ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Μελέτη του αλγόριθμου παραγοντοποίησης Quadratic Sieve

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Μαρία Νεφέλη Νταντουρή
ΑΕΜ: 3073

Επιβλέπων: Δρ. Κωνσταντίνος Δραζιώτης

ARISTOTLE UNIVERSITY OF THESSALONIKI
FACULTY OF SCIENCES
SCHOOL OF INFORMATICS

Study of Quadratic Sieve factorization algorithm

Graduate Thesis

Maria Nefeli Ntantouri
AEM: 3073

Supervisor: Dr. Konstantinos Draziotis

Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Σχολή Θετικών Επιστημών
Τμήμα Πληροφορικής

Το περιεχόμενο αυτής της εργασίας δεν απηχεί απαραίτητα τις απόψεις του Τμήματος, του Επιβλέποντα, ή της επιτροπής που την ενέκρινε.

Υπεύθυνη Δήλωση

(Υπογραφή) .....

Μαρία Νεφέλη Νταντουρή

Ευχαριστίες

Η πτυχιακή αυτή εργασία δεν θα είχε ολοκληρωθεί χωρίς την βοήθεια και υποστήριξη του καθηγητή μου κ. Κ. Δραζιώτη, ο οποίος ήταν πρόθυμος να λύσει κάθε απορία μου και ήταν διαθέσιμος για συχνή και άμεση επικοινωνία καθόλη τη διάρκεια διεκπεραίωσης της. Θα ήθελα επίσης να ευχαριστήσω τους γονείς μου για τη βοήθεια τους κατά τη διάρκεια των σπουδών μου.

Περίληψη

Η εργασία αυτή αποτελεί μία υλοποίηση του αλγόριθμου παραγοντοποίησης Quadratic Sieve (QS). Ο QS ήταν ο πιο γρήγορος αλγόριθμος παραγοντοποίησης, μέχρι την ανακάλυψη του Number Field Sieve (NFS) το 1993. Για ακέραιους με λιγότερα από 110 δεκαδικά ψηφία, ο QS παραμένει ο πιο γρήγορος, ενώ είναι αρκετά πιο απλός από τον NFS. Γίνεται αρχικά μία ιστορική αναφορά στη παραγοντοποίηση και στη συνέχεια επισημαίνεται η σημασία της παραγοντοποίησης στις μέρες μας. Τέλος, αναλύεται ο αλγόριθμος QS και το μαθηματικό υπόβαθρο που χρειάζεται.

Λέξεις Κλειδιά.

Quadratic Sieve, παραγοντοποίηση, τετραγωνικά υπόλοιπα

Summary

This thesis is an implementation of the Quadratic Sieve factorization algorithm (QS). QS was the fastest factorization algorithm until the invention of the Number Field Sieve (NFS) in 1993. For integers with less than 110 decimal digits, QS remains the fastest, and it is considerably simpler than NFS. At first, the historical background of factorization is mentioned and then the importance of factoring nowadays is highlighted. Finally, the QS algorithm is discussed, and the necessary mathematical background is presented.

Key Words. Quadratic Sieve, factorization, quadratic residues

Περιεχόμενα

ΚΕΦΑΛΑΙΟ 1

Εισαγωγή

Στην εργασία αυτή υλοποιήθηκε ο αλγόριθμος Quadratic Sieve για τη παραγοντοποίηση RSA-modulus, δηλαδή ακέραιους αριθμούς της μορφής $N = p \cdot q$, όπου οι p και q είναι πρώτοι αριθμοί. Η υλοποίηση έγινε σε γλώσσα προγραμματισμού Python. Ένα από τα βήματα του αλγόριθμου απαιτεί να βρεθεί, με τη χρήση γραμμικής άλγεβρας, ένα πεπερασμένο σώμα, του οποίου το γινόμενο ισούται με 0 πάνω στο $\mathbb{Z}_2$. Ένα από τα βασικά προβλήματα που αντιμετωπίσαμε ήταν οι τεράστειες απαιτήσεις μνήμης. Για εξασφάλιση μνήμης, τροποποιήσαμε τμήματα των αλγορίθμων, ώστε οι υπολογισμοί να γίνονται επί τόπου, χωρίς να χρειάζεται η αποθήκευση των δεδομένων σε κάποια δομή. Επιπλέον, καθώς οι αριθμοί που προκύπτουν κατά την εκτέλεση του αλγόριθμου είναι πολύ μεγάλοι, έγινε χρήση της βιβλιοθήκης **gmpy2**. Η gmpy2 είναι μια κωδικοποιημένη σε C ενότητα επέκτασης της Python που υποστηρίζει αριθμητική πολλαπλής ακρίβειας. Η gmpy2 είναι ο διάδοχος της αρχικής βιβλιοθήκης gmpy. Η τελευταία, υποστήριζε μόνο τη βιβλιοθήκη πολλαπλής ακρίβειας GMP [9], ενώ η gmpy2 προσθέτει υποστήριξη για τις βιβλιοθήκες MPFR [7] και MPC.

Η υλοποίηση του Quadratic Sieve μπορεί να βρεθεί εδώ.

ΚΕΦΑΛΑΙΟ 2

Ιστορικό Υπόβαθρο

2.1 Η Ιστορία της παραγοντοποίησης

Η σημασία της παραγοντοποίησης και του ελέγχου για πρώτους αριθμούς (primality testing) είναι εξαιρετικά σημαντική στη σημερινή κοινωνία, καθώς σε αυτές τις τεχνικές βασίζονται πολλές μέθοδοι για ασφαλή μεταφορά δεδομένων.

Ο πρώτος καταγεγραμμένος ορισμός των πρώτων αριθμών δόθηκε από τον Ευκλείδη γύρω στο 300 π.Χ. στη μαθηματική πραγματεία του "Στοιχεία" [1]. Ωστόσο, υπάρχουν ενδείξεις ότι η έννοια των πρώτων αριθμών ήταν γνωστή πολύ παλιότερα, για παράδειγμα στον Πυθαγόρα και τους ακολούθους του [29].

Παρόλο που η έννοια των πρώτων αριθμών μας είναι γνωστή για χιλιετίες [17], πολύ πρόσφατα καταφέραμε να ανακαλύψουμε αποδοτικούς τρόπους για να ελέγξουμε αν ένας αριθμός είναι πρώτος. Αυτό το φαινομενικά ασήμαντο έργο είναι στην πραγματικότητα πολύ πιο δύσκολο από ότι φαίνεται.

Μία έννοια που χρησιμοποιείται συχνά για αυτούς τους ελέγχους είναι η έννοια του "κόσκινου" (sieve). "Κόσκινο" θεωρείται η διαδικασία που ακολουθείται για να βρεθούν αριθμοί με κάποια συγκεκριμένα χαρακτηριστικά (για παράδειγμα πρώτοι) ψάχνοντας μεταξύ όλων των ακέραιων αριθμών μέχρι ένα συγκεκριμένο όριο, και απορρίπτοντας υποψήφιους μέχρις ότου μείνουν μόνο οι επιθυμητοί αριθμοί.

Παράδειγμα

Για να βρεθούν όλοι οι πρώτοι αριθμοί μικρότεροι ή ίσοι με το 30. Πρώτα, δημιουργούμε μία λίστα ακέραιων από το 2 μέχρι το 30. Ο πρώτος αριθμός είναι το 2, διαγράφουμε κάθε δεύτερο αριθμό στη λίστα μετά το 2, μετρώντας από το 2 και αυξάνοντας κατά 2:

2 3 ~~4~~ 5 ~~6~~ 7 ~~8~~ 9 ~~10~~ 11 ~~12~~ 13 ~~14~~ 15 ~~16~~ 17 ~~18~~ 19 ~~20~~ 21 ~~22~~ 23 ~~24~~ 25 ~~26~~ 27 ~~28~~ 29 ~~30~~

Ο επόμενος αριθμός μετά το 2 είναι το 3, διαγράφουμε κάθε τρίτο αριθμό από τη λίστα, αυτά θα είναι τα πολλαπλάσια του 3 στη λίστα:

2 3 ~~4~~ 5 ~~6~~ 7 ~~8 9 10~~ 11 ~~12~~ 13 ~~14 15 16~~ 17 ~~18~~ 19 ~~20 21 22~~ 23 ~~24~~ 25 ~~26 27 28~~ 29 ~~30~~

Ο επόμενος αριθμός στη λίστα που δεν έχει διαγραφεί είναι το 5, αντίστοιχα διαγράφουμε όλα τα πολλαπλάσια του 5:

2 3 ~~4~~ 5 ~~6~~ 7 ~~8 9 10~~ 11 ~~12~~ 13 ~~14 15 16~~ 17 ~~18~~ 19 ~~20 21 22~~ 23 ~~24 25 26 27 28~~ 29 ~~30~~

Επαναλαμβάνουμε την διαδικασία μέχρι ο αριθμός που ελέγχεται, υψωμένος στο τετράγωνο, να είναι μεγαλύτερος από το 30. Ο επόμενος πρώτος αριθμός που θα ελέγχαμε είναι το 7. Δεν χρειάζεται να ελεγχθεί όμως, καθώς $7^2 = 49 > 30$. Επομένως, ο αλγόριθμος τερματίζει και το σύνολο των πρώτων αριθμών μικρότερων του 30 είναι:

2 3 5 7 11 13 17 19 23 29

Αλγόριθμος 2.1.1 *Το κόσκινο του Ερατοσθένη [32]*
Είσοδος: *ένας ακέραιος* $n > 1$
΄Εξοδος: *όλοι οι πρώτοι αριθμοί από το* 2 *μέχρι το* n

1. Έστω A ένας πίνακας από Boolean τιμές, δεικτοδοτημένος από το 2 μέχρι το n, αρχικοποιημένες σε **true**
2. **for** $i = 2, 3, 4, \cdots \leq \sqrt{n}$:
3. if $A[i]$ **true**
4. for $j = i^2, i^2 + i, i^2 + 2i, i^2 + 3i, \cdots, \leq n$:
5. $A[j]$:= **false**

6. **return** όλα τα i για τα οποία το $A[i]$ είναι **true**.

Παράδειγμα κώδικα σε python:

```
def SieveOfEratosthenes(n):
    import math

    A = [True for i in range(n+1)]
    for i in range(2,int(math.sqrt(n))+1):
        if A[i]==True:
            for j in range(i**2,n+1,i):
                A[j]=False
    B=[]
    for i in range(2,len(A)):
        if A[i]==True:
            B.append(i)
    return B

if __name__ == '__main__':
    n = 30
    print("Following are the prime numbers smaller than or equal to", n)
    primes = SieveOfEratosthenes(n)
    print(primes)
```

Το κόσκινο του Ερατοσθένη αντιπροσωπεύει τον μοναδικό γνωστό αλγόριθμο από την αρχαιότητα που θα μπορούσαμε να χαρακτηρίσουμε ως έλεγχο πιστοποίησης πρώτων αριθμών, αλλά είναι πολύ αναποτελεσματικός και δεν μπορούσε να πλησιάσει την επαλήθευση μερικών από τους πρώτους αριθμούς που είναι γνωστοί σήμερα. Ο αριθμός $2^{6972593} - 1$, που έχει αποδειχθεί πως είναι πρώτος το 1999, έχει 2.098.960 δεκαδικά ψηφία. Η χρήση του κόσκινου του Ερατοσθένη για τον έλεγχο πιστοποίησης πρώτων αυτού του αριθμού θα διαρκούσε περισσότερο από το προσδόκιμο ζωής του ήλιου μας χρησιμοποιώντας τους ταχύτερους υπολογιστές που είναι γνωστοί σήμερα. Οι σύγχρονες τεχνικές που κατάφεραν να αποδείξουν τέτοιου είδους εντυπωσιακά αποτελέσματα βασίζονται σε ιδέες πρωτοπόρων, των οποίων η συνεισφορά θα επισημανθεί στη συνέχεια.

2.2 Δοκιμαστική Διαίρεση

Ο Leonardo of Pisa, γνωστός ως Fibonacci, στο πρώτο και πιο γνωστό του βιβλίο, "Book of Calculation" [30] (1202), έδωσε έναν αλγόριθμο παραγοντοποίησης ακέραιων αριθμών. Αποτελεί τον πιο απλό τρόπο παραγοντοποίησης και

ονομάζεται δοκιμαστική διαίρεση (trial division). Ο n διαιρείται διαδοχικά από φυσικούς αριθμούς μέχρι $\sqrt{n}$.

Ένα παράδειγμα εκτέλεσης του αλγόριθμου δοκιμαστικής διαίρεσης σε Python είναι το εξής:

```
def  trial_division(n):
    a = []
    while n%2 ==0:
        a.append(2)
        n //=2
    f = 3
    while f*f <= n:
        if n % f == 0:
            a.append(f)
            n //= f
        else:
            f += 2
    if n != 1:
        a.append(n)
    return a
```

Η μέθοδος αυτή μπορεί να χρησιμοποιηθεί για να ελέγξει αν ένας αριθμός n είναι πρώτος ή σύνθετος (primality testing). Αν οποιοσδήποτε από τους αριθμούς στο διάστημα $[2, \sqrt{n}]$ είναι διαιρέτης του n, τότε ο n είναι σύνθετος, διαφορετικά είναι πρώτος.

Επιπλέον, ο Fibonacci έφερε στην επιφάνεια την γνωστή κλάση των αριθμών Fibonacci, $\{F_n\}$, που ορίζεται από την αναγωγική ακολουθία:

$$F_1 = F_2 = 1, \quad F_n = F_{n-1} + F_{n-2} \quad (n \geq 3).$$

Αυτή η ακολουθία αριθμών έπαιξε σημαντικό ρόλο στους αλγόριθμους για έλεγχο πιστοποίησης πρώτων [17].

2.3 Πιστοποίηση πρώτων και τέλειοι αριθμοί

Ένα άλλο σύνολο αριθμών το οποίο επηρέασε σημαντικά την εξέλιξη των αλγόριθμων ελέγχου πρώτων αριθμών είναι το σύνολο των τέλειων αριθμών [33]. Τέλειος χαρακτηρίζεται ένας ακέραιος αριθμός n που ισούται με το άθροισμα των κανονικών διαιρετών του (τα $m \in \mathbb{N}$ όπου $m|n$ αλλά $m \neq n$). Για παράδειγμα

το 6 είναι τέλειος αριθμός, καθώς $6 = 3 + 2 + 1$. Οι τέλειοι αριθμοί εμφανίζονται στα "Στοιχεία" του Ευκλείδη, επομένως γνωρίζουμε πως η ιδέα υπάρχει εδώ και τουλάχιστον δύο χιλιάδες χρόνια. Ο αριθμός $2^{n-1}(2^n - 1)$ είναι τέλειος για $n = 2, 3, 5, 7$, και αυτοί είναι οι 4 πρωτοι τέλειοι αριθμοι: 6, 28, 496 και 8128, αντιστοίχως.

Ο Piedro Antonio Cataldi (1548-1626) ανέπτυξε μία αλγοριθμική προσέγγιση στον έλεγχο πρώτων [4]. Απέδειξε πως ο πέμπτος, έκτος και έβδομος τέλειος αριθμός είναι αντίστοιχα:

$$33550336 = 2^{12}(2^{13} - 1)$$

$$8589869056 = 2^{16}(2^{17-1})$$

$$137438691328 = 2^{18}(2^{19} - 1)$$

Δεν είναι σίγουρο ότι ο Caraldi ήταν ο πρώτος που τους ανακάλυψε, αλλά αυτές είναι οι πρώτες γνωστές αποδείξεις. Ήταν επίσης ο πρώτος που παρατήρησε πως αν ο $2^n - 1$ είναι πρώτος αριθμός τότε και ο n πρέπει να είναι πρώτος.

Θεώρημα 2.3.1 (Τέλειοι αριθμοί) *Αν ο $2^n - 1$ είναι πρώτος, τότε ο n είναι πρώτος και ο $N = 2^{n-1}(2^n - 1)$ είναι τέλειος αριθμός.*

Απόδειξη Καθώς $(2^m - 1)|(2^n - 1)$ όποτε ισχύει $m|n$, τότε ο n πρέπει να είναι πρώτος όποτε ο $2^n - 1$ είναι πρώτος. Σημειώνεται πως γενικά, αν $n = lm$, τότε για κάθε $b \in \mathbb{N}$,

$$b^n - 1 = (b^m - 1)\sum_{j=1}^{l} b^{l-j}.$$

Έστω S_1 είναι το άθροισμα όλων των διαιρετών του 2^{n-1} και έστω S_2 το άθροισμα όλων των διαιρετών του πρώτου αριθμού $2^n - 1$. Τότε το άθροισμα S όλων των διαιρετών του $2^{n-1}(2^n - 1)$ δίνεται από:

$$S = \sum_{l|2^{n-1}(2^n-1)} l = \sum_{l|2^{n-1}, l'|(2^n-1)} l \cdot l' = \sum_{l|2^{n-1}} l \sum_{l'|(2^n-1)} l' = S_1 S_2.$$

Επιπλέον, $S_1 = \sum_{j=0}^{n-1} 2^j$, και έτσι

$$S_1 = 2^n - 1.$$

Τέλος, καθώς $2^n - 1$ είναι πρώτος, τότε $S_2 = 2^n$. Επομένως,

$$S = 2^n(2^n - 1),$$

άρα ο $2^{n-1}(2^n - 1)$ είναι τέλειος.

■

Πολύ αργότερα από τον Cataldi, ο Euler απέδειξε πως κάθε άρτιος τέλειος αριθμός έχει τη μορφή από το παραπάνω Θεώρημα 2.3.1. Δεν είναι γνωστό αν υπάρχουν περιττοί τέλειοι αριθμοί, και η έρευνα για αυτούς έχει ξεπεράσει το όριο του 10^{300} [19, 24].

Το θεώρημα 2.3.1 δείχνει πως η έρευνα για άρτιους τέλειους αριθμούς είναι στη πραγματικότητα αναζήτηση για πρώτους της μορφής:

$$M_p = 2^p - 1, \text{ όπου ο } p \text{ είναι πρώτος.}$$

Αυτής της μορφής οι πρώτοι αριθμοί ονομάζονται Mersenne primes, από τον Marin Mersenne. Ο Pierre de Fermat, απέδειξε πως $223|(2^{37} - 1) = M_{37}$.

Θεώρημα 2.3.2 (FERMAT'S LITTLE THEOREM) *Αν ο p είναι ένας πρώτος αριθμός, τότε για κάθε ακέραιο a ισχύει $a^p \equiv a \mod p$.*

Απόδειξη [Με διωνυμικό θεώρημα] Με επαγωγή, θα αποδείξουμε το θεώρημα για όλους τους ακέραιους αριθμούς $a \geq 0$. Για $a = 0$ είναι τετριμμένο καθώς έχουμε $0^p \equiv 0 \mod p$. Έπειτα, πρέπει να δείξουμε πως το θεώρημα, αν αληθεύει για $a = k$, τότε αληθεύει και για $a = k + 1$. Για αυτό το επαγωγικό βήμα, χρειαζόμαστε το ακόλουθο λήμμα.

Λήμμα 2.3.1 *Για κάθε ακέραιο x και y και για κάθε πρώτο αριθμό p, ισχύει*

$$(x + y)^p \equiv x^p + y^p \mod p.$$

Θεωρούμε $k^p \equiv k \mod p$ και εργαζόμαστε για $(k + 1)^p$. Από το λήμμα 2.3.1 έχουμε πως

$$(k + 1)^p \equiv k^p + 1^p \mod p.$$

Χρησιμοποιώντας την υπόθεση επαγωγής, έχουμε πως $k^p \equiv k \mod p$, και $1^p = 1$. Επομένως,

$$(k+1)^p \equiv k+1 \mod p,$$

που είναι η δήλωση του θεωρήματος για $a = k+1$. ■

Το έργο του Fermat είχε σημαντική συνεισφορά στη παραγοντοποίηση. Γενικά, ο έλεγχος αν ένας αριθμός $n > 1$ είναι σύνθετος ή πρώτος, είναι πιο εύκολος από την παραγοντοποίηση. Επομένως, πρέπει πρώτα να ελέγχεται αν ο αριθμός είναι σύνθετος (primality test) και στη συνέχεια να του εφαρμόζεται ένας αλγόριθμος παραγοντοποίησης.

Το 1643, ο Fermat ανέπτυξε μία μέθοδο παραγοντοποίησης η οποία βασίζεται σε μία απλή παρατήρηση. Αν ο $n = rs$ είναι ένας περιττός φυσικός αριθμός με $r < \sqrt{n}$, τότε

$$n = a^2 - b^2, \text{ όπου } a = (s+r)/2 \text{ και } b = (s-r)/2.$$

Η μέθοδος αυτή είναι αποτελεσματική αν ο n έχει κάποιον παράγοντα κοντά στο $\sqrt{n}$. Για να βρεθεί ένας παράγοντας του n, ελέγχουμε τις διάφορες τιμές του $a^2 - n$ καθώς ο a κυμαίνεται στις τιμές $a = \lfloor \sqrt{n} \rfloor, \lfloor \sqrt{n} \rfloor + 1, \cdots, \lfloor (n+9)/6 \rfloor + 1$ μέχρις ότου βρεθεί ένα τέλειο τετράγωνο, το οποίο θα παίξει τον ρόλο του b^2. Όταν καθοριστούν οι κατάλληλες τιμές των a και b, μπορούμε να βρούμε τους παράγοντες r και s. Αυτό ονομάζεται διαφορά τετραγώνων (difference of squares).

Αλγόριθμος 2.3.1 *Μέθοδος Fermat*
Είσοδος: *n θετικός περιττός ακέραιος, $k > 0$ μικρός ακέραιος*
Έξοδος: *Ένας μη τετριμμένος διαιρέτης του n*

1. ***for*** $\lfloor \sqrt{kn} \rfloor \le a \le \lfloor (n+9)/6 \rfloor$ ***do***
2. $\quad b \leftarrow \sqrt{a^2 - kn}$
3. $\quad$ ***if*** *b είναι ακέραιος* ***then***
4. $\quad\quad$ *print* $\gcd(a-b, n)$
5. $\quad$ ***end***
6. ***end***

2.4 Τετραγωνικά υπόλοιπα

Το 1830, μία πολύτιμη τεχνική για παραγοντοποίηση οποιουδήποτε περιττού ακέραιου n ανακαλύφθηκε από τον Andrien-Marie Legendre, χρησιμοποιώντας τη θεωρία των τετραγωνικών υπολοίπων (quadratic residues). Αυτή η μέθοδος μελετήθηκε από τον Euler, αναπτύχθηκε ιδιαιτέρως από τον Gauss, και εφαρμόστηκε από τον Legendre στην ανάπτυξη μιας νέας μεθόδου κοσκινίσματος.

Ορισμός 2.4.1 *΄Ενας ακέραιος c ονομάζεται τετραγωνικό υπόλοιπο modulo $n \in \mathbb{N}$ αν υπάρχει ένας ακέραιος x τέτοιος, ώστε*

$$c \equiv x^2 \mod n.$$

Έστω ότι ψάχνουμε τους πρώτους διαιρέτες ενός ακεραίου n. Για διαφορετικούς πρώτους αριθμούς p, ο Legendre μελέτησε τις ισοτιμίες (congruences) της μορφής

$$x^2 \equiv \pm p \mod n.$$

Ορισμός 2.4.2 (Legendre symbol) *΄Εστω $p \in \mathbb{N}$. Τότε το σύμβολο του Legendre ορίζεται ως:*

$$\left(\frac{n}{p}\right) = \begin{cases} 1 & \text{αν το } a \text{ είναι τετραγωνικό υπόλοιπο} \mod p \\ 0 & \text{αν } p|a \\ -1 & \text{αν το } a \text{ είναι μη-τετραγωνικό υπόλποιπο} \mod p \end{cases} \tag{1}$$

Οι πιο βασικές ιδιότητες του σύμβολου του Legendre δίνονται στο παρακάτω θεώρημα, δείτε [27, Theorem 2.59.].

Θεώρημα 2.4.1 *΄Εστω p ένας περιττός πρώτος αριθμός και έστω r και s ακέραιοι. Τότε:*

1. $\left(\frac{rs}{p}\right) = \left(\frac{r}{p}\right)\left(\frac{s}{p}\right)$
2. *Αν* $r \equiv s \mod p$, *τότε* $\left(\frac{r}{p}\right) = \left(\frac{s}{p}\right)$
3. *Αν* $p \nmid r$, *τότε* $\left(\frac{r^2}{p}\right) = +1$ *και* $\left(\frac{r^2 s}{p}\right) = \left(\frac{s}{p}\right)$

4. $r^{(p-1)/2} \equiv \left(\frac{r}{p}\right) \mod p$

5. $\left(\frac{-1}{p}\right) = +1$ *όταν* $p \equiv 1 \mod 4$ *και* $\left(\frac{-1}{p}\right) = -1$ *όταν* $p \equiv 3 \mod 4$

6. $\left(\frac{2}{p}\right) = +1$ *όταν* $p \equiv 1$ *ή* $7 \mod 8$, *και* $\left(\frac{2}{p}\right) = -1$ *όταν* $p \equiv 3$ *ή* $5 \mod 8$

7. *Αν* q *είναι ένας περιττός πρώτος διαφορετικός από τον* p, *τότε* $\left(\frac{p}{q}\right) = \left(\frac{q}{p}\right)$ *όταν* $p \equiv 1 \mod 4$ *ή* $q \equiv 1 \mod 4$, *και* $\left(\frac{p}{q}\right) = -\left(\frac{q}{p}\right)$ *όταν* $p \equiv q \equiv 3 \mod 4$

Έστω ότι η λύση της ισοτιμίας $x^2 \equiv \pm p \mod n$ μπορούσε να βρεθεί. Αυτό συνεπάγεται ότι ο $\pm p$ είναι τετραγωνικό υπόλοιπο modulo όλων των πρώτων παραγόντων του n. Το γεγονός αυτό μπορεί να χρησιμοποιηθεί για να μειώσει σημαντικά την αναζήτηση για τους πρώτους διαιρέτες του n λαμβάνοντας υπόψη μόνο τους πρώτους αριθμούς q για τους οποίους το p είναι επίσης ένα τετραγωνικό υπόλοιπο $\mod q$.

Κάποια από τα αποτελέσματα του Euler είχαν στην πραγματικότητα προβλέψει το έργο του Legendre. Θεώρησε δύο αναπαραστάσεις του n:

$$n = x^2 + ay^2 = z^2 + aw^2$$

έτσι,

$$(xw)^2 \equiv (n - ay^2)w^2 \equiv nw^2 - ay^2w^2 \equiv -ay^2w^2 \equiv (z^2 - n)y^2 \equiv (zy)^2 \mod n$$

και έχουμε έναν πιθανό παράγοντα του n. Η βασική ιδέα είναι πως, για ένα δεδομένο n, είναι πιθανό πως αν βρούμε τους ακεραίους x, y τέτοιους ώστε

$$x^2 \equiv y^2 \mod n \text{ και } x \not\equiv \pm y \mod n,$$

τότε ο $\gcd(x - y, n)$ είναι ένας μη τετριμμένος παράγοντας του n. Αυτή η ιδέα χρησιμοποιείται από πολλούς αλγόριθμους: Pollard's $p - 1$ algorithm, continued fraction algorithm, quadratic sieve, και τον ισχυρό number field sieve.

Μια εκδοχή της μεθόδου του Legendre για παραγοντοποίηση αναπτύχθηκε από τον μαθηματικό Carl Friederich Gauss, στο έργο του "Disquisitiones Arithmetica" [8].

Ίσως μία από τις προσωπικότητες του 19ου αιώνα που επηρέασαν περισσότερο την περιοχή της πιστοποίησης πρώτων, ήταν ο Francois Edouard Anatole Lucas.

Ο Lucas ενδιαφερόταν για τα πρακτικά μαθηματικά (recreational mathematics), όπως η εφεύρεσή του γνωστή ως "Πύργοι του Ανόι" [11]. Ωστόσο, το βασικό του ενδιαφέρον ήταν η θεωρία αριθμών, και συγκεκριμένα η Διοφαντική ανάλυση. Παρόλο που πέρασε μόνο τα χρόνια 1875-1878 πάνω σε προβλήματα παραγοντοποίησης και primality testing, η συνεισφορά του ήταν εντυπωσιακή. Μελέτησε του αριθμούς Fibonacci και μέχρι το 1877 είχε παραγοντοποιήσει πλήρως τους πρώτους 60 από αυτούς.

Ο Maurice Borisovich Kraitchik, είναι αυτός ο οποίος προσπάθησε να συνδέσει τους πρώτους αριθμούς για να δημιουργήσει ένα τετράγωνο. Ο Kraitchik, γύρω στο 1920, βασιζόμενος στη μέθοδο του Fermat, αιτιολόγησε ότι είναι αρκετό να βρεθεί ένα πολλαπλάσιο του n ως διαφορά τετραγώνων. Επέλεξε μία τετραγωνική εξίσωση της μορφής $kn = ax^2 \pm by^2$ για κάποιο $k \in \mathbb{N}$. Στην απλούστερη μορφή με $k = a = b = 1$, εφάρμοσε το "κόσκινο" πάνω στο τετραγωνικό πολυώνυμο $x^2 - n$ για $x \geq \lfloor \sqrt{n} \rfloor$. Αυτή είναι η βασική ιδέα πίσω από τον quadratic sieve. Αυτό που έκανε στη πραγματικότητα ο Kraitchik είναι ότι χρησιμοποίησε την εύρεση δύο διακριτών υπολοίπων σε ένα δεδομένο πρώτο αριθμό για να σχηματίσει ένα τετράγωνο, το οποίο δεν μπόρεσε να κάνει ο Legendre.

Το 1931, οι D. H. Lehmer and R. E. Powers περιέγραψαν έναν γενικού-σκοπού αλγόριθμο γνωστό ως μέθοδο παραγοντοποίησης με συνεχή κλάσματα (continued fraction factorization method) [13]. Το 1975, οι Brillhart-Morrison υλοποίησαν τον continued fraction factoring αλγόριθμο [13] και κατάφεραν να παραγοντοποιήσουν έναν 50-ψήφιο αριθμό, ενώ μέχρι το 1970 ήταν μετά βίας δυνατόν να παραγοντοποιηθεί ένας 20-ψήφιος αριθμός. Ο αλγόριθμος αυτός αποτελεί τον πρώτο αλγόριθμο παραγοντοποίησης υποεκθετικής πολυπλοκότητας. Η μέθοδος αυτή προσπαθεί να δημιουργήσει μία ισοτιμία της μορφής $x^2 \equiv y^2 \mod n$ με την προϋπόθεση πως $x \not\equiv \pm y \mod n$. Τότε, ο n μπορεί να παραγοντοποιηθεί: $n = \gcd(x + y, n) \cdot \gcd(x - y, n)$.

Έστω N ένας περιττός, σύνθετος αριθμός με $N > 1$. Τότε ο αλγόριθμος μπορεί να περιγραφεί με τρία βήματα [18] :

Αλγόριθμος 2.4.1 *Μέθοδος συνεχών κλασμάτων των Morrison-Brillhart*

1. *Ανέπτυξε το $\sqrt{N}$, ή το $\sqrt{kN}$ για κάποιο κατάλληλα επιλεγμένο $k \geq 1$, σε ένα*

απλό συνεχές κλάσμα

$$\sqrt{kN} = q_0 + \cfrac{1}{q_1 + \cfrac{1}{q_2 + \cfrac{1}{\cdots \cfrac{1}{\cfrac{\sqrt{kN}+P_n}{Q_n}}}}}$$

για κάποιο σημείο $n = n_0$. *Για κάθε τιμή* $n, 1 \leq n \leq n_0$, *η ταυτότητα*

$$A_{n-1}^2 - kNB_{n-1}^2 = (-1)^n Q_n$$

όπου A_n/B_n *είναι η* n*-ιοστή συγκλίνουσα, συνεπάγεται την ισοδυναμία*

$$A_{n-1}^2 \equiv (-1)^n Q_n \mod N.$$

Ονομάζουμε τα ζευγάρια (A_{n-1}, Q_n) *ένα* $A - Q$ *ζευγάρι. Σημειώνεται πως σε αυτό το βήμα θεωρούμε πως ο* n_0 *είναι αρκετά μεγάλο ώστε να δημιουργηθούν αρκετά* $A - Q$ *ζευγάρια.*

2. *Βρες μεταξύ των* $A-Q$ *ζευγαριών από το βήμα 1, ορισμένα ζευγάρια, τα οποία ονομάζουμε S-σύνολα, τέτοια ώστε το γινόμενο* $\prod_i (-1)^i Q_i$ *να είναι τετράγωνο. Αν δεν μπορεί να δημιουργηθεί τέτοιο σύνολο, επέστρεψε στο βήμα ένα και μεγάλωσε το* $\sqrt{kN}$.

3. *Κάθε S-σύνολο που βρέθηκε στο βήμα 2 υποκύπτει στη παρακάτω ισοτιμία:*

$$A^2 = \prod_i A_{i-1}^2 = \prod_i (-1)^i Q_i = Q^2 \mod N$$

όπου $1 \leq A < N$. *Υπολόγισε τα* A *και* Q *από την ισοτιμία και στη συνέχεια υπολόγισε το* $\gcd(A - Q, N) = D$ *για όλα τα S-σύνολα τα οποία έχουν παραχθεί. Αν* $1 < D < N$ *για κάποιο S-σύνολο τότε ο αλγόριθμος τερματίζει και ο* D *είναι ένας μη τετριμμένος διαιρέτης του* N. *Αλλιώς επέστρεψε στο βήμα 1 και μεγάλωσε το* $\sqrt{kN}$.

Μία βελτίωση της μέθοδου του Fermat, παρουσιάστηκε το 1981 από τον John D. Dixon. Η βελτίωση του αλγόριθμου αυτού είναι η μέθοδος του τετραγωνικού κοσκίνου (Quadratic Sieve). Το 1984 Carl Bernard Pomerance, παρουσίασε τον Quadratic Sieve αλγόριθμο, ο οποίος σήμερα είναι ο δεύτερος πιο γρήγορος αλγόριθμος παραγοντοποίησης. Το 1994 ο QS κατάφερε να παραγοντοποιήσει το διάσημο RSA-129. Το RSA-129, όπως προκύπτει και από το όνομα του, έχει 129 δεκαδικά ψηφία (426 bits), παραγοντοποιήθηκε με τη χρήση περίπου

1600 υπολογιστών από περίπου 600 εθελοντές συνδεδεμένους στο διαδίκτυο. Ο αριθμός και η παραγοντοποίηση του είναι η εξής:

RSA-129 = 114381625757888867669235779976146612010218296721242362562
561842935706935245733897830597123563958705058989075147599290026879
543541

$$\text{RSA} - 129 =$$

$$3490529510847650949147849619903898133417764638493387843990820577\times$$

$$3276913299326670954996198819083446141317764296799294253979828853 3$$

Η δοκιμασία περιλάμβανε και ένα κρυπτογραφημένο μήνυμα με το RSA-129. Όταν αποκρυπτογραφήθηκε χρησιμοποιώντας την παραγοντοποίηση, το μήνυμα που αποκαλύφθηκε ήταν: "The Magic Words are Squeamish Ossifrage" [31].

Ο Pomerance είναι επίσης ένας από τους εφευρέτες μίας μεθόδου πιστοποίησης πρώτων, γνωστή ως Adleman–Pomerance–Rumely. Ένας αλγόριθμος που καθορίζει αν ένας αριθμός είναι πρώτος ή όχι. Σε αντίθεση με άλλους αλγόριθμους για τον ίδιο σκοπό, αποφεύγει τη χρήση τυχαίων αριθμών, το οποίο καθιστά τον αλγόριθμο ντετερμινιστικό. Αργότερα βελτιώθηκε από τους Henri Cohen και Hendrik Willem Lenstra, για αυτό και αναφέρεται συνήθως ως APR-CL [2]. Η πολυπλοκότητα του αλγόριθμου για έλεγχο ενός αριθμού n είναι:

$$(\log n)^{O(\log\log\log n)}.$$

Το 1990 ο quadratic sieve διπλασίασε το μήκος των αριθμών που μπορούν να παραγοντοποιηθούν, παραγοντοποιόντας έναν 116-ψήφιο αριθμό.

Το 1993, ο number field sieve εκθρόνισε τον QS, καθώς με επιτυχία παραγοντοποίησε το 130-RSA σε περίπου 15% του χρόνου που θα χρειαζόταν ο QS. Η αρχική ιδέα του αλγόριθμου προέρχεται από τον Pollard και αργότερα βελτιώθηκε από τους Lenstra, Pomerance και Coppersmith [6]. Ευριστικά, η πολυπλοκότητά [23] του για παραγοντοποίηση ενός ακεραίου n (με $\lfloor \log_2 n \rfloor + 1$ bits) είναι

$$\exp\left(\left(\sqrt[3]{\frac{64}{3}} + o(1)\right)(\ln n)^{\frac{1}{3}}(\ln\ln n)^{\frac{2}{3}}\right) = L_n\left[\frac{1}{3}, \sqrt[3]{\frac{64}{9}}\right].$$

Παρ' όλ' αυτά, για αριθμούς μέχρι 100 δεκαδικά ψηδία, ($\sim$ 332 bits) ο QS παραμένει πιο γρήγορος από τον GNFS, και σημαντικά απλούστερος. Ο QS

είναι ένας γενικού-σκοπού αλγόριθμος παραγοντοποίησης, που σημαίνει πως ο χρόνος εκτέλεσής του εξαρτάται μόνο από το μέγεθος του ακέραιου αριθμού που πρέπει να παραγοντοποιηθεί και όχι σε άλλες ιδιότητες [21].

Για περισσότερη μελέτη των αλγορίθμων παραγοντοποίησης, προτείνεται το βιβλίο των R. Crandall και C. Pomerance: Prime numbers [5].

ΚΕΦΑΛΑΙΟ 3

Η σημασία της παραγοντοποίησης

3.1 Εισαγωγή

Η εποχή μας κυριαρχείται από πληροφορίες και η ανάγκη για μυστικότητα είναι ύψιστης σημασίας. Καθώς στέλνουμε emails, μηνύματα και οικονομικά δεδομένα, ελπίζουμε να παραμείνουν ιδιωτικά. Η παραγοντοποίηση και η πιστοποίηση πρώτων παίζουν κυρίαρχο ρόλο στην ανάπτυξη σύγχρονων κρυπτογραφικών τεχνικών. Σε καθημερινή βάση χρησιμοποιούμε μία κάρτα για τις συναλλαγές μας. Εμπιστευόμαστε πως τα προσωπικά μας δεδομένα θα παραμείνουν κρυφά, πως θα πληρωθεί το σωστό ποσό και πως τα χρήματα θα μεταφερθούν στον κατάλληλο λογαριασμό.

Όλα αυτά είναι εφικτά χάρη στην κρυπτογράφηση δημοσίου κλειδιού. Σε ένα κρυπτοσύστημα δημοσίου κλειδιού, το κλειδί κρυπτογράφησης είναι δημόσιο και διαφορετικό από το κλειδί αποκρυπτογράφησης, το οποίο είναι ιδιωτικό. Ένας χρήστης δημιουργεί και δημοσιεύει ένα δημόσιο κλειδί. Ο καθένας μπορεί να κρυπτογραφήσει ένα μήνυμα, με τη χρήση του δημόσιου κλειδιού, αλλά το μήνυμα αυτό μπορεί να αποκρυπτογραφηθεί μόνο από αυτόν που γνωρίζει τους πρώτους αριθμούς. Τέτοιου είδους κρυπτοσυστήματα βασίζονται σε δύο προβλήματα τα οποία θεωρούνται άλυτα σε πολυωνυμικό χρόνο, το Diffie-Hellman πρόβλημα και τη παραγοντοποίηση σε πρώτους παράγοντες (prime factorization). Ένα από τα πιο γνωστά κρυπτοσυστήματα δημοσίου κλειδιού είναι το RSA. Ο αλγόριθμος περιλαμβάνει 4 βήματα: την παραγωγή των κλειδιών, τον διαμοιρασμό των κλειδιών, την κρυπτογράφηση και την αποκρυπτογράφηση.

3.2 RSA

Το RSA είναι ένα κρυπτοσύστημα δημοσίου κλειδιού, το οποίο χρησιμοποιείται ευρέως για την ασφαλή μεταφορά δεδομένων. Το ακρωνύμιο προέρχεται από τα επώνυμα των Ron Rivest, Adi Shamir και Leonard Adleman, οι οποίοι δημοσίευσαν τον αλγόριθμο το 1978 [26].

3.2.1 Παραγωγή Κλειδιού Κρυπτογράφησης

Αρχικά επιλέγονται δύο διαφορετικοί πρώτοι αριθμοί p και q. Οι αριθμοί αυτοί είναι τυχαίοι και πρέπει να είναι του ίδιου δυαδικού μεγέθους αλλά να διαφέρουν για μερικά δεκαδικά ψηφία, ώστε να είναι πιο δύσκολη η παραγοντοποίηση [26]. Τα p και q παραμένουν κρυφά. Στη συνέχεια υπολογίζεται ο $n = pq$, ο οποίος χρησιμοποιείται ως modulus για το δημόσιο και το ιδιωτικό κλειδί και δημοσιεύεται. Υπολογίζεται ο $\lambda(n)$, όπου λ είναι η συνάρτηση Carmichael, ο οποίος παραμένει κρυφός.

Ορισμός 3.2.1 (Συνάρτηση Carmichael) *Η συνάρτηση Carmichael συσχετίζει με κάθε θετικό ακέραιο n έναν θετικό ακέραιο $\lambda(n)$, ο οποίος ορίζεται ως ο μικρότερος θετικός ακέραιος m τέτοιος, ώστε:*

$$a^m \equiv 1 \mod n$$

για κάθε ακέραιο a μεταξύ 1 και n, με $\gcd(a, n) = 1$ (δείτε [20]).

Καθώς, $n = pq$, το $\lambda(n) = \text{lcm}(\lambda(p), \lambda(q))$, όπου lcm είναι το ελάχιστο κοινό πολλαπλάσιο. Εφόσον ο p και ο q είναι πρώτοι, προκύπτει $\lambda(p) = \varphi(p) = p - 1$ και αντίστοιχα $\lambda(q) = q - 1$. Επομένως, $\lambda(n) = \text{lcm}(p - 1, q - 1)$.

Ορισμός 3.2.2 (Συνάρτηση του Euler) *Η συνάρτηση του Euler μετράει τους θετικούς ακέραιους αριθμούς μέχρι έναν δεδομένο ακέραιο n που είναι σχετικά πρώτοι ως προς τον n. Συμβολίζεται ως $\varphi(n)$ (δείτε [14, 3.2.2-Ορισμός 15]).*

Στη συνέχεια επιλέγεται ένας ακέραιος e τέτοιος, ώστε $1 < e < \lambda(n)$, και $\gcd(e, \lambda(n)) = 1$. Ο e δημοσιεύεται ως κομμάτι του δημοσίου κλειδιού.

Ορίζεται ο d ως $d \equiv e^{-1} \mod \lambda(n)$, ο οποίος είναι ο αντίστροφος του e modulo $\lambda(n)$. Ο d μπορεί να υπολογιστεί με τη χρήση του Εκτεταμένου ευκλείδειου αλγόριθμου. Ο d παραμένει κρυφός ως εκθέτης του ιδιωτικού κλειδιού.

Ο Euler έχει αποδείξει ότι:

Θεώρημα 3.2.1 (Euler) *Για κάθε θετικό ακέραιο m πρώτο προς τον ακέραιο a, ισχύει*

$$a^{\varphi(m)} \equiv 1 \mod m.$$

Το θεώρημα 3.2.1 [34] μπορεί να χρησιμοποιηθεί για τον υπολογισμό αντίστροφων στοιχείων. Έστω ότι πρέπει να υπολογιστεί το $a^{-1} \mod m$, τότε έχουμε

$$a^{-1} \equiv a^{\varphi(m)-1} \mod m.$$

Αν $m = p$ είναι πρώτος και $p \nmid a$ προκύπτει

$$a^{p-1} \equiv 1 \mod p.$$

Και ισοδύναμα

$$a^{p} \equiv a \mod p,$$

για κάθε ακέραιο a και πρώτο p.

Επομένως, το δημόσιο κλειδί αποτελείται από το modulus n και τον δημόσιο εκθέτη e. Το ιδιωτικό κλειδί περιλαμβάνει τον εκθέτη d. Τα p, q και $\lambda(n)$ πρέπει επίσης να παραμείνουν κρυφά, καθώς μπορούν να χρησιμοποιηθούν για τον υπολογισμό του d.

3.2.2 Κρυπτογράφηση

Για να κρυπτογραφηθεί ένα μήνυμα M, πρέπει πρώτα να μετατραπεί σε έναν ακέραιο αριθμό m, έτσι ώστε $0 \leq m < n$, χρησιμοποιώντας ένα συμφωνημένο αντιστρεπτό πρωτόκολλο, γνωστό ως σχήμα συμπλήρωσης (padding scheme). Έπειτα, υπολογίζεται το κρυπτοκείμενο c, με τη χρήση του δημοσίου εκθέτη e, σύμφωνα με την εξίσωση:

$$m^{e} \equiv c \mod n.$$

3.2.3 Αποκρυπτογράφηση

Ο παραλήπτης μπορεί να ανακτήσει το m από το c χρησιμοποιώντας τον εκθέτη d του ιδιωτικού του κλειδιού. Έτσι, υπολογίζει

$$c^d \equiv (m^e)^d \equiv m \mod n.$$

Έχοντας το m, μπορεί να ανακτήσει το αρχικό κείμενο M με τη χρήση του σχήματος συμπληρώματος.

3.2.4 Ασφάλεια

Το RSA κρυπτοσύστημα βασίζει την ασφάλειά (security) του στη δυσκολία παραγοντοποίησης ακέραιων. Αναφέρθηκε παραπάνω η επιτυχημένη προσπάθεια παραγοντοποίησης ενός 129-ψήφιου RSA modulus. Σήμερα ένα RSA modulo με 512 bits, ή με περίπου 155 ψηφία θα ήταν εφικτό να παραγοντοποιηθεί, και στην πραγματικότητα έχει όντως γίνει. Τον Αύγουστο του 1999, μία ομάδα που συμπεριλάμβανε τον Arjen Lenstra και τον Peter Montgomery παραγοντοποίησαν ένα RSA modulus των 512 bits χρησιμοποιώντας τον Number Field Sieve με χρήση εκατοντάδων υπολογιστών και σε χρόνο περίπου 7 μηνών. Το 2015, παραγοντοποιήθηκε 512-bit RSA modulus σε 4 ώρες, με χρήση υπηρεσιών της Amazon και υλοποίηση του GNFS [16]. Επομένως, είναι απαγορευτική η χρήση τόσο μικρών αριθμών στη κρυπτογραφία. Για εταιρική χρήση, προτείνεται η χρήση πρώτων παραγόντων των 1024 bits και 2048 bits για περισσότερη ασφάλεια. Αυτές οι προτάσεις λαμβάνουν υπόψη τις πιθανές εξελίξεις στις τεχνικές παραγοντοποίησης και την αύξηση της ταχύτητας επεξεργασίας. Ο Riesel [25] δείχνει ότι είναι δυνατόν να δημιουργηθεί ένας αλγόριθμος που παραγοντοποιεί ακέραιους αριθμούς σε σχεδόν πολυωνυμικό χρόνο, οπότε υπάρχει σίγουρα περιθώριο για βελτιώσεις. Ωστόσο, αν ένας κβαντικός υπολογιστής κατασκευαστεί ποτέ με επαρκή αριθμό qubits, ο Peter Shor έχει ανακαλύψει έναν αλγόριθμο για παραγοντοποίηση ακέραιων αριθμών σε πολυωνυμικό χρόνο [28]. Τότε το RSA θα έπρεπε να αποσυρθεί, καθώς για να είναι ασφαλές, το RSA-modulus θα έπρεπε να είναι τόσο μεγάλο που δεν θα ήταν βολικό.

ΚΕΦΑΛΑΙΟ 4

Quadratic Sieve

4.1 Η Βασική Ιδέα

Έστω n ο αριθμός που πρέπει να παραγοντοποιηθεί. Ο QS [12] προσπαθεί να βρει δύο αριθμούς x και y τέτοιους ώστε $x \not\equiv y \mod n$ και $x^2 \equiv y^2 \mod n$, βασισμένος στην ιδέα του Fermat, που αναφέρθηκε παραπάνω. Αυτό σημαίνει πως $(x-y)(x+y) \equiv 0 \mod n$ και απλά υπολογίζουμε $\gcd(x-y, n)$ (μέγιστο κοινό διαιρέτη) χρησιμοποιώντας τον Ευκλείδειο αλγόριθμο ώστε να δούμε αν είναι ένας μη τετριμμένος διαιρέτης.

Πρώτα ορίζουμε,

$$Q(x) = (x + \lfloor \sqrt{n} \rfloor)^2 - n.$$

Υπολογίζουμε τα $\{Q(i) : i = 1, 2, \cdots m\}$, για m που θα προσδιοριστεί αργότερα. Έστω, $x_1, \cdots, x_k \in \{1, 2, \cdots, m\}$, τέτοια ώστε

$$Q(x_1)Q(x_2)\cdots Q(x_k) \equiv Y^2 \mod n. \tag{2}$$

Από την άλλη όμως έχουμε $Q(x_i) = (x_i + \lfloor \sqrt{n} \rfloor)^2 - n \equiv \tilde{x_i}^2 \mod n$ όπου

$$\tilde{x_i} = x_i + \lfloor \sqrt{n} \rfloor.$$

Θέτουμε $\tilde{x_1} \cdots \tilde{x_k} = X$. Οπότε από την (2) έχουμε $X^2 \equiv Y^2 \mod n$. Αν ισχύει $X \equiv \pm Y \mod n$ τότε $\gcd(X - Y, n)$ μας δίνει ένα πρώτο παράγοντα του n. Το τελευταίο ισχύει με πιθανότητα $\geq 1/2$.

4.2 Factor Base

Πρέπει να βρεθεί ένας αποτελεσματικός τρόπος για να καθορίζονται τα x_i, ώστε το γινόμενο των $Q(x_i)$ να είναι τετράγωνο. Για να ελεγχθεί αν το γινόμενο (2) είναι τετράγωνο, οι εκθέτες των πρώτων παραγόντων του γινομένου πρέπει να είναι όλοι άρτιοι. Άρα πρέπει να παραγοντοποιηθούν όλα τα $Q(x_i)$. Επομένως, είναι προτιμότερο να παραγοντοποιούνται σε ένα σύνολο μικρών πρώτων αριθμών, συμπεριλαμβανομένου του -1. Το σύνολο αυτό ονομάζεται βάση παραγόντων, την οποία θα συμβολίζουμε ως $\mathcal{S}$ (factor base).

Έστω ο $x \in [-M, M]$ για κάποιο $M > 0$, και κάποιος πρώτος p διαιρεί το Q(x), τότε $(x + \lfloor \sqrt{n} \rfloor)^2 \equiv n \mod p$. Άρα ο n είναι τετραγωνικό υπόλοιπο $\mod p$. Επομένως, οι πρώτοι αριθμοί στη βάση μας (factor base) πρέπει να είναι τέτοιοι ώστε το σύμβολο του Legendre $\left(\frac{n}{p}\right) = 1$. Επίσης, πρέπει οι πρώτοι αυτοί να είναι μικρότεροι από ένα όριο B, το οποίο εξαρτάται από το μέγεθος του n και θα προσδιοριστεί αργότερα. Έτσι, η βάση παραγόντων είναι $\mathcal{S} = \{-1, p_1, \cdots, p_t\}$.

4.3 Sieving

Αφού δημιουργήσουμε τη βάση μας (factor base), παίρνουμε τα x από το διάστημα κοσκινίσματος $[-M, M]$ και υπολογίζουμε τα $Q(x)$. Κατόπιν, ελέγχουμε αν παραγοντοποιούνται πλήρως με τη βάση μας. Αν ένα $Q(x_i)$ παραγοντοποιείται λέμε ότι είναι B-smooth. Αν δεν παραγοντοποιείται τότε το αφήνουμε και προχωράμε στο επόμενο στοιχείο του διαστήματος κοσκινίσματος. Αν η βάση ωστόσο είναι μεγάλη, είναι άκρως μη αποτελεσματικό να ελέγχουμε ένα στοιχείο τη φορά για να δούμε αν παραγοντοποιείται με τη βάση. Αντιθέτως μπορούμε να δουλέψουμε με ολόκληρο το διάστημα κοσκινίσματος. Αν δουλεύουμε παράλληλα, κάθε επεξεργαστής θα δούλευε πάνω σε διαφορετικό υποδιάστημα.

4.4 Smoothness

Ένας αριθμός m χαρακτηρίζεται λείος (smooth) αν όλοι οι πρώτοι παράγοντές του είναι μικροί. Συγκεκριμένα, λέμε ότι ο m είναι B-smooth αν όλοι οι πρώτοι παράγοντές του είναι μικρότεροι από B. Μία πρώτη παρατήρηση είναι ότι αν ένας αριθμός στην ακολουθία μας δεν είναι smooth, τότε είναι απίθανο να χρησιμοποιηθεί σε μία υποακολουθία με γινόμενο τετράγωνο.

Λήμμα 4.4.1 *Αν $m_1, m_2, \cdots, m_k$ είναι θετικοί B-smooth ακέραιοι, και αν $k > \pi(B)$, όπου $\pi(B)$ δηλώνει τον αριθμό των πρώτων στο διάστημα $[1, B]$, τότε κάποιο μη κενό υποσύνολο του $\{m_1, \cdots, m_k\}$ έχει γινόμενο που είναι τετράγωνο.*

Απόδειξη Από "Smooth numbers and the quadratic sieve" [22]. Για κάποιο B-smooth αριθμό m, παίρνουμε το διάνυσμα των εκθετών $\vec{v}(m)$. Αν το m έχει την παραγοντοποίηση των πρώτων

$$m = \prod_{i=1}^{\pi(B)} {p_i}^{u_i}$$

όπου p_i είναι ο i-οστός πρώτος αριθμός και κάθε εκθέτης u_i είναι ένας μη αρνητικός ακέραιος, τότε $\vec{v}(m) = (v_1, v_2, \cdots, v_{\pi_{(B)}})$. Τότε ένα υποσύνολο $m_{i_1}, \cdots, m_{i_t}$ έχει τετραγωνικό γινόμενο, αν και μόνο αν $\vec{v}(m_{i_1}) + \cdots + \vec{v}(m_{i_t})$ έχει μόνο άρτια στοιχεία. Αυτό συμβαίνει, αν και μόνο αν το άθροισμα των διανυσμάτων είναι το μηδενικό διάνυσμα $\bmod 2$. Τώρα ο χώρος των διανυσμάτων ${F_2}^{\pi(B)}$, όπου F_2 είναι το πεπερασμένο σώμα με 2 στοιχεία, έχει διαστάσεις $\pi(B)$. Έχουμε $k > \pi(B)$ διανύσματα. Επομένως αυτή η ακολουθία διανυσμάτων είναι γραμμικά εξαρτώμενη σε αυτό στο χώρο διανυσμάτων. ■

Ένας αλγόριθμος που μας δίνει αυτό το υποσύνολο είναι για παράδειγμα η Γκαουσιανή απαλοιφή (Gaussian reduction of a matrix) [10]. Με μία συλλογή smooth αριθμών, δημιουργούμε τα διανύσματα των εκθετών $\bmod 2$, και έπειτα χρησιμοποιούμε Γκαουσιανή απαλοιφή για να βρούμε ένα μη κενό υποσύνολο με άθροισμα το μηδενικό διάνυσμα $\bmod 2$.

Να σημειωθεί πως η γνώση του ολοκληρωμένου διανύσματος εκθετών ενός αριθμού m ισούται με τη γνώση της παραγοντοποίησης πρώτων διαιρετών του m (complete prime factorization of m).

4.5 Η επιλογή του B: Ένα πρόβλημα βελτιστοποίησης

Φυσικά το όριο B πρέπει να οριστεί κατάλληλα, ώστε ο χρόνος παραγοντοποίησής του n να είναι ελάχιστος. Αν το B επιλεγεί πολύ μικρό, τότε δεν χρειάζεται να βρούμε πολλούς B-smooth αριθμούς στο "κόσκινο" και ο πίνακας θα είναι μικρός. Αλλά οι B-smooth αριθμοί με πολύ μικρή B τιμή είναι πολύ αραιά κατανεμημένοι μεταξύ των φυσικών αριθμών και επομένως μπορεί να χρειαστεί να διασχίσουμε μία πολύ μεγάλη ακολουθία x τιμών για να πάρουμε έστω και

μία B-smooth τιμή $x^2 - n$, πόσο μάλλον το απαιτούμενο πλήθος. Από την άλλη, αν η τιμή B είναι πολύ μεγάλη, τότε το πλήθος των B-smooth αριθμών θα είναι μεγαλύτερο και δεν θα δυσκολευτούμε να τους βρούμε στην πολυωνυμική ακολουθία $x^2 - n$. Έτσι όμως, απαιτείται η αποθήκευση περισσότερων αριθμών, το οποίο απαιτεί πολλή μνήμη στον υπολογιστή.

Για να λύσουμε αυτό το πρόβλημα πρέπει να μετράμε με κάποιο τρόπο τη πιθανότητα η τιμή $x^2 - n$ να είναι B-smooth. Αυτό είναι ένα πολύ δύσκολο πρόβλημα στη θεωρία αριθμών. Ορίζουμε,

$$B \leftarrow \lfloor L(n)^{1/2} \rfloor, \text{ όπου } L(x) = e^{\sqrt{\log x * \log\log x}}$$

4.6 Παραλληλοποίηση

Αν p είναι ένας πρώτος παράγοντας του $Q(x)$, τότε $p|Q(x+p)$. Αν $x \equiv y \mod p$, τότε $Q(x) \equiv Q(y) \mod p$. Έτσι για κάθε p αρκεί να λύσουμε

$$Q(x) = s^2 \equiv 0 \mod p, \text{ για } x \in \mathbb{Z}_p.$$

Αυτό μπορούμε να το κάνουμε χρησιμοποιώντας τον αλγόριθμο Shanks-Tonelli (δείτε [15, 2.1 Tonelli and Shanks Algorithm]). Βρίσκουμε δύο λύσεις s_{1_p} και $s_{2_p} = p - s_{1_p}$. Τα $Q(x_i)$ με x_i από το διάστημα κοσκινίσματος διαιρούνται με p για κάποιο ακέραιο k.

Υπάρχουν διάφοροι τρόποι για να γίνει το "κοσκίνισμα" από εδώ και πέρα. Ένας τρόπος είναι να επιλεγεί ένα υποδιάστημα, και να τοποθετηθεί το $Q(x_i)$ σε έναν πίνακα για κάθε x_i του υποδιαστήματος. Για κάθε p, ξεκινάμε από s_{1_p} και s_{2_p} και διαιρούμε με τη μεγαλύτερη δυνατή δύναμη του p για κάθε στοιχείο του πίνακα, καταγράφοντας τις αντίστοιχες δυνάμεις $\mod 2$ του p σε ένα διάνυσμα. Θα δημιουργηθεί ένα διάνυσμα για κάθε παραγοντοποιήσιμο $Q(x_i)$ και κάθε στοιχείο του διανύσματος αντιστοιχεί σε ένα μοναδικό πρώτο από τη βάση. Το διάνυσμα των δυνάμεων μπορεί να μπει σε έναν πίνακα A. Επαναλαμβάνουμε την διαδικασία μέχρι να έχουμε αρκετές εισόδους στον A.

Ένας άλλος τρόπος είναι, αντί να εργαζόμαστε με τις τιμές του $Q(x)$ πάνω σε κάποιο υποδιάστημα, να καταγράφουμε τον αριθμό των bits του $Q(x_i)$ σε έναν πίνακα.

4.7 Το βήμα της γραμμικής άλγεβρας

Για κάθε αριθμό x_i στο διάστημα κοσκινίσματος $[-M, M]$, υπολογίζουμε τα $Q(x_i) = (x_i + \lfloor \sqrt{n} \rfloor)^2 - n$ των οποίων όλοι οι διαιρέτες ανήκουν στη βάση παραγόντων $\mathcal{S}$. Έστω S_1 αυτό το σύνολο. Σταματάμε όταν βρεθούν $t+2$ τέτοια $Q(x_i)$, όπου $t+1$ είναι το πλήθος των στοιχείων της βάσης παραγόντων $S = \{-1, p_1, \cdots, p_t\}$.

Για κάθε στοιχείο $y \in S_1$, δημιουργούμε ένα διάνυσμα $\vec{a} = (a_1, \cdots, a_{t+1})$ με τους εκθέτες $\mod 2$ των πρώτων παραγόντων του y. Τα διανύσματα αυτά τοποθετούνται σε έναν πίνακα A. Η κάθε γραμμή του πίνακα αναπαριστά ένα $Q(x_i)$, και οι στήλες αναπαριστούν τους εκθέτες $\mod 2$ των πρώτων αριθμών από τη βάση. Πρέπει το γινόμενο των $Q(x_i)$ να είναι τέλειο τετράγωνο, επομένως θέλουμε το άθροισμα των εκθετών του κάθε πρώτου από τη βάση να είναι άρτιο, και άρα ισότιμο με $0 \mod 2$.

Έτσι, αν $\vec{a_i}$ είναι η γραμμή του πίνακα A που αντιστοιχεί στο $Q(x_i)$, τότε θέλουμε

$$\vec{a_1}e_1 + \vec{a_2}e_2 + \cdots + \vec{a_k}e_k \equiv \vec{0} \mod 2, \quad k = t+1$$

όπου e_i είναι είτε 0 είτε 1. Έστω S_2 το σύνολο των διανυσμάτων $\vec{a}$. Αυτό σημαίνει πως πρέπει να λύσουμε το γραμμικό σύστημα,

$$\vec{e}A = \vec{0} \mod 2,$$

όπου

$$\vec{e} = (e_1, e_2, \cdots, e_k).$$

Βρίσκουμε μία μη-μηδενική λύση του συστήματος. Η λύση αυτή θα μας δώσει ένα γραμμικά εξαρτόμενο υποσύνολο του S_2. Π.χ. αν $\vec{a}_2 + \vec{a}_5 = 0 \mod 2$, τότε η 2η και 5η γραμμή είναι γραμμικά εξαρτημένες.

Αν $Q(x_1), \cdots, Q(x_r)$ αντιστοιχούν στις γραμμές του πίνακα οι οποίες ανήκουν στην λύση, τότε ορίζουμε ως

$$x^2 = Q(x_1) \cdots Q(x_r) \mod n$$

και

$$z = x_1 \cdots x_r \mod n.$$

Υπολογίζουμε τον $d \leftarrow \gcd(x - z, n)$ και ελέγχουμε αν είναι παράγοντας του n.

4.8 Αλγόριθμος

Βασιστήκαμε στον αλγόριθμο από το βιβλίο "Handbook of Applied Crytpography" [3]. Έστω n ο ακέραιος προς παραγοντοποίηση. Επίσης, $m = \lfloor \sqrt{n} \rfloor$, και θεωρούμε το πολυώνυμο $q(x) = (x+m)^2 - n$. Παρατηρείται πως

$$q(x) = x^2 + 2mx + m^2 - n \approx x^2 + 2mx,$$

το οποίο είναι μικρό συγκριτικά με το n, αν η απόλυτη τιμή του x είναι μικρή. Ο QS διαλέγει ένα $a_i = (x+m)$ και ελέγχει αν το $b_i = (x+m)^2 - n$ είναι p_t-smooth. Σημειώνεται πως

$$a_i{}^2 = (x+m)^2 \equiv b_i \mod n.$$

Επίσης, σημειώνεται πως αν ένας πρώτος αριθμός p διαιρεί το b_i τότε

$$(x+m)^2 \equiv n \mod p,$$

και επομένως το n είναι τετραγωνικό υπόλοιπο p. Άρα, η βάση παραγοντοποίησης χρειάζεται να περιέχει μόνο τους πρώτους p, για τους οποίους το σύμβολο του Legendre $\left(\frac{n}{p}\right)$ ισούται με 1. Επιπλέον, καθώς το b_i μπορεί να είναι αρνητικό, το -1 συμπεριλαμβάνεται στη βάση παραγοντοποίησης.

Αλγόριθμος 4.8.1 *Quadratic sieve*
Είσοδος: *ένας σύνθετος ακέραιος n που δεν είναι δύναμη πρώτου αριθμού*
Έξοδος: *ένας μη τετριμμένος παράγοντας d του n*

1. Επιλογή της βάσης παραγόντων $S = \{p_1, p_2, \cdots, p_t\}$, όπου $p_1 = -1$ και $p_j (j \geq 2)$ είναι ο $(j-1)^{στος}$ πρώτος p για τον οποίο το n είναι τετραγωνικό υπόλοιπο modulo p
2. Υπολογισμός του $m = \lfloor \sqrt{n} \rfloor$
3. Συλλογή t+1 ζευγαριών (a_i, b_i). Οι τιμές x επιλέγονται με τη σειρά $0, \pm 1, \pm 2, \cdots$ Ορίζουμε $i \leftarrow 1$. Όσο $i \leq t+1$, επανάλαβε:

   3.1. Υπολόγισε $b = q(x) = (x+m)^2 - n$, και έλεγξε χρησιμοποιώντας trial division από τα στοιχεία του S αν το b είναι p_t-smooth. Αν όχι, διάλεξε ένα καινούριο x και επανέλαβε το βήμα 3.1

3.2. Αν το b είναι p_t-smooth, και $b \prod_{j=1}^{t} p_j^{e_{ij}}$, τότε όρισε $a_i \leftarrow (x+m), b_i \leftarrow b$, και $u_i = (u_{i_1}, u_{i_2}, \cdots, u_{i_t})$, όπου $u_{ij} = e_{ij} \mod 2$ για $1 \leq j \leq t$

3.3. $i \leftarrow i+1$

4. Με τη χρήση γραμμικής άλγεβρας πάνω στο $\mathbb{Z}_2$, να βρεθεί ένα μη-κενό υποσύνολο $T \subseteq \{1, 2, \cdots, t+1\}$, τέτοιο ώστε $\sum_{i \in T} u_i = 0$

5. Υπολόγισε $x = \prod_{i \in T} a_i \mod n$

6. Για κάθε $j, 1 \leq j \leq t$, υπολόγισε $l_j = (\sum_{i \in T} e_{ij})/2$

7. Υπολόγισε $y = \prod_{j=1}^{t} p_j^{l_j} \mod n$

8. Αν $x \equiv \pm y \mod n$, τότε βρες ένα άλλο μη-κενό υποσύνολο $T \subseteq \{1, 2, \cdots, t+1\}$, τέτοιο ώστε $\sum_{i \in T} u_i = 0$, και πάνε στο βήμα 5. (Στην απίθανη περίπτωση που ένα τέτοιο υποσύνολο T δεν υπάρχει, αντικατέστησε μερικά από τα (a_i, b_i) ζευγάρια με καινούρια (βήμα 3), και πάνε στο βήμα 4)

9. Υπολόγησε $d = \gcd(x-y, n)$ και return(d)

Παράδειγμα

Παράδειγμα εκέτλεσης του Quadratic sieve αλγόριθμου, για να βρεθεί ένας μη τετριμμένος διαιρέτης του $n = 25651$.

1. Επιλογή της βάσης παραγόντων $S = \{-1, 2, 3, 5, 17, 19, 23\}$ μεγέθους $t = 7$. (Οι 7, 11 και 13 δεν ανήκουν στη βάση S καθώς $\left(\frac{n}{p}\right) = -1$ για αυτούς τους πρώτους αριθμούς)

2. Υπολογίζουμε $m = \lfloor \sqrt{25651} \rfloor = 160$

3. Ακολουθεί ο πίνακας για τις πρώτες $t+1$ τιμές του x για τις οποίες το $q(x)$ είναι 23-smooth

i	x	q(x)	factorization of q(x)	a_i	u_i
1	0	-51	$-3 \cdot 17$	160	(1, 0, 1, 0, 1, 0, 0)
2	1	270	$2 \cdot 3^3 \cdot 5$	161	(0, 1, 1, 1, 0, 0, 0)
3	3	918	$2 \cdot 3^3 \cdot 17$	163	(0, 1, 1, 0, 1, 0, 0)
4	-9	-2850	$-2 \cdot 3 \cdot 5^2 \cdot 19$	151	(1, 1, 1, 0, 0, 1, 0)
5	10	3249	$3^2 \cdot 19^2$	170	(0, 0, 0, 0, 0, 0, 0)
6	-11	-3450	$-2 \cdot 3 \cdot 5^2 \cdot 23$	149	(1, 1, 1, 0, 0, 0, 1)
7	12	3933	$3^2 \cdot 19 \cdot 23$	172	(0, 0, 0, 0, 0, 1, 1)
8	-14	-4335	$-3 \cdot 5 \cdot 17^2$	146	(0, 0, 1, 1, 0, 0, 0)

4. Προκύπτει πως $\vec{u_4} + \vec{u_6} + \vec{u_7} = 0$, δηλαδή $T = \{4, 6, 7\}$

5. Υπολογίζεται $x = (a_4 a_6 a_7 \mod n) = 22178$

6. Υπολογίζεται $l_1 = 1, l_2 = 1, l_3 = 2, l_4 = 2, l_5 = 0, l_6 = 1, l_7 = 1$

7. Υπολογίζεται $y = -2 \cdot 3^2 \cdot 5^2 \cdot 19 \cdot 23 \mod n = 8558$

8. Ισχύει, $22178 \not\equiv \pm 8558 \mod n$, άρα υπολογίζεται $\gcd(x+y, n) = \gcd(30736, 25651) = 113$. Επομένως, δύο μη τετριμμένοι διαιρέτες του 25651 είναι το 113 και το 227.

Αναφορές

[1] Euclid . *The Elements of Geometrie*. Henry Billingsley, 1st edition, 1570.

[2] Leonard M. Adleman, Carl Pomerance, and Robert S. Rumely. On distinguishing prime numbers from composite numbers. *Annals of Mathematics*, 117(1):173–206, 1983.

[3] Paul C. van Oorschot Alfred J. Menezes and Scott A. Vanstone CRC Press. *Handbook of Applied Cryptography*. 5 edition, 1996.

[4] P. A. Cataldi. *Trattato de nvmeri perfetti di Pietro Antonio Cataldo*, volume 1. Presso gli heredi di Giouanni Rossi, 1603.

[5] Richard Crandall and Carl Pomerance. *Prime numbers. A Computational Perspective*. Springer-Verlag New York, 2005.

[6] K. Draziotis. *Textbook for the cryptography course*. 2021.

[7] Laurent Fousse, Guillaume Hanrot, Vincent Lefèvre, Patrick Pélissier, and Paul Zimmermann. MPFR: A Multiple-Precision Binary Floating-Point Library With Correct Rounding. Research Report RR-5753, INRIA, 2005.

[8] Carl Friedrich Gauss and William C. Waterhouse. Disquisitiones arithmeticae. page 472. Springer-Verlag New York, 1986.

[9] T. Granlund and Gmp Development Team. *GNU MP 6.0 Multiple Precision Arithmetic Library*. Samurai Media Limited, 2015.

[10] Joseph F. Grcar. Mathematicians of gaussian elimination. *Notices of the American Mathematical Society*, 58:782–792, 2011.

[11] Andreas M. Hinz, Sandi Klavzar, Uros Milutinovic, and Ciril Petr. *The Tower of Hanoi - Myths and Maths*. Birkhäuser Basel, 2013.

[12] Eric Landquist. The Quadratic Sieve Factoring Algorithm. MATH 488: Cryptographic Algorithms, 2001.

[13] D. H. Lehmer and R. E. Powers. On factoring large numbers. *Bulletin of the American Mathematical Society*, 37(10):770 – 776, 1931.

[14] Calvin T. Long. *Elementary Introduction to Number Theory*. [D. C. Heath and Company], 2 edition, 1972.

[15] A. Uma Maheswari and Prabha Durairaj. An algorithm to find square roots of quadratic residues modulo p. *Global Journal of Pure and Applied Mathematics*, 13(4):1223–1239, 2017.

[16] Aleksandra V. Markelova. Vulnerability of rsa algorithm. Information Security Department Bauman Moscow State Technical University, 2017.

[17] Richard A. Mollin. A brief history of factoring and primality testing b. c. (before computers). *Mathematics Magazine*, 75(1):18–29, February 2002.

[18] Michael A. Morrison and John Brillhart. A method of factoring and the factorization of f7. *MATHEMATICS OF COMPUTATION*, 29(129):183–205, January 1975.

[19] Pascal Ochem and Michaël Rao. *Mathematics of Computation*, 81(279):1869–1877, 2012.

[20] Carl Pomerance Paul Erdős and Eric Schmutz. Carmichael's lambda function. *Acta Arithmetica*, 58:363–385, 1991.

[21] C. Pomerance. Computational methods in number theory: Part 1. *Mathematical Centre Tracks 154*, pages 89–139, 1982. [PDF].

[22] C. Pomerance. Smooth numbers and the quadratic sieve. *Algorithmic Number Theory - MSRI Publications*, 44:72, 2008.

[23] Carl Pomerance. A tale of two sieves. *NOTICES AMER. MATH. SOC*, 43:1473–1485, 1996.

[24] G. L. Cohen R. P. Brent and H. J. J. te Riele. *Mathematics of Computation*, 57(196):857–868, 1991.

[25] Hans Riesel. *Prime Numbers and Computer Methods for Factorization*. Birkhäuser Basel, 2 edition, 1994.

[26] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 1978.

[27] Jr. Samuel S. Wagstaff. *The Joy of Factoring*, volume 68 of *Student Mathematical Librabry*. American Mathematical Society, 2013.

[28] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[29] Martyn Shuttleworth. Pythagoras. Explorable. Think Outside of the box, 2010. Retrieved Sep 12, 2021, [Article].

[30] L. E. (Laurence E.) Sigler. *Fibonacci's Liber Abaci: A Translation into Modern English of Leonardo Pisano's Book of Calculation*. Sources and studies in the history of mathematics and physical sciences. 2002.

[31] Singh Simon. *The code book : the science of secrecy from ancient Egypt to quantum cryptography*. Anchor, 1999.

[32] Jonathan Sorenson. An introduction to prime number sieves. Department of Computer Sciences University of Wisconsin-Madison, January 1990. [PDF].

[33] J. Voight. Perfect numbers: An elementary introduction. 2000. [PDF].

[34] Eric W. Weisstein. Euler's totient theorem. From MathWorld–A Wolfram Web Resource. [Link].