich badaniach nad rozwiązaniem Problemu Ogólnego, skupił się też nad wyznaczeniem rzędu wielkości generowanych parametrów, dla których można udowodnić, że proponowane metody algorytmiczne działają w czasie wielomianowym. Przyjęcie powyższych założeń powoduje, że przeprowadzone badania autora tego autoreferatu możemy również zakwalifikować do prac z podstaw informatyki teoretycznej. Ze względu na rozpatrywane systemy kryptograficzne z kluczem publicznym wyniki dotychczasowych badań autora tego autoreferatu możemy podzielić na cztery grupy.

1. Efektywne kryptosytemy w ciałach skończonych.

W ostatnich latach zaproponowano klasę kryptosystemów z kluczem publicznym, w której wykorzystuje się efektywny sposób reprezentowania ciała skończonego $\mathbb{F}_{p^n}$ za pomocą elementów jego podciała $\mathbb{F}_{p^d}^*$, gdzie $d \mid n$, $d < n$ [32], [44], [43], [56], [57], [71]. W tej metodzie elementy $\mathbb{F}_{p^n}$ reprentowane są za pomocą ich śladu względem podciała $\mathbb{F}_{p^d}$. Dzięki temu reprezentacja ciała $\mathbb{F}_{p^n}$ wymaga mniejszej liczby bitów niż w klasycznych reprezentacjach tego ciała. W pracy [56] wprowadzono pojęcie *torus-based cryptography*, która obejmuje całą rodzinę kryptosystemów tego typu. Niech $\Phi_n(X) \in \mathbb{Z}[X]$ oznacza n-ty wielomian podziału koła, gdzie n jest ustaloną liczbą naturalną. Niech $G \subset \mathbb{F}_{p^n}$ będzie podgrupą grupy $\mathbb{F}_{p^n}^*$ rzędu $q > k$, gdzie $q \mid \Phi_k(p)$. Wtedy $G \not\subset \mathbb{F}_{p^d}^*$, dla $d \mid n$, $d < n$ [45]. Dla kryptosystemu z rodziny *torus-based* Problem Ogólny ma następującą postać.

Problem 2. *Znajdź dwie liczby pierwsze p oraz q takie, że*

$$q \mid \Phi_n(p), \qquad \frac{\Phi_n(p)}{q} = s, \tag{1}$$

gdzie s jest możliwie małe. Zaproponowana metoda powinna działać w czasie wielomianowym ze względu na liczbę bitów p oraz q.

Dla $n = 6$, w pracy [44] zaproponowano dwie, szybko działające w praktyce, metody generowania liczb pierwszych postaci (1). Parametry tej postaci są wykorzystywane w kryptosystemie XTR. Pierwsza z nich działa w następujący sposób: algorytm znajduje liczbę $r \in \mathbb{N}$ taką, że $\Phi_6(r) = q$ jest liczbą pierwszą. Następnie generuje on $k \in \mathbb{N}$ takie, że $p = qk + r$ jest liczbą pierwszą. W ten sposób algorytm zwarca parę liczb pierwszych p oraz q spełniających (1). Druga strategia znajdowania p oraz q wykonuje następujące kroki: algorytm generuje losowo liczbę pierwszą $q \equiv 7 \pmod{12}$. Następnie, znaduje rozwiązanie

r równania $\Phi_n(X) \equiv 0 \pmod{q}$ i w ostatnim kroku wybiera $k \in \mathbb{N}$ takie, że $p = qk + r$ jest liczbą pierwszą. W wyniku wykonania powyższych kroków otrzymujemy liczby pierwsze p i q spełniające (1). Uogólnienia powyższej techniki generowania parametrów (1) oraz systemu XTR znajdujemy w pracach [31], [56], [62], [71], dla $n = 5, 10, 15, 30, 210$.

Analiza złożoności obliczeniowych powyższych metod wymaga założenia prawdziwości pewnych znanych hipotez z teorii liczb. Mianowicie, ustalmy $n \geq 2$. Nie wiadomo czy istnieje nieskończenie wiele liczb $r \in \mathbb{N}$ takich, że $\Phi_n(r)$ jest liczbą pierwszą. Przy założeniu hipotezy Bouniakowskiego oraz Schinzela [12], [59] oraz oszacowań heurystycznych [7] znalezienie liczby $r \in \mathbb{N}$ takiej, że $\Phi_n(r)$ jest pierwsza jest możliwe w czasie wielomianowym [D1], [D2]. Kolejna trudność, którą napotykamy podczas analizy powyższych metod związana jest z założeniem występującym w Problemie 2, który mówi o tym, że s powinno być możliwie małe. W ten sposób wymaga się, aby liczba p była *blisko* liczby q, a więc postuluje wybór niewielkiego k w obu powyższych metodach. Niech $p(q, r)$ będzie najmniejszą liczbą pierwszą w postępie arytmetycznym $qk + r$, $k \geq 0$. Niech $p(q) = \max\{p(q, r) : (q, r) = 1, 1 \leq r < q\}$. Heath-Brown udowodnił, że $p(q) \ll q^{5.5}$ dla dostatecznie dużego q [36]. Ostatnio wynik ten został poprawiony przez Xylourisa [75]. Warto tutaj dodać, że znane twierdzenie Siegela-Walfisza odnosi się tylko do liczb pierwszych $p \equiv r \pmod{q}$, gdzie $p \leq x$, $q \leq (\log x)^N$, $(r, q) = 1$, $N > 0$, dla dostatecznie dużego x [74]. Przy założeniu prawdziwości pewnych hipotez teorii liczb krok ten algorytm wykonuje w czasie wielomianowym [35], [73], [D1]. Zwóćmy również uwagę na krok algorytmu, w którym oblicza się pierwiastek r równania $\Phi_6(X) \equiv 0 \pmod{q}$. Dla $q \equiv 3 \pmod 4$ można go obliczyć w sposób deterministyczny. W ogólności jednak, nie jest znany algorytm deterministyczny, który znajduje rozwiązania równania $\Phi_n(X) \equiv 0 \pmod{q}$, dla $q \equiv 1 \pmod{n}$ w czasie wielomianowym ze względu na liczbę bitów q. Przy założeniu prawdziwości Uogólnionej Hipotezy Riemanna pierwiastek równania $\Phi_n(X) \equiv 0 \pmod{q}$ może być wyznaczony deterministycznie w czasie $O(n(\log q)^5)$, por. Twierdzenie 7.8.7 w pracy [3]. W praktyce wykorzystuje się probabilistyczne metody obliczania pierwiastka pierwotnego n-tego stopnia z jedności modulo q.

W pracy [H1] zaproponowano metodę algorytmiczną, która rozwiązuje Problem 2 dla $n = 5, 10$. Parametry tej postaci są składnikami klucza kryptosystemu Gulianiego-Gonga [31]. Algorytm autora tego autoreferatu działa według na następującego schematu. Niech $K = \mathbb{Q}(\omega)$, $\omega = (1 + \sqrt{5})/2$ będzie rzeczywistym ciałem kwadratowym z pierścieniem liczb całkowitych $\mathcal{O}_K = \{a + b\omega : a, b \in \mathbb{Z}\}$. Niech $N_{K/\mathbb{Q}}(a + b\omega) = a^2 + ab - b^2$ będzie normą $a + b\omega \in \mathcal{O}_K$ względem ciała $\mathbb{Q}$. W pierwszym kroku algorytm znajduje $a, b \in \mathbb{Z}$ takie, że $q = |N_{K/\mathbb{Q}}(a + b\omega)| \equiv 11 \pmod{20}$ jest liczbą pierwszą. Następnie dla danego a, b oraz q algorytm znajduje $r \pmod{q}$ pierwiastek równania $\Phi_n(X) \equiv 0 \pmod{q}$, gdzie $n = 5, 10$. W pracy [H1] autor udowodnił, że krok ten można wykonać w deterministyczny sposób w czasie $O((\log q)^3)$. W ostatniej fazie proponowanej metody, algorytm wybiera $k \in \mathbb{N}$ takie, że $p = qk + r$ jest liczbą pierwszą. Dzięki zastosowanej technice algorytmicznej autor udowodnił między innymi, że kroki pierwszy i drugi działają w czasie wielomianowym ze względu na liczbę bitów q, przy czym pierwszy krok osiągamy z prawdopodobieństwem bliskim jedności, a krok drugi jest deterministyczny [H1]. Ostatni krok tego algorytmu nie był analizowany przez autora. Poniżej przedstawiamy kompletny algorytm autora tego autoreferatu, który składa się trzech procedur. Niech $\varrho(\alpha)$ oznacza geometryczną reprezentację liczby $\alpha \in K$ oraz niech $\mathcal{X}$ będzie obszarem fundamentalnym K [11]. Definiujemy

$$R(x) = \{\alpha \in \mathcal{O}_K : \quad x \leq |N_{K/\mathbb{Q}}(\alpha)| \leq 2x, \quad \varrho(\alpha) \in \mathcal{X}\},$$

$$S(x) = \{\beta \in \mathcal{O}_K : \quad \beta = \omega^i \alpha, \quad \alpha \in R(x), \quad i = 0, \ldots, 59\}.$$

Procedura FindPrimeQ(k,l,x). Ustalmy $k,l \in \mathbb{N}$ takie, że $(k,l)=1$, $k^2+kl-l^2 \equiv 11 \pmod{20}$ oraz liczba $k+l\omega$ jest całkowicie dodatnia. Niech $x>1$. Procedura znajduje $a+b\omega \in S(x)$, gdzie $a \equiv k \pmod{20}$, $b \equiv l \pmod{20}$ oraz $a+b\omega$ jest całkowicie dodatnia taką, że $N_{K/\mathbb{Q}}(a+b\omega)=q$ jest liczbą pierwszą oraz $x \leq q \leq 2x$.

krok 1 Wylosuj $a+b\omega \in S(x)$ takie, że $a \equiv k \pmod{20}$, $b \equiv l \pmod{20}$ oraz $a+b\omega$ jest całkowicie dodatnia.

krok 2 Oblicz $q=N_{K/\mathbb{Q}}(a+b\omega)$. Jeśli q jest liczbą pierwszą, to skok do kroku 3. W przeciwnym przypadku skok do kroku 1.

krok 3 Zwróć a,b oraz q.

Niech $m \in \mathbb{N}$. Oznaczmy przez $\mathcal{PT}$ liczbę operacji na bitach potrzebnych do wykonania deterministycznego testu pierwszości liczby m [1]. Załóżmy, że $\mathcal{PT}$ wymaga co najmniej $O(\log^3 m)$ operacji na bitach

Twierdzenie 1. *Ustalmy $k,l \in \mathbb{N}$ takie, że $(k,l)=1$, $k^2+kl-l^2 \equiv 11 \pmod{20}$ oraz liczba $k+l\omega$ jest całkowicie dodatnia. Istnieją dwie stałe $c>0$ oraz $x_0>0$ takie, że dla każdego $x \geq x_0$ oraz dowolnego $\lambda \geq 1$, procedura* FindPrimeQ *znajduje $c+d\omega \in S(x)$, gdzie $c+d\omega$ jest całkowicie dodatnia oraz $c \equiv k \pmod{20}$, $d \equiv l \pmod{20}$ taką, że $q=N_{K/\mathbb{Q}}(c+d\omega)$ jest liczbą pierwszą, z prawdopodobieństwem wiekszym lub równym $1-e^{-\lambda}$ po powtórzeniu $[c\lambda(\log x)]$ kroków procedury. Każdy krok tej procedury wymaga wykonania co najwyżej $\mathcal{PT}$ operacji na bitach.*

Dowód: zobacz [H1] (Twierdzenie 6.1).

Procedura FindRootModuloQ(a,b,q,n). Niech $n=5$ lub $n=10$. Dla danego q oraz $\alpha=a+b\omega$ takiego, że $q=N_{K/\mathbb{Q}}(\alpha) \equiv 11 \pmod{20}$, procedura oblicza r pierwiastek wielomianu $\Phi_n(x)$ modulo q.

krok 1 Oblicz $z \equiv (a^2-4b^2)^{(q+1)/4} \pmod{q}$.

krok 2 Oblicz $w \equiv (z-a)(-2b)^{-1} \pmod{q}$.

krok 3 Jeśli $n=5$, to $r=w$. jeśli $n=10$, to $r \equiv -w \pmod{q}$.

krok 4 Zwróć r.

Twierdzenie 2. *Procedura* FindRootModuloQ *wymaga wykonania nie więcej niż $O((\log q)^3)$ operacji na bitach.*

Dowód: zobacz [H1] (Twierdzenie 7.1).

Procedura FindPrimeP(r,q). Niech liczba pierwsza q oraz $r<q$ będą dane. Procedura znajduje $p \equiv r \pmod{q}$.

krok 1 Wylosuj $m \in \mathbb{N}$.

krok 2 Oblicz $p=qm+r$. Jeśli p jest liczbą pierwszą, to zakończ procedurę. W przeciwnym przypadku idź do kroku 1.

krok 3 Zwróć p.

Algorytm 1 Ustalmy $k,l \in \mathbb{N}$ takie, że $(k,l)=1$, $k^2+kl-l^2 \equiv 11 \pmod{20}$ oraz liczba $k+l\omega$ jest całkowicie dodatnia. Dla danych $n=5$ lub $n=10$ oraz $x>1$ algorytm znajduje liczby pierwsze p oraz q takie, że $q \mid \Phi_n(p)$.

krok 1 $a,b,q :=$ FindPrimeQ(k,l,x);

krok 2 $r :=$ FindRootModuloQ(a,b,q,n);

krok 3 $p := \text{FindPrimeP}(r, q)$;
krok 4 Zwróć p oraz q.

Twierdzenie 3. *Niech $n = 5$ lub $n = 10$. Algoritm 1 generuje liczby pierwsze p oraz q takie, że q dzieli $\Phi_n(p)$.*

Dowód: zobacz [H1] (Twierdzenie 5.1).

W pracy [H2] zaproponowano algorytm, który rozwiązuje Problem 2 dla dowolnej ustalonej liczby naturalnej $n \geq 2$. Niech $B > 0$ będzie ustalone. Autor udowodnił, że dla $n < B$, jego metoda działa w czasie wielomianowym ze względu na liczbę bitów q oraz p dla prawie wszystkich liczb pierwszych q. Chociaż główne kroki algorytmu z pracy [H2] odpowiadają tym z [H1], to metoda konstrukcji parametrów (1) istotnie się różni od tej proponowanej w [H1]. W szczególności różnica dotyczy sposobu obliczania pierwiastków pierwotnych n-tego stopnia z jedności modulo q.

Niech ω będzie pierwiastkiem pierwotnym n-tego stopnia z jedności. Niech $K = \mathbb{Q}(\omega)$ będzie ustalonym ciałem cyklotomicznym stopnia $[K : \mathbb{Q}] = 2t$, gdzie t jest liczbą urojonych włożeń ciała K w ciało $\mathbb{C}$. Niech $\mathcal{O}_K$ będzie pierścieniem liczb algebraicznych całkowitych ciała K. Idea proponowanej techniki obliczeniowej zaproponowanej przez autora tego autoreferatu jest następująca. W pierwszym kroku algorytm znajduje losową liczbę $\alpha \in \mathcal{O}_K$ taką, że jej norma $N_{K/\mathbb{Q}}(\alpha) = q$ jest liczbą pierwszą. Następnie, dla danych α, q algorytm znajduje $r \pmod q$ takie, że $\Phi_n(r) \equiv 0 \pmod q$. W ostatnim kroku algorytm losuje $k \in \mathbb{N}$ takie, że $p = qk + r$ jest liczbą pierwszą. Jak wspomniano wcześniej, proponowana technika obliczeniowa oraz jej analiza różni się zasadniczo od tej opisanej w [H1] w trzech aspektach. Liczba q jest generowana w poprzez elementy ciała cyklotomicznego, które jest naturalnie związane z pierwiatkami wielomianu $\Phi_n(X)$. Dzięki temu, autor skonstruował nową (nie będącą uogólnieniem wcześniejszgo algorytmu) metodę znajdowania rozwiązania $r \pmod q$ równania $\Phi_n(X) \equiv 0 \pmod q$, dla danego α oraz q. Autor udowodnił, że krok ten jest deterministyczy i działa w czasie wielomianowym dla ustalonego n i dostatecznie dużego q. Ponadto, autor przeprowadził analizę algorytmu i wykazał, że można wygenerować w probalistyczny sposób liczby pierwsze spełniające (1) takie, że $x \leq q \leq 2x$ oraz $q \leq p \ll q^2(\log q)^{20}$ dla dostatecznie dużego x oraz prawie wszystkich q z kroku pierwszego algorytmu.

Algorytm generejący liczby pierwsze spełniające założenia Problemu 2 wymaga wykonania trzech poniższych procedur [H2].

Procedura $\text{FindPrimeQ}(n)$. Ustalmy $n \in \mathbb{N}$. Niech ω będzie pierwiatskiem pierwotnym n-tego stopnia z jedności. Niech $K = \mathbb{Q}(\omega)$ będzie ustalonym ciałem cyklotomicznym stopnia $[K : \mathbb{Q}] = 2t$, gdzie t jest liczbą urojonych włożeń ciała K w ciało $\mathbb{C}$. Niech $\varepsilon_1, \ldots, \varepsilon_r$, będzie systemem fundamentalnym jednostek ciała K, gdzie $r = t-1$. Definiujemy

$$M = M(n) = \max_{1 \leq i \leq r} \{\log |\sigma_j(\varepsilon_i)|, \quad j = 1, \ldots, t\}.$$

Niech $\omega_1 = \omega, \omega_2, \ldots \omega_{\varphi(n)}$ będą elementami sprzężonymi z ω. Ponadto, definiujemy

$$C = C(n) = \max\{|v_{i,j}|, \quad i = 1, \ldots, \varphi(n), \quad j = 1, \ldots, \varphi(n)\}. \tag{2}$$

gdzie

$$\begin{bmatrix} 1 & \omega_1 & \cdots & \omega_1^{\varphi(n)-1} \\ 1 & \omega_2 & \cdots & \omega_2^{\varphi(n)-1} \\ \vdots & \vdots & \ddots & \cdots \\ 1 & \omega_{\varphi(n)} & \cdots & \omega_{\varphi(n)}^{\varphi(n)-1} \end{bmatrix}^{-1} = \begin{bmatrix} \upsilon_{1,1} & \upsilon_{1,2} & \cdots & \upsilon_{1,\varphi(n)} \\ \upsilon_{2,1} & \upsilon_{2,2} & \cdots & \upsilon_{2,\varphi(n)} \\ \vdots & \vdots & \ddots & \cdots \\ \upsilon_{\varphi(n),1} & \upsilon_{\varphi(n),2} & \cdots & \upsilon_{\varphi(n),\varphi(n)} \end{bmatrix}$$

Procedura znajduje liczbę $\alpha = \sum_{i=1}^{\varphi(n)} a_i \omega^{i-1} \in \mathcal{O}_K$ taką, że $N(\alpha) \equiv 1 \pmod{n}$ jest liczbą pierwszą oraz $x \leq N_{K/\mathbb{Q}}(\alpha) \leq 2x$ and $|a_i| < C\varphi(n)e^{rM}(2x)^{1/\varphi(n)}$.

krok 1. Dla $i = 1, \ldots, \varphi(n)$, losuj $a_i \in \mathbb{Z}$ takie, że $|a_i| < C\varphi(n)e^{rM}(2x)^{1/\varphi(n)}$. Niech $\alpha = a_1 + a_2\omega + \ldots + a_{\varphi(n)}\omega^{\varphi(n)-1} \in \mathcal{O}_K$.

krok 3. Oblicz $q = N_{K/\mathbb{Q}}(\alpha)$. Jeśli $q < x$ lub $q > 2x$, to idź do kroku 1.

krok 4. Jeśli q jest liczbą pierwszą, to koniec procedury. W przeciwnym przypadku idź do kroku 1.

krok 5. Zwróć $a_1, \ldots, a_{\varphi(n)}$, q oraz $A(\alpha)$ takie, że $\det(A(\alpha)) = q$.

Twierdzenie 4. *Dla danego $n \in \mathbb{Z}$, $n \geq 2$, istnieją dwie stałe $c_0 > 0$ oraz $x_0 > 0$ takie, że każdego $x \geq x_0$ i dowolnej liczby rzeczywistej $\lambda \geq 1$, procedura* FindPrimeQ *znajduje*

$$\alpha = \sum_{i=1}^{\varphi(n)} a_i \omega^{i-1} \in \mathcal{O}_K, \quad |a_i| < C\varphi(n)e^{rM}(2x)^{1/\varphi(n)}$$

takie, że

$$N_{K/\mathbb{Q}}(\alpha) \equiv 1 \pmod{n}, \quad x \leq N(\alpha) \leq 2x,$$

jest liczbą pierwszą, z prawdopodobieństwem większym lub równym $1 - e^{-\lambda}$ po powtórzeniu $[c_0\lambda(\log x)]$ kroków procedury. Każdy krok tej procedury wymaga wykonania co najwyżej $\mathcal{PT}$ operacji na bitach.

Dowód: zobacz [H2] (Twierdzenie 2.1).

Procedura FindRootModQ$(\alpha, A(\alpha), q)$. Znajdź $n \in \mathbb{Z}$, $n \geq 1$. Dla danego $\alpha \in \mathcal{O}_K$ i liczby pierwszej q takiej, że $N_{K/\mathbb{Q}}(\alpha) = q \equiv 1 \pmod{n}$, gdzie $N_{K/\mathbb{Q}}(\alpha) = \det(A(\alpha))$, procedura znajduje pierwiastek równania $\Phi_n(x) \equiv 0 \pmod{q}$.

krok 1. Wyznacz macierz $M = [A(\alpha)^T | C]_{\varphi(n)\times\varphi(n)+1}$, która jest macierzą otrzymaną z macierzy $A(\alpha)^T$ przez dołączenie wektora C, gdzie $C^T = [y, -1, 0, \ldots, 0]_{1\times\varphi(n)}$.

krok 2. Za pomocą algorytmu eliminacji Gaussa sprowadź macierz M do postaci górnej trójkątnej.

$$M' = \left[\begin{array}{cccc|c} a'_{1,1} & a'_{2,1} & \cdots & a'_{\varphi(n),1} & c'_1 \\ 0 & a'_{2,2} & \cdots & a'_{\varphi(n),2} & c'_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a'_{\varphi(n),\varphi(n)} & c'_{\varphi(n)} \end{array}\right],$$

gdzie $c'_i = c'_i(y)$ są wielomianami stopnie nie większego niż 1.

krok 3. Dla każdego $i = \varphi(n), \ldots, 1$ wykonuj:

a) Wyznacz

$$b_i = \frac{1}{a'_{ii}} \left(c'_i(y) - \sum_{j=i+1}^{\varphi(n)} a'_{ij} b_j \right) = \frac{r_i y + s_i}{t_i}, \quad \text{where} \quad q | t_i$$

b) Jeśli $(r_i, q) = 1$, to oblicz $y \equiv -s_i r_i^{-1} \pmod{q}$ idź do kroku 4. W przeciwnym przypadku idź do kroku 2.

Krok 4. Zwróć $y \pmod{q}$.

Twierdzenie 5. *Niech $n > 2$. Dla danego $\alpha \in O_K$, macierzy $A(\alpha)$ i liczby pierwszej q takiej, że $q = N_{K/\mathbb{Q}}(\alpha) \equiv 1 \pmod{n}$ oraz $N_{K/\mathbb{Q}}(\alpha) = \det(A(\alpha))$, Procedura* FindRootModQ *jest deterministyczna i znajduje pierwiastek równania $\Phi_n(x) \equiv 0 \pmod{q}$ po wykonaniu $O(\varphi(n)^3 \log^3 q)$ operacji na bitach.*

Dowód: zobacz [H2] (Twierdzenie 2.2).

Procedure FindPrimeP(r, q). Dla danej liczby pierwszej q oraz liczby naturalnej $r < q$, procedura znajduje liczbę pierwszą $p \equiv r \pmod{q}$.

krok 1. Wylosuj $k \in \mathbb{N}$ taką, że $k \in [10, (2^{40}q^2(\log 2^{20}q)^{20}) - r)q^{-1}]$.

krok 2. Oblicz $p = qk + r$. Jeśli p nie jest liczbą pierwszą, to idź do kroku 1.

krok 3. Zwróć p.

Twierdzenie 6. *Niech $q \in [x, 2x]$ będzie wyjściem z procedury* FindPrimeQ *oraz niech $r < q$. Dla dostatecznie dużego $q \geq 2^{32}$ oraz dowolnej liczby rzeczywistej $\lambda \geq 1$, procedura* FindPrimeP *znajduje $k \in \mathbb{N}$ oraz liczbę pierwszą $p = qk + r$ taką, że*

$$k \in [0, (2^{40}q^2(\log 2^{20}q)^{20}) - r)q^{-1}], \quad q \leq p \leq 2^{40}q^2(\log 2^{20}q)^{20}$$

z prawdopodobieństwem większym lub równym $1 - e^{-\lambda}$ po powtórzeniu $[\lambda 8 \log(2^{20}q)]$ kroków procedury z wyjątkiem co najwyżej $O(x(\log x)^{-2})$ liczb pierwszych q. Każdy krok procedury wymaga wykonania co najwyżej $\mathcal{PT}$ operacji na bitach.

Dowód: zobacz [H2] (Twierdzenie 2.3).

Algorithm 2. Dla danego $n > 2$, $n \in \mathbb{N}$, algorytm znajduje liczby pierwsze p oraz q takie, że $q \mid \Phi_n(p)$

krok 1. $a_1, \ldots, a_{\varphi(n)}$, q , $A(\alpha) :=$ FindPrimeQ(n)

krok 2. $y :=$ FindRootModQ$(\alpha, A(\alpha), q)$

krok 3. $p :=$ FindPrimeP(y, q)

krok 4. Return p, q;

Twierdzenie 7. *Algorytm 2 znajduje dwie liczby pierwsze p oraz q takie, że $q \mid \Phi_n(p)$.*

Dowód: zobacz [H2] (Twierdzenie 2.4).

2. Kryptografia na krzywej eliptycznej

Niech E będzie krzywę eliptyczną zdefiniowaną nad ciałem $\mathbb{F}_p$, gdzie p jest liczbą pierwszą. Niech $E(\mathbb{F}_p)$ będzie grupą punktów krzywej E nad $\mathbb{F}_p$ oraz niech $\sharp E(\mathbb{F}_p)$ oznacza rząd grupy $E(\mathbb{F}_p)$. Grupa $E(\mathbb{F}_p)$ jest wykorzystywana do implementacji protokołów kryptograficznych, których bezpieczeństwo zależy od trudności obliczania logarytmu dyskretnego w $E(\mathbb{F}_p)$ [17]. Logarytm dysktetny w grupie $E(\mathbb{F}_p)$ jest trudny do obliczenia, jeżeli grupa ta posiada podrupę dużego rzędu [17]. Z tego powodu, w praktycznym zastosowaniu, należy generować krzywą E nad $\mathbb{F}_p$, której rząd posiada duży dzielnik pierwszy q. W przypadku protokołów kryptograficznych opartych na krzywej eliptycznej Problem Ogólny ma postać.

Problem 3. *Znajdź dwie liczby pierwsze p oraz q takie, że*

$$q \mid \sharp E(\mathbb{F}_p), \qquad \frac{\sharp E(\mathbb{F}_p)}{q} = s, \tag{3}$$

gdzie s jest możliwie małe. Zaproponowana metoda powinna działać w czasie wielomianowym ze względu na liczbę bitów p oraz q.

Generowanie krzywej E nad $\mathbb{F}_p$, odpowiedniej do celów kryptograficznych, może być realizowane według dwóch różnych strategii. W pierwszej z nich wybieramy dużą liczbę pierwszą p i losujemy współczynniki krzywej $a, b \in \mathbb{F}_p$. Następnie, za pomocą algorytmu Schoofa [60], obliczamy $N = \sharp E(\mathbb{F}_p)$. Akceptujemy krzywą $E : y^2 = x^3 + ax + b \pmod p$, jeśli $N = q$ lub $N = qs$, gdzie q jest liczbą pierwszą, a liczba s jest mała. Podczas analizy złożoności obliczeniowej powyższego algorytmu napotykamy na problem związany z istnienie generowanych obiektów. Nie wiadomo czy istnieje nieskończenie wiele liczb pierszych $q = \sharp E(\mathbb{F}_p)$, gdy $p \to \infty$. Jednak, przy założeniu prawdziwości pewnych hipotez z teorii liczb i oszacowań heurystycznych analiza złożoności obliczeniowej powyższej strategii jest możliwa [5], [30], [61]. Iwaniec oraz Urroz udowodnili twierdzenie związane z oszacowaniem liczby generowanych przez powyżej opisany algorytm liczb pierwszych p oraz q [37]. Niech P_2 oznacza liczbę całkowitą bezkwadratową posiadającą co najwyżej dwa dzielniki pierwsze. Autorzy [37] udowodnili, że dla $x \geq 5$ liczba liczb pierwszych $p \leq x$, $p \equiv 1 \pmod 4$ takich, że $\frac{1}{8}|E(\mathbb{F}_p)| = P_2$ jest co najmniej równa $cx/(\log x)^2$, gdzie E jest krzywą eliptyczną nad $\mathbb{Z}$ postaci $y^2 = x^3 - x$, gdzie $c > 0$ jest pewną stałą.

Drugim podejściem do generowania krzywej E nad ciałem $\mathbb{F}_p$ jest metoda mnożenia zespolonego (CM metoda) [25], [64]. Wprowadźmy następującą definicję.

Definicja 1. *Niech $\Delta < 0$ będzie ustaloną liczbą bezkwadratową. Liczby pierwsze p oraz q są* CM-pierwsze *względem Δ jeśli istnieją liczby całkowite f oraz t takie, że*

$$|t| \leq 2\sqrt{p}, \quad q \mid p + 1 - t, \quad 4p - t^2 = \Delta f^2. \tag{4}$$

Dla danych liczb CM-pierwszych q oraz p równanie krzywej E nad $\mathbb{F}_p$, gdzie $q \mid \sharp E(\mathbb{F}_p)$ można otrzymać metodą mnożenia zespolonego po wykonaniu $O(|\Delta|^{1+\epsilon})$ operacji arytmetycznych [25]. W praktyce jest możliwe wygenerowanie krzywych dla których $|\Delta| \leq 10^{12}$ [64]. W literaturze istnieje wiele różnych wariantów znajowania liczb CM-pierwszych [14], [15], [40], [42], [58]. Jednak, analiza złożoności obliczeniowej algorytmów w cytowanych pracach zależy od pewnych heurystycznych hipotez lub nie jest podana.

W pracy [H4] zaproponowano algorytm, który konstruje liczby CM-pierwsze. Autor udowodnił w niej, że wprowadzona metoda generowania liczb postaci (4) działa w czasie wielomianowym ze względu na liczbę bitów p oraz q. Idea proponowanej metody jest następująca [H4]. Niech $\Delta < 0$ będzie bezkwadratową liczbą całkowitą. Ustalmy ciało $K = \mathbb{Q}(\sqrt{\Delta})$ wraz z pierścieniem liczb całkowitych $\mathcal{O}_K = \{a + b\omega : a, b \in \mathbb{Z}\}$, gdzie $\omega = \frac{1+\sqrt{\Delta}}{2}$, gdy $\Delta \equiv 1 \pmod 4$ oraz $\omega = \sqrt{\Delta}$, gdy $\Delta \equiv 2, 3 \pmod 4$. W pierwszym kroku algorytm znajduje $\alpha \in \mathcal{O}_K$ takie, że $N_{K/\mathbb{Q}}(\alpha) = q$ jest liczbą pierwszą. Następnie, dla danego α, algorytm znajduje liczbę $\beta \in \mathcal{O}_K$ w postępie arytmetycznym $\beta \equiv 1 \pmod{(\alpha)}$ taką, że $N_{K/\mathbb{Q}}(\beta) = p$ jest liczbą pierwszą. Autor tego autoreferatu wykazał, że w ten sposób wygenerowane liczby pierwsze są CM-pierwsze i spełniają (4). Ponadto, autor oszacował rząd wielkości generowanych liczb pierwszych p oraz q. Wykazał on, że $x < q < 2x$ oraz $q < p < (2x)^{5/(2-5\varepsilon)}$, gdzie $0 < \varepsilon < \frac{2}{5}$ dla dostecznie dużego x i dla prawie wszytkich q wygenerownych w pierwszym kroku algorytmu. Procedura znajdowania powyższej liczby pierwszej q jest podobna do tej przedstawionej w [H1], [H2]. Jednak druga faza metody obliczeniowej wprowadzonej w [H4] jest różna od tych wcześniej stosowanych i dzięki niej

udało się autorowi uzyskać rezultat, w którym $(\log p)/(\log q) \ll 5/(2-5\varepsilon)$ dla dostatecznie dużych p oraz q oraz $0 < \varepsilon < \frac{2}{5}$. Algorytm autora składa się z dwóch głównych procedur i wraz z metodą mnożenia zespolonego realizuje Problem 3 [H4].

Procedure FindPrimeQ$(n, m, \Delta, x, \gamma)$. Niech będą dane $n, m \in \mathbb{N}$, $(m, n) = 1$, bezkwadratowa liczba $\Delta \in \mathbb{Z}$, $\Delta < 0$ oraz dostatecznie duże $x \in \mathbb{R}$. Ustalmy $K = \mathbb{Q}(\sqrt{\Delta})$ oraz pierścień liczb całkowitych tego ciała $\mathcal{O}_K = \{a + b\omega : a, b \in \mathbb{Z}\}$, gdzie $\omega = \frac{1+\sqrt{\Delta}}{2}$, gdy $\Delta \equiv 1 \pmod 4$ oraz $\omega = \sqrt{\Delta}$, gdy $\Delta \equiv 2, 3 \pmod 4$. Niech $\gamma = f + g\omega \in \mathcal{O}_K$ będzie takie, że $|f|, |g| \le n$, $N_{K/\mathbb{Q}}(\gamma) \equiv m \pmod n$. Procedura znajduje $\alpha = a + b\omega \in \mathcal{O}_K$ takie, że $q = N_{K/\mathbb{Q}}(\alpha) \equiv m \pmod n$ jest liczbą pierwszą, $x \le q \le 2x$.

krok 1. Wylosuj $u, v \in \mathbb{Z}$ takie, że

$$\begin{array}{lll} |u| \le (\frac{\sqrt{1-\Delta}}{\sqrt{-\Delta}}(2x)^{1/2} - f)n^{-1}, & |v| \le (\frac{2}{\sqrt{-\Delta}}(2x)^{1/2} - g)n^{-1} & \text{jeśli} \quad \Delta \equiv 1 \pmod 4, \\ |u| \le ((2x)^{1/2} - f)n^{-1}, & |v| \le (\frac{1}{\sqrt{-\Delta}}(2x)^{1/2} - g)n^{-1} & \text{jeśli} \quad \Delta \equiv 2, 3 \pmod 4. \end{array}$$

krok 2. Oblicz $a = nu + f$ oraz $b = nv + g$

krok 3. Oblicz

$$\begin{array}{ll} q = a^2 + ab + \frac{1-\Delta}{4}b^2 & \text{jeśli} \quad \Delta \equiv 1 \pmod 4, \\ q = a^2 - \Delta b^2 & \text{jeśli} \quad \Delta \equiv 2, 3 \pmod 4. \end{array}$$

Jeśli $q < x$ lub $q > 2x$, to idź do kroku 1.

krok 4. Jeśli q jest liczbą pierwszą, to zakończ procedurę. W przeciwnym przypadku idź do kroku 1.

krok 5. Zwróć $\alpha = a + b\omega$, q.

Twierdzenie 8. *Niech dane będą $n, m \in \mathbb{N}$, $(m, n) = 1$ oraz bezkwadratowa liczba całkowita $\Delta < 0$. Ustalmy $K = \mathbb{Q}(\sqrt{\Delta})$ wraz pierścieniem liczb całkowitych $\mathcal{O}_K$. Niech $\mathfrak{f} = n\mathcal{O}_K$. Istnieje $x_0 > 0$ takie, że dla każdego $x \ge x_0$ oraz dowolnego $\lambda \ge 1$ procedura* FindPrimeQ *znajduje $\alpha = a + b\omega \in \mathcal{O}_K$ takie, że $N_{K/\mathbb{Q}}(\alpha) \equiv m \pmod n$ jest liczbą pierwszą oraz $x \le N_{K/\mathbb{Q}}(\alpha) \le 2x$, gdzie*

$$\begin{array}{lll} |a| \le \frac{\sqrt{1-\Delta}}{\sqrt{-\Delta}}(2x)^{\frac{1}{2}}, & |b| \le \frac{2}{\sqrt{-\Delta}}(2x)^{\frac{1}{2}} & \textit{if} \quad \Delta \equiv 1 \pmod 4, \\ |a| \le (2x)^{\frac{1}{2}}, & |b| \le \frac{1}{\sqrt{-\Delta}}(2x)^{\frac{1}{2}} & \textit{if} \quad \Delta \equiv 2, 3 \pmod 4. \end{array}$$

z prawdopodobieństwem większym lub równym $1 - e^{-\lambda}$ po powtórzeniu $[c_1\lambda(\log x)]$ kroków procedury, gdzie $c_1 = \frac{16\sqrt{1-\Delta}h^*_{\mathfrak{f}}(K)}{-\Delta n^2}$, dla $\Delta \equiv 1 \pmod 4$, oraz $c_1 = \frac{16h^*_{\mathfrak{f}}(K)}{\sqrt{-\Delta}n^2}$, dla $\Delta \equiv 2, 3 \pmod 4$. Każdy krok procedury wymaga wykonania co najwyżej $\mathcal{PT}$ operacji na bitach.

Dowód: zobacz [H4] (Twierdzenie 2.1).

Procedure FindPrimeP(α, q, Δ, x). Ustalmy $0 < \varepsilon < 2/5$ oraz ciało $K = \mathbb{Q}(\sqrt{\Delta})$ wraz pierścieniem liczb całkowitych $\mathcal{O}_K$. Niech dane będzie $\alpha = a + b\omega \in \mathcal{O}_K$ takie, że $q = N_{K/\mathbb{Q}}(\alpha) \equiv m \pmod n$, gdzie $(m, n) = 1$, $x \le q \le 2x$, jest liczbą pierwszą. Procedura znajduje $\beta \in \mathcal{O}_K$ takie, że $\beta \equiv 1 \pmod{\alpha\mathcal{O}_K}$ oraz $N_{K/\mathbb{Q}}(\beta)$ jest liczbą pierwszą.

krok 1. Wylosuj $s, t \in \mathbb{Z}$.

Jeśli $\Delta \equiv 1 \pmod 4$,

$$|s| \le \frac{\sqrt{1-\Delta}}{\sqrt{-\Delta}}(2x)^{(3+5\varepsilon)/(4-10\varepsilon)}, \quad |t| \le \frac{2}{\sqrt{-\Delta}}(2x)^{(3+5\varepsilon)/(4-10\varepsilon)}.$$

Jeśli $\Delta \equiv 2,3 \pmod 4$

$$|s| \leq (2x)^{(3+5\varepsilon)/(4-10\varepsilon)}, \quad |t| \leq \frac{1}{\sqrt{-\Delta}}(2x)^{(3+5\varepsilon)/(4-10\varepsilon)}$$

krok 2. Oblicz

$$\begin{aligned} c &= as - \frac{1-\Delta}{4}bt + 1, \quad d = bs + (a+b)t \quad \text{jeśli} \quad \Delta \equiv 1 \pmod 4, \\ c &= as + \Delta bt + 1, \quad d = bs + at \quad \text{jeśli} \quad \Delta \equiv 2,3 \pmod 4. \end{aligned}$$

krok 3. Oblicz

$$\begin{aligned} &p = c^2 + cd + \tfrac{1-\Delta}{4}d^2 \qquad \text{if} \quad \Delta \equiv 1 \pmod 4, \\ &p = c^2 - \Delta d^2 \qquad\qquad\quad \text{if} \quad \Delta \equiv 2,3 \pmod 4. \end{aligned}$$

Jeśli $p < x$ or $p > (2x)^{5/(2-5\varepsilon)}$, to idź do kroku 1.

krok 4. Jeśli p jest liczbą pierwszą, to zakończ procedurę. W przeciwnym przypadku idź do kroku 1.

krok 5. Zwróć $\beta = c + d\omega$, p.

Twierdzenie 9. *Niech $\Delta < 0$ będzie bezkwadratową liczbą całkowitą. Ustalmy $K = \mathbb{Q}(\sqrt{\Delta})$ wraz pierścieniem liczb całkowitych $\mathcal{O}_K$ oraz $0 < \varepsilon < \frac{2}{5}$. Niech $\alpha \in \mathcal{O}_K$ oraz $x \leq q \leq 2x$ będzie wyjściem procedury* FindPrimeQ. *Procedura* FindPrimeP, *do której wejścią są α, q oraz Δ, ma następujące własności: istnieje $x_0 > 0$ taka, że dla każdego $x \geq x_0$, dowolnego $\lambda \geq 1$ oraz dowolnej stałej $A > 2$, procedura znajduje $\beta \in \mathcal{O}_K$ takie, że,*

$$\beta = c + d\omega, \quad p = N_{K/\mathbb{Q}}(\beta) \text{ jest liczbą pierwszą}, \quad x \leq N_{K/\mathbb{Q}}(\beta) \leq (2x)^{5/(2-5\varepsilon)},$$

z prawdopodobieństwem większym lub równym $1 - e^{-\lambda}$ po powtórzeniu $[c_2\lambda(\log 2x)]$ kroków procedury, gdzie $c_2 = \frac{80h(K)\sqrt{1-\Delta}}{-(2-5\varepsilon)w(K)\Delta}$, gdy $\Delta \equiv 1 \pmod 4$ oraz $c_2 = \frac{40h(K)}{(2-5\varepsilon)w(K)\sqrt{-\Delta}}$ dla $\Delta \equiv 2,3 \pmod 4$, dla prawie wszystkich α z wyjątkiem co najwyżej $O(x(\log x)^{-A})$ wartości α. Każdy krok procedury wymaga wykonania co najwyżej $\mathcal{PT}$ operacji na bitach.

Dowód: zobacz [H4] (Twierdzenie 2.2).

Algorytm 3

krok 1. $\alpha, q :=$ FindPrimeQ(n, Δ, x, γ).

krok 2. $\beta, p :=$ FindPrimeP(α, q, Δ, x).

krok 3. Zwróć p, q, α, β.

Twierdzenie 10. *Niech będą dane $n, m \in \mathbb{N}$, $(m, n) = 1$ oraz bezkwadratowa liczba całkowita $\Delta < 0$. Ustalmy $K = \mathbb{Q}(\sqrt{\Delta})$ wraz z pierścieniem liczb całkowitych $\mathcal{O}_K$. Jeśli Algorytm 3 się wykona, to zwraca on parę liczb $\alpha, \beta \in \mathcal{O}_K$, $\beta = c + d\omega$ takich, że $q = N_{K/\mathbb{Q}}(\alpha) \equiv m \pmod n$, $N_{K/\mathbb{Q}}(\beta) = p$ są CM-pierwsze.*

Dowód: zobacz [H4] (Twierdzenie 2.3).

Uwaga. Niech dane będą liczby CM-pierwsze $q = N_{K/\mathbb{Q}}(\alpha)$, $p = N_{K/\mathbb{Q}}(\beta)$, gdzie $\alpha = a + b\omega$, $\beta = c + d\omega \in \mathcal{O}_K$. Istnieje krzywa eliptyczna E nad $\mathbb{F}_p$ z mnożeniem zespolonym przez ordynek $\mathcal{O}_d = [1, d\omega] \subseteq K$ taka, że q dzieli

$$\begin{aligned} &|E(\mathbb{F}_p)| = p + 1 - 2c - d \qquad \text{jeśli} \quad \Delta \equiv 1 \pmod 4, \\ &|E(\mathbb{F}_p)| = p + 1 - 2c \qquad\quad \text{jeśli} \quad \Delta \equiv 2,3 \pmod 4. \end{aligned}$$

Ustalmy $c > 0$ oraz niech $\Delta = O((\log p)^c)$. Dla dowolnego $\epsilon > 0$ konstrukcja krzywej E nad $\mathbb{F}_p$ metodą mnożenia zespolonego wymaga wykonania nie więcej niż $O((\log p)^{c(1+\epsilon)/2})$ operacji arytmetycznych $\mathbb{F}_p$.

3. Efektywne oszacowanie czasu działania algorytmów.

Analiza złożoności czasowej wielu algorytmów stosowanych w obliczeniowej teorii liczb, w tym w kryptologii, często wykorzystuje twierdzenia dotyczące istoty obiektów teorioliczbowych: ich istnienia, rozmieszczenia oraz gęstości. Tego typu rezultaty zazwyczaj zachodzą dla dostatecznie dużych parametrów oraz postulują istnienie stałych bez podania ich numerycznej wartości, bo nie jest ona istotna z punktu widzenia przeprowadznego dowodu twierdzenia. Jednak w obliczeniowej teorii liczb wartości numeryczne stałych mają znaczenie w praktycznym zastosowaniu. Na przykład, wyznaczenie dokładnych wartości stałej Hermite'a γ_n jest istotna z punktu widzenia bezpieczeństwa kryptosystemów implementowanych na kratach i zrozumienia istoty wielu algorytmów znajdujących wektory, których długość jest bliska najkrótszemu wektorowi w zadanej kracie [50]. Do dziś znane są jedynie stałe γ_n dla $n < 9$, $n = 24$.

Generując liczby pierwsze, rozmaitych postaci, parametry klucza do systemów kryptograficznych, istotna jest wiedza o tym, jaki powinny posiadać one rząd wielkości, aby mieć pewność, że wybierane są one ze zbioru, w którym ich gęstość jest wystarczająca dla bezpieczeństwa kryptosystemu i efektywności algorytmu. Numeryczne testy pozwalają nam na wyznaczanie gęstości parametrów kluczy tylko w niewielkim zakresie (10^{12}) ograniczonym przez czas i wydajność komputerów. Z tego powodu nie dotyczą one wymaganych dla bezpieczeństwa krytosystemów rzędów wielkości parametrów. Dowodząc twierdzeń, w których stałe wyznaczone są expicite i na ich podstawie dokonując analizy algorytmów, dostajemy wiedzę o tym w jakich zakresach projektowane metody będą zawsze działać zgodnie z tezą udowodnionego twierdzenia. Chęć uzyskania takiej wiedzy związana jest z kolejną pracą autora tego autoreferatu [H6].

W dostępnej literaturze odnajdujemy wiele prac, w których stałe wyznaczone są explicite [2], [21], [22], [23], [34], [41], [46], [70]. W pracy [41] przedstawiono efektywną wersję twierdzenia Czebotariewa, która wykorzystywana jest do analizy wielu algorytmów w teorii liczb. Jednak twierdzenie to dotyczy szerszej klasy obiektów teorii liczb, niż tych które występują w algorytmach autora tego autoreferatu. Ponadto, w pracy [41] nie wszystkie wartości numeryczne stałych są wyznaczone, a teza głównego twiedzenia mówi jedynie o ich istnieniu i możliwości efektywnego obliczenia stałych. Z drugiej strony istnieją efektywne twierdzenia o liczbach pierwszych [21], [22], [23] lub o liczbach pierwszych w postępie arytmetycznym [46], w których wartości numeryczne stałych są podane. Jednak nie nadają się one do analizy algorytmów zaproponowanych w pracach [H1], [H2], [H4], w których znajduje się liczby pierwsze bedące normami liczb algebraicznych całkowitych ustalonego urojonego ciała liczbowego. W pracy [H6] autor uzyskuje efektywne twierdzenie, które można wykorzystać do analizy złożoności procedury FindPrimeQ z prac [H1], [H2], [H4], [H3], [H5]. Metody dowodu wykorzystywane w pracach [21], [22], [23], [41], [46] nie pozwalają na optymalne oszacowanie stałych numerycznych występujących w otrzymanych twierdzeniach. Dlatego, w pracy [H6] autor nie powtarza klasycznego dowodu twierdzenia lecz obiera ścieżkę dowodową w ten sposób, aby uzyskać jak najlepsze oszacowanie stałych występujących w twierdzeniu.

Niech K będzie ustalonym, całkowicie urojonym, algebraicznym ciałem liczbowym o wyróżniku $\Delta = \Delta(K)$ oraz stopniu $[K : \mathbb{Q}] = 2r_2$, gdzie $2r_2$ jest liczbą sprzężonych ciał urojonych z K. Niech $\mathcal{O}_K$ będzie pierścieniem liczb całkowitych ciała K, a $\mathfrak{f}$ ustalonym, niezerowym

ideałem całkowitym tego pierścienia. Oznaczmy przez $H = H \pmod{\mathfrak{f}}$ grupę klas ideałów modulo $\mathfrak{f}$ rozpatrywaną w wąskim sensie. Niech $h^*_{\mathfrak{f}}(K)$ będzie liczbą elementów H. Dla ustalonej klasy $X \in H$ definiujemy funkcję,

$$\Psi(x, X) = \sum_{\substack{x \leq N\mathfrak{p}^m \leq 2x \\ \mathfrak{p}^m \in X}} \log N\mathfrak{p},$$

gdzie $\mathfrak{p}$ przebiega we wszystkie ideały pierścienia $\mathcal{O}_K$, a $N\mathfrak{p}$ oznacza normę ideału $\mathfrak{p}$. Niech χ_0 oznacza charakter główny modulo $\mathfrak{f}$. Ponadto, niech

$$E_0 = E_0(\chi) = \begin{cases} 1 & \text{dla} \quad \chi = \chi_0 \\ 0 & \text{dla} \quad \chi \neq \chi_0 \end{cases}$$

W pracy [H6] autor uzyskał poniższy rezultat.

Twierdzenie 11. *Niech K, Δ, $\mathfrak{f}$, $\zeta(s, \chi)$ oznaczają odpowiednio ciało liczb algebraicznych stopnia $[K : \mathbb{Q}] = 2r_2$, wyróżnik ciała K, dowolny ideał całkowity K oraz funkcję dzeta Heckego-Landaua z charaktemem χ modulo $\mathfrak{f}$. Ustalmy $0 < \varepsilon < 1$. Jeśli $|\Delta| \geq 9$ oraz obszar*

$$\sigma \geq 1 - 0.0795 \left(\log |\Delta| + 0.7761 \log \left((|t| + 1)^{2r_2} \, (N\mathfrak{f})^{1 - E_0} \right) \right)^{-1}, \tag{5}$$

jest wolny od zer funkcji $\zeta(s, \chi)$, to

$$\Psi(x, X) \geq \frac{x(1 - \varepsilon)}{h^*_{\mathfrak{f}}(K)},$$

dla

$$\log x \geq \left(23.148 \sqrt{r_2} \left(1 + \left(2 \log \left(\frac{c_1 \sqrt{r_2}}{0.117 \varepsilon} \right) \right)^{\frac{1}{2}} + \frac{2}{3} \log \left(\frac{c_1 \sqrt{r_2}}{0.117 \varepsilon} \right) \right) \right)^2,$$

gdzie

$$c_1 = \left(40506.547 |\Delta|^{\frac{1.933}{r_2}} + 15061.779 |\Delta|^{\frac{1.289}{r_2}} (N\mathfrak{f})^{\frac{1}{r_2}} h^*_{\mathfrak{f}}(K) \right) r_2^2 \log(|\Delta| N\mathfrak{f}).$$

Dowód: zobacz [H6] (Twierdzenie 1).

Uwaga. Dla rzeczywistego charakteru $\chi \pmod{\mathfrak{f}}$ funkcja $\zeta(s, \chi)$ może mieć pojedyncze zero rzeczywiste w obszarze (5). Takie zero, o ile istnieje, nazywamy zerem Siegela. Możemy sprawdzić numerycznie, czy funkcja ustalona $\zeta(s, \chi)$ posiada zero Siegela w obszarze (5) za pomocą skryptu dołączonego do pracy [54].

4. Kryptosystemy oparte na odwzorowaniach dwuliniowych

Iloczyny dwuliniowe Weila oraz Tate'a na krzywych eliptycznych lub na jakobianach krzywych hipereliptycznych rodzaju 2 miały wpływ na powstanie wielu nowych protokołów kryptograficznych. Wykorzystując odwzorowania dwuliniowe zaprojektowano między innymi protokół, który tylko w jednej rundzie uzgadnia sekret między trzema stronami [38].

Iloczyny dwuliniowe wykorzystuje się w protokołach generowania krótkich podpisów cyfrowych [8], oraz przyczyniły się one do ulepszena protokołów szyfrowania opartych na tożsamości [10]. Na punktach q-torsyjnych krzywej eliptycznej lub jakobianu krzywej hipereliptycznej rodzaju g nad $\mathbb{F}_p$, odwzorowania dwuliniowe przyjmują wartości w pewnym roszerzeniu skończonym $\mathbb{F}_{p^n}$ ciała $\mathbb{F}_p$. Liczbę n nazywamy stopniem zanurzenia krzywej względem q. Aby iloczyny dwuliniowe mogły być efektywnie obliczane w praktyce stopień zanurzenia n względem q nie może być duży [17]. Z drugiej strony iloczyny dwuliniowe przenoszą problem logarytmu dyskretnego z krzywej (lub jakobianu krzywej hipereliptycznej rodzaju g) nad $\mathbb{F}_p$ do ciała $\mathbb{F}_{p^n}$. Dlatego, dla małego n oraz niewielkiego p, problem obliczania logarytmu dyskretnego na krzywej (jakobianie) może być atakowany metodami dedykowanymi dla ciał skończonych [17], [33], [53]. Z tego powodu w praktyce implementacja protokołów opartych na iloczynach dwuliniowych wymaga generowania specjalnych krzywych rodzaju $g = 1, 2$ o małym stopniu zanurzenia n względem q, których rząd (jakobian) nad $\mathbb{F}_p$ posiada duży dzielnik pierwszy q, a liczba p jest odpowiednio duża. Dodajmy, że istnieje niewielkie prawdopodobieństwo tego, że losowo generowana krzywa nad $\mathbb{F}_p$ ma mały stopień zanurzenia [5]. W celu omówienia metod konstrukcji powyższych krzywych rozważymy dwa przypadki.

(A) Krzywe eliptyczne.

Definicja 2. *Niech n będzie ustaloną liczbą naturalną oraz niech $\Delta < 0$ będzie liczbą bezkwadratową. Liczby pierwsze p oraz q są typu PF (pairing-friendly) względem n i Δ jeśli istnieją liczby całkowite f oraz t takie, że*

$$|t| \leq 2\sqrt{p}, \quad q \mid p + 1 - t, \quad q \mid \Phi_n(p), \quad 4p - t^2 = \Delta f^2. \tag{6}$$

Dla danych liczb pierwszych p oraz q, które są typu PF względem n i Δ, konstrukcja krzywej elipcznej $E(\mathbb{F}_p)$ odbywa się za pomocą metody mnożenia zespolonego [25], [64]. Istnieją dwa główne podejścia znajdowania liczb pierszych p oraz q typu PF. Pierwsza z nich związana jest z obserwacją dotyczącą faktoryzacji pewnych wielomianów specjalnej postaci nad $\mathbb{Z}[x]$ [6], [13], [20], [28], [49]. W metodzie tej znajduje się liczby pierwsze, które są wartościami ustalonych, nierozkładalnych nad $\mathbb{Z}[x]$ wielomianów stopnia większego lub równego 2. Z tego powodu analiza złożoności obliczeniowej powyższych metod jest możliwa przy założeniu prawdziwości pewnych hipotez teorii liczb [7], [59]. Drugim sposobem generowania liczb pierwszych p oraz q, które są typu PF względem n i Δ, jest konstrukcja zaproponowana przez Cocksa-Pincha. Metoda ta nie została opublikowana, jednak znana jest wśród kryptologów [28]. Algorytm Cocksa-Pincha wykonuje następujące kroki. Dla danego n, znajduje on liczbę pierwszą $q \equiv 1 \pmod{n}$ oraz bezkwadratową liczbę $\Delta < 0$ taką, że Δ jest kwadratem modulo q. Następnie oblicza on pierwiastek pierwotny w_n stopnia n z jedynki modulo q oraz wyznacza $t' \equiv w_n + 1 \pmod{q}$ oraz $f' \equiv (t' - 2)(\sqrt{\Delta})^{-1} \pmod{q}$. W ostatniej fazie, algorytm znajduje $t \equiv t' \pmod{q}$ oraz $f \equiv f' \pmod{q}$ takie, że $p = \frac{1}{4}(t^2 - \Delta f^2)$ jest liczbą pierwszą. Analiza złożoności obliczeniowej tej metody nie jest znana.

W pracy [H3] autor proponuje algorytm znajdujący, dla danego n oraz bezkwadrotowej liczby $\Delta < 0$, liczby p oraz q, które są typu PF względem n i Δ. Zaproponowana metoda jest uogólnieniem metody Cocksa-Pincha. Niech $K = \mathbb{Q}(\sqrt{\Delta})$ oraz niech $H(K)$ oznacza grupę klas ideałów ciała K. Niech $h(K)$ będzie liczbą elementów $H(K)$. Analiza algorytmu proponowanego w [H3] pokazuje, że dla $\Delta \equiv 2, 3 \pmod{4}$ oraz $h(K) = 1$ metoda Cocksa-Pincha jest szczególnym przypadkiem algorytmu autora tego autoreferatu. Autor w [H3]

rozważa również przypadek $\Delta \equiv 1 \pmod 4$, w którym $q = a^2 + ab + \frac{1-\Delta}{4}b^2$, $p = t^2 + tf + \frac{1-\Delta}{4}f^2$, gdzie $a, b, t, f \in \mathbb{Z}$. Przypadek ten nie był wcześniej badany.

W pracy [9] metoda Cocksa oraz Pincha została zaadoptowana do generowania krzywych eliptycznych nad $\mathbb{F}_p$, których rząd jest podzielny przez iloczyn dużych liczb pierwszych. W tym przypadku należy generować parametry spełniające poniższe warunki.

Definicja 3. *Niech n będzie ustaloną liczbą naturalną oraz niech $\Delta < 0$ będzie liczbą bezkwadratową. Liczba pierwsza p oraz liczba naturalna N są typu PF względem n i Δ jeśli istnieją liczby całkowite f oraz t takie, że*

$$|t| \leq 2\sqrt{p}, \quad q \mid p + 1 - t, \quad N \mid \Phi_n(p), \quad 4p - t^2 = \Delta f^2.$$

W pracy [H3] autor proponuje konstrukcję algorytmiczną zwracającą parę liczb p, N typu PF, gdzie N jest liczbą złożoną. Metoda ta może być alternatywą do metod generowania parametrów opisanych w [9]. Algorytm autora tego autoreferatu znajdujący liczby pierwsze spełniających (6) składa się dwóch poniższych procedur.

Procedura FindPrimeQ(n, Δ, γ). Niech będą dane $n \in \mathbb{N}$ oraz bezkwadratowa liczba $\Delta \in \mathbb{Z}$, $\Delta < 0$. Niech $K = \mathbb{Q}(\sqrt{\Delta})$ oraz niech $\mathcal{O}_K$ oznacza pierścień liczb całkowitych K. Niech dane będzie $\gamma = f + g\omega \in \mathcal{O}_K$ takie, że $|f|, |g| \leq n$, $N_{K/\mathbb{Q}}(\gamma) \equiv 1 \pmod n$. Procedura znajduje $\alpha = a + b\omega \in \mathcal{O}_K$ takie, że $N_{K/\mathbb{Q}}(\alpha) \equiv 1 \pmod n$ oraz $N_{K/\mathbb{Q}}(\alpha) = q$ jest liczbą pierwszą.

krok 1. Wylosuj $u, v \in \mathbb{Z}$.

krok 2. Oblicz $a = nu + f$ oraz $b = nv + g$.

krok 3. Oblicz $q = N_{K/\mathbb{Q}}(a + b\omega)$.

krok 4. Jeśli q jest liczbą pierwszą, to zakończ procedurę. W przeciwnym przypadku idź do kroku 1.

krok 5. Zwróć $\alpha = a + b\omega$, q.

Procedura FindPrimeP(α, q, Δ). Niech $K = \mathbb{Q}(\sqrt{\Delta})$ oraz niech $\mathcal{O}_K$ będzie pierścieniem liczb całkowitych K. Niech dane będzie $\alpha = a + b\omega \in \mathcal{O}_K$ takie, że $q = N_{K/\mathbb{Q}}(\alpha) \equiv 1 \pmod n$ jest liczbą pierwszą. Procedura znajduje $\beta \in \mathcal{O}_K$ takie, że $N_{K/\mathbb{Q}}(\beta) \equiv w_n \pmod q$, gdzie w_n jest pierwiastkiem pierwotnym n-tego stopnia z jedności modulo q oraz $N_{K/\mathbb{Q}}(\beta)$ jest liczbą pierwszą.

krok 1. Oblicz w_n pierwiastek n-tego stopnia z jedności modulo q.

krok 2. Oblicz $r \equiv a(-b)^{-1} \pmod q$.

krok 3. Oblicz k oraz l modulo q.

Jeśli $\Delta \equiv 1 \pmod 4$,

$$k \equiv (1 - (1 + \omega_n)r)(1 - 2r)^{-1} \pmod q, \qquad l \equiv (\omega_n - 1)(1 - 2r)^{-1} \pmod q.$$

Jeśli $\Delta \equiv 2, 3 \pmod 4$

$$k \equiv (1 - \omega_n)2^{-1} \pmod q, \qquad l \equiv (1 + \omega_n)(2r)^{-1} \pmod q.$$

krok 4. Wylosuj $s, t \in \mathbb{Z}$.

krok 5. Oblicz $c = qs + k$ oraz $d = qt + l$.

krok 6. Oblicz $p = N_{K/\mathbb{Q}}(c + d\omega)$. Jeśli p jest liczbą pierwszą, to zakończ procedurę. W przeciwnym przypadku idź do kroku 3.

krok 7. Zwróć $\beta = c + d\omega$, p.

Algorytm 4

krok 1. $\alpha, q := \text{FindPrimeQ}(n, \Delta, \gamma)$.

krok 2. $\beta, p := \text{FindPrimeP}(\alpha, q, \Delta)$.

krok 3. Zwróć p, q, α, β.

Twierdzenie 12. *Niech $n \in \mathbb{N}$ oraz bezkwadratowa liczba całkowita $\Delta < 0$ będą dane. Niech $K = \mathbb{Q}(\sqrt{\Delta})$ oraz niech $\mathcal{O}_K$ będzie pierścieniem liczb całkowitych ciała K. Algorytm 4 zwraca $\alpha, \beta \in \mathcal{O}_K$, $\beta = c + d\omega$ takie, że $N_{K/\mathbb{Q}}(\alpha) = q$, $N_{K/\mathbb{Q}}(\beta) = p$ są liczbami pierwszymi, które są typu PF względem n oraz Δ.*

Dowód: zobacz [H3] (Twierdzenie 1).

Uwaga. W pracy [H4] autor udowodnił, że powyższa procedura $\text{FindPrimeQ}(n, \Delta, \gamma)$ działa w czasie wielomianowym ze względu na liczbę bitów p oraz q.

(B) Krzywe hipereliptyczne rodzaju $g \geq 2$.

Niech K będzie CM-ciałem stopnia $[K : \mathbb{Q}] = 2g$. Oznaczmy przez $\mathcal{O}_K$ pierścień liczb algebraicznych całkowitych ciała K. Mówimy, że π jest p-liczbą Weila jeśli $\pi \in \mathcal{O}_K$ oraz dla każdego urojonego włożenia $\sigma : K \to \mathbb{C}$ mamy $|\sigma(\pi)| = \sqrt{p}$.

Definicja 4. *Niech n będzie ustaloną liczbą naturalną. Liczby pierwsze q oraz p są typu PF względem n jeśli istnieje liczba algebraiczna całkowita $\pi \in \mathcal{O}_K$ taka, że π jest p-liczbą Weila, oraz*

$$p = \pi\overline{\pi}, \quad q \mid \Phi_n(\pi\overline{\pi}), \quad q \mid N_{K/\mathbb{Q}}(\pi - 1). \tag{7}$$

Załóżmy, że dane są dwie liczby pierwsze q oraz p, które są typu PF względem n, oraz p-liczba Weila $\pi \in \mathcal{O}_K$. Wtedy istnieje prosta i zwyczajna rozmaitość abelowa $A/\mathbb{F}_p$ odpowiadająca p-liczbie Weila π, która ma stopień zanurzenia n względem q [29]. Rozmaitość abelową $A/\mathbb{F}_p$ można skonstruować za pomocą metody mnożenia zespolonego, gdy $[K : \mathbb{Q}] = 4$ [24], [48], [63].

Istnieje wiele metod znajdowania liczb pierwszych, które są typu PF względem n [18], [19], [26], [27], [29]. Niestety we wszystkich zacytowanych pracach autorzy nie szacują złożoności obliczenowej swoich algorytmów, bądź dokonują analizy algorytmów przy heurystycznych założeniach.

W pracy [29] algorytm działa według następującej zasady. Niech K będzie ustalonym CM-ciałem stopnia $[K : \mathbb{Q}] = 2g \geq 4$ oraz niech K' będzie reflex ciałem K względem ustalonego CM-typu Φ, gdzie $[K' : \mathbb{Q}] = 2g'$. Niech $n \in \mathbb{N}$ będzie wejściem do algorytmu. W pierwszej fazie algorytm znajduje liczbę pierwszą $q \equiv 1 \pmod{n}$, która rozpada się całkowicie w K. Następnie, dla danego $q \equiv 1 \pmod{n}$ znajduje on $\xi \in \mathcal{O}_{K'}$ takie, że $N_{K'/\mathbb{Q}}(\xi) = p$ jest liczbą pierwszą. W drugiej fazie algorytm generuje p-liczbę Weila $\pi \in \mathcal{O}_K$. W tym celu dokonuje on faktoryzacji ideału (q) w ciele $\mathcal{O}_{K'}$. Można tego dokonać za pomocą algorytmu faktoryzacji w $\mathbb{F}_q[X]$, por. Twierdzenie 4.8.13, str. 199 w pracy [16].

W pracy [H5] zaprezentowano algorytmiczną metodę znajdowania liczb pierwszych q oraz p, które są typu PF względem n. Wprowadzona metoda łączy ze sobą oraz rozwija idee stosowane w pracach [29], [H2], [H3]. Algorytm autora tego autoreferatu różni się w istotnych fragmentach od sposobu generowania parametrów prezentowanego w [29]. W pracy [H5], w pierwszej fazie swojego działania, algorytm losuje $\alpha \in \mathcal{O}'_K$ takie, że $q = N_{K'/\mathbb{Q}}(\alpha)$

jest liczbą pierwszą . Dzięki takiemu krokowi nie trzeba wykonywać w dalszej części algorytmu faktoryzacji ideału (q) w ciele $\mathcal{O}_{K'}$, jak to było w [29]. W algorytmie autora tego autoreferatu wykorzystuje się elementarne metody algebry liniowej, co znacznie ułatwia jego implementację. Ponadto, w pracy [H5] wprowadzono deterministyczną metodę obliczania pierwiatków pewnych wielomianów nad $\mathbb{F}_q$, gdy dane są q, α takie, że $q = N_{K'/\mathbb{Q}}(\alpha)$. Metoda ta jest uogólnieniem algorytmicznego sposobu wyznaczania pierwiastków z [H2]. Algorytm znajdowania liczb pierwszych spełniajacych (7) składa się z trzech poniższych procedur.

Procedura FindPrimeQ$(\mathcal{O}_K, \gamma, v, k)$ Niech $k, v \in \mathbb{N}$, gdzie $(v, k) = 1$, $v < k$ będą dane. Niech K będzie algebraicznym ciałem liczbowym stopnia $[K : \mathbb{Q}] = m$ wraz pierścieniem liczb całkowitym $\mathcal{O}_K = \{\sum_{j=1}^{m} b_i\omega_i, \omega_1 = 1, b_i \in \mathbb{Z}\}$. Niech $\gamma = \sum_{j=1}^{m} g_j\omega_j \in \mathcal{O}_K$ będzie takie, że $|g_j| \leq k$, oraz $N_{K/\mathbb{Q}}(\gamma) \equiv v \pmod{k}$. Procedura znajduje $\alpha = \sum_{j=1}^{m} a_j\omega_j \in \mathcal{O}_K$ takie, że $N_{K/\mathbb{Q}}(\alpha) \equiv v \pmod{n}$ jest liczbą pierwszą.

krok 1. Wylosuj $h_j \in \mathbb{Z}$, gdzie $j = 1, \ldots, m$.

krok 2. Dla $j = 1, \ldots, m$ oblicz $a_j = h_j k + g_j$.

krok 3. Zapisz $\alpha = \sum_{j=1}^{m} a_j\omega_j \in \mathcal{O}_K$ i oblicz $q = N_{K/\mathbb{Q}}(\alpha)$.

krok 4. Jeśli r jest liczbą pierwszą, to zakończ procedurę. W przeciwnym przypadku idź do kroku 1.

krok 5. Zwróć $\alpha \in \mathcal{O}_K$, q oraz $A(\alpha)$ takie, że $N_{K/\mathbb{Q}}(\alpha) = \det(A(\alpha))$.

Lemat 2. *Procedura* FindPrimeQ *znajduje $\alpha \in \mathcal{O}_K$ takie, że $N_{K/\mathbb{Q}}(\alpha) \equiv v \pmod{k}$ jest liczbą pierwszą.*

Dowód: zobacz [H5] (Lemat 2.1).

Uwaga. Można udowodnić, że procedura FindPrimeQ działa w czasie wielomianowym ze względu na liczbę bitów q i w ten sposób otrzymać analogiczne rezultaty jak w pracach [H4], [H2].

Niech $K = \mathbb{Q}(\theta)$ będzie algebraicznym ciałem liczbowym stopnia m and $\mathbb{Q}$. Niech

$$\mathcal{O}_K = \{a_1\omega_1 + \ldots + a_m\omega_m, \quad \omega_1 = 1, \quad a_i \in \mathbb{Z}\}, \tag{8}$$

będzie pierścieniem liczb całkowitych K, gdzie $\theta \in \mathcal{O}_K$. Niech $\theta = \theta_1, \theta_2, \ldots, \theta_m$ będą liczbami sprzężonymi do θ over $\mathbb{Q}$. Istnieje s różnych ciał spośród m sprzężonych ciał z K i każde różne ciało występuje m/s razy. Niech $K = K_1 = \ldots = K_{m/s}$ będą ciałami identycznymi z K. Ponadto, niech

$$\sigma_j : K \longrightarrow K_j, \qquad \theta \longmapsto \theta_j$$

będą izomorfizmami powyższych ciał, dla $j = 1, \ldots, m$. Niech L będzie domknięciem normalnym ciała K. Istnieją izomorfizmy

$$\tau_j : L \longrightarrow L,$$

takie, że $\tau_j(\gamma) = \sigma_j(\gamma)$ dla każdego $\gamma \in K$. Załóżmy, że $\alpha \in \mathcal{O}_K$ jest takie, że $q = N_{K/\mathbb{Q}}(\alpha) \equiv 1 \pmod{n}$ jest liczbą pierwszą, która nie dzieli $d(K)$ wyróżnika ciała K. Ponadto, załóżmy, że $\mathfrak{p} = \alpha\mathcal{O}_K$ rozpada się całkowicie w L. Stąd, ideał $\mathfrak{p}$ rozkłada się w $\mathcal{O}_L$ oraz

$$\mathfrak{p}\mathcal{O}_L = \mathfrak{P}_1 \cdots \mathfrak{P}_{[L:K]}, \qquad N_{L/\mathbb{Q}}(\mathfrak{P}_v) = q,$$

gdzie $\mathfrak{P}_v$ są różnymi ideałami pierwszymi $\mathcal{O}_L$ dla $v = 1, \ldots, [L : K]$. Ustalmy ideał $\mathfrak{P}_v$. Niech

$$\varphi : \mathcal{O}_L \longrightarrow \mathcal{O}_L/\mathfrak{P}_v, \qquad \gamma \longmapsto \gamma + \mathfrak{P}_v$$

będzie homomorfizmem, oraz niech

$$\lambda : \mathcal{O}_L/\mathfrak{P}_v \longrightarrow \mathbb{Z}/q\mathbb{Z}, \qquad 1 + \mathfrak{P}_v \longmapsto 1 + q\mathbb{Z}$$

będzie izomorfizmem. Ponieważ $\mathbb{Z} \cap \mathfrak{P}_v = q\mathbb{Z}$, więc istnieje $x_{jk} \in \mathbb{Z}$ takie, że

$$\lambda \circ \varphi(\tau_j(\omega_k)) \quad = \quad \lambda(\tau_j(\omega_k) + \mathfrak{P}_v) = \lambda(x_{jk} + \mathfrak{P}_v) = x_{jk} + q\mathbb{Z}, \tag{9}$$

dla $j = 1, \ldots, m$, $k = 1, \ldots, m$. Kolejna procedura znajduje liczby $x_{jk} \pmod q$ postaci (9), gdzie $j = 1, \ldots, m$, $k = 1, \ldots, m$.

ProceduraFindBase$(K, \theta, F, A(\alpha), q)$ Ustalmy $1 \leq l \leq m$. Niech $K = \mathbb{Q}(\theta)$ będzie algebraicznym ciałem liczbowym stopnia $[K : \mathbb{Q}] = m$, oraz niech s będzie liczbą nieizomorficznych ciał z K stopnia m. Niech L będzie rozszerzeniem normalnym ciała K, oraz niech $F(X) \in \mathbb{Z}[X]$ będzie wielomianem minimalnym θ. Ponadto, niech $\alpha \in \mathcal{O}_K$ będzie takie, że $q = N_{K/\mathbb{Q}}(\alpha) = \det(A(\alpha))$, gdzie q jest liczbą pierwszą oraz ideał $\mathfrak{p} = \alpha\mathcal{O}_K$ rozpada się całkowicie w L. Procedura oblicza $x_{jk} + q\mathbb{Z}$ postaci (9) dla $j = 1, \ldots, m$, $k = 1, \ldots, m$.

krok 1. Oblicz $t_{1l} = (-1)^{1+l} \det D_{1l}(\alpha)$ oraz $s_{jl} = (-1)^{j+l+1} \det E_{jl}(\alpha)$.

krok 2. Niech $x_{11} + q\mathbb{Z} = 1 + q\mathbb{Z}$.

krok 3. Oblicz $x_{1k} + q\mathbb{Z} = -s_{jl}(t_{1l})^{-1} + q\mathbb{Z}$, gdzie $k = 2, \ldots, m$

krok 4. Zapisz $\sigma_j(\omega_k) = \sum_{i=1}^{m} c_i\omega_i \in \mathcal{O}_K$, gdzie $j = 2, \ldots, m/s$, $k = 1, \ldots, m$

krok 5. Oblicz $x_{jk} + q\mathbb{Z} = \sum_{i=1}^{m} c_i x_{1i} + q\mathbb{Z}$, gdzie $j = 2, \ldots, m/s$, $k = 1, \ldots, m$, .

krok 6. jeśli $s > 1$, to

6.1 Oblicz $z_i + q\mathbb{Z} = \lambda \circ \varphi(\tau_i(\theta))$, gdzie $i = m/s + 1, \ldots, m$.

6.2 Wykorzystaj $z_i + q\mathbb{Z}$ do obliczenia $x_{jk} + q\mathbb{Z}$ dla $k = 1, \ldots, m$, $j = m/s + 1, \ldots, m$.

krok 7. Zwróć $x_{jk} + q\mathbb{Z}$, gdzie $k = 1, \ldots, m$, $j = 1, \ldots, m$.

Lemat 3. *Procedura* FindBases *oblicza* $x_{jk} + q\mathbb{Z}$ *postaci (9) dla* $j = 1, \ldots, m$, $k = 1, \ldots, m$.

Dowód: zobacz [H5] (Lemat 3.2).

Uwaga. Macierze $D_{1l}(\alpha)$ oraz $E_{jl}(\alpha)$ zdefiniowane są [H5] (zobacz (11)).

ProceduraFindPrimeP$(K, \theta, F, \Phi_K, A(\alpha), q, n)$ Niech $K = \mathbb{Q}(\theta)$ będzie CM-ciałem stopnia $2g$ wraz pierścieniem liczb całkowitym $\mathcal{O}_K = \{\sum_{k=1}^{2g} a_k\omega_k, \quad \omega_1 = 1, a_i \in \mathbb{Z}\}$, gdzie $\theta \in \mathcal{O}_K$. Niech (K, Φ_K) będzie pierwotnym CM-typem. Niech $\alpha \in \mathcal{O}_K$ będzie taka, że $q = N_{K/\mathbb{Q}}(\alpha) \equiv 1 \pmod n$ jest liczbą pierwszą, oraz niech ideał $\mathfrak{p} = \alpha\mathcal{O}_K$ rozpada się całkowicie w L. Procedura znajduje liczbę pierwszą p oraz p-liczbę Weila $\pi \in \mathcal{O}_{K'}$ taką, że $\Phi_n(\pi\overline{\pi}) \equiv 0 \pmod r$ oraz $N_{K'/\mathbb{Q}}(\pi - 1) \equiv 0 \pmod q$.

krok 1. Znajdź $\zeta_n + q\mathbb{Z}$ pierwiastek pierwotny stopnia n z jedności w $\mathbb{Z}/q\mathbb{Z}$.

krok 2. Użyj Procedury FindBase$(K, \theta, F, A(\alpha), q)$ do obliczenia $x_{jk} + q\mathbb{Z}$, gdzie $j, k = 1, \ldots, 2g$.

krok 3. Wylosuj $c_j + r\mathbb{Z}$, $d_j + q\mathbb{Z}$, gdzie $j = 2, \ldots, g$.

krok 4. Oblicz $c_1 + r\mathbb{Z} = \prod_{j=2}^{g} c_j^{-1} + r\mathbb{Z}$, oraz $d_1 + q\mathbb{Z} = \zeta_n \prod_{j=2}^{g} d_j^{-1} + q\mathbb{Z}$

krok 5. Znajdź rowiązania $b_k + q\mathbb{Z}$, $k = 1, \ldots, 2g$, układu równań liniowych nad $\mathbb{Z}/q\mathbb{Z}$

$$\begin{cases} \sum_{k=1}^{2g}(x_{1,k} + q\mathbb{Z})(b_k + q\mathbb{Z}) & = & c_1 + q\mathbb{Z}, \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^{2g}(x_{g,k} + q\mathbb{Z})(b_k + q\mathbb{Z}) & = & c_g + q\mathbb{Z}, \\ \sum_{k=1}^{2g}(x_{g+1,k} + q\mathbb{Z})(b_k + q\mathbb{Z}) & = & d_1 + q\mathbb{Z}, \\ \vdots & \ddots & \vdots \\ \sum_{k=1}^{2g}(x_{2g,k} + q\mathbb{Z})(b_k + q\mathbb{Z}) & = & d_g + q\mathbb{Z}. \end{cases} \tag{10}$$

krok 6. Niech $\beta = b_1 + b_2\omega_2 + \ldots + b_{2g}\omega_{2g} \in \mathcal{O}_K$.

krok 7. Oblicz $p = N_{K/\mathbb{Q}}(\beta)$. Jeśli p jest liczbą pierwszą, to $\pi = N_{\Phi_K}(\beta)$. W przeciwnym przypadku idź do kroku 3.

krok 8. Jeśli π nie generuje ciała K', to idź do kroku 3.

krok 9. Zwróć π oraz p.

Lemat 4. *Procedura* FindPrimeP *znajduje liczbą pierwszą p oraz p-liczbę Weila $\pi \in \mathcal{O}_{K'}$ taką, że $\Phi_n(\pi\overline{\pi}) \equiv 0 \pmod q$ oraz $N_{K'/\mathbb{Q}}(\pi - 1) \equiv 0 \pmod q$.*

Dowód: zobacz [H5] (Lemat 4.2).

Algorytm 5$(K, \mathcal{O}_K, \theta, F, \Phi_K, n, u, w)$. Ustalmy $n, u, w \in \mathbb{N}$ takie, że $(n, w) = 1$ or $(n, w) > 1$, $u = 1$. Niech $K = \mathbb{Q}(\theta)$ będzie CM-ciałem $[K : \mathbb{Q}] = 2g$ wraz z pierścieniem liczb algebraicznych całkowitych $\mathcal{O}_K = \{\sum_{j=1}^{2g} a_j\omega_j, \quad \omega_1 = 1, \quad a_j \in \mathbb{Z}\}$, gdzie $\theta \in \mathcal{O}_K$. Niech $F(X) \in \mathbb{Z}[X]$ będzie wielomianem minimalnym θ, oraz niech (K, Φ_K) będzie pierwotnym CM-typem ciała K. Algorytm $\alpha \in \mathcal{O}_K$ oraz $\pi \in \mathcal{O}_{K'}$ takie, że $N_{K/\mathbb{Q}}(\alpha) = q$, $N_{K'/\mathbb{Q}}(\pi) = p$ są liczbami pierwszymi, które są typu PF względem n.

krok 1. Jeśli $(n, w) = 1$, to $k = nw$ oraz znajdź $v \pmod k$ rowiązanie układu równań $v \equiv 1 \pmod n$, $v \equiv u \pmod w$.

krok 2. Jeśli $(n, w) > 1$ oraz $u = 1$, to $v = 1$ oblicz $k = [n, w]$ najmniejszą wspólną wielokrotność w oraz n.

krok 3. $\alpha, q, A(\alpha) :=$ Procedura FindPrimeQ(O_K, γ, v, k).

krok 4. $\pi, p :=$ Procedura FindPrimeP $(K, \theta, F, \Phi_K, A(\alpha), q)$.

krok 5. Zwróc q, p, α, π.

Twierdzenie 13. *Algorytm 5 znajduje $\alpha \in \mathcal{O}_K$ oraz $\pi \in \mathcal{O}_{K'}$ takie, że $N_{K/\mathbb{Q}}(\alpha) = r$, $N_{K'/\mathbb{Q}}(\pi) = q$ są liczbami pierwszymi, które są typu PF względem n.*

Dowód: zobacz [H5] (Twierdzenie 5.1).

5. **Omówienie pozostałych osiągnięć naukowo - badawczych**.

Niech K będzie ustalonym, całkowicie urojonym, ciałem liczb algebraicznym o wyróżniku $\Delta = \Delta(K)$ oraz stopniu $[K : \mathbb{Q}] = 2r_2$, gdzie $2r_2$ jest liczbą sprzężonych ciał z K. Niech $\mathcal{O}_K$ będzie pierścieniem liczb całkowitych ciała K, a $\mathfrak{f}$ ustalonym, niezerowym ideałem całkowitym $\mathcal{O}_K$. Oznaczmy przez $H \pmod{\mathfrak{f}}$ dowolną klasę ideałów modulo $\mathfrak{f}$ rozpatrywaną w wąskim sensie. Niech χ_0 oznacza charkter główny modulo $\mathfrak{f}$. Niech $\chi(H)$ będzie charakterem grupy klas ideałów $H \pmod{\mathfrak{f}}$, oraz niech $\chi(\mathfrak{a})$ będzie zwyczajnym rozszerzeniem charkteru $\chi(H)$. Niech $s = \sigma + it$. Niech $\zeta(s, \chi)$ będzie funkcja Heckego-Landaua odpowiadającą charakterowi χ, Niech $N_\chi(T)$ oznacza liczbę zer funkcji $\zeta(s, \chi)$ w prostokącie $0 \le \sigma \le 1$, $|t| \le T$. W pracy [C1] udowodniono następujące twierdzenie.

Twierdzenie 14. *Niech $T \geq 1$ oraz $\chi \neq \chi_0$ będzie charakterem pierwotnym modulo $\mathfrak{f}$. Wtedy*

$$\left| N_\chi(T) - \frac{T}{\pi} \log\left(\left(\frac{T}{2\pi e}\right)^{2r_2} |\Delta| N\mathfrak{f} \right) \right| \leq 2r_2(A_1 \log T + A_2 \log\log(T+5)) + \\ + 2r_2(A_3 \log(|\Delta| N\mathfrak{f}) + A_4 \log\log(|\Delta| N\mathfrak{f})^{\frac{1}{2r_2}} + A_5),$$

gdzie

$$A_1 = \frac{1}{2\pi \log 2}, \quad A_2 = \frac{2}{\log 2}, \quad A_3 = \frac{1}{4\pi \log 2}, \quad A_4 = \frac{1}{\log 2},$$

$$A_5 = A_5(T, \Delta, \mathfrak{f}) = \frac{1}{\pi \log 2}\left(1 + \frac{3}{\log(T+3)} + \frac{\log(|\Delta| N\mathfrak{f})^{\frac{1}{2r_2}}}{\log(T+3)}\right)^{-1} + \\ + \frac{1}{2\pi \log 2}(1 + 2\eta(T)) \log\left(1 + \frac{5}{T}\right) + \frac{1}{4\pi \log 2}\eta(T) \log(|\Delta| N\mathfrak{f}) + \\ + \frac{3r_2}{2 \log 2} \log\left(\left(1 + \frac{4}{\log((|\Delta| N\mathfrak{f})^{\frac{1}{2r_2}}(T+3))}\right)\left(1 + \frac{\log((|\Delta| N\mathfrak{f})^{\frac{1}{2r_2}})}{\log(T+3))}\right)\right) \\ + 3.347190,$$

oraz

$$\eta(T) = \left(\log\left(e^3 (|\Delta| N\mathfrak{f})^{\frac{1}{2r_2}}(T+3)\right)\right)^{-1} \leq \frac{1}{4}.$$

Dowód: zobacz [C1] (Twierdzenie 1.1).

Znaczenie oraz wykorzystanie twierdzeń, w których wszystkie stałe wyznaczone są explicite, zostało omówione w części autoreferatu dotyczącej pracy [H6]. Rezultaty powyższego typu dla funkcji dzeta Riemanna zostały udowodnione w pracach [4], [55], [67], [68]. Twiedzenia związane z funkcjami L-Dirichleta oraz z funkcją dzeta Dedekinda znajdziemy w pracach [39], [46], [69]. W pracy [C1] zastosowano metody dowodowe z [69] do oszacownia liczby zer $\zeta(s, \chi)$ oraz wprowadzono pewne modyfikacje dowodu dzięki czemu nieznacznie poprawiono najbardziej znaczące stałe występujące w twierdzeniu niż otrzymano by metodami z pracy [69].

Zagadnienia badane w pracach [D1], [D2], [D3], [D5], [D6] dotyczą rozwiązania problemu obliczeniowego zdefinowanego wcześniej w tym autoreferacie jako Problem 2. W pracy [D1] przeprowadzono analizę złożoności obliczeniowej algorytmów, które zaproponowane zostały przez autorów systemu kryptograficznego z kluczem publicznym XTR [44]. Dla systemu XTR wymaga się wygenerowania dwóch liczb pierwszych p oraz q postaci (1), gdzie $n = 6$. Metoda konstrukcji takich parametrów została opisana w Rozdziale 1 tego autoreferatu. W pracy [D1] autor udowodnił, że znalezienie liczb pierwszych powyższej postaci jest możliwe w czasie wielomianowym ze względu na liczbę bitów p oraz q. Jednak, analiza tego algorytmu wymagała założenia hipotezy Buniakowskiego-Schinzla [59] oraz hipotezy dotyczącej najmniejszej liczby pierwszej w ustalonym postępie arytmetycznym pochodzącej od Heatha-Browna [36]. W pracy [D2] zaproponowaną nową technikę znajdowania liczb pierwszych będących parametrami klucza w kryptosystemie XTR. Idea kostrukcji p oraz q związana jest z następującym faktem dotyczącym faktoryzacji wielomianów. Niech $P(X), Q(X) \in Z[X]$ będą pewnymi, ustalonymi wielomianami stopnia 2. Autor tego autoreferatu udowodnił, że $Q(X)$ dzieli $\Phi_6(P(X))$. Wykorzystując powyższą własność autor tego autoreferatu zbudował algorytm, który znajduje $r \in \mathbb{N}$ takie, że $p = P(r)$ oraz $q = Q(r)$ są jednocześnie liczbami pierwszymi [D2] . Przy założeniu hipotezy Buniakowskiego-Schinzla udowodniono, że taka metoda działa w czasie wielomianowym ze względu na liczbę bitów p oraz q.

Niech p będzie liczbą pierwszą oraz niech $k \in \mathbb{N}$ będzie ustalone. W 1999 wykorzystano ciągi rekurencyjne stopnia 3 nad ciałem $\mathbb{F}_{p^{3k}}$ o okresie $\Phi_3(p^k) = p^{2k} + p^k + 1$ do konstrukcji systemu kryptograficznego z kluczem publicznym [32]. Konstrukcja kluczy do kryptosystemu wymagała znalezienia odpowiednio dużej liczby pierwszej p oraz znajomości czynników pierwszych $\Phi_3(p^k)$. Dzięki takim parametrom można łatwo znaleźć elementy rzędu dzielącego $\Phi_3(p^k)$ czego wymagała konstrukcja algorytmu kryptograficznego. Aby problem logarytmu dykretnego był trudny do rozwiązania w wykorzystywanej strukturze algebraicznej liczba $\Phi_3(p^k)$ powinna posiadać dostatecznie duży dzielnik pierwszy. W pracy [D4] zaproponowano metody generowania liczby pierwszej p oraz czynników pierwszych $\Phi_3(p^k)$ będących składnikami klucza sytemu z [32].

W pracach [D3], [D5], [D6], rozważano przypadki znajdowania rozwiąnia Problemu 2 dla parametrów $n = 5, 10, 7, 12, 14$. Przegląd wyników autora tego autoreferatu wraz listą problemów otwartych znajduje się w pracy [D7].

Spis publikacji własnych nie wchodzących w skład osiągnięcia, o którym mowa w pkt. 4.

Publikacje wymienione w Załączniku 3 w punkcie II A (w czasopismach z bazy JCR):

[C1] Maciej Grześkowiak. Explicit zero-counting theorem for Hecke-Landau zeta-functions. *Bulletin of the Australian Mathematical Society*, 95:1–12, 2017.

Publikacje wymienione w Załączniku 3 w punkcie II C (monografie, artykuły w czasopismach spoza JCR):

[D1] Maciej Grześkowiak. Analysis of algorithms of generating key parameters for the XTR cryptosystem. *Tatra Mt. Math. Publ.*, 33:189–198, 2006.

[D2] Maciej Grześkowiak. New key generation algorithms for the XTR cryptosytem. *OTM Conferences (1), Lecture Notes in Computer Science*, 4277:439–449, 2006.

[D3] Maciej Grześkowiak. Generating a large prime factor of $p^4 + p^2 + 1$ in polynomial time. *On the Move to Meaningful Internet Systems: OTM 2008, Lecture Notes in Computer Science*, 5332:1140–1149, 2008.

[D4] Maciej Grześkowiak. On generating elements of orders dividing $p^{2k} \pm p^k + 1$. *IWSEC 2008, Lecture Notes in Computer Science*, 5312:1–19, 2008.

[D5] Maciej Grześkowiak. Algorithm for generating primes for the Giuliani-Gong public key system. *Journal of Internet Services and Information Security (JISIS)*, 1(2/3):21–31, 2011.

[D6] Maciej Grześkowiak. Generating elements of orders dividing $p^6 \pm p^5 + p^4 \pm p^3 + p^2 + p \pm 1$. *Ann. UMCS, Inf.*, 11(2):113–125, January 2011.

[D7] Maciej Grześkowiak. Prime numbers and cryptosystems based on discrete logarithms. *Studia Bezpieczeństwa Narodowego*, pages 163–176, 2014.

Spis cytowanych powyżej prac innych autorów.

[1] M. Agrawal, N. Kayal, and N. Saxena. Primes is in P. *Annals of Mathematics*, 160(2):781–793, September 2004.

[2] K. Akbary, A. Hambrook. A variant of the Bombieri-Vinogradov theorem with explicit constants and applications. *Math. Comp.*, 84(294):1901–1932, 2015.

[3] J. Bach, E. Shallit. *Algorithmic Number Theory, Volume I: Efficient Algorithms*. MIT Press, 1996.

[4] R. Backlund. Über die Nullstellen der Riemannschen Zetafunction. *Acta Math.*, 41:345–375, 1918.

[5] N. Balasubramanian, R. Koblitz. The improbability that an elliptic curve has subexponential discrete log problem under the menezes—okamoto—vanstone algorithm. *Journal of Cryptology*, 11(2):141–145, 1998.

[6] M. Barreto, P. Naehrig. Pairing-friendly elliptic curves of prime order. *Selected Areas in Cryptography, SAC 2005, Lecture Notes in Computer Science*, 3897:319–331, 2006.

[7] R. Bateman, P. Horn. A heuristic asymptotic formula concerning the distribution of prime numbers. *Math. Comp.*, 16:363–367, 1962.

[8] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing,. *Journal of Cryptology*, 17(4):297–319, 2004.

[9] D. Boneh, K. Rubin, and A. Silverberg. Finding composite order ordinary elliptic curves using the cocks-pinch method. *Journal of Number Theory*, 131(5):832–841, 2011.

[10] M. Boneh, D. Franklin. Identity-based encryption from the Weil pairing,. *SIAM J. Comput.*, 32(3):586–615, 2003.

[11] I. Borevich, Z. Shafarevich. Number theory. *Academic Press*, 1966.

[12] V. Bouniakowsky. Nouveaux théoremes relatifs à la distinction des nombres premiers et à la de composition des entiers en facteurs. *Sc. Math. Phys.*, 6:305–329, 1857.

[13] A. Brezing, F. Weng. Elliptic curves suitable for pairing based cryptography. *Des. Codes. Crypt.*, 1(37):133–141, 2005.

[14] P. Bröker, R. Stevenhagen. Efficient CM-constructions of elliptic curves over finite fields. *Math. Comp.*, 76:2161–2179, 2007.

[15] P. Bröker, R. Stevenhagen. Constructing elliptic curves of prime order. *Contemp. Math.*, 463:17–28, 2008.

[16] H. Cohen. A course in computational algebraic number theory. *Springer-Verlag*, 1996.

[17] H. Cohen, G. Frey, R. Avanzi, Ch. Doche, T. Lange, K. Nguyen, and Vercauteren F. Handbook of elliptic and hyperelliptic curve cryptography, second edition. *Chapman & Hall/CRC*, 2012.

[18] R. Dryło. A new method for constructing pairing-friendly abelian surfaces. *Proceedings of the 4th International Conference on Pairing-based Cryptography, Pairing'10, Lecture Notes in Computer Science*, 6487:298–311, 2010.

[19] R. Dryło. On constructing families of pairing-friendly elliptic curves with variable discriminant. *Proceedings of the 12th International Conference on Cryptology in India INDOCRYPT'11, Lecture Notes in Computer Science*, pages 310–319, 2011.

[20] A. Morain F. Dupont, R. Enge. Building curves with arbitrary small MOV degree over finite prime fields. *Journal of Cryptology*, 18(2):79–89, 2005.

[21] P. Dusart. Estimates of $\theta(x;k,l)$ for large values of x. *Math. Comp.*, 71(239):1137–1168, 2002.

[22] P. Dusart. Estimates of ψ, θ for large values of x without the Riemann hypothesis. *Math. Comp.*, 298(85):875–888, 2016.

[23] P. Dusart. Explicit estimates of some functions over primes. *Ramanujan J.*, 45(1):227—-251, 2018.

[24] K. Eisentraeger, K. Lauter. A crt algorithm for constructing genus 2 curves over finite fields. *Proceedings of Arithmetic, Geometry, and Coding Theory*, 21, 2004.

[25] A. Enge. The complexity of class polynomial computation via floating point approximations. *Math. Comp.*, 78(266):1089–1107, 2009.

[26] D. Freeman. Constructing pairing-friendly elliptic curves with embedding degree 10. *Algorithmic Number Theory ANTS, Lecture Notes in Computer Science*, 4076:452–465, 2006.

[27] D. Freeman. A generalized brezing-weng algorithm for constructing pairing-friendly ordinary abelian varieties. *Pairing-Based Cryptography, Pairing 2008, Lecture Notes in Computer Science*, 5209:146–163, 2008.

[28] D. Freeman, M. Scott, and E. Teske. A taxonomy of pairing-friendly elliptic curves. *Journal of Cryptology*, 23(2):224–280, 2010.

[29] D. Freeman, P. Stevenhagen, and M. Streng. Abelian varieties with prescribed embedding degree. *Algorithmic Number Theory ANTS, Lecture Notes in Computer Science*, 5011:60–73, 2008.

[30] J. Galbraith, S. McKee. The probability that the number of points on an elliptic curve over a finite field is prime. *J. London Math. Soc.*, 62(3):671–684, 2000.

[31] G. Giuliani, K. Gong. Analogues to the Gong-Harn and XTR cryptosystems. (34), 2003.

[32] Harn L.: Gong, G. Public-key cryptosystems based on cubic finite field extensions. *IEEE Trans. Inform. Theory*, 45(7):2601—-2605, 1999.

[33] D. Gordon. Discrete logarithms in $gf(p)$ using the number field sieve. *SIAM J. Discret. Math.*, 6(1):124 –138, 1993.

[34] G. Grenié, L. Molteni. Explicit versions of the prime ideal theorem for Dedekind zeta functions under grh. *Math. Comp.*, 298(85):889—-906, 2016.

[35] D. R Heath-Brown. Almost-primes in arithmetic progression and short intervals. *Proc. London Proc. Cambridge Phil. Soc.*, 86:357–375, 1978.

[36] D. R. Heath-Brown. Artin's conjecture for primitive roots. *The Quarterly Journal of Mathematics*, 37:27–38, 1986.

[37] J. Iwaniec, H. Urroz. Orders of CM elliptic curves modulo p with at most two primes. *Ann. Sc. Norm. Super. Pisa Cl. Sci.*, 5(9):815–832, 2010.

[38] A. Joux. A one round protocol for tripartite diffie-hellman. *J. Cryptology*, 17(4):263–276, 2004.

[39] N. Kadiri, H. Ng. Explicit zero density theorems for Dedekind zeta-functions. *J. Number Theory*, 132(4):748—775, 2012.

[40] E. Konstantinou, Y. Stamatiou, and C. Zaroliagis. On the efficient generation of elliptic curves over prime fields. *Cryptographic Hardware and Embedded Systems—CHES 2002, LNCS 2523*, pages 233–348, 2002.

[41] A. M. Lagarias, J. C. Odlyzko. Effective versions of the Chebotarev density theorem. *Algebraic number fields: L-functions and Galois properties (Proc. Sympos., Univ. Durham, Durham, 1975), Academic Press*, pages 409—-464, 1977.

[42] H. G. Lay, G. J. Zimmer. Constructing elliptic curves with given group order over large finite fields. *Algorithmic Number Theory ANTS 1994, Lectures Notes in Computer Scienc*, 877:250–263, 1994.

[43] E. Lenstra, A. Verheul. An overview of the XTR public key system. *In: Publickey cryptography and computational number theory (Warsaw, 2000), de Gruyter*, page 151–180, 2001.

[44] E. Lenstra, Verheul. The XTR public key system. *Proc. of 20th Annual International Cryptology Conference Crypto 2000, Advances in Cryptology (CRYPTO 2000), Lecture Notes in Computer Science*, 1880:1–19, 2000.

[45] K. Lenstra, A. Using cyclotomic polynomials to construct efficient discrete logarithm cryptosystems over finite fields. *Information Security and Privacy, Lecture Notes in Computer Science*, 1270:126–138, 1997.

[46] K. McCurley. Explicit estimates for the error term in the prime number theorem for arithmetic progressions. *Math. Comp.*, 42(165):265–285, 1984.

[47] A. Menezes, J. van Oorschot Katz, and S. A. P. C. Vanstone. Handbook of applied cryptography. *CRC Press*, 1996.

[48] J. F. Mestre. Construction de courbes de genre 2 à partir de leurs modules. *Effective Methods in Algebraic Geometry, Birkhäuser Boston*, pages 313–334, 1991.

[49] A. Miyaji, M. Nakabayashi, and S. Takano. New Explicit conditions of elliptic curve traces for FR-reduction. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 84(5):1234–1243, 2001.

[50] B. Phong, N. Valle. *The LLL Algorithm: Survey and Applications*. Springer Publishing Company, Incorporated, 1st edition, 2009.

[51] M. Pohlig, S. Hellman. An improved algorithm for computing logarithms over $gf(p)$ and its cryptographic significance. *IEEE Trans. Inf. Theor.*, 24(1):106–110, 2006.

[52] J.M. Pollard. Monte carlo methods for index computation mod p). 32, 07 1978.

[53] C. Pomerance. Fast, rigorous factorization and discrete logarithm algorithms. *In Discrete Algorithms and Complexity, Academic Press*, pages 119—-143, 1987.

[54] M. Radziejewski. On the distribution of algebraic numbers with prescribed factorization properties. *Acta Arith.*, 116(2):153—171, 2005.

[55] J. Rosser. Explicit bounds for some functions of prime numbers. *American Journal of Mathematics*, 63:211–232, 1941.

[56] A. Rubin, K. Silverberg. Torus-based cryptography. *In: Boneh D. (eds) Advances in Cryptology - CRYPTO 2003,Lecture Notes in Computer Science*, 2729:349–365, 2003.

[57] A. Rubin, K. Silverberg. Using primitive subgroups to do more with fewer bits. *In: Buell D. (eds) Algorithmic Number Theory. ANTS 2004, Lecture Notes in Computer Science*, 3076:18–41, 2004.

[58] E. Savaş, T. A. Schmidt, and C. K. Koç. Generating elliptic curves of prime order. *Cryptographic Hardware and Embedded Systems—CHES 2001, LNCS 2162*, pages 145—-161, 2001.

[59] W. Schinzel, A. Sierpiński. Sur certaines hypothèses concernment les nombres premiers. *Acta Arith.*, 4:185–208, 1958. Erratum **5** (1958).

[60] R. Schoof. Elliptic curves over finite fields and the computation of square roots (mod p). *Math. Comp.*, 44(170):483—-494, 1985.

[61] A. Shparlinski, I. Sutherland. On the distribution of Atkin and Elkies primes. *Foundations of Computational Mathematics*, 14(2):285–297, 2014.

[62] A. Silverberg. Future directions in algorithmic number theory. *The American Institute of Mathematics, https://aimath.org/WWN/primesinp/primesinp.pdf*, 2003.

[63] M. Streng. Computing igusa class polynomials. *Math. Comp.*, 83:275–309, 2014.

[64] A. Sutherland. Computing hilbert class polynomials with the chinese remainder theorem. *Math. Comp.*, 80:501–538, 2011.

[65] D. Talbot, J. Welsh. Complexity and cryptography. an introduction. *Cambridge University Press*, 2006.

[66] E. Teske. Square-root algorithms for the discrete logarithm problem (a survey). *In Public Key Cryptography and Computational Number Theory, Walter de Gruyter*, pages 283–301, 2001.

[67] T. Trudgian. An improved upper bound for the argument of the Riemann zeta-function on the critical line. *Math. Comp.*, 81(278):1053—1061, 2012.

[68] T. Trudgian. An improved upper bound for the argument of the Riemann zeta-function on the critical line ii. *J. Number Theory*, (134):280—-292, 2014.

[69] T. Trudgian. An improved upper bound for the error in the zero-counting formulae for Dirichlet L-functions and Dedekind zeta-functions. *Math. Comp.*, 84(293):1439–1450, 2015.

[70] T. Trudgian. An improved upper bound for the error in the zero-counting formulae for Dirichlet L-functions and dedekind zeta-functions. *Math. Comp.*, 84(293):1439—-1450, 2015.

[71] M. van Dijk, R. Granger, D. Page, K. Rubin, A. Silverberg, M. Stam, and D. Woodruff. Practical cryptography in high dimensional tori. *Proc. of 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Advances in Cryptology - (EUROCRYPT'05),Lecture Notes in Computer Science*, 3494:234–250, 2005.

[72] I. von zur Gathen, J. Shparlinski. Generating safe primes. *Journal of Mathematical Cryptology*, 7(4):333–365, 2013.

[73] S. Wagstaff. Greatest of the least primes in arithmetic progressions having a given modulus. *Math. Comp.*, 33:1073–1080, 1979.

[74] A. Walfisz. Zur additiven zahlentheorie. ii. *Mathematische Zeitschrift*, 1:592—-607, 1936.

[75] T. Xylouris. On the least prime in an arithmetic progression and estimates for the zeros of dirichlet l-functions. *Acta Arith.*, 150(1):65—-91, 2011.

Maciej Grześkowiak