cryptography

MDPI

Article

Montgomery Reduction for Gaussian Integers †

Malek Safieh and Jürgen Freudenberger *

Institute for System Dynamics (ISD), HTWG Konstanz, University of Applied Sciences, 78462 Konstanz, Germany; msafieh@htwg-konstanz.de

* Correspondence: jfreuden@htwg-konstanz.de; Tel.: +49-7531-206-647

† This paper is an extended version of our paper Safieh, M.; Freudenberger, J. Montgomery Modular Arithmetic over Gaussian Integers. In Proceedings of the 24th International Information Technology Conference (IT), Zabljak, Montenegro, 18–22 February 2020; pp. 1–4.

Abstract: Modular arithmetic over integers is required for many cryptography systems. Montgomery reduction is an efficient algorithm for the modulo reduction after a multiplication. Typically, Montgomery reduction is used for rings of ordinary integers. In contrast, we investigate the modular reduction over rings of Gaussian integers. Gaussian integers are complex numbers where the real and imaginary parts are integers. Rings over Gaussian integers are isomorphic to ordinary integer rings. In this work, we show that Montgomery reduction can be applied to Gaussian integer rings. Two algorithms for the precision reduction are presented. We demonstrate that the proposed Montgomery reduction enables an efficient Gaussian integer arithmetic that is suitable for elliptic curve cryptography. In particular, we consider the elliptic curve point multiplication according to the randomized initial point method which is protected against side-channel attacks. The implementation of this protected point multiplication is significantly faster than comparable algorithms over ordinary prime fields.

Keywords: public-key cryptography; elliptic curve point multiplication; Gaussian integers; Montgomery modular reduction

Citation: Safieh, M.; Freudenberger, J. Montgomery Reduction for Gaussian Integers. *Cryptography* **2021**, *5*, 6. https://doi.org/10.3390/cryptography5010006

Received: 13 January 2021
Accepted: 29 January 2021
Published: 1 February 2021

1. Introduction

Montgomery reduction is an efficient method that performs modulo reduction after an integer multiplication. The reduction algorithm uses multiplication and simple bit-wise operations instead of an expensive division [1]. Public-key cryptography based on the Rivest-Shamir-Adleman (RSA) system benefits from the efficient Montgomery modulo reduction [2,3]. Similarly, elliptic curve cryptography (ECC) systems over prime fields apply Montgomery reduction to support arbitrary prime curves [4–14].

In this work, we consider the modulo reduction for Gaussian integers. Gaussian integers are complex numbers such that the real and imaginary parts are integers. Arithmetic over Gaussian integers for the RSA system was proposed in [15–18]. In [17,18], it was shown for the RSA system that a significant complexity reduction can be achieved with Gaussian integers. Due to the isomorphism between Gaussian integer rings and ordinary integer rings, this arithmetic is suitable for many cryptography systems. The Rabin cryptography system and elliptic curve cryptography over Gaussian integers were considered in [19–22]. Moreover, coding applications over Gaussian integers use Gaussian integer arithmetic [23–28].

To reduce the complexity of the modular reduction, we investigate Montgomery reduction algorithms for finite Gaussian integer rings. However, it is not trivial to generalize Montgomery reduction to Gaussian integers. The final step of the reduction algorithm is based on the total order of integers which does not exist for complex numbers. In [29], a Montgomery reduction algorithm was proposed that utilizes the absolute value to measure the size of a Gaussian integer. This study is an extension of our paper [29]. In this work, we

present a new approach that aims on reducing the complexity of the reduction presented in [29]. In this algorithm, the absolute value is replaced by the Manhattan weight. The calculation of the Manhattan weight requires only a single addition, whereas the calculation of the absolute value requires addition and two squaring operations. On the other hand, the reduction using the Manhattan weight may not always obtain a unique solution. There are at most two possible solutions that are congruent. For many applications uniqueness is not an issue for the intermediate results in calculations, which require many subsequent reduction steps. For the final result, the correct representative can be calculated using absolute values.

The proposed concept of Montgomery reduction was applied in [22] for the efficient calculation of the elliptic curve point multiplication. In particular, an area-efficient co-processor was designed in [22]. This processor uses the proposed Montgomery modular arithmetic over Gaussian integers. It is shown in [21,22] that a key representation with a Gaussian integer expansion is beneficial for the calculation of the point multiplication. This key representation reduces the computational complexity and the memory requirements of a secure hardware implementation, which is protected against side-channel attacks [30–37]. In this work, we provide the theoretical justification for the reduction algorithms applied in [22]. Furthermore, we present new results for protected implementations of the point multiplication. In particular, we consider the randomized initial point method proposed in [38]. This method is protected against several side-channel attacks, i.e., differential power analysis (DPA), zero-value point attacks (ZPA), and refined power analysis (RPA) [32,38]. We present synthesis results for a field-programmable gate array (FPGA) based on the architecture proposed in [22]. The protected implementation of the point multiplication is significantly faster with Gaussian integers. Moreover, it requires less memory and fewer flip-flops than the PM algorithms over ordinary prime fields reported in [39,40].

This publication is organized as follows. We review the Montgomery multiplication of ordinary integers and discuss some properties of Gaussian integers in Section 2. In Section 3, Montgomery reduction based on the absolute value is investigated. We present a low-complexity reduction algorithm based on the Manhattan weight in Section 4. To demonstrate that the arithmetic over Gaussian integers is useful, we consider the complexity for the elliptic curve point multiplication according to the randomized initial point method in Section 5. In Section 6, we discuss the results and conclude our work.

2. Preliminaries and Problem Statement

In this section, we briefly discuss Montgomery reduction of ordinary integers. Moreover, we review some properties of Gaussian integer rings.

2.1. Montgomery Reduction of Ordinary Integers

Montgomery reduction is considered in several ECC and RSA cryptography systems to decrease the complexity of the arithmetic in prime fields [9,11,14] as well as in rings [2,3]. The Montgomery multiplication uses the arithmetic over a ring $\mathcal{R}_n$ that is isomorphic to the integer ring $\mathbb{Z}_n = \{x \bmod n : x = 0, \ldots, n-1, x \in \mathbb{Z}\}$ [1]. Montgomery reduction algorithm reduces the complexity of the modulo reduction. The expensive calculation mod n is replaced by mod R, where $R > n$ is a power of two. Hence, the modulo reduction in the ring $\mathcal{R}_n$ is a simple bitwise AND operation with $R - 1$.

For the Montgomery arithmetic, an element $x \in \mathbb{Z}_n$ is mapped to the Montgomery domain as

$$X = xR \bmod n. \tag{1}$$

Addition in the Montgomery domain is isomorphic to ordinary integer modular arithmetic, because

$$(X + Y) \bmod n = (xR + yR) \bmod n = (x + y)R \bmod n. \tag{2}$$

The multiplication in the Montgomery domain needs Montgomery reduction function $\mu(Z) = ZR^{-1} \bmod n$ described in Algorithm 1. This function defines the inverse mapping, since $\mu(X) = xRR^{-1} \bmod n = x \bmod n$. Using the reduction function, we obtain the product

$$\mu(XY) = XYR^{-1} \bmod n = xyR \bmod n \tag{3}$$

in the Montgomery domain.

Algorithm 1 Montgomery reduction according to [1].

input: Z, with $0 \leq Z < nR$, $n' = -n^{-1} \bmod R$, and $R = 2^l \geq n$
output: $M = \mu(Z) = ZR^{-1} \bmod n$

1: $t = Zn' \bmod R$ // bitwise AND with $R - 1$
2: $q = (Z + tn) \operatorname{div} R$ // shift right by l
3: **if** $(q \geq n)$ **then**
4: $\quad M = q - n$
5: **else**
6: $\quad M = q$
7: **end if**

Typically, all variables are mapped at the beginning of the calculation into the Montgomery domain using this reduction function as

$$X = \mu(xR^2) = xR^2R^{-1} \bmod n = xR \bmod n, \tag{4}$$

since only one precomputed value $R^2 \bmod n$ is required.

We illustrate the Montgomery reduction of ordinary integers with an example. The binary representation of positive integers is denoted by brackets $(\cdot)_2$ with the subscript 2.

Example 1. *We consider the product of two numbers* $x = 14$ *and* $y = 7$ *from* $\mathbb{Z}_{29}$. *The two integers x and y are mapped to the integers* $X = xR \bmod n = 13$ *and* $Y = yR \bmod n = 21$ *with* $R = 32 = 2^5 > n = 29$ *in the Montgomery domain. Hence, all elements of the Montgomery domain can be represented with five binary digits. The product* $xy \bmod n = 98 \bmod 29 = 11$ *corresponds to* $xyR \bmod n = 4$ *in the Montgomery domain.*

With Algorithm 1, we can reduce the product $Z = XY = 273$ *as follows. With* $n' = 11$, *we first calculate* $t = Zn' \bmod R = 3003 \bmod 32 = 27$. *Note that the modulo reduction of the binary representation is a simple bitwise AND operation with* $R - 1 = 31$ *or equivalently the truncation of all bits except the five least significant bits. The integer 3003 has the binary representation* $(1011\,1011\,1011)_2$ *and the five least significant bits* $(1\,1011)_2$ *represent the value* $t = 27$.

Next, we calculate the quotient $q = (Z + tn) \operatorname{div} R = 1056 \operatorname{div} 32 = 33$. *For* $R = 32$, *the division by R without remainder in the binary representation is a simple right-shift by five bits or equivalently the truncation of the five least significant bits. The integer 1056 has the binary representation* $(100\,0010\,0000)_2$. *The division without remainder results in* $(10\,0001)_2$ *which is the binary representation of 33. The final result in the Montgomery domain is* $M = \mu(Z) = q - 29 = 4$ *because* $q = 33 > n = 29$. *To obtain the result in* $\mathbb{Z}_{29}$, *we apply Montgomery reduction for the inverse mapping* $M' = \mu(M) = \mu(4) = 11$.

2.2. Rings of Gaussian Integers

The modulo arithmetic over Gaussian integers is based the modulo function

$$x \bmod \pi = x - \left\lfloor \frac{x\pi^*}{\pi\pi^*} \right\rceil \cdot \pi, \tag{5}$$

where x and π are Gaussian integers. π^* is the conjugate of the Gaussian integer π. The brackets $\lfloor \cdot \rceil$ denote rounding to the closest Gaussian integer [23], i.e., for a complex number $x = c + d\mathrm{i}$, we have $\lfloor x \rceil = \lfloor c \rceil + \lfloor d \rceil \mathrm{i}$. The set

$$\mathcal{G}_p = \{x \bmod \pi : x = 0, \ldots, p-1, x \in \mathbb{Z}\} \tag{6}$$

is a finite ring. For primes p of the form $p \equiv 1 \bmod 4$, $\mathcal{G}_p$ is a finite field which is isomorphic to the prime field $GF(p)$ over ordinary integers [23]. Such fields are suitable for elliptic curve cryptography [21,22]. A prime of the form $p \equiv 1 \bmod 4$ is the sum of two perfect squares, i.e., $p = \pi\pi^* = |a|^2 + |b|^2$ with the Gaussian integer $\pi = a + b\mathrm{i}$.

Moreover, for integers $n = cd$ such that c and d are primes of the form $c \equiv d \equiv 1 \bmod 4$ and $c \neq d$, the set $\mathcal{G}_n$ is a ring that is isomorphic to the ring $\mathbb{Z}_n$ over ordinary integers [26]. Such Gaussian integers are suitable for the RSA public-key system [17,18].

Consider Montgomery reduction in Algorithm 1. Lines 3 to 7 of this algorithm uniquely determine the smallest integer that is congruent to $ZR^{-1} \bmod n$. However, complex numbers cannot be totally ordered [24]. Hence, it is not possible to directly apply this reduction algorithm to Gaussian integers. This reduction problem is not regarded in [17,18].

In this work, we consider two reduction strategies using different norms. One algorithm uses the absolute value $|x| = \sqrt{|c|^2 + |d|^2}$ of the Gaussian integer $x = c + d\mathrm{i}$. The second approach is based on the Manhattan weight $\|x\| = |c| + |d|$. Calculating the Manhattan weight is less complex than calculating the absolute value, because only addition is required, whereas squaring is necessary to determine $|x|$. Note that

$$|x| \leq \|x\| \tag{7}$$

holds for any x which follows from squaring both sides of the inequality.

2.3. Finding Primes of the Form $p = a^2 + b^2$

An algorithm for primes of the form $p \equiv 1 \bmod 4$ to find a, b such that $p = a^2 + b^2$ is proposed in [23]. This algorithm consists of two steps.

- Find x such that $x^2 \equiv -1 \bmod p$, which can be done using the Tonelli-Shanks algorithm [41].
- Use the Euclidean algorithm to determine the greatest common divisor of p and x. The first two remainders less than $\sqrt{p}$ are a and b.

On the other hand, there exist many primes of the form $p \equiv 1 \bmod 4$ such that $p = a^2 + (a-1)^2$ or $p = a^2 + 1$. Exploiting this observation we can search for a such that the sum $p = a^2 + 1$ or $p = 2a^2 - 2a + 1$ is prime. This can significantly reduce the search complexity. Table 1 illustrates some examples of such primes with sufficient bit lengths for ECC applications.

Table 1. Examples for primes of the form $p = a^2 + b^2$ with $b = 1$ or $b = a - 1$.

bits	p	a	b
156	$8 \times 10^{46} + 74 \times 10^{24} + 17113$	$2 \times 10^{23} + 93$	$2 \times 10^{23} + 92$
169	$4 \times 10^{50} + 216 \times 10^{25} + 2917$	$2 \times 10^{25} + 54$	1
170	$8 \times 10^{50} + 6 \times 10^{26} + 113$	$2 \times 10^{25} + 8$	$2 \times 10^{25} + 7$
190	$8 \times 10^{56} + 3 \times 10^{30} + 2813$	$2 \times 10^{28} + 38$	$2 \times 10^{28} + 37$
256	$8 \times 10^{76} + 2516 \times 10^{38} + 197821$	$2 \times 10^{38} + 315$	$2 \times 10^{38} + 314$
382	$9 \times 10^{114} + 384 \times 10^{57} + 4097$	$3 \times 10^{57} + 64$	1

3. Montgomery Reduction for Gaussian Integers

In this section, we consider some important properties of Gaussian integers and derive Montgomery reduction for Gaussian integers using the absolute value. First, we show that the set $\mathcal{G}_n$ is symmetric and that the absolute value of any element of this ring is bounded.

Lemma 1. *For any $x \in \mathcal{G}_n$ we have the following symmetry*

$$-x, \mathrm{i}x, -\mathrm{i}x \in \mathcal{G}_n. \tag{8}$$

The absolute value $|x|$ is bounded by

$$|x| < \frac{|\pi|}{\sqrt{2}}. \tag{9}$$

Moreover, for any $x' \notin \mathcal{G}_n$ with $x = x' \bmod \pi$ we have

$$|x| < |x'|. \tag{10}$$

Proof. Consider the quotient $\frac{x}{\pi} = \frac{x\pi^*}{n} = c + d\mathrm{i}$, where c and d are the real part and the imaginary part of $\frac{x}{\pi}$. For $x \in \mathcal{G}_n$ we have $x = x \bmod \pi$. Hence, the rounded quotient satisfies

$$\left[\frac{x\pi^*}{\pi\pi^*}\right] = 0.$$

This implies $[c] = [d] = 0$ and $|c| < 1/2$, $|d| < 1/2$. Hence, we can bound the absolute value of the quotient $\frac{x}{\pi}$ as

$$\left|\frac{x}{\pi}\right| < \frac{1}{\sqrt{2}}.$$

Multiplying both sides by $|\pi|$ results in (9). Note that $x, -x, ix$, or $-ix$ have the same absolute value. Hence, (8) holds.

Finally, consider a Gaussian integer $x' \notin \mathcal{G}_n$ which is congruent to x, i.e. $x = x' \bmod \pi$. We use the notation $\frac{x'}{\pi} = c' + d'\mathrm{i}$. Note that $x = x' \bmod \pi$ implies $c = c' - [c']$ and $d = d' - [d']$. Moreover, at least one rounded value $[c']$ or $[d']$ is nonzero because $x' \notin \mathcal{G}_n$, where $[c'] \neq 0$ results in $|c| < 1/2 < |c'|$. Similarly, $[d'] \neq 0$ implies $|d| < 1/2 < |d'|$. Hence, we have (10). □

The next example demonstrates that $|x| < \frac{|\pi|}{\sqrt{2}}$ is not a sufficient condition for $x \in \mathcal{G}_n$.

Example 2. *We consider the finite field $\mathcal{G}_{29}$ for the Gaussian integer $\pi = 5 + 2\mathrm{i}$. All elements of this field are depicted in Figure 1. Consider $q' = 3$ with $q = 3 \bmod \pi = -2 - 2\mathrm{i}$, i.e., $q' \notin \mathcal{G}_{29}$*

and $q \in \mathcal{G}_{29}$. *Both Gaussian integers satisfy* $|q'| < \frac{|\pi|}{\sqrt{2}}$ *and* $|q| < \frac{|\pi|}{\sqrt{2}} \approx 3.8079$. *Hence, it is not possible to determine the representative based on the bound (9). The representative* $q = -2 - 2\mathrm{i}$ *follows from (10), as* $|q| \approx 2.8284 < |q'| = 3$.

Next, we consider now Montgomery reduction. The reduction for Gaussian integers can be performed according to Algorithm 2, where

$$\alpha' = \underset{\alpha \in \{0, \pm 1, \pm \mathrm{i}\}}{\operatorname{argmin}} |q - \alpha\pi|, \tag{11}$$

$$\alpha'' = \underset{\alpha \in \{\pm 1, \pm \mathrm{i}, \pm 1 \pm \mathrm{i}\}}{\operatorname{argmin}} |q - \alpha\pi|. \tag{12}$$

Steps 1 and 2 in Algorithm 2 are applied separately for the real- and imaginary part. We demonstrate that Algorithm 2 always obtains a precision reduction resulting in the correct representative.

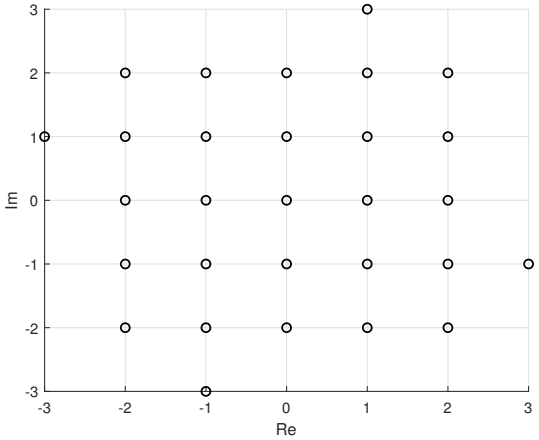

Figure 1. Elements of the Gaussian integer field $\mathcal{G}_{29}$ with $\pi = 5 + 2\mathrm{i}$.

Algorithm 2 Montgomery reduction for Gaussian integers

input: $Z = XY$, $\pi' = -\pi^{-1} \bmod R$, $R = 2^l > \frac{|\pi|}{\sqrt{2}}$
output: $M = \mu(Z) = ZR^{-1} \bmod \pi$

1: $t = Z\pi' \bmod R$ // bitwise AND of Re, Im with $R - 1$
2: $q = (Z + t\pi) \operatorname{div} R$ // shift Re, Im right by l
3: **if** $(|q| < \frac{\sqrt{2}-1}{\sqrt{2}}|\pi|)$ **then**
4: $M = q$
5: **else if** $(|q| < \frac{|\pi|}{\sqrt{2}})$ **then**
6: determine α' according to (11)
7: $M = q - \alpha'\pi$
8: **else**
9: determine α'' according to (12)
10: $M = q - \alpha''\pi$
11: **end if**

Proposition 1. *For M and Z according to Algorithm 2, we have*

$$M = ZR^{-1} \bmod \pi. \tag{13}$$

Proof. We consider the sum $Z + t\pi$ in step 2 of Algorithm 2, where

$$\begin{aligned} Z + t\pi &\equiv Z + Z\pi'\pi \bmod R \\ &\equiv Z + Z(-\pi^{-1})\pi \bmod R \\ &\equiv Z - Z \bmod R \equiv 0, \end{aligned}$$

implies that R divides $Z + t\pi$. Considering the corresponding quotient $q = (Z + t\pi) \operatorname{div} R$, we have

$$\begin{aligned} q \bmod \pi &\equiv (Z + t\pi) \operatorname{div} R \bmod \pi \\ &\equiv (Z + t\pi)R^{-1} \bmod \pi \\ &\equiv ZR^{-1} + t\pi R^{-1} \bmod \pi \\ &\equiv ZR^{-1} \bmod \pi. \end{aligned}$$

This shows that q is congruent to $ZR^{-1} \bmod \pi$ or $ZR^{-1} = q - \alpha\pi$. The absolute value of $ZR^{-1} \bmod \pi$ is bounded, i.e.,

$$\left|ZR^{-1}\right| = |XY|R^{-1} \leq \frac{|\pi|^2}{2R} < \frac{R|\pi|}{\sqrt{2}R} = \frac{|\pi|}{\sqrt{2}}.$$

As shown in Example 2, the quotient q may not be the correct representative $ZR^{-1} \bmod \pi$ even for $|q| < \frac{|\pi|}{\sqrt{2}}$. The congruent values $q - \alpha\pi$ with $\alpha \in \{\pm 1, \pm \mathrm{i}\}$ are also possible candidates. However, for any $x \in \mathcal{G}_n$ and $\alpha \in \{\pm 1, \pm \mathrm{i}\}$, the lower bound

$$|x - \alpha\pi| \geq ||\pi| - |x|| > \frac{\sqrt{2}-1}{\sqrt{2}}|\pi|,$$

follows from the triangle inequality and Lemma 1. Consequently, the quotient q is the unique solution for $|q| < \frac{\sqrt{2}-1}{\sqrt{2}}|\pi|$. Furthermore, for any $x \in \mathcal{G}_n$ the condition $|\alpha| > \sqrt{2}$ implies

$$|x - \alpha\pi| \geq ||\alpha\pi| - |x|| > \frac{|\pi|}{\sqrt{2}}.$$

Hence, only the candidates according to (11) are possible for $\frac{\sqrt{2}-1}{\sqrt{2}}|\pi| \leq |q| < \frac{|\pi|}{\sqrt{2}}$. Using Lemma 1, it follows that the correct candidate can be found by minimizing the absolute value among all candidates according to (11).

If $|q| \geq \frac{|\pi|}{\sqrt{2}}$, the result M is calculated as $M = q - \alpha''\pi$ with α'' according to (12). To demonstrate that (12) is correct, we derive the bound $|\alpha| \leq \sqrt{2}$. Consider again the quotient $q = (Z + t\pi) \operatorname{div} R$. We aim to compensate the offset $t\pi R^{-1}$ by $\alpha\pi$, where the absolute values are bounded by

$$|\alpha\pi| = \left|t\pi R^{-1}\right| = |t||\pi|R^{-1} \leq \sqrt{2}|\pi|.$$

This follows from $|t| \leq \sqrt{R^2 + R^2} = \sqrt{2}R$ due to the reduction mod R and implies the bound $|\alpha| \leq \sqrt{2}$. □

Example 3. *As an example for Montgomery reduction, we consider the product of two numbers $x = 6$ and $y = 7$ from $\mathbb{Z}_{29}$ or the corresponding field $\mathcal{G}_{29}$ of Gaussian integers with $\pi = 5 + 2\mathrm{i}$.*

The two integers x and y are mapped to the Gaussian integers $X = xR \bmod \pi = 2 - \mathrm{i}$ and $Y = yR \bmod \pi = -2$ with $R = 8$ in the Montgomery domain. The product $xy \bmod p = 42 \bmod 29 = 13$ corresponds to $xyR \bmod \pi = -\mathrm{i}$ in the Montgomery domain.

With Algorithm 2, we can reduce the product $Z = XY = -4 + 2\mathrm{i}$ as follows. With $\pi' = -1 + 2\mathrm{i}$, we first calculate $t = Z\pi' \bmod R = -10\mathrm{i} \bmod 8 = -2\mathrm{i}$. Note that the modulo reduction is a simple bitwise AND operation with $R - 1 = 7$.

Next, we calculate the quotient $q = (Z + t\pi) \operatorname{div} R = -8\mathrm{i} \operatorname{div} 8 = -\mathrm{i}$. For $R = 8 = 2^3$, the division by R without remainder is a simple right-shift by three bits. The quotient $M = \mu(Z) = q = -\mathrm{i}$ is the final result without further reduction steps because $|q| = 1 < \frac{\sqrt{2}-1}{\sqrt{2}}|\pi| = 1.5773$.

The product $xy \bmod p = 13$ corresponds to $xy \bmod \pi = 1 + \mathrm{i}$ in the field $\mathcal{G}_{29}$. To obtain this result, we apply Montgomery reduction for the inverse mapping $M' = \mu(M) = MR^{-1}$. With $M = -\mathrm{i}$, we have $t = Z\pi' \bmod R = 2 + \mathrm{i} \bmod 8 = 2 + \mathrm{i}$ and the result $M' = q = (Z + t\pi) \operatorname{div} R = 8 + 8\mathrm{i} \operatorname{div} 8 = 1 + \mathrm{i}$. Again, no reduction is required because the condition $|q| = \sqrt{2} < \frac{\sqrt{2}-1}{\sqrt{2}}|\pi| = 1.5773$ is satisfied.

We illustrate the final reduction procedure of Algorithm 2 in the following example. There are up to eight possible values α according to (11) and (12). However, not all potential values have to be considered for the reduction. The number of valid candidates α can be reduced based on the signs of the real and imaginary part of q.

Example 4. *We consider again the finite field $\mathcal{G}_{29}$ of Gaussian integers with $\pi = 5 + 2\mathrm{i}$. Figure 2 depicts all possible quotients q in the Montgomery domain after the first two steps of Algorithm 2. For $q = 3$ we have $\frac{\sqrt{2}-1}{\sqrt{2}}|\pi| \approx 1.5773 < |q| = 3 < \frac{|\pi|}{\sqrt{2}} \approx 3.8079$. Thus, α' should be determined according to (11). However, there are only two possible values for α, i.e., $\alpha \in \{0, 1\}$, as q is in the first quadrant. Calculating $|q - 0| = 3$ and $|q - \pi| \approx 2.3852$ leads to $\alpha' = 1$. Consequently, we obtain the representative $M = q - \alpha'\pi = -2 - 2\mathrm{i}$.*

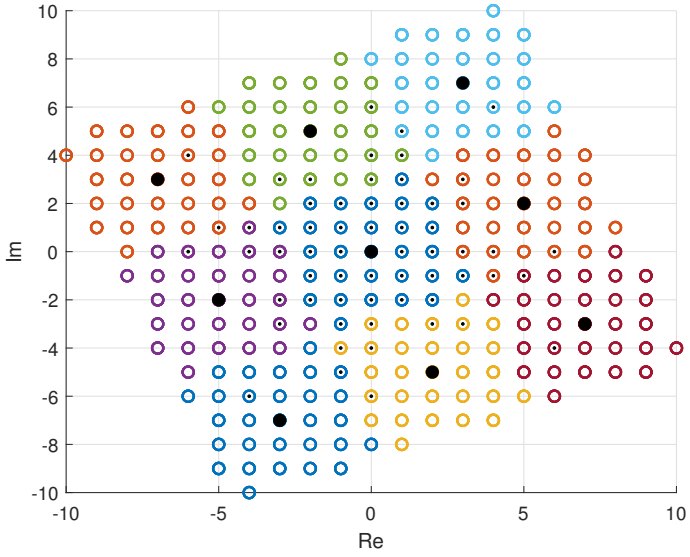

Figure 2. Illustration of the the Montgomery domain for $p = 29$ and $\pi = 5 + 2\mathrm{i}$. Circles centered with a point are possible quotients q after Montgomery reduction and filled circles are possible offsets $\alpha\pi$.

Algorithm 2 achieves the desired precision reduction, where the final reduction step always results in the correct representative $M = q \bmod \pi$. This result satisfies the bound $|M| \leq \frac{|\pi|}{\sqrt{2}}$. The number of possible candidates α is restricted depending on the absolute value of q, as illustrated in (11) and (12).

Nonetheless, the reduction according to Algorithm 2 can be rather complex due to the squaring to determine the absolute values. This complexity may not be required in all applications of the Montgomery multiplication. For applications in cryptography, it is adequate to find the unique representative $ZR^{-1} \bmod \pi$ only in the last calculation step, whereas for intermediates results a reduction that achieves $M \equiv ZR^{-1} \bmod \pi$ is sufficient. This motivates a low complexity Montgomery reduction based on the Manhattan weight, which is considered in the next section.

4. Precision Reduction for Gaussian Integers Using the Manhattan Weight

In this section, we present a low complexity precision reduction for Gaussian integers based on the Manhattan weight, i.e., the absolute value in the reduction algorithm is replaced by the Manhattan weight. The calculation of the absolute value requires squaring, whereas the Manhattan weight requires only addition. On the other hand, this algorithm may not obtain a unique solution. However, there are at most two possible solutions that are congruent. In ECC or RSA systems that require many reduction steps, this algorithm can be used for the intermediate results.

Without loss of generality we consider $\pi = a + b\mathrm{i}$ with $a > b \geq 1$. The precision reduction is described in Algorithm 3, where

$$\hat{\alpha} = \underset{\alpha \in \{\pm 1, \pm \mathrm{i}, \pm 1 \pm \mathrm{i}\}}{\operatorname{argmin}} \|q - \alpha\pi\|, \tag{14}$$

and $N = a - 1$. We demonstrated that this algorithm always obtains $M \equiv ZR^{-1} \bmod \pi$, where $\|M\| \leq N$.

Algorithm 3 Precision reduction for Gaussian integers

input: $Z = XY$, $\pi' = -\pi^{-1} \bmod R$, $R = 2^l > N$
output: $M \equiv \mu(Z) = ZR^{-1} \bmod \pi$

1: $t = Z\pi' \bmod R$ // bitwise AND of Re, Im with $R - 1$
2: $q = (Z + t\pi) \operatorname{div} R$ // shift Re and Im right by l
3: **if** $(\|q\| \leq N)$ **then**
4: $\quad M = q$
5: **else**
6: $\quad$ determine $\hat{\alpha}$ according to (14)
7: $\quad M = q - \hat{\alpha}\pi$
8: **end if**

First, we note that using the symmetry according to Lemma 1, we can restrict the following derivations to the elements of the first quadrant. Next, we consider some important properties of the Manhattan weight.

Lemma 2. *For $x \in \mathcal{G}_p$ the Manhattan weight is upper bounded by*

$$\|x\| \leq a - 1 = N. \tag{15}$$

Proof. Let c and d be the real part and the imaginary part of $\frac{x}{\pi}$. For $x \in \mathcal{G}_p$ we have $x = x \bmod \pi$ and consequently

$$\left\lceil \frac{x\pi^*}{\pi\pi^*} \right\rfloor = 0. \tag{16}$$

This implies $\lceil c \rfloor = \lceil d \rfloor = 0$ and $|c| < 1/2$, $|d| < 1/2$. Hence, we consider $(1/2 + 1/2\mathrm{i})\pi$ to upper bound the real and imaginary parts of x as

$$\begin{aligned} |\mathrm{Re}\{\mathrm{x}\}| &< \left| \frac{a-b}{2} \right|, \\ |\mathrm{Im}\{\mathrm{x}\}| &< \left| \frac{a+b}{2} \right|. \end{aligned}$$

Note that either a is odd and b is even or vice versa because p is an odd prime. Furthermore, the real- and imaginary parts of x are integers. Consequently, we have

$$\begin{aligned} |\mathrm{Re}\{\mathrm{x}\}| &\leq \left| \frac{a-b}{2} - \frac{1}{2} \right|, \\ |\mathrm{Im}\{\mathrm{x}\}| &\leq \left| \frac{a+b}{2} - \frac{1}{2} \right|. \end{aligned}$$

We can bound the Manhattan weight of x as

$$\|x\| \leq \left\| \frac{a-b-1}{2} + \frac{a+b-1}{2}\mathrm{i} \right\|.$$

With $a > b \geq 1$, the Manhattan weight of x is upper bounded by

$$\|x\| \leq \frac{a-b-1}{2} + \frac{a+b-1}{2} = a - 1. \tag{17}$$

Hence (15) holds. □

Next, we consider bounds on the Manhattan weight for the sum and product of two elements $x, y \in \mathcal{G}_p$. Note that we consider arithmetic without modulo reduction.

Lemma 3. *For $x, y \in \mathcal{G}_p$ we have the upper bounds*

$$\begin{aligned} \|x + y\| &\leq 2N, & (18) \\ \|xy\| &\leq N^2, & (19) \end{aligned}$$

for arithmetic without modulo reduction.

Proof. The bound on the sum follows from the triangle inequality and (15), i.e.,

$$\|x + y\| \leq \|x\| + \|y\| \leq 2N. \tag{20}$$

Without loss of generality we consider two elements $x = c + d\mathrm{i}$ and $y = e + f\mathrm{i}$ from the first quadrant for the product in (19). This implies $\|x\| = c + d \leq N$ or $d \leq N - c$. Similarly, we have $f \leq N - e$. For the product xy we have

$$xy = (ce - df) + (ed + cf)\mathrm{i}. \tag{21}$$

First, consider the absolute value of the imaginary part $\mathrm{Im}\{\mathrm{xy}\}$

$$|\mathrm{Im}\{\mathrm{xy}\}| = ed + cf \leq e(N - c) + c(N - e) = eN + cN - 2ce.$$

To determine the maximum value, we consider the bivariate function $g(e,c) = eN + cN - 2ce$ and its partial derivatives

$$\frac{\partial g(e,c)}{\partial e} = N - 2c = 0, \tag{22}$$

$$\frac{\partial g(e,c)}{\partial c} = N - 2e = 0. \tag{23}$$

This results in a maximum for $c = e = N/2$ and the bound $|\mathrm{Im}\{xy\}| = ed + cf \leq N^2/2$. Due to symmetry this bound also holds for the absolute value of the real part. Hence, we have

$$\|xy\| \leq \frac{N^2}{2} + \frac{N^2}{2} = N^2. \tag{24}$$

□

The following proposition demonstrates that Algorithm 3 results in the desired reduction.

Proposition 2. *For M and Z according to Algorithm 3, we have*

$$M \equiv ZR^{-1} \bmod \pi, \tag{25}$$

$$\|M\| \leq N. \tag{26}$$

Proof. The first two steps are identical in Algorithm 3 and Algorithm 2. Hence, the first statement (25) follows from the same arguments as in the proof of Proposition 1.

For $\|q\| \leq N$, we immediately have (26). For $\|q\| > N$, we consider again the corresponding quotient $q = (Z + t\pi) \operatorname{div} R$. The Manhattan weight of $ZR^{-1} \bmod \pi$ is bounded by

$$\left\|\frac{Z}{R}\right\| = \frac{\|Z\|}{R} = \frac{\|XY\|}{R} \leq \frac{N^2}{R} \leq \frac{NR}{R} = N, \tag{27}$$

where we have used (24) and the assumption $R \geq N$.

Similar to the proof of Proposition 1, we aim to compensate $t\pi R^{-1}$ with $\alpha\pi$. From Proposition 1 follows that $\|\alpha\| \leq \sqrt{2}$. Hence, the minimization in (14) over $\alpha \in \{\pm 1, \pm \mathrm{i}, \pm 1 \pm \mathrm{i}\}$ is sufficient. From (27) follows that there is at least one solution with $\|M\| = \|q - \hat{\alpha}\pi\| \leq N$. Consequently, the minimization in (14) will find a solution satisfying (26). □

Finally, we can now describe the final reduction step of Algorithm 3. Note that there are eight possible values for α according to (14), but not all potential values have to be considered. The reduction procedure is demonstrated in the following example.

Example 5. *Again, we consider the finite field $\mathcal{G}_{29}$ of Gaussian integers with $\pi = 5 + 2\mathrm{i}$, where all possible values of q after the first two steps of Algorithm 3 are depicted in Figure 2.*

For instance, consider the product of $x = 19$ and $y = 7$ from $\mathbb{Z}_{29}$. The two integers x and y are mapped to the Gaussian integers $X = xR \bmod \pi = 2 - 2\mathrm{i}$ and $Y = yR \bmod \pi = -2$ with $R = 8$ in the Montgomery domain. The product $xy \bmod p = 133 \bmod 29 = 17$ corresponds to $xyR \bmod \pi = 3 - \mathrm{i}$ in the Montgomery domain.

With Algorithm 2, we can reduce the product $Z = XY = -4 + 4\mathrm{i}$ as follows. We calculate $t = Z\pi' \bmod R = 4 + 4\mathrm{i}$ and the quotient $q = (Z + t\pi) \operatorname{div} R = 1 + 4\mathrm{i}$. A valid solution of the reduction is a Gaussian integer M with $\|M\| \leq N$. We have $\|q\| = 5 > N = 4$. Hence, further reduction is required. As q is in the first quadrant, we have three possible values for α, i.e., $\alpha \in \{1, \mathrm{i}, 1 + \mathrm{i}\}$. Calculating $q - \alpha\pi$ results in $-4 + 2\mathrm{i}, 3 - \mathrm{i}, -2 - 3\mathrm{i}$, where only the solution $M = 3 - \mathrm{i}$ satisfies the condition $\|M\| \leq N = 4$. Hence, we choose $M = 3 - \mathrm{i}$ as the final result.

The final reduction step results in a Gaussian integer x with $\|x\| \leq N$. Hence, Algorithm 3 achieves the desired precision reduction using the Manhattan weight. However,

the result may not always be an element of the ring. For instance, for some π the value $x = a - 1$ is a ring element. Alternatively, the point $x' = x - \pi = -1 - b\mathrm{i}$ can be a ring element. Both values are congruent and can satisfy $\|x\| = a - 1 \leq N$, $\|x'\| = b + 1 \leq N$ depending on π. The ring element is the value with the smallest absolute value. There exist at most two congruent values with Manhattan weight less or equal N. Consequently, the correct representation can be selected by minimizing the absolute value of these two congruent values. Depending on the application this step might be required once to obtain the final result $x = q \bmod \pi$.

5. Elliptic Curve Point Multiplication

In this section, we consider the elliptic curve point multiplication to demonstrate that the arithmetic over Gaussian integers enables an efficient calculation. Calculations of the elliptic curve point multiplication are prone to side-channel attacks. For example, an attacker can estimate the secret key based on the required power consumption of the different arithmetic operations over time. There are many known attacks on the point multiplication like timing attacks, simple power analysis, differential power analysis, refined power analysis, and zero-value point attacks [32,38].

The concept for the point multiplication over Gaussian integers was introduced in [22], where an area-efficient coprocessor design was proposed with an arithmetic unit that enables Montgomery modular arithmetic over Gaussian integers. It is shown in [22] that a key representation with a Gaussian integer expansion is beneficial to reduce the computational complexity and the memory requirements of a secure hardware implementation considering timing attacks and simple power analysis. This architecture supports different point multiplication algorithms. In contrast to [22], we consider the randomized initial point method which is additionally protected against differential power analysis, refined power analysis, and zero-value point attacks [32,38]. We refer to [22] for the architecture and the implementation details.

In the following, we consider the elliptic curve

$$y^2 = x^3 + \alpha x + \beta, \tag{28}$$

which is recommended for prime fields $GF(p)$ [32,42]. The parameters α and β are constant coefficients. The pair x and y defines the coordinates of a point $P(x, y)$ on the curve. The one-way function of elliptic-curve cryptography is the point multiplication, i.e., kP, where P is a point on the elliptic curve and k is an integer. The point multiplication is typically implemented based on the binary expansion $k = \sum_{i=0}^{r-1} k_i 2^i$ of the integer k with binary digits $k_i \in \{0, 1\}$, where r is the length in bits and $k_{r-1}, k_{r-2}, \ldots, k_0$ is the binary representation of the key (the integer k). This method requires two different point operations, the point addition (ADD) and the point doubling operation (DBL) [32]. Let $P(x_1, y_1)$ and $Q(x_2, y_2)$ be two distinct points on an elliptic curve (28), then the sum $S(x_3, y_3) = P(x_1, y_1) + Q(x_2, y_2)$ is calculated as

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2, \quad y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1. \tag{29}$$

Similarly, the point doubling operation $S(x_3, y_3) = 2P(x_1, y_1)$ is defined as

$$x_3 = \left(\frac{3x_1^2 + \alpha}{2y_1}\right)^2 - 2x_1, \quad y_3 = \left(\frac{3x_1^2 + \alpha}{2y_1}\right)(x_1 - x_3) - y_1. \tag{30}$$

The calculations in (29) and (30) are typically performed using arithmetic over prime fields. However, we consider arithmetic over Gaussian integer fields, where $x_i, y_i, \alpha, \beta \in \mathcal{G}_p$.

As an alternative to the binary expansion key representation, τ-adic expansions of the integer k with a non-binary basis τ were proposed to speed up the point multiplication

(PM) and to improve the resistance against attacks [38,43–47]. The τ-adic expansion results in the point multiplication

$$kP = \sum_{i=0}^{l-1} \kappa_i \tau^i P = \tau(\ldots\tau(\tau\kappa_{l-1}P + \kappa_{l-2}P) + \ldots) + \kappa_0 P, \tag{31}$$

where we consider digits κ_i from a Gaussian integer field.

To demonstrate that a τ-adic expansion with arithmetic over Gaussian integers can reduce the computational complexity, we consider an example based on the curve (28) with $\beta = 0$. For the products $\kappa_i P(x, y)$ we use the endomorphism $\mathrm{i}P(x, y) = P(-x, \mathrm{i}y)$ for prime fields [32,48]. Note that $P(-x, \mathrm{i}y)$ is a point on the curve (28) if $P(x, y)$ is a valid point. The negation of any point $-P(x, y)$ is $P(x, -y)$ according to [32]. Hence, for the products $\kappa_i P(x, y)$ with $\kappa_i \in \{0, \pm 1, \pm \mathrm{i}\}$ we have

$$P = P(x, y), \tag{32}$$
$$-P = P(x, -y), \tag{33}$$
$$\mathrm{i}P = P(-x, \mathrm{i}y), \tag{34}$$
$$-\mathrm{i}P = P(-x, -\mathrm{i}y). \tag{35}$$

Using this endomorphism, we can calculate other products τP, e.g., $\tau P = (2 + \mathrm{i})P = 2P + \mathrm{i}P$ requires a DBL, the mapping $\mathrm{i}P$, and an ADD operation. Several other endomorphisms for different curves over prime fields can be fund in [32,48].

For the point multiplication, we consider the randomized initial point method according to Algorithm 4. This method introduces a random point R into the calculation of the point multiplication. Consequently, all intermediate results in the calculation of the point multiplication become randomized. The algorithm starts with a precomputation phase, where the points $S_{l-1}, \ldots, S_1, S_0$ are computed in steps 1 to 4 and stored in a memory. The point addition in step 5 results in the point $Q = S_{l-1} + R$. This point is multiplied with τ and then added to the corresponding precomputed point S_j in the loop in steps 6 to 9 of this algorithm. Due to the second product of the precomputations $-(\tau - 1) \cdot R$, we obtain the point $Q + R$ after each iteration of this loop. Correspondingly, the final result is the point $Q + R = k \cdot P + R$. Hence, we obtain the correct result of the point multiplication $k \cdot P$ by subtracting R from Q in step 10. The point addition in step 8 is computed in each iteration since the τ-adic expansion of the key according to [22] excludes all zero elements. This prevents SPA and timing attacks. Furthermore, all stored points $S_{l-1}, \ldots, S_1, S_0$ are randomized due to the subtracting the random point $-(\tau - 1) \cdot R$. Thus, the resistance against DPA, RPA, and ZPA is increased.

We can use the reduction based on the Manhattan weight in Algorithm 3 for all interim results in the point multiplication. The aim of the reduction of interim results is finding a Gaussian integer $\tilde{x}$ that has a Manhattan weight satisfying $\|\tilde{x}\| \leq N$ and is congruent to the actual representative x. The reduction algorithm can be stopped once a value $\tilde{x} = q - \alpha\pi$ with $\|\tilde{x}\| \leq N$ is found. Hence, not all offsets in (14) have to be considered in every reduction. Table 2 presents numerical results on the number of required reduction steps for different field sizes. The four columns on the right in Table 2 provide the percentage for the number of required offset reduction steps after an arithmetic operation, where no reduction is required if the result q already satisfies $\|q\| \leq N$. These results illustrate that sequential processing of the offset reduction is suitable for the point multiplication because 91% of all operations require no reduction since $\|q\| \leq N$ is satisfied. About $4 - 5\%$ of all operations require a single reduction step. Two or three calculation steps were required in approximately 4% of all cases.

Algorithm 4 Point multiplication according to the randomized initial point method from [38].

input: $P, k = (\kappa_{l-1}, \ldots, \kappa_1, \kappa_0)_\tau$
output: $k \cdot P$

1: $R = random\ point$ // randomized initial point
2: **for** $(j = l - 1$ **down to** $0)$ **do**
3: $S_j = \kappa_j \cdot P - (\tau - 1) \cdot R$ // store all precomputed points in memory
4: **end for**
5: $Q = S_{l-1} + \tau \cdot R$; // ADD $\kappa_{l-1} \cdot P - (\tau - 1) \cdot R$ and $\tau \cdot R$
6: **for** $(j = l - 2$ **down to** $0)$ **do**
7: $Q = \tau \cdot Q$ // use DBL and ADD operations since $\tau \in \mathbb{Z}[\mathrm{i}]$
8: $Q = Q + S_j$ // ADD Q and $\kappa_j \cdot P - (\tau - 1) \cdot R$ that is read from the memory
9: **end for**
10: **return** $Q - R$

Table 2. Primes of the form $p = a^2 + b^2$ suitable for elliptic curve cryptography (ECC) applications and the percentage of occurrences of offset reduction steps.

$\log_2(p)$	a	b	No Reduction	1 Reduction	2 Reductions	3 Reductions
188	$2^{94} - 150$	1	91.7%	4.4%	2.4%	1.5%
189	$2^{94} - 94$	$a - 1$	91.9%	5.0%	1.5%	1.6%
198	$2^{99} - 34$	1	91.6%	4.1%	2.5%	1.6%
199	$2^{99} - 37$	$a - 1$	91.6%	5.3%	1.5%	1.7%
208	$2^{104} - 120$	1	91.5%	4.0%	2.7%	1.8%
209	$2^{104} - 10$	$a - 1$	91.6%	4.2%	2.7%	1.6%

On the other hand, the different execution times of the reduction steps may cause concerns about side-channel vulnerabilities. However, the probability for additional reduction steps is relativity low. Moreover, the randomized initial point method also randomizes the occurrence of these additional steps.

Next, we demonstrate that determining the point multiplication using Gaussian integers reduces the computational complexity. The number of required point operations per iteration of the point multiplication is similar to the results in [22]. However, some additional computations are required to subtract the random point. As in [22], we estimate the computational complexity for the point multiplication in terms of multiplication equivalent operations M. Table 3 illustrates two examples for the complexity with different bases τ. The parameter r denotes the maximum key length in bits and the value l is the number of iterations per point multiplication. A larger value of $|\tau|^2$ reduces the number of iterations and speeds up the computation compared with a conventional binary PM, but requires more storage for the precomputed points.

Table 3. Complexity results for Algorithm 4 with different τ-adic expansions in comparison with ordinary integer expansions from [38] for a binary key of length $r = 163$.

Reference	$\lvert\tau\rvert^2$ or 2^w	l	M per PM incl. Precomputations
proposed	5	$0.430r$	2372
[38]	4	$0.506r$	3054
proposed	17	$0.245r$	1866
[38]	16	$0.2515r$	2763

The results are compared with the point multiplication according to randomized initial point but using ordinary integer expansions as proposed in [38] for comparable base values w. For instance, we can compare the complex base $\tau = 4 + \mathrm{i}$ with $|\tau|^2 = 17$ digits and the base $w = 4$ with $2^w = 16$ digits from [38]. For these values, the number M of multiplications is reduced by 32.5%. This reduction results mainly from the simpler calculation of $\tau \cdot Q$ for Gaussian integers due to the used endomorphism. In [21], a similar PM with τ-adic expansions over Gaussian integers was proposed. This algorithm additionally reduces the number of precomputed and stored points. However, this method considers only timing and simple power analysis attacks. It is more vulnerable to statistical attacks than the randomized initial point method.

To illustrate the efficiency of Gaussian integer arithmetic, we consider latencies for the point multiplication according to Algorithm 4 using both norms in Table 4. The table provides results for a Xilinx Virtex-7 field-programmable gate array (FPGA) based on the architecture from [22]. The hardware requirements are represented by the number of look-up tables (LUT), flip-flops (FF), slices, and digital signal processor (DSP) units, as well as by the RAM size. The maximum clock frequency is denoted by f_{clk}. These point multiplications are significantly faster than the results from [39,40]. For instance, the design in [40] considers key lengths up to $r = 256$ and the unprotected PM. This design is optimized for the use of DSP units which reduces the number of LUT. The number 1990 of LUT is similar to the proposed design with key length $r = 253$ using DSP units. It employs much more flip-flops (1786) and memory (234 kbit). An unprotected PM of length $r = 256$ requires 23.5ms whereas the proposed protected PM requires only 6.87ms for $r = 253$.

Note that the latency values for the reduction algorithms depend on the hardware architecture. The calculation of the Manhattan weight requires one addition which is performed in a single clock cycle with the architecture from [22]. For the absolute value, two additional multiplications are required which need four clock cycles each. Hence, the latency for calculating the Manhattan weight is only 11% of the time for the absolute value. The results in Table 4 illustrate the impact of this complexity reduction on the total latency of a point multiplication. Applying the Manhattan weight reduces the latency of a PM by 13% compared with the reduction algorithm from [29].

Table 4. Field-programmable gate array (FPGA) synthesis results for the point multiplication according to Algorithm 4 using the absolute value (abs.) and the Manhattan weight (Man).

τ	r	LUT	FF	RAM [kbit]	Slices	DSP	f_{clk} [MHz]	Protected PM Latency [ms]	
								abs.	Man.
$2 + \mathrm{i}$	189	1540	521	17.3	426	4	227	7.59	6.60
$4 + \mathrm{i}$	189	1540	521	19.1	426	4	227	5.68	4.93
$4 + \mathrm{i}$	253	1891	678	20.4	532	4	212	7.92	6.87

6. Conclusions

Gaussian integers are suitable for RSA [17,18] and ECC applications [22]. Furthermore, Gaussian integers have applications in coding theory [23,26]. The Montgomery multiplication for Gaussian integers was previously proposed in [17,18]. However, no reduction algorithm was advised.

The generalization of Montgomery reduction to Gaussian integers is not trivial, because complex numbers cannot be totally ordered. In this work, we have presented two algorithms for Montgomery reduction for Gaussian integers which use different norms. The first algorithm utilizes the absolute value to measure the size of a Gaussian integer. This algorithm can uniquely determine the correct representative. The second algorithm is based on the Manhattan weight of Gaussian integers, which reduces the computational complexity for the modulo reduction. However, two congruent solutions may occur. It is not possible to determine the correct candidate based on the Manhattan weight. The correct representative is the candidate with the smaller absolute value. Hence, the first algorithm is required to determine the final result, e.g., for the inverse mapping from the Montgomery domain to the original field representation.

It is shown in [21,22] that a key representation with a Gaussian integer expansion is beneficial to reduce the computational complexity and the memory requirements of elliptic curve point multiplication algorithms that are protected against side-channel attacks. However, only timing and simple power analysis attacks were considered. This PM is vulnerable to statistical attacks [38]. As an alternative, we have shown that Gaussian integer expansions can be used with the randomized initial point method. This method is protected against statistical attacks like DPA, RPA, and ZPA. The FPGA implementation of this protected PM with Gaussian integer expansions is significantly faster than the PM algorithms over ordinary prime fields reported in [39,40].

However, Gaussian integer fields can only be constructed for primes of the form $p \mod 4 = 1$, hence a generalization of this work to Eisenstein integers could be beneficial. Eisenstein integers are complex numbers of the form $x = a + b\omega$, where $\omega = \frac{1}{2} \cdot \left(1 + \sqrt{-3}\right)$, and Eisenstein integer fields can be constructed for primes of the form $p \mod 6 = 1$. An elliptic curve point multiplication using Eisenstein integers was considered in [49] showing similar properties as the point multiplication over Gaussian integers. Up to now no efficient modulo operation for Eisenstein integers is known. We believe that a generalization of the Montgomery modular multiplication to Eisenstein integers would be a promising direction for further research.

Author Contributions: The research for this article was exclusively undertaken by M.S. and J.F. Conceptualization and investigation, M.S. and J.F.; writing–review and editing, M.S. and J.F.; writing–original draft preparation, M.S.; supervision, project administration, and funding acquisition J.F.; All authors have read and agreed to the published version of the manuscript.

Funding: The German Federal Ministry of Economics and Technology (ZF4559701ED8) supported the research for this article.

Acknowledgments: The authors would like to thank Hyperstone GmbH, Konstanz for supporting the research for this article.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Montgomery, P.L. Modular multiplication without trial division. *Math. Comput.* **1985**, *44*, 519–521. [CrossRef]
2. Mahapatra, P.P.; Agrawal, S. RSA Cryptosystem with Modified Montgomery Modular Multiplier. In Proceedings of the IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, India, 14–16 December 2017; pp. 1–6. [CrossRef]
3. Parihar, A.; Nakhate, S. Fast Montgomery modular multiplier for Rivest-Shamir-Adleman cryptosystem. *IET Inf. Secur.* **2019**, *13*, 231–238. [CrossRef]

4. Koblitz, N. Elliptic Curve Cryptosystems. *Math. Comput.* **1987**, *48*, 203–209. [CrossRef]
5. Loi, K.C.C.; Ko, S. Scalable Elliptic Curve Cryptosystem FPGA Processor for NIST Prime Curves. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2015**, *23*, 2753–2756. [CrossRef]
6. Khan, Z.U.A.; Benaissa, M. Throughput/Area-efficient ECC Processor Using Montgomery Point Multiplication on FPGA. *IEEE Trans. Circuits Syst. II Express Briefs* **2015**, *62*, 1078–1082. [CrossRef]
7. Hossain, M.S.; Saeedi, E.; Kong, Y. High-Speed, Area-Efficient, FPGA-Based Elliptic Curve Cryptographic Processor over NIST Binary Fields. In Proceedings of the IEEE International Conference on Data Science and Data Intensive Systems, Sydney, NSW, Australia, 11–13 December 2015; pp. 175–181. [CrossRef]
8. Bosmans, J.; Roy, S.S.; Jarvinen, K.; Verbauwhede, I. A Tiny Coprocessor for Elliptic Curve Cryptography over the 256-bit NIST Prime Field. In Proceedings of the 29th International Conference on VLSI Design (VLSID), Kolkata, India, 4–8 January 2016; pp. 523–528. [CrossRef]
9. Ma, Y.; Zhang, Q.; Liu, Z.; Tu, C.; Lin, J. Low-Cost Hardware Implementation of Elliptic Curve Cryptography for General Prime Fields. In *Information and Communications Security*; Lecture Notes in Computer Science; Lam, K.Y.; Chi, C.H.; Qing, S., Eds.; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 292–306.
10. Mozhi, S.A.; Ramya, P. Efficient bit-parallel systolic multiplier over GF(2^m). In Proceedings of the International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), Chennai, India, 3–5 March 2016; pp. 4899–4902. [CrossRef]
11. Amiet, D.; Curiger, A.; Zbinden, P. Flexible FPGA-Based Architectures for Curve Point Multiplication over GF(p). In Proceedings of the Euromicro Conference on Digital System Design (DSD), Limassol, Cyprus, 31 August–2 September 2016; pp. 107–114. [CrossRef]
12. Khan, Z.U.A.; Benaissa, M. High-Speed and Low-Latency ECC Processor Implementation Over GF(2^m) on FPGA. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2017**, *25*, 165–176. [CrossRef]
13. Safieh, M.; Thiers, J.P.; Freudenberger, J. Area Efficient Coprocessor for the Elliptic Curve Point Multiplication. In Proceedings of the 12th International ITG Conference on Systems, Communications and Coding (SCC), Rostock, Germany, 11–14 February 2019; pp. 1–6.
14. Roy, D.B.; Mukhopadhyay, D. High-Speed Implementation of ECC Scalar Multiplication in GF(p) for Generic Montgomery Curves. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* **2019**, *27*, 1587–1600. [CrossRef]
15. Elkamchouchi, H.; Elshenawy, K.; Shaban, H. Extended RSA cryptosystem and digital signature schemes in the domain of Gaussian integers. In Proceedings of the 8th International Conference on Communication Systems (ICCS), Singapore, 28 November 2002; Volume 1, pp. 91–95. [CrossRef]
16. Koval, A.; Verkhovsky, B.S. Analysis of RSA over Gaussian Integers Algorithm. In Proceedings of the Fifth International Conference on Information Technology: New Generations (ITNG), Las Vegas, NV, USA, 7–9 April 2008; pp. 101–105. [CrossRef]
17. Koval, A. *Security Systems Based on Gaussian Integers: Analysis of Basic Operations and Time Complexity of Secret Transformations*; New Jersey Institute of Technology: Newark, NJ, USA, 2011.
18. Koval, A. Algorithm for Gaussian Integer Exponentiation. In *Information Technology: New Generations*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; pp. 1075–1085.
19. Bhargava, K.; Soni, V. A novice cryptosystem based on nth root of Gaussian integers. In Proceedings of the 2017 International Conference on Computer, Communications and Electronics (Comptelix), Jaipur, India, 1–2 July 2017; pp. 271–274. [CrossRef]
20. Awad, Y.; El-Kassar, A.N.; Kadri, T. Rabin Public-Key Cryptosystem in the Domain of Gaussian Integers. In Proceedings of the International Conference on Computer and Applications (ICCA), Beirut, Lebanon, 25–26 August 2018; pp. 336–340. [CrossRef]
21. Safieh, M.; Thiers, J.; Freudenberger, J. Side Channel Attack Resistance of the Elliptic Curve Point Multiplication using Gaussian Integers. In Proceedings of the Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 26–27 May 2020; pp. 231–236.
22. Safieh, M.; Thiers, J.; Freudenberger, J. A Compact Coprocessor for the Elliptic Curve Point Multiplication over Gaussian Integers. *Electronics* **2020**, *9*, 2050. [CrossRef]
23. Huber, K. Codes over Gaussian integers. *IEEE Trans. Inf. Theory* **1994**, *40*, 207–216. [CrossRef]
24. Martinez, C.; Beivide, R.; Gabidulin, E. Perfect Codes for Metrics Induced by Circulant Graphs. *IEEE Trans. Inf. Theory* **2007**, *53*, 3042–3052. [CrossRef]
25. Quilles, C.; Palazzo, R. Quasi-Perfect Geometrically Uniform Codes Derived from Graphs over Gaussian Integer Rings. In Proceedings of the IEEE International Symposium on Information Theory (ISIT), Austin, TX, USA, 13–18 June 2010.
26. Freudenberger, J.; Ghaboussi, F.; Shavgulidze, S. New Coding Techniques for Codes over Gaussian Integers. *IEEE Trans. Commun.* **2013**, *61*, 3114–3124. [CrossRef]
27. Freudenberger, J.; Ghaboussi, F.; Shavgulidze, S. Set Partitioning and Multilevel Coding for Codes Over Gaussian Integer Rings. In Proceedings of the 9th International ITG Conference on Systems, Communications and Coding (SCC), Munich, Germany, 21–24 January 2013; pp. 1–5.
28. Rohweder, D.; Freudenberger, J.; Shavgulidze, S. Low-Density Parity-Check Codes over Finite Gaussian Integer Fields. In Proceedings of the 2018 IEEE International Symposium on Information Theory (ISIT), Vail, CO, USA, 17–22 June 2018; pp. 481–485. [CrossRef]
29. Safieh, M.; Freudenberger, J. Montgomery Modular Arithmetic over Gaussian Integers. In Proceedings of the 24th International Information Technology Conference (IT), Zabljak, Montenegro, 18–22 February 2020; pp. 1–4.

30. Kocher, P. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 18–22 August 1996.
31. Kocher, P.; Jaffe, J.; Jun, B. Differential Power Analysis. In Proceedings of the Annual International Cryptology Conference, Santa Barbara, CA, USA, 15–19 August 1999.
32. Hankerson, D.; Menezes, A.J.; Vanstone, S. *Guide to Elliptic Curve Cryptography*; Springer: New York, NY, USA, 2003.
33. Goubin, L. A Refined Power-Analysis Attack on Elliptic Curve Cryptosystems. In *Public Key Cryptography—PKC 2003*; Desmedt, Y.G., Ed.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 199–211.
34. Akishita, T.; Takagi, T. Zero-Value Point Attacks on Elliptic Curve Cryptosystem. In *Information Security*; Boyd, C.; Mao, W., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; pp. 218–233.
35. Mulder, E.D.; Örs, S.B.; Preneel, B.; Verbauwhede, I. Differential power and electromagnetic attacks on a FPGA implementation of elliptic curve cryptosystems. *Comput. Electr. Eng.* **2007**, *33*, 367–382. [CrossRef]
36. Lerman, L.; Bontempi, G.; Markowitch, O. Side channel attack: An approach based on machine learning. In Proceedings of the Proc. 2nd International Workshop on Constructive Side-Channel Analysis and Secure Design, Darmstadt, Germany, 14 February 2011; pp. 29–41.
37. Maghrebi, H.; Portigliatti, T.; Prouff, E. Breaking cryptographic implementations using deep learning techniques. In Proceedings of the International Conference on Security, Privacy, and Applied Cryptography Engineering, Hyderabad, India, 14–18 December 2016; pp. 3–26.
38. Hedabou, M.; Pinel, P.; Bénéteau, L. Countermeasures for Preventing Comb Method Against SCA Attacks. In *Information Security Practice and Experience*; Deng, R.H., Bao, F., Pang, H., Zhou, J., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 85–96.
39. Salman, A.; Ferozpuri, A.; Homsirikamol, E.; Yalla, P.; Kaps, J.; Gaj, K. A scalable ECC processor implementation for high-speed and lightweight with side-channel countermeasures. In Proceedings of the International Conference on ReConFigurable Computing and FPGAs (ReConFig), Cancun, Mexico, 4–6 December 2017; pp. 1–8. [CrossRef]
40. Matutino, P.M.; Araújo, J.; Sousa, L.; Chaves, R. Pipelined FPGA coprocessor for elliptic curve cryptography based on residue number system. In Proceedings of the International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS), Pythagorion, Greece, 17–20 July 2017; pp. 261–268. [CrossRef]
41. Diekert, V.; Kufleitner, M.; Rosenberger, G.; Hertrampf, U. *Discrete Algebraic Methods: Arithmetic, Cryptography, Automata and Groups*; De Gruyter: Berlin, Germany, 2016.
42. Krisell, M. *Elliptic Curve Digital Signatures in RSA Hardware*; Scholar's Press: Riga, Latvia 2012.
43. Jarvinen, K.; Tommiska, M.; Skytta, J. A scalable architecture for elliptic curve point multiplication. In Proceedings of the IEEE International Conference on Field- Programmable Technology, Brisbane, NSW, Australia, 6–8 December 2004; pp. 303–306. [CrossRef]
44. Okeya, K.; Takagi, T.; Vuillaume, C. Efficient Representations on Koblitz Curves with Resistance to Side Channel Attacks. In *Information Security and Privacy*; Boyd, C., Gonza'lez Nieto, J.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2005; pp. 218–229.
45. Thériault, N. SPA Resistant Left-to-Right Integer Recodings. In *Selected Areas in Cryptography*; Preneel, B., Tavares, S., Eds.; Springer: Berlin/Heidelberg, Germany, 2006; pp. 345–358.
46. Tao, Z.; Mingyu, F.; Xiaoyu, Z. Secure and efficient elliptic curve cryptography resists side-channel attacks. *J. Syst. Eng. Electron.* **2009**, *20*, 660–665.
47. Liu, S.; Yao, H.; Wang, X.A. SPA Resistant Scalar Multiplication Based on Addition and Tripling Indistinguishable on Elliptic Curve Cryptosystem. In Proceedings of the 10th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), Krakow, Poland, 4–6 November 2015; pp. 785–790. [CrossRef]
48. Gallant, R.; Lambert, R.; Vanstone, S. Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms. In *Advances in Cryptology—CRYPTO 2001*; Springer: Berlin/Heidelberg, Germany, 2001; pp. 190–200.
49. Thiers, J.P.; Safieh, M.; Freudenberger, J. Side Channel Attack Resistance of the Elliptic Curve Point Multiplication using Eisenstein Integers. In Proceedings of the IEEE 10th International Conference on Consumer Electronics (ICCE), Berlin, Germany, 9–13 November 2020.