An Inverse Factorization of Minimum Mersenne Number with a Specified Factor

Harunori Nakayama and Seiji Anbe

***Abstract*—Currently, the square-free and Wieferich prime problems of number theory can be solved only via computational means. Because an efficient Wieferich prime exploration algorithm involves the investigation of a squared factor of Mersenne numbers with a prime exponent, we propose an inverse factorization algorithm to obtain both the exponent and another factor of the minimum Mersenne number with a specified factor. If we specify any prime, we can detect whether it is a Wieferich prime. We demonstrated the procedure with suitable examples and discussed the application of the classical baby-step giant-step algorithm to this limitation. Moreover, the inverse factorization is generalized not only to Mersenne numbers but also to repunits and repdigits. Finally, we briefly discuss cipher applications by applying our algorithm to a concrete example of inverse factorization of Mersenne numbers. This encryption algorithm expands the bit length by the nonlinearity between plaintext and ciphertext. The block length of the ciphertext becomes the decryption key. Block lengths that are less than the decryption key include a computation load maximization block, which improves the security.**

***Index Terms*— block cipher, inverse factorization, repunits, Wieferich primes**

I. Introduction

Mersenne numbers, which are related to perfect numbers [1], [2], have been studied for a long time and can be expressed in the form $M_n = 2^n - 1$ for some $(n \in \mathbb{N})$ [3]. However, it remains unclear whether all Mersenne numbers that have prime exponents have squared factors; this is known as the so-called square-free problem (SFP) [4]. In relation to the Mersenne numbers are the Wieferich primes, which are prime numbers p that satisfy $2^{p-1} \equiv 1 \pmod{p^2}$. In 2005, it was reported that no Wieferich prime is less than $1.25 \cdot 10^{15}$, except 1093 and 3511 [5], [6]; this situation is unchanged to date. The issue of whether an infinite number of Wieferich primes exist is known as the Wieferich prime problem (WPP). Observations based on computer experimentation [7] suggest that only a finite number of Wieferich primes exist. However, because there is no theoretical definitive algorithm to solve SFP and WPP, they are approached through computational means. The work of Keller and Richstein [8] may be referred for more generalized $a^{p-1} \equiv 1 \pmod{p^r}$.

SFP can be examined using the results of the Mersenne prime search. If a prime p is specified, $M_p = 2^p - 1$ can be computed. When M_p is determined to be a composite number using the Lucas–Lehmer test [3] or a primality test [9] using an elliptic curve [10], it is possible to examine duplicate factors via factorization [2], [11]. However, the SFP is related to the WPP. As reported by Warren and Bray in 1967 [12], if a Fermat or Mersenne number is not square-free, for any prime factor p whose square divides the given number, $2^{p-1} \equiv 1 \pmod{p^2}$. Thus, modulo p^2 can be used to examine whether $2^{p-1} \equiv 1 \pmod{p^2}$ is satisfied for a prime factor p [5], indicating that the Mersenne number $M_{p-1} = 2^{p-1} - 1$ has p^2 as a factor. Therefore, the efficient strategy is to determine the Wieferich prime p before approaching the SFP. Although p is prime, $p - 1$ is an even number when $p \neq 2$. When $2hn = p - 1$ $(h \in \mathbb{N})$, the prime exponent n of the minimum Mersenne number with factor p^2 may be determined such that $2^{p-1} = 2^{2hn} \equiv 2^n \equiv 1 \pmod{p^2}$. Research on arithmetic sequences for the exponents of composite Mersenne numbers suggests the presence of infinitely multiple composite Mersenne numbers with a prime exponent [13]. Moreover, research of Carlitz module analogs of Mersenne primes demonstrates that infinitely many composite Mersenne numbers exist [14]. This includes the existence of composite Mersenne numbers with a prime exponent, which provides a counterexample to the SFP by WPP. Furthermore, the numerical factorization of Mersenne numbers with a specific exponent n has been examined in certain approaches [2], [11]. However, few computational approaches are available to determine the exponent n of a Mersenne number factored to have a specified factor p. Therefore, identifying an approach to efficiently obtain a Mersenne number with a specified odd number p as a factor without (numerical) factorization should be useful to provide a computational solution for both SFP and WPP.

In this study, we develop an inverse factorization algorithm for obtaining another factor q and exponent n of a Mersenne number $M_n = pq$ with a specified odd p and show how this procedure can be applied to SFP and WPP. By applying an expansion (the inverse of factorization), n can be obtained if p and q are properly specified. When $M_n = 2^n - 1 = pq$, the factorization is denoted as f, expansion as f^{-1}, and inverse factorization as $f^{\sim 1}$. In particular,

$$\begin{aligned} f &: n \to (p, q) \\ f^{-1} &: (p, q) \to n \\ f^{\sim 1} &: p \to (n, q). \end{aligned} \tag{1}$$

Manuscript received November 3, 2021; revised July 3, 2022.
H. Nakayama is an undergraduate student at the Faculty of Liberal Arts in the Open University of Japan. (corresponding author to provide e-mail: 1710745506@campus.ouj.ac.jp)
S. Anbe is the President of a cram school, namely, Learning Forest "English and Mathematics Seminar." (e-mail: 5zp7ac@bma.biglobe.ne.jp)

We now discuss the structure of this study. To prepare the groundwork, we introduce the ordered pairs (cells) for handling a 1D real number in a 2D real-number space. Next, we demonstrate the facility of the inverse transform of the original 1D real number for such ordered pairs. We call the ordered pairs "cells" in this study. Next, the multiplication of a cell is defined, and an outer algorithm is introduced. The algebraic space comprising cells is known as a real cell space. We show that a number can be decomposed into two components using multiplication as the primary operation. The inverse factorization algorithm is developed by applying the binary representation [6] of Mersenne numbers based on common multiplication because Mersenne numbers are repunits [15]. In the "Methods" section, proposed algorithms for approaching the SFP and WPP are sequentially presented, including examples and pseudocodes. Using the exponent n_{I} of the Mersenne number with a prime factor p, the exponent n_{II} of the Mersenne number with a factor p^2 is shown to usually be estimated by pn_{I}. In the "Results" section, focusing on known Wieferich primes, we applied the inverse factorization to all two known Wieferich and 20 non-Wieferich primes and tabulate the results and confirmed that Wieferich and non-Wieferich primes can be distinct. Moreover, there is a public table [16] that listed up 104 Wieferich numbers based on Agoh, Dilcher, and Skula's definition [17]. Using this data, certain detection results of the Wieferich composites are presented. A primality test is required to distinguish between Wieferich primes and composites. In the "Discussion" section, we consider the time complexity of the proposed algorithm with similar known algorithms and present certain future challenges. Furthermore, the inverse factorization algorithm of Mersenne numbers is generalized to repunits and repdigits [15]. Finally, we discuss cipher applications by applying the inverse factorization algorithm to a concrete example of inverse factorization of Mersenne numbers.

II. Preliminaries

We then introduce the conversion of 1D numbers to 2D numbers and vice versa, following which we provide an algebraic structure to 2D numbers.

A. Introduction of ordered pairs (cells)

Given a number s, an exponent k can be generated as the logarithm of s with base m. The relationship between s and k is as follows:

$$s \in S_0, k, x, y \in S_1, \gamma \in S_2, \alpha \in S_3, m \in \mathbb{R}^+ - \{1\},$$

where

$$s = m^k = m^{x+y}. \tag{2}$$

S_0 is the zeroth-number space; x and y are elements of the first number space S_1; γ lies in the second number space S_2, $S_2 \subseteq S_1$; α is a parameter of the distribution ratio, which is an element of the third number space S_3; and $\mathbb{R}^+ \cup \{0\}$ is the nonnegative real-number space.

We then apply an additive decomposition [18] to s using the following two properties of additive decompositions:

Additive decomposition using a weighted average [19]:

Let u, v be the weights of the weighted average.

For $k \in S_1$, $u, v \in S_2$, and $u + v \neq 0$, because

$\alpha = u/(u+v)$ and $1 - \alpha = v/(u+v)$, then

$$k = \alpha k + (1-\alpha)k = \frac{u}{u+v}k + \frac{v}{u+v}k. \tag{3}$$

Additive decomposition of the zeroth element:

For $\gamma \in S_2$,

$$0 = (+\gamma) + (-\gamma). \tag{4}$$

Any real number k can be expressed using the additive operation as follows:

$$k = [\,k\,] + [\,0\,] = [\,\alpha k + (1-\alpha)k\,] + [\,(+\gamma) + (-\gamma)\,]$$
$$= [\,\alpha k + \gamma\,] + [\,(1-\alpha)k + (-\gamma)\,]. \tag{5}$$

This additive decomposition defines the ordered pair of cells $(x, y) \in {S_1}^2$ as follows:

$$(x, y) := [\alpha k, (1-\alpha)k\,] \times (\gamma, -\gamma) = [\alpha k + \gamma, (1-\alpha)k - \gamma],$$
$$k = x + y, \gamma = (1-\alpha)x - \alpha y \tag{6}.$$

B. Inverse cell conversion

2D cells can then be reverted to their original 1D form using

$$|(x, y)| := m^{x+y} \tag{7},$$

which is known as the "value" of the cell (x, y).

C. Definition of multiplication

The multiplicative operation $\times$ in a cell is defined as natural multiplication. For $S_0 = \mathbb{R}^+$, with $\mathbb{R}$ as the real-number space, $S_1, S_2 = \mathbb{R}$, if $x_1, x_2, y_1, y_2 \in S_1$ and the set of the cells $F = \mathbb{R}^2$. Multiplication is then provided by the mapping $\times\colon F \times F \longrightarrow F$, and the algebraic system $(F, \times)$ of the binary relation $(c_1, c_2) \in F \times F$ is defined as follows:

$$c_1 \times c_2 = (x_1, y_1) \times (x_2, y_2) := (x_1 + x_2, y_1 + y_2) \tag{8}.$$

Multiplication is associative and commutative; therefore, if $\mathrm{j} = 1, 2, 3, x_{\mathrm{j}}, y_{\mathrm{j}} \in S_1, n \in \mathbb{R}^+ - \{1\}$, $c_{\mathrm{j}} \in F, \alpha > 0, 1 - \alpha > 0$, and $c_{\mathrm{j}} = (x_{\mathrm{j}}, y_{\mathrm{j}})$, then

$$c_1 \times (c_2 \times c_3) = (c_1 \times c_2) \times c_3, \tag{9}$$
$$c_1 \times c_2 = c_2 \times c_1. \tag{10}$$

D. Definitions of binary operations like a vector space

In the real cell space $(F, +, \times)$, we define the binary operation on the field K by considering it as the real space $\mathbb{R}$; i.e., using the mapping $\circ : K \times F \longrightarrow K \times F$, we define the algebraic system $(F, +, \times, \circ)$ of the binary relation $(r, c) \in K \times F$ and the additive inverse using the ring structure of the field K.

Let $S_0 = S_1 = S_2 = S_3 = \mathbb{R}$, $r \in \mathbb{R}, x, y \in S_1, m \in \mathbb{R}^+ - \{1\}$, $c \in F = \mathbb{R}^2$, the operation $+1 \circ (x, y)$ is shortened as +1, and the cell relationship is defined as $+1 \circ (x, y) := (x, y)$ (i.e., $1 \circ c = c$).

For certain cell value s, the definition is extended as follows:

$$|c| := rs = r|(x, y)| = rm^{x+y} \in \mathbb{R}. \tag{11}$$

For $\gamma \in S_2, \alpha \in S_3, \alpha > 0$, and $1 - \alpha > 0$, one has

$$c = r \circ (x, y) := rm^{x+y} \circ (\gamma, -\gamma) = |c| \circ (\gamma, -\gamma). \tag{12}$$

For $\mathrm{j} = 1, 2, r, r_{\mathrm{j}} \in \mathbb{R}, x, y, x_{\mathrm{j}}, y_{\mathrm{j}} \in S_1, c, c_{\mathrm{j}} \in F, \alpha > 0$, and $1 - \alpha > 0$, the following holds:

$$(r_1 + r_2) \circ c = r_1 \circ c + r_2 \circ c, \tag{13}$$
$$r_1(r_2 \circ c) = (r_1 r_2) \circ c, \tag{14}$$
$$r \circ (c_1 \times c_2) = (r \circ c_1) \times c_2. \tag{15}$$

When $r > 0$, we have

$$c = r \circ (x, y) = (\alpha \log_m r + x, (1-\alpha) \log_m r + y)$$

(16).

III. METHODS

A. Inverse factorization of Mersenne numbers

Because $m \in \mathbb{N}$, any natural number s can be expressed using a series expansion [20] of l terms, as shown in Equation 17 (i.e., as a sum of cell values), namely,

$$\begin{aligned} s &= \sum_{\mathrm{j}=0}^{l-1} r_{\mathrm{j}} m^{\mathrm{j}} \\ &= r_0 m^0 + r_1 m^1 + r_2 m^2 + \cdots + r_{l-1} m^{l-1} \\ &= r_0|c_0| + r_1|c_1| + r_2|c_2| + \cdots + r_{l-1}|c_{l-1}| \\ &= \textstyle\sum_{\mathrm{j}=0}^{l-1} r_{\mathrm{j}}|c_{\mathrm{j}}|, \end{aligned} \tag{17}$$

where r_{j} are included in $\mathbb{N} \cup \{0\}$.

Remark 3.1. *If s is a Mersenne number, all coefficients r_j are 1, and base $m = 2$ is assumed.*

Example 3.2. *If $m = 3$ and $s = 23$ in Equation 17, the series expansion of the cell values that contain only the second component is*

$$\begin{aligned} 23 &= 2 \cdot 3^0 + 1 \cdot 3^1 + 2 \cdot 3^2 \\ &= 2 \cdot |(0,0)| + 1 \cdot |(0,1)| + 2 \cdot |(0,2)|. \end{aligned}$$

Theorem 3.3. *Let $p_y, q_x \in \mathbb{N} \cup \{0\}$ for all $x, y \in \mathbb{N} \cup \{0\}, m \in \mathbb{N} - \{1\}$ and denote $p = \sum_{y=0}^{l_b-1} p_y m^y$ and $q = \sum_{x=0}^{l_a-1} q_x m^x$. Then, we have*

$$\begin{aligned} pq = &\sum_{k=0}^{l_b-1} \sum_{y=0}^{k} p_y q_{k-y} |(k-y, y)| \\ &+ \sum_{k=l_b}^{l_a-2} \sum_{y=0}^{l_b-1} p_y q_{k-y} |(k-y, y)| \\ &+ \sum_{k=l_a-1}^{n-1} \sum_{y=k-(l_a-1)}^{l_b-1} p_y q_{k-y} |(k-y, y)|, \end{aligned} \tag{18}$$

where $x + y = k \in \mathbb{N} \cup \{0\}$, $n = l_a + l_b - 1$ (Fig. 1 for details).

Proof. *Given $p = \sum_{y=0}^{l_b-1} p_y m^y = \sum_{y=0}^{l_b-1} p_y |(0, y)|$ and $q = \sum_{x=0}^{l_a-1} q_x m^x = \sum_{x=0}^{l_a-1} q_x |(x, 0)|$,*

$$pq = \sum_{y=0}^{l_b-1} p_y |(0, y)| \sum_{x=0}^{l_a-1} q_x |(x, 0)|$$

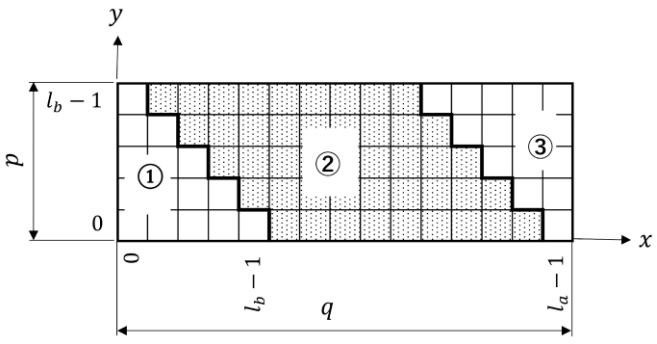

Fig. 1. View of three-region decomposition using multiplication in cell space (each square represents one cell)
$pq = ① + ② + ③$

$$\begin{aligned} &= \sum_{y=0}^{l_b-1} \sum_{k-y=0}^{l_a-1} p_y q_{k-y} |(k-y, y)| \\ &= \sum_{k=0}^{l_b-1} \sum_{y=0}^{k} p_y q_{k-y} |(k-y, y)| \\ &\quad + \sum_{k=l_b}^{l_a-2} \sum_{y=0}^{l_b-1} p_y q_{k-y} |(k-y, y)| \\ &\quad + \sum_{k=l_a-1}^{n-1} \sum_{y=k-(l_a-1)}^{l_b-1} p_y q_{k-y} |(k-y, y)|, \end{aligned}$$

where $x + y = k \in \mathbb{N} \cup \{0\}, n = l_a + l_b - 1$. □

Remark 3.4. *If $m = 2$ in Equation 18, $p_{l_b-1} = q_{l_a-1} = 1$. Moreover, both p_y and q_x can only consider the values 0 and 1. In particular, if pq is odd, $p_0 = q_0 = 1$.*

Here, $k \geq 0$, and m^k corresponds to the place of pq, $x + y = k, k = 0, 1, 2, \ldots, l_a - 1, \ldots, l_a + l_b - 2$. Furthermore, $n = l_a + l_b - 1$. However, l_a and n remain unknown if only p is specified.

Example 3.5. *Consider a case of a Mersenne number with an odd prime exponent. If $m = 3$, $p = 23$, and $q = 89$ in Equation 18, $pq = 2047 = M_{11}$, as is shown below:*

$$\begin{aligned} pq = &\sum_{k=0}^{2} \sum_{y=0}^{k} p_y q_{k-y} |(k-y, y)| \\ &+ \sum_{k=3}^{3} \sum_{y=0}^{2} p_y q_{k-y} |(k-y, y)| \\ &+ \sum_{k=4}^{6} \sum_{y=k-4}^{2} p_y q_{k-y} |(k-y, y)|, \end{aligned}$$

$$\begin{aligned} p_0 q_0 = 4, p_0 q_1 = 4, p_1 q_0 = 2, p_0 q_2 = 0, p_1 q_1 = 2, p_2 q_0 \\ = 4, \\ p_0 q_3 = 0, p_1 q_2 = 0, p_2 q_1 = 4, \\ p_0 q_4 = 2, p_1 q_3 = 0, p_2 q_2 = 0, p_1 q_4 = 1, p_2 q_3 = \\ 0, p_2 q_4 = 2. \end{aligned}$$

We use the representations of p:

$$\begin{aligned} p = 23 &= 2 \cdot |(0,0)| + 1 \cdot |(0,1)| + 2 \cdot |(0,2)| \\ &= \sum_{y=0}^{2} p_y |(0, y)|, \end{aligned}$$

where $p_0 = 2, p_1 = 1, p_2 = 2, l_b - 1 = 2$, and

$$\begin{aligned} q = 89 = 2 \cdot |(0,0)| + 2 \cdot |(1,0)| + 0 \cdot |(2,0)| + 0 \cdot |(3,0)| \\ + 1 \cdot |(4,0)| = \sum_{x=0}^{4} q_x |(x, 0)|, \end{aligned}$$

where $q_0 = 2, q_1 = 2, q_2 = 0, p_3 = 0, q_4 = 1, l_a - 1 = 4$, $n - 1 = 6$.

Consequently,

$$\begin{aligned} pq &= \sum_{y=0}^{2} p_y |(0, y)| \sum_{x=0}^{4} q_x |(x, 0)| \\ &= \sum_{y=0}^{2} \sum_{k-y=0}^{4} p_y q_{k-y} |(k-y, y)| \end{aligned}$$

$$= \sum_{k=0}^{2}\sum_{y=0}^{k} p_y q_{k-y}|(k-y,y)| + \sum_{k=3}^{3}\sum_{y=0}^{2} p_y q_{k-y}|(k-y,y)| + \sum_{k=4}^{6}\sum_{y=k-4}^{2} p_y q_{k-y}|(k-y,y)|$$

$$= p_0 q_0|(0,0)| + \sum_{y=0}^{1} p_y q_{1-y}|(1-y,y)| + \sum_{y=0}^{2} p_y q_{2-y}|(2-y,y)| + \sum_{y=0}^{2} p_y q_{3-y}|(3-y,y)| + \sum_{y=0}^{2} p_y q_{4-y}|(4-y,y)| + \sum_{y=1}^{2} p_y q_{5-y}|(5-y,y)| + \sum_{y=2}^{2} p_y q_{6-y}|(6-y,y)|$$

$$\begin{aligned}
&= p_0 q_0|(0,0)| \\
&+(p_0 q_1|(1,0)| + p_1 q_0|(0,1)|) \\
&+(p_0 q_2|(2,0)| + p_1 q_1|(1,1)| + p_2 q_0|(0,2)|) \\
&+(p_0 q_3|(3,0)| + p_1 q_2|(2,1)| + p_2 q_1|(1,2)|) \\
&+(p_0 q_4|(4,0)| + p_1 q_3|(3,1)| + p_2 q_2|(2,2)|) \\
&+(p_1 q_4|(4,1)| + p_2 q_3|(3,2)|) \\
&+p_2 q_4|(4,2)|
\end{aligned}$$

In this example, the sum of coefficients of each digit k, which is

$$\sigma_k := \begin{cases} \sum_{y=0}^{k} p_y q_{k-y} & (k \le l_b - 1) \\ \sum_{y=0}^{l_b-1} p_y q_{k-y} & (l_b \le k \le l_a - 2) \\ \sum_{y=k-(l_a-1)}^{l_b-1} p_y q_{k-y} & (l_a - 1 \le k \le l_a + l_b - 2) \end{cases} \quad (19)$$

can be determined as follows:

$$\sigma_0 = p_0 q_0 = 4$$
$$\sigma_1 = p_0 q_1 + p_1 q_0 = 4 + 2 = 6$$
$$\sigma_2 = p_0 q_2 + p_1 q_1 + p_2 q_0 = 0 + 2 + 4 = 6$$
$$\sigma_3 = p_0 q_3 + p_1 q_2 + p_2 q_1 = 0 + 0 + 4 = 4$$
$$\sigma_4 = p_0 q_4 + p_1 q_3 + p_2 q_2 = 2 + 0 + 0 = 2$$
$$\sigma_5 = p_1 q_4 + p_2 q_3 = 1 + 0 = 1, \sigma_6 = p_2 q_4 = 2$$

When $\sigma_k \ge m$, a carry-up is required. The carry-up is the operation of adding $\lfloor \sigma_k/m \rfloor$ to the next digit σ_{k+1} and executing $\sigma_k - \lfloor \sigma_k/m \rfloor$.

Example 3.6. *We then use a binary representation because of the nature of Mersenne numbers and show the application of Equation 19. If $m = 2, p = 23, q = 89$ in Equation 18, $pq = 2047 = M_{11}$, we have*

$$p = 23 = 1\cdot|(0,0)| + 1\cdot|(0,1)| + 1\cdot|(0,2)| + 0\cdot|(0,3)| + 1\cdot|(0,4)| = \sum_{y=0}^{4} p_y|(0,y)|,$$

where $p_0 = 1, p_1 = 1, p_2 = 1, p_3 = 0, p_4 = 1, l_b - 1 = 4$, and

$$q = 89 = 1\cdot|(0,0)| + 0\cdot|(1,0)| + 0\cdot|(2,0)| + 1\cdot|(3,0)| + 1\cdot|(4,0)| + 0\cdot|(5,0)| + 1\cdot|(6,0)| = \sum_{x=0}^{6} q_x|(x,0)|,$$

where $q_0 = 1, q_1 = 0, q_2 = 0, p_3 = 1, q_4 = 1, q_5 = 0, q_6 = 1, l_a - 1 = 6, n - 1 = 10$.

Consequently,

$$pq = \sum_{y=0}^{4} p_y|(0,y)| \sum_{x=0}^{6} q_x|(x,0)| = \sum_{k=0}^{4}\sum_{y=0}^{k} p_y q_{k-y}|(k-y,y)| + \sum_{k=5}^{5}\sum_{y=0}^{4} p_y q_{k-y}|(k-y,y)| + \sum_{k=6}^{10}\sum_{y=k-6}^{4} p_y q_{k-y}|(k-y,y)|$$

$$\begin{aligned}
&= p_0 q_0|(0,0)| \\
&+(p_0 q_1|(1,0)| + p_1 q_0|(0,1)|) \\
&+(p_0 q_2|(2,0)| + p_1 q_1|(1,1)| + p_2 q_0|(0,2)|) \\
&+(p_0 q_3|(3,0)| + p_1 q_2|(2,1)| + p_2 q_1|(1,2)| + p_3 q_0|(0,3)|) \\
&+(p_0 q_4|(4,0)| + p_1 q_3|(3,1)| + p_2 q_2|(2,2)| + p_3 q_1|(1,3)| + p_4 q_0|(0,4)|) \\
&+(p_0 q_5|(5,0)| + p_1 q_4|(4,1)| + p_2 q_3|(3,2)| + p_3 q_2|(2,3)| + p_4 q_1|(1,4)|) \\
&+(p_0 q_6|(6,0)| + p_1 q_5|(5,1)| + p_2 q_4|(4,2)| + p_3 q_3|(3,3)| + p_4 q_2|(2,4)|) \\
&+(p_1 q_6|(6,1)| + p_2 q_5|(5,2)| + p_3 q_4|(4,3)| + p_4 q_3|(3,4)|) \\
&+(p_2 q_6|(6,2)| + p_3 q_5|(5,3)| + p_4 q_4|(4,4)|) \\
&+(p_3 q_6|(6,3)| + p_4 q_5|(5,4)|) \\
&+p_4 q_6|(6,4)|.
\end{aligned}$$

The coefficients of each digit sum are as follows:

$$\begin{aligned}
&\sigma_0 = p_0 q_0 = 1, \\
&\sigma_1 = p_0 q_1 + p_1 q_0 = 0 + 1 = 1, \\
&\sigma_2 = p_0 q_2 + p_1 q_1 + p_2 q_0 = 0 + 0 + 1 = 1, \\
&\sigma_3 = p_0 q_3 + p_1 q_2 + p_2 q_1 + p_3 q_0 = 1 + 0 + 0 + 0 = 1, \\
&\sigma_4 = p_0 q_4 + p_1 q_3 + p_2 q_2 + p_3 q_1 + p_4 q_0 = 1 + 1 + 0 + 0 + 1 = 3, \\
&\sigma_5 = p_0 q_5 + p_1 q_4 + p_2 q_3 + p_3 q_2 + p_4 q_1 = 0 + 1 + 1 + 0 + 0 = 2, \\
&\sigma_6 = p_0 q_6 + p_1 q_5 + p_2 q_4 + p_3 q_3 + p_4 q_2 = 1 + 0 + 1 + 0 + 0 = 2, \\
&\sigma_7 = p_1 q_6 + p_2 q_5 + p_3 q_4 + p_4 q_3 = 1 + 0 + 0 + 1 = 2, \\
&\sigma_8 = p_2 q_6 + p_3 q_5 + p_4 q_4 = 1 + 0 + 1 = 2, \\
&\sigma_9 = p_3 q_6 + p_4 q_5 = 0 + 0 = 0, \\
&\sigma_{10} = p_4 q_6 = 1.
\end{aligned}$$

Then, let $R_k \in \mathbb{N} \cup \{0\}$ be the carry-up from digit k to digit $k+1$ and let $T_k \in \mathbb{N} \cup \{0\}$ be the sum of the total cell values of digit k and R_{k-1}. However, for convenience, we then set $R_{-1} := 0$, which yields the following:

$$T_k = R_{k-1} + \sigma_k. \quad (20)$$

Furthermore, let $V_k \in \mathbb{N} \cup \{0\}$ be the difference between T_k and the carry-up R_k; i.e.,

$$V_k = T_k - mR_k. \quad (21)$$

Because p, p_y, and l_b are known, q_{k-y} and n are determined based on the property that all coefficients of Mersenne numbers are unity for the base $m = 2$. Thus, all $V_k = 1$, and the initial values are $R_0 = 0$, $V_0 = 1$, and $T_0 = 1$. R_k, T_k, and V_k, are sequentially calculated. If q_k is determined such that $V_k = 1$ for k, V_{k+1} to V_{k+l_b-1} are unity, $R_{k+l_b-1} = 0$, and the calculation is finished.

Note that k is maximal when the calculation ends, which is when k = $n - 1$. Usually, M_n, in which we are interested, is

the positional system [21] of V_k in binary.

Because $p_0 = 1$, we have $q_k = 0$ or 1, and $T_k \equiv V_k \equiv 1 \pmod 2$. Subsequently, we can determine that

$$R_k = \lfloor T_k/m \rfloor. \quad (22)$$

Thus, the Mersenne numbers M_n are calculated as follows:

$$M_n = pq = p\sum_{x=0}^{l_a-1} q_x|(x,0)|. \quad (23)$$

Moreover, this algorithm can calculate the Mersenne number with p^2 rather than the specified prime factor p. Even if p is an odd number that is not a prime, the inverse factorization of the Mersenne numbers is possible. If the specified p is prime and $l_a = 1$, p is a Mersenne prime.

Proposition 3.7. *Let $p_y, q_x \in \mathbb{N} \cup \{0\}$ for all $x, y \in \mathbb{N} \cup \{0\}$, $p = \sum_{y=0}^{l_b-1} p_y|(0,y)|$, and $q = \sum_{x=0}^{l_a-1} q_x|(x,0)|$ for a base $m \in \mathbb{N} - \{1\}$, and let $pq = \sum_{y=0}^{l_b-1} p_y|(0,y)| \sum_{x=0}^{l_a-1} q_x|(x,0)|$.*
$R_{l_a+l_b-2}$ is then less than $m-1$.

Proof. *Because $R_{l_a+l_b-2}$ is a maximum when $p_y, q_x = m-1$, each coefficient in Equation 18 is $(m-1)^2$.*

When $k = 0$, $\sigma_0 = T_0 = (m-1)^2$ and $R_0 = m-2, V_0 = 1$.

When $1 \le k \le l_b - 1$,
$\sigma_k = (k+1)(m-1)^2$,
$T_k = (k+1)m^2 - (k+2)m$,
$R_k = (k+1)m - (k+2)$,
$V_k = 0$.

When $l_b \le k \le l_a - 1$,
$\sigma_k = l_b(m-1)^2$
$T_k = l_b m^2 - l_b m - 1$,
$R_k = l_b m - (l_b + 1)$,
$V_k = m - 1$.

When $l_a \le k \le l_a + l_b - 2$,
$\sigma_k = (l_b - k + l_a - 1)(m-1)^2$,
$T_k = (l_b - k + l_a - 1)m^2 - (l_b - k + l_a - 1)m - V_k$,
$R_k = (l_b - k + l_a - 1)(m-1)$,

$$V_k = \begin{cases} m-2 \ (k = l_a) \\ m-1 \ (l_a + 1 \le k \le l_a + l_b - 2). \end{cases}$$

Therefore, $R_{l_a+l_b-2} \le m-1$. □

Corollary 3.8. *Let $k_a \in \mathbb{N} \cup \{0\}, 0 \le k_a \le l_a - 1$. If all $V_k = q_k$ are determined for $k \le k_a$, $q_{(2)} = V_{k_a}V_{k_a-1}\cdots V_1V_0$, and $pq_{(2)} = R_{k_a+l_b-1}V_{k_a+l_b-1}V_{k_a+l_b-2}\cdots V_1V_0$, and then $R_{k_a+l_b-1} \le m-1$ for $q_{k_a} \ne 0$.*

Example 3.9. *If $m = 2$, $p = 23, q = 89$, and $pq = 2047 = M_{11}$, we apply the carry-up technique below.*

Using the results of Example 4, one has

$T_0 = R_{-1} + \sigma_0 = 0 + 1 = 1, R_0 = \lfloor T_0/2 \rfloor = \lfloor 1/2 \rfloor = 0, V_0 = T_0 - 2R_0 = 1 - 0 = 1,$
$T_1 = R_0 + \sigma_1 = 0 + 1 = 1, R_1 = \lfloor T_1/2 \rfloor = \lfloor 1/2 \rfloor = 0, V_1 = T_1 - 2R_1 = 1 - 0 = 1,$
$T_2 = R_1 + \sigma_2 = 0 + 1 = 1, R_2 = \lfloor T_2/2 \rfloor = \lfloor 1/2 \rfloor = 0, V_2 = T_2 - 2R_2 = 1 - 0 = 1,$
$T_3 = R_2 + \sigma_3 = 0 + 1 = 1, R_3 = \lfloor T_3/2 \rfloor = \lfloor 1/2 \rfloor = 0, V_3 = T_3 - 2R_3 = 1 - 0 = 1,$
$T_4 = R_3 + \sigma_4 = 0 + 3 = 3, R_4 = \lfloor T_4/2 \rfloor = \lfloor 3/2 \rfloor = 1, V_4 = T_4 - 2R_4 = 3 - 2 = 1,$
$T_5 = R_4 + \sigma_5 = 1 + 2 = 3, R_5 = \lfloor T_5/2 \rfloor = \lfloor 3/2 \rfloor = 1, V_5 = T_5 - 2R_5 = 3 - 2 = 1,$
$T_6 = R_5 + \sigma_6 = 1 + 2 = 3, R_6 = \lfloor T_6/2 \rfloor = \lfloor 3/2 \rfloor = 1, V_6 = T_6 - 2R_6 = 3 - 2 = 1,$
$T_7 = R_6 + \sigma_7 = 1 + 2 = 3, R_7 = \lfloor T_7/2 \rfloor = \lfloor 3/2 \rfloor = 1, V_7 = T_7 - 2R_7 = 3 - 2 = 1,$
$T_8 = R_7 + \sigma_8 = 1 + 2 = 3, R_8 = \lfloor T_8/2 \rfloor = \lfloor 3/2 \rfloor = 1, V_8 = T_8 - 2R_8 = 3 - 2 = 1,$
$T_9 = R_8 + \sigma_9 = 1 + 0 = 1, R_9 = \lfloor T_9/2 \rfloor = \lfloor 1/2 \rfloor = 0, V_9 = T_9 - 2R_9 = 1 - 0 = 1,$
$T_{10} = R_9 + \sigma_{10} = 0 + 1 = 1, R_{10} = \lfloor T_{10}/2 \rfloor = \lfloor 1/2 \rfloor = 0, V_{10} = T_{10} - 2R_{10} = 1 - 0 = 1.$

Therefore, using the binary positional system, $pq_{(2)} = V_{10}V_9V_8V_7V_6V_5V_4V_3V_2V_1V_0 = 11111111111_{(2)} = 2^{11} - 1$.

Then, the framework of Theorem 3.3 is restrained to $p_{l_b-1}q_{l_a-1} \ne 0$ and $0 \le p_y, q_x \le m-1$ for all $x, y \in \mathbb{N} \cup \{0\}$.

Theorem 3.10. *When p and q are odd, consider the composite Mersenne number $M_n = 2^n - 1 = pq$. The exponent n of the minimum Mersenne number that has the specified factor p and another factor q can be uniquely determined.*

Proof. *Because $m = 2$, we obtain for the specified factor p*

$$pq = \sum_{k=0}^{l_b-1}\sum_{y=0}^{k} p_y q_{k-y}|(k-y,y)| + \sum_{k=l_b}^{l_a-2}\sum_{y=0}^{l_b-1} p_y q_{k-y}|(k-y,y)| + \sum_{k=l_a-1}^{n-1}\sum_{y=k-(l_a-1)}^{l_b-1} p_y q_{k-y}|(k-y,y)|.$$

$$\begin{aligned}
&= p_0q_0|(0,0)| \\
&+(p_0q_1|(1,0)| + p_1q_0|(0,1)|) \\
&+(p_0q_2|(2,0)| + p_1q_1|(1,1)| + p_2q_0|(0,2)|) \\
&+(p_0q_3|(3,0)| + p_1q_2|(2,1)| + p_2q_1|(1,2)| \\
&\qquad + p_3q_0|(0,3)|) \\
&\qquad \vdots \\
&+\big(p_0q_{l_b-1}|(l_b-1,0)| + p_1q_{l_b-2}|(l_b-2,1)| + \cdots \\
&\qquad + p_{l_b-2}q_1|(1,l_b-2)| \\
&\qquad + p_{l_b-1}q_0|(0,l_b-1)|\big) \\
&+\big(p_0q_{l_b}|(l_b,0)| + p_1q_{l_b-1}|(l_b-1,1)| + \cdots \\
&\qquad + p_{l_b-2}q_2|(2,l_b-2)| \\
&\qquad + p_{l_b-1}q_1|(1,l_b-1)|\big) \\
&+\big(p_0q_{l_b+1}|(l_b+1,0)| + p_1q_{l_b}|(l_b,1)| + \cdots \\
&\qquad + p_{l_b-2}q_3|(3,l_b-2)| \\
&\qquad + p_{l_b-1}q_2|(2,l_b-1)|\big) \\
&\qquad \vdots \\
&+\big(p_0q_{l_a-3}|(l_a-3,0)| + p_1q_{l_a-4}|(l_a-4,1)| + \cdots \\
&\qquad + p_{l_b-2}q_{l_a-l_b-1}|(l_a-l_b-1,l_b-2)| \\
&\qquad + p_{l_b-1}q_{l_a-l_b-2}|(l_a-l_b-2,l_b-1)|\big) \\
&+\big(p_0q_{l_a-2}|(l_a-2,0)| + p_1q_{l_a-3}|(l_a-3,1)| + \cdots \\
&\qquad + p_{l_b-2}q_{l_a-l_b}|(l_a-l_b,l_b-2)| \\
&\qquad + p_{l_b-1}q_{l_a-l_b-1}|(l_a-l_b-1,l_b-1)|\big)
\end{aligned}$$

$$+\big(p_0q_{l_a-1}|(l_a-1,0)|+p_1q_{l_a-2}|(l_a-2,1)|$$
$$+\cdots+p_{l_b-2}q_{l_a-l_b+1}|(l_a-l_b+1,l_b-2)|$$
$$+p_{l_b-1}q_{l_a-l_b}|(l_a-\ l_b,l_b-1)|\big)$$
$$\vdots$$
$$+\big(p_{l_b-3}q_{l_a-1}|(l_a-1,l_b-3)|$$
$$+p_{l_b-2}q_{l_a-2}|(l_a-2,l_b-2)|$$
$$+p_{l_b-1}q_{l_a-3}|(l_a-3,l_b-1)|\big)$$
$$+\big(p_{l_b-2}q_{l_a-1}|(l_a-1,l_b-2)|$$
$$+p_{l_b-1}q_{l_a-2}|(l_a-2,l_b-1)|\big)$$
$$+p_{l_b-1}q_{l_a-1}|(l_a-1,l_b-1)|.$$

Moreover,

$$\sigma_0=p_0q_0,$$
$$\sigma_1=p_0q_1+p_1q_0,$$
$$\sigma_2=p_0q_2+p_1q_1+p_2q_0,$$
$$\sigma_3=p_0q_3+p_1q_2+p_2q_1+p_3q_0,$$
$$\vdots$$
$$\sigma_{l_b-1}=p_0q_{l_b-1}+p_1q_{l_b-2}+\cdots+p_{l_b-2}q_1+p_{l_b-1}q_0,$$
$$\sigma_{l_b}=p_0q_{l_b}+p_1q_{l_b-1}+\cdots+p_{l_b-2}q_2+p_{l_b-1}q_1,$$
$$\sigma_{l_b+1}=p_0q_{l_b+1}+p_1q_{l_b}+\cdots+p_{l_b-2}q_3+p_{l_b-1}q_2,$$
$$\vdots$$
$$\sigma_{l_a-3}=p_0q_{l_a-3}+p_1q_{l_a-4}+\cdots+p_{l_b-2}q_{l_a-l_b-1}$$
$$+p_{l_b-1}q_{l_a-l_b-2},$$
$$\sigma_{l_a-2}=p_0q_{l_a-2}+p_1q_{l_a-3}+\cdots+p_{l_b-2}q_{l_a-l_b}$$
$$+p_{l_b-1}q_{l_a-l_b-1},$$
$$\sigma_{l_a-1}=p_0q_{l_a-1}+p_1q_{l_a-2}+\cdots+p_{l_b-2}q_{l_a-l_b+1}$$
$$+p_{l_b-1}q_{l_a-l_b}$$
$$\vdots$$
$$\sigma_{l_a+l_b-4}=p_{l_b-3}q_{l_a-1}+p_{l_b-2}q_{l_a-2}+p_{l_b-1}q_{l_a-3}$$
$$\sigma_{l_a+l_b-3}=p_{l_b-2}q_{l_a-1}+p_{l_b-1}q_{l_a-2}$$
$$\sigma_{l_a+l_b-2}=p_{l_b-1}q_{l_a-1},$$

and

$$T_0=\sigma_0=1,R_0=\lfloor T_0/2\rfloor=0,V_0=T_0-2R_0=1,$$
$$T_1=R_0+\sigma_1,R_1=\lfloor T_1/2\rfloor,V_1=T_1-2R_1=1,$$
$$T_2=R_1+\sigma_2,R_2=\lfloor T_2/2\rfloor,V_2=T_2-2R_2=1,$$
$$T_3=R_2+\sigma_3,R_3=\lfloor T_3/2\rfloor,V_3=T_3-2R_3=1,$$
$$\vdots$$
$$T_{l_b-1}=R_{l_b-2}+\sigma_{l_b-1},$$
$$R_{l_b-1}=\lfloor T_{l_b-1}/2\rfloor,V_{l_b-1}=T_{l_b-1}-2R_{l_b-1}=1,$$
$$T_{l_b}=R_{l_b-1}+\sigma_{l_b},$$
$$R_{l_b}=\lfloor T_{l_b}/2\rfloor,V_{l_b}=T_{l_b}-2R_{l_b}=1,$$
$$T_{l_b+1}=R_{l_b}+\sigma_{l_b+1},$$
$$R_{l_b+1}=\lfloor T_{l_b+1}/2\rfloor,V_{l_b+1}=T_{l_b+1}-2R_{l_b+1}=1,$$
$$\vdots$$
$$T_{l_a-3}=R_{l_a-4}+\sigma_{l_a-3},$$
$$R_{l_a-3}=\lfloor T_{l_a-3}/2\rfloor,V_{l_a-3}=T_{l_a-3}-2R_{l_a-3}=1,$$
$$T_{l_a-2}=R_{l_a-3}+\sigma_{l_a-2},$$
$$R_{l_a-2}=\lfloor T_{l_a-2}/2\rfloor,V_{l_a-2}=T_{l_a-2}-2R_{l_a-2}=1,$$
$$T_{l_a-1}=R_{l_a-2}+\sigma_{l_a-1},$$
$$R_{l_a-1}=\lfloor T_{l_a-1}/2\rfloor,V_{l_a-1}=T_{l_a-1}-2R_{l_a-1}=1,$$
$$\vdots$$
$$T_{l_a+l_b-4}=R_{l_a+l_b-5}+\sigma_{l_a+l_b-4},$$
$$R_{l_a+l_b-4}=\lfloor T_{l_a+l_b-4}/2\rfloor,V_{l_a+l_b-4}=T_{l_a+l_b-4}-2R_{l_a+l_b-4}=1,$$
$$T_{l_a+l_b-3}=R_{l_a+l_b-4}+\sigma_{l_a+l_b-3},$$
$$R_{l_a+l_b-3}=\lfloor T_{l_a+l_b-3}/2\rfloor,V_{l_a+l_b-3}=T_{l_a+l_b-3}-2R_{l_a+l_b-3}=1,$$
$$T_{l_a+l_b-2}=R_{l_a+l_b-3}+\sigma_{l_a+l_b-2},$$
$$R_{l_a+l_b-2}=\lfloor T_{l_a+l_b-2}/2\rfloor,V_{l_a+l_b-2}=T_{l_a+l_b-2}-2R_{l_a+l_b-2}=1.$$

First, because $p_0,p_1,p_2,\dots,p_{l_b-1}$ are known and $p_0,q_0=1$, it is evident that $V_0=1$. Next, the unknown values q_1 and T_1 are either 0 or 1; however, these are uniquely determined because $V_1=1$ is required. Assuming that $q_1=0$, we can obtain that $q_1=0$ or 1 if T_1 is odd or even. Next, based on the determined q_1, T_1 is calculated again, and so R_1 and V_1 are calculated. We then repeat the procedure from q_2, T_2 to q_{l_a-1}, T_{l_a-1}, which can be similarly determined. Consequently, although l_a-1 is unknown, we can determine T_{l_a-1}, R_{l_a-1}, and V_{l_a-1} when $k=l_a-1$. Furthermore, using Corollary 3.8, we can determine all T_k, R_k, and V_k for $l_a\le k\le l_a+l_b-2$. The multiplication operation is complete when all $V_k=1$ for $l_a\le k\le l_a+l_b-2$ and $R_{l_a+l_b-2}=0$. The exponent of the minimum Mersenne number with the specified p as a factor is $n=l_a+l_b-1$. Using the binary positional system, we obtain another factor $q_{(2)}$ as $q_{l_a-1}q_{l_a-2}\cdots q_1q_0=V_{l_a-1}V_{l_a-2}\dots V_1V_0$. □

Theorem 3.10 shows that the inverse factorization of Mersenne numbers is possible; the corresponding algorithm is provided in Algorithm 3.11.

Algorithm 3.11: Inverse factorization of Mersenne numbers

INPUT: Specify an odd number p that is a factor of the given Mersenne number.

OUTPUT: The decimal exponent n of the Mersenne number with the decimal factor p and another binary factor q.

1: Specify the divisor $p_{(10)}$ in decimals.

2: Express p in a binary expansion, i.e., a series expansion with the term number l_b:
$p_{(10)}=\sum_{y=0}^{l_b-1}p_y\cdot 2^y=p_{l_b-1}\,p_{l_b-2}\cdots\,p_{0\,(2)}$.

3: Assign each digit of p in the binary positional system to the cell $p_yq_0\circ(0,y)$.
For $y=0$ to l_b-1; $l_b-1=\lfloor\log_2 p_{(10)}\rfloor$
$p_y\leftarrow p_y$; initial condition
Next y

4: Let $R_{-1}=0$, $\sigma_0=0$, and $T_0=0$. Let $V_c=$ null.

5: **For** $k=0$ to $p-1$

6: Determine q_k under the condition $V_k=1$.
Note that any coefficient of a digit with k greater than $k=k_a$ in the calculation is set to zero.
If T_k is odd, **then**
For $y=0$ to l_b-1
$q_k\leftarrow\ 0$; thus $p_yq_k\leftarrow\ 0$
Next y
Else if T_k is even, **then**
For $y=0$ to l_b-1
$q_k\leftarrow 1$; thus $p_yq_k\leftarrow p_y$
Next y
End if

7: Calculate $\sigma_k,T_k,$ and R_k. Let $V_c=1$.
For $y=0$ to l_b-1,
$\sigma_{k+y}\leftarrow\sigma_{k+y}+p_yq_k$,

$T_{k+y} \leftarrow R_{k+y-1} + \sigma_{k+y}$,
$R_{k+y} \leftarrow \lfloor T_{k+y}/2 \rfloor$,
$V_{k+y} \leftarrow T_{k+y} - 2\,R_{k+y}$,
$V_c \leftarrow V_c * V_{k+y}$. This step is the preparation to examine the completion of the calculation using Corollary 3.8.

Next y

8: Obtain a string of divisors $q_{(2)}$ and let $\mathrm{arr}(k)$ be a string variable.
$\mathrm{arr}(k) \leftarrow q_k \& \mathrm{arr}(k)$, where "&" is the string concatenation operator.

9: Use Corollary 3.8 to examine the completion of the calculation.
If $V_c = 1$, and $R_{k+l_b-1} = 0$, **then go to** 11

10: **Next** k

11: Output $n = k + l_b$

12: Output $\mathrm{arr}(k)$ as $q_{(2)}$, which is a positional system in binary representation in the order of decreasing exponent.
Or convert $q_{(2)}$ to a decimal using the binary expansion and output $q_{(10)}$.

13: **End**

Corollary 3.12. *For the base $m = 2$, let p_y, q_x for all $x, y \in \mathbb{N} \cup \{0\}, p = \sum_{y=0}^{l_b-1} p_y |(0,y)|$, and $q = \sum_{x=0}^{l_a-1} q_x |(x,0)|$. When the minimum Mersenne number $pq = 2^n - 1$ and $n = l_a + l_b - 1$ with the specified factor p is known, all q_{k_a} such that $l_a \le k_a \le l_a + l_b - 2$ are zero. Therefore, $q_{l_a}, q_{l_a+1}, \ldots, q_{l_a+l_b-2}$ are zero.*

Proof. *From Theorem 3.10, the specified p with the highest order $l_b - 1$ yields another factor q such that all $V_{k_a} = 1$ for $l_a \le k_a \le l_a + l_b - 2$. The highest order of q is $l_a - 1$, and $pq = 2^{l_a+l_b-1} - 1$. Therefore, all q_{k_a} such that $l_a \le k_a \le l_a + l_b - 2$ are zero.* □

In Example 3.13, applying Corollary 3.8 to a Mersenne number and confirming the completion of the calculation complicates the explanation, we apply Corollary 3.12 instead.

Example 3.13. *Let us use inverse factorization to determine the exponent n of the minimum Mersenne number with $p = 23$ as a factor and another divisor q. We have*

$$p = 23 = 10111_{(2)},$$

$l_b - 1 = 4$,

$$p_0|(0,0)| = 1 \cdot |(0,0)|, p_1|(0,1)| = 1 \cdot |(0,1)|, p_2|(0,2)| = 1 \cdot |(0,2)|, p_3|(0,3)| = 0 \cdot |(0,3)|, p_4|(0,4)| = 1 \cdot |(0,4)|.$$

Thus, $p_0 = 1, p_1 = 1, p_2 = 1, p_3 = 0, p_4 = 1$.

We then consider each of these values in turn.
When $k = 0, q_0 = 0$, then $\sigma_0 = p_0 q_0 = 0$,
$T_0 = R_{-1} + \sigma_0 = 0 + 0 = 0$, which is even.
Therefore, $q_0 = 1$ is determined, *and* so $p_0 q_0 = 1$, $T_0 = 1$, and

$$R_0 = \lfloor T_0/2 \rfloor = \lfloor 1/2 \rfloor = 0.$$

Thus, $\sigma_0 = 1, T_0 = 1, R_0 = 0, V_0 = 1$.

When $k = 1$, if $q_1 = 0$, then $\sigma_1 = p_0 q_1 + p_1 q_0 = 0 + 1 = 1$,
$T_1 = R_0 + \sigma_1 = 0 + 1 = 1$, which is odd.
Consequently, $q_1 = 0$ is determined, so $p_0 q_1 = p_1 q_1 = p_2 q_1 = p_3 q_1 = p_4 q_1 = 0$, $T_1 = 1$, and

$$R_1 = \lfloor T_1/2 \rfloor = \lfloor 1/2 \rfloor = 0.$$

Hence,

$$\sigma_1 = 1, T_1 = 1, R_1 = 0, V_1 = 1.$$

Remark 3.14. *Consider the case where q_k is assumed to be 1 rather than 0. If $q_k = 1$, we obtain $V_k = 1$ when T_k is odd, so $q_k = 1$. If $q_k = 1$, we obtain $V_k = 0$ when T_k is even, so $q_k = 0$.*

When $k = 1$, if $q_1 = 1$, then $\sigma_1 = p_0 q_1 + p_1 q_0 = 1 + 1 = 2$,
$T_1 = R_0 + \sigma_1 = 0 + 2 = 2$, which is even.
Thus, $q_1 = 0$ is determined, $p_0 q_1 = p_1 q_1 = p_2 q_1 = p_3 q_1 = p_4 q_1 = 0$, $T_1 = 1$,
$R_1 = \lfloor T_1/2 \rfloor = \lfloor 1/2 \rfloor = 0$,

$$\sigma_1 = 1, T_1 = 1, R_1 = 0, V_1 = 1.$$

When $k = 2$ and $q_2 = 0$, one has $\sigma_2 = p_0 q_2 + p_1 q_1 + p_2 q_0 = 0 + 0 + 1 = 1$, followed by
$T_2 = R_1 + \sigma_2 = 0 + 1 = 1$, which is again an odd number.
Thus, $q_2 = 0$ is determined, $p_0 q_2 = p_1 q_2 = p_2 q_2 = p_3 q_2 = p_4 q_2 = 0$, $T_2 = 1$,

$$R_2 = \lfloor T_2/2 \rfloor = \lfloor 1/2 \rfloor = 0,$$

$$\sigma_2 = 1, T_2 = 1, R_2 = 0, V_2 = 1.$$

When $k = 3$, if $q_3 = 0$, $\sigma_3 = p_0 q_3 + p_1 q_2 + p_2 q_1 + p_3 q_0 = 0 + 0 + 0 + 0 = 0$,
$T_3 = R_2 + \sigma_3 = 0 + 0 = 0$, which is even.
Thus, $q_3 = 1, p_0 q_3 = 1, p_1 q_3 = 1, p_2 q_3 = 1, p_3 q_3 = 0, p_4 q_3 = 1, T_3 = 1$,
$R_3 = \lfloor T_3/2 \rfloor = \lfloor 1/2 \rfloor = 0$,
$\sigma_3 = 1, T_3 = 1, R_3 = 0, V_3 = 1$.

Finally, when $k = 4$ and $q_4 = 0$, we obtain $\sigma_4 = p_0 q_4 + p_1 q_3 + p_2 q_2 + p_3 q_1 + p_4 q_0 = 0 + 1 + 0 + 0 + 1 = 2$, followed by
$T_4 = R_3 + \sigma_4 = 0 + 2 = 2$, which is even.
Thus, $q_4 = 1, p_0 q_4 = 1, p_1 q_4 = 1, p_2 q_4 = 1, p_3 q_4 = 0, p_4 q_4 = 1, T_4 = 3$,
$R_4 = \lfloor T_4/2 \rfloor = \lfloor 3/2 \rfloor = 1$,

$$\sigma_4 = 3, T_4 = 3, R_4 = 1, V_4 = 1.$$

When $k = 5$ and $q_5 = 0$, $\sigma_5 = p_0 q_5 + p_1 q_4 + p_2 q_3 + p_3 q_2 + p_4 q_1 = 0 + 1 + 1 + 0 + 0 = 2$, and
$T_5 = R_4 + \sigma_5 = 1 + 2 = 3$, which is odd.
Thus, $q_5 = 0, p_0 q_5 = p_1 q_5 = p_2 q_5 = p_3 q_5 = p_4 q_5 = 0$, $T_5 = 3$,
$R_5 = \lfloor T_5/2 \rfloor = \lfloor 3/2 \rfloor = 1$,
$\sigma_5 = 2, T_5 = 3, R_5 = 1, V_5 = 1$.

When $k = 6$ and $q_6 = 0$, $\sigma_6 = p_0 q_6 + p_1 q_5 + p_2 q_4 + p_3 q_3 + p_4 q_2 = 0 + 0 + 1 + 0 + 0 = 1$, and
$T_6 = R_5 + \sigma_6 = 1 + 1 = 2$, which is even.
Thus, $q_6 = 1, p_0 q_6 = 1, p_1 q_6 = 1, p_2 q_6 = 1, p_3 q_6 = 0, p_4 q_6 = 1, T_6 = 3$,
$R_6 = \lfloor T_6/2 \rfloor = \lfloor 3/2 \rfloor = 1$,
$\sigma_6 = 2, T_6 = 3, R_6 = 1, V_6 = 1$.

When $k = 7$ and $q_7 = 0$, then $\sigma_7 = p_0q_7 + p_1q_6 + p_2q_5 + p_3q_4 + p_4q_3 = 0 + 1 + 0 + 0 + 1 = 2$, and
$T_7 = R_6 + \sigma_7 = 1 + 2 = 3$, which is odd.
Thus, $q_7 = 0, p_0q_7 = p_1q_7 = p_2q_7 = p_3q_7 = p_4q_7 = 0$, $T_7 = 3$,
$R_7 = \lfloor T_7/2 \rfloor = \lfloor 3/2 \rfloor = 1$,
$\sigma_7 = 2, T_7 = 3, R_7 = 1, V_7 = 1$.

When $k = 8$ and $q_8 = 0$, then $\sigma_8 = p_0q_8 + p_1q_7 + p_2q_6 + p_3q_5 + p_4q_4 = 0 + 0 + 1 + 0 + 1 = 2$, and
$T_8 = R_7 + \sigma_8 = 1 + 2 = 3$, which is odd.
Thus, $q_8 = 0, p_0q_8 = p_1q_8 = p_2q_8 = p_3q_8 = p_4q_8 = 0, T_8 = 3$,
$R_8 = \lfloor T_8/2 \rfloor = \lfloor 3/2 \rfloor = 1$,
$\sigma_8 = 2, T_8 = 3, R_8 = 1, V_8 = 1$.

When $k = 9$ and $q_9 = 0$, then $\sigma_9 = p_0q_9 + p_1q_8 + p_2q_7 + p_3q_6 + p_4q_5 = 0 + 0 + 0 + 0 + 0 = 0$, and
$T_9 = R_8 + \sigma_9 = 1 + 0 = 1$, which is odd.
Thus, $q_9 = 0, p_0q_9 = p_1q_9 = p_2q_9 = p_3q_9 = p_4q_9 = 0, T_9 = 1$,
$R_9 = \lfloor T_9/2 \rfloor = \lfloor 1/2 \rfloor = 0$,
$\sigma_9 = 0, T_9 = 1, R_9 = 0, V_9 = 1$.

When $k = 10$ and $q_{10} = 0$, then
$\sigma_{10} = p_0q_{10} + p_1q_9 + p_2q_8 + p_3q_7 + p_4q_6 = 0 + 0 + 0 + 0 + 1 = 1$, and
$T_{10} = R_9 + \sigma_{10} = 0 + 1 = 1$, which is odd.
Thus, $q_{10} = 0, p_0q_{10} = p_1q_{10} = p_2q_{10} = p_3q_{10} = p_4q_{10} = 0, T_{10} = 1$,
$R_{10} = \lfloor T_{10}/2 \rfloor = \lfloor 1/2 \rfloor = 0$,
$\sigma_{10} = 1, T_{10} = 1, R_{10} = 0, V_{10} = 1$.

Because $q_6 \neq 0$ and $q_7 = q_8 = q_9 = q_{10} = 0$, then $l_a - 1 = 6$ and $l_b - 1 = 4$, where all q_{k_a} for which $l_a \leq k_a \leq l_a + l_b - 2$ are 0. Therefore, based on Corollary 3.12, the multiplication calculation is complete and $n = 11$. Using the binary positional system, another factor is

$$q = q_6q_5q_4q_3q_2q_1q_0 = 1011001_{(2)}$$
$$= 1 \cdot 2^6 + 0 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 89.$$

Similar to Example 3.13, the exponent of the Mersenne number with $p^2 = 23^2 = 529$ as a factor becomes $n = 253$.

B. Adapting the algorithm for Wieferich primes

Using inverse factorization, we can determine the exponent n of the minimum Mersenne number with a factor of p^2, p is odd. However, there are innumerable Mersenne numbers with a factor of p^2, including Wieferich primes. There are two algorithms to determine only the Wieferich primes.

In the first method, specifying a prime factor p, after determining the exponent n_{I} of the minimum Mersenne number with p^2 as a specified factor using inverse factorization, we determine the exponent n_{II} of the minimum Mersenne number with p^2 as a factor. If p is a Wieferich prime, then $n_{\mathrm{II}} = n_{\mathrm{I}}$. If it is not a Wieferich prime, the calculation can be ended when the calculation of n_{II} reaches n_{I}.

Proposition 3.15. *By the inverse factorization of Mersenne numbers, let the exponent n_I be from the specified prime factor p and let exponent n_{II} be from the factor p^2. If $n_I = n_{II}$, p is a Wieferich prime.*

Proof. *Let the exponent of the Mersenne number with a prime factor p be n_I and the other factor be q_I. Let the exponent of the Mersenne number with a factor of p^2 be n_{II} and the other factor be q_{II}. Then,*

$$M_{n_{\mathrm{I}}} = 2^{n_{\mathrm{I}}} - 1 = pq_{\mathrm{I}}, \quad (24)$$
$$M_{n_{\mathrm{II}}} = 2^{n_{\mathrm{II}}} - 1 = p^2q_{\mathrm{II}}. \quad (25)$$

Moreover, using proportional coefficients $h_{\mathrm{I}} \in \mathbb{N}$,

$$p = h_{\mathrm{I}}n_{\mathrm{I}} + 1 \quad (26)$$

if $n_{\mathrm{I}} = n_{\mathrm{II}}$, then $2^{n_{\mathrm{I}}} - 1 = pq_{\mathrm{I}} = p^2q_{\mathrm{II}}$. Thus, $2^{n_{\mathrm{I}}} \equiv 1 \pmod{p^2}$. Using Equation 26, $2^{(p-1)/h_{\mathrm{I}}} \equiv 2^{p-1} \equiv 1 \pmod{p^2}$. Therefore, p is a Wieferich prime. □

Next, we confirm that the inverse factorization of non-Wieferich primes can be applied in another means. In the past, research into non-Wieferich primes assumed the abc conjecture [22], [23]; however, we handle the scenario that holds regardless of the abc conjecture.

Proposition 3.16. *Consider a non-Wieferich prime p, $n_I \in \mathbb{N} - \{1\}$. If $2^{n_I} \equiv 1 \,(mod\ p)$ with the minimum exponent n_I, $2^{pn_I} \equiv 1 \,(mod\ p^2)$ with the minimum exponent pn_I.*

Proof. *Because p is a non-Wieferich prime, Equation 24 gives $q_I \not\equiv 0 \,(mod\ p)$. We can then obtain $(2^{n_I})^h = (pq_I + 1)^h$; herein, $h \in \mathbb{N} - \{1\}$. Using binomial coefficients, we have*

$(pq_{\mathrm{I}} + 1)^h = \binom{h}{0}(pq_{\mathrm{I}})^h + \binom{h}{h-1}(pq_{\mathrm{I}})^{h-1} + \binom{h}{h-2}(pq_{\mathrm{I}})^{h-2} + \cdots + \binom{h}{h-1}(pq_{\mathrm{I}})^1 + \binom{h}{h}(pq_{\mathrm{I}})^0$.
Thus,
$(2^{n_{\mathrm{I}}})^h - 1 = (pq_{\mathrm{I}} + 1)^h - 1 = \binom{h}{0}(pq_{\mathrm{I}})^h + \binom{h}{h-1}(pq_{\mathrm{I}})^{h-1} + \binom{h}{h-2}(pq_{\mathrm{I}})^{h-2} + \cdots + \binom{h}{h-1}(pq_{\mathrm{I}})^1$.
When more than two terms have exponents (pq_{I}), they have p^2 as a factor. Moreover, all binomial coefficients are natural numbers. Consequently, we have at least $\binom{h}{h-1} \equiv 0 \pmod{p}$ for $(2^{n_{\mathrm{I}}})^h - 1 = (pq_{\mathrm{I}} + 1)^h - 1 \equiv 0 \pmod{p^2}$. Therefore, $\binom{h}{h-1} = h \equiv 0 \pmod{p}$. When $h = p$, $2^{pn_{\mathrm{I}}} \equiv 1 \pmod{p^2}$ with the minimum exponent pn_{I}. □

It is self-evident that Proposition 3.16 can be applied to odd numbers except for Wieferich primes. This is a convenient algorithm for obtaining from the specified prime factor p except for Wieferich primes and through inverse factorization, the n_{II} exponent of the Mersenne number that has p^2 as a factor.

Another algorithm for determining only the Wieferich primes uses Corollary 3.17 after determining the exponent n_{II} of the minimum Mersenne number with the squared specified prime factor p, p^2, using inverse factorization.

Corollary 3.17. *If p is a non-Wieferich prime with minimum exponent n_I and $n_{II} = pn_I \neq p-1$, then $n_{II} - p > 0$. If p is a Wieferich prime, then $n_{II} - p < 0$.*

Proof. *Proposition 3.16 gives $n_{II} - p = p(n_I - 1) > 0$.* However, if p is a Wieferich prime, then $n_{\mathrm{II}} = n_{\mathrm{I}}$ so that $n_{\mathrm{II}} - p = n_{\mathrm{I}} - p < 0$ because $h_{\mathrm{I}} - 1 \geq 0$ and $p - n_{\mathrm{I}} = (h_{\mathrm{I}} - 1)n_{\mathrm{I}} + 1 > 0$ from Equation 26. □

Proposition 3.18. *If p is a non-Wieferich prime with minimum exponent n_I and $n_{II} = pn_I \neq p-1$,*
$1 + 2^{n_{\mathrm{I}}} + 2^{2n_{\mathrm{I}}} + \cdots + 2^{(p-1)n_{\mathrm{I}}} \equiv 0 \pmod{p}$ (27).

Proof. *Because $n_{II} = pn_I \neq p-1$ and $2^{pn_I} \equiv 1\ (mod\ p^2)$,*

$$M_{pn_{\mathrm{I}}} = 2^{pn_{\mathrm{I}}} - 1 = (2^{n_{\mathrm{I}}} - 1)\left(1 + 2^{n_{\mathrm{I}}} + 2^{2n_{\mathrm{I}}} + \cdots + 2^{(p-1)n_{\mathrm{I}}}\right) \equiv 0 \pmod{p^2}.$$

From Equation 24, $2^{n_{\mathrm{I}}} - 1$ has only one p as a factor such that it must be $1 + 2^{n_{\mathrm{I}}} + 2^{2n_{\mathrm{I}}} + \cdots + 2^{(p-1)n_{\mathrm{I}}} \equiv 0 \pmod{p}$. □

C. An algorithm for the square-freeness decision on Mersenne numbers with prime exponents

First, we use the method discussed in Section B to determine the Wieferich prime and the value of n of the associated Mersenne number. We use the trial division [24] to determine whether the exponent n is prime. If n is prime, a counterexample of the SFP is reported; hence, the problem is solved. Algorithm 3.19 is one possible trial division algorithm [25].

Algorithm 3.19: Trial division algorithm

INPUT: The exponent n of the Wieferich prime p obtained from the inverse factorization of Algorithm 3.11.

OUTPUT: Exponent n, either prime or composite.

1: Check if n is even.
$n_0 \leftarrow n/2 - \lfloor n/2 \rfloor$
If $n_0 = 0$ **then goto** 3
2: Check if n is divisible by an odd number less than or equal to $\lfloor \sqrt{n} \rfloor$.
For $d = 3$ **to** $\lfloor \sqrt{n} \rfloor$ **step** 2
$n_0 \leftarrow n/d - \lfloor n/d \rfloor$
If $n_0 = 0$, **then goto** 3
Next d
3: Output "prime" or "composite"
If $n_0 = 0$, **then**
Output "composite"
Else if $n_0 \neq 0$, **then**
Output "prime"
End if
4: **End**

TABLE I
DETECTION OF WIEFERICH PRIME 1093

p	n_{I}	p^2	n_{II}	$n_{\mathrm{II}} - p$
1061	1060	1125721	1124660	1123599
1063	531	1129969	564453	563390
1069	356	1142761	380564	379495
1087	543	1181569	590241	589154
1091	1090	1190281	1189190	1188099
1093	364	1194649	364	−729
1097	274	1203409	300578	299481
1103	29	1216609	31987	30884
1109	1108	1229881	1228772	1227663
1117	1116	1247689	1246572	1245455
1123	1122	1261129	1260006	1258883

TABLE II
DETECTION OF WIEFERICH PRIME 3511

p	n_{I}	p^2	n_{II}	$n_{\mathrm{II}} - p$
3463	577	11992369	1998151	1994688
3467	3466	12020089	12016622	12013155
3469	3468	12033961	12030492	12027023
3491	3490	12187081	12183590	12180099
3499	3498	12243001	12239502	12236003
3511	1755	12327121	1755	−1756
3517	3516	12369289	12365772	12362255
3527	1763	12439729	6218101	6214574
3529	882	12453841	3112578	3109049
3533	3532	12482089	12478556	12475023
3539	3538	12524521	12520982	12517443

TABLE III
PRIMALITY JUDGMENT FOR n_{II} OF WIEFERICH PRIMES

Wieferich prime p	n_{II}	Judgment for n_{II}
1093	$364 = 2^2 \cdot 7 \cdot 13$	composite
3511	$1755 = 3^3 \cdot 5 \cdot 13$	composite

D. Computer implementation

We implemented Algorithm 3.11 for the inverse factorization of Mersenne numbers in a computer running Windows 10 Home, version 1909, with 8.00 GB (7.39 GB available) of RAM and using an AMD E2-9000 RADEON R2, 4 COMPUTE CORE 2C+2G 1.80-GHz microprocessor. We then calculated the corresponding values using MS Excel from Microsoft Office Personal Premium and Algorithm 3.11 coded in Visual Basic for Applications (VBA).

VBA requires the *a priori* declaration of variables for the Mersenne number $M_n = pq$, and $(p-1)\lceil \log_2 p \rceil$ pcs as cells and $\log_2 p$ pcs as the number of digits of p in binary. Moreover, $p-1$ pcs each of σ_k, T_k, R_k, and V_k are declared in advance. Furthermore, $p-1$ pcs array variables $\mathrm{arr}(x)$ are declared for outputting q as a string.

Moreover, a prime factor p or its square was required in Algorithm 3.11 and was determined using Algorithm 3.19.

IV. RESULTS

Tables I and II present the detection results using inverse factorization for the two known Wieferich primes, 1093 and 3511, and ten samples before and after them. The inverse factorizations were run via Algorithm 3.11 based on Theorem 3.10. These tables show the Mersenne number exponent n_{I} with the prime factor p and the Mersenne number exponent n_{II} with the factor p^2. Moreover, $n_{\mathrm{II}} - p$ is given. These results show that the two known Wieferich primes can be detected based on both Proposition 3.15 and Corollary 3.17. Moreover, the example that follows shows an accurate sequential calculation.

Table III shows an investigation of SFP using the results of detected Wieferich primes. These exponents n_{II} are both composites, not counterexamples, and agree with the known facts.

Table IV shows the detection results with the inverse factorization using certain Wieferich numbers from the public table [16]. This shows that even if $n_{\mathrm{I}} = n_{\mathrm{II}}$, it is not necessarily Wieferich primes. We can see that Wieferich composite numbers take $n_{\mathrm{II}} - p < 0$, indicating that a primality test is necessary for the specified p to distinguish between a Wieferich prime and Wieferich composite. However, if $n_{\mathrm{I}} = n_{\mathrm{II}}$ or $n_{\mathrm{II}} - p < 0$ with a specified odd number p, it indicates either Wieferich prime or Wieferich composite; therefore, we can detect a Wieferich number.

Example 4.1. Given $\boldsymbol{p = 23, n_I = 11, q_I = 1011001_{(2)} = 89, p^2 = 529}$, ***we use Proposition 3.15 to determine whether p is a Wieferich prime number.***

$p^2 = 529 = 1000010001_{(2)}$,

$l_b - 1 = 9$,

$p_0 = 1, p_1 = 0, p_2 = 0, p_3 = 0, p_4 = 1, p_5 = 0, p_6 = 0, p_7 = 0, p_8 = 0, p_9 = 1$.

Assume

$k = 0, \sigma_0 = 1, p_0 q_0 = 1, T_0 = 1, R_0 = 0, V_0 = 1$.

When $k = 1$, if $q_1 = 0$, then $\sigma_1 = p_0 q_1 + p_1 q_0 = 0 + 0 = 0$, and

$T_1 = R_0 + \sigma_1 = 0 + 0 = 0$, which is even.

Thus, $q_1 = 1, p_0 q_1 = 1, p_1 q_1 = 0, p_2 q_1 = 0, p_3 q_1 = 0, p_4 q_1 = 1, p_5 q_1 = 0, p_6 q_1 = 0, p_7 q_1 = 0, \quad p_8 q_1 = 0, p_9 q_1 = 1, T_1 = 1$,

$R_1 = \lfloor T_1/2 \rfloor = \lfloor 1/2 \rfloor = 0$,

$\sigma_1 = 1, T_1 = 1, R_1 = 0, V_1 = 1$.

When $k = 2$, if $q_2 = 0$, then $\sigma_2 = p_0 q_2 + p_1 q_1 + p_2 q_0 = 0 + 0 + 0 = 0$, and

$T_2 = R_1 + \sigma_2 = 0 + 0 = 0$, which is even.

Thus, $q_2 = 1, p_0 q_2 = 1, p_1 q_2 = 0, p_2 q_2 = 0, p_3 q_2 = 0, p_4 q_2 = 1, p_5 q_2 = 0, p_6 q_2 = 0, p_7 q_2 = 0, p_8 q_2 = 0, p_9 q_2 = 1, T_2 = 1$,

$R_2 = \lfloor T_2/2 \rfloor = \lfloor 1/2 \rfloor = 0$,

$\sigma_2 = 1, T_2 = 1, R_2 = 0, V_2 = 1$.

When $k = 3$, if $q_3 = 0$, then $\sigma_3 = p_0 q_3 + p_1 q_2 + p_2 q_1 + p_3 q_0 = 0 + 0 + 0 + 0 = 0$, and

$T_3 = R_2 + \sigma_3 = 0 + 0 = 0$, which is even.

Thus, $q_3 = 1, p_0 q_3 = 1, p_1 q_3 = 0, p_2 q_3 = 0, p_3 q_3 = 0, p_4 q_3 = 1, p_5 q_3 = 0, p_6 q_3 = 0, p_7 q_3 = 0, p_8 q_3 = 0, p_9 q_3 = 1, T_3 = 1$,

$R_3 = \lfloor T_3/2 \rfloor = \lfloor 1/2 \rfloor = 0$,

$\sigma_3 = 1, T_3 = 1, R_3 = 0, V_3 = 1$.

When $k = 4$, if $q_4 = 0$, then $\sigma_4 = p_0 q_4 + p_1 q_3 + p_2 q_2 + p_3 q_1 + p_4 q_0 = 0 + 0 + 0 + 0 + 1 = 1$, and

$T_4 = R_3 + \sigma_4 = 0 + 1 = 1$, which is odd.

Thus, $q_4 = 0, p_0 q_4 = p_1 q_4 = p_2 q_4 = p_3 q_4 = p_4 q_4 = p_5 q_4 = p_6 q_4 = p_7 q_4 = 0, p_8 q_4 = p_9 q_4 = 0, T_4 = 1$,

$R_4 = \lfloor T_4/2 \rfloor = \lfloor 1/2 \rfloor = 0$,

$\sigma_4 = 1, T_4 = 1, R_4 = 0, V_4 = 1$.

When $k = 5$, if $q_5 = 0$, then $\sigma_5 = p_0 q_5 + p_1 q_4 + p_2 q_3 + p_3 q_2 + p_4 q_1 + p_5 q_0 = 0 + 0 + 0 + 0 + 1 + 0 = 1$, and

$T_5 = R_4 + \sigma_5 = 0 + 1 = 1$, which is odd.

Thus, $q_5 = 0, p_0 q_5 = p_1 q_5 = p_2 q_5 = p_3 q_5 = p_4 q_5 = p_5 q_5 = p_6 q_5 = p_7 q_5 = 0, p_8 q_5 = p_9 q_5 = 0, T_5 = 1$,

$R_5 = \lfloor T_5/2 \rfloor = \lfloor 1/2 \rfloor = 0$,

$\sigma_5 = 1, T_5 = 1, R_5 = 0, V_5 = 1$.

When $k = 6$, if $q_6 = 0$, then $\sigma_6 = p_0 q_6 + p_1 q_5 + p_2 q_4 + p_3 q_3 + p_4 q_2 + p_5 q_1 + p_6 q_0 = 0 + 0 + 0 + 0 + 1 + 0 + 0 = 1$, and

$T_6 = R_5 + \sigma_6 = 0 + 1 = 1$, which is odd.

Thus, $q_6 = 0, p_0 q_6 = p_1 q_6 = p_2 q_6 = p_3 q_6 = p_4 q_6 = p_5 q_6 = p_6 q_6 = p_7 q_6 = 0, p_8 q_6 = p_9 q_6 = 0, T_6 = 1$,

$R_6 = \lfloor T_6/2 \rfloor = \lfloor 1/2 \rfloor = 0$,

$\sigma_6 = 1, T_6 = 1, R_6 = 0, V_6 = 1$.

When $k = 7$, if $q_7 = 0$, then $\sigma_7 = p_0 q_7 + p_1 q_6 + p_2 q_5 + p_3 q_4 + p_4 q_3 + p_5 q_2 + p_6 q_1 + p_7 q_0 = 0 + 0 + 0 + 0 + 1 + 0 + 0 + 0 = 1$, and

$T_7 = R_6 + \sigma_7 = 0 + 1 = 1$, which is odd.

Thus, $q_7 = 0, p_0 q_7 = p_1 q_7 = p_2 q_7 = p_3 q_7 = p_4 q_7 = p_5 q_7 = p_6 q_7 = p_7 q_7 = 0, p_8 q_7 = p_9 q_7 = 0, T_7 = 1$,

$R_7 = \lfloor T_7/2 \rfloor = \lfloor 1/2 \rfloor = 0$,

$\sigma_7 = 1, T_7 = 1, R_7 = 0, V_7 = 1$.

When $k = 8$, if $q_8 = 0$, then $\sigma_8 = p_0 q_8 + p_1 q_7 + p_2 q_6 + p_3 q_5 + p_4 q_4 + p_5 q_3 + p_6 q_2 + p_7 q_1 + p_8 q_0 = 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 = 0$, and

$T_8 = R_7 + \sigma_8 = 0 + 0 = 0$, which is even.

Thus, $q_8 = 1, p_0 q_8 = 1, p_1 q_8 = 0, p_2 q_8 = 0, p_3 q_8 = 0, p_4 q_8 = 1, p_5 q_8 = 0, p_6 q_8 = 0, p_7 q_8 = 0, p_8 q_8 = 0, p_9 q_8 = 1, T_8 = 1$,

$R_8 = \lfloor T_8/2 \rfloor = \lfloor 1/2 \rfloor = 0$,

$\sigma_8 = 1, T_8 = 1, R_8 = 0, V_8 = 1$.

When $k = 9$, if $q_9 = 0$, then

$\sigma_9 = p_0 q_9 + p_1 q_8 + p_2 q_7 + p_3 q_6 + p_4 q_5 + p_5 q_4 + p_6 q_3 + p_7 q_2 + p_8 q_1 + p_9 q_0 = 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 = 1$, and

$T_9 = R_8 + \sigma_9 = 0 + 1 = 1$, which is odd.

Thus, $q_9 = 0, p_0 q_9 = p_1 q_9 = p_2 q_9 = p_3 q_9 = p_4 q_9 = p_5 q_9 = p_6 q_9 = p_7 q_9 = 0, p_8 q_9 = p_9 q_9 = 0, T_9 = 1$,

$R_9 = \lfloor T_9/2 \rfloor = \lfloor 1/2 \rfloor = 0$, and

$\sigma_9 = 1, T_9 = 1, R_9 = 0, V_9 = 1$.

When $k = 10$, if $q_{10} = 0$, then $\sigma_{10} = p_0 q_{10} + p_1 q_9 + p_2 q_8 + p_3 q_7 + p_4 q_6 + p_5 q_5 + p_6 q_4 + p_7 q_3 + p_8 q_2 + p_9 q_1 = 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 0 + 1 = 1$, and

$T_{10} = R_9 + \sigma_{10} = 0 + 1 = 1$, which is odd.

Thus, $q_{10} = 0, p_0 q_{10} = p_1 q_{10} = p_2 q_{10} = p_3 q_{10} = p_4 q_{10} = p_5 q_{10} = p_6 q_{10} = p_7 q_{10} = 0 \quad , \quad p_8 q_{10} = p_9 q_{10} = 0, T_{10} = 1$,

$R_{10} = \lfloor T_{10}/2 \rfloor = \lfloor 1/2 \rfloor = 0$,

$\sigma_{10} = 1, T_{10} = 1, R_{10} = 0, V_{10} = 1$.

Because $q_8 = 1$, all coefficients of $l_b - 1 = 9$ in Corollary 3.12 are nonzero. The tentative exponent is $n_{\mathrm{II}} = 11$; however, the inverse factorization calculation cannot be completed. Therefore, because $n_{\mathrm{I}} \neq n_{\mathrm{II}}$, p is not a Wieferich prime.

Example 4.2. When $p = 23, n_I = 11, q_I = 1011001_{(2)} = 89, and\ p^2 = 529$, we use Corollary 3.17 to determine whether p is a Wieferich prime. We skip the calculations, which are available in the software of Algorithm 3.11.

When inverse factorization is performed with the specified factor $p^2 = 529$, the exponent is $n_{\mathrm{II}} = 253$. Therefore, because $n_{\mathrm{II}} - p = 253 - 23 = 230 > 0$, p is a not a Wieferich prime.

Example 4.3. When $p = 1093 = 10001000101_{(2)}$, we use Proposition 3.15 to determine whether p is a Wieferich prime.

The pairs of quantities $n_{\mathrm{I}}, q_{\mathrm{I}}$ and $n_{\mathrm{II}}, q_{\mathrm{II}}$ are calculated by a computer that implements Algorithm 3.11 (Subsection 3.4).

For $n_{\mathrm{I}} = 364$, one has

$q_{\mathrm{I}} =$ 1110111111 0101101100 0111000101 0111001000
0100000110 0101001011 1010011011 1111110100
1100000111 1011101010 1010111110 1010011100
1110110100 0001110100 0010110000 0010000110
1110100011 0011111111 1100010000 0010100100
1110001110 1010001101 1110111110 0110101101
0001011001 0000000010 1100111110 0001000101
0101010000 0101011000 1100010010 1111100010
1111010011 1111011110 0100010111 0011 $_{(2)}$.

$n_{\mathrm{II}} = 364$ and

$q_{\mathrm{II}} =$ 1110000010 1100101011 1011111011 0000011111
0010100000 1000001000 1001001100 0101100000
1111100100 1001001110 1101111100 1000001101
1000001111 1001000010 0011010111 1001101010
0011111111 1111111111 1100011111 0100110101
0001000001 0011111000 0011010111 1101111101
1101101100 1110100111 1100000110 1101101100
0100100000 1101111100 1001111100 0001101111
0111001010 0001100101 0111 $_{(2)}$.

Because $n_{\mathrm{I}} = n_{\mathrm{II}}$, we conclude that $p = 1093$ is a Wieferich prime. This confirms the known result in terms of this number.

TABLE IV
DETECTION OF WIEFERICH PRIME AND COMPOSITE NUMBERS

Wieferich number p	n_{I}	n_{II}	$n_{\mathrm{II}} - p$
1093 (prime)	364	364	−729
3279	364	1092	−2187
3511 (prime)	1755	1755	−1756
7651	1092	1092	−6559
10533	3510	3510	−7023
14209	1092	1092	−13117
17555	7020	7020	−10535
22953	1092	1092	−21861
31599	3510	3510	−28089
42627	1092	1092	−41535
45643	7020	7020	−38623
52665	7020	7020	−45645
68859	1092	9828	−59031
94797	3510	31590	−63207
99463	1092	1092	−98371
127881	1092	9828	−118053
136929	7020	7020	−129909
157995	7020	7020	−150975
228215	7020	7020	−221195
298389	1092	1092	−297297
410787	7020	7020	−403767
473985	7020	63180	−410805
684645	7020	7020	−677625
895167	1092	9828	−885339
1232361	7020	63180	−1169181

Example 4.3 uses Proposition 3.15 to confirm whether p is a Wieferich prime by obtaining each exponent n_{I} and n_{II} of the minimum Mersenne number with the specified p and p^2. Example 4.4 uses Corollary 3.17 to confirm whether p is a Wieferich prime and not by determining the minimum Mersenne number for a specified p but by determining the exponent n_{II} of the minimum Mersenne number with p^2 as a factor.

Example 4.4. When $p = 1093$, we use Corollary 3.17 to determine whether p is a Wieferich prime.

When $p^2 = 1194649 = 10010001110101001 1001_{(2)}$, we have $n_{\mathrm{II}} = 364$. Therefore, because $n_{\mathrm{II}} - p = 364 - 1093 = -729 < 0$, we conclude that p is a Wieferich prime.

Moreover, $n_{\mathrm{II}} = 364$ is a composite number and not a counterexample of the SFP.

V. DISCUSSION

A. Evaluation and expansion of the algorithm

The proposed algorithm of inverse factorization of Mersenne numbers produces correct results for both SFP and WPP. When examining the SFP without going through the WPP, the exponent must pass the primality test; however, the result of the test is much less than the result of the primality test applied to the Mersenne numbers itself. Nevertheless, the SFP or WPP only requires to determine the power of the Mersenne number of the specified factor, and not the other factors. Therefore, the SFP and WPP can be investigated using existing algorithms for solving the discrete logarithm problem (DLP) [26], [27]. This can be accomplished, e.g., using classical algorithms, such as the baby-step giant-step (BSGS) algorithm [28], [29], developed by Shanks in 1969. In general, the DLP is $m^n \equiv a \pmod p$ and delivers n for a given m, a, and p.

Herein, we consider our algorithm and BSGS algorithm in terms of time complexity. Using a hash table, the BSGS algorithm was improved from $O(\sqrt{p}\log p)$ to $O(\sqrt{p})$. However, the time complexity of our algorithm is $O(p \log p)$, which is less efficient than the BSGS algorithm. Nevertheless, for the WPP, the BSGS algorithm converges at a rate of $O(\sqrt{p^2}) = O(p)$, which is an improvement over the previous convergence rate of $O(\sqrt{p^2}\log p^2) = O(p \log p)$. With Proposition 3.15, the inverse factorization for the WPP converges at a rate of $O(p \log p) + O(p \log p) = O(p \log p)$, whereas with Corollary 3.17, we achieve even $O(p^2 \log p^2) + O(1) = O(p^2 \log p)$. Therefore, the proposed algorithm based on Proposition 3.15 performs similar to the BSGS algorithm before improvement.

Because $\gcd(2, p) = 1$, we can consider $2^{n_{\mathrm{I}}} \equiv 2^{p-1} \equiv 1 \pmod p$ a reduced residue class group $(\mathbb{Z}/p\mathbb{Z})^{\times}$. The order of the group is $p - 1$. We can factorize $p - 1$ with the known (numerical) factorization algorithm. Let p_{j} be prime, for Equation 26, we will be able to gain $p - 1 = h_{\mathrm{I}} n_{\mathrm{I}} = 2^{h_1} \cdot 3^{h_2} \cdot 5^{h_3} \cdot 7^{h_4} \cdot \cdots \cdot p_{\mathrm{j}}^{h_{\mathrm{j}}} \equiv n_{\mathrm{I}}$, $h_{\mathrm{j}} \in \mathbb{N} \cup \{0\}$. Then, we can obtain $2^{n_{\mathrm{I}}} \equiv 1 \pmod p$ from a proper combination of these factors. First, the time complexity of factorizing the order of

the group is $O(\sqrt{p})$ for instance using trial division. Next, we consider the time complexity for search to obtain the exponent of the minimum Mersenne number. The maximum search amount is the case when all excluding h_1 are zero, $p = 2^{h_1} + 1$. Consequently, we can use binary search, $O(\log h_1) = O(\log\log p)$. To summarize, the worst time complexity is $O(\sqrt{p}) + O(\log\log p) = O(\sqrt{p})$. This is a similar result to the BSGS algorithm after improvement, $O(p)$ for the WPP. Moreover, to obtain the other factor, q requires to use division. Note that this procedure is different

TABLE V
TIME COMPLEXITY AND OUTPUT

	Time Complexity	Time Complexity for WPP	Output
Proposition 3.15	$O(p\log p)$	$O(p\log p)$	$n, q_{(2)}$
Corollary 3.17	$O(p\log p)$	$O(p^2\log p)$	$n, q_{(2)}$
BSGS after (hash table)	$O(\sqrt{p})$	$O(p)$	n
BSGS before (no hash table)	$O(\sqrt{p}\log p)$	$O(p\log p)$	n
Trial division for $p-1$	$O(\sqrt{p})$	$O(p)$	n

from our aim without (numerical) factorization. Table V summarizes the time complexity and output.

Next, we discuss algorithms to improve the efficiency of the inverse factorization algorithm. For $2^n - 1 \equiv 0 \pmod p$, assume a composite number such as $n = hn_3$ is on the same base $m = 2$, $h, n_3 \in \mathbb{N} - \{1\}$, with $2^n = (2^h)^{n_3} \equiv 1 \pmod p$. When inverse factorization is applied in the cell space with $m = 2^h$ as the base, the minimum exponent is n_3, which is expected to be h times more efficient. Thus, if $m = 2^h$, as per its positional system, the Mersenne numbers are $(2^h - 1)(2^h - 1)\cdots(2^h - 1)$. In general, in base m, each digit of q is obtained from $\{0, 1, \dots, m-1\}$, and the condition that satisfies all $V_k = m - 1$ is uniquely determined. When $p = 23, m = 2$ is not a primitive root [30], [31] of p. For example, the results of inverse factorization with $p = 23$ for bases $m = 2,3,4,5,6,7,8,9$ are as follows:

$2^{11} - 1 = 10111_{(2)} \times 1011001_{(2)}$,
$3^{11} - 1 = 212_{(3)} \times 101120021_{(3)}$,
$4^{11} - 1 = 113_{(4)} \times 230201121_{(4)}$,
$5^{22} - 1 = 43_{(5)} \times 1020413321434240311 23_{(5)}$,
$6^{11} - 1 = 35_{(6)} \times 1322030441_{(6)}$,
$7^{22} - 1 = 32_{(7)} \times 2062511343646041553 23_{(7)}$,
$8^{11} - 1 = 27_{(8)} \times 2620544131_{(8)}$,
$9^{11} - 1 = 25_{(9)} \times 3462311507_{(9)}$,
where $p = 23_{(10)} = 10111_{(2)} = 212_{(3)} = 113_{(4)} = 43_{(5)} = 35_{(6)} = 32_{(7)} = 27_{(8)} = 25_{(9)}$,

$2^{11} \equiv 3^{11} \equiv 4^{11} \equiv 5^{22} \equiv 6^{11} \equiv 7^{22} \equiv 8^{11} \equiv 9^{11} \equiv 1 \pmod{23}$. However, when $p = 13$, $m = 2$ is a primitive root of p. For example, the results of the inverse factorization with $p = 13$ for base $m = 2,3,4,5,6,7,8,9$ are as follows:

$2^{12} - 1 = 1101_{(2)} \times 100111011_{(2)}$,
$3^3 - 1 = 111_{(3)} \times 2_{(3)}$,
$4^6 - 1 = 31_{(4)} \times 10323_{(4)}$,
$5^4 - 1 = 23_{(5)} \times 143_{(5)}$,
$6^{12} - 1 = 21_{(6)} \times 24340531215_{(6)}$,
$7^{12} - 1 = 16_{(7)} \times 35245631421_{(7)}$,
$8^4 - 1 = 15_{(8)} \times 473_{(8)}$,
$9^3 - 1 = 14_{(9)} \times 62_{(9)}$,
where $p = 13_{(10)} = 1101_{(2)} = 111_{(3)} = 31_{(4)} = 23_{(5)} = 21_{(6)} = 16_{(7)} = 15_{(8)} = 14_{(9)}$, $2^{12} \equiv 3^3 \equiv 4^6 \equiv 5^4 \equiv 6^{12} \equiv 7^{12} \equiv 8^4 \equiv 9^3 \equiv 1 \pmod{13}$. These results indicate that multiple relationships exist between exponents obtained by inverse factorization for different bases m. For example, if $m = 2$ is a primitive root of the specified factor p, $p - 1$ would be implied by Fermat's little theorem [32]. If we do not know whether a number is a primitive root and if we can determine the proportionality coefficient of the exponent of $m = 2$, the efficiency should improve because reducing the number of cells reduces the number of sums required to calculate σ_k. The base with $1 < h \in \mathbb{N}$ and $m = 2^h$ performs the inverse factorization of the specified p to obtain its exponent n_3. Excluding the instance when $m = 2$ is not a primitive root and $(p-1)/n_3 = h$, if $(p-1)/n_3 = h$, $2^n - 1 = 2^{hn_3} - 1$. If $(p-1)/n_3 \neq h$, we conjecture $2^n - 1 = 2^{n_3} - 1$. For example, in the case of $p = 23, m = 4$, it fails. However, the larger the p and the more bases $m = 2^h$ we can select, the lower the probability of failure. Moreover, the generalization of the inverse factorization to any base, $m \in \mathbb{N}, m \neq 1$, is applied to repunits $R_n := (m^n - 1)/(m - 1) = pq$. Moreover, we can use repunits as repdigits $gR_n = p(gq), m - 1 \geq g \in \mathbb{N}$, where the Mersenne numbers are $m = 2$ and $g = 1$. The inverse factorization of repdigits expanded based on Algorithm 3.11 is shown to Algorithm 5.1.

Algorithm 5.1: Inverse factorization of repdigits

INPUT: Specify Base m and an odd number p that is a factor of the given repdigit, the coefficient g.

OUTPUT: The decimal exponent n of the repdigit with the decimal factor p, and another factor q in base m.

1: Specify the divisor $p_{(10)}$ in decimal, and m, g.
2: Express p in a m-adic expansion; i.e., a series expansion with term number l_b:
$p_{(m)} = \sum_{y=0}^{l_b-1} p_y \cdot m^y = p_{l_b-1}\, p_{l_b-2} \cdots\ p_{0\,(m)}$.
3: Assign each digit of p in the base m positional system to cell $p_y q_0 \circ (0, y)$.
For $y = 0$ **to** $l_b - 1$; $l_b - 1 = \lfloor \log_m p_{(10)} \rfloor$
$p_y \leftarrow p_y$; initial condition
Next y
4: Let $R_{-1} = 0$, $\sigma_0 = 0$, and $T_0 = 0$. Let $V_c =$ null.
5: **For** $k = 0$ **to** $p - 1$
6: Determine q_k under the condition $V_k = g$;
Note that any coefficient of a digit with k greater than k_a in the calculation is set to zero.
$T_k \leftarrow R_{k-1} + \sigma_k$
For $q_k = 0$ **to** $m - 1$
If $T_k + p_0 q_k \equiv g\ (mod\ m)$ **then**, **goto** 7: Determine q_k.
Next q_k
7: Calculate $\sigma_k, T_k,$ and R_k. Let $V_c = 1$.
For $y = 0$ **to** $l_b - 1$,
Calculate $p_y q_k$.
$\sigma_{k+y} \leftarrow \sigma_{k+y} + p_y q_k$,
$T_{k+y} \leftarrow R_{k+y-1} + \sigma_{k+y}$,

$R_{k+y} \leftarrow \lfloor T_{k+y}/m \rfloor$,
$V_{k+y} \leftarrow T_{k+y} - m\,R_{k+y}$,

8: This step is the preparation to examine the completion of the calculation using Corollary 3.8.
If $V_{k+y} = g$ **then**
$V_c \leftarrow V_c * V_{k+y}/g$.
Else if $V_{k+y} \neq g$ **then**
$V_c \leftarrow 0$
End if
Next y

9: Get a string of divisors $q_{(m)}$ and let $\mathrm{arr}(k)$ be a string variable.
If $Len(q_k) = 1$, **then**
$\mathrm{arr}(k) \leftarrow q_k \& \mathrm{arr}(k)$, where "&" is the string concatenation operator.
Else if $Len(q_k) > 1$, **then**: This function gets the length of the string.
$\mathrm{arr}(k) \leftarrow$ "("$\&q_k\&$ ")"$\&\ \mathrm{arr}(k)$: Notation by the positional system in $m > 10$.
End if

10: Use Corollary 3.8 to examine the completion of the calculation.
If $V_c = 1$, and $R_{k+l_b-1} = 0$, **then goto** 12

11: **Next** k

12: Output $n = k + l_b$

13: Output $\mathrm{arr}(k)$ as $q_{(m)}$, which is a positional system in base m representation in order of decreasing exponent. Or convert $q_{(m)}$ to decimal by using the binary expansion and output $q_{(10)}$.

14: **End**

Note that another factor q in the inverse factorization generally differs for different bases. Let $h, h' \in \mathbb{N}$ and $h \neq h'$. When the base $m = 2$ is a primitive root of the specified factor p, for any two bases 2^h and $2^{h'}$, the other factors obtained by the inverse factorization with a specified factor p are equal, i.e., $q_{(2^h)} = q_{(2^{h'})}$. Nevertheless, the relationship between $q_{(2^h)}$ and $q_{(2^{h'})}$ is not equal in other cases. This is an improvement considering that the DLP does not require the identification of another factor q. Moreover, the ability to determine another factor that the DLP does not require indicates that there is a possibility of the application to the algorithm for factoring Mersenne numbers. However, another factor obtained using inverse factorization is binary, which we must be converted to decimal if necessary, as performed in Example 3.13. Depending on the size of q, this process increases computation time and creates overflow issues; therefore, it should be improved. Furthermore, depending on the computing environment, the BSGS algorithm can cause exponent calculation overflow, but not in the inverse factorization.

The basic theory of inverse factorization is simpler than the BSGS algorithm, which requires advanced mathematical knowledge such as finite fields [28], [29] and group theory [28], [29]. Precisely, lattice multiplication [33] can be considered as an alternative to cell algebra. However, the proposed approach should reduce the barriers to entry and allow beginners to research the SFP and WPP.

B. Application to encryption

Recently, information security has become increasingly relevant; research has been conducted on the secure storage of passwords [34], the complex Vernam cipher [35], and the efficient key exchange protocol [36]. Note that inverse factorization of Mersenne numbers constitutes an encryption algorithm. In conventional block encryption, the plaintext is divided into multiple blocks with the same bit length and converted into ciphertext of the same bit length [37]–[40]. Moreover, encryption may be used multiple times to strengthen security. Furthermore, security may be strengthened by converting plaintext blocked to the same bit length into ciphertext of a different bit length, even if the text is encrypted only once.

We now study whether this approach works for the inverse factorization of Mersenne numbers, which encrypts odd p of plaintext to the other factor q. Moreover, the bit length of each block of the ciphertext is not the same, and so the bit length is increased to exceed that of the plaintext (Table V). When ciphertexts divided into blocks are concatenated, the bit length of each block is unknown, thereby making decryption difficult. Thus, the sequence of bit lengths of the ciphertexts is the decryption key. Therefore, we strengthen security against ciphertext-only attacks [41] because the attacker requires to divide each concatenated ciphertext to determine each bit length to decrypt.

Note that this composite key has limited reusability and each plaintext block length is not a key. Information regarding the block lengths that divide the plaintext is unnecessary for decryption, and the block lengths can be changed. The decryption key sequence for the decryption is generated. This approach strengthens security against known-plaintext attacks [42].

Moreover, we must ensure encryption. For example, we preprocess by adding w_p =1 to the prefix and w_s =$1101_{(2)}$ to the suffix of each divided plaintext. The former is a measure against digit loss, so 0100 is recognized as 100, and the latter is a measure against even numbers and the types of Mersenne and Fermat numbers, such as $1010101_{(2)}$, $1001001_{(2)}$, and $110011_{(2)}$. These cannot be decoded. When p = 0100, the preprocessing to enable inverse factorization is

$w_p \& p \& w_s = 1\&0100\&1101 = 101001101_{(2)}$.

(w_p, w_s) is the common key.

Conventionally, Σ is the alphabet, the encryption function is $\Sigma^l \to \Sigma^l$, $l \in \mathbb{N}$ [31], but the inverse factorization of Mersenne numbers has the property $\Sigma^{l_b} \to \Sigma^{l_a}$, $l_a, l_b \in \mathbb{N}$. Cases exist where $l_a \leq l_b$, but cases where $l_a > l_b$ are more prevalent in Table VI.

Moreover, the post processing replaces one character of the first and last of the ciphertext with its equivalent of the plaintext. For instance, the inverse factorization of $101001101_{(2)}$ is $11000100110011100000001111011_{(2)}$, which is replaced by $01000100110011100000001111010_{(2)}$ because the prefix and suffix of the plaintext p are both 0. We simply write this postprocessing as $w_c = [\text{first character } w_f, \text{last character } w_l]$, this case is $w_c =$

[0,0].

Figure 2 shows the flowchart for encryption using inverse factorization, which is detailed in Algorithm 5.3. Moreover, Example 5.2 is a simple example of enciphering.

Example 5.2. ***The plaintext*** $\boldsymbol{p_{(2)}} =$ **0100111011110010** ***is divided into 3 bits to show the ciphertext*** $\boldsymbol{q}$ ***by inverse factorization of Mersenne numbers,*** $\boldsymbol{f^{\sim 1}}$***. Let prefix*** $\boldsymbol{w_p}$ ***=1 and suffix*** $\boldsymbol{w_s}$ ***=1101.***

Because $p_{(2)} = 010\&011\&101\&111\&001\&0$, dividing each element block p_{j} for $\mathrm{j} \in \mathbb{N}$ gives

$$p_1 = 1\&010\&1101 = 10101101 = 173_{(10)},$$
$$p_2 = 1\&011\&1101 = 10111101 = 189_{(10)},$$
$$p_3 = 1\&101\&1101 = 11011101 = 221_{(10)},$$
$$p_4 = 1\&111\&1101 = 11111101 = 253_{(10)},$$
$$p_5 = 1\&001\&1101 = 10011101 = 157_{(10)},$$
$$p_6 = 1\&0\&1101 = 101101 = 45_{(10)}.$$

Moreover, the postprocessings are

$w_{c1} = [0,0], w_{c2} = [0,1], w_{c3} = [1,1], w_{c4} = [1,1], w_{c5} = [0,1], w_{c6} = [0,0]$.

To encrypt, make p_{j} and $l_{b\mathrm{j}}$ correspond to p and l_b. As per Line 3 of Algorithm 3.11 for the inverse factorization of Mersenne numbers, we have q_{j}, n_{j}, and $l_{a\mathrm{j}}$. Thus,

$q_1 =$ 1011110101101001000100000010001110000011101 1001100001101010100010110001100100100111111110100 0010100101101110111111011100011111000100110011110 010101011101001110011011011, $n_1 = 172, l_{a1} = 165$,

q_2 =10101101011, $n_2 = 18, l_{a2} = 11$,

q_3 =10010100010001011, $n_3 = 24, l_{a3} = 17$,

$q_4 =$ 1000000110000100100011011010100011111101011 1100001101001001110111011001100011001010010111111 0001110101011, $n_4 = 110, l_{a4} = 103$,

$q_5 =$ 1101000010110110100111111100101111101001001 011, $n_5 = 52, l_{a5} = 45$,

q_6 =1011011, $n_6 = 12, l_{a6} = 7$.

Therefore, the ciphertext is

$w_c(q_{(2)})$
$= w_{c1}(q_1)\&w_{c2}(q_2)\&w_{c3}(q_3)\&w_{c4}(q_4)\&w_{c5}(q_5)\&w_{c6}(q_6)$
$=$

0011110101101001000100000010001110000011101100110000110101010001011000110010010011111110100001010010110111011111101110001111100010011001111001010101110100111001101101000101101011100101000100010111000000110000100100011011010100011111101011110000110100100111011101100110001100101001011111000111010101101010000101101101001111111001011110100 10010110011010,

and the decryption key sequence is $l_{a1}, l_{a2}, l_{a3}, l_{a4}, l_{a5}, l_{a6} = 165, 11, 17, 103, 45, 7$. Thus, this example of 16 bits of plaintext produces an expanded ciphertext of 348 bits.

TABLE VI
BIT LENGTH OF CIPHERTEXT CORRESPONDING TO PLAINTEXT OF BIT LENGTH 6

$p_{(2)}$	$q_{(2)}$	l_a
100001	11111	5
100011	1110101	7
100101	1101110101100111110010001010011	31
100111	1101001	7
101001	110001111100111	15
101011	101111101	9
101101	1011011	7
101111	101011100100110001	18
110001	1010011100101111	16
110011	101	3
110101	10011010100100001110011111011001010110111100011	47
110111	100101001111001	15
111001	1000111110111	13
111011	10001010110110001111001011111011101010010011100001101	53
111101	1000011001001011100010100111110111100110110100011101011	55
111111	1	1

Algorithm 5.3: Encryption using the inverse factorization of Mersenne numbers

INPUT: A plaintext as a number $p_{(2)}$, its divided bits l, and common keys (w_p, w_s).

OUTPUT: The ciphertext $q_{(2)}$ and the decryption key sequence $K_d = l_{a1}, \ldots, l_{a\omega}$, where ω is the number of encryption blocks.

1: Let $\mathrm{j} \in \mathbb{N} \cup \{0\}$ be a count number, set initial value $\mathrm{j} = 0$.

2: **Do**

3: Plaintext $p_{(2)}$ is divided into blocks of l bits from the beginning and assigned to array variables $p(\mathrm{j})$ in order.
$p_{(2)} = p_1\&p_2\&\cdots\&p_{\mathrm{j}}\&\cdots\&p_\omega$
$l_s = \mathrm{j} \cdot l + 1$
$p(\mathrm{j}) \longleftarrow \mathrm{Mid}(p_{(2)}, l_s, l)$: A function that yields l pieces of character as a base point the l_sth character from the left end of the string $p_{(2)}$.

4: As the preprocessing, concatenate the prefix w_p and the suffix w_s into each plaintext $p(\mathrm{j})$ block and assign it to $p(\mathrm{j})$.
$p(\mathrm{j}) \longleftarrow w_p\&p(\mathrm{j})\&w_s$

5: The bit length l_b of $p(\mathrm{j})$,
$l_{b\mathrm{j}} = \lfloor \log_{10} p(\mathrm{j}) \rfloor + 1$.

6: Read the first and last characters of each plaintext for post processing: $w_{c\mathrm{j}} := [w_f(\mathrm{j}), w_l(\mathrm{j})]$.
$w_f(\mathrm{j}) \longleftarrow \mathrm{Left}(p_{\mathrm{j}}, 1)$: Get one character from the left side of the p_{j} string.
$w_l(\mathrm{j}) \longleftarrow \mathrm{Right}(p_{\mathrm{j}}, 1)$: Get one character from the right side of the p_{j} string.

7: To encrypt, make $p(\mathrm{j})$ and $l_{b\mathrm{j}}$ correspond to p and l_b, to be taken from Line 3 of the Algorithm 3.11 for the inverse factorization of Mersenne numbers.
$(n_{\mathrm{j}}, q_{\mathrm{j}}) = f^{\sim 1}(p_{\mathrm{j}})$.

8: Get element $l_{a\mathrm{j}}$ of the encryption key sequence,
$l_{a\mathrm{j}} = n_{\mathrm{j}} - l_{b\mathrm{j}} + 1$

9: $\mathrm{j} \longleftarrow \mathrm{j} + 1$

10: **Loop until** $p(\mathrm{j}) = \mathrm{null}$: Iterates from Line 2 to Line 9 and exits the loop when $p(\mathrm{j})$ is null.

11: $\omega \longleftarrow \mathrm{j} - 1$

12: Using the obtained $(l_{a\mathrm{j}}, q_{\mathrm{j}})$, compound the ciphertext $q_{(2)}$ and the decryption key sequence $K_d = l_{a1}, l_{a2}, \cdots, l_{a\omega}$.
Let $K_d = \mathrm{null}$ initially.
For $\mathrm{j} = 1$ **to** ω
$q_{\mathrm{j}} \leftarrow w_f(\mathrm{j})\ \&\ \mathrm{Mid}(q_{\mathrm{j}}, 2, l_{a\mathrm{j}} - 1)\ \&\ w_l(\mathrm{j}) :\ w_{c\mathrm{j}}(q_{\mathrm{j}})$
$q_{(2)} \longleftarrow q_{(2)}\&q_{\mathrm{j}}$
$K_d \longleftarrow K_d\&","\&l_{a\mathrm{j}}$

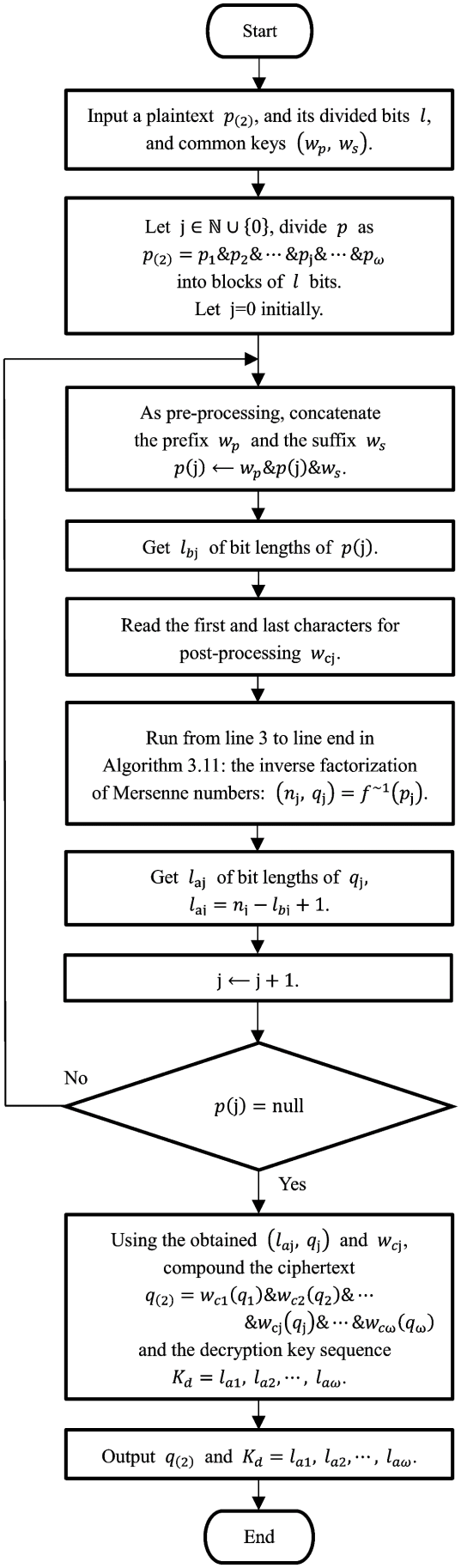

Fig. 2. Flowchart for encryption using the inverse factorization of Mersenne numbers.

Next j
13: Output $q_{(2)}$
14: Output $K_d = l_{a1}, l_{a2}, \cdots, l_{a\omega}$.
15: **End**

The decryption is executed using the reverse process. First, using the decryption key, we can divide each element block of the ciphertext corresponding to the bit length. Next, read the first and last characters of each ciphertext; if any of these characters are 0, and replace them with 1. Then, executing as per Line 3 of Algorithm 3.11 for the inverse factorization of Mersenne numbers for each element block, we can obtain the plaintexts divided into blocks. Remove the prefix w_p and suffix w_s from the plaintext of each block. Finally, concatenating the block plaintexts in order, we can obtain the original plaintext.

Considering the computational complexity of inverse factorization and that blocks of ciphertext are not sorted by bit length, we assume that decoding is attempted in the ascending order of bit length. Let the ciphertext be l_a bits long. From the beginning of the block, the inverse factorization is performed bit by bit until the decipher succeeds, and then the inverse factorization is repeated. Next, we perform the same from the beginning of the next block. Let l_{a0} be a unit bit and $m^{l_{a0}-1}$ be the ciphertext corresponding to the unit bit (herein, it is $m = 2$). If the ω_j[th] search can decrypt the j[th] block search, with $z \le \omega_j \in \mathbb{N}$ being a parameter, the time complexity is

$$O(2^{l_{a0}-1}\log 2^{l_{a0}-1}) + O(2^{2l_{a0}-1}\log 2^{2l_{a0}-1}) + O(2^{3l_{a0}-1}\log 2^{3l_{a0}-1}) + \cdots + O\left(2^{\omega_j l_{a0}-1}\log 2^{\omega_j l_{a0}-1}\right) = \sum_{z=1}^{\omega_j} O\left(2^{zl_{a0}-1}\cdot(zl_{a0}-1)\right) = \max_{z\le\omega_j\in\mathbb{N}} O(zl_{a0}\cdot 2^{zl_{a0}}).$$

Therefore, set $l_{a0} = 1$. If the i[th] search in this search of the j[th] block is the maximum computational load, the time complexity is $\max_{z\le\omega_j\in\mathbb{N}} O(z\cdot 2^z) = O(i\cdot 2^i)$. Then, the decryption load is dominated by the block with the maximum exponent n of the discrete logarithm. Generally, there is a risk that the common key for block cipher with a fixed bit length l will be decrypted by an exhaustive search of 2^l [37]. However, a cipher using the inverse factorization can have $z \le l$ such that $z\cdot 2^z = 2^{z+\log z} \ge 2^l$, so $z + \log z \ge l$. If $z = i \ne \omega_j$, the proposed cipher can decrypt without its dominant step because the receiver has the decryption key sequence. The bit length $z = \omega_j$ of the ciphertext can be generated from the plaintext with a smaller bit length.

If the BSGS algorithm after improvement or factoring is used, the time complexity is $\sum_{z=1}^{\omega_j} O\left(2^{z/2}\right) = \max_{z\le\omega_j\in\mathbb{N}} O\left(2^{z/2}\right)$ by similar consideration.

Let the fixed bit length of the key of the conventional block cipher be l_T, and the mean bit length of the ciphertext by the inverse factorization be l_I. At least $l_I \ge 2l_T$ to become more secure than conventional block ciphers under an exhaustive search is necessary for the inverse factorization cipher under exhaustive factorization attack.

Encryption using the inverse factorization of Mersenne numbers is expected to be useful for encrypting short sentences because the bit length of plain text is extended.

However, for a long sentence, an appropriate way to determine the bit lengths of the plaintexts and the prefix and suffix of the preprocessing such that the encryption processing time is within the practical range is a subject for future research.

VI. Conclusion

Because no theoretical solution is yet available for the SFP and WPP, one must depend on a computational solution. If the proposed inverse factorization obtains other Wieferich primes, then the WPP is a step closer to being solved. Moreover, if the exponent of the minimum Mersenne number with any Wieferich prime factor is prime, the SFP is solved as a counterexample.

Thus, inverse factorization of Mersenne numbers is a hybrid approach of factorization and expansion. Although it seems to be less efficient than the classical DLP algorithm (the BSGS algorithm), the application of the proposed algorithm to the WPP produces results similar to those of the BSGS algorithm before the latter's improvement in terms of time complexity. Moreover, this inverse factorization algorithm was generalized to repdigits, including Mersenne numbers. Improving the efficiency of the proposed algorithm is a potential research topic. Moreover, the question of whether the factorization algorithm may be improved by applying the inverse factorization computation in reverse remains open.

Note that the practical applicability of the inverse factorization of Mersenne numbers as an encryption algorithm. As opposed to block ciphers, the proposed algorithm allows the bit length to be expanded. The bit lengths of plaintext require not be constant and do not serve as encryption keys. Thus, the sender and receiver require to not agree regarding the bit lengths of plaintext used as the encryption keys. By changing the plaintext block length, we can prevent the same plaintext from being encrypted into the same ciphertext. Another feature is that the decryption key cannot be used for other ciphertexts because it is prevented by nonlinearity between plaintext and ciphertext via the inverse factorization of Mersenne numbers. The block length of the ciphertext is the decryption key, and block lengths less than the decryption key have a block that maximizes the computation load, because of which the security is improved. However, to equate the security of conventional block ciphers under an exhaustive search and our suggested cipher under an exhaustive factorization attack, the mean bit length of the ciphertext should be at least more than twice that of conventional block ciphers. These features should allow the cipher systems to be developed based on the inverse factorization of Mersenne numbers. However, to maintain the encryption processing time within a practical range, an algorithm of balancing the bit length of the plaintext that has been divided and preprocessed with the total number of blocks, which will be a subject for future research. It will be relevant to compare security with conventional cryptography and investigate techniques to strengthen encryption against other attacks. More secure encryption will be expected with the inverse factorization of repdigits.

Finally, the cell space introduced in this study is an operation based on multiplication. The computation rules of the inverse factorization of Mersenne numbers are simple to understand, and it should lower the barrier to entry for beginners by allowing them to approach the topic without prior knowledge of finite fields or group theory. The concrete description of addition and investigations into the algebraic structure of the cell space are both subjects for future work.

Data Availability

The Algorithm data used to support the results of this research are included within the article. Furthermore, the software data based on Algorithm 3.11 have been deposited in the figshare repository (https://doi.org/10.6084/m9.figshare.14495235.v4).

Acknowledgment

The authors are grateful to Masahiro Kumabe of the Open University of Japan for commenting on this work and for advice pertaining to this research. The authors would like to thank Enago (www.enago.jp) for the English language review.

References

[1] P. Pollack and V. Shevelev, "On perfect and near-perfect numbers," *J. Number Theor.*, vol. 132, no. 12, pp3037–3046, 2012.

[2] T. Kleinjung, J. W. Bos and A. K. Lenstra, "Mersenne factorization factory" in P. Sarkar and T. Iwata, Eds. Berlin: Springer, Dec. 2014 (eds.) *International Conference on the Theory and Application of Cryptology and Information Security*, pp358–377.

[3] L. Debnath and K. Basu, "Some analytical and computational aspects of prime numbers, prime number theorems and distribution of primes with applications," *Int. J. Appl. Comput. Math.*, vol. 1, no. 1, pp3–32, 2015.

[4] K. R. Guy, "Prime numbers" in *Unsolved Problems in Number Theory*. New York: Springer, Jul. 2004, pp3–69.

[5] J. Knauer and J. Richstein, "The continuing search for Wieferich primes," *Math. Comput.*, vol. 74, no. 251, pp1559–1564, 2005.

[6] R. Crandall, K. Dilcher and C. Pomerance, "A search for Wieferich and Wilson primes," *Math. Comput.*, vol. 66, no. 217, pp433–450, 1997.

[7] T. Dupuy and Weirich, "Bits of in binary, Wieferich primes and a conjecture of Erdős," *J. Number Theor.*, vol. 158, pp268–280, 2016.

[8] W. Keller and J. Richstein, "Solutions of the congruence $a^{p-1} \equiv 1 \pmod{p^r}$," *Math. Comput.*, vol. 74, no. 250, pp927–936, 2004.

[9] E. L. Roettger, H. C. Williams and R. K. Guy, "Some primality tests that eluded Lucas," *Des. Codes Cryptogr.*, vol. 77, no. 2–3, pp515–539, 2015.

[10] S. Y. Yan and G. James, "Testing Mersenne primes with elliptic curves" in *Computer Algebra in Scientific Computing*, V. G. Ganzha, E. W. Mayr and E. V. Vorozhtsov, Eds. Berlin, Heidelberg: Springerpp, Sep. 2006, pp303–312.

[11] X. Wang, "Factorization of large numbers via factorization of small numbers," *Glob. J. Pure Appl. Math.*, vol. 6, pp5157–5173, 2016.

[12] L. J. Warren and H. G. Bray, "On the square-freeness of Fermat and Mersenne numbers," *Pac. J. Math.*, vol. 22, no. 3, pp563–564, 1967.

[13] S. Davis, "Arithmetical sequences for the exponents of composite Mersenne numbers," *Notes Number Theor. Discrete Math.*, vol. 20, pp19–26, 2014.

[14] N. N. Dong Quan and D. Quan, "Carlitz module analogues of Mersenne primes, Wieferich primes, and certain prime elements in cyclotomic function fields," *J. Number Theor.*, vol. 145, pp181–193, 2014.

[15] K. Broughan, S. G. Sanchez, and F. Luca, "Perfect repdigits," *Math. Comput.*, vol. 82, no. 284, pp2439-2459, 2013.

[16] N. J. A. Sloane, "Wieferich numbers (1): n > 1 such that 2^A000010(n) == 1 (mod n^2).," The on-line encyclopedia of integer sequences. Available: https://oeis.org/A077816/b077816.txt

[17] T. Agoh, K. Dilcher and L. Skula, "Fermat Quotients for Composite Moduli," *J. Number Theor.*, vol. 66, no. 1, pp29-50, 1997.

[18] C. Esholtz, "A survey on additive and multiplicative decompositions of sumsets and of shifted sets" in *I Combinatorial Number Theory and Additive Group Theory*, 2009, Adv. Courses Math. C. R. M. Barcelona, *(Centre de Recerca Matemàtica)*. Base: Birkhäuser, pp213–231.

[19] A. Bonfietti and M. Lombardi, "The weighted average constraint" in. *Intl. Conf. on Princ. and Pract. of Constraint Program*. Berlin, Heidelberg, M. Milano, Ed. Springer, 2012, pp191–206.

[20] H. Riesel, "The recognition of primes" in *Prime Numbers and Computer Methods for Factorization.*Birkhuser, J. Coates and S. Helgason, Eds. Boston: Modern Birkhuser Classics, 2011, pp84–140.
[21] A. Slinko, "Integers" in *Springer Undergrad. Math. S.*, Jun. 2015, pp1–36.
[22] H. Graves and M. R. Ram Murty, "The abc conjecture and non-Wieferich primes in arithmetic progressions," *J. Number Theor.*, vol. 133, no. 6, pp1809–1813, 2013.
[23] Y. G. Chen and Y. Ding, "Non-Wieferich primes in arithmetic progressions," *Proc. Am. Math. Soc.*, vol. 145, no. 5, pp1833–1836, 2017.
[24] H. Riesel, "The number of primes below a given limit" in *Prime Numbers and Computer Methods for Factorization*, J. Oesterlé and A. Weinstein, Eds., 2011. B. Birkhuser, *Modern Birkhuser Classics*, pp1–36.
[25] H. Riesel, "Classical method of factorization" in *Prime Numbers and Computer Methods for Factorization*, J. Oesterlé and A. Weinstein, Eds., 2011. B. Birkhuser, *Modern Birkhuser Classics*, pp141–172.
[26] B. Kacsmar et al., "Computing low-weight discrete logarithms" in *Sel. Areas Cryptogr.*, C. Adams and J. Camenisch, Eds. Springer, pp106–126, 2017.
[27] J. H. Cheon and T. Kim, "A new approach to the discrete logarithm problem with auxiliary Inputs," *LMS J. Comput. Math.*, vol. 19, no. 1, pp1–15, 2016.
[28] P. A. Kameswari et al., "Shank's baby-step giant-step attack extended to discrete log with Lucas sequences," *IOSR-JM*, vol. 12, pp9–16, 2016.
[29] S. Y. Yan, "Logarithm based cryptography" in *Cybercryptography, Applicable Cryptography for Cyberspace Security*. Springer, 2019, pp287–341.
[30] J. Ha, "On the least prime primitive root," *J. Number Theor.*, vol. 133, no. 11, pp3645–3669, 2013.
[31] S. D. Cohen and T. Trudgian, "On the least square-free primitive root modulo p," *J. Number Theor.*, vol. 170, pp10–16, 2017.
[32] G. Effinger, "On generalizing a corollary of Fermat's little theorem," *Math. Intelligencer*, vol. 41, no. 4, pp10–12, 2019.
[33] E. Boag, "Lattice multiplication," *Hist. Math.*, vol. 22, no. 3, pp182–184, 2007.
[34] C. Somboonpattanakit and N. Wisitpongphan, "Secure password storing using prime decomposition," *IAENG Int. J. Comput. Sci.*, vol. 48, no. 1, pp52-160, 2021.
[35] E. B. Nababan, G. T. Simbolon, O. S. Sitompul, "Multi-LSB and modified vernam cipher to enhance document file security," *IAENG Int. J. Comput. Sci.*, vol. 47, no. 4, pp705-712, 2020.
[36] A. Krikun and A. Levina, "Parallelized Montgomery exponentiation in GF(2^k) for Diffie–Hellman Key Exchange Protocol," *Engineering Letters*, vol. 29, no. 2, pp645-649, 2021.
[37] L. R. Knudsen and M. J. B. Robshaw, "The block cipher companion" in *Inf. Sec. Cryptogr.* Berlin, Heidelberg: Springer, Oct. 2011, pp1–12.
[38] L. R. Knudsenand and M. J. B. Robshaw, "Using block ciphers" in *Inf. Sec. Cryptogr.* Berlin, Heidelberg: Springer, pp65–94, 2011.
[39] S. G. Singh, "A study of encryption algorithms (RSA, DES, 3DES and AES) for information security," *Int. J. Comput. Appl.*, vol. 67, pp33–38, 2013.
[40] T. Xiang et al., "A novel block cryptosystem based on iterating a chaotic map," *Phys. Lett. A*, vol. 349, no. 1–4, pp109–115, 2006.
[41] A. Biryukov and E. Kushilevitz, "From differential cryptanalysis to ciphertext-only attacks" in *Lect. Notes Comput. Sci*, vol. 1462, H. Krawczyk, Ed. Berlin, Heidelberg: Springer, 2017, pp72–88.
[42] P. C. van Oorschot and M. J. Wiener, "A known-plaintext attack on two-key triple encryption" in *Sci,* Advances in Cryptology — EUROCRYPT" 90, I. B. Damgård, Ed. *Lect. Notes Comput.* Berlin, Heidelberg: Springer, 1990, pp318–325.

Harunori Nakayama completed his Bachelor of Engineering from Department of Chemistry and Chemical Engineering, Faculty of Engineering, Niigata University. He worked as a mechanical engineer at an extrusion plant manufacturer for 18 years. Moreover, he pursued his masters degree in Arts and Sciences at the School of Graduate Studies, the Open University of Japan. Currently, he is a sole proprietor and runs a retail business. He is an undergraduate student at Faculty of Liberal Arts, the Open University of Japan.

Seiji Anbe completed his Bachelor of Science from Tokyo University of Science Faculty of Science Division, Department of Mathematics. He was engaged as a mathematics teacher at several high schools for 37 years. Moreover, he completed Bachelor of Arts from Waseda University Department of English Literature. He was a student at the School of Graduate Studies, the Open University of Japan, for 5 years. He had supervised a cram school after retirement: Learning Forest "English and Mathematics Seminar". Currently, he is a President of the cram school.