# THE DISTRIBUTION OF SOLUTIONS TO $XY = N$ (MOD $A$) WITH AN APPLICATION TO FACTORING INTEGERS

**Michael O. Rubinstein** [1]
*Pure Mathematics, University of Waterloo, Waterloo, Ontario, Canada*
mrubinst@uwaterloo.ca

*Received: 3/6/09, Revised: 9/21/12, Accepted: 3/6/13, Published: 3/15/13*

**Abstract**

We consider the uniform distribution of solutions $(x, y)$ to $xy = N \mod a$, and obtain a bound on the second moment of the number of solutions in squares of length approximately $a^{1/2}$. We use this to study a new factoring algorithm that factors $N = UV$ provably in $O(N^{1/3+\epsilon})$ time, and discuss the potential for improving the runtime to sub-exponential.

## 1. Introduction

Let $\gcd(a, N) = 1$. A classic application of Kloosterman sums shows that the points $(x, y) \mod a$ satisfying $xy = N \mod a$ become uniformly distributed in the square of side length $a$ as $a \to \infty$. In this paper we investigate an application of this fact to the problem of factoring integers. We give a new method to factor the integer $N$ which beats trial division, and prove that it runs in time $O(N^{1/3+\epsilon})$. Unlike many existing factoring algorithms that attempt to factor $N$ by considering various congruences modulo $N$, the method presented here revisits trial division and exploits information available on a failed trial division, namely it considers the remainder of $N$ when divided by an integer $a$, and also by its neighbor, $a - 1$.

While the complexity of our method is not exciting, considering the existence of several probabilistic sub-exponential factoring algorithms, the runtime here is provable. For comparison, the best known *provable* factoring algorithm, Pollard-Strassen, runs in time $O(N^{1/4+\epsilon})$. Shank's class group method runs in time $O(N^{1/5+\epsilon})$ assuming the GRH. It has a comparable runtime to Lehman's method [7], but the method is different. Our algorithm is described in Section 2.

Furthermore, proving this runtime requires understanding the finer distribution of solutions to $xy = N \mod a$, and our results in this regards are interesting in their own right. We discuss the problem on uniform distribution in Sections 4 and 5.

Finally, all existing sub-exponential factoring algorithms have grown out of much weaker exponential algorithms, and we hope that the factoring ideas presented here

[1] Support for work on this paper was provided by an NSERC Discovery Grant

will be improved. In Section 3 we discuss some needed improvements to achieve a better runtime.

We have not implemented the algorithms described in this paper. The purpose of this paper is to present a new approach to factoring integers and analyze its runtime.

## 2. Algorithm – Hide and Seek

Let $N$ be a positive integer that we wish to factor. Say $N = UV$ where $U$ and $V$ are positive integers, not necessarily prime, with $1 < U \leq V$. For simplicity, assume $V < 2U$, so that $V < (2N)^{1/2}$. The general case, without this restriction, will be handled at the end of this section.

The idea behind the algorithm is to perform trial division of $N$ by a couple of integers, and to use information about the remainder to determine the factors $U$ and $V$.

Let $a$ be a positive integer, $1 < a < N$. By the division algorithm, write

$$\begin{aligned} U &= u_1 a + u_0, \quad \text{with } 0 \leq u_0 < a \\ V &= v_1 a + v_0, \quad \text{with } 0 \leq v_0 < a. \end{aligned} \tag{1}$$

Assume that $u_0$ is relatively prime to $a$, and likewise for $v_0$, since otherwise we easily extract a factor of $N$ by taking $\gcd(a, N)$. If, for a given $a$, we can determine $u_0, u_1, v_0, v_1$ then we have found $U$ and $V$.

Consider $N = u_0 v_0 \mod a$. One cannot simply determine $u_0$ and $v_0$ from the value of $N \mod a$, because $\phi(a)$ pairs of integers $(x, y) \mod a$ satisfy $xy = N \mod a$ (if $x = m u_0 \mod a$, then $y = m^{-1} v_0 \mod a$, where $\gcd(m, a) = 1$).

However, say $a$ is large, $a \geq \lceil (2N)^{1/3} \rceil > V^{2/3}$, so that $v_1$ and $u_1$ are comparatively small, $u_1, v_1 \leq V^{1/3}$, i.e., both are $< a^{1/2}$. If we consider $N \mod a - \delta$

$$N = UV = (u_1 \delta + u_0)(v_1 \delta + v_0) \mod a - \delta, \tag{2}$$

for $\delta = 0, 1$, we get, as solutions $(x, y)$ to $xy = N \mod a - \delta$, two nearby points, $(u_0, v_0)$ and $(u_0 + u_1, v_0 + v_1)$, whose coordinates are within $a^{1/2}$ of one another. This pair of points is just one pair amongst the many pairs of solutions to the above equations, for $\delta = 0, 1$. However, the fact that the solutions are nearby reduces the amount of checking that we need to do in order to find the pair of points, $(u_0, v_0)$ and $(u_0 + u_1, v_0 + v_1)$, that we seek.

Figures 1 and 2 illustrate this fact, for $N = 1910861 = 1061 \times 1801$, and $a = 157$. Thus, $U = 1061$, $u_0 = 119$, $u_1 = 8$, and $V = 1801$, $v_0 = 74$, $v_1 = 15$. Rather than just depict the solutions to $xy = N \mod a - \delta$, for $\delta = 0, 1$, we also plot the solutions for $\delta = 2, 3$ (though our algorithm below only makes use of solutions for

$\delta = 0, 1$). Plotting four sets of solutions, for $\delta = 0, 1, 2, 3$ makes it easier for the human eye to tell the points $(u_0, v_0) = (119, 74)$, $(u_0 + u_1, v_0 + v_1) = (125, 85)$, $(u_0 + 2u_1, v_0 + 2v_1) = (131, 96)$, and $(u_0 + 3u_1, v_0 + 3v_1) = (137, 107)$ from the random coincidences of nearby points as these all lie equally spaced apart and on one line.

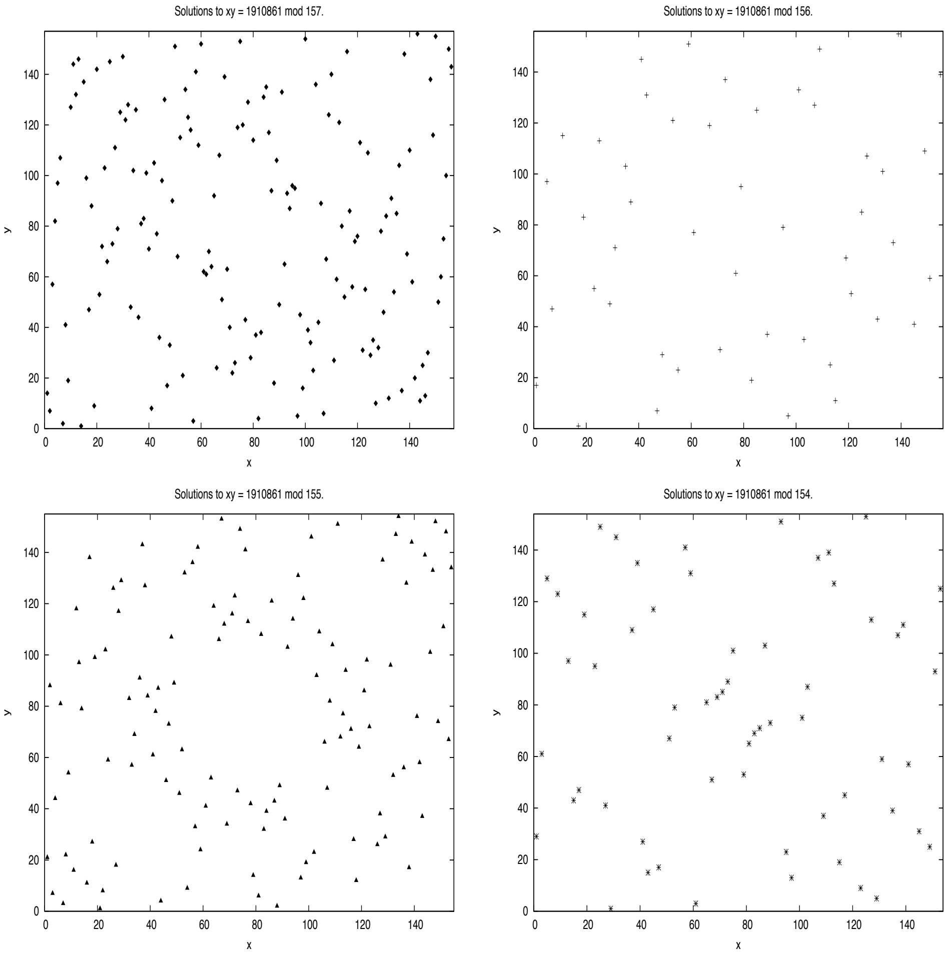

Figure 1: The solutions $(x, y)$ to $xy = 1910861 \mod 157 - \delta$, for $\delta = 0, 1, 2, 3$

So, we can set, say, $a = \lceil (2N)^{1/3} \rceil$, and partition the Cartesian plane into squares of side length $a^{1/2}$, each square being of the form $\{(x, y) \in \mathbb{R}^2 | ma^{1/2} \leq x < (m + 1)a^{1/2}, na^{1/2} \leq y < (n + 1)a^{1/2}\}$, where $m, n \in \mathbb{Z}$.

We then list *all* $\phi(a)$ pairs of integers $(x, y)$, with $1 \leq x, y \leq a$, that satisfy $xy = N \mod a$, throwing them into our squares of side lengths $a^{1/2}$. We can

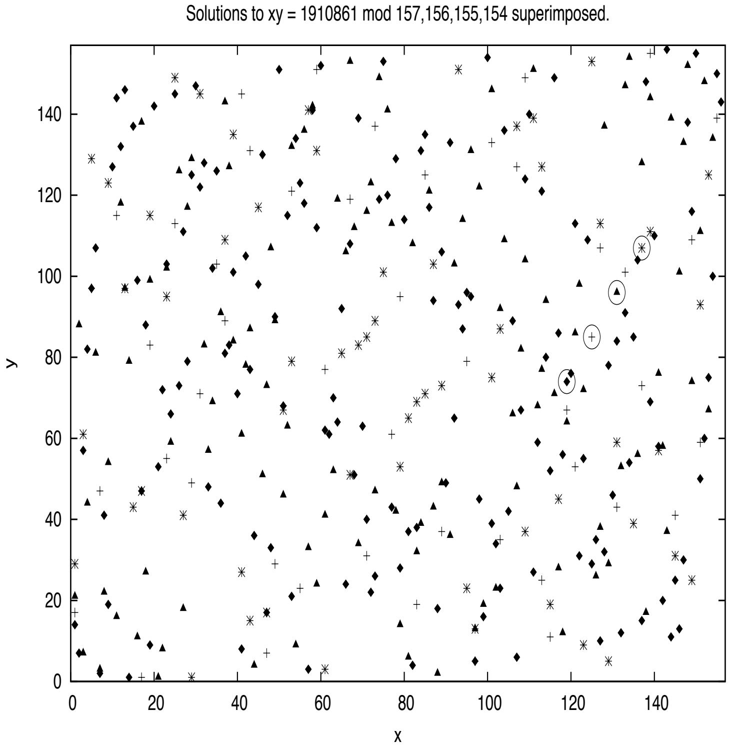

Figure 2: The solutions $(x, y)$ to $xy = 1910861 \mod 157 - \delta$, for $\delta = 0, 1, 2, 3$, superimposed. The four circled points are $(u_0, v_0) = (119, 74)$, $(u_0 + u_1, v_0 + v_1) = (125, 85)$, $(u_0 + 2u_1, v_0 + 2v_1) = (131, 96)$, and $(u_0 + 3u_1, v_0 + 3v_1) = (137, 107)$.

assume that $\gcd(a, N) = 1$, because, otherwise we easily extract a factor of $N$.

We can compute all inverses mod $a$, and hence all $(x, y) = (x, x^{-1}N) \mod a$ in $O(a)$ operations mod $a$. To compute all inverses, start with $m = 2$, multiply mod $a$ by $m$ until we arrive at 1, or hit a residue class already encountered (in which case $m$ is not invertible). Then, take the first residue not yet encountered and repeat the previous step until all residue classes are exhausted.

Having produced all solutions for the modulus $a$, we then repeat the process for the modulus $a - 1$. For each solution $(x_1, y_1)$ to $xy = N \mod a - 1$, we determine which $a^{1/2} \times a^{1/2}$ square it falls within, and consider all nearby (with each coordinate within $a^{1/2}$, wrapping to the opposite side of the larger $a \times a$ square if needed) solutions $(x_0, y_0)$ to $xy = N \mod a$ from our list of stored solutions. We set $\mu_0 = x_0$, $\nu_0 = y_0$, $\mu_1 = x_1 - \mu_0$, $\nu_1 = y_1 - \nu_0$, and check whether $(\mu_1 a + \mu_0)(\nu_1 a + \nu_0) = N$. If so, we have determined a non-trivial factor of $N$ and quit.

How much work does comparing pairs of points $(x_0, y_0)$ and $(x_1, y_1)$ entail? There are $\phi(a-1)$ solutions to $xy = N \mod a-1$, and, typically, we expect there to be only a handful of solutions to $xy = N \mod a$ whose coordinates are each within $a^{1/2}$. Each such pair of solutions gives us candidate values $\mu_0, \nu_0$ and $\mu_1, \nu_1$ for $u_0, v_0$ and $u_1, v_1$, and we check to see whether they produce $N = (u_1 a + u_0)(v_1 a + v_0)$. On average, each $a^{1/2} \times a^{1/2}$ square contains $O(1)$ points, the overall time to check all squares and points is roughly predicted to be $O(a)$. In Section 4 we obtain a runtime bound of $O(a^{1+\epsilon})$. This algorithm terminates successfully when the true points $(u_0, v_0)$ and $(u_0 + u_1, v_0 + v_1)$ are found. Since $a = O(N^{1/3})$ this gives a running time that is provably $O(N^{1/3+\epsilon})$.

The idea that lies behind the algorithm suggests the name 'Hide and Seek'. The solutions that we seek $(u_0, v_0)$ and $(u_0 + u_1, v_0 + v_1)$ are hiding amongst many solutions in the large $a \times a$ square, but, like children who have hidden next to one another while playing the game Hide and Seek, they have become easier to spot.

We summarize the above in the following algorithm.

**Algorithm 2.1.** (Hide and Seek) *Let $N = UV$ be a positive integer, and assume that $1 < U \leq V < 2U$, with $U, V \in \mathbb{Z}$. Thus $V < (2N)^{1/2}$. For given positive integers $N, r$, define*

$$H_{N,a} = \{(x, y) \mid xy = N \mod a,\ 0 \leq x, y < a\}. \tag{3}$$

*Step 1* *Set $a = \lceil (2N)^{1/3} \rceil$.*

*Step 2* *Use the Euclidean algorithm to compute* $\gcd(N, a-\delta)$ *for $\delta = 0, 1$. If either gcd is $> 1$ then we have determined a non-trivial factor of $N$ and quit.*

*Step 3* *Compute and store in an array all $\phi(a)$ points of $H_{N,a}$. This can be done using $O(a)$ arithmetic operations mod $a$ as described above.*

*Step 4* *For $0 \leq m, n < a^{1/2}$, initialize a doubly indexed array, 'Bin'. Each element, Bin[m,n], will contain a list of points and be used to partition $H_{N,a}$. Each is initially set to empty.*

*Step 5* *Partition the elements of $H_{N,a}$ according to squares of side length $a^{1/2}$ by computing, for each $(x, y) \in H_{N,a}$, the values $m = \lfloor x/a^{1/2} \rfloor$ and $n = \lfloor y/a^{1/2} \rfloor$, and appending the point $(x, y)$ to Bin[m,n].*

*Step 6* *Compute the $\phi(a-1)$ elements of $H_{N,a-1}$. For each $(x_1, y_1) \in H_{N,a-1}$:*

*Step 6a* *Determine which bin it corresponds to by computing $m = \lfloor x_1/a^{1/2} \rfloor$ and $n = \lfloor y_1/a^{1/2} \rfloor$.*

*Step 6b* *Loop through the nearby points $(x_0, y_0)$ of $H_{N,a}$ whose coordinates lie, left and downwards, within $a^{1/2}$. Typically, this entails examining the four bins Bin[$m - \epsilon_1, n - \epsilon_2$], where $\epsilon_1, \epsilon_2 \in \{0, 1\}$. However, slight care* *is needed when crossing over an edge of the $a \times a$ square- one should wrap to the opposite side of the square.*

*Step 6c* *Set $\mu_0 = x_0$, $\nu_0 = y_0$, $\mu_1 = x_1 - \mu_0$, $\nu_1 = y_1 - \nu_0$, and check whether $(\mu_1 a + \mu_0)(\nu_1 a + \nu_0) = N$. If so, we have determined a non-trivial factor of $N$ and quit.*

The storage requirement of $O(N^{1/3})$ can be improved to $O(N^{1/6})$ by generating the solutions $(x, y)$ to $xy = N \mod a - \delta$ lying in one vertical strip of width $O(a^{1/2})$ at a time (easy to do since we can choose $x$ as we please, which then determines $y$). In general, we are then no longer free to generate all modular inverses at once, and must compute inverses in intervals of size $a^{1/2}$, one at a time, at a cost, using the Euclidean algorithm, of $O(a^{\epsilon})$ per inverse.

A slight improvement on the runtime of the algorithm can be achieved my modifying the choice of $a$ so that $\phi(a)$ or $\phi(a-1)$ are comparatively small. This can be achieved by selecting an $a$ or $a-1$ with smaller distinct prime factors.

### 2.1. Variant: $1 < U \leq V < N$ Without Restriction

Say $U = N^{\alpha}$, $V = N^{1-\alpha}$, with $1/3 < \alpha \leq 1/2$. We may assume that $\alpha > 1/3$, for, if not, we can find $U$ by performing $O(N^{1/3})$ trial divisions.

Let $a = \lceil 2N^{1/3} \rceil$ (we do, here, mean $2N^{1/3}$, rather than $(2N)^{1/3}$ of the previous section, as explained below).

Instead of working with small squares of side length $a^{1/2}$, partition the $a \times a$ square into rectangles of width $w$ and height $h$, with $wh = N^{1/3}$. We would like to select $w$ roughly equal to $N^{\alpha-1/3}$ and hence $h = N^{1/3}/w$ roughly equal to $N^{2/3-\alpha}$.

These rough values of $w$ and $h$ are needed to make sure that, using the same notation as before, $(u_0, v_0)$ and $(u_0 + u_1, v_0 + v_1)$ are in the same, or neighboring, rectangles. More precisely, say $N^{\alpha-1/3} < w \leq 2N^{\alpha-1/3}$. Then $h = N^{1/3}/w \geq N^{2/3-\alpha}/2$. Then, in (1), $u_1 = \lfloor U/a \rfloor \leq N^{\alpha}/\lceil 2N^{1/3} \rceil \leq N^{\alpha-1/3}/2 < w$, and $v_1 = \lfloor V/a \rfloor \leq N^{1-\alpha}/\lceil 2N^{1/3} \rceil \leq N^{2/3-\alpha}/2 \leq h$. Thus the $x$-coordinates of $(u_0, v_0)$ and $(u_0 + u_1, v_0 + v_1)$ are $< w$ apart and the $y$-coordinates are $\leq h$ apart.

Since we do not, a priori know $\alpha$, we cannot simply set $w$ and $h$. Instead, we use an exponentially increasing set of $w$'s, for example starting with $w = 2$, and, repeatedly applying the above procedure, each time doubling the size of $w$, until $w > N^{\alpha-1/3}$ and one successfully factors $N$.

The area of each rectangle is $N^{1/3}$, and of the $a \times a$ square is approximately $N^{2/3}$, so there are $O(N^{1/3})$ rectangles (at the top and right edges these will typically be truncated), and, on average, each contains $O(1)$ solutions to $xy = N \mod a - \delta$. Running through each rectangle and its immediate neighbors, checking all pairs of points in these rectangles suggests $O(N^{1/3})$ operations are needed for a particular choice of $w$ and $h$. Since we might have to repeat this a few times, doubling the size of $w$, the overall running time gets multiplied by $O(\log N)$ which is $O(N^{\epsilon})$.

In Section 5, a running time equal to $O(N^{1/3+\epsilon})$ is proven.

The steps described in this section are summarized below.

**Algorithm 2.2.** *Let $N = UV$ be a positive integer, and assume that $1 < U \leq V < N$, with $U, V \in \mathbb{Z}$.*

```
Step 1 Carry out trial division on $N$ up to $N^{1/3}$. If a non-trivial factor of $N$ is found, quit.
Step 2 Set $a = \lceil 2N^{1/3} \rceil$.
Step 3 Use the Euclidean algorithm to compute $\gcd(N, a - \delta)$ for $\delta = 0, 1$. If either gcd is $> 1$ then we have determined a non-trivial factor of $N$ and quit.
Step 4 Compute and store, in two arrays, all the points of $H_{N,a}$ and $H_{N,a-1}$.
Step 5 Set $j = 0$. While we have not succeeded in finding a non-trivial factor of $N$:
  Step 5a Increment $j$ by 1 and set $w = 2^j$ and $h = N^{1/3}/w$.
  Step 5b For $0 \leq m < a/w$ and $0 \leq n < a/h$, initialize a doubly indexed array, 'Bin', whose elements, Bin[m,n], will contain lists of points and be used to partition $H_{N,a}$. Each bin is initially set to empty.
  Step 5c Partition the elements of $H_{N,a}$ according to rectangles of width $w$ and height $h$ by computing, for each $(x, y) \in H_{N,a}$, the values $m = \lfloor x/w \rfloor$ and $n = \lfloor y/h \rfloor$, and appending the point $(x, y)$ to Bin[m,n].
  Step 5d For each $(x_1, y_1) \in H_{N,a-1}$:
    Step 5d1 Determine which bin it corresponds to by computing $m = \lfloor x_1/w \rfloor$ and $n = \lfloor y_1/h \rfloor$.
    Step 5d2 Loop through the nearby points $(x_0, y_0)$ of $H_{N,a}$ whose coordinates lie, left and downwards, within $w$ and $h$ respectively. Typically, this entails examining the four bins Bin[$m - \epsilon_1, n - \epsilon_2$], where $\epsilon_1, \epsilon_2 \in \{0, 1\}$. However, slight care is needed when crossing over an edge of the $a \times a$ square- one should wrap to the opposite side of the square.
    Step 5d3 Set $\mu_0 = x_0$, $\nu_0 = y_0$, $\mu_1 = x_1 - \mu_0$, $\nu_1 = y_1 - \nu_0$, and check whether $(\mu_1 a + \mu_0)(\nu_1 a + \nu_0) = N$. If so, we have determined a non-trivial factor of $N$ and quit.
  Step 5e Free up the memory used by 'Bin'.
```

## 3. Towards a Subexponential Bound

The above algorithm exploits the fact that when $a$ is large, and $\delta$ is small, the points with coordinates $(U, V) \mod a - \delta$ are close to one another. In fact they lie equally spaced on a line with common horizontal difference $u_1$, and vertical difference $v_1$.

An obvious thing to try is to reduce the size of $a$. However, as $a$ decreases, $u_1$ and $v_1$ increase so that not only do the points $(u_0, v_0)$ and $(u_0 + u_1, v_0 + v_1)$ move far apart, the latter point soon falls far outside the square of side length $a$.

To fix this, one can view (1) as the base $a$ expansion of $U$ and $V$. When $a$ is smaller, one could instead use a polynomial expansion

$$\begin{aligned} U &= u_{d_1}a^{d_1} + \ldots + u_1 a + u_0, \quad 0 \le u_i < a \\ V &= v_{d_2}a^{d_2} + \ldots + v_1 a + v_0, \quad 0 \le v_i < a, \end{aligned} \tag{4}$$

with $u_{d_1} \neq 0$ and $v_{d_1} \neq 0$. For simplicity in what follows, assume that the degrees of both polynomials are equal, $d_1 = d_2 = d$, so that both $U$ and $V$ satisfy $a^d \le U, V < a^{d+1}$.

A polynomial of degree $d$ is determined uniquely by $d+1$ values. Imitating the approach in Section 1, we evaluate $N \mod a - \delta$ for $d+1$ values of $\delta$. A natural choice might be $\delta = 0, \pm 1, \pm 2, \ldots$, but, to keep our polynomial values positive, we consider non-negative values of $\delta$, and, for good measure, take extra values, $\delta = 0, 1, 2, \ldots 2d$ (by extra, we mean $\delta \le 2d$ rather than $d\, leqd$). Now,

$$N = UV = (u_d\delta^d + \ldots + u_1\delta + u_0)(v_d\delta^d + \ldots + v_1\delta + v_0) \mod a - \delta. \tag{5}$$

Since $0 \le u_j < a$, we have

$$u_d\delta^d + \ldots + u_1\delta + u_0 < a\lambda(d, \delta) \tag{6}$$

where

$$\lambda(d, \delta) = \delta^d + \delta^{d-1} + \ldots + 1 = (\delta^{d+1} - 1)/(\delta - 1) \sim \delta^d, \quad \text{as } \delta \to \infty. \tag{7}$$

and similarly for the $v_j$'s.

For each $\delta$ one lists all solutions $(x, y)$ to

$$xy = N \mod a - \delta \tag{8}$$

$$0 < x, y < a\lambda(d, \delta). \tag{9}$$

The number of points $(x, y)$ for a given $\delta$ is $\phi(a - \delta)$ per $a \times a$ square, and hence, overall, equals

$$\phi(a - \delta)\lambda(d, \delta)^2 = O(a(2d)^{2d}). \tag{10}$$

We are again assuming that $\gcd(a - \delta, N) = 1$, otherwise one easily pulls out a factor of $N$.

We need a method to recognize the solutions that we seek $(u_d\delta^d + \ldots + u_0, v_d\delta^d + \ldots + v_0)$ hiding amongst all the $(x, y)$'s. This leads to the question:

Let $X > 0$ and let $S_0, S_1, \ldots, S_{2d}$ be $2d + 1$ sets of points $\in \mathbb{Z}^2$ all of whose coordinates are positive and $\le X$. Assume that amongst these points there exists

$2d+1$ points, one from each $S_\delta$, whose coordinates are described by polynomials $u(\delta), v(\delta) \in \mathbb{Z}[\delta]$ of degree $d$. More precisely, for each $0 \le \delta \le 2d$ there exists a point $(x_\delta, y_\delta) \in S_\delta$ such that

$$\begin{aligned} x_\delta &= u(\delta) = u_d\delta^d + \ldots + u_0 \\ y_\delta &= v(\delta) = v_d\delta^d + \ldots + v_0. \end{aligned} \tag{11}$$

Can one find these $2d+1$ points much more efficiently than by exhaustively searching through all possible $2d+1$ tuples of points? For example, can one find these points in time $O(X^\alpha d^{\beta d})$ for some $\alpha, \beta > 0$?

In our application, $X = O(a(2d)^{2d})$. Since $N = UV$ and $a^d \le U < V < a^{d+1}$, we have $a < N^{1/(2d)}$. Assuming that there is an $O(X^\alpha d^{\beta d})$ time algorithm for finding points with polynomial coordinates, on taking $d$ proportionate to

$$\left(\frac{\log N}{\log\log N}\right)^{1/2} \tag{12}$$

one gets a factoring algorithm requiring

$$\exp\left(\gamma(\log N \log\log N)^{1/2}\right) \tag{13}$$

time and storage, for some $\gamma > 0$.

One can cut back a bit on the search space, by noting, for example, that the coefficients of $u(\delta)$ and $v(\delta)$ are integers (this imposes a divisibility restriction on finite differences between points lying on the polynomial), and, in our particular application, that the coefficients are non-negative and bounded, and this restricts the rate of growth of the polynomials. However, to get down to a running time polynomial in $X$, one needs to do much better.

## 4. Uniform Distribution

Let $\gcd(a, N) = 1$. A classic application of Kloosterman sums shows that the points $(x, y) \mod a$ satisfying $xy = N \mod a$ become uniformly distributed in the square of side length $a$ as $a \to \infty$. While the tools used in this section are fairly standard, they will also be applied in the next section to estimate the running time of the Hide and Seek algorithm. Similar theorems can be found in the literature ([1], [2], [3], [4], [8], [9], [11]), often with restrictions to prime values of $a$ or to $N = 1$.

Consider the following identity which detects pairs of integers $(x, y)$ such that $xy = N \mod a$:

$$\frac{1}{a}\sum_{k=0}^{a-1} e\left(\frac{k}{a}(y - \bar{x}N)\right) = \begin{cases} 1 & \text{if } xy = N \mod a \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

where $e(z) = \exp(2\pi i z)$, and where $\bar{x}$ stands for any integer congruent to $x^{-1}$ mod $a$, if the inverse exists. Recall that we have assumed $\gcd(a, N) = 1$ so that any solution to $xy = N \mod a$ must have $\gcd(x, a) = 1$. Thus, for such solutions, $x^{-1} \mod a$ exists.

Let $R$ be the rectangle bounded horizontally by $x_1, x_2 \in \mathbb{Z}$ and vertically by $y_1, y_2 \in \mathbb{Z}$, where $0 \le x_1 < x_2 \le a$ and $0 \le y_1 < y_2 \le a$:

$$R = R(x_1, x_2, y_1, y_2) = \{(x, y) \in \mathbb{Z}^2 | x_1 \le x < x_2, y_1 \le y < y_2\}. \qquad (15)$$

Let $c_R(N, a)$ denote the number of pairs of integers $(x, y)$ that lie in the rectangle $R$, and satisfy $xy = N \mod a$:

$$c_R(N, a) = \sum_{\substack{(x,y)\in R \\ xy=N \mod a}} 1. \qquad (16)$$

The identity above gives

$$c_R(N, a) = \frac{1}{a} \sum_{k=0}^{a-1} \sum_{\substack{(x,y)\in R \\ \gcd(x,a)=1}} e\left(\frac{k}{a}(y - \bar{x}N)\right). \qquad (17)$$

Notice that we only need to restrict $x$ to $\gcd(x, a) = 1$ and that $y$ runs over all residues in $y_1 \le y < y_2$. This will allow us to deal with the sum over $y$ as a geometric series.

The $k = 0$ term provides the main contribution while the other terms can be estimated using bounds for Kloosterman sums. We require two lemmas. The first considers the main contribution, and the second bounds the remaining terms.

**Lemma 4.1.** *The $k = 0$ term in (17) equals*

$$\frac{area(R)}{a^2} \phi(a) + O(a^\epsilon) \qquad (18)$$

*for any $\epsilon > 0$.*

*Proof.* The $k = 0$ term is

$$\frac{1}{a} \sum_{\substack{(x,y)\in R \\ \gcd(x,a)=1}} 1 = \frac{(y_2 - y_1)}{a} \sum_{\substack{x_1 \le x < x_2 \\ \gcd(x,a)=1}} 1. \qquad (19)$$

Using the Mobius function we have

$$\begin{aligned} \sum_{\substack{x_1 \le x < x_2 \\ \gcd(x,a)=1}} 1 &= \sum_{x_1 \le x < x_2} \sum_{d | \gcd(x,a)} \mu(d) = \sum_{d|a} \mu(d) \sum_{x_1/d \le x < x_2/d} 1 \\ &= \sum_{d|a} \mu(d)((x_2 - x_1)/d + O(1)) = (x_2 - x_1) \prod_{p|a} (1 - 1/p) + O(\tau(a)). \qquad (20) \end{aligned}$$

where $\tau(a)$ equals the number of divisors of $a$ and is $O(a^\epsilon)$ for any $\epsilon > 0$. This implies that the $k = 0$ contribution to $c_R(N, a)$ equals

$$\frac{\text{area}(R)}{a^2}\phi(a) + O((y_2 - y_1)a^{-1+\epsilon}) \tag{21}$$

which gives the lemma. □

The next lemma bounds the contribution of the $k \geq 1$ terms in (17).

**Lemma 4.2.** *For any $\epsilon > 0$ we have*

$$\frac{1}{a}\sum_{k=1}^{a-1} \sum_{\substack{(x,y)\in R \\ \gcd(x,a)=1}} e\left(\frac{k}{a}(y - \bar{x}N)\right) = O(a^{1/2+\epsilon}). \tag{22}$$

*Proof.* One can separate the sum over $y$ and evaluate it as a geometric series obtaining for the left-hand side above

$$\frac{1}{a}\sum_{k=1}^{a-1} \frac{e\left(\frac{k}{a}y_2\right) - e\left(\frac{k}{a}y_1\right)}{e\left(\frac{k}{a}\right) - 1} \sum_{\substack{x_1 \leq x < x2 \\ \gcd(x,a)=1}} e\left(\frac{-k}{a}\bar{x}N\right). \tag{23}$$

Taking absolute values we get an upper bound of

$$\frac{1}{a}\sum_{k=1}^{a-1} \frac{\left|\sin\left(\frac{\pi k}{a}(y_2 - y_1)\right|\right.}{\left|\sin\left(\frac{\pi k}{a}\right)\right|} \left|\sum_{\substack{x_1 \leq x < x2 \\ \gcd(x,a)=1}} e\left(\frac{-k}{a}\bar{x}N\right)\right|. \tag{24}$$

Next, notice that the terms $k$ and $a - k$ give the same contribution, so we may restrict our attention to just the terms $1 \leq k \leq (a-1)/2$. If $a-1$ is odd, the middle term is left out at a cost of $O(1)$, and the bound becomes

$$\frac{2}{a}\sum_{1\leq k\leq (a-1)/2} \frac{\left|\sin\left(\frac{\pi k}{a}(y_2 - y_1)\right|\right.}{\left|\sin\left(\frac{\pi k}{a}\right)\right|} \left|\sum_{\substack{x_1 \leq x < x2 \\ \gcd(x,a)=1}} e\left(\frac{-k}{a}\bar{x}N\right)\right| + O(1). \tag{25}$$

The second sum above over $x$ can be expressed in terms of Kloosterman sums, and using estimates for Kloosterman sums one has

$$\sum_{\substack{x_1 \leq x < x2 \\ \gcd(x,a)=1}} e\left(\frac{-k}{a}\bar{x}N\right) = O(a^{1/2+\epsilon}\gcd(k,a)^{1/2}). \tag{26}$$

For a proof, see Lemma 4 on page 36 of Hooley's book [5] where a proof is given (his $r$ corresponds to our $a$, and his $l$ is $-kN$. Also recall that we are assuming $\gcd(N, a) = 1$ so that $N$ does not appear in the gcd of the $O$ term).

Furthermore, using the Taylor expansion of $\sin(x)$ one obtains the two inequalities

$$\begin{aligned} \sin(x) &\le \min(x,1), \quad x \ge 0, \\ 1/\sin(x) &< 2/x, \quad 0 < x < \pi/2. \end{aligned} \tag{27}$$

For the second inequality, use $0 < x/2 < x - x^3/3! < \sin(x)$ in the stated interval.

Applying (27) and (26) gives an upper bound for (25) of

$$O\left( a^{-1/2+\epsilon} \sum_{1 \le k \le (a-1)/2} \min(\frac{\pi k}{a}(y_2 - y_1), 1) \frac{2a}{\pi k} \gcd(k,a)^{1/2} + 1 \right). \tag{28}$$

Breaking up the sum into $1 \le k \le a/(\pi(y_2 - y_1))$ and $a/(\pi(y_2-y_1)) < k \le (a-1)/2$, the sum over $k$ in the $O$ term equals

$$2(y_2 - y_1) \sum_{1 \le k \le a/(\pi(y_2-y_1))} \gcd(k,a)^{1/2} + \frac{2a}{\pi} \sum_{a/(\pi(y_2-y_1)) < k \le (a-1)/2} \gcd(k,a)^{1/2}/k. \tag{29}$$

Both kinds of sums can be easily handled (the first can also be found in Hooley). Let $X > 0$. Then,

$$\sum_{1 \le k \le X} \gcd(k,a)^{1/2} \le \sum_{d|a} d^{1/2} \sum_{\substack{1 \le k \le X \\ d|k}} 1 \le X \sum_{d|a} d^{-1/2} = O(X a^{\epsilon}). \tag{30}$$

Next, let $0 < X_1 < X_2$. Then

$$\begin{aligned} \sum_{X_1 < k \le X_2} \gcd(k,a)^{1/2}/k &\le \sum_{d|a} d^{1/2} \sum_{\substack{X_1 < k \le X_2 \\ d|k}} 1/k \\ &= O\left( \log(X_2 - X_1 + 2) \sum_{d|a} d^{-1/2} \right) \end{aligned} \tag{31}$$

which equals

$$O(\log(X_2 - X_1 + 2) a^{\epsilon}). \tag{32}$$

Applying (30) and (32) to (29), we have that (28) is

$$O(a^{1/2+\epsilon}), \tag{33}$$

completing the proof. □

These two lemmas together give the following theorem.

**Theorem 4.3.** *Let* $\gcd(N,a) = 1$ *and* $R$ *as described in (15). Then,* $c_R(N,a)$*, the number of solutions* $(x,y)$ *to* $xy = N \mod a$ *with* $(x,y)$ *lying in the rectangle* $R$*, is equal to*

$$\frac{\mathit{area}(R)}{a^2} \phi(a) + O(a^{1/2+\epsilon}) \tag{34}$$

*for any* $\epsilon > 0$.

This theorem shows that the points $(x, y)$ satisfying $xy = N \mod a$ are uniformly dense in the sense that the rectangle $R$ contains its fair share of solutions, so long as the area of $R$ is of larger size than $a^{3/2+\epsilon}$.

For example, if $R$ is a square, it needs to have side length at least $a^{3/4+\epsilon}$ to contain its fair share of points. This is considerably larger than the side length of $a^{1/2}$ that is used in the algorithm of Section 1.

The papers of Shparlinski ([8], [9]) contain many references to the problem of uniform distribution and discusses improved results on average over $N$.

## 5. Second Moment and Running Time

We now examine the assertion made in Section 1 that $O(a^{1+\epsilon})$ time is needed to scan across all $a$ squares of side length $a^{1/2}$ and their immediate neighbors, comparing all pairs of points contained in said squares.

The running time of the algorithm in Section 2.1, i.e., in the case $U \le V < 2U$ depends on how the solutions to $xy = N \mod a$ and $x'y' = N \mod (a-1)$ are distributed amongst the small squares of side length $a^{1/2}$. In Section 5.1 we will consider the running time of the variant in Section 2.1 which is used for the general situation $1 < U \le V < N$.

Let $S$ denote one such square, i.e., of side length $a^{1/2}$. Then the running time needed to examine just the square $S$, looking at all pairs of points $(x, y)$, $(x', y')$ in $S$ is $O(c_S(N, a)c_S(N, a-1))$, which, by the arithmetic geometric inequality is $O(c_S(N, a)^2 + c_S(N, a-1)^2)$. The algorithm also requires us to compare points in neighboring squares, say $S_1$ and $S_2$, which, similarly, takes $O(c_{S_1}(N, a)^2 + c_{S_2}(N, a-1)^2)$ time. Hence, the overall running time to compare pairs of points is

$$O\left(\sum_S c_S(N, a)^2 + c_S(N, a-1)^2\right), \tag{35}$$

the sum being over the roughly $a$ squares of side length $a^{1/2}$ that partition the $a \times a$ square $\{(x, y) \in \mathbb{Z}^2 | 0 \le x, y < a\}$ (at the top and right edges we get rectangles, unless $a^{1/2}$ is an integer).

Consider now the contribution from the points $\mod a$:

$$\sum_S c_S(N, a)^2. \tag{36}$$

For convenience, rather than deal with squares $S$ of side length $a^{1/2}$, we will estimate (36) by making a small adjustment and partitioning the $a \times a$ square into squares of side length

$$b = \lceil a^{1/2} \rceil. \tag{37}$$

We also assume that $\gcd(b, a) = 1$. If not, replace $b$ with $b + 1$ until this condition holds. By equation (20), this will not take long to occur, so that, for any $\epsilon > 0$, $b = a^{1/2} + O(a^\epsilon)$.

Thus, consider the squares

$$B_{ij} = \{(x, y) \in \mathbb{Z}^2 | ib \leq x < (i + 1)b, jb \leq y < (j + 1)b\} \tag{38}$$

with $0 \leq i, j < a/b - 1$.

Since $b \nmid a$ , these will not entirely cover the $a \times a$ square, but the number of points $(x, y) \in \mathbb{Z}^2$ satisfying $xy = N \mod a$ that are neglected at the right most and top portions of the $a \times a$ square is, by (20), $O(\phi(a)b/a)$, and these therefore contribute $O(\phi(a)^2 b^2/a^2) = O(\phi(a))$ to (36).

The points $(x, y) \in \mathbb{Z}^2$ belonging to an $a^{1/2} \times a^{1/2}$ square $S$ are contained entirely in at most four squares, say $B_{i_1 j_1}, B_{i_2 j_2}, B_{i_1 j_3}, B_{i_4 j_4}$, of side length $b$. Therefore,

$$c_S(N, a)^2 \leq (c_{B_{i_1 j_1}}(N, a) + c_{B_{i_2 j_2}}(N, a) + c_{B_{i_3 j_3}}(N, a) + c_{B_{i_4 j_4}}(N, a))^2 \tag{39}$$

which, by the Cauchy-Schwartz inequality is

$$\leq 4(c_{B_{i_1 j_1}}(N, a)^2 + c_{B_{i_2 j_2}}(N, a)^2 + c_{B_{i_3 j_3}}(N, a)^2 + c_{B_{i_4 j_4}}(N, a)^2). \tag{40}$$

Since each $B$ square overlaps with $O(1)$ $S$ squares, we thus have that

$$\sum_S c_S(N, a)^2 = O\left(\phi(a) + \sum_B c_B(N, a)^2\right), \tag{41}$$

the $\phi(a)$ accounting for the contribution from the neglected portion at the right most and top portions of the $a \times a$ square.

A similar consideration for the points satisfying $xy = N \mod (a-1)$, partitioning the larger $a \times a$ square into squares $D$ of side length $d$, where $d$ is the smallest integer greater than $\lceil (a-1)^{1/2} \rceil$ which is coprime to $a$, gives the same kind of sum

$$\sum_S c_S(N, a-1)^2 = O\left(\phi(a-1) + \sum_D c_D(N, a-1)^2\right). \tag{42}$$

Therefore, we need to estimate the second moment

$$\sum_B c_B(N, a)^2 \tag{43}$$

where $B$ ranges over all $\lfloor a/b \rfloor^2$ squares of the form (38). To prove that the running time of the hide and seek algorithm of Section 1 is $O(N^{1/3+\epsilon})$ we need to prove that (43) is $O(a^{1+\epsilon})$.

**Theorem 5.1.** *Let*

$$P = \{B_{ij}\}_{0\le i,j<a/b-1} \tag{44}$$

*Then*

$$\sum_{B\in P} c_B(N,a)^2 = O(a^{1+\epsilon}). \tag{45}$$

*Proof.* Rather than look at just the $a\times a$ square, it is helpful to consider the $ba\times ba$ square $\{(x,y)\in\mathbb{Z}^2 | 0\le x,y<ba\}$. The advantage of looking at the larger square will become apparent when we turn to the discrete Fourier transform, and will be summing over all the $a$th roots of unity.

This larger square can be partitioned into $b^2$ squares of side length $a$. Because the solutions to $xy = N \mod a$ repeat $\mod a$, we can count each $c_B(N,a)^2$ once per $a\times a$ square, by summing $c_{B'}(N,a)^2 = c_B(N,a)^2$ over all $b^2$ translates $B' = B + (r_1a, r_2a)$ of $B$, with $0\le r_1,r_2<b$.

On the other hand, we can also partition the $ba\times ba$ square into $a^2$ squares of side length $b$:

$$P_2 = \{B_{ij}\}_{0\le i,j\le a-1} \tag{46}$$

with $B_{ij}$ given by (38).

Each translate of a $B$ square, $B' = B + (r_1a, r_2a)$, is covered by at most four $B_{ij}\in P_2$, and each $B_{ij}\in P_2$ overlaps at most four such translates of $B$.

Hence, applying the Cauchy-Schwartz inequality as before,

$$b^2\sum_{B\in P} c_B(N,a)^2 = O\left(\sum_{B\in P_2} c_B(N,a)^2\right). \tag{47}$$

To study $c_B(N,a)^2$ we multiply equation (17) by its conjugate, giving

$$c_B(N,a)^2 = \frac{1}{a^2}\sum_{0\le k_1,k_2\le a-1}\ \sum_{\substack{(x_1,y_1)\in B\\ \gcd(x_1,a)=1}}\ \sum_{\substack{(x_2,y_2)\in B\\ \gcd(x_2,a)=1}} e\left(\frac{k_1}{a}(y_1-\bar{x_1}N) - \frac{k_2}{a}(y_2-\bar{x_2}N)\right). \tag{48}$$

Next, sum over all $B_{ij}\in P_2$, and break up each sum over $(x,y)\in B_{ij}$ into a double sum $ib\le x<(i+1)b$, $jb\le y<(j+1)b$,

$$\begin{aligned}\sum_{B\in P_2} c_B(N,a)^2 = \frac{1}{a^2}\sum_{0\le k_1,k_2\le a-1} &\left(\sum_{i=0}^{a-1}\ \sum_{\substack{ib\le x_1,x_2<(i+1)b\\ \gcd(x_1,a)=\gcd(x_2,a)=1}} e\left(-\frac{N}{a}(k_1\bar{x_1}-k_2\bar{x_2})\right)\right)\\ \times &\left(\sum_{j=0}^{a-1}\ \sum_{jb\le y_1,y_2<(j+1)b} e\left(\frac{k_1y_1-k_2y_2}{a}\right)\right).\end{aligned} \tag{49}$$

Now, the inner most sum,

$$\sum_{jb\leq y_1,y_2<(j+1)b} e\left(\frac{k_1y_1-k_2y_2}{a}\right), \tag{50}$$

is a product of two geometric series and equals

$$e((k_1-k_2)jb/a)\frac{e(k_1b/a)-1}{e(k_1/a)-1}\frac{e(-k_2b/a)-1}{e(-k_2/a)-1}. \tag{51}$$

We understand $\frac{e(kb/a)-1}{e(k/a)-1}$ to equal $b$ if $k=0 \mod a$. Summing (51) over $0\leq j\leq a-1$ gives

$$\begin{cases} a\left|\frac{e(k_1b/a)-1}{e(k_1/a)-1}\right|^2 & \text{if } k_1=k_2 \mod a\\ 0 \quad \text{otherwise}\end{cases}$$

(recall that we have chosen $b$ so that $\gcd(b,a)=1$). Therefore, only the terms with $k_1=k_2$ contribute to (49) and it equals

$$\frac{1}{a}\sum_{k=0}^{a-1}\left(\sum_{i=0}^{a-1}\sum_{\substack{ib\leq x_1,x_2<(i+1)b\\ \gcd(x_1,a)=\gcd(x_2,a)=1}} e\left(-\frac{N}{a}(k(\bar{x_1}-\bar{x_2}))\right)\right)\left|\frac{e(kb/a)-1}{e(k/a)-1}\right|^2. \tag{52}$$

The $k=0$ term gives, on separating the sum over $x_1$ and $x_2$,

$$\frac{b^2}{a}\sum_{i=0}^{a-1}\left(\sum_{\substack{ib\leq x<(i+1)b\\ \gcd(x,a)=1}} 1\right)^2 \tag{53}$$

which, by (20) and using $b\sim a^{1/2}$, equals

$$b^2(\phi(a)b/a+O(a^\epsilon))^2=O(\phi(a)^2). \tag{54}$$

Next, we deal with the terms $1\leq k\leq a-1$. The sum over $i$ in (52) equals

$$\sum_{i=0}^{a-1}\sum_{ib\leq x_1,x_2<(i+1)b} A_{x_1,x_2}(-Nk), \tag{55}$$

where

$$A_{x_1,x_2}(t)=\begin{cases}0 \quad \text{if } \gcd(x_1x_2,a)>1\\ e(t(\bar{x_1}-\bar{x_2})/a) \quad \text{otherwise.}\end{cases}$$

To analyze this sum, we use the two dimensional discrete Fourier transform

$$\hat{A}_{m_1,m_2}(t)=\sum_{0\leq x_1,x_2\leq a-1} A_{x_1,x_2}(t)e\left(-\frac{m_1x_1+m_2x_2}{a}\right), \tag{56}$$

so that

$$A_{x_1,x_2}(t) = \frac{1}{a^2} \sum_{0 \le m_1, m_2 \le a-1} \hat{A}_{m_1,m_2}(t) e\left(\frac{m_1 x_1 + m_2 x_2}{a}\right), \tag{57}$$

and (55) equals, on changing order of summation,

$$\frac{1}{a^2} \sum_{0 \le m_1, m_2 \le a-1} \hat{A}_{m_1,m_2}(-Nk) \left( \sum_{0 \le i \le a-1} \sum_{ib \le x_1, x_2 < (i+1)b} e\left(\frac{m_1 x_1 + m_2 x_2}{a}\right) \right). \tag{58}$$

The bracketed sum over $i$ is similar to the sum over $j$ worked out above and equals

$$\begin{cases} a \left| \frac{e(m_1 b/a) - 1}{e(m_1/a) - 1} \right|^2 & \text{if } m_2 = -m_1 \mod a \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, (58) equals

$$\frac{1}{a} \sum_{m=0}^{a-1} \hat{A}_{m,a-m}(-Nk) \left| \frac{e(mb/a) - 1}{e(m/a) - 1} \right|^2. \tag{59}$$

So, (52), and hence (49), equals

$$\frac{1}{a^2} \sum_{k=0}^{a-1} \sum_{m=0}^{a-1} \hat{A}_{m,a-m}(-Nk) \left| \frac{e(mb/a) - 1}{e(m/a) - 1} \right|^2 \left| \frac{e(kb/a) - 1}{e(k/a) - 1} \right|^2. \tag{60}$$

But,

$$\begin{aligned} \hat{A}_{m,a-m}(-Nk) &= \sum_{\substack{0 \le x_1, x_2 \le a-1 \\ \gcd(x_1 x_2, a) = 1}} e\left(-\frac{Nk}{a}(\bar{x_1} - \bar{x_2})\right) e\left(-\frac{m x_1 - m x_2}{a}\right) \\ &= \left| \sum_{\substack{0 \le x \le a-1 \\ \gcd(x,a) = 1}} e\left(-\frac{Nk\bar{x} + mx}{a}\right) \right|^2. \end{aligned} \tag{61}$$

However, the sum on the right-hand side is a Kloosterman sum,

$$\sum_{\substack{0 \le x \le a-1 \\ \gcd(x,a) = 1}} e\left(-\frac{Nk\bar{x} + mx}{a}\right) = S(-m, -Nk, a), \tag{62}$$

and are known [10] [6] to satisfy the bound

$$|S(-m, -Nk, a)| \le \tau(a) \gcd(m, k, a)^{1/2} a^{1/2} = O(a^{1/2+\epsilon} \gcd(k, a)^{1/2}) \tag{63}$$

(recall we are assuming that $\gcd(N, a) = 1$ so that $N$ does not appear on the right-hand side of this inequality). Applying this bound to $\hat{A}_{m,a-m}(-Nk)$, shows that (60) is

$$O\left(\frac{a^\epsilon}{a}\sum_{k=1}^{a-1}\sum_{m=0}^{a-1}\gcd(k,a)\left|\frac{e(mb/a)-1}{e(m/a)-1}\right|^2\left|\frac{e(kb/a)-1}{e(k/a)-1}\right|^2+\phi(a)^2\right). \qquad (64)$$

The $\phi(a)^2$ terms comes from the $k = 0$ contribution, (54). We must isolate this term, otherwise the estimate below will be too large.

Separating sums gives

$$O\left(\frac{a^\epsilon}{a}\left(\sum_{k=1}^{a-1}\gcd(k,a)\left|\frac{e(kb/a)-1}{e(k/a)-1}\right|^2\right)\left(\sum_{m=0}^{a-1}\left|\frac{e(mb/a)-1}{e(m/a)-1}\right|^2\right)+\phi(a)^2\right). \qquad (65)$$

Both sums can be bounded using the same approach as for (24) in the previous section, namely: combining terms $k$ and $a-k$ (similarly for the $m$ sum, but taking the $m = 0$ term alone), breaking up the sum into the terms with $k \le a/(\pi b) \sim a^{1/2}/\pi$ (respectively, $m$), applying inequalities (27), estimating the resulting sums, using $b \sim a^{1/2}$, we find, for any $\epsilon > 0$, that (65) equals

$$O(b^2 a^{1+\epsilon}). \qquad (66)$$

We have thus estimated the sum that appears on the right-hand side of (47). The sum that we wish to bound appears on the left-hand side of (47) but with an extra factor of $b^2$. Hence, dividing the above by $b^2$ gives $O(a^{1+\epsilon})$ for the sum in theorem. □

Remark: In certain cases, such as when $a = p^2$, with $p$ prime, one can improve the above estimate for the second moment to $O(a)$ by taking $b = p$ and, for $x = jp + l$, with $\gcd(l, p) = 1$, using $\bar{x} = \bar{l}^2(l - jp)$.

### 5.1. Running Time of the Variant for $1 < U \le V < N$

Instead of partitioning the $a \times a$ square into smaller squares of side length $b \sim a^{1/2}$, we partition it into rectangles $R$ of width $w < a$ and height $h < a$, where $w, h \in \mathbb{Z}$ and, for convenience, $\gcd(w, a) = \gcd(h, a) = 1$.

We partition the $a \times a$ square and also the larger $wa \times ha$ rectangle into smaller rectangles $R$:

$$\begin{aligned} R &= R_{ij} = \{(x,y) \in \mathbb{Z}^2 | iw \le x < (i+1)w, jh \le y < (j+1)h\} \\ Q &= \{R_{ij}\}_{\substack{0\le i<a/w-1 \\ 0\le j<a/h-1}} \\ Q_2 &= \{R_{ij}\}_{0\le i,j\le a-1}. \end{aligned}$$

As in Section 4.1, we have

$$wh \sum_{R \in Q} c_R(N,a)^2 = O\left( \sum_{R \in Q_2} c_R(N,a)^2 \right). \tag{67}$$

with $wh$ appearing on the left-hand side since the large $wa \times ha$ rectangle has that many copies of the $a \times a$ square.

Using the discrete Fourier transform, as before,

$$\sum_{R \in Q_2} c_R(N,a)^2 = \frac{1}{a^2} \sum_{k=0}^{a-1} \sum_{m=0}^{a-1} |S(-m,-Nk,a)|^2 \left| \frac{e(mw/a)-1}{e(m/a)-1} \right|^2 \left| \frac{e(kh/a)-1}{e(k/a)-1} \right|^2. \tag{68}$$

This useful identity expresses the second moment for the larger $wa \times ha$ rectangle as a sum involving Kloosterman sums.

The $k = 0$ term can be estimated as in (54) and asymptotically equals

$$\frac{h^2 w^2}{a^2} \phi(a)^2. \tag{69}$$

For the $k \geq 1$ terms, we use bound (63) to estimate the Kloosterman sums and separate the double sum above to get a contribution of

$$O\left( \frac{a^\epsilon}{a} \left( \sum_{k=1}^{a-1} \gcd(k,a) \left| \frac{e(kh/a)-1}{e(k/a)-1} \right|^2 \right) \left( \sum_{m=0}^{a-1} \left| \frac{e(mw/a)-1}{e(m/a)-1} \right|^2 \right) \right). \tag{70}$$

The first sum is estimated to equal $O(\tau(a)ah)$ while the second sum is $O(aw)$, giving, for $k \geq 1$, a contribution of

$$O(a^{1+\epsilon} wh) \tag{71}$$

for any $\epsilon > 0$. Putting (69) and (71) together, then dividing the left-hand side of (67) by $wh$ gives the following estimate for the second moment:

**Theorem 5.2.** *Let $1 < w, h < a$, with $\gcd(w,a) = \gcd(h,a) = 1$. Then, using the notation above, we have an estimate for the second moment that depends on the area $wh$ of the rectangles $R$:*

$$\sum_{R \in Q} c_R(N,a)^2 = \begin{cases} O(a^{1+\epsilon}) & \text{if } wh = O(a^{1+\epsilon}), \\ O(wh\phi(a)^2/a^2) & \text{if } wh \gg a^\lambda \text{ for some } \lambda > 1. \end{cases} \tag{72}$$

Remark: if $\gcd(w,a) = \gcd(h,a) = 1$ does not hold, one can bound the left-hand side of (72) by comparing with the same kind of sum, but where $w$ and $h$ are incremented, as before, by at most $O(a^\epsilon)$ until this gcd condition holds. So long as $w, h \gg a^\epsilon$ to begin with, the estimates in the above theorem are unaffected.

In Section 1.2, our choice of $w$ and $h$ has $wh = O(a)$, and the estimate for the second moment is thus $O(a^{1+\epsilon})$, as in the previous section.

The second estimate of the theorem (not relevant for our particular application), $O(wh\phi(a)^2/a^2)$, can probably be turned into an asymptotic formula and a central limit theorem proven. This will remain an inquiry for the future.

**Acknowledgements** I wish to thank Andrew Granville, Jeffrey Lagarias, Carl Pomerance, and Matthew Young for helpful feedback, and the referee for making a number of suggested improvements to the presentation and description of the algorithms.

## References

[1] J. Beck, and M. R. Khan, *On the uniform distribution of inverses modulo n*, Periodica Mathematica Hungarica **44** (2002), 147–155.

[2] F. P. Boca, C. Cobeli and A. Zaharescu, *Distribution of Lattice Points Visible from the Origin*, Commun. Math. Phys. **213** (2000), 433–470.

[3] A. Fujii, and Y.Kitaoka, *On plain lattice points whose coordinates are reciprocals modulo a prime*, Nagoya Math. J. **147** (1997), 137–146.

[4] D.R. Heath-Brown, *Arithmetic applications of Kloosterman sums*, Nieuw Arch. Wiskd. (5) **1** (2000), no. 4, 380–384.

[5] C. Hooley, *Applications of sieve methods to the theory of numbers*, Cambridge University Press (1976).

[6] H. Iwaniec, E. Kowalski, *Analytic Number Theory*, Colloquium Publications **53**, American Math. Soc., Providence, RI, 2004, Chapter 11.

[7] R.S. Lehman, *Factoring large integers*, Math. Comp. 28 (1974), 637–646.

[8] I. Shparlinski, *Distribution of points on modular hyperbolas*, Sailing on the Sea of Number Theory: Proc. 4th China-Japan Seminar on Number Theory, Weihai, 2006, World Scientific, 2007, 155–189.

[9] I. Shparlinski, *Modular hyperbolas*, `http : //arxiv.org/abs/1103.2879`.

[10] A. Weil, *On some exponential sums*, Proc. Nat. Acad. Sci. USA **34** (1948), 204–207.

[11] W. Zhang, *On the distribution of inverses modulo n*, Journal of Number theory **61** (1996), 301–310.