Journal of Computer Science 5 (9): 674-679, 2009
ISSN 1549-3636

Integer Factorization: Solution via Algorithm for Constrained Discrete Logarithm Problem

Boris S. Verkhovsky
Department of Computer Science,
New Jersey Institute of Technology, Newark, NJ, USA

Abstract: Problem statement: During the last thirty years many public-key cryptographic protocols based on either the complexity of integer factorization of large semiprimes or the Discrete Logarithm Problem (DLP) have been developed. **Approach:** Although several factorization algorithms with sub-exponential complexity have been discovered, the recent RSA Factoring Challenge demonstrated that it was still necessary to use several thousand computers working in a coordinated effort for several months to factor an integer n that was a product of two primes. **Results:** In this research it was demonstrated how to find integer factors of n using an algorithm for a constrained DLP. Several numerical examples illustrate details of the algorithms. One of these algorithms has $O(\sqrt[3]{n})$ complexity and, if the search is balanced, it has complexity $O(n^{1/3}\log^{1/\alpha} n)$, where *alpha*>1.

Key words: Balanced search, subexponential complexity, integer factorization, constrained discrete logarithm problem, RSA factoring challenge, public key cryptography

INTRODUCTION

Attempts to find efficient algorithms for integer factorization of a semiprime $n = pq$ have a long history. Pierre Fermat[9], Leonhard Euler[8] and other great mathematicians of the past suggested various algorithms. Unfortunately, the complexities of their algorithms do not allow for efficient factoring of semiprimes with hundreds decimal digits. During the last twenty five years various factorization algorithms were discovered[6,7,10,11,13,16,20-22]. Several of these algorithms have a sub-exponential complexity[7,11,13]. Yet, the recent RSA Factoring Challenge[12] showed that it required the coordinated efforts of many researchers, using several thousand computers for many months, to factor a single semiprime. The study[1] presented a non-deterministic polynomial-time algorithm which shows that for factoring n it is sufficient to compute discrete logarithms modulo n. The research[4] shows that a xedni-calculus attack on the DLP for elliptic curves[14] can also be used to factor integers. A deterministic algorithm for factoring of semiprimes is provided in this research.

MATERIALS AND METHODS

Reduction of factorization to constrained DLP: Now we define the integer factorization problem and propose the discrete logarithm problem as a method of its solution.

Factorization: Suppose p and q are unknown distinct primes and

$$n = pq \tag{1}$$

If the product n is known, the problem is to determine the primes p and q.

Definition of constrained DLP: Let g, h, n, E and T be known integers that satisfy the equation

$$g^x \bmod n = h \tag{2}$$

where $E<x<T<n$ is an unknown integer. In this case the problem is how to find max$x$, if x exists.

Multiplicative inverse modulo *n*: If g and n are co-prime, then there exists a unique integer $0<b<n$ such that

$$gb \bmod n = 1 \tag{3}$$

The algorithm: Let's assume that there exists an algorithm A which efficiently solves the Eq. 2.

Step 1: Using the Extended Euclid Algorithm[5] or the algorithm[15], proposed by the author of this study, find the multiplicative inverse b of g modulo n (3)

Step 2: Using the algorithm A, solve the DLP[23-28]:

$g^{v} \bmod n = b$ (4)

where b satisfies (3)

Step 3: Let $h := n-$ (5)

Step 4: Solve the quadratic equation:

$z^2 - hz + n = 0$ (6)

then

$p := z_1; \quad q := z_2$ (7)

Modular Multiplicative Inverse (MMI): The algorithm for the MMI consists of two stages: Down-stage and Up-stage.

Step 5: *count*:=0; *T:=n*; *B:=b*

Step 6: {*Down-stage*}: *count*:=*count*+1 (8)

$H := T \bmod B; F := (T - H) / B$ (9)

store all values of *F* in a *stack*;

Step 7: If *H*=0, then the MMI inverse does not exist; {as a result, *F*=gcd(*n, b*)};

while *H*>1, re-assign *T*:=*B*; *B*:=*H*; (10)

repeat **Step 6**

countess:=*count*; (11)

Initialize *T*:=0; *B*:=1 (12)

Step 8:{*Up-stage*}: *count*:=*count*-1 (13)

pop up *F* from the *stack*;

H: =*BF*+*T* (14)

Step 9: while *count*>1, re-assign *T*:=*B*; *B*:=*H*;

repeat **Step 8**;

if *count*=0 and *countess* is *odd*,

then *MMI*:=*H* else *MMI*:=*n-H* (15)

Algorithm validation: Let:

$B := \varphi(n) = (p-1)(q-1)$ (16)

Euler's theorem[3] implies that:

$b = g^{n-p-q} \bmod n$ (17)

Indeed:

$bg = g^{n-p-q} g = g^{(p-1)(q-1)} \pmod{pq} = 1$ (18)

Therefore, $v = n-p-q$.

Thus, Eq. 7 can be re-written as:

$z^2 - (p+q) z + pq = 0$ (19)

Finally, Viète's theorem[17] implies the validity of (8).

Remark 1: (17) implies that the solution of equation (4) always exists.

Q.E.D.

Let's illustrate the algorithm.

RESULTS AND DISCUSSION

Numeric illustration: Let n = 97965643. Select an integer g = 22.

The multiplicative inverse of g modulo n equals b = 40076854, (Table1).

Verification shows that b = 40076854 is indeed the multiplicative inverse of g = 22:

$$\text{namely, } 22 \times 40076854 \bmod 97965643 = 1$$

Applying algorithm A to solve the DLP:

$$22^{v} \bmod 97965643 = 40076854$$

We determine that v = 97945847 and $h = n-v$ = 19796.

Solving the quadratic equation:

$$z^2 - hz + n = z^2 - 19796z + 97965643 = 0$$

We determine that:

$$z_{1,2} = 9898 \pm 69$$

Table 1: Computation of MMI of g = 22 modulo n

T = 97965643	*B* = 22	*H* = 17	5	2	1
Stack	*F* = 4452983	1	3	2	**
b=40076854	9	7	2	1	0

Therefore:

p: = z_1= 9967, q: = z_2 = 9829

Direct verification: indeed:

pq = 97965643

Dealing with multiplicity of DLP solutions: If n is a prime and g is a generator (primitive root), then the DLP (2) has a unique solution and as a result, there is a unique multiplicative inverse in (3). However, if n is a composite, then the following identity holds for every g that is relatively prime with $n = pq$:

$$g^L \bmod pq = 1 \tag{20}$$

Where:

$$L := (p-1)(q-1) / \gcd(p-1, q-1) \tag{21}$$

Therefore, there exists more then one solution to Eq. 4. Indeed, if:

$$b = g^{L-1} \bmod pq \tag{22}$$

then for every integer m that satisfies the inequality:

$$1 \le mL - 1 \le \varphi(n) - 1 < pq = n \tag{23}$$

also holds:

$$b = g^{mL-1} \bmod pq \tag{24}$$

For instance, there are several solutions of the DLP in the numerical example provided above. One of them is v=18017. To avoid ambiguity it is essential to find on [E, T] the largest integer v that satisfies Eq. 4.

Proposition1: Let:

$$M \le \min(p, q);\ B := \varphi(n) \tag{25}$$

and

$$T := \left\lfloor \left(\sqrt{n} - 1\right)^2 \right\rfloor \tag{26}$$

Then for every $n = pq$ the following inequalities hold:

$$n(M-1)/M - M \le B \le T < n \tag{27}$$

and, if $M = \sqrt[3]{n}$, then:

$$E := \left\lceil \sqrt[3]{n}\left[-1/4 + \left(\sqrt[3]{n} - 1/2\right)^2 - 1\right] \right\rceil \le \varphi(n) \le \left\lfloor \left(\sqrt{n} - 1\right)^2 \right\rfloor \tag{28}$$

Example 2: Let n = 868575847.

Let's select g = 2.

Then its multiplicative inverse b modulo n equals b = 434287924.

There are sixteen solutions that satisfy the equation:

$$g^v \bmod n = 434287924$$

Here are listed three of them: v = 54280434, 108560869,…,***868486959*** (largest one smaller than n). To avoid values of v smaller than E the search for v must be strictly on the interval [E, T].

The maximal solution satisfies the inequalities (27) with the upper bound on φ(n) <T = 868516904.

On the other hand, if $M = \left\lfloor \sqrt[3]{n} \right\rfloor$ = 954, then from the inequalities (28) φ(n) 867664550.

Therefore, from the algorithm we find that v=868486959<n; h=n-v=88888 and from the equation:

$$z^2 - 88888z + 868575847 = 0$$

We derive:

$z_{1,2}$ = 44444±33267,
i.e., p: = z_1= 77711, q: = z_2 = 11177.

Algorithm (4)-(15) revisited: Step2 of the Algorithm (4)-(15) can be modified. From Euler's identity it follows that:

$$g^{p+q} = g^{n-1} \pmod{pq} \tag{29}$$

Therefore, modify (4) as follows:
using the algorithm A, solve the DLP:

$$g^v = g^{n-1} \pmod{n} \tag{30}$$

and then solve the equation:

$$z^2 - vz + n = 0 \tag{31}$$

Finally, $p = z_1$ and $q = z_2$.

In order to decide which of two algorithms is better we need to compare the time complexity $T(g)$ to compute g^{n-1} in (30) and the time complexity $T(b)$ to compute the multiplicative inverse b in (4).

Computer experiments demonstrate that the average number of required steps for computation of the multiplicative inverse b is much smaller than the corresponding average number of steps (3log$n$)/2 required for exponentiation g^{n-1}.

General purpose O($\sqrt[3]{n}$) factoring algorithm: Let us demonstrate the algorithm.

Example 3: Let n = 1003939.

Step 1: Verify that n is not divisible by primes smaller than or equal to $M = \left\lfloor \sqrt[3]{n} \right\rfloor = 89$, otherwise n is factorized after at most $M/\log M$ trials;

Step2: Compute the upper bound T and lower bound E on $\varphi(n)$: $T = \left\lceil \left(\sqrt{n} - 1\right)^2 \right\rceil = 1001836$;

$E = \left\lfloor n - \sqrt[3]{n^2} - \sqrt[3]{n} \right\rfloor = 993811$;

Step 3: Select an integer $g<M$ that is relatively prime with n; {all integers smaller than M are co-prime with n}; let g=2;

Step4: Using the algorithm (8)-(15), find the multiplicative inverse b of g; {b=501970};

Step 5: Solve the DLP:

$$g^v \bmod n = b;\ (4),\ \text{where } v \in [E, T]$$

Remark 2: The DLP problem (4) may be solved using any known algorithm for the DLP, including Pollard's rho-algorithm[10].

Solution of DLP via baby-step giant-step algorithm:

Step 6: Let v:=E+Sy+z; where $S := \left\lceil \sqrt{T - E} \right\rceil = 90$ and $0 \le z \le S$; $0 \le y \le S - 1$ are unknown integers.

Remark 3: If the Baby-Step Giant-Step (BSGS) algorithm is used[16], then the values $g^{E+z} \bmod n$ are pre-computed and stored {these are baby steps) for z from 0 to S and $\left(g^{-S}\right)^y b \bmod n$ are computed for y from 0 to S-1 {these are giant steps}.

Step 7: Solve the problem $g^{(E+z)+Sy} = b \pmod n$;

{$2^{(993811+z)+90y} \bmod 1003939 = 501970$;
as a result, we find y=73 and z=74};

Step 8: Compute v= E+Sy+z=1000455;

Remark 4: Another solution of the DLP v=500227 is excluded by the condition that $v>E$;

Step 9: Compute h:=n-v=3484, (5);

Step 10: Solve the equation $z^2 - hz + n = 0$ (6):

$$p := z_1 = 3167; \quad q := z_2 = 317$$

Indeed, n=pq=1003939.

Complexity of algorithm: Both variables y and z {see Step6.2} are changing on the interval [0, S], where

$$S := \left\lceil \sqrt{T - E} \right\rceil \ge \sqrt{\left(\sqrt{n} - 1\right)^2 - \left(n - \sqrt[3]{n^2} - \sqrt[3]{n}\right)} = \sqrt{\sqrt[3]{n^2} - 2\sqrt{n} + \sqrt[3]{n} + 1} = \sqrt[3]{n}\left[1 - o\left(\sqrt[3]{n}\right)\right] = \Theta\left(\sqrt[3]{n}\right) \quad (32)$$

In addition, $M/\log M$ divisions/trials are used in Step 1. Therefore, the algorithm described in Steps 1-10 has time-space complexity:

$$O\left(\sqrt[3]{n} / \log n\right) + \Theta\left(\sqrt[3]{n}\right) = O\left(\sqrt[3]{n}\right) \quad (33)$$

CONCLUSION

It is essential to stress that a seemingly simple algorithm for integer factorization (Step 1-8} is based on the strong assumption that we know a computationally efficient algorithm A for solution of the DLP. The discussed algorithms based on this assumption imply that the complexity of the integer factorization problem cannot be higher than the complexity of the DLP. It is important to emphasize that we are comparing the information-based complexities of problems, not the algorithms used to solve them. A specific algorithm is a method that after a

finite number of well-defined and executable steps provably delivers a solution to a class of problems. Unless it is an optimal algorithm[18,19], it is plausible that its computational complexity can be later reduced.

In contrast, the information-based complexity of a specific problem is an intrinsic characteristic of the problem itself. Presently, there are no strict proofs demonstrating that integer factorization and/or the DL problem is intrinsically complex**.** We can only plausibly assume that they are not computationally "simple" problems. The proposed Algorithm (4)-(15) implies that the integer factorization problem has either the same complexity as the DLP or is less complex than the DLP. The algorithm described in the Step 1-10 has a computational complexity $O(\sqrt[3]{n})$. Furthermore, if the search is balanced, it has complexity:

$$O(n^{1/3}\log^{1/\alpha} n) \quad (34)$$

where *alpha*>1 is an integer.

ACKNOWLEDGEMENT

I express my appreciation to A. J. Menezes for advice, X. Ma and Md. S. Sadik for assistance in computer experiments and P. Fay for assistance and suggestions that improved the style of this research.

REFERENCES

1. Bach, E., 1984. Discrete logarithms and factoring. Technical Report: CSD-84-186, UC-Berkeley, USA. http://portal.acm.org/citation.cfm?id=894497
2. Buhler, J. and N. Koblitz, 1998. Lattice basis reduction, Jacobi sums and hyperelliptic cryptosystems. Bull. Aust. Math. Soc., 58: 147-154. http://www.zentralblattmath.org/ioport/en/?id=427888&type=txt
3. Gauss, C.F., 1986. Disquisitiones Arithmeticae. Yale University Press, ISBN: 0387962549, pp: 472.
4. Jacobson, M.J., N. Koblitz, J.H. Silverman, A. Stein and E. Teske, 2000. Analysis of the xedni calculus attack. Des. Codes Cryptogr., 20: 41-64. http://portal.acm.org/citation.cfm?id=377938
5. Knuth, D., 1997. The Art of Computer Programming: Fundamental Algorithms. 2nd Edn., Addison Wesley, Reading, MA., USA., ISBN: 0201896834, pp: 650-652.
6. Lenstra Jr., H.W., 1987. Factoring integers with elliptic curves. Ann Math., 2: 649-673. https://openaccess.leidenuniv.nl/bitstream/1887/3826/1/346_086.pdf
7. Lenstra, A.K. and J.H.W. Lenstra, 1993. The development of the number field sieve. Lecture Notes Math., 1554: 131-131. http://cat.inist.fr/?aModele=afficheN&cpsidt=61493
8. James, M., 1996. Turning Euler's factoring method into a factoring algorithm. Bull. London Math. Soc., 4: 351-355. http://blms.oxfordjournals.org/cgi/content/abstract/28/4/351
9. McKee, J., 1999. Speeding Fermat's factoring method. Math. Comput., 68: 1729-1737. http://portal.acm.org/citation.cfm?id=333551
10. Pollard, J.P., 1975. Monte Carlo method for factorization. BIT. Num. Math., 15: 331-334. DOI: 10.1007/BF01933667
11. Pomerance, C., 1985. The quadratic sieve factoring algorithm. Proceeding of the EUROCRYPT 84 Workshop on Advances in Cryptology: Theory and Application of Cryptographic Techniques, (ACT'85), Springer-Verlag, Berlin, Paris, France, pp: 169-182. http://portal.acm.org/citation.cfm?id=20194
12. Frenke, J., 2004. Mathematicians collaborate to solve RSA factoring challenge. High Performance Comput., 13. http://www.tgc.com/hpcwire/hpcwireWWW/04/0430/107585.html
13. Crandall, R. and C. Pomerance, 2001. Prime Numbers: A Computational Perspective. 1st Edn., Springer, ISBN: 0-387-94777-9, pp: 227-244.
14. Silverman, J., 2000. The xedni calculus and the elliptic curve discrete logarithm problem. Des. Codes Cryptogr., 20: 5-40. http://portal.acm.org/citation.cfm?id=377935.377937
15. Verkhovsky, B., 1999. Multiplicative inverse algorithm and its complexity. Proceeding of the International Conference on InterSYMP-99, July 28-30, Baden-Baden Germany, pp: 62-67.
16. Verkhovsky, B., 2008. Generalized baby-step giant-step algorithm for discrete logarithm problem. Advances in Decision Technology and Intelligent Information Systems, 2008, IIAS, Baden-Baden, Germany, ISBN: 978-1-897233-26-9, pp: 88-89.
17. Viète, F. and F. van Schooten, 1970. Opera Mathematica. http://www.amazon.de/exec/obidos/ASIN/B0000BTZBF/ref=nosim/mathworld02-21
18. Traub, J.F., 1980. A General Theory of Optimal Algorithms. ACM Monograph Series, ISBN: 9780126976502.
19. Traub, J.F., G.W. Wasilkowski and H. Wozniakowski, 1983. Information, Uncertainty, Complexity. Addison-Wesley, Reading, MA., ISBN: 0201078902.

20. Pomerance, C., J.W. Smith and R. Tuler, 1988. A pipeline architecture for factoring large integers with the quadratic sieve algorithm. SIAM J. Comput., 17: 387-403. http://portal.acm.org/citation.cfm?id=45486
21. Lenstra, A.K., 2000. Integer factoring. Des. Codes Cryptogr., 19: 101-128. DOI: 10.1023/A:1008397921377
22. Seysen, M., 1987. A probabilistic factorization algorithm with quadratic forms of negative discriminant. Math. Comput., 48: 757-780. http://www.jstor.org/stable/2007842
23. Schirokauer, O., 2000. Using number fields to compute logarithms in finite fields. Math. Comput., 69: 1267-1283. http://portal.acm.org/citation.cfm?id=349887
24. LaMacchia, B.A. and A.M. Odlyzko, 1991. Computation of discrete logarithms in prime fields. Des. Codes Cryptogr., 19: 47-62. DOI: 10.1007/BF00123958
25. Adleman, L.M. and J. DeMarrais, 1993. A subexponential algorithm for discrete logarithms over all finite fields. Math. Comput., 61: 1-15. http://www.jstor.org/stable/2152932
26. Enge, A. and P. Gaudry, 2000. A general framework for sub-exponential discrete logarithm algorithms. Research Report LIX/RR/00/04, LIX. http://www.math.uniaugsburg.de/~enge/vorabdrucke/subexp.ps.gz
27. Müller, V., A. Stein and C. Thiel, 1999. Computing discrete logarithms in real quadratic congruence function fields of large genus. Math. Comput., 68: 807-822. http://portal.acm.org/citation.cfm?id=312069
28. Zuccherato, R., 1998. The equivalence between elliptic curve and quadratic function field discrete logarithms in characteristic 2. Lecture Notes Comput. Sci., 1423: 621-638. http://portal.acm.org/citation.cfm?id=749880