Scientific Research Publishing

Advances in Pure Mathematics, 2025, 15(9), 629-642
https://www.scirp.org/journal/apm
ISSN Online: 2160-0384
ISSN Print: 2160-0368

Kernel Factor Pairs for Semiprime Factorization

Han-Lin Li[1], Shu-Cherng Fang[2], Way Kuo[3], Nianrui Lin[4]

[1]Department of Computer Science, City University of Hong Kong, Hong Kong, China
[2]Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, USA
[3]Hong Kong Institute for Advanced Study and Department of Data Science, City University of Hong Kong, Hong Kong, China
[4]Department of Systems Engineering, City University of Hong Kong, Hong Kong, China
Email: nianrui.lin@my.cityu.edu.hk

How to cite this paper: Li, H.-L., Fang, S.-C., Kuo, W. and Lin, N.R. (2025) Kernel Factor Pairs for Semiprime Factorization. *Advances in Pure Mathematics*, **15**, 629-642.
https://doi.org/10.4236/apm.2025.159032

Received: August 26, 2025
Accepted: September 19, 2025
Published: September 22, 2025

Abstract

We show that any semiprime number can be factorized as the product of two prime numbers in the form of a kernel factor pair of two out of 48 root numbers. Specifically, each natural number without factors of 2, 3, 5 and 7 can be traced back to one unique number of a total of 48 root numbers falling in $[11, 220]$ in periods of length 210. Unlike the commonly used sieve-based methods, under no preconditions, will the proposed kernel-factor-pair-based algorithm be guaranteed to successfully factorize any given semiprime α by searching over $1/2\log\alpha$ binary variables. The proposed method is well structured for factorization in breaking RSA encryption and is readily applicable for parallel computation.

Keywords

Semiprimes, Factorization, Factor-Pairs Table

1. Introduction

One of the greatest theorems of mathematics states that any natural number can be represented uniquely as a product of primes [1] [2]. Over years, prime numbers have been an important topic of number theory [3] [4]. Primes exhibit numerous interesting properties and are widely used in many fields in recent years, such as data science, cryptography [3], color theory [5] and reliability design [4].

Semiprime factorization methods are to decompose a semiprime number into its two prime factors, which serve as the key to cryptography including RSA public key encryption and RSA digital signature [6]-[12].

To the best of our knowledge, almost all available semiprime factorization

methods are sieve-based, which can be classified into the following three categories [13] [14]:

- Category 1: The Trivial Sieve Method [15] comes from the well-known Aristotle sieve technique of utilizing exhaustive brute force to factorize a semiprime number. The Wheel Sieve Method [16] [17] identifies potential prime numbers starting from a small wheel and expanding to larger ones which improves the Trivial Sieve Method.
- Category 2: The Quadratic Sieve Method [18] includes the Pollard's Rho [19] and Lenstra Elliptic Curve techniques [20]-[22].
- Category 3: The General Sieve Method [8] [23] includes the General Number Field Sieve (GNFS) and Schnorr methods [24]-[26].

Each of the above-mentioned sieve-based methods may suffer from the following limitations:

i) They often employ heuristic techniques to factorize a semiprime number, guaranteeing no convergence to a feasible solution. For example, the GNFS method depends on the specification of two polynomial functions [13] [23]. Similarly, since Pollard's Rho method [19] is a probabilistic algorithm, it may not reach a convergent solution.

ii) Some require a good deal of pre-known information in the factorization process. For example, the Trivial Sieve method needs to know all prime numbers smaller than $\sqrt{\alpha}$ beforehand to factorize a semiprime number α, while the GNFS method likely uses more than half of its computing time to detect whether a number is smooth or not [13].

iii) Sieve methods often induce high space complexity. For example, to factorize a semiprime α, the Trivial Sieve method needs to reserve a large memory space to store $\sqrt{\alpha}$ pre-known primes for computations. The space complexity gets higher as α grows larger.

iv) Often needed by some of the methods are the special structure for a given semiprime. For example, the GNFS method can only factorize a semiprime with limited smoothness values [13]. Similarly, Pollard's Rho [19] and Elliptic Curve [22] methods are special purpose algorithms, whose running time depends on the size of the smaller prime factor of a given semiprime number.

v) Wheel Sieve is an improvement of the method of Sieve of Eratosthenes. Such methods are intended to construct a prime list using the first few primes. Pritchard's method [16], proposed in 1982, is one of the early works that explored the concept of wheels for factorization. The idea has been elaborated by others such as [17] into various forms. The wheel-sieve-based algorithms in general suffer from the difficulties of clearly identifying or distinguishing primes and composites. Consequently, the solutions usually get fuzzy when the involved numbers become large.

Recently, Li, Fang and Kuo discovered the Periodic Table of Primes (PTP) [27] which reveals all primes and composites with no factors of 2, 3, 5 and 7 exclusively in a compact table. One unique feature of the PTP is to identify the cyclic appearance of composites through the Cyclic Table of Composites (CTC) [27], and then

differentiate primes and composites from a feasible set like that used in the wheel sieve, but subject to no preconditions. The PTP clearly shows that the ratio of the number of primes to that of the composites approaches zero when the table size grows. An early version of the SSRN archive [28] also provided examples to highlight the potential of using the PTP for factorization.

To overcome the general difficulties and limitations of sieve-based factoring methods, we now propose an innovative scheme for semiprime factorization. By extending our previous work of building the PTP [27], we introduce the concepts of "factor pairs" and "kernel factor pairs" for fast factorization. A short table of all factor pairs listed in 48 rows and 28 columns is constructed to steer the factorization of any given semiprime under no preconditions. With the Factor-Pairs Table (**Table 1**) in hand, there is no need to know, calculate, and store any primes that come before $\sqrt{\alpha}$ for a given semiprime α. This may significantly reduce the space complexity issue faced by some sieve-based methods. The number of potential factor pairs involved in selecting a kernel factor pair corresponding to a given semiprime α naturally leads to the design of a highly parallelable algorithm for semiprime factorization.

Table 1. Factor-pairs table (FT_{210}).

i	r_i	Pair 1	2	3	4	5	6	7	8	9	10	11	12	13	14
		$q_{j\|i}$													
		$q_{\hat{j}\|i}$						$q_{j\|i}$	$q_{\hat{j}\|i}$						
1	11	11 211	13 17	19 89	23 37	29 109	31 41	43 137	47 103	53 127	59 139	61 131	67 113	71 151	73 167
2	13	11 173	13 211	17 149	19 67	23 101	29 167	31 163	37 199	41 113	43 181	47 179	53 131	59 107	61 193
3	17	11 97	13 179	17 211	19 23	29 73	31 197	37 131	41 67	43 59	47 121	53 139	61 107	71 157	79 173
4	19	11 59	13 163	17 137	19 211	23 83	29 131	31 109	37 97	41 149	43 103	47 197	53 143	61 169	67 157
5	23	11 193	13 131	17 199	19 167	23 211	29 37	31 143	41 103	43 191	47 139	53 151	59 157	61 83	67 179
6	29	11 79	13 83	17 187	19 101	23 193	29 211	31 89	37 137	41 139	43 113	47 157	53 163	59 61	67 107
7	31	11 41	13 67	17 113	19 79	23 47	29 59	31 211	37 103	43 157	53 167	61 121	71 101	73 127	83 137
8	37	11 137	13 19	17 101	23 29	31 157	37 211	41 47	43 79	53 179	59 143	61 97	67 151	71 107	73 199
9	41	11 61	13 197	17 163	19 179	23 157	29 139	31 191	37 143	41 211	43 167	47 193	53 187	59 79	67 173
10	43	11 23	13 181	17 89	19 157	29 197	31 103	37 109	41 83	43 211	47 59	53 191	61 73	67 79	71 113
11	47	11 157	13 149	17 151	19 113	23 139	29 103	31 137	37 41	43 89	47 211	53 199	59 193	61 197	67 101
12	53	11 43	13 101	17 139	19 47	23 121	29 67	31 83	37 149	41 73	53 211	59 97	61 173	71 193	79 107
13	59	11 139	13 53	17 127	19 191	23 103	29 31	37 47	41 109	43 143	59 211	61 149	67 167	71 199	73 113
14	61	11 101	13 37	17 53	19 169	23 167	29 89	31 151	41 191	43 187	47 113	59 179	61 211	67 73	71 131
15	67	11 197	13 199	17 41	19 103	23 149	29 53	31 97	37 121	43 109	47 131	59 83	61 187	67 211	71 137
16	71	11 121	13 167	17 103	19 59	23 67	29 169	31 131	37 53	41 181	43 197	47 73	61 101	71 211	79 179
17	73	11 83	13 151	17 29	19 37	23 131	31 43	41 53	47 149	59 197	61 163	67 139	71 143	73 211	79 187

Continued

18	79	11 179	13 103	17 17	19 181	23 113	29 191	31 199	37 127	41 89	43 163	47 167	53 53	59 101	61 139
19	83	11 103	13 71	17 79	19 137	23 31	29 97	37 59	41 43	47 109	53 61	67 89	73 191	83 211	101 163
20	89	11 199	13 23	17 67	19 71	29 61	31 179	37 167	41 79	43 173	47 127	53 73	59 151	83 163	89 211
21	97	11 47	13 169	17 191	19 193	23 59	29 83	31 37	41 197	43 139	53 89	61 67	71 167	73 79	97 211
22	101	11 181	13 137	17 43	19 149	23 187	29 199	31 71	37 173	41 151	47 163	53 97	59 169	61 191	67 83
23	103	11 143	13 121	17 179	19 127	23 41	29 47	31 193	37 139	43 61	53 101	59 137	67 199	71 173	73 151
24	107	11 67	13 89	17 31	19 83	23 169	29 163	37 71	41 187	43 149	47 181	53 109	59 73	61 167	79 113
25	109	11 29	13 73	17 167	19 61	23 23	31 139	37 37	41 59	43 193	47 47	53 113	67 127	71 179	79 121
26	113	11 163	13 41	17 19	23 151	29 127	31 173	37 179	43 71	47 199	53 121	59 187	61 143	67 149	73 131
27	121	11 11	13 187	17 143	19 139	23 197	29 149	31 31	37 43	41 131	47 83	53 137	59 59	61 181	67 193
28	127	11 107	13 139	17 131	19 73	23 179	29 113	31 187	37 151	41 167	43 169	47 101	53 149	59 173	61 157
29	131	11 31	13 107	17 193	19 29	23 97	37 83	41 121	43 47	53 157	59 109	61 71	67 143	73 137	79 209
30	137	11 127	13 59	17 181	19 173	23 79	29 193	31 167	37 191	41 157	43 179	47 61	53 169	67 71	73 209
31	139	11 89	13 43	17 107	19 151	23 143	29 41	31 79	37 157	47 137	53 173	59 191	61 109	67 187	71 209
32	143	11 13	17 169	19 107	23 61	29 157	31 113	37 89	41 193	43 101	47 79	53 181	59 127	67 209	71 73
33	149	11 109	13 173	17 157	19 41	23 43	29 121	31 59	37 197	47 97	53 193	61 209	67 137	71 79	73 143
34	151	11 71	13 157	17 83	19 19	23 107	29 179	31 181	37 163	41 101	43 67	47 173	53 197	59 209	61 61
35	157	11 167	13 109	17 71	19 163	23 89	29 143	31 127	37 61	41 137	43 199	47 191	53 209	59 113	67 181
36	163	11 53	13 61	17 59	19 97	23 71	29 107	31 73	37 169	41 173	43 121	47 209	67 109	79 127	83 131
37	167	11 187	13 29	17 121	19 53	23 199	31 107	37 101	41 127	43 209	47 151	59 163	61 137	67 131	71 97
38	169	11 149	13 13	17 47	19 31	23 53	29 71	37 67	41 209	43 43	59 131	61 199	73 103	79 151	83 83
39	173	11 73	13 191	17 109	19 197	23 181	29 187	31 53	37 209	41 163	43 131	47 169	59 67	61 113	71 103
40	179	11 169	13 143	17 97	19 131	23 163	29 151	31 209	37 107	41 199	43 53	47 187	59 181	61 89	67 197
41	181	11 131	13 127	17 23	19 109	29 209	31 121	37 73	41 71	43 97	47 53	59 149	61 151	67 103	79 199
42	187	11 17	13 79	19 43	23 209	29 173	31 67	37 181	41 107	47 71	53 59	61 127	73 109	83 149	89 113
43	191	11 151	13 47	17 73	19 209	23 127	29 79	31 101	37 113	41 61	43 107	53 67	59 199	71 121	83 187
44	193	11 113	13 31	17 209	19 187	23 191	29 137	37 79	41 143	43 151	47 89	53 71	59 167	61 103	67 169
45	197	11 37	13 209	17 61	19 143	23 109	29 43	31 47	41 97	53 79	59 103	67 191	71 127	73 89	83 139
46	199	11 209	13 193	17 197	19 121	23 173	29 101	31 169	37 187	41 179	43 73	47 107	53 83	59 71	61 79
47	209	11 19	13 113	17 37	23 73	29 181	31 149	41 169	43 83	47 67	53 103	59 121	61 179	71 139	79 101
48	211	11 191	13 97	17 173	19 199	23 137	29 29	31 61	37 193	41 41	43 127	47 143	53 107	59 89	67 163

i	r_i	Pair	15	16	17	18	19	20	21	22	23	24	25	26	27	28
		q_{ji}														
		$q_{\hat{j}i}$							$q_{j\|i}$	$q_{\hat{j}\|i}$						
1	11		79 149	83 157	97 143	101 181	107 163	121 191	169 179	173 187	193 197	199 209				
2	13		71 83	73 121	79 157	89 137	97 169	103 151	109 187	127 139	143 191	197 209				

Continued

3	17	83 109 89 163 101 127 103 149 113 199 137 181 143 169 151 167 187 191 193 209
4	19	71 89 73 193 79 181 101 179 107 167 113 173 121 139 127 187 151 199 191 209
5	23	71 163 73 101 79 197 89 97 107 169 109 137 113 121 127 149 173 181 187 209
6	29	71 169 73 173 97 167 103 143 109 191 121 179 127 197 131 199 149 151 181 209
7	31	89 149 97 193 107 173 109 139 131 191 143 197 151 181 163 187 169 199 179 209
8	37	83 89 103 139 109 193 113 149 121 127 131 167 163 169 173 209 181 187 191 197
9	41	71 181 73 107 83 127 89 109 97 113 101 121 103 137 131 151 149 199 169 209
10	43	97 139 101 173 107 179 121 193 127 169 131 143 137 149 151 163 167 209 187 199
11	47	71 187 73 179 79 83 97 191 107 181 109 143 121 167 127 131 163 209 169 173
12	53	89 187 103 131 109 197 113 151 127 179 137 169 143 181 157 209 163 191 167 199
13	59	79 131 83 193 89 121 97 137 101 169 107 187 151 209 157 197 163 173 179 181
14	61	79 139 83 107 97 163 103 127 109 199 121 181 137 143 149 209 157 193 173 197
15	67	73 139 79 163 89 173 101 167 107 191 113 179 127 151 143 209 157 181 169 193
16	71	83 97 89 199 107 193 109 149 113 127 137 163 139 209 143 187 151 191 157 173
17	73	89 107 97 109 101 113 103 121 127 199 137 209 157 169 167 179 173 191 181 193
18	79	67 67 71 149 73 73 79 211 83 173 97 187 107 197 109 151 121 169 131 209 137 137 143 143 157 157 193 193
19	83	107 199 113 181 121 143 127 209 131 193 139 197 149 157 151 173 167 169 179 187
20	89	97 107 101 109 103 113 121 209 131 169 137 157 139 191 143 193 149 181 187 197
21	97	101 107 103 109 113 209 121 157 127 181 131 137 143 149 151 187 163 199 173 179
22	101	73 197 79 89 101 211 103 107 109 209 113 157 121 131 127 143 139 179 167 193
23	103	79 97 83 191 89 197 103 211 107 209 109 157 113 131 149 167 163 181 169 187
24	107	97 131 101 157 103 209 107 211 121 197 127 191 137 151 139 173 143 199 179 193
25	109	83 143 89 131 97 157 101 209 103 103 107 107 109 211 137 197 149 191 151 169 163 163 173 173 181 199 187 187
26	113	79 137 83 181 89 157 97 209 101 103 107 109 113 211 139 167 169 197 191 193
27	121	71 191 73 157 79 169 89 209 97 103 101 101 107 113 109 109 121 211 127 163 151 151 167 173 179 179 199 199
28	127	67 121 71 197 79 193 83 209 89 143 97 181 103 199 109 163 127 211 137 191
29	131	89 169 101 151 103 197 113 187 127 173 131 211 139 149 163 167 179 199 181 191
30	137	83 199 89 103 97 101 107 121 109 113 131 187 137 211 139 143 149 163 151 197
31	139	73 163 83 113 97 127 101 149 103 193 121 199 131 179 139 211 167 197 169 181
32	143	83 151 97 179 103 191 109 167 121 173 131 163 137 139 143 211 149 187 197 199
33	149	83 103 89 181 101 199 107 127 113 163 131 139 149 211 151 179 167 187 169 191
34	151	73 97 79 79 89 89 103 187 109 169 113 137 121 121 127 193 131 131 139 199 143 167 149 149 151 211 191 191
35	157	73 169 79 103 83 179 97 151 101 197 107 131 121 187 139 193 149 173 157 211
36	163	89 167 101 143 103 181 113 191 137 179 139 187 149 197 151 193 157 199 163 211
37	167	73 149 79 143 83 169 89 193 103 179 109 173 113 139 157 191 167 211 181 197

Continued

38	169	89	101	97	97	107	137	109	121	113	113	127	127	139	181	143	173	157	187	163	193	167	167	169	211	179	191	197	197
39	173	79	167	83	121	89	127	97	149	101	193	107	139	137	199	143	151	157	179	173	211								
40	179	71	109	73	83	79	191	101	139	103	173	113	193	121	149	127	137	157	167	179	211								
41	181	83	197	89	179	101	191	107	143	113	167	137	173	139	169	157	163	181	211	187	193								
42	187	97	121	101	137	103	169	131	197	139	163	143	179	151	157	167	191	187	211	193	199								
43	191	89	139	97	173	103	167	109	179	131	181	137	193	143	157	149	169	163	197	191	211								
44	193	73	181	83	101	97	199	107	149	109	127	121	163	131	173	139	157	179	197	193	211								
45	197	101	187	107	151	113	169	121	137	131	157	149	193	163	179	167	181	173	199	197	211								
46	199	67	97	89	191	103	163	109	181	113	143	127	157	131	149	137	167	139	151	199	211								
47	209	89	151	97	197	107	157	109	131	127	167	137	187	143	163	173	193	191	199	209	211								
48	211	71	71	73	187	79	109	83	167	101	131	103	157	113	197	121	151	139	139	149	179	169	169	181	181	209	209	211	211

The proposed method is well structured for semiprimes factorization when breaking RSA encryption and readily applicable for parallel computation.

The rest of the paper is organized as follows. In Section 2, we introduce the basic theory of semiprime factorization based on the concepts of factor pair and kernel factor pair. In Section 3, we provide a framework of designing a Linear-search Factor-pair Kernel (LFK) algorithm for semiprime factorization. The proposed LFK algorithm is presented in Section 4 while a complexity analysis is given in Section 5. An illustrative example is provided in Section 6. The paper ends with conclusions and discussions in Section 7.

The following notations are used throughout the paper:

- α : a semiprime number
- $N_0 = \{0,1,2,\cdots\}$: all non-negative integers
- $N = \{1,2,3,\cdots\}$: all natural numbers
- $\Lambda = \{n \in N \mid n > 1$ and n has no factors of 2, 3, 5 and 7$\}$
- $P = \{n \in N \mid n > 1$ and n is a prime$\}$: all prime numbers
- $P_S = \{n \in N \mid n$ is a product of two primes$\}$: all semiprime numbers
- $[a,b]_N = \{n \in N \mid a \le n \le b\}$
- FT_{210} = a table of 48 rows and 28 columns consisting of all factor-pairs
- $\lceil x \rceil$ = ceiling function of the smallest integer greater than or equal to x
- $\lfloor x \rfloor$ = floor function of the largest integer smaller than or equal to x

2. Basic Theory

For a given number $n \in N$, it is relatively easy to identify if it has any factor of 2, 3, 5 and 7. Since $2 \times 3 \times 5 \times 7 = 210$, we may group every 210 numbers in one period starting from 11, *i.e.*, $[11, 220]_N$, $[221, 430]_N$, $[431, 640]_N$, … Checking the numbers in $[11, 220]_N$, we can identify 48 numbers which have no factors of 2, 3, 5 and 7, *i.e.*, $[11, 220]_N \cap \Lambda = 48$. These 48 numbers are listed as the roots (root numbers) in the following set:

$$\begin{aligned}S=\{&r_1=11, r_2=13, r_3=17, r_4=19, r_5=23, r_6=29, r_7=31, r_8=37, r_9=41,\\ &r_{10}=43, r_{11}=47, r_{12}=53, r_{13}=59, r_{14}=61, r_{15}=67, r_{16}=71, r_{17}=73,\\ &r_{18}=79, r_{19}=83, r_{20}=89, r_{21}=97, r_{22}=101, r_{23}=103, r_{24}=107,\\ &r_{25}=109, r_{26}=113, r_{27}=121, r_{28}=127, r_{29}=131, r_{30}=137, r_{31}=139,\\ &r_{32}=143, r_{33}=149, r_{34}=151, r_{35}=157, r_{36}=163, r_{37}=167, r_{38}=169,\\ &r_{39}=173, r_{40}=179, r_{41}=181, r_{42}=187, r_{43}=191, r_{44}=193, r_{45}=197,\\ &r_{46}=199, r_{47}=209, r_{48}=211\}.\end{aligned}$$

A quick observation shows that 5 roots in S are composite numbers, namely, $r_{27}=121=11\times 11$, $r_{32}=143=11\times 13$, $r_{38}=169=13\times 13$, $r_{42}=187=11\times 17$ and $r_{47}=209=11\times 19$, and the rest 43 root numbers in S are primes.

An immediate result shows that any natural number α without factors of 2, 3, 5 and 7, no matter it is a prime or composite number, can be traced back to a unique root $r_i\in S$.

Theorem 1.

A number $\alpha\in\Lambda$ if and only if there exists a unique $r_i\in S$ and $k\in N_0$ such that $\alpha=r_i+210\times k$.

Proof.

i) If $\alpha=r_i+210\times k$ for any $r_i\in S$ and $k\in N_0$, then $\alpha\in\Lambda$ because r_i is the residue of α divided by 2, 3, 5 and 7.

ii) If $\alpha\in\Lambda$, then we let $k=\lfloor(\alpha-10)/210\rfloor\in N_0$ and $\alpha-210\times k\in[11, 220]_N\cap\Lambda$. Hence there exists a unique $r_i\in S$ such that $\alpha=r_i+210\times k$. □

Theorem 1 ensures that every number α without factors of 2, 3, 5 and 7 is an offspring of one unique root $r_i\in S$ in the k-th period of 210. Here we emphasize that any prime number greater than 10 is rooted back to a unique $r_i\in S$. Moreover, we enlist all numbers without factors of 2, 3, 5 and 7 as below:

Period $k=0$, $[11, 220]_N\cap\Lambda=\{q_1=r_1, q_2=r_2, \cdots, q_{48}=r_{48}\}$.

Period $k=1$, $[221, 430]_N\cap\Lambda=\{q_1+210, q_2+210, \cdots, q_{48}+210\}$,

$\vdots$

Period $k=\bar{k}$,

$\left[11+210\bar{k}, 220+210\bar{k}\right]_N\cap\Lambda=\left\{q_1+210\times\bar{k}, q_2+210\times\bar{k}, \cdots, q_{48}+210\times\bar{k}\right\}$.

We now focus on semiprime numbers. Since any semiprime $\alpha\in P_S$ is a product of two primes and it is easy to identify if α has any factor of 2, 3, 5 or 7, without loss of generality, we may limit our consideration to $\alpha\in P_S\cap\Lambda$. In this case, from Theorem 1, there exists a unique $r_i\in S$, $k_i\in N_0$ such that

$$\alpha=r_i+210k_i. \tag{1}$$

Moreover, there must exist r_j, $r_{\hat{j}}\in S$ and k_j, $k_{\hat{j}}\in N_0$ such that

$$\alpha=\left(r_j+210k_j\right)\times\left(r_{\hat{j}}+210k_{\hat{j}}\right). \tag{2}$$

Note that $r_j=r_{\hat{j}}$ and/or $k_j=k_{\hat{j}}$ are allowed.

Equation (1) and Equation (2) require that

$$\frac{r_j\times r_{\hat{j}}-r_i}{210}=k_i-\left(k_j\times r_{\hat{j}}+k_{\hat{j}}\times r_j+210k_j\times k_{\hat{j}}\right).$$

Therefore, $r_j \times r_{\hat{j}} - r_i$ has to be a multiple of 210. We call $\left(r_j, r_{\hat{j}}\right) \in S \times S$ a factor pair with respect to $r_i \in S$ and denote the pair by $\left(q_{j|i}, q_{\hat{j}|i}\right)$ where $q_{j|i} = r_j$ and $q_{\hat{j}|i} = r_{\hat{j}}$. When $r_j = r_{\hat{j}}$, we call $\left(q_{j|i} = r_j, q_{\hat{j}|i} = r_{\hat{j}}\right)$ a co-factor pair with respect to r_i.

Obtained through some elaborative calculations, the Factor-Pairs Table FT_{210} (Table 1) shows all factor/co-factor pairs $\left(q_{j|i}, q_{\hat{j}|i}\right) \in S \times S$ with respect to each $r_i \in S$ in the arrangement of $q_{j|i} \le q_{\hat{j}|i}$. For each $r_i \in S$, $i = 1, 2, \cdots, 48$, we define $Q(i) = \left\{\left(q_{j|i}, q_{\hat{j}|i}\right) \in S \times S \mid q_{j|i} \le q_{\hat{j}|i} \text{ and } q_{j|i} \times q_{\hat{j}|i} - r_i \text{ is a multiple of } 210\right\}$.

From FT_{210}, we see that (i) for $i = 18, 25, 27, 34, 38$ and 48, $|Q(i)| = 28$ with 4 co-factor pairs, (ii) for the rest 42 r_i's, $|Q(i)| = 24$ with no co-factor pairs, and (iii) $q_{j|i} \times q_{\hat{j}|i} \ge r_i$ for all $r_i \in S$.

Summarizing the above, we reach the next result.

Theorem 2.

For any given semiprime α *with no factors of* 2, 3, 5 a*nd* 7, *i.e.,* $\alpha \in P_S \cap \Lambda$, *there exists a unique* $r_i \in S$ *with* $k_i \in N_0$ *and a factor pair* $\left(q_{j|i}, q_{\hat{j}|i}\right) \in Q(i)$ *with* k_j, $k_{\hat{j}} \in N_0$ *such that*

$$\alpha = r_i + 210k_i = \left(q_{j|i} + 210k_j\right) \times \left(q_{\hat{j}|i} + 210k_{\hat{j}}\right). \tag{3}$$

It is important to note that since the factorization of a semiprime α is unique, the corresponding factor pair of α, *i.e.*, $\left(q_{j|i}, q_{\hat{j}|i}\right) \in Q(i)$, is unique. We call it the kernel factor pair associated with the semiprime α.

3. Algorithm Design

Based on Theorem 2, we introduce the overall design of an algorithm for semiprime factorization based on the kernel factor pair.

For a given semiprime α, it is easy to check if α has a factor of 2, 3, 5 or 7. Without loss of generality, we may assume that $\alpha \in P_S \cap \Lambda$. The root number $r_i \in S$ can be readily identified by computing

$$r_i = \alpha - \left\lfloor \frac{\alpha - 10}{210} \right\rfloor \times 210 \tag{4}$$

with

$$k_i = \left\lfloor \frac{\alpha - 10}{210} \right\rfloor.$$

Once r_i is determined for $i \in \{1, \cdots, 48\}$, then we check each factor pair $\left(q_{j|i}, q_{\hat{j}|i}\right) \in Q(i)$ to see if Equation (3) is satisfied by certain k_j, $k_{\hat{j}} \in N_0$. If the answer is Yes, then $(q_{j|i}, q_{\hat{j}|i})$ is a kernel factor pair for factoring α, and the existence and uniqueness of a kernel factor pair in $Q(i)$ for $\alpha \equiv r_i \pmod{210}$ is assured by Theorem 2.

For a given factor pair $\left(q_{j|i}, q_{\hat{j}|i}\right) \in Q(i)$, to check if it is a kernel factor pair for α is equivalent to finding θ and $\hat{\theta} \in N_0$ such that

$$\alpha = \left(q_{j|i} + 210 \times \theta\right)\left(q_{\hat{j}|i} + 210 \times \hat{\theta}\right),$$

or equivalently,

$$\frac{\alpha - q_{j|i} \times q_{\hat{j}|i}}{210} = q_{j|i} \times \hat{\theta} + q_{\hat{j}|i} \times \theta + 210 \times \theta \times \hat{\theta}.$$

Denoting

$$\sigma_{j|i} = \frac{\alpha - \left(q_{j|i} \times q_{\hat{j}|i}\right)}{210} \left(\le \frac{\alpha - 11^2}{210} \right), \tag{5}$$

we have

$$\sigma_{j|i} = q_{j|i} \times \hat{\theta} + q_{\hat{j}|i} \times \theta + 210 \times \theta \times \hat{\theta}. \tag{6}$$

Consequently, we may perform a less desirable "quadratic search" on $\theta, \hat{\theta} \in \left\{0, 1, \cdots, \left\lceil \sigma_{j|i} / 11 \right\rceil\right\}$ simultaneously to see if $\left(q_{j|i}, q_{\hat{j}|i}\right)$ is a kernel factor pair for α.

4. LFK—A Linear Search Algorithm

To reduce the complexity involved in the quadratic search, we conduct further analysis to reach a "linear search" algorithm. The algorithm presented here is closely related to the theoretical framework developed in [29], which involves an innovative algorithm for general primality testing. In this paper, matching the concept of "factor pairs" with the characteristics of "semiprime", we focus on developing an elaborated algorithm specifically for semiprime factorization.

Without loss of generality, we assume that $\theta \ge \hat{\theta} \ge 0$ in Equation (6) which leads to

$$\sigma_{j|i} \ge 210 \times \theta \times \hat{\theta} \ge 210\hat{\theta}^2.$$

Together with Equation (5), we have

$$\hat{\theta}^2 \le \frac{\sigma_{j|i}}{210} \le \frac{\alpha - 11^2}{210^2}.$$

Hence $\hat{\theta} \le \sqrt{\alpha} / 210$. This means that it is sufficient to check for $\hat{\theta} \in \left\{0, 1, \cdots, \left\lceil \sqrt{\alpha} / 210 \right\rceil\right\}$.

Once $\hat{\theta}$ is selected, Equation (6) indicates that

$$\theta = \frac{\sigma_{j|i} - q_{j|i} \times \hat{\theta}}{q_{\hat{j}|i} + 210\hat{\theta}} \tag{7}$$

should be an integer greater than or equal to $\hat{\theta}$.

Our approach leads to the following LFK method for semiprime factorization.

Step1: Input a semiprime number α. If α can be fully divided by 2, 3, 5 or 7, then stop with a simple factorization.

Step2: Determine the root number $r_i \in S$.

Set $r_i = \alpha - \left\lfloor \frac{\alpha - 10}{210} \right\rfloor \times 210$.

Step3: Determine the kernel factor pair in $Q(i)$.

Pick one unchecked factor pair a time in $Q(i)$ from FT_{210}, say $\left(q_{j|i}, q_{\hat{j}|i}\right)$.

Set $\sigma_{j|i} = \left(\alpha - q_{j|i} \times q_{\hat{j}|i}\right) / 210$.

Step4: Check for possible $\theta \ge \hat{\theta} \ge 0$.

For $\hat{\theta} \in \left\{0, 1, \cdots, \left\lceil \sqrt{\alpha}/210 \right\rceil\right\}$, compute θ using Equation (7).

If θ is an integer and $\theta \ge \hat{\theta}$, then output $\left(q_{j|i}, q_{\hat{j}|i}\right)$ as the kernel factor pair and $\alpha = \left(q_{j|i} + 210 \times \theta\right) \times \left(q_{\hat{j}|i} + 210 \times \hat{\theta}\right)$. Stop the algorithm.

Step5: Check for possible $0 \le \theta \le \hat{\theta}$.

For $\theta \in \left\{0, 1, \cdots, \left\lceil \sqrt{\alpha}/210 \right\rceil\right\}$, compute

$$\hat{\theta} = \frac{\sigma_{j|i} - q_{\hat{j}|i} \times \theta}{q_{j|i} + 210\theta}.$$

If $\hat{\theta}$ is an integer and $\hat{\theta} > \theta$, then output $\left(q_{j|i}, q_{\hat{j}|i}\right)$ as the kernel factor pair and $\alpha = \left(q_{j|i} + 210 \times \theta\right) \times \left(q_{\hat{j}|i} + 210 \times \hat{\theta}\right)$. Stop the algorithm.

Step6: Mark the current factor pair as "checked" and Return to Step 3 for the next unchecked factor pair in $Q(i)$.

As a direct consequence of Theorem 2, we obtain the next result.

Corollary 3.

The proposed LFK algorithm always terminates in finite steps with a unique kernel factor pair for any given semiprime α.

5. Complexity Analysis

For any given semiprime number $\alpha \in P_S$, we know $\alpha = p_1 \times p_2$ for a unique pair of $p_1 \in P$ and $p_2 \in P$. Except that p_1 or p_2 is 2, 3, 5 or 7, which can be easily verified, α has no factors of 2, 3, 5 and 7, *i.e.*, $\alpha \in P_S \cap \Lambda$. Theorem 1 assures that α has a unique root number $r_i \in S$. Table FT_{210} shows that there are either 24 or 28 factor pairs in $Q(i)$ for a semiprime rooted at r_i.

Theorem 2 further confirms the existence of a unique kernel factor pair $\left(q_{j|i}, q_{\hat{j}|i}\right) \in Q(i)$ such that α can be factorized as the product of $\left(q_{j|i} + 210 \times \theta\right)$ and $\left(q_{\hat{j}|i} + 210 \times \hat{\theta}\right)$.

Steps 1 - 3 of the LFK Algorithm set up the platform with simple calculations. Considering the possible relations between θ and $\hat{\theta}$, the desired kernel factor pair can be identified by searching through $1 + \left\lceil \sqrt{\alpha}/210 \right\rceil$ integer values for $\hat{\theta}$ first and then, when needed, searching through $1 + \left\lceil \sqrt{\alpha}/210 \right\rceil$ integer values for θ. Therefore, the LFK algorithm involves at most $28 \times 2 = 56$ linear searches over $1 + \left\lceil \sqrt{\alpha}/210 \right\rceil$ integer values. In other words, a rough estimation of the required work for factoring a semiprime α involves no more than searching through $56 \times \sqrt{\alpha}/210 \approx \sqrt{\alpha}/4$ integer values. The previous analysis leads to the next result.

Theorem 4.

The LFK algorithm requires at most 56 linear searches over the integer values in $\left[0, \left\lceil \sqrt{\alpha}/210 \right\rceil\right]$ *to find a unique kernel factor pair for any given semiprime* α.

It is worthy to point out that the tasks for searching the kernel factor pair out of the 28 (or 24) factor pairs in $Q(i)$ can be conducted independently in parallel.

6. Illustrating Example

We illustrate the LFK algorithm using the following example to factor an 11-digit semiprime $\alpha = 12648677849$.

Step1: Since α cannot be fully divided by 2, 3, 5 and 7, we know $\alpha \in P_S \cap \Lambda$.

Step2: Since $\alpha = 59 + 210 \times 60231799$, $r_{13} = 59$ is the root number of α.

Step3: We sequentially check the 24 factor-pairs in $Q(13)$ from FT_{210}, namely, $Q(13) = \{(11,139),(13,53),(17,127),\cdots,(71,199),\cdots,(179,181)\}$. After taking 12 iterations going through Steps 4 and 5 to check $\left(q_{j|13}, q_{\hat{j}|13}\right)$ for $j = 1,2,\cdots,12$ without finding any feasible solution, we check $\left(q_{13|13}, q_{\hat{13}|13}\right) = (71,199)$. Calculate $\sigma_{13|13} = \dfrac{\alpha - 71 \times 199}{210} = 60231732$.

Step4: Check for possible $\theta \geq \hat{\theta} \geq 0$.

Compute

$$\theta = \frac{\sigma_{j|i} - q_{j|i} \times \hat{\theta}}{q_{\hat{j}|i} + 210\hat{\theta}} = \frac{60231732 - 71\hat{\theta}}{199 + 210\hat{\theta}}$$

for each $\hat{\theta} \in \left\{0, 1, \cdots, \left\lceil \sqrt{\alpha}/210 \right\rceil\right\} = \{0, 1, \cdots, 536\}$. Since none of the $\hat{\theta}$'s are integers, continue to Step 5.

Step5: Check for possible $0 \leq \theta \leq \hat{\theta}$.

Compute $\hat{\theta} = \dfrac{\sigma_{j|i} - q_{\hat{j}|i} \times \theta}{q_{j|i} + 210\theta} = \dfrac{60231732 - 199\theta}{71 + 210\theta}$ for each $\theta \in \{0, 1, \cdots, 536\}$.

When $\theta = 473$, $\hat{\theta} = \dfrac{60231732 - 199 \times 473}{71 + 210 \times 473} = 605 > \theta$ is an integer. Output (71, 199) as the kernel factor pair and

$\alpha = 12648677849 = (71 + 210 \times 473) \times (199 + 210 \times 605) = 99401 \times 127249$.

7. Conclusions and Discussions

In this paper, we introduce the concepts of factor pairs and kernel factor pairs to form a new framework for designing an efficient semiprime factoring algorithm that serves as the key input to the RSA algorithm for computer encryption. Unlike commonly developed sieve-based methods, the proposed kernel-factor-pair-based LFK algorithm is proven to successfully factorize any given semiprime α by taking at most 56 linear searches over the integer values in [0, $\left\lceil \sqrt{\alpha}/210 \right\rceil$]. The overall computational complexity is less than searching through $\left\lceil \sqrt{\alpha}/4 \right\rceil$ integer values with simple calculations. The LFK algorithm works under no pre-conditions and finds the exact factors for any given semiprime. The required search can be conducted naturally in parallel for fast computations.

As pointed out in [27], instead of eliminating the factors of 2, 3, 5 and 7 for consideration, an "extended" PTP table can be easily built by eliminating fewer factors, such as 2, 3, 5 only, or by eliminating more factors, such as 2, 3, 5, 7 and 11. Since the construction of the PTP is not sieve-based, it requires no prior knowledge or processing efforts of any other primes. In fact, an extended PTP that eliminates more factors for consideration will have a longer period but with more roots. The concepts of factor pairs and kernel factor pair follow accordingly for the LFK algorithm to prevail.

Some further discussions are as follows:

1) Notice that the LFK searches θ (and $\hat{\theta}$) through the integer values from 0 to $\lceil \sqrt{\alpha}/210 \rceil$. When $\alpha \approx 2^d$ becomes a huge number in d digits, the searching range could be a concern. In this case, by adopting a prime-logarithmic technique in [4], we can simply find an integer m such that $2^m \geq \lceil \sqrt{\alpha}/210 \rceil \geq 2^{m-1}$ and introduce m ($\sim \frac{1}{2}\log\alpha$) 0 - 1 binary variables $u_1, \cdots, u_m$ such that θ can be represented by

$$\theta = 2^0 u_1 + 2^1 u_2 + 2^2 u_3 + \cdots + 2^{m-1} u_m \quad \text{for} \quad u_i \in \{0, 1\}.$$

In other words, the task of searching θ through the integer values from 0 to $\lceil \sqrt{\alpha}/210 \rceil$ can be replaced by checking the values of m (about $1/2\log\alpha$) binary variables. For instance, to factorize a semiprime $\alpha = 2^{32} + 1 = 4294967297$, we only need 16 binary variables to find the solution being

$$2^{32} + 1 = (11 + 210 \times 3) \times (157 + 210 \times 31906) = 641 \times 6700417.$$

However, the performance of the prime-logarithmic technique developed by Li *et al.* [4] strongly depends on, firstly, the way of formulating the related linear binary programming problem, and secondly, the software package used to solve the linear binary programming problem. This will remain for further study.

2) Compared with the commonly used sieve-based semiprime factoring methods, we observe that:

(1) Heuristic versus deterministic: Many sieve-based methods are heuristic and may not converge to a feasible solution. However, the proposed LFK algorithm is a deterministic approach which guarantees to factorize any semiprime number into two prime factors.

(2) High space complexity versus low space complexity: The Trial Sieve Method requires a high space complexity to store all primes that come before $\sqrt{\alpha}$, while, with the FT_{210} in hand, the LFK algorithm needs at most 28 memory spaces to store the corresponding factor pairs.

(3) Hard-to-manage versus manageable parallel computing: Sieve-based methods may generate numerous subproblems which are hard to arrange for parallel processing. The LFK algorithm can easily arrange all subproblems into 24 or 28 processes to be computed parallelly.

In conclusion, fast semiprime factorization is essential for breaking RSA encryption. The kernel-factor-based LFK algorithm provides a new angle for further investigation on designing efficient factoring methods for information security.

Acknowledgements

We thank Professor Xiaohua Jia of City University of Hong Kong and an anonymous reviewer for reviewing and providing valuable comments to the article. We are grateful for the in-kind-support from NTU and the financial support for Way Kuo from the Hong Kong Institute for Advanced Study (CityU 9610556).

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Zagier, D. (1977) The First 50 Million Prime Numbers. *The Mathematical Intelligencer*, **1**, 7-19. https://doi.org/10.1007/bf03351556

[2] Zagier, D. (1997) Newman's Short Proof of the Prime Number Theorem. *The American Mathematical Monthly*, **104**, 705-708. https://doi.org/10.1080/00029890.1997.11990704

[3] Dickson, L.E. (2005) History of the Theory of Numbers, Volume II: Diophantine Analysis. Dover Publications.

[4] Li, H., Huang, Y., Fang, S. and Kuo, W. (2021) A Prime-Logarithmic Method for Optimal Reliability Design. *IEEE Transactions on Reliability*, **70**, 146-162. https://doi.org/10.1109/tr.2020.3020597

[5] Li, H., Fang, S., Lin, B.M.T. and Kuo, W. (2023) Unifying Colors by Primes. *Light: Science & Applications*, **12**, Article No. 32. https://doi.org/10.1038/s41377-023-01073-x

[6] Konheim, A.G. (2006) Computer Security and Cryptography. Wiley. https://doi.org/10.1002/0470083980

[7] Li, D., Luo, M., Zhao, B. and Che, X. (2018) Provably Secure APK Redevelopment Authorization Scheme in the Standard Model. *Computers, Materials & Continua*, **56**, 447-465. https://doi.org/10.3970/cmc.2018.03692

[8] RSA Laboratories (2013) The RSA Factoring Challenge. https://web.archive.org/web/20130921043459/http://www.emc.com/emc-plus/rsa-labs/historical/the-rsa-factoring-challenge.htm

[9] Shor, P.W. (1997) Polynomial-time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, **26**, 1484-1509. https://doi.org/10.1137/s0097539795293172

[10] Shoshina, A.V., Borzunov, G.I. and Ivanova, E.Y. (2021) Application of Bio-Inspired Algorithms to the Cryptanalysis of Asymmetric Ciphers on the Basis of Composite Number. 2021 *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering* (*ElConRus*), St. Petersburg, 26-29 January 2021, 2399-2403. https://doi.org/10.1109/elconrus51938.2021.9396242

[11] Upadhyay, S. and Gupta, V.K. (2022) A Literature Review on the Concept of Cryptography and RSA Algorithm. *International J of Advance and Innovative Research*, **9**, 237-240.

[12] Vandersypen, L.M.K., Steffen, M., Breyta, G., Yannoni, C.S., Sherwood, M.H. and Chuang, I.L. (2001) Experimental Realization of Shor's Quantum Factoring Algorithm Using Nuclear Magnetic Resonance. *Nature*, **414**, 883-887. https://doi.org/10.1038/414883a

[13] Boudot, F., Gaudry, P., Guillevic, A., Heninger, N., Thome, E. and Zimmermann, P. (2022) The State of the Art in Integer Factoring and Breaking Public-Key Cryptography. *IEEE Security & Privacy*, **20**, 80-86. https://doi.org/10.1109/msec.2022.3141918

[14] Zhang, X., Li, M., Jiang, Y. and Sun, Y. (2019) A Review of the Factorization Problem of Large Integers. In: Sun, X., Pan, Z. and Bertino, E., Eds., *Artificial Intelligence and Security*, Springer, 202-213. https://doi.org/10.1007/978-3-030-24268-8_19

[15] Pomerance, C. and Erdös, P. (1996) A Tale of Two Sieves. *Notices of the American Mathematical Society*, **43**, 1473-1485.

[16] Pritchard, P. (1982) Explaining the Wheel Sieve. *Acta Informatica*, **17**, 477-485. https://doi.org/10.1007/bf00264164

[17] Wikipedia Contributors (2025) Wheel Factorization. https://en.wikipedia.org/w/index.php?title=Wheel_factorization&oldid=1279299441

[18] Atkin, A.O.L. and Bernstein, D.J. (2003) Prime Sieves Using Binary Quadratic Forms. *Mathematics of Computation*, **73**, 1023-1030. https://doi.org/10.1090/s0025-5718-03-01501-1

[19] Pollard, J.M. (1993) The Lattice Sieve. In: Lenstra, A.K. and Lenstra, H.W., Eds., *The Development of the Number Field Sieve*, Springer, 43-49. https://doi.org/10.1007/bfb0091538

[20] Dixon, B. and Lenstra, A.K. (1994) Factoring Integers Using SIMD Sieves. In: Helleseth, T., Ed., *Advances in Cryptology—EUROCRYPT*'93. *EUROCRYPT* 1993, Springer, 28-39.

[21] Kleinjung, T., Aoki, K., Franke, J., Lenstra, A.K., Thomé, E., Bos, J.W., *et al.* (2010) Factorization of a 768-Bit RSA Modulus. In: Rabin, T., Ed., *Advances in Cryptology—CRYPTO* 2010, Springer, 333-350. https://doi.org/10.1007/978-3-642-14623-7_18

[22] Menezes, A. and Vanstone, S.A. (1993) Elliptic Curve Cryptosystems and Their Implementation. *Journal of Cryptology*, **6**, 209-224.

[23] Bai, S., Gaudry, P., Kruppa, A., Thomé, E. and Zimmermann, P. (2016) Factorisation of RSA-220 with CADO-NFS. https://inria.hal.science/hal-01315738

[24] Schnorr, C.P. (2013) Factoring Integers by CVP Algorithms. In: Fischlin, M. and Katzenbeisser, S., Eds., *Number Theory and Cryptography*, Springer, 73-93. https://doi.org/10.1007/978-3-642-42001-6_6

[25] Schnorr, C.P. (2021) Fast Factoring Integers by SVP Algorithms. https://eprint.iacr.org/2021/933

[26] Tang, X., Xu, J. and Duan, B. (2018) A Memory-Efficient Simulation Method of Grover's Search Algorithm. *Computers, Materials & Continua*, **57**, 307-319. https://doi.org/10.32604/cmc.2018.03693

[27] Li, H., Fang, S. and Kuo, W. (2024) The Periodic Table of Primes. *Advances in Pure Mathematics*, **14**, 394-419. https://doi.org/10.4236/apm.2024.145023

[28] Li, H., Fang, S. and Kuo, W. (2024) The Periodic Table of Primes. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.4742238

[29] Li, H., Fang, S., Kuo, W. and Lin, N. (2025) Listing Prime Numbers Periodically. *Advances in Pure Mathematics*, **15**, 247-268. https://doi.org/10.4236/apm.2025.154012