Factoring with an Oracle

Ueli M. Maurer

Institute for Theoretical Computer Science
ETH Zürich
CH-8092 Zürich, Switzerland
Email address: maurer@inf.ethz.ch

Abstract. The problem of factoring integers in polynomial time with the help of an (infinitely powerful) oracle who answers arbitrary questions with yes or no is considered. The goal is to minimize the number of oracle questions. Let N be a given composite n-bit integer to be factored. The trivial method of asking for the bits of the smallest prime factor of N requires $n/2$ questions in the worst case. A non-trivial algorithm of Rivest and Shamir requires only $n/3$ questions for the special case where N is the product of two $n/2$-bit primes. In this paper, a polynomial-time oracle factoring algorithm for general integers is presented which, for any $\epsilon > 0$, asks at most ϵn oracle questions for sufficiently large N. Based on a conjecture related to Lenstra's conjecture on the running time of the elliptic curve factoring algorithm it is shown that the algorithm fails with probability at most $N^{-\epsilon/2}$ for all sufficiently large N.

1. Introduction

An interesting direction of research in complexity theory is to determine the complexity of a problem under the assumption that an oracle is available who answers questions about the particular instance of the problem to be solved. Clearly, to introduce such an oracle can make sense only if some restrictions are placed on the questions that may be asked; otherwise every problem could trivially be solved by asking the oracle for the solution. For a given problem

R.A. Rueppel (Ed.): Advances in Cryptology - EUROCRYPT '92, LNCS 658, pp. 429-436, 1993.

that is believed to be difficult there exists a trade-off between the restriction on the questions and the running time of an algorithm solving the problem with such restricted use of the oracle.

A natural restriction is to allow arbitrary questions with a binary answer (yes/no) but to restrict the number of questions. Note that every question having a d-ary answer can easily be simulated by $\lceil \log_2 d \rceil$ questions with binary answer. Clearly, in order to be of interest, the number of questions must be smaller than the size in bits of the solution.

It is common practice in theoretical computer science to distinguish as a coarse classification for the feasibility of an algorithm between polynomial and superpolynomial running time. The goal of the research described in this paper is to find a polynomial-time algorithm for a given problem (here integer factorization) asking as few questions as possible.

One motivation for considering this problem is to determine whether or not the difficulty of a certain problem can be concentrated in a few difficult bits, leading to a new complexity-theoretic classification of problems. Another motivation is the fact that if the number of questions could be reduced to $O(\log n)$ where n is the input size, then all possible oracle answers could be checked in polynomial time; this would result in a polynomial-time algorithm (without access to the oracle) for the original problem.

Motivated by a paper by Rivest and Shamir [5], this paper is concerned with the problem of factoring integers, which is widely believed to have no polynomial-time algorithm for its solution. In fact, several cryptographic systems (e.g., [6]) rely on the difficulty of factoring. A non-trivial factor of every n-bit integer N can easily be determined by asking $n/2$ questions, namely, "What is the i-th bits of the smallest prime factor of N?", for $i = 1, \ldots, n/2$. For the special case of integers that are the product of two primes of roughly equal size, Rivest and Shamir [5] described a polynomial-time algorithm based on integer programming which asks at most $n/3$ question. In this paper, a polynomial-time algorithm is presented that, for any given $\epsilon > 0$, asks at most ϵn questions. The claim that the algorithm fails only with exponentially small probability is based on a plausible number-theoretic conjecture about the distribution of smooth numbers in certain intervals and is closely related to a conjecture by Lenstra that he used in the (heuristic) running time analysis of his elliptic curve factoring algorithm [3].

The major motivation of Rivest and Shamir for investigating this problem was that an adversary often has some side-information about the secret parameters of a cryptographic system. Our analysis corresponds to a worst-case scenario in which the adversary can choose precisely what side-information he would like to obtain. This paper demonstrates that cryptosystems whose security is based on the difficulty of factoring a modulus (e.g., the RSA system [6]) could be

broken if an adversary were allowed to obtain only ϵn bits of information of his choice about the modulus. However, because oracles do not exist in reality, the results of this paper have no direct implication on the security of existing cryptographic systems.

2. Preliminaries

The following lemma shows that in a sequence of k pairwise independent events, each having probability p of occurring, the probability that none of these events occurs is at most $(1-p)/(kp)$. The events are pairwise independent if for any two events A and B, $P[A \cap B] = P[A] \cdot P[B]$.

Lemma. *Let $X_1, \ldots, X_k$ be pairwise independent binary random variables where $P[X_i = 1] = p$ for $1 \leq i \leq k$. Then*

$$P[X_1 = X_2 = \cdots = X_k = 0] \leq \frac{1-p}{kp}.$$

Proof. Note that the expected value and the variance of X_i are given by $E[X_i] = p$ and $\mathrm{Var}[X_i] = p(1-p)$, respectively. Let S be the integer sum of $X_1, \ldots, X_k$, i.e., $S = X_1 + \cdots + X_k$. Hence $S = 0$ if and only if $X_1 = \cdots = X_k = 0$, and $E[S] = kp$. It is not difficult to prove (cf. [2]) that the variance of the sum of several pairwise independent random variables is equal to the sum of the individual variances. Thus $\mathrm{Var}[S] = kp(1-p)$. For every real-valued random variable Y we have

$$\mathrm{Var}[Y] \geq P[Y = 0] \cdot E[Y]^2$$

since the right-hand side is only one of several positive terms summing to the variance. We conclude that $P[S = 0] \leq \mathrm{Var}[S]/E[S]^2 = (1-p)/(kp)$. □

Remarks. It is well-known that the expected value of the sum of several random variables is equal to the sum of their expected values, and that the variance of the sum is equal to the sum of the variances if the random variables are statistically independent. It is less well-known that a sufficient condition for the rth moment of the sum to be equal to the sum of the rth moments is that the random variables be only r-wise statistically independent. For fixed p and $k \to \infty$ the proved bound on the probability that $X_1 = \cdots = X_k = 0$ is $O(1/k)$, which is optimal.

It is well-known that a polynomial of degree at most d over a field can be interpolated from any set of $d+1$ distinct arguments and their corresponding polynomial values. For the case of a finite field $GF(q)$ with q elements (where q is a prime power) this observation leads to a construction of a sequence of q $(d+1)$-wise independent random variables: When the $d+1$ coefficients of the

polynomial are selected randomly and independently from $GF(q)$ with uniform distribution, then the polynomial's values for any set of $d+1$ arguments are also statistically independent and uniformly distributed. We will make use only of the special case $d = 1$ (pairwise independence).

For a prime $p > 3$ the elliptic curve over $GF(p)$ with parameters a and b satisfying $4a^3 + 27b^2 \neq 0$ is defined as the set of points (x, y) with $x, y \in GF(p)$ satisfying the congruence equation

$$y^2 \equiv x^3 + ax + b \pmod{p}, \tag{1}$$

together with a special element denoted $\mathcal{O}$ and called the point at infinity. This curve is denoted as $E_{a,b}(p)$. It is well-known that a group operation, which is called addition, can be defined on the set of points of the elliptic curve $E_{a,b}(p)$. Let P and Q be two points on $E_{a,b}(p)$. The point $P + Q$ is defined according to the following rules. $P + \mathcal{O} = \mathcal{O} + P = P$ for all P on E (i.e., $\mathcal{O}$ is the identity element of $E_{a,b}(p)$). Let $P = (x_1, y_1)$ and $Q = (x_2, y_2)$. If $x_1 = x_2$ and $y_1 = -y_2$, then $P + Q = \mathcal{O}$ (i.e., the negative of the point (x, y) is the point $(x, -y)$). In all other cases the coordinates of $P + Q = (x_3, y_3)$ are computed as follows. Let λ be defined by

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} & \text{if } x_1 \neq x_2 \\ \dfrac{3x_1^2 + a}{2y_1} & \text{if } x_1 = x_2, \end{cases}$$

where all operations are to be computed modulo p. (When $P + Q \neq \mathcal{O}$ then the denominator is not zero and thus the quotient is defined.) The resulting point $P + Q = (x_3, y_3)$ is defined by

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 \\ y_3 &= \lambda(x_1 - x_3) - y_1. \end{aligned}$$

The prime p can be replaced by a composite N in the above definition and equations. However, $E_{a,b}(N)$ defined in this manner is not a group, but it can be extended to form a group by adjoining a small fraction of additional elements. (In the case where $N = p_1 \cdots p_r$ is the product of distinct primes > 3, $E_{a,b}(N)$ is the direct product of the corresponding elliptic curves over $GF(p_1), \ldots, GF(p_r)$.) Nevertheless, the addition operation, which is in this case called pseudo-addition, can be performed as long as it is defined, i.e., when the denominator is relatively prime to N, and it corresponds in fact to the addition operation on the extended curve. We refer to [3] for further information on elliptic curves. Note that in [3] points (x, y) are represented in projective coordinates as triples $(x : y : 1)$, and $\mathcal{O}$ is represented as $(0 : 1 : 0)$.

Unless stated otherwise, logarithms in this paper are to the natural base e. The cardinality of a set S is denoted by $\#S$.

3. The Oracle Factoring Algorithm

Let N be a given composite n-bit integer and let $\epsilon < 0.5$ be an arbitrary given positive constant. If N is not known to be composite, a simple probabilistic compositeness test such as the Miller-Rabin test [4] can be used to prove the compositeness of N. In the sequel a polynomial-time (in n) algorithm is described for finding a non-trivial divisor d of N $(1 < d < N)$ which, for all sufficiently large N, succeeds with probability at least $1 - N^{-\epsilon/2}$ and asks at most ϵn oracle questions.

The algorithm consists of four steps.

(i) (Special cases.) If 2 or 3 divides N or if N is a prime power $N = q^t$, output 2, 3 or q, respectively, and stop.

(ii) (Setup.) Choose δ with $0 < \delta < \epsilon$ as an arbitrary positive constant and let

$$c = \frac{1}{\epsilon - \delta}$$

and

$$w = (\log N)^c.$$

Let further

$$h = \prod_{r \le w,\, r \text{ prime}} r^{e(r)}, \tag{2}$$

where $e(r)$ is the largest integer m with $r^m \le N^{1/2} + 2N^{1/4} + 1$. Choose s and t randomly from $GF(2^{3n})$. Fix a natural enumeration of the elements of $GF(2^{3n})$: $\alpha_1, \alpha_2, \ldots, \alpha_{2^{3n}}$. For a given natural representation of the elements of $GF(2^{3n})$ as triples of n-bit integers, let $(a_k, x_k, y_k) \in Z_{2^n} \times Z_{2^n} \times Z_{2^n}$ be the triple corresponding to $s\alpha_k + t$ where α_k is the k-th element of $GF(2^{3n})$, and let $b_k \in Z_N$ be defined by

$$b_k \equiv y_k^2 - x_k^3 - a_k x_k \pmod{N}.$$

Remarks. $N^{1/2} + 2N^{1/4} + 1$ is an upper bound on the order of an elliptic curve over $GF(p)$, where p is the smallest prime factor of N. As mentioned in Section 2 the above construction guarantees that the triples (a_k, x_k, y_k) are pairwise statistically independent. Instead of the field $GF(2^{3n})$ any other finite field with cardinality greater than N^3 could be used to create an appropriate list of pairwise independent triples (a_k, x_k, y_k). Only triples for which all three components are smaller than N will actually be of possible use.

(iii) (Oracle questions.) Now ask the oracle the following question. If there exists a positive integer $k < 2^{\lfloor \epsilon n \rfloor}$ such that

(1) for the smallest prime factor p of N,

$$4a_k^3 + 27b_k^2 \not\equiv 0 \pmod{p},$$

and each prime factor r dividing $\#E_{a_k,b_k}(p)$ satisfies $r \leq w$,

(2) and for some prime factor $q \neq p$ of N,

$$4a_k^3 + 27b_k^2 \not\equiv 0 \pmod{q}$$

and $\#E_{a_k,b_k}(q)$ is not divisible by the largest prime number dividing the order of the point $(x_k : y_k : 1)$ on the elliptic curve $E_{a_k,b_k}(p)$,

where a_k, x_k, y_k and b_k are defined in step (ii), then output (the binary representation of) the smallest such k, else output 0.

Remark. Of course, this question can easily be transformed into $\lfloor \epsilon n \rfloor$ questions with a yes/no answer.

(iv) (Factorization.) If the oracle's answer is 0, stop. In this case, the algorithm fails. If the oracle's answer is some $k > 0$, proceed as follows. Compute (a_k, x_k, y_k) and b_k as described in step (ii). Let $P = (x_k : y_k : 1)$ be a point on $E_{a_k,b_k}(N)$ (which is not a group). Try to compute $h \cdot P$ using the pseudo-addition method described in (2.4) of [3], pretending that N is prime. At some point during this computation the addition of two points $(x' : y' : 1)$ and $(x'' : y'' : 1)$ will fail because $\gcd(x' - x'', N) > 1$. Output this divisor of N.

4. Analysis of the Algorithm

We need to prove first that the algorithm works and runs in polynomial time and second that the failure probability is upper bounded by $N^{-\epsilon/2}$.

Theorem 1. *If the oracle's answer is $k > 0$ then the algorithm runs in polynomial time and always finds a non-trivial divisor of N.*

Proof. That the algorithm runs in polynomial time follows from the facts that the pseudo-addition can be performed in time $O(n^2)$ and that the number of pseudo-additions required for computing $h \cdot P$ is at most $2\lceil \log_2 h \rceil - 1$ which is polynomial in n since according to (2),

$$\log_2 h \;=\; \sum_{r \leq w,\, r \text{ prime}} e(r) \log_2 r \;\leq w \log_2 w$$

and $w = O(n^c)$. N is guaranteed to have a prime factor smaller than $\sqrt{N}$ and hence Proposition (2.6) in [3] for $\nu = \sqrt{N}$ implies that the algorithm always succeeds. □

It follows from the Corollary to Theorem 3.1 of Canfield, Erdös and Pomerance [1] that the probability that a random positive integer $s \leq x$ has all its prime factors $\leq L(x)^\alpha$, where $L(x) = e^{\sqrt{\log x \log\log x}}$, is $L(x)^{-1/(2\alpha)+o(1)}$, for $x \to \infty$. In the analysis of the elliptic curve factoring algorithm [3], Lenstra stated the conjecture that the same result is valid if s is a random integer in the interval $(x+1-\sqrt{x}, x+1+\sqrt{x})$. We will need a similar conjecture with a smaller smoothness bound. One can prove that the mentioned result of Canfield et al. implies that the probability that a random positive integer $s \leq x$ has all its prime factors $\leq (\log x)^c$ for $c > 1$ is greater than $x^{-1/c-\beta}$ for all $\beta > 0$ and for all sufficiently large x. The conjecture we will need is that the same result is valid if $1/c + \beta < 1/2$ and s is a random integer in the interval $(x+1-\sqrt{x}, x+1+\sqrt{x})$. We believe that our conjecture appears to be equally plausible as Lenstra's conjecture. Note that $c > 2$ in our algorithm but that for $c < 2$ the conjecture cannot be true since the expected number of smooth integers in the given interval is less than 1.

Theorem 2. *If the described conjecture is true, then the oracle outputs* 0 *(and hence the oracle factoring algorithm fails) with probability at most* $N^{-\epsilon/2}$.

Proof. Let p be the smallest prime divisor of N, and let U be the number of integers in the interval $(p+1-\sqrt{p}, p+1+\sqrt{p})$ for which no prime factor is greater than $w = (\log N)^c$. According to our conjecture with $\beta = \delta/2$, U is lower bounded by

$$U > (2\lfloor\sqrt{p}\rfloor + 1)p^{-1/c-\delta/2},$$

for all sufficiently large p. Note that $-1/c - \delta/2 = -\epsilon + \delta/2$. It follows from proposition (2.7) of [3] that the number T of triples $(a,x,y) \in Z_N \times Z_N \times Z_N$ that are successful in step (iii) of our algorithm is, for sufficiently large p, lower bounded by

$$\begin{aligned} T &> N^3 \frac{C_1}{\log p} \cdot \frac{U-2}{2\lfloor\sqrt{p}\rfloor + 1} \\ &> N^3 \frac{C_2}{\log p} \cdot p^{-\epsilon+\delta/2}, \end{aligned}$$

where C_1 and C_2 are positive constants. Hence the probability that a triple selected randomly from $Z_{2^n} \times Z_{2^n} \times Z_{2^n}$ is successful is equal to $T/2^{3n}$. Because the triples (a_k, x_k, y_k), $1 \leq k \leq 2^{3n}$, are pairwise independent, it follows from the lemma in Section 2 that the probability Q that none of the triples (a_k, x_k, y_k), for $1 \leq k < 2^{\lfloor \epsilon n \rfloor} - 1$, is successful (and therefore the oracle answers 0) is upper

bounded by

$$\begin{aligned} Q &< \frac{1}{(T/2^{3n}) \cdot (2^{\lfloor \epsilon n \rfloor} - 1)} \\ &< \frac{1}{\frac{1}{8}\frac{C_2}{\log p} \cdot p^{-\epsilon+\delta/2} \cdot (\frac{1}{2}N^{\epsilon} - 1)}, \end{aligned}$$

where we have made use of $N^3/2^{3n} > 1/8$ and $2^{\lfloor \epsilon n \rfloor} > 2^{\epsilon n - 1} > \frac{1}{2}N^{\epsilon}$. Since $p \leq N^{1/2}$ the last expression is smaller than $N^{-\epsilon/2}$ for all sufficiently large N. □

Acknowledgments

It is a pleasure to thank Uri Feige and Andrew Odlyzko for helpful discussions and an anonymous program committee member for a comment on our conjecture. Claus Schnorr has pointed out that a result similar to ours could be obtained by using the class group factoring algorithm of [7] instead of the elliptic curve factoring algorithm.

References

[1] E.R. Canfield, P. Erdös and C. Pomerance, On a problem of Oppenheim concerning "Factorisatio Numerorum", *J. Number Theory*, Vol. 17, pp. 1-28, 1983.

[2] B. Chor and O. Goldreich, On the power of two-point based sampling, *Journal of Complexity*, Vol. 5, No. 1, pp. 96-106, 1989.

[3] H.W. Lenstra, Jr., Factoring integers with elliptic curves, *Annals of Mathematics*, Vol. 126, pp. 649-673, 1987.

[4] M.O. Rabin, Probabilistic algorithm for testing primality, *Journal on Number Theory*, Vol. 12, pp. 128-138, 1980.

[5] R.L. Rivest and A. Shamir, Efficient factoring based on partial information, *Advances in Cryptology - EUROCRYPT '85*, Lecture Notes in Computer Science, Vol. 219, Berlin: Springer-Verlag, pp. 31-34, 1986.

[6] R.L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM*, Vol. 21, No. 2, pp. 120-126, 1978.

[7] C.P. Schnorr and H.W. Lenstra, A Monte Carlo factoring algorithm with linear storage, *Mathematics of Computation*, Vol. 43, No. 167, pp. 289-311, July 1984.