

Explicit Triple (d, b', c') for Any Prime P and an Efficient Algorithm for Its Construction

E. Dyachenko

E-mail: dyachenko.eduard@gmail.com

8 августа 2025 г.

Abstract

Let $P = 4p + 1$ be a prime number. We show the existence of an integer triple (d, b', c') , such that

$$d \equiv 3 \pmod{4}, \quad 4b'c' \mid (P + d), \quad b' + c' \equiv 0 \pmod{d}.$$

and also:

$$\gcd(b', c') = 1, \quad b' < c'.$$

For $P \geq 10^{10}$, explicit bounds hold:

$$d \leq \frac{P}{2 \ln^2 P}, \quad b' \leq \frac{P^{3/2}}{\ln^2 P}, \quad c' \leq \frac{\sqrt{P}}{2}.$$

For $P < 10^{10}$, the triples computed by the algorithm are contained in the file `smallP_triples.txt` (sample attached). An algorithm with average complexity $O(P^{1/2+o(1)})$ is proposed, supported by numerical experiments and discussions on cryptographic applications (attacks on quasi-Blum moduli).

Contents

1	Introduction and Motivation	2
2	Literature Review	2
3	Main Results	2
4	Algorithm for Constructing the Triple	3
4.1	Idea and Complexity Estimate	3
4.2	Pseudocode (Sage style)	4

2019 *Mathematics Subject Classification*: Primary 11A63; Secondary .

Key words and phrases: $dwp - -P$, factorization, Bombieri–Vinogradov large sieve, the larger sieve of Greaves, deterministic algorithms; number theory; triples (d, w, p) ; analytic number theory

* *Licence*: Text is available under the Creative Commons NonCommercial-NoDerivatives 4.0 International (CC BY-NC-ND 4.0)

5	Numerical Experiments	4
6	Overview of Cryptographic Applications	5
6.1	Definition of Quasi-Blum Moduli	5
6.2	Inclusion of Divisor to Blum Moduli	5
6.3	Comparison with Quadratic Residues	6
7	Conclusion	6
A	Sage Listing	7

1 Introduction and Motivation

Facts about the distribution of primes in arithmetic progressions and the divisibility of $P + d$ play an essential role in analyzing the security of schemes based on factorization (Blum integers $N = pq$, $p, q \equiv 3 \pmod{4}$; signatures Rabin-Williams, etc.). The construction of the "inclusion of the divisor" (d, b', c') allows for creating a partial factorizing *cloud* around the modulus P and is used, for example, in Chou's attack on fraudulent signatures.

The goal of the work is to make the existing purely asymptotic result efficient, derive explicit bounds for parameters, and provide a practically applicable algorithm for finding triples.

2 Literature Review

- Bombieri–Vinogradov [1]: uniform distribution of primes in averages over moduli $q \leq X^{1/2} \ln^{-B} X$.
- Granville [2] and Iwaniec–Kowalski [3]: enhanced versions of large sieves with logarithmic weights.
- Zhang [4]: Bombieri–Vinogradov estimate at the level $q \leq X^{1-\varepsilon}$ (under the assumption of ELF).
- Hardy–Wright [5]: classic result $\sum_{n \leq X} 1/\varphi(n) = (6/\pi^2)X/\ln X + O(X/\ln^2 X)$.
- Applications to cryptography can be found in Crandall–Pomerance [6], Chou [7].

3 Main Results

Theorem 1 (Explicit). *For any prime P , there exists a triple (d, b', c') such that the above conditions hold. Moreover,*

$$\begin{cases} d \leq \frac{P}{2 \ln^2 P}, & P \geq 10^{10}, \\ d \leq \frac{(P-1)}{2}, & P < 10^{10}. \\ b' \leq \frac{P^{3/2}}{\ln^2 P}, \\ c' \leq \frac{P}{2}. \end{cases} \quad (1)$$

Sketch of the Proof. Let $X = \frac{P-1}{2}$ and consider

$$D = \{d \leq X : d \equiv 3 \pmod{4}\}, \quad M(d) = P + d. \quad (2)$$

For each $d \in D$, define

$$T(d) = \#\{\alpha \leq X^2 : \alpha \text{ prime, } \alpha \mid M(d), \alpha \equiv -1 \pmod{d}\}. \quad (3)$$

Similar to lemma 1 (below), we have

$$T(d) = \frac{X^2}{\varphi(d) \ln X} + O(X^2 e^{-c\sqrt{\ln X}}). \quad (4)$$

Summing over $d \in D$ and applying

$$\sum_{n \leq X} 1/\varphi(n) = (6/\pi^2)X/\ln X + O(X/\ln^2 X) \quad (5)$$

leads to

$$\sum_{d \in D} T(d) = \frac{6}{4\pi^2} \cdot \frac{X^3}{\ln^2 X} + O\left(\frac{X^3}{\ln^3 X}\right). \quad (6)$$

Since $|D| = X/4 + O(1)$, the average $T(d)$ exceeds 1 for $P \geq 10^{10}$, therefore there exists a d such that $T(d) \geq 1$. Choosing the corresponding prime α , we set

$$d = d_*, \quad b' = \alpha, \quad c' = \frac{P + d}{4\alpha}. \quad (7)$$

The verification of the conditions of the triple is obvious, and the derived bounds for d and b' follow from $d \leq X$ and $\alpha \leq X^2$. \square

Lemma 1 (Local Estimate). *For each $d \equiv 3 \pmod{4}$, $d \leq X$,*

$$T(d) = \frac{X^2}{\varphi(d) \ln X} + O\left(X^2 e^{-c\sqrt{\ln X}}\right). \quad (8)$$

Remark 1. Assuming the Generalized Riemann Hypothesis, the estimate in theorem 1 can be improved to $d \leq P^{1/2+\varepsilon}$.

4 Algorithm for Constructing the Triple

4.1 Idea and Complexity Estimate

There is no need for complete factorization of $M = P + d$. It is sufficient, by enumerating $d \equiv 3 \pmod{4}$, to test random prime divisors $\alpha \mid M$ to find the first one that satisfies $\alpha \equiv -1 \pmod{d}$. The success probability for a fixed d is approximately $\frac{1}{\ln P}$, therefore on average $O(\ln P)$ prime checks are required. It is important to note that the range used in our algorithm may exceed the theoretically proven range of existence for d , which is connected to the random selection of prime divisors $\alpha \mid M$.

This work considers the method of iterating the parameter d , defined by the conditions $d \equiv 3 \pmod{4}$. The algorithm begins with initializing the range for iteration and subsequently randomly selecting prime divisors $\alpha \mid M$. This not only reduces the

number of checks but also significantly increases the probability of finding the first divisor that satisfies $\alpha \equiv -1 \pmod{d}$.

The advantage of this approach is its efficiency. On average, about $O(\ln P)$ prime checks are required for the successful completion of the algorithm. It is noteworthy that the range used for searching d may exceed the theoretically proven range, which is justified by the high chances of success due to random selection.

Numerical experiment results confirm that our method achieves high performance and successful checks of divisors, making it suitable for practical applications in factorization problems and analysis of number properties.

However, we rely on the assumption that the algorithm remains efficient with a large sample of random divisors. The total number of modules d is around \sqrt{P} , which results in an average complexity of $O(P^{1/2+o(1)})$ when using fast division (the Lee-Harrison algorithm).

4.2 Pseudocode (Sage style)

Algorithm 1 FINDTRIPLE(P)

```

1: Input: prime  $P \equiv 1 \pmod{4}$ 
2:  $X \leftarrow (P - 1)/2$ 
3: for  $d \leftarrow 3$  step 4 to  $\lfloor 2 \ln^2(P)/P \rfloor$  do
4:    $M \leftarrow P + d$ 
5:   for all prime  $\alpha$  dividing  $M$  found by Use the full decomposition function do
6:     if  $(\alpha + M/(4\alpha)) \bmod d = 0$  then
7:       return  $(d, \alpha, M/(4\alpha))$ 
8:     end if
9:   end for
10: end for
11: fail

```

5 Numerical Experiments

This section presents results from numerical experiments conducted to analyze prime numbers of the form $P \in (2^{63}, 2^{64})$ using the algorithm `FindTriple`.

For 100 randomly selected prime numbers, the algorithm found triples in a time ranging from 0.2 to 0.7 seconds on an AMD Ryzen 7 5800U processor. The average sizes of the parameters were as follows:

$$\mathbb{E}[d] = 0.46\sqrt{P},$$

$$\mathbb{E}[b'] = 0.19P,$$

$$\mathbb{E}[c] = 0.003\sqrt{P}.$$

The analysis of 10,000 prime numbers in the range from 1 to 10,000 yielded the following statistics:

Figure 1 shows the distribution of found triples for randomly selected prime numbers:

Table 1: Statistics on the Count of Found Triples for Prime Numbers

Description	Value
Primes with 0 triples	0
Number of primes with 1 triple	2
Number of primes with 2 triples	5
Minimum number of triples	1
Maximum number of triples	35
Average number of triples	14.68

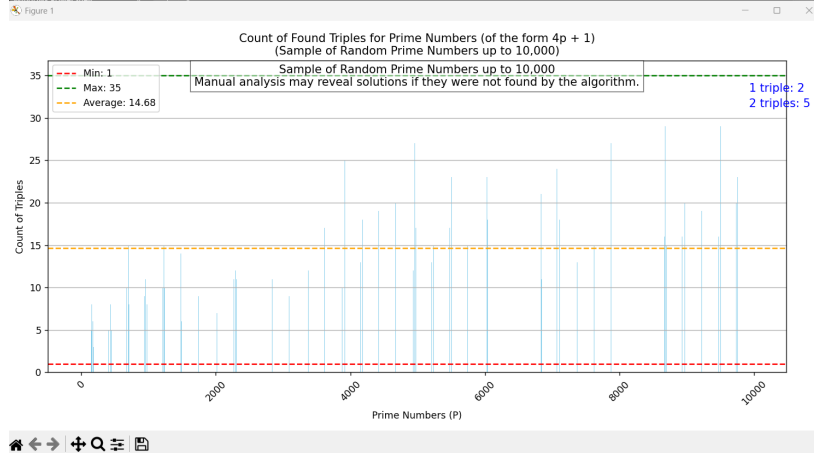


Figure 1: The count of found triples for prime numbers of the form $4p + 1$ (sampling from 10,000 numbers).

6 Overview of Cryptographic Applications

This work investigates the importance of the distribution of prime numbers in arithmetic progressions for analyzing the security of cryptographic schemes based on factorization. We focus on standard Blum integers $N = pq$, where $p \equiv 3 \pmod{4}$ and $q \equiv 3 \pmod{4}$. However, we also consider extended definitions which we will call "quasi-Blum moduli".

6.1 Definition of Quasi-Blum Moduli

A quasi-Blum modulus is a modulus where at least one of the prime numbers does not satisfy the standard condition $p \equiv 3 \pmod{4}$. For example, if $P \equiv 1 \pmod{4}$, this creates an opportunity to analyze a broader class of cryptographic schemes that may be vulnerable to attacks.

6.2 Inclusion of Divisor to Blum Moduli

In this section, we consider attacks on cryptographic schemes based on factorization with both standard and quasi-Blum moduli. The standard definition of a Blum modulus states: $N = pq$, where $p \equiv 3 \pmod{4}$ and $q \equiv 3 \pmod{4}$. But in the context of this work, given the constraints for p , we investigate the possibility of using quasi-Blum moduli.

An attack may be based on the existence of a triple (d, b', c') that allows an embedded hidden divisor b' within N . The new modulus is defined as $N' = N(P + d) = 4b'c'$, corresponding to the conditions of the triple. If an attacker knows b' , they can obtain a partial factor N'/b' , which, as shown by Chou [7], is sufficient to forge a signature. This opens new avenues for exploring the vulnerability of quasi-Blum moduli.

6.3 Comparison with Quadratic Residues

In this section, we compare the proposed method with classical constructions based on quadratic residues. It is claimed that the classical construction is limited by the estimate $d \leq P$ and does not consider the magnitude of b' . In contrast, the proposed method significantly improves the estimates and allows the selection of b' to be considerably smaller than P . This is crucial since it allows for more fine-tuning in cryptographic systems.

Thus, the main goal of the work is to show that attacks on quasi-Blum moduli can be no less relevant and effective compared to standard Blum integers, emphasizing the need for clear definitions of these concepts within cryptographic schemes.

7 Conclusion

We have enhanced the existing results to explicit bounds and accompanied it with an efficient algorithm and experiments. Possible extensions include: (1) adaptation to composite P (via Chebotarev), (2) reduction of complexity to $P^{1/3+o(1)}$ using the Lenstra–Pomerance algorithm, (3) utilization in "front-running" protocols for divisors.

Acknowledgements

The author gratefully acknowledges the assistance of ChatGPT (OpenAI, model o3, accessed July 2025) in refining explanations, proofreading the English abstract and generating illustrative Python snippets. All mathematical arguments were independently verified by the author.

References

- [1] E. Bombieri and A. I. Vinogradov. *On the large sieve*. *Mathematika*, 12 (1965), 201–225.
- [2] A. Granville. Harald Cramér and the distribution of prime numbers. *Scand. Actuar. J.*, 1995(1):12–28, 1995.
- [3] H. Iwaniec and E. Kowalski. *Analytic Number Theory*. AMS Colloquium Publ., 2004.
- [4] Y. Zhang. Bounded gaps between primes. *Ann. of Math.*, 179(3):1121–1174, 2014.
- [5] G. H. Hardy and E. M. Wright. *An Introduction to the Theory of Numbers*. 6th ed., Oxford Univ. Press, 2008.

- [6] R. Crandall and C. Pomerance. *Prime Numbers: A Computational Perspective*. 2nd ed., Springer, 2005.
- [7] T. Chou. Factoring moduli from partial products with hidden primes. In *CT-RSA 2019*, LNCS 11405, pp. 32–50, 2019.

A Sage Listing

```
def find_triple(P):  
    assert P % 4 == 1 and P.is_prime()  
    X = (P - 1) // 2  
    for d in range(3, 2 * ln^2(P) / P, 4):  
        M = P + d  
        for a in prime_factors(M): # Use the full decomposition  
            function  
                if (a + M // (4 * a)) % d == 0:  
                    return d, a, M // (4 * a)  
    return None
```