2009

Nontrivial Solutions to the Cubic Sieve Congruence Problem: $x^3 \equiv y^2 z \bmod p$

Maitra, Subhamoy; Rao, Subba, Y.V.; Stănică, Pantelimon; Gangopadhyay, Sugata

Maitra, S., Subba Rao, Y.V., Stanica, P. & Gangopadhyay, S. 2009, "Nontrivial solutions to the cubic sieve congruence problem x3=y2 z \pmod p", Special Issue on Applied Cryptography & Data Security in Journal of "Computacion y Sistemas" (eds. F. Rodriguez-Henriquez, D. Chakraborty), vol. 12, no. 3, pp. 253--266.

https://hdl.handle.net/10945/56988

Nontrivial Solutions to the Cubic Sieve Congruence Problem: $x^3 \equiv y^2 z \bmod p$

Soluciones no Triviales al Problema de Congruencia de Criba Cúbica: $x^3 \equiv y^2 z \bmod p$

Subhamoy Maitra[1], **Y. V. Subba Rao**[2], **Pantelimon Stanica**[3] **and Sugata Gangopadhyay**[4]

[1] Applied Statistics Unit, Indian Statistical Institute,
203, B T Road, Calcutta 700 108, INDIA
subho@isical.ac.in

[2] School of Maths. and Comp./Info. Sciences, University of Hyderabad,
P. O. Central University, Gachbowli, Hyderabad 500 046, INDIA
yvsrcs@uohyd.ernet.in

[3] Applied Mathematics Department
Graduate School of Engineering & Applied Sciences (GSEAS)
Naval Postgraduate School, Monterey, CA 93943, USA
pstanica@nps.edu

[4] Mathematics Department, Indian Institute of Technology, Roorkee,
Haridwar, Uttaranchal, INDIA
sugata70@rediffmail.com

Article received on March 1, 2008, accepted on June 14, 2008

Abstract
In this paper we discuss the problem of finding nontrivial solutions to the Cubic Sieve Congruence problem, that is, solutions of $x^3 \equiv y^2 z \pmod p$, where $x, y, z < p^{\frac{1}{2}}$ and $x^3 \neq y^2 z$. The solutions to this problem are useful in solving the Discrete Log Problem or factorization by index calculus method. Apart from the cryptographic interest, this problem is motivating by itself from a number theoretic point of view. Though we could not solve the problem completely, we could identify certain sub classes of primes where the problem can be solved in time polynomial in $\log p$. Further we could extend the idea of Reyneri's sieve and identify some cases in it where the problem can even be solved in constant time. Designers of cryptosystems should avoid all primes contained in our detected cases.
Keywords: Cubic Sieve Congruence, Discrete Log Problem, Prime Numbers.

Resumen
En este artículo se discute el problema de cómo encontrar soluciones no triviales al problema de congruencia de la criba cúbica, esto es, soluciones a la ecuación: $x^3 \equiv y^2 z \pmod p$, donde $x, y, z < p^{\frac{1}{2}}$ y $x^3 \neq y^2 z$. Las soluciones a este problema resultan útiles para resolver el problema del logaritmo discreto o el de factorización entera cuando se utiliza el método de *index calculus*. Además del evidente interés criptográfico, este problema tiene también relevancia desde el punto de vista de la teoría elemental de números. Aunque no logramos resolver totalmente el problema, sí pudimos identificar ciertas subclases de primos donde el problema puede ser resuelto en tiempo polinomial en $\log p$. Asimismo, extendimos la idea de cribado de Reyneri e identificamos algunas clases en donde el problema puede ser resuelto en tiempo constante. Los diseñadores de cripto-esquemas deben evitar utilizar cualquiera de los primos contenidos en los casos aquí detectados.
Palabras Claves: Congruencia de criba cúbica, problema del logaritmo discreto, números primos.

1 Introduction

Index calculus method (Menezes and Oorschot and Vanstone 1997; Coppersmith, Odlyzko and Schroeppel 1986; Das 1999; Das and Madhavan 2005) appears to be applicable in solving the Discrete Log Problem (DLP) (Menezes and

Oorschot and Vanstone 1997). One variant of this is the cubic sieve method (Coppersmith, Odlyzko and Schroeppel 1986; Lenstra and Lenstra 1990; Das 1999; Das and Madhavan 2005). In the cubic sieve method, one needs a 'known' solution (in positive integers) of the Diophantine equation

$$x^3 \equiv y^2 z \bmod p,$$

such that $x^3 \neq y^2 z$ with x, y, z of order p^α for some $\frac{1}{3} \leq \alpha < \frac{1}{2}$, where p is a prime number. We call this the Cubic Sieve Congruence (CSC) problem and x, y, z will be called a solution of CSC. We refer to (Das 1999, Section 3.2.3) for the logic behind the suggested range of α towards the solution of discrete log problem.

Though the problem was first presented back in mid eighties (Coppersmith, Odlyzko and Schroeppel 1986), to the best of our knowledge the next serious attempt to the problem was made in (Das 1999, Chapter 5) where heuristic estimates about the density of the solutions were studied in great details. We briefly present the results of (Das 1999, Chapter 5) in Section 2 with some more experimental evidence to support the conjectured claims of (Das 1999). However, no effort has yet been made to design a nontrivial algorithm for this problem and we attempt some solutions in Sections 3, 4. It has been stated in (Coppersmith, Odlyzko and Schroeppel 1986) that "We don't see any easy way to find such a triple in general" and in (Das 1999) that "in spite of all these theoretical and experimental exercises, the question of existence or otherwise a solution of the CSC for some $\frac{1}{3} \leq \alpha < \frac{1}{2}$ continues to remain unanswered".

It is well known that the "Number Field Sieve" (see (Lenstra and Lenstra 1993; Pomerance 1996)) is faster than the cubic sieve among index calculus type methods used in solving DLP. Let $L_p[v, c] = \exp((c+o(1))(\log p)^v(\log\log p)^{1-v})$. It is worth mentioning that once a solution of the cubic sieve is known, the running time of the cubic sieve discrete logarithm and factorization algorithm in $GF(p)$ is $L_p[\sqrt[3]{2/3}, 1/2] = \exp((0.816\ldots+o(1))(\log p \log\log p)^{1/2})$ (Coppersmith, Odlyzko and Schroeppel 1986). This could be potentially better than the Number Field Sieve, which has a running time of $L_p[1.923\ldots, 1/3]$. Thus it is important to answer where exactly the contribution of this work stands from a cryptographic point of view. We find polynomial and constant time algorithms (input size $\log p$, when p is the prime) to solve the CSC problem for different subclasses of primes. Though these subclasses are very small compared to the complete set of primes, the primes in these subclasses should not be chosen for any secure cryptosystem which is based on hardness of DLP as easy solution of CSC presents a potential weakness.

Further, this problem is interesting in itself from a number theoretic point of view. An easy attempt to solve CSC is to choose $x, y < p^{\frac{1}{2}}$ at random and then check whether $z < p^{\frac{1}{2}}$ too. As it will be clearer later in this paper, this random attempt is not going to succeed at all. Thus one needs to consider carefully designed methods to attack this problem.

We study this problem in parametric form $x = v^2 z \,\%p$ and $y = v^3 z \,\%p$. By $a \,\%b$ we mean the remainder when the integer a is divided by the integer b (the operator $\%p$ is always applied to the preceding expression, so $v^2 z \,\%p$ means $(v^2 z) \,\%p$). In Section 3, we show that it is possible to find a solution in time polynomial in $\log p$ (we denote this by $\mathcal{P}(\log p)$) if there exists a suitable $v > p^{0.25}$ having a value $p^{0.25} + O(\mathcal{P}(\log p))$. We show that this happens for approximately $\frac{N^{\frac{1}{4}}}{\log N}$ many primes $p \leq N$. In Section 4 we extend the idea of Reyneri's sieve and present precise solutions for CSC when the prime p satisfies $n^3 < lp < M < lp + p^\epsilon$, where $M = n^2(n+i)$, $i = 1, 2, 3$ or $(n+1)^3$, $0 < l < p^{0.5-3\epsilon} - p^{\epsilon-1}$ and $0 \leq \epsilon < \frac{1}{6}$. This idea works for approximately $\sum_{j=2}^{N^{\frac{1}{3}}} \frac{4}{3} \frac{j^{\frac{1}{2}}}{\log j}$ many primes $p \leq N$. The ideas used in this paper seem to be extendable for larger subclasses of primes and we are currently working in that direction.

2 Existing Results

We begin by introducing some notations as in (Das 1999). Fix a prime number p. Let

- $S = \{(x, y, z) \mid x^3 \equiv y^2 z \bmod p, 1 \leq x, y, z < p\}$
- $S_= = \{(x, y, z) \mid (x, y, z) \in S \text{ and } x^3 = y^2 z\}$

- $S_{\neq} = \{(x, y, z) \mid (x, y, z) \in S \text{ and } x^3 \neq y^2 z\}$
- $S_{\alpha} = \{(x, y, z) \in S_{\neq} \mid 1 \leq x, y, z < p^{\alpha}\}$

Throughout this paper, we use the Vinogradov symbols $\gg$, $\ll$ and the Landau symbols O, Θ and o with their usual meanings (see also (Das 1999; Coppersmith, Odlyzko and Schroeppel 1986; Menezes and Oorschot and Vanstone 1997) for details). We recall that $A \ll B,\ B \gg A$ and $A = O(B)$ are all equivalent and mean that $|A| < c|B|$ holds with some constant c, while $A = \Theta(B)$ means that both $A \ll B$ and $B \ll A$ hold. For a positive real number x we write $\log x$ for the maximum between 1 and the natural logarithm of x. We let $\lfloor x \rfloor$ be the largest integer $\leq x$, and let $\{x\} = x - \lfloor x \rfloor$ be the fractional part of $x > 0$.

It is clear that the CSC problem (see also (Das 1999, Chapter 5)), ignoring the bounds on x, y, z, has exactly $(p-1)^2$ number of solutions, since one can choose any x, y from $[1, p-1]$ and immediately z will be obtained. Thus, $\#S = (p-1)^2 = \Theta(p^2)$. Further it has been presented in (Das 1999, Chapter 5)) that $\#S_{=} \leq \frac{3}{2}(p-1)\ln(p-1) + (3\gamma - \frac{3}{2})(p-1) + O(\sqrt{p}) = O(p \ln p)$, and $\#S_{=} \geq \frac{3}{2}p + O(p^{\frac{2}{3}})$, that is, $\#S_{=} = \Omega(p)$.

Here γ is the Euler's constant defined as $\gamma = \lim_{n\to\infty}(1 + \frac{1}{2} + \ldots + \frac{1}{n} - \ln(n)) = 0.57721566\ldots$. Since S is the disjoint union of $S_{=}$ and $S_{\neq}$, from above one gets, $\#S_{\neq} \geq (p-1)^2 - \frac{3}{2}(p-1)\ln(p-1) + O(p)$, and so, $\#S_{\neq} \leq (p-1)^2 - \frac{3}{2}p + O(p^{\frac{2}{3}})$. In particular, $\#S_{\neq} = \Theta(p^2)$.

We are more interested in the value of $\#S_{\alpha}$, which is estimated by the following conjecture in (Das 1999, Chapter 5).

Conjecture 1 *The expected cardinality of S_{α} is asymptotically equal to $\chi p^{3\alpha-1}$ for all $0 \leq \alpha \leq 1$ and for some constant $\chi \approx 1$.*

Table 1. Primes 4268002919 (left) and 4213586771 (middle) and average values over 50 primes of 30-bit length (right)

α	# sol	$\frac{2}{3}p^{3\alpha-1}$	$p^{3\alpha-1}$
0.34	0	0	1
0.35	0	2	3
0.36	2	3	5
0.37	6	7	11
0.38	16	14	22
0.39	27	28	43
0.40	69	56	84
0.41	154	109	164
0.42	283	212	319
0.43	573	413	620
0.44	1135	804	1206
0.45	2223	1564	2347
0.46	4407	3043	4565
0.47	8639	5919	8879
0.48	16910	11513	17270
0.49	33179	22392	33589
0.50	65137	43552	65329

α	# sol	$\frac{2}{3}p^{3\alpha-1}$	$p^{3\alpha-1}$
0.34	0	0	1
0.35	2	2	3
0.36	4	3	5
0.37	5	7	11
0.38	13	14	22
0.39	27	28	43
0.40	54	56	84
0.41	126	108	163
0.42	257	211	317
0.43	547	412	618
0.44	1080	800	1201
0.45	2150	1557	2336
0.46	4235	3028	4543
0.47	8300	5888	8832
0.48	16427	11448	17172
0.49	32244	22258	33387
0.50	63262	43274	64911

α	Mean	Std.Dev
0.34	0.2800000	0.6074369
0.35	0.4400000	0.5115004
0.36	0.5340000	0.4082616
0.37	0.6622222	0.4120630
0.38	0.7054902	0.3139408
0.39	0.7988400	0.2547877
0.40	0.8296789	0.1910907
0.41	0.8618105	0.1410821
0.42	0.8903438	0.1060304
0.43	0.9261365	0.0804415
0.44	0.9389463	0.0643277
0.45	0.9533673	0.0441644
0.46	0.9686826	0.0338940
0.47	0.9745897	0.0261893
0.48	0.9799228	0.0207219
0.49	0.9840180	0.0138331
0.50	0.9883767	0.0111183

The conjecture is certainly believable, since if x, y are selected at random, then the probability that $z = x^3/z^2 \leq p^{\alpha}$ is expected to be p^{α}/p and so the size of S_{α} is about $p^{3\alpha-1}$. We also make a good number of experimental verifications with various sizes of primes ranging from 15 bits to 32 bits to support the above conjecture. In (Das

1999, Chapter 5), experimental results have been tabulated for the primes 32263723 (25 bits) and 1034302223 (30 bits). We tabulate in Table 1 experimental results for two 32-bit primes. In this first column we give the values of α. Second column contains the number of solutions with $x, y, z < p^\alpha$. Third column contains the value of $\lfloor \frac{2}{3} p^{3\alpha-1} \rfloor$ and fourth column contains the value of $\lfloor p^{3\alpha-1} \rfloor$. These results indicate that as α increases, the number of solutions get closer to $p^{3\alpha-1}$ and also for sufficiently large α depending on the size of prime (in case of 32-bit primes this α is 0.41), $\lfloor \frac{2}{3} p^{3\alpha-1} \rfloor$ gives a lower bound to the number of solutions.

To continue our verification, we calculate $\frac{\text{Number of solutions } <p^\alpha}{p^{3\alpha-1}}$ for α ranging from 0.34 to 0.50 for fifty randomly chosen primes of 30 bits. Then in Table 1 (rightmost) we have tabulated information as α in first column, the mean of fifty fractions for that α in second column. In the last column the standard deviation of the same values is given. Results here indicate that as α is increasing to 0.50, the mean is getting closer to 1.0 and standard deviation is getting closer to 0.0. This justifies Conjecture 1 further.

In (Coppersmith, Odlyzko and Schroeppel 1986, Page 13) it was noted that Reyneri's sieve applied to $p = x^3 - z$, with z small generates an easy solution having $y = 1$. So the idea is to take $x = \lceil \sqrt[3]{p} \rceil$, that is, the minimum x such that $x^3 > p$. If $x^3 - p < p^{0.5}$, then put $z = x^3 - p$ and $y = 1$. This gives a solution with $x, y, z < p^{0.5}$. However, getting such a solution is not possible in general. It may very well happen that the first x for which $x^3 > p$ is such that $x^3 - p \geq p^{0.5}$. As example, take $p = 125000003$. In that case, the first x such that $x^3 > p$ is $x = 501$. So $x^3 - p = 125751501 - 125000003 = 751498 > p^{\frac{2}{3}}$ and we can not get a solution according to our need, as for $y = 1$, $z = x^3 - p \geq p^{0.5}$. However, we note that there are many solutions with the constraint $x, y, z < p^{0.5}$ for this prime and one such example is $x = 56, y = 605, z = 1025$.

A simple algorithm to find a solution for any prime is as follows.

Algorithm 1

```
1.  for x = 1 to p^a, x = x + 1 {
2.    for y = 1 to p^b, y = y + 1 {
3.      calculate 0 < y_1 < p, such that y y_1 ≡ 1 mod p;
4.      calculate z = x^3 y_1^2 %p;
5.      if z < p^0.5 output solution (x, y, z);
6.    }
7.  }
```

Note that, by the previous analysis, it is clear that if we take $a = b = 0.35$, then it is expected to get a solution with $x, y, z < p^{0.35}$ for any large prime p. Further, step 3 of Algorithm 1 needs $O(\log p)$ time. Thus, the overall complexity becomes $O(p^{0.7} \log p)$. On the other hand, we have also experimentally observed that it is possible to get a solution with $y < p^{0.5}$ when x is very small compared to the large prime p. Considering this assumption and then letting $a = \epsilon$, a very small quantity and $b = 0.5$, it is expected to get a solution where $x, y, z < p^{0.5}$ with time complexity $O(p^{0.5+\epsilon} \log p)$. However, given a very large p, this algorithm is not a practical one.

3 Parametric form for CSC

To have a better understanding of the problem, we express it in parametric form. We rewrite the congruence in the form $\left(\frac{y}{x}\right)^2 \equiv \frac{x}{z} \pmod{p}$. That suggests the parametrization

$$x = v^2 z \,\%p \text{ and } y = v^3 z \,\%p \tag{1}$$

Note that in this parametric form the sets $S, S_{\neq}, S_\alpha$ (as defined in the previous section) can be rewritten as

- $S = \{(x, y, z) \mid x = v^2 z \,\%p, y = v^3 z \,\%p, 1 \leq x, y, z, v < p\}$,

- $S_{\neq} = \{(x, y, z) \mid x = v^2 z \,\%p, y = v^3 z \,\%p, 1 \leq x, y, z, v < p, x^3 \neq y^2 z\}$,
- $S_{\alpha} = \{(x, y, z) \mid x = v^2 z \,\%p, y = v^3 z \,\%p, 1 \leq x, y, z < p^{\alpha}, 1 \leq v < p, x^3 \neq y^2 z\}$.

However, the condition $x^3 \neq y^2 z$ in CSC needs to be tackled carefully in this parametric form. First we present a technical result.

Proposition 1 *If $(x, y, z) \in S_{0.5}$ satisfy (1), then $v > p^{0.25}$.*

Proof : Let $v \leq p^{0.25}$. Then $x = v^2 z \,\%p = v^2 z$ since $v^2 z < p^{2(0.25)+0.5} = p$ as $z < p^{.05}$. Also $y = vx \,\%p = vx$, since $x < p^{0.5}$ and $v \leq p^{0.25}$. Thus $x^3 = y^2 z$ which violates the requirement $x^3 \neq y^2 z$.

In the rest of the paper, we consider the specific constraint $p^{0.25} < v < p^{0.5}$. Further we need solutions of the form $x, y, z < p^{0.5}$. Under these constraints, $x^3 \neq y^2 z$ in CSC is equivalent to $x \neq v^2 z$ (see Proposition 2 below). This serves our purpose, since as presented in Proposition 1, we have $v > p^{0.25}$ for any solution with $x, y, z < p^{0.5}$ and further we concentrate on the cases when $v < p^{0.5}$ too.

Proposition 2 *Let $p^{0.25} < v < p^{0.5}, 1 \leq x, y, z < p^{0.5}$. Then the condition $x^3 \neq y^2 z$ is equivalent to $x \neq v^2 z$.*

Proof : Suppose x, y, z is a solution for CSC such that $v < p^{0.5}$ and $x^3 \neq y^2 z$. Since, $x, v < p^{0.5}$, so $y = vx < p$. Assume that $x = v^2 z = \frac{y^2}{x^2} z$. This implies that $x^3 = y^2 z$ which is a contradiction to $x^3 \neq y^2 z$. Thus we get $x \neq v^2 z$.

Conversely, let x, y, z, v be a solution to the system $x \equiv v^2 z \bmod p, y \equiv vx \bmod p, x \neq v^2 z$, with $p^{0.25} < v < p^{0.5}, 1 \leq x, y, z < p^{0.5}$. Then $y = vx$ and $x = v^2 z + lp$, with $l \neq 0$. So, $x = \frac{y^2}{x^2} z + lp$, which implies $x^3 = y^2 z + (lx^2)p$, that is, $x^3 \equiv y^2 z \bmod p$, but $x^3 \neq y^2 z$.

Thus, to find a solution for the CSC problem it suffices to find a solution to

$$x \equiv v^2 z \bmod p, y \equiv vx \bmod p, \text{ where } p^{0.25} < v < p^{0.5}, x \neq v^2 z, 1 \leq x, y, z < p^{\frac{1}{2}}. \tag{2}$$

It is clear that the set of these solutions is a subset of $S_{0.5}$. Further it should be noted that for these solutions, y is an exact integral multiple of x.

Definition 1 *We call a solution x, y, z of CSC as given in equation (2) a valid solution.*

Henceforth, we write $v = p^{\delta}$ and $z = p^{\beta}$ for δ, β real.

Conjecture 1 claims that there are approximately $\chi p^{3\alpha - 1}$ many solutions $(\chi \approx 1)$ where $x, y, z < p^{\alpha}$. For $\alpha = 0.5$, the number of solutions is approximately $p^{0.5}$. We randomly took 25 primes of length 30-bit and checked that for these solutions, when turned to parametric domain, the cases when $v < p^{0.5}$ is extremely low. The number of solutions for 30-bit primes is approximately 2^{15}. However, in Table 2 we observe that the number of solutions having $v < p^{0.5}$ is extremely low compared to 2^{15}. In the most favorable result, we get 19 solutions only for the prime 759828683. Also it should be noted that there are cases when there is no solution with $v < p^{0.5}$ as happened for the prime 741799451 (note that $x^3 + p$ has the required form, for $x = 731, 929, 3034, 6039$, however, y/x is not an integer). *Thus there are very few solutions, which, in the parametric form, give $x, y, z, v < p^{0.5}$. Still we attempt to find those solutions here as the range in which we need to vary v is much smaller than $O(p)$ and show that the analysis produces favorable results in certain cases.*

Lemma 1 *For any valid solution of CSC, if $v = p^{\delta} < p^{0.5}$ then $x < p^{0.5-\delta} < p^{0.25}$.*

Proof : Since $\delta < 0.5$ and for a valid solution $x < p^{0.5}$, the congruence $y \equiv vx \bmod p$ is an equality, that is, $y = vx$. From this we have $vx < p^{0.5}$, therefore $x < \frac{p^{0.5}}{v} = p^{0.5-\delta}$. From Proposition 1, $\delta > 0.25$, hence the result.

Table 2. Number of solutions with $x, y, z < p^{0.5}$ and $v < p^{\delta}$

δ Primes	$0 \leq \delta < .3$	$.3 \leq \delta < .35$	$.35 \leq \delta < .4$	$.4 \leq \delta < .45$	$.45 \leq \delta < .5$
895917131	2	0	0	0	0
593554447	0	0	0	1	1
551556059	0	0	2	0	0
774712823	0	0	1	1	0
961344259	0	1	2	1	0
1052502491	1	1	0	0	0
877166131	0	1	0	1	0
669150091	1	0	0	2	2
721235807	0	0	0	1	0
997165739	1	0	0	0	0
777782111	0	0	3	2	1
601873567	0	2	0	7	6
976974643	0	1	1	0	0
561998999	6	2	1	0	0
784308199	0	0	0	0	1
604718867	1	1	0	0	0
920692687	0	0	2	1	1
678600491	1	0	0	1	0
1066913867	0	1	0	1	0
741799451	0	0	0	0	0
1014893507	3	0	4	1	0
678813823	3	1	2	0	0
759828683	0	0	14	4	1
548375899	0	1	3	0	0
917289047	0	2	6	1	2

Lemma 2 *For a fixed $v = p^{\delta} < p^{0.5}$, that is part of a valid solution, we have $z > p^{1-2\delta}$.*

Proof : From the fact that $p^{0.25} < v < p^{0.5}$, we have $p^{0.5} < v^2 < p$. Now, if we assume that $z \leq p^{1-2\delta}$, then without taking modular operations $p^{0.5} < v^2 z = p^{2\delta} z \leq p^{2\delta} p^{1-2\delta} = p$. Therefore $x = v^2 z$ can not be less than $p^{0.5}$. This proves that $z > p^{1-2\delta}$.

Putting together Proposition 1, Lemma 1, 2, we obtain the following result.

Theorem 1 *Let there be a valid solution (recall that $x, y, z < p^{0.5}$, in that case) with $p^{0.25} < v = p^{\delta} < p^{0.5}$. Then $x < p^{0.5-\delta} \leq p^{0.25}$ and $z \geq p^{1-2\delta}$.*

In light of the above discussion, let us present the following result which will be used for the algorithms we discuss next.

Proposition 3 *For some v, z such that $p^{0.25} < v = p^{\delta} < p^{0.5}$ and $p^{1-2\delta} < z < p^{0.5}$, if there exists an $x < p^{0.5-\delta}$, then $y < p^{0.5}$, that is, we have a valid solution.*

As we have already mentioned, an important question at this point is: "is it guaranteed that for any prime p there will be a solution of the form $x, y, z, v < p^{0.5}$?" The answer is no, though for almost all the primes we have considered, it is possible to get such a solution. We have some experimental results for 25 primes in Table 2 where there is only one prime 741799451 for which there is no solution of the form $x, y, z, v < p^{0.5}$.

In this section we assume that the considered primes will have solutions of the form $x, y, z, v < p^{0.5}$ and present an algorithm based on that. The observation from Theorem 1 presents the basis of the algorithm we propose now. Here for each fixed $v = p^{\delta}$ in the range $p^{0.25}$ to $p^{0.5}$, we vary z in the range $p^{1-2\delta} = \frac{p}{v^2}$ to $p^{0.5}$ and compute x for each pair (v, z). Once the suitable x is found, with $x < p^{0.5-\delta}$, we output the solution.

Algorithm 2

```
1.  for v = p^0.25 to p^0.5, v = v + 1 {
2.    for z = p/v^2 to p^0.5, z = z + 1 {
3.      calculate x = v^2 z %p;
4.      if x < p^0.5 / v output the solution (x, y = vx, z);
5.    }
6.  }
7.  Output no solution with x, y, z, v < p^0.5;
```

If there is no solution $x, y, z, v < p^{0.5}$, our Algorithm 2 fails. However, that is not the case in general. Note that in the worst case, the time complexity of Algorithm 2 is $O(p)$, which is worse than the trivial Algorithm 1. However, it should be noted that Algorithm 2 is extremely efficient when there is a solution where v is close to $p^{0.25}$. Before proceeding further, let us present some nontrivial improvement over Algorithm 2.

From Theorem 1, we can see that for fixed v, smallest z that can be considered is $\lceil p^{1-2\delta} \rceil$. We represent this as z_1 and also write $z_1 = p^{\beta_1}$ for some real $\beta_1 < 0.5$. For this z_1, we have

$$v^2 z_1 = p^{2\delta+\beta_1} = p + k_1, \tag{3}$$

for some $0 \leq k_1 < p$. Now we have two possible cases:
Case 1: $k_1 < p^{0.5-\delta}$. In this case our problem is solved by letting $x = k_1$. Because, from our earlier analysis we know that if $v, z < p^{0.5}$ and $x < p^{0.5-\delta}$, then we can have a solution just by taking $y = vx$.
Case 2: $k_1 \geq p^{0.5-\delta}$. In this case we may try for the 'next suitable' z in increasing order. Let that be $z_2 = p^{\beta_2}$ of the form $z_2 = z_1 + t_1$. Also, we need z_2 to be such that

$$v^2 z_2 = p^{2\delta+\beta_2} = 2p + k_2, \text{ and } v^2(z_2 - 1) < 2p, \tag{4}$$

for some $0 \leq k_2 < p$. This is because, if we take any other z_2', such that $z_1 < z_2' < z_2$, then $p + k_1 < v^2 z_2' = p + k_2' < 2p$ and hence $k_1 < k_2' < p$. Thus if $x = k_1$ is not a valid solution, $x = k_2'$ can not be a valid solution, as well. So we consider, $v^2 z_2 = 2p + k_2$ which gives $v^2(z_1 + t_1) = 2p + k_2$. This gives us $v^2 t_1 = 2p + k_2 - v^2 z_1 = 2p + k_2 - (p + k_1) = (p - k_1) + k_2$, and so, $t_1 = \frac{(p-k_1)+k_2}{v^2}$. Since our aim is to minimize k_2, we can take $t_1 = \lceil \frac{(p-k_1)}{v^2} \rceil$. Again, as above, we have two cases.
Case 2a: $k_2 < p^{0.5-\delta}$, which leads to a solution.
Case 2b: $k_2 \geq p^{0.5-\delta}$, we can continue to the next z, say $z_3 = z_2 + t_2$ where $t_2 = \lceil \frac{(p-k_2)}{v^2} \rceil$.

We can repeat this process until it terminates by giving us a 'valid' solution or it reaches a stage where $z_r \geq p^{0.5}$ in some r^{th} cycle. Then we can restart with $v = v + 1$ till $v < p^{0.5}$. Based on this we present the following algorithm.

Algorithm 3

I	$Min = \lceil p^{0.25} \rceil$;
II	$Max = \lfloor p^{0.5} \rfloor$;
III	Start with $v = Min$;
IV	while($v \leq Max$){
IVa	$z = \lceil \frac{p}{v^2} \rceil$;
IVb	$k = v^2 z \% p$;
IVc	if $(k < \lfloor \frac{Max}{v} \rfloor)$
	Output solution as $(x = k, y = kv, z, v)$ and terminate;
IVd	$t = \lceil \frac{p-k}{v^2} \rceil$;
IVe	$z = z + t$;
IVf	While $(z \leq Max)$ {
	$k = v^2 z \% p$;
	if $(k < \lfloor \frac{Max}{v} \rfloor)$
	Output solution as $(x = k, y = kv, z, v)$ and terminate;
	$t = \lceil \frac{p-k}{v^2} \rceil$;
	$z = z + t$;
	}
IVg	$v = v + 1$;
	}
V	Output no solution with $x, y, z, v \leq \lfloor p^{0.5} \rfloor$;

In Algorithm 3 we increase z by a step of t instead of 1, as was done in Algorithm 2. This gives the improvement. However, as v becomes larger the worst case complexity of Algorithm 3 becomes $O(p)$, which is again theoretically worse than the trivial method described in Algorithm 1. On the other hand, it is important to note that Algorithm 3 is much more efficient than Algorithm 1 when there is a solution where v is close to $p^{0.25}$. We shall now use Algorithm 3 for a few arbitrary primes, which are hard to solve using Algorithm 1. Note that the last but one row in Table 3 contains a 77-bit prime and the last row contains a 98-bit prime. We run Algorithm 3 implemented using C programming language and GMP (GNU Multi Precision) facility. The operating system is Redhat Linux 8.0 and the machine contains Pentium IV processor with 1 GByte RAM. It took approximately 20 minutes to have a solution for the 77-bit prime and 5 minutes for the 98-bit one. If one uses Algorithm 1, it seems very hard to find solutions in these cases with present day machines. As in Table 2, all the primes presented in Table 3 are selected at random. We have chosen five 77-bit primes and obtained a solution every time within half an hour. For 98-bit, we have taken two randomly chosen primes, out of which one is in Table 3, the other one has not given any solution in 3 hours.

Table 3. Experimental Results running Algorithm 3

p	$p^{0.25}$	$p^{0.5}$	v	x	y	z
145678132176163	3475	12069719	27009	17	459153	9785284
145678132176162513743	109863	12069719639	115472	18609	2148818448	10925491628
23456543676548754325781	391351	153155292682	1440247	48034	69180824398	147005442243
66666555558888899999267	508133	258198674587	11225651	16104	180777883704	117974951645
16544909312689742347064453653 7	20168152	406754340022202	52165306	5171691	269782843552446	303998105265466

Theorem 2 *Assume that for a prime p, there exists a valid solution (recall Definition 1 and equation (2)) with $v = \Theta(p^{0.25+\epsilon})$. Then Algorithm 3 requires $\Theta(p^{0.25+3\epsilon})$ time complexity.*

Proof : We assume $p-k$ is $\Theta(p)$. If v is $\Theta(p^{0.25+\epsilon})$, then t is $\Theta(\frac{p}{p^{0.50+2\epsilon}})$, that is, $\Theta(p^{0.50-2\epsilon})$. So z takes $\Theta(\frac{p^{0.50}}{p^{0.50-2\epsilon}})$, which is, $\Theta(p^{2\epsilon})$ steps for each v. Hence the total time complexity is $\Theta(p^{0.25+3\epsilon})$.

From Table 2, we see that there are solutions for $\delta < 0.3$ for 9 primes out of 25 and the time complexity is $O(p^{0.4})$ in these cases. It should also be noted that this method is extremely effective when v is $\Theta(p^{0.25})$.

Now let us see under what conditions Algorithm 3 works in time $O(\mathcal{P}(\log p))$, that is, in time polynomial in $\log p$. This directly follows from the proof of Theorem 2.

Corollary 1 *Assume that for a given prime p, there is a solution $x, y, z < p^{0.5}$ (as in (2)) with $v = p^{0.25} + O(\mathcal{P}(\log p))$. Then Algorithm 3 runs in $O(\mathcal{P}(\log p))$ time.*

Proof : If $v = p^{0.25} + O(\mathcal{P}(\log p))$, then t is $\Theta(\frac{p}{(p^{0.25}+O(\mathcal{P}(\log p)))^2})$. Now z takes $\Theta(\frac{p^{0.5}}{t})$ steps, and considering $\frac{\mathcal{P}(\log p)}{p^{0.25}}$ is negligible, one can assume that z takes constant number of steps for each v. This gives the proof.

Algorithm 3 uses a suitable gap in z for a fixed v. In a similar way one can try to work with a suitable gap in v for a fixed z. However, we believe a much better improvement could be achieved by finding a 'better' (v_1, z_1) pair for given (v_0, z_0) pair. Here by 'better' we aim at having $k_1 < k_0 < p$, where $v_1^2 z_1 = l_1 p + k_1$ and $v_0^2 z_0 = l_0 p + k_0$. A strategy in this direction may improve Algorithm 3 further.

Now one important question is what proportion of primes will have a solution as mentioned in Corollary 1. This is not clear at this point and needs further investigation.

It should be noted that the primes in Table 3 are selected at random. However, it is possible to identify very large primes for which Algorithm 3 will give a solution very fast. We first decide on a bound for p, say N, and then select any v of $O(N^{0.25})$. Now choose a prime p which lies between $(v-1)^2v^2 - v + 1 < p < (v-1)^2v^2$. Thus v is $\Theta(p^{0.25})$. Take $z = (v-1)^2$ and note that $z < p^{0.5}$. It is easy to see that $x, y < p^{0.5}$.

As an example we present an 160 digit prime $p =$ 176137087374777815393637069
274127644687309130845043890914502471120716308007100351639864691570824
459843834241066823375464624824608726598154401499019151812451283 9. Note that
$\lceil p^{0.25} \rceil =$ 64783245678901234567437892136453865642 73,
$\lfloor p^{0.5} \rfloor =$ 41968689206928754804764822743102551198400852553442630150 37
855720242846177345425 5, $v =$ 64783245678901234567437892136453865642 73,
$x =$ 697, $y =$ 45153922238194160493504210819108344352982 81, and
$z =$ 41968689206928754804 76 48227431025511983943742288747400269218134 13
214816386889984.

Proposition 4 *Consider a prime p such that $(v-1)^2v^2 - v + 1 < p < (v-1)^2v^2$. Then we get a valid solution of (2) for $z = (v-1)^2$.*

Proof : Since $(v-1)^4 < (v-1)^2v^2 - v + 1 < p$, we get $z = (v-1)^2 < p^{0.5}$. Now $x = v^2 z \,\%p = v^2(v-1)^2 \,\%p$. This gives, $x \leq v - 2 < p^{0.25}$. Hence, $y = vx = v(v-2) < (v-1)^2 < p^{0.5}$.

The Prime Number Theorem (see reference (Menezes and Oorschot and Vanstone 1997)) states that there are approximately $\frac{N}{\log N}$ many primes less than or equal to N. Proposition 4 implies that, for approximately $\frac{(v-1)^2v^2}{\log((v-1)^2v^2)} - \frac{(v-1)^2v^2-v+1}{\log((v-1)^2v^2-v+1)} \approx \frac{v}{\log v^4} \approx \frac{N^{\frac{1}{4}}}{\log N}$ many primes less than N, one can get a fast solution to CSC using Algorithm 3. Thus we have the following result from the above discussion and Corollary 1.

Corollary 2 *There are approximately $\frac{N^{\frac{1}{4}}}{\log N}$ many primes $p \leq N$ for which we get a valid solution of CSC in $O(\mathcal{P}(\log p))$ time using Algorithm 3.*

4 Further extension with respect to Reyneri's sieve

We have already discussed an application of Reyneri's sieve to CSC in Section 2. Here we use an extension of that idea to get fast solutions of CSC for certain kind of primes.

Let p be a given prime then take $n = \lfloor p^{\frac{1}{3}} \rfloor$. So, we have $n^3 < p < (n+1)^3$. Now let $k = (n+1)^3 - p$. If $k < \frac{p^{0.5}}{n+1}$, by letting $v = n+1$ and $z = n+1$, we have the required solution as seen earlier. One can also consider the cases when $n^3 < p < n^2(n+i)$ for $i = 1, 2, 3$. Consider that some particular a^2b satisfies $a^2b > p$ and $k = a^2b - p < \frac{p^{0.5}}{a}$. Then we have a solution by taking $v = a$ and $z = b$. Now we look into this idea more carefully.

Theorem 3 *Given a prime p, assume that there exists l and i such that for $n = \lfloor \sqrt[3]{lp} \rfloor$ we have*

(i) $n^3 < lp < (n+1)^3 < lp + p^\epsilon$, *or*

(ii) $n^3 < lp < n^2(n+i) < lp + p^\epsilon$, *where $i = 1, 2$, or 3 and $i \leq p^{0.5} - p^{0.5-\epsilon}$,*

where $0 < l < p^{0.5-3\epsilon} - p^{\epsilon-1}$. Then there is a valid solution of (2) with

(i) $v = z = n+1$,

(ii) $v = n, z = n+i$,

respectively. Further $l > 0$ implies $0 < \epsilon < \frac{1}{6}$.

Proof : First we prove (i). Take $v = z = n+1$. Then $lp < v^2 z = (n+1)^3 < lp + p^\epsilon$. Thus, $x \equiv v^2 z \bmod p$, $x < p^\epsilon$. Now $y = vx < (n+1)p^\epsilon < (\sqrt[3]{lp + p^\epsilon})p^\epsilon < (\sqrt[3]{(p^{0.5-3\epsilon})p + p^\epsilon})p^\epsilon = (\sqrt[3]{p^{1.5-3\epsilon} - p^\epsilon + p^\epsilon})p^\epsilon = p^{0.5-\epsilon}p^\epsilon = p^{0.5}$. Similarly, $z = n + 1 < p^{0.5-\epsilon}$.

Now we prove (ii). In this case, $n^3 < lp < n^2(n+i) < lp + p^\epsilon$, $i = 1, 2, 3$. Take $v = n$, $z = n + i$. Then we obtain $v^2 z = n^2 z = n^2(n+i) < lp + p^\epsilon$. Since, $x \equiv v^2 z \bmod p$, $x < p^\epsilon$. Further $y = vx < np^\epsilon < (lp + p^\epsilon)^{1/3} p^\epsilon \leq ((p^{0.5-3\epsilon} - p^{\epsilon-1})p + p^\epsilon)^{1/3} p^\epsilon = (p^{1.5-3\epsilon} - p^\epsilon + p^\epsilon)^{1/3} p^\epsilon = (p^{1.5-3\epsilon})^{1/3} p^\epsilon = p^{0.5-\epsilon} p^\epsilon = p^{0.5}$. Lastly, we have to show that $z < p^{0.5}$ given that $z = n + i$. Since $n^3 < lp$, we have $n < (lp)^{1/3} < ((p^{0.5-3\epsilon} - p^{\epsilon-1})p)^{1/3} = (p^{1.5-3\epsilon} - p^\epsilon)^{1/3} < p^{0.5-\epsilon}$. So, $n + i < p^{0.5-\epsilon} + i \leq p^{0.5}$, if $i \leq p^{0.5} - p^{0.5-\epsilon}$.

Based on Theorem 3, we present Algorithm 4. Before stating the step by step algorithm, we discuss the following few issues. Let us consider a prime p and some l. It is clear that we can immediately calculate $n = \lfloor \sqrt[3]{lp} \rfloor$. Now to get a solution using Theorem 3, one needs $lp + p^\epsilon > M$, where $M = n^2(n+i)$, $i = 1, 2, 3$ or $M = (n+1)^3$. Thus lp must be greater than $M - p^\epsilon$. That is why the requirement is $M - p^\epsilon < lp < M$.

Now we need to check whether there exists any l for which this is possible. So we calculate $l = \lfloor \frac{M}{p} \rfloor$, and so, $lp < M < (l+1)p$. Given this l, we calculate the maximum ϵ in the range $0 < \epsilon < \frac{1}{6}$ such that $l < p^{0.5-3\epsilon} - p^{\epsilon-1}$. There are various ways to calculate such an ϵ. For instance, labeling $A = \sqrt{p}, X = p^\epsilon$, we can solve for X satisfying the inequality $A^3X^4 - lA^2X - 1 > 0$. (We can also use the next alternative approach: since $\epsilon - 1 < 0$ and $0.5 - 3\epsilon > 0$, then the term $p^{0.5-3\epsilon}$ will dominate $p^{\epsilon-1}$ and so, for p sufficiently large, we can only solve the inequality $l < p^{0.5-3\epsilon}$, instead, which will give $p^\epsilon = \sqrt[3]{p^{0.5}/l}$.) For that maximum ϵ, if $lp + p^\epsilon$ becomes greater than M, then we get a valid solution. Thus, we do not need to check all integer l in the range $0 < l < p^{0.5-3\epsilon} - p^{\epsilon-1}$, but we can only check the values of l as $l = \lfloor \frac{n^2(n+i)}{p} \rfloor$, for $i = 1, 2, 3$ and $l = \lfloor \frac{(n+1)^3}{p} \rfloor$ in the prescribed range. Also it is clear that as we increase l, the value of ϵ becomes smaller. Thus the expectation of getting a solution decreases as l is increased. Based on this we present the following algorithm.

Algorithm 4

I	$n = \lfloor p^{1/3} \rfloor$; $v = \lfloor p^{1/2} \rfloor$;
II	$l = 1$; $M_1 = n^2(n+1)$; $M_2 = n^2(n+2)$; $M_3 = n^2(n+3)$; $M_4 = n^3$;
III	while($l \leq v$){
IIIa	$z_1 = n+1$; $z_2 = n+2$; $z_3 = n+3$; $z_4 = n$;
IIIb	for ($i = 1, 2, 3, 4$){
IIIb(i)	$l = \lfloor \frac{M_i}{p} \rfloor$;
IIIb(ii)	Calculate ϵ such that $l = \lfloor p^{0.5-3\epsilon} - p^{\epsilon-1} \rfloor$; $g = \lfloor p^{\epsilon} \rfloor$;
IIIb(iii)	if $(M_i - lp) < g$
	report $v = n, z = z_i, x = v^2 z \,\%p, y = v^3 z \,\%p$ and terminate;
IIIc	}
IIId	$n = n + 1$;
IV	}
V	Report no solution of this form.

Now it is important to analyze what proportion of primes are covered by Algorithm 4. We only take the case when $l = 1$ which gives a lower bound on the number of primes that are being covered by this algorithm and the algorithm will stop just after the first iteration. That is, for these primes, we have a constant time algorithm. For $l = 1$, $\epsilon = \frac{1}{6}$. Thus if we have $M - p^{\frac{1}{6}} < p < M$, then there is a valid solution of CSC for the prime p. We can take $p \approx n^3$. The range between n^3 and $(n+1)^3$ is $3n^2 + 3n + 1$. In this range p can have the value in the range $M - p^{\frac{1}{6}} < p < M$, where $M = n^2(n+i)$, $i = 1, 2, 3$ or $M = (n+1)^3$ to have a solution by Algorithm 4 in one step. Thus there are 4 different regions, each of length $p^{\frac{1}{6}}$, where we get a one step solution using Algorithm 4. Thus in the range of $3n^2 + 3n + 1$ integers, we are interested in the 4 intervals containing $4p^{\frac{1}{6}} \approx 4n^{\frac{1}{2}}$ many integers in total. Now we can approximate the number of primes in these intervals by $\sum_{i=1}^{4} \left(\frac{M_i}{\log M_i} - \frac{M_i - n^{\frac{1}{2}}}{\log(M_i - n^{\frac{1}{2}})} \right)$, where the M_i's are as described in step II of Algorithm 4. Taking $N \approx n^3 \approx M_i$, we can approximate this by $4 \left(\frac{N}{\log N} - \frac{N - n^{\frac{1}{2}}}{\log(N - n^{\frac{1}{2}})} \right) \approx 4 \left(\frac{N}{\log N} - \frac{N - n^{\frac{1}{2}}}{\log N} \right) \approx 4 \frac{n^{\frac{1}{2}}}{\log N} \approx 4 \frac{n^{\frac{1}{2}}}{\log (n+1)^3} \approx \frac{4}{3} \frac{n^{\frac{1}{2}}}{\log n}$.

Similarly one can look at the interval between $(n-1)^3$ and n^3. Thus one can approximate the total number of such primes up to $(n+1)^3$ by $\sum_{j=2}^{n} \frac{4}{3} \frac{j^{\frac{1}{2}}}{\log j} \approx \sum_{j=2}^{N^{\frac{1}{3}}} \frac{4}{3} \frac{j^{\frac{1}{2}}}{\log j}$. We summarize the previous analysis in the following corollary.

Corollary 3 *There are approximately* $\sum_{j=2}^{N^{\frac{1}{3}}} \frac{4}{3} \frac{j^{\frac{1}{2}}}{\log j}$ *many primes* $p \leq N$ *for which we get a valid solution of CSC in one step by using Algorithm* 4.

To further motivate our sieving approach, we now attempt to find some necessary conditions on primes p which fail Reyneri's sieve, but pass ours. From its construction, a prime p will pass Reyneri's sieve when $x^3 - p < p^{\frac{1}{2}}$, where $x = \lceil \sqrt[3]{p} \rceil$. On the other hand, a prime p will pass our sieve if there is some l, satisfying the conditions of Theorem 3.

We first discuss the case with $l = 1$. Given some n, we concentrate on the interval of integers from n^3 to $(n+1)^3$. Take the cases when (1) $(n+1)^3 - p^{\frac{1}{6}} < p < (n+1)^3$ or (2) $n^2(n+3) - p^{\frac{1}{6}} < p < n^2(n+3)$. In these two cases, considering $n \approx p^{\frac{1}{3}}$, one can see the following solution using Reyneri's sieve. Take $x = \lceil p^{\frac{1}{3}} \rceil$, $z = x^3 - p$ and $y = 1$. In these two cases, $x^3 = (n+1)^3$ and hence $z = x^3 - p < x^3 - n^2(n+3) + p^{\frac{1}{6}} = 3n + 1 + p^{\frac{1}{6}} < p^{\frac{1}{2}}$. Thus one can get a solution with $x, y, z < p^{\frac{1}{2}}$. However, note that the solutions we get using Algorithm 4 are different from the ones using Reyneri's sieve, since y cannot be 1 in our cases, as $y > x$, in fact a multiple of x.

Now consider the other two cases when (3) $n^2(n+2)-p^{\frac{1}{6}} < p < n^2(n+2)$ or (4) $n^2(n+1)-p^{\frac{1}{6}} < p < n^2(n+1)$. In these two cases, $z = x^3 - p > x^3 - n^2(n+2) = 3n^2 + 3n + 1 > p^{\frac{1}{2}}$. Thus these primes have solution for CSC with our sieving method, but not by Reyneri's sieving.

As an experimental result, we tried with $n = 100000$ and found 16 primes as in the cases (1), (2) which pass Reyneri's sieve and 18 primes as in the cases (3), (4) which do not pass Reyneri's sieve.

The cases considering $l > 1$ are not simple to analyze and need further investigation. However, we have experimented with a few cases and the results show that the primes do not pass the Reyneri's sieve. As example, we tried with $n = 100000$. For $2 \leq l \leq 9$, we got the solutions for 30 primes according to Theorem 3 and none of them can be approached by Reyneri's sieve.

Now we extend slightly the notion of valid solution to CSC to include all solutions satisfying $x, y, z = O(p^{\frac{1}{2}})$ (in our previous definition the constant understood was 1).

Theorem 4 *Let p be a prime. Assume that there exist integers a, b with $c_1 p^{\frac{1}{3}} \leq a \leq c_2 p^{0.5-\epsilon}$ (for some fixed constants $c_1 \geq c_2$; due to the reason $c_1 p^{\frac{1}{3}} < c_2 p^{\frac{1}{2}-\epsilon}$, $0 < \epsilon < \frac{1}{6} - \log_p(\frac{c_1}{c_2})$) and $b > \frac{lp}{a^2}$ such that $lp < a^2 b < lp + p^\epsilon$, for some $1 \leq l \leq c_3 p^{\frac{1}{6}}$. Then there is a valid solution of CSC with $v = a$, $z = b$.*

Proof : Take $v = a$, $z = b$. It can be checked that $x^3 = y^2 z \bmod p$ and $x^3 \neq y^2 z$. Since $lp < a^2 b < lp + p^\epsilon$ and $x \equiv a^2 b \bmod p$, it follows that $x = a^2 b \,\%p < p^\epsilon < p^{\frac{1}{6}}$. Similarly, using $alp < a^3 b < alp + ap^\epsilon$ and $y \equiv a^3 b \bmod p$, we gather that $y = a^3 b \,\%p < ap^\epsilon < c_2 p^{\frac{1}{2}-\epsilon} p^\epsilon = c_2 p^{\frac{1}{2}-\epsilon}$. Furthermore, $z = b < \frac{lp}{a^2} + \frac{p^\epsilon}{a^2} < \frac{lp}{c_1^2 p^{\frac{2}{3}}} + \frac{p^\epsilon}{c_1^2 p^{\frac{2}{3}}} < \frac{c_3}{c_1^2} p^{\frac{1}{2}} + 1 < \left(\frac{c_3}{c_1^2} + 1\right) p^{\frac{1}{2}}$. Therefore, x, y, z are all $O(p^{\frac{1}{2}})$ and they are solutions to CSC.

Clearly the result of Theorem 4 covers a lot more primes than Theorem 3. However, it is not clear how to write an algorithm to get l very fast when the results of Theorem 3 or Theorem 4 are applied. Algorithm 4 works efficiently (in fact in constant time) when one gets a solution for low values of l (bounded by a constant), however as l increases, the complexity of the algorithm increases.

5 Conclusion

In this paper we identify some subsets of the set of primes where the Cubic Sieve Congruence problem can be solved very fast. The solutions to this problem help in solving the Discrete Log Problem (DLP) by index calculus method. Thus we could identify some subclasses of primes which should not be used in the design of cryptosystems where the hardness of DLP provides the security. Apart from a cryptographic interest, this problem is motivating by itself from a number theoretic point of view. We could only provide partial solutions to this problem. Solving it completely seems to be an extremely challenging task. Thus, getting some more partial solutions to this problem presents an important research direction.

References

D. Coppersmith, A. Odlyzko and R. Schroeppel. Discrete logarithms in GF(p). Algorithmica, 1:1–15, 1986.

A. Das. Galois Field Computations: Implementation of a library and a study of the discrete logarithm problem. *Ph. D. Thesis*, Indian Institute of Science, Bangalore, India, 1999. Available at: http://www.cse-web.iitkgp.ernet.in/ abhij/download/doc/thesis.ps.gz [last accessed June 16, 2008]

A. Das and C. E. Veni Madhavan. On the cubic sieve method for computing discrete logarithms over prime fields. International Journal of Computer Mathematics, 82(12):1481–1495, December, 2005.

A. K. Lenstra and H. W. Lenstra. Algorithms in Number Theory. In *Handbook of Theoretical Computer Science*, pages 675–715, 1990.

A. K. Lenstra and H. W. Lenstra. The Development of the Number Field Sieve. Berlin: Springer-Verlag, 1993.

A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.

C. Pomerance. A Tale of Two Sieves. Notices Amer. Math. Soc., 43:1473–1485, 1996.

Subhamoy Maitra *received his Bachelor of Electronics and Telecommunication Engineering degree in the year 1992 from Jadavpur University, Kolkata and Master of Technology in Computer Science in the year 1996 from Indian Statistical Institute, Kolkata. He has completed Ph.D. from Indian Statistical Institute in 2001. Currently he is an Associate Professor at Indian Statistical Institute. His research interest is in Cryptology.*

Subba Rao Y. V. *is working as faculty since 2004, at Department of Computer and Information Sciences, University of Hyderabad, India. He completed Master of Technology in Computer Science in the year 2003 from Indian Statistical Institute, Kolkata. His research interests are in Cryptology and Theory of Computer science.*

Pantelimon Stanica *obtained his Ph.D. from State University of NewYork at Buffalo and the Institute of Mathematics of the Romanian Academy in 1998. He is currently with the Applied Mathematics Department at Naval Postgraduate School, in Monterey, CA, USA. His main interests are in cryptographic Boolean functions, combinatorics, discrete mathematics, and number theory.*

Sugata Gangopadhyay *received his B.Sc. degree in Mathematics from the University of Calcutta in 1991, M.Sc. and Ph.D. degrees in Mathematics from the Indian Institute of Technology Kharagpur in 1993 and 1998. He is a faculty member at the Indian Institute of Technology Roorkee. His research interest is in Boolean functions and Cryptology.*