USING LUCAS SEQUENCES TO FACTOR LARGE INTEGERS NEAR GROUP ORDERS

Zhenxiang Zhang*
Dept. of Math., Anhui Normal University, 241000 Wuhu, Anhui, P.R. China
e-mail: zhangzhx@mail.ahwhptt.net.cn

(*Submitted March 1999-Final Revision August 1999*)

Dedicated to the memory of P. Erdös (1913-1996)

1. INTRODUCTION

Factoring large integers into primes is one of the most important and most difficult problems of computational number theory (the twin problem is primality testing [13]). Trial division, Fermat's algorithm [1], [3], [8], Pollard's $p-1$ method [6], Williams' $p+1$ method [11], Lenstra's elliptic curve method (ECM) [5], Pomerance's quadratic sieve (QS) [7], [10], and Pollard's number field sieve (NFS) [4] are commonly used methods for factorization.

Trial division and Fermat's method are two of the oldest systematic methods of factoring integers. Although, in general, both methods are not very efficient, it is worthwhile attempting them before other methods. Trial division consists of making trial divisions of the integer N by the small primes; it succeeds when

$$N = pq, \text{ prime } p \text{ is small.} \tag{1.1}$$

The practical limit for trial division to locate a prime factor of large N is about 8-10 digits. Fermat's algorithm works in the opposite direction from trial division. It works quickly when N is the product of two almost equal factors, i.e.,

$$N = pq, \ |q-p| \text{ is small.} \tag{1.2}$$

Integers whose largest prime factor is small are called *smooth*. The $p-1$ method succeeds when

$$p-1 \text{ is smooth for some prime divisor } p \text{ of } N. \tag{1.3}$$

The method is based on the consequence of Fermat's Little Theorem: if M is a multiple of $p-1$ and if p does not divide a, then p divides $\gcd(N, a^M-1)$. If $p-1$ is smooth, then we can find a suitable M by taking the product of small primes and powers of very small primes.

In 1982, Hugh Williams [11] showed how to use the structure of Lucas sequences to factor N when

$$p+1 \text{ is smooth for some prime divisor } p \text{ of } N. \tag{1.4}$$

His method is based on the following fact about Lucas sequences (Theorem 12.8, [1]). If we choose an integer u and define a pair of Lucas sequences $U_n = U_n(u)$ and $V_n = V_n(u)$ by

$$\begin{cases} U_0 = 0, \ U_1 = 1, \ V_0 = 2, \ V_1 = u, \\ U_n = uU_{n-1} - U_{n-2}, \ V_n = uV_{n-1} - V_{n-2} \ \text{ for } n \geq 2, \end{cases} \tag{1.5}$$

and, if $D = u^2 - 4$, then, for any odd prime p, p divides both $\gcd(N, U_M)$ and $\gcd(N, V_M - 2)$ whenever M is a multiple of $p-(D/p)$, where (D/p) is the Legendre symbol. If $(D/p) = -1$

* Supported by the China State Educational Commission Science Foundation and by NSF of China Grant 10071001.

and $p+1$ is smooth, then we can find a suitable M by taking the product of small primes and powers of very small primes.

So far, either textbooks [1], [3], [8] or survey papers [2], [12] on factorization treated the above four methods separately. In this paper we present algorithms not only to unify but also to enhance these four methods. We state our main results as the following two theorems.

Theorem 1: There exists an algorithm (Algorithm 1) for finding prime divisors $p<q$ of N in $O(\log^3 N + |r|\log^2 N)$ bit operations, provided

$$N = pq \text{ with } q = k(p-1)+r \text{ and } |r| < (p-3)/2. \tag{1.6}$$

Theorem 2: There exists an algorithm (Algorithm 2) for finding prime divisors $p<q$ of N in $O(\log^3 N + |r|\log^2 N)$ bit operations, provided

$$N = pq \text{ with } q = k(p+1)+r \text{ and } |r| < (p+1)/2. \tag{1.7}$$

Remark 1.1: Clearly, Algorithm 1 finds prime factors p and q of N quickly when

$$N = pq \text{ with } q = k(p-1)+r \text{ and } |r| \text{ small}, \tag{1.8}$$

and Algorithm 2 finds prime factors p and q of N quickly when

$$N = pq \text{ with } q = k(p+1)+r \text{ and } |r| \text{ small}. \tag{1.9}$$

We remark that condition (1.8) can be relaxed to

$$N = pq \text{ with } q = k'd+r',\ |r'| \text{ small, and } (p-1)/d \text{ smooth}, \tag{1.10}$$

where d is a divisor of $p-1$; whereas condition (1.9) can be relaxed to

$$N = pq \text{ with } q = k'd+r',\ |r'| \text{ small, and } (p+1)/d \text{ smooth}, \tag{1.11}$$

where d is a divisor of $p+1$. We see that conditions (1.1), (1.2), and (1.3) are contained in condition (1.10); whereas conditions (1.1), (1.2), and (1.4) are contained in condition (1.11). Thus, we have a unified approach for trial division, Fermat's method, and Pollard's $p-1$ method; and a unified approach for trial division, Fermat's method, and Williams' $p+1$ method.

2. PROOF OF THEOREM 1

To prove Theorem 1 we need two lemmas.

Lemma 2.1: Let $N = pq$ be the product of two primes $p<q$ with $q = k(p-1)+r$, where $|r| < (p-3)/2$. Let M be the number of positive integers a modulo N with

$$\gcd(a, N) = 1 \quad \text{and} \quad a^N \equiv a^r \bmod N. \tag{2.1}$$

Then we have $M < N/2$.

Proof: Since $p < q = k(p-1)+r$ and $|r| < (p-3)/2$, we have $k \geq 2$, or $k = 1$ and $r \geq 2$. In both cases, we have $p-r < q-1$. Thus,

$$\gcd(N-r, q-1) = \gcd(p(q-1)+p-r, q-1) = \gcd(p-r, q-1) \leq (q-1)/2.$$

The number of such bases a satisfying (2.1) is the number of solutions (mod N) of the congruence $f(x) = x^{N-r} - 1 \equiv 0 \bmod N$. It is well known that congruence $f(x) \equiv 0 \bmod p$ has $\gcd(N-r, p-1) = \gcd((p-1)(q+k), p-1) = p-1$ distinct solutions (mod p), and congruence $f(x) \equiv 0$ mod

q has $\gcd(N-r, q-1) \le (q-1)/2$ distinct solutions (mod q). According to the Chinese Remainder Theorem, we have

$$M = \gcd(N-r, p-1)\gcd(N-r, q-1) \le (p-1)(q-1)/2 < N/2. \ \square$$

Lemma 2.2: Let $N = pq$ with p prime and $q = k(p-1)+r$ not necessarily prime. Let $a > 1$ with $\gcd(a, N) = 1$ and $u = a^N - a^r \bmod N$. Then we have:

(a) $p \mid \gcd(u, N)$;

(b) If q is prime and $u \ne 0$, then $\gcd(u, N) = p$.

Proof:

(a) This follows from the fact that $a^N = a^{pq} \equiv a^q = a^{k(p-1)+r} \equiv a^r \bmod p$.

(b) Since $u \ne 0 \bmod N$ and $u \equiv 0 \bmod p$, $u \ne 0 \bmod q$. Thus, we have $\gcd(u, N) = p$. $\square$

Example 2.1: Let $N = 26544669$. Then

$$2^N \equiv 19445336 \bmod N, \text{ and } \gcd(19445336 - 2^9, N) = 2823 = 941 \cdot 3.$$

In fact, $N = 941 \cdot 28209$, where 941 is prime, whereas $28209 = 30(941-1)+9 = 3 \cdot 9403$ is not prime.

Example 2.2: Let $N = 8848223$. Then

$$2^N \equiv 864787 \bmod N, \text{ and } \gcd(864787 - 2^3, N) = 941.$$

Thus, $N = 941 \cdot 9403$, where both 941 and $9403 = 10(941-1)+3$ are primes.

Example 2.3: Let $N = 8836931$. Then

$$2^N \equiv 4892191 \bmod N, \ 2^{-1} \equiv 4418466 \bmod N,$$

$$2^{-9} \equiv 2571685 \bmod N, \text{ and } \gcd(4892191 - 2571685, N) = 941.$$

Thus, $N = 941 \cdot 9391$, where both 941 and $9391 = 10(941-1) - 9$ are primes.

Now we are ready to prove Theorem 1.

Proof of Theorem 1: Suppose condition (1.6) holds. We present *Algorithm* 1 as follows:

We first select a random integer a with $1 < a < N$ and $\gcd(a, N) = 1$; and do the modular exponentiation $b = a^N \bmod N$ and calculate $a^{-1} \bmod N$ via the Euclidean algorithm. Then, for $i = 1, 2, \ldots$, calculate $a^i = a^{i-1}a \bmod N$ and $a^{-i} = a^{-(i-1)}a^{-1} \bmod N$ by recurrence, and calculate $\gcd(b - a^i, N)$ and $\gcd(b - a^{-i}, N)$ via the Euclidean algorithm.

By Lemma 2.1, the probability that a random integer a modulo N satisfies

$$a^N \ne a^r \bmod N \tag{2.2}$$

is at least $1/2$. Suppose (2.2) holds for the chosen a. By Lemma 2.2, we have

$$\gcd(a^N - a^r, N) = p \quad \text{and} \quad q = N/p.$$

It is well known that it takes $O(\log^3 N)$ bit operations for modular exponentiation [9] and $O(\log^2 N)$ bit operations to do a gcd with naive arithmetic (Euclidean algorithm) [3]. Thus, in total, it takes $O(\log^3 N + |r|\log^2 N)$ bit operations to find prime divisors $p < q$ of N. This completes the proof. $\square$

Example 2.4: Let $N = 89603 \cdot 10^{198} + 51701096 \cdot 10^{99} + 7457884581$ (203 digits). Using Algorithm 1, we obtain $N = pq$, where both

$$p = \gcd(2^N - 2^{165}, N) = 10^{99} + 289 \text{ (100 digits)}$$

and

$$q = N / p = 89603(p-1) + 165 = 89603 \cdot 10^{99} + 25805829 \text{ (104 digits)}$$

are primes. Our Pascal program (with multi-precision package partially written in Assembly language) ran about eighteen seconds on my PC 486/66 to get the desired results.

3. COMBINED WITH POLLARD'S $p-1$ METHOD

The following *Extended Algorithm* 1 combines *Algorithm* 1 presented in the proof of Theorem 1 with Pollard's $p-1$ algorithm, thus it unifies trial division, Fermat's method, and the $p-1$ method.

Extended Algorithm 1: We first select a random integer g with $1 < g < N$ and $\gcd(g, N) = 1$. Then calculate $a = g^M$ mod N, where M is the product of all small primes and some powers of very small primes. If $1 < \gcd(a-1, N) < N$, then a nontrivial factor is found (the $p-1$ algorithm ends up here); otherwise, calculate $b = a^N$ mod N. If condition (1.10) holds, then the prime divisor p could be found quickly, since in this case we would most likely have $\gcd(b - a^{r'}, N) = p$.

Example 3.1: Let $N =$

21599677 4125459698 7880191329 6573463347 1444517931 6954707436
3533196547 4958078521 1295059800 6895461157 4586337662 0125667872
2212935015 1101826633 4121506661 8644391868 2033158453 4956423476
3200995905 4369044649 0215558908 4213065793 (218 digits).

Let $a = 2^M$ mod N, where M is the product of the first 120 primes. We obtain $N = pq$, where $p = \gcd(a^N - a^{23}, N) =$

2912 4205259383 1345758783 9106248908 4606333874 4736995720
6878160308 4991206875 7497656678 0499080822 1052741991 (104 digits)

and $q = N / p =$

7416 4006262753 0801524787 1419019374 7405994078 1097519023
9058213161 4441575950 4705008092 8187116939 4073700000 0000000023
(114 digits).

The whole calculation took about twenty seconds on my PC 486. We find that

$$q = k(p-1) + r = 10^{15}d + 23,$$

where $(p-1)/d = 2 \cdot 3 \cdot 5 \cdot 7 \cdot 11 \cdot 17$ is smooth, and $r' = 23$ is small, although $r = 245 \ldots 39493$ (103 digits) is large.

4. PROOF OF THEOREM 2

In this section and the following section, we need the pair of Lucas sequences $U_n = U_n(u)$ and $V_n = V_n(u)$ to the parameter u as defined by (1.5). When there is no doubt as to the values of the parameter u, we often omit it. Moreover, the U's and V's are calculated modulo N, and the words "mod N" are often omitted, where N is the integer to be factored. We shall use both $|\ |$ and # to denote cardinality of a set, reserving the latter symbol for a set written with braces. Legendre's symbol is denoted by $(\frac{*}{p})$ of $(*/p)$ with p an odd prime.

To prove Theorem 2 and describe *Algorithm* 2, we need four lemmas.

Lemma 4.1: Let $N = pq$, with p an odd prime and q not necessarily prime. Then we have

$$\#\left\{u : 0 \le u < N, \left(\frac{u^2-4}{p}\right) = -1\right\} = \frac{p-1}{p}\cdot\frac{N}{2}.$$

Proof: It is well known that

$$\sum_{u=0}^{p-1}\left(\frac{u^2-4}{p}\right) = -1.$$

Since

$$\#\left\{u : 0 \le u < p, \left(\frac{u^2-4}{p}\right) = 0\right\} = 2,$$

we have

$$\#\left\{u : 0 \le u < p, \left(\frac{u^2-4}{p}\right) = -1\right\} = (p-1)/2,$$

and

$$\#\left\{u : 0 \le u < p, \left(\frac{u^2-4}{p}\right) = 1\right\} = (p-3)/2.$$

Thus,

$$\#\left\{u : 0 \le u < p, \left(\frac{u^2-4}{p}\right) = -1\right\} = \frac{p-1}{2}\cdot\frac{N}{p} = \frac{p-1}{p}\cdot\frac{N}{2}. \quad \square$$

Lemma 4.2: (Lemma 12.15 of [1], see also Section 2 of [11].) If p is an odd prime, $m \in \mathbb{Z}^+$, and $\varepsilon = (\frac{u^2-4}{p})$, then we have $U_{m(p-\varepsilon)} \equiv 0 \bmod p$, and $V_{m(p-\varepsilon)} \equiv 2 \bmod p$.

Lemma 4.3: Let $N = pq$ with p an odd prime and $q = k(p+1)+r$ not necessarily prime. Let integer u be such that $(\frac{u^2-4}{p}) = -1$. Then we have $U_{N+r} \equiv 0 \bmod p$, and $V_{N+r} \equiv 2 \bmod p$.

Proof: Since $N + r = pq + r = (p+1)(pk+r)$, the lemma follows by Lemma 4.2. □

In Lemma 4.4 below we investigate the number of integers u satisfying

$$\left(\frac{u^2-4}{q}\right) = 1,\ U_{N+r}(u) \equiv 0 \bmod q,\ \text{ and }\ V_{N+r}(u) \equiv 2 \bmod q; \tag{4.1}$$

or satisfying

$$\left(\frac{u^2-4}{q}\right) = -1,\ U_{N+r}(u) \equiv 0 \bmod q,\ \text{ and }\ V_{N+r}(u) \equiv 2 \bmod q. \tag{4.2}$$

Lemma 4.4: Let N, p, q, k, and r be as given in Theorem 2, with $k \ge 7$,

$$S_1 = \{u : 0 \le u < N, u \text{ satisfies } (4.1)\}, \quad \text{and} \quad S_2 = \{u : 0 \le u < N, u \text{ satisfies } (4.2)\}.$$

Then we have $|S_1 \cup S_2| < N/4$.

Proof: It is easy to see that condition (4.1) is equivalent to

$$\left(\frac{u^2-4}{q}\right)=1, \text{ there exists a } w\in GF^*(q) \text{ with } w^2-uw+1=0 \text{ and } w^{N+r}=1, \qquad (4.1')$$

and condition (4.2) is equivalent to

$$\left(\frac{u^2-4}{q}\right)=-1, \text{ there exists a } \xi\in GF^*(q^2) \text{ with } \xi^2-u\xi+1=0 \text{ and } \xi^{N+r}=1. \qquad (4.2')$$

Since $q=k(p+1)+r>(k-1/2)(p+1)$, we have $p+1<\frac{q}{k-1/2}$.

In the group $GF^*(q)$, the number of solutions of the equation $x^{N+r}=1$ is

$$\gcd(N+r,q-1)=\gcd(p+r,q-1)\le p+r<3(p+1)/2<3q/(2k-1)<q/4.$$

Since every u (mod q) of the set S_1 corresponds to two different w's, and different u (mod q)'s correspond to different pairs of w's, we have

$$|S_1|<(1/2)\cdot(q/4)\cdot(N/q)=N/8.$$

In the group $GF^*(q^2)$, if ξ is such that

$$\xi^2-u\xi+1=0 \text{ with } \left(\frac{u^2-4}{q}\right)=-1,$$

then $\xi^{q+1}=1$. The number of solutions of the system of equations

$$\begin{cases} x^{q+1}=1,\\ x^{N+r}=1,\end{cases}$$

is

$$\gcd(N+r,q+1)=\gcd(p-r,q+1)\le p-r<3(p+1)/2<q/4.$$

Since every u (mod q) of the set S_2 corresponds to two different ξ's, and different u (mod q)'s correspond to different pairs of ξ's, we have

$$|S_2|<(1/2)\cdot(q/4)\cdot(N/q)=N/8.$$

Therefore, we have $|S_1\cup S_2|<(N/8)+(N/8)=N/4$. □

Now we are ready to prove Theorem 2.

Proof of Theorem 2: Suppose condition (1.7) holds. We present *Algorithm* 2 as follows: Select a random integer u with $0\le u<N$, $u\ne 2$, $u\ne N-2$ and

$$\gcd(u,N)=1 \text{ and } \gcd(D,N)=1, \text{ where } D=u^2-4. \qquad (4.3)$$

If (4.3) does not hold, then a nontrivial factor of N is found. Suppose (4.3) holds.

By Lemma 4.1, for a random integer u, the probability that $(D/p)=-1$ is about $1/2$. Suppose $(D/p)=-1$ with the chosen u.

We first calculate the pair U_N and V_N via the formulas (cf. Lemma 12.5 of [1]):

$$U_{2i}=U_iV_i, \qquad V_{2i}=V_i^2-2,$$
$$U_{2i+1}=U_{i+1}V_i-1, \quad V_{2i+1}=V_{i+1}V_i-u.$$

(This is something like doing modular exponentiation in Algorithm 1.) Then calculate

$$U_{N+1} = (uU_N + V_N)/2, \quad V_{N+1} = (uV_N + DU_N)/2,$$
$$U_{N-1} = uU_N - U_{N+1}, \qquad V_{N-1} = uV_N - V_{N+1}.$$

For $i = 2, 3, \ldots,$ calculate by recurrence,

$$U_{N+i} = uU_{N+i-1} - U_{N+i-2}, \quad V_{N+i} = uV_{N+i-1} - V_{N+i-2},$$
$$U_{N-i} = uU_{N-i+1} - U_{N-i+2}, \quad V_{N-i} = uV_{N-i+1} - V_{N-i+2}.$$

By Lemma 4.3, we have $p \mid U_{N+r}$ and $p \mid (V_{N+r} - 2)$.

Suppose $k \geq 7$. (If $k < 7$, then $N = pq$ with $q = k(p+1) + r = k(p-1) + 2k + r$ can be easily factored by Algorithm 1.) By Lemma 4.4, for a random u, the probability that

$$q \mid U_{N+r}(u) \quad \text{and} \quad q \mid (V_{N+r}(u) - 2) \tag{4.4}$$

is less than $1/4$. Suppose (4.4) does not hold for the chosen u. Then we have $\gcd(U_{N+r}, N) = p$ and/or $\gcd(V_{N+r} - 2, N) = p$, and $N/p = q$.

The time taken by computer for the calculation of $U_N \pmod N$ and $V_N \pmod N$ as explained above and for the calculation of $a^N \pmod N$ are the same order of magnitude for large values of N. So, as analyzed in Theorem 1, bit operations used here is also $O(\log^3 N + |r|\log^2 N)$, but with a larger constant related to the big O-notation than that in Theorem 1. □

Example 4.1: Let $N = 525837811$, $u = 6$, and $D = u^2 - 4 = 32$. Then

$$U_N = 128529829, \; V_N = 365916885, \; U_{N-9} = 154978947, \text{ and } V_{N-9} = 215276907.$$

We have $N = pq$, where

$$p = \gcd(U_{N-9}, N) = \gcd(V_{N-9} - 2, N) = 1621 \text{ and } q = N/p = 324391.$$

We find that $q = 200(p+1) - 9$, $(D/p) = -1$, and $(D/q) = 1$.

Example 4.2: Let $N = 262940789$, $u = 6$, and $D = u^2 - 4 = 32$. Then

$$U_N = 90848206, \; V_N = 211151910, \; U_{N+9} = 256455168, \text{ and } V_{N+9} = 78409393.$$

We have $N = pq$, where

$$p = \gcd(U_{N+9}, N) = \gcd(V_{N+9} - 2, N) = 1621, \text{ and } q = N/p = 162209.$$

We find that $q = 100(p+1) + 9$, $(D/p) = -1$, and $(D/q) = 1$.

Remark 4.1: In Algorithm 2, if the integer u happens to be selected with $(D/p) = 1$ (with probability about $1/2$), then $p \mid U_{p-1}$ and $p \mid (V_{p-1} - 2)$, instead of $p \mid U_{p+1}$ and $p \mid (V_{p+1} - 2)$. If k in (1.7) is small or condition (1.6) holds, we would have $\gcd(U_{N-r}, N) = \gcd(V_{N-r} - 2, N) = p$, where r satisfies (1.6). In this case, Algorithm 2 acts essentially as Algorithm 1 does.

Example 4.3: Let $N = 13157657$, $u = 6$, and $D = u^2 - 4 = 32$. Then

$$U_N = 2945491, \qquad V_N = 1183255,$$
$$U_{N+7} = 3350607, \quad V_{N+7} = 6668796.$$

We have $N = pq$, where

$$p = \gcd(U_{N+7}, N) = \gcd(V_{N+7} - 2, N) = 1621, \text{ and } q = N/p = 8117.$$

We find that $q = 5(p+1)+7$, $(D/p) = (D/q) = -1$.

If $u = 3$ is selected, then $D = u^2 - 4 = 5$ and

$$\begin{array}{ll} U_N = 6604163, & V_N = 281690, \\ U_{N-17} = 12418481, & V_{N-17} = 3076660. \end{array} \quad D = u^2 - 4 = 5$$

We have $N = pq$, where

$$p = \gcd(U_{N-17}, N) = \gcd(V_{N-17} - 2, N) = 1621, \text{ and } q = N/p = 8117.$$

We find that $q = 5(p-1)+17$, $(D/p) = 1$, and $(D/q) = -1$. This example explains Remark 4.1.

Example 4.4: Let

$$N = 10^{224} + 67\cdot 10^{198} + 579\cdot 10^{125} + 39052\cdot 10^{99} + 8381\cdot 10^{27} + 5690121 \text{ (225 digits)},$$

$u = 4$, and $D = u^2 - 4 = 12$. Then we have $N = pq$, where

$$p = \gcd(U_{N+259}, N) = \gcd(V_{N+259} - 2, N) = 10^{99} + 289$$

and

$$q = N/p = 10^{125} + 67\cdot 10^{99} + 29\cdot 10^{27} + 19689.$$

The entire calculation took about forty seconds on my PC 486. We find that

$$q = (10^{26} + 67)(p+1) + 259, \ (D/p) = -1, \text{ and } (D/q) = 1.$$

Remark 4.2: One may calculate only the V's using Algorithm 8.3 of [1]. It takes a little less time than calculating both U's and V's. However, it might happen that $\gcd(N, V_{N+r} - 2) = N$, but $\gcd(N, U_{N+r}) = p$ (cf. Lemma 4.4). So we prefer to calculate both U's and V's.

5. COMBINED WITH WILLIAMS' $p+1$ METHOD

The following *Extended Algorithm* 2 combines *Algorithm* 2 presented in the proof of Theorem 2 with Williams' $p+1$ method, thus it unifies trial division, Fermat's method, and the $p+1$ method.

Extended Algorithm 2: Let u, D be as given in the proof of Theorem 2. Calculate $a = U_M(u)$ and $b = V_M(u)$, where M is the product of all small primes and some powers of very small primes. If $1 < \gcd(a, N) < N$ or $1 < \gcd(b-2, N) < N$, then a nontrivial factor of N is found (the $p+1$ algorithm ends up here). Otherwise, calculate (cf. Lemma 12.14 of [1]) $U_{MN}(u) = a\cdot U_N(b)$ and $V_{MN}(u) = V_N(b)$. Then, for $i = 1, 2, \ldots$, calculate

$$\begin{aligned} U_{M(N+i)}(u) &= \tfrac{1}{2}(b\cdot U_{M(N+i-1)}(u) + a\cdot V_{M(N+i-1)}(u)), \\ V_{M(N+i)}(u) &= \tfrac{1}{2}(b\cdot V_{M(N+i-1)}(u) + D\cdot a\cdot U_{M(N+i-1)}(u)), \\ U_{M(N-i)}(u) &= \tfrac{1}{2}(b\cdot U_{M(N-i+1)}(u) - a\cdot V_{M(N-i+1)}(u)), \\ V_{M(N-i)}(u) &= \tfrac{1}{2}(b\cdot V_{M(N-i+1)}(u) - D\cdot a\cdot U_{M(N-i+1)}(u)). \end{aligned}$$

If condition (1.11) holds, even though $|r|$ in (1.9) is large, the prime divisor p could be found quickly, since in this case we would most likely have

$$\gcd(U_{M(N+r)}(u), N) = p \ \text{ and/or } \ \gcd(V_{M(N+r)}(u) - 2, N) = p.$$

Example 5.1: Let

$$N = 3041465128\cdot 10^{219} + 355851419976\cdot 10^{198} + 1757966843983\cdot 10^{120}$$
$$+ 206120091724443\cdot 10^{99} + 25402620895539900002984764006054838 7 \text{ (229 digits)},$$

$u = 9$, and $D = u^2 - 4 = 77$. Let

$$M = \prod_{p \text{ prime},\ p^{m_p} < 32768} p^{m_p}.$$

Then we have $N = pq$, where

$$p = \gcd(U_{M(N+144)}(u), N) = \gcd(V_{M(N+144)}(u) - 2, N)$$
$$= 3041465128\cdot 10^{99} + 878983421991 \text{ (109 digits)}$$

and

$$q = N/p = 10^{120} + 117\cdot 10^{99} + 289\cdot 10^{21} + 33957 \text{ (121 digits)}.$$

The entire calculation took about twenty minutes on my PC 486. We found that

$$q = k(p+1) + r = (10^{21} + 117)d + 144 \text{ and } (D/p) = (D/q) = -1,$$

where $d = 10^{99} + 289$, $(p+1)/d = 2^3\cdot 13\cdot 19\cdot 47\cdot 32749$, which is smooth and divides *M*, and $r' = 144$ is small, whereas $|r| = -r = 1369...51187$ (109 digits) is large.

Remark 5.1: As mentioned in Remark 4.1, if the integer u happens to be selected with $(D/p) = 1$, then Extended Algorithm 2 acts essentially as Extended Algorithm 1 does. Thus, Extended Algorithm 2 not only unifies trial division, Fermat's method, Pollard's $p-1$ method, and Williams' $p+1$ method, but also enhances these four methods.

6. CONCLUSIONS

The algorithms we have presented each operate in an Abelian group. Algorithm 1 uses the multiplicative group $GF^*(p)$ of nonzero elements of $GF(p)$. The work on Lucas sequences in Algorithm 2 is really arithmetic done in a subgroup, with order $p+1$, of the multiplicative group $GF^*(p^2)$ of nonzero elements of $GF(p^2)$. The prime factor p of N can be found quickly when $N = pq$ satisfies one of the four conditions (1.8), (1.9), (1.10), and (1.11) or, in other words, when $N = pq$ is near the related group orders $p \pm 1$. Moreover, it is easy to see that the "factoring large integers near group orders" idea can be used to Lenstra's Elliptic Curve Method [5] to enhance the ability of ECM for factoring more large integers near the order d_p of the group E_p, elliptic curve E modulo p.

Our algorithms not only unify trial division, Fermat's method, Pollard's $p-1$ method, and Williams' $p+1$ method, but also can quickly factor a class of large integers, which could not be factored by other available methods (such as QS or NFS) within a reasonable amount of time. Thus, such integers should be excluded from RSA moduli candidates.

ACKNOWLEDGMENTS

I thank Professors D. M. Bressoud, R. K. Guy, A. K. Lenstra, H. W. Lenstra, Jr., C. Pomerance, G. Robin, S. S. Wagstaff, Jr., and H. C. Williams for their sending me reprints concerning this subject or other topics in computational number theory. Special thanks go to the editors and the referees for kind and helpful comments that improved the presentation of this paper.

REFERENCES

1. D. M. Bressoud. *Factorization and Primality Testing*. New York: Springer-Verlag, 1989.
2. John D. Dixon. "Factorization and Primality Tests." *Amer. Math. Monthly* **91** (1984):333-352.
3. D. E. Knuth. *The Art of Computer Programming: Semi-Numerical Algorithms*. Vol. 2. 2nd ed. Reading, MA: Addison-Wesley, 1981.
4. A. K. Lenstra & H. W. Lenstra, Jr. *The Development of the Number Field Sieve*. Lecture Notes in Math., 1554. Berlin: Springer-Verlag, 1993.
5. H. W. Lenstra, Jr. "Factoring Integers with Elliptic Curves." *Ann. of Math.* **126** (1987):649-673.
6. J. M. Pollard. "Theorems on Factorization and Primality Testing." *Proc. Cambridge Philos. Soc.* **76** (1974):521-28.
7. C. Pomerance. "Factoring." *Proc. SIAM* **42** (1990):27-47.
8. H. Riesel. *Prime Numbers and Computer Methods for Factorization*. Boston: Birkhaüser, 1985.
9. K. H. Rosen. *Elementary Number Theory and Its Applications*. Reading, MA: Addison-Wesley, 1984.
10. R. D. Silverman. "The Multiple Pollynomial Quadratic Sieve." *Math. Comp.* **48** (1987):329-39.
11. H. C. Williams. "A $p+1$ Method of Factoring." *Math. Comp.* **39** (1982):225-34.
12. H. C. Williams & J. O. Shallit. "Factoring Integers before Computers." *Proc. of Symposia in Applied Mathematics* **48** (1994):481-531.
13. Zhenxiang Zhang. "Finding Strong Pseudoprimes to Several Bases." *Math. Comp.* **70** (2001): 863-72.

AMS Classification Numbers: 11Y05, 11A51, 11B39

❖❖❖