Journal of Computer Science 5 (9): 614-618, 2009
ISSN 1549-3636


# Accelerated Search for Gaussian Generator Based on Triple Prime Integers

[1]Boris S. Verkhovsky and [2]Md Shiblee Sadik
[1]Department of Computer Science, New Jersey Institute of Technology, USA
[2] Department of Computer Science, University of Oklahoma, Norman, Oklahoma, USA

**Abstract: Problem statement:** Modern cryptographic algorithms are based on complexity of two problems: Integer factorization of real integers and a Discrete Logarithm Problem (DLP). **Approach:** The latter problem is even more complicated in the domain of complex integers, where Public Key Cryptosystems (PKC) had an advantage over analogous encryption-decryption protocols in arithmetic of real integers modulo p: The former PKC have quadratic cycles of order O ($p^2$) while the latter PKC had linear cycles of order O(p). **Results:** An accelerated non-deterministic search algorithm for a primitive root (generator) in a domain of complex integers modulo triple prime p was provided in this study. It showed the properties of triple primes, the frequencies of their occurrence on a specified interval and analyzed the efficiency of the proposed algorithm. **Conclusion:** Numerous computer experiments and their analysis indicated that three trials were sufficient on average to find a Gaussian generator.

**Key words:** Communication network security, crypto-immunity, primitive root, public-key cryptography

## INTRODUCTION

A Discrete Logarithm problem, {DLP, for short}, is defined as follows: For real integers g>1, p and h>0 to find an integer x such that satisfies the equation:

$$g^x \bmod p = h \quad (1)$$

This is a computationally formidable problem[7,8,10] especially if the integer g is a primitive root (generator)[1,2]. The complexity of the DLP is the basis for secret-key establishment in modern cryptography[3,6,11,12]. An RSA cryptographic algorithm in the domain of complex integers is described in[4].

The DLP in the domain of complex integers (called Gaussian integers) is an extension of the problem (1): To find a real integer x such that holds

$$G^x \bmod p = h \quad (2)$$

Where:
G and H = Gaussian integers
p = A prime

As in (1), a solution of Eq. 2 is computationally intense especially if the Gaussian integer G is a primitive root or generator as described in Definition 2 below.

**Definition 1:** If X is a Gaussian integer and m is the smallest positive integer, for which the following relation holds:

$$X^m \bmod p = (1,0) = 1 \quad (3)$$

then m is defined as the order of X:

$$\mathrm{ord}(X) = m \quad (4)$$

**Definition 2:** If the order of Gaussian integer G equals:

$$m = p^2-1 \quad (5)$$

then G is called a Gaussian primitive root or generator, (GG, for short).

There are currently no known deterministic algorithms that compute a GG. However, if p is appropriately selected, then the search for a GG can be substantially simplified.

**Definition 3:** A prime p is called a Triple Prime (TP) if both integers:

$$q := (p+1)/4 \text{ and } r := (p-1)/6 \quad (6)$$

are also primes.

**Corresponding Author:** Boris S. Verkhovsky, Department of Computer Science, New Jersey Institute of Technology, USA

Table 1: TPs and corresponding primes q and r

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| p | 19 | 43 | 67 | 283 | 787 | 907 | 1,867 | 9,643 | 99,907 | 991,987 | 1,998,643 |
| q | 5 | 11 | 17 | 71 | 197 | 227 | 467 | 2,411 | 24,977 | 247,997 | 499,661 |
| r | 3 | 7 | 11 | 47 | 131 | 151 | 311 | 1,697 | 16,651 | 165,331 | 333,107 |

Table 2: Frequencies of TPs on intervals [104m, 104 (m+1)]

| m | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 12 | 14 | 16 | 18 | 20 | 25 | 30 | 40 | 50 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| No. of TP | 8 | 6 | 8 | 3 | 5 | 7 | 4 | 7 | 6 | 7 | 3 | 7 | 5 | 3 | 1 | 3 | 3 | 5 | 2 |

**Remark:** If both p and q are primes, then neither (p-1)/2 nor (p-1)/3 are primes; {for details Lemma 2}.

The Table 1 provides examples of several triple primes.

## MATERIALS AND METHODS

**Triple primes and their properties:** The algorithm searching for Gaussian generator (9-12) is based on the following properties of triple primes.

**Lemma 1:** If a prime $p \geq 43$, then for every TP the following condition holds:

$$p \bmod 60 = 7 \text{ or } 43 \quad (7)$$

If (7) does not hold, then either q or r are not integers or not primes. For applications, it is necessary to know the occurrence of the TP (their density) for large p. The Table 2 provides such information.

**Remark:** CPU times T required to find a D digit-long TP are provided in milliseconds (ms).

**Lemma 2:** If n is an odd integer and:

- $n \geq 5$ is not divisible by 3, then 24 divides $n^2-1$
- $n \geq 23$ and $(n+1)/4$ is a prime, then $(n-1)/6$ is an integer

**Proof:** n-1 and n+1 are two consecutive even integers, hence one of them is divisible by 4 and their product is divisible by 8. In addition, either n-1 or n+1 is divisible by 3. Therefore, if n is a Blum prime, then:

$$(n^2-1)/24 = [(n+1)/4][(n-1)/6] \quad (8)$$

where both factors in (8) are integers.

Suppose there is $n_1 \geq 23$, for which $q_1 = (n_1+1)/4$ is a prime, but $r_1 = (n_1-1)/6$ is not an integer. Since $n_1-1$ is even, hence 3 does not divide $n_1-1$. Therefore three divides the prime $q_1$. This contradiction proves Lemma 2. More details are provided in Fig. 1.

Table 3: Length in decimal digits (D), TPs and average time (T) to compute it

| D | TP p | T (ms) |
|---|---|---|
| 5 | 11,443 | 48.36 |
| 6 | 100,483 | 217.75 |
| 7 | 1,006,267 | 469.44 |
| 8 | 10,009,267 | 1087.07 |
| 9 | 100,019,923 | 7829.67 |
| 10 | 1,000,013,107 | 9205.34 |

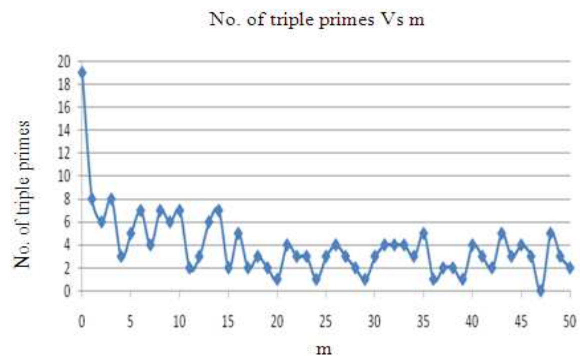


Fig. 1: Number of TPs on intervals $[10^4 m, 10^4 (m+1)]$

## RESULTS AND DISCUSSION

**Algorithm searching for generator:**

**Step1**: Select a triple prime $p \geq 19$:

$$\text{compute } q := (p+1)/4$$

and

$$r := (p-1)/6 \quad (9)$$

**Step 2:** Select integers such that hold:

$$a \neq b;\ 1 \leq a,b \leq p-1;\ a+b \neq p$$

and

$$(a^2 + b^2) \bmod p \neq 1$$

**Step 3:** For k = {2, 3, q} compute:

$$(c,d) := (a,b)^k \bmod p \quad (10)$$

**Step 4:** If $k \neq q$ and $\{c = 0$ or $d = 0$ or $|c| = |d|\}$, then goto Step 2; {(a, b) is not a generator}.

**Step 5:** If $k = q$ and $\{c = 0$ or $d = 0\}$, then goto Step 2; {(a, b) is not a generator}.

**Step 6:** Compute:

$$e := -4c^4 \bmod p \quad (11)$$

**Step 7:** If $e^f \bmod p = \pm 1$ (12)

then go to Step 2; {(a, b) is not a generator};
else output G = (a, b); {(a, b) is a generator}.

**Analysis of basic results:** Numerous computer experiments demonstrated that the algorithm (9-12) finds a Gaussian generator after three trials of (a, b) on average.

**Algorithm validation:**

**Lemma 3:** Suppose (a, b) is a Gaussian integer (Gaussian, for short), p is a TP and:

$$(c,d) := (a,b)^q \bmod p \quad (13)$$

If a component in (c, d) equals zero, then (a, b) is not a GG.

**Proof:** If $(a,b)^q \bmod p = (c,0)$, then:

$$(a,b)^{(p\wedge 2-1)/4} \bmod p = c^{p-1 \bmod p} = 1 \quad (14)$$

Hence, the order of (a, b) is smaller than $p^2 -1$:

$$\text{therefore, } ord(a,b) \le (p^2 - 1)/4 \quad (15)$$

If $(a,b)^q \bmod p=(0,d)$, then:

$$(a,b)^{2q} = (0,d)^2 = (-d^2) \pmod p \quad (16)$$

Therefore:

$$(a,b)^{(p\wedge 2-1)/2} \bmod p == (-d^2)^{p-1} \bmod p = 1 \quad (17)$$

Thus, the order of (a, b) is smaller than $p^2 -1$; as a result:

$$ord(a,b) \le (p^2 -1)/2 \quad (18)$$

Hence in both cases, (14) and (16), (a, b) is not a GG. Q.E.D.

**Theorem 4:** Suppose (a, b) is a Gaussian integer, p is a TP:

$$(a,b)^q \equiv (c,p\pm c) \pmod p \quad (19)$$

Let:

$$e := (-4c^4) \bmod p = -(\pm 2c^2)^2 \bmod p \quad (20)$$

If:

$$e^{(p-1)/6} \bmod p = \pm 1 \quad (21)$$

then (a, b) is not a GG.

**Proof:** First of all:

$$(a,b)^{4q} = (c,\pm c)^4 = -(\pm 2c^2)^2 \pmod p = e \quad (22)$$

Let us define:

$$v_{k :=} e^{(p-1)/k} \bmod p \quad (23)$$

then $v_1 = 1$ and $v_1 = \pm 1$. Therefore (21) implies that:

$$(-4c^d)^r = (c,\pm c)^{dr} = (a,b)^{4qr}$$
$$= (a,b)^{(p\wedge 2-1)/6} = \pm 1 \pmod p \quad (24)$$

Hence:

$$ord(a,b) \le (p^2 - 1)/3 \quad (25)$$

i.e., (a, b) is not a generator.

Suppose that condition (21) does not hold, i.e., let:

$$v_6 \ne \pm 1 \quad (26)$$

Let us analyze two sub-cases:

$$v_3 = \pm 1 \text{ or } v_2 = \pm 1 \quad (27)$$

**Case A:** $v_6^2 \bmod p = v_3 = -1$ is impossible, otherwise $v_1 = -1$ and that contradicts Fermat's theorem. On the other hand $v_6^2 \bmod p = v_3 = 1$ implies that $v_6 = \pm 1$, which contradicts the assumption (26).

**Case B:** Let us demonstrate that:

$$v_2 = 1 \quad (28)$$

is also impossible. Indeed, consider:

$$v_2 = e^{(p-1)/2} = (-1)^{(p-1)/2} [(2c^2)^2]^{(p-1)/2}$$
$$= (-1)^{(p-1)/2} \bmod p = 1 \quad (29)$$

Therefore, this implies that (a, b) is a GG. Q.E.D.

Suppose that N(p) is the number of Gaussian generators.

Table 4: Required number of Real Integer Multiplications (RIM)

| Operations | $(c,d):= (a,b)^q \bmod p$ | $e^r \bmod p$ |
|---|---|---|
| Squarings | log q complex squarings require 2logq RIM | Requires logr RIM |
| Multiplications | On average (log q)/2 complex multiplications require 3 (logq)/2 RIM | Requires on average (logr)/2 RIM |
| Overall | 3.5 log q RIM | 1.5 log r RIM |

**Theorem 5:** For large p:

$$N(p) \to (p^2 - 1)/3 \quad (30)$$

**Proof:** If G is a GG, then $G^z \bmod p$ is also a GG if z is co-prime with $p^2$-1.

Euler's totient function shows how many integers z satisfy this requirement:

$$\varphi(p^2 -1) = \varphi(24qr) = \varphi(24)\varphi(q-1)\varphi(r-1)$$
$$=(p-3)(p-7)/3=[ (p-5)^2 -4] /3 \quad (31)$$

Therefore, $N(p) = [(p-5)^2 -4]/3 = (p^2 -1)/3 - o(p)$.

**Corollary:** For a large p, if a Gaussian (a, b) is selected randomly, then the probability of it being a primitive root (generator) is close to 1/3. After t trials, (a, b) is likely to be a GG with probability $1-(2/3)^t$.

If t = 3, then the probability is $1-(2/3)^t = 0.703703\ldots$ that it is a GG.

Numerous computer experiments demonstrate that, on average, the algorithm (9-12) finds a GG after a mere three trials, with a standard deviation of 2.44.

**Complexity analysis of algorithm:** The following Table 4 facilitates the analysis.

Indeed, from[5] $(x,y)^2 = ((x+y)(x-y),2xy)(\bmod p)$ (32)

and $(u,w)(x,y)=(ux-wy,(u+w)(x+y)-ux-wy) \bmod p$ (33)

therefore, the squaring requires two RIM and the product of two complex integers requires three RIM (32). Thus, the total number of required RIM is equal to $3.5\times\log(p+1)/4+1.5\times\log(p-1)/6<5\log(p+1)/4 = \Theta(\log p)$. Further reduction of complexity can be achieved via application of Toom's algorithm for computation of multi-digit long integers[9].

**Illustrative example:** The following numeric example demonstrates the most important features of the search algorithm and quadratic order of the GG.

Suppose a triple prime p = 11443.

**Step 1:** p = 11443, q = 2861 and r = 1907; {all three integers are primes}.

Table 5: TPs, Gaussian generators and their orders

| TP p | GG | ord (GG) | T (µs) |
|---|---|---|---|
| 11,443 | (3801, 7240) | 130,942,248 | 35.21 |
| 11,587 | (9925, 3113) | 134,258,568 | 17.07 |
| 12,163 | (4761, 10711) | 147,938,568 | 8.78 |
| 14,107 | (3598, 1763) | 199,007,448 | 13.31 |
| 15,187 | (10173, 12371) | 230,644,968 | 6.32 |
| 99,907 | (12209, 93518) | 9,981,408,648 | 7.34 |
| 100,987 | (58921, 38436) | 10,198,374,168 | 61.50 |
| 103,387 | (21044, 47275) | 10,688,871,768 | 6.17 |
| 104,947 | (10289, 17250) | 11,013,872,808 | 5.70 |
| 991,987 | (473412, 476250) | 984,038,208,168 | 61.26 |
| 1,030,867 | (665172, 814725) | 1,062,686,771,688 | 5.89 |
| 1,038,523 | (322164, 825494) | 1,078,530,021,528 | 6.02 |
| 1,998,643 | (372339, 931799) | 3,994,573,841,448 | 3.26 |
| 2,004,763 | (574467, 342161) | 4,019,074,686,168 | 12.83 |

**Step 2:** {First trial}:

Select randomly (a, b) = (2446, 1893);
Step 3: Repeat for k = {2, 3, q}
Step 4 {iteration for k = 2}:
Compute (c, d) = (7880, 3169);
if c = 0 or d = 0 or |c| = |d|,
then goto the next trial;
Step 4 {iteration for k = 3}:
Compute (c, d) = (1683, 11074);
if c = 0 or d = 0 or |c| = |d|,
then goto the next trial;
Step4 {iteration for k = q}:
Compute (c, d) = (0, 11074);
if c = 0 or d = 0, then goto the next trial;
Step 2: {Second trial}:
Select randomly (a, b) = (3801, 7240);
Step 3: Repeat for k = {2, 3, q}
Step4 {iteration for k = 2}:
Compute (c, d) = (9318, 9093);
if c = 0 or d = 0 or |c| = |d|,
then goto the next trial;
Step4: {iteration for k = 3}:
Compute (c, d) = (11335, 10468);
if c = 0 or d = 0 or |c| = |d|,
then goto the next trial;
Step4: {iteration for k = q}:
Compute (c, d) = (9571, 9571);
if c = 0 or d = 0, then goto the next trial;
Step 5: Compute e = 938;
Step 6: f = 2932; {see (11) and (12)};
Step 7: If $f \neq \pm1$, then output (a, b) = (3801, 7240) is a Gaussian generator.

It is easy to verify that the order of the GG equals: ord (3801, 7240) = $p^2$-1 = $11443^2$-1= 130,942,248; (Table 5).

**Remark:** CPU times T required to compute a Gaussian generator are provided in micro-seconds (µs).

All computations were performed on a PC with the following specifications: Intel Pentium dual-core processor; 2.16 GHz, 1 MB l2 cache and 2GB DDR2Main Memory.

Table 5 shows that if a GG=(574467, 342161) modulo triple prime p=2,004,763, then ord(GG) = 4,019,074,686,168.

## CONCLUSION

In various public-key cryptographic protocols users select a large prime and a corresponding generator g that are computationally-intense problems. Selection of a triple prime p is a computationally challenging task. Fortunately, the triple prime p must be selected only on a system-design level. After the triple prime p of a specified size is computed, the system designer can periodically change Gaussian generators. This policy provides additional cyber-immunity to cryptographic protocols.

## ACKNOWLEDGEMENT

The first author expresses his appreciation to Md. S. Sadik for his dedicated programming support and assistance in extensive computer experiments and gratitude to P. Fay for his valuable comments that improved the style of this study.

## REFERENCES

1. Boneh, D., 1998. The decision diffie-hellman problem. Lecture Notes Comput. Sci., 1423: 48-63.
2. Diffie, W. and M. Hellman, 1976. New directions in cryptography. IEEE. Trans. Inform. Theor., 22: 644-654. http://www.citeulike.org/user/junin/article/504050
3. ElGamal, T., 1985. A Public-key cryptosystem and a digital signature scheme based on discrete logarithms. IEEE Trans. Inform. Theory, 31: 469-472. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=1057074&isnumber=22749
4. Elkamchouchi, H., K. Elshenawy and H. Shaban, 2002. Extended RCA cryptosystem and digital signature schemes in the domain of gaussian integers. Proceeding of the 8th International Conference on Communication Systems, Nov. 25-28, IEEE Xplore Press, USA., pp: 91-95. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1182444
5. Karatsuba, A. and Y. Ofman, 1962. Multiplication of many-digital numbers by automatic computers. Doklady Akad. Nauk SSSR, 145: 293-294.
6. Katz, J. and Y. Lindell, Introduction to Modern Cryptography, Chapman and Hall CRC, London, New York, 2008. ISBN: 1584885513, pp: 534.
7. Pollard, J., 1978. Monte carlo methods for index computation. Math. Comput., 32: 918-924. http://www.jstor.org/stable/2006496
8. Shank, D., 1971. Class number, a theory of factorization and genera. Proc. Symp. Pure Math., 20: 415-440.
9. Toom, A., 1963. The complexity of a scheme of functional elements realizing the multiplication of integers. Doklady Akad. Nauk SSSR., 150: 496-498. http://www.de.ufpe.br/~toom/articles/engmat/MULT-E.PDF
10. Verkhovsky, B., 2008. Generalized baby-step giant-step algorithm for discrete logarithm problem. Adv. Decis. Technol. Intell. Inform. Syst., IX: 88-90.
11. Verkhovsky, B., 2009. Accelerated cybersecure communication based on reduced encryption/ decryption and information assurance protocols. J. Telecomm. Manage., 2 (to appear).
12. Verkhovsky, B., 2008. Information assurance protocols: efficiency analysis and implementation for secure communication. J. Inform. Assurance Secur., 3: 263-269.