AN APPLICATION OF HIGHER RECIPROCITY TO COMPUTATIONAL NUMBER THEORY

(Abstract)

Leonard M. Adleman†
Robert McDonnell†

University of Southern California

I. INTRODUCTION

The Higher Reciprocity Laws are considered to be among the deepest and most fundamental results in number theory. Yet, they have until recently played no part in number theoretic algorithms. Their first use occured in [A1][APR], where they were of critical importance in the primality algorithm described there. In this paper we continue to explore the power of the laws in algorithms.

The problem we consider is part of a group of well studied problems about roots in finite fields and rings. Let $\mathbb{F}$ denote a finite field, let $\mathbb{R}$ denote a direct product of finite fields. Consider the following problems:

Problem 1. Is $X^n = a$ solvable in $\mathbb{F}$

Problem 2. If $X^n = a$ is solvable in $\mathbb{F}$ find X

Problem 3. Is $X^n = a$ solvable in $\mathbb{R}$

Problem 4. If $X^n = a$ solvable in $\mathbb{R}$ find X.

Problem 1. A polynomial time algorithms was already known to Legendre and Gauss by 1801.

Problem 2. For fixed n, a random polynomial (in the size of the field) time algorithm was discovered by Berlekamp [B] and improved by Rabin [R1]. If Extended Riemann hypothesis is assumed and n = 2 then a polynomial time algorithm exists [AMM] [S]. When n is not fixed, then the Berlekamp & Rabin algorithms are strictly exponential in n, but an algorithm for discrete logarithms [A2] yields a subexponential algorithm.

Problem 4. Rabin [R2] has shown that when n = 2, the problem is equivalent to factoring. This result has provided the most compelling evidence to date that a varient of the RSA public key cryptosystems [RSA] is secure. Rabin's result generalizes to establish that for all n, Problem 4 is equivalent to factoring

†Research supported by National Science Foundation Grant # MCS-8022533

those integers m with $n|\phi(m)$. Generalizing this result without the constraint that $n|\phi(m)$ is an important open problem which, among other things, would establish (up to the difficulty of factoring) the security of the RSA cryptosystem.

It is Problem 3 which this paper deals with. We will establish, under a weak technical assumption, that Problem 3 is as difficult as factoring "paired composites" i.e., those of the form $n = q\hat{q}$ with q, $\hat{q}$ prime. Our reduction does not run in polynomial time but rather $e^{O[\ell n \ell n(n) \ell n \ell n \ell n(n)^2]}$. (Those familiar with [A1][APR] will recognize that this is close to the current upper bound on primality testing). Formally we show:

Assumption A: There are natural numbers c,d such that for all natural numbers n:

1. If $z = d\ell n(\ell n(n))$ then
   a) $\prod_{i=1}^{z} p_i \leq e^{c\ell n \ell n(n) \ell n \ell n \ell n(n)}$, where p_i is the i^{th} prime.
   b) $p_z \leq c\ell n \ell n(n) \ell n \ell n \ell n(n)$
   c) $S = \{E | E-1 = p_1^{e_1} p_2^{e_2} \ldots p_z^{e_z}, e_i \in \{0,1\}, i = 1,2,\ldots,z\}$ then the product of all primes in S exceeds $n^{\frac{1}{2}}$.

Theorem: On Assumption A:

There is a natural number c and a Turing machine M with oracle for problem 3 (above) such that: On input $n = q\hat{q}$ with q and $\hat{q}$ prime, M outputs q and $\hat{q}$ within $e^{c[\ell n \ell n(n) \ell n \ell n \ell n(n)]^2}$ steps.

It follows from the above theorem (on Assumption A) that unless very fast algorithms for factoring exist, no fast algorithms for problem 3 exist.

The result has cryptographic significance in the following sense: there are variants of the RSA cryptosystem, where it is not only equivalent to factoring to decipher cryptograms, but is equivalent to factoring even to distinguish cryptograms from non-cryptograms.

II. OUTLINE OF PROOF

We will assume the reader is familiar with [A1] and [APR]. Since many of the ideas used in reducing Factoring of paired composites to problem 3 are the same as in the primality test of [A1] and [APR], we will only present an algorithm to take care of what is new (III below). The rest will be indicated by two sequences of reasoning (I and II below).

I. Assume $n = \hat{q}q$ with $q,\hat{q}$ prime and $q \leq n^{\frac{1}{2}}$. We begin by showing how q's identity can be broken into a number of very small pieces. Later using an oracle for problem 3 we will find these pieces. The algorithms used to reconstruct q from them are straightforward.

1. Let $p_0, p_1, \ldots, p_z$ be an initial segment of primes (z to be determined later). These primes will be called initial primes (e.g. $p_0 = 2$, $p_1 = 3$, $p_2 = 5$).

2. Let $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_w$ be all the primes formed by taking a product of a subset of initial primes and adding one. (e.g. $2+1 = 3$, $2\cdot3+1 = 7$, $2\cdot5+1 = 11$, $2\cdot3\cdot5+1 = 31$). Choose z above so that $\prod_{i=1}^{w} \varepsilon_i \geq n^{\frac{1}{2}}$. These primes will be called Euclidean primes.

3. To know q it is enough to know q MOD($\prod_{i=1}^{w} \varepsilon_i$) since $\prod_{i=1}^{w} \varepsilon_i \geq q$.

4. To know q MOD($\prod_{i=1}^{w} \varepsilon_i$) it is enough to know for each Euclidean prime ε, q MOD(ε) (this is the Chinese Remainder Theorem).

5. To know q MOD(ε) it is enough to know for some generator g of $Z^*/(\varepsilon)$ the index of q with respect to g in $Z^*/(\varepsilon) \triangleq \text{IND}(q,\varepsilon,g)$.

6. To know IND(q,ε,g) it is enough to know IND(q,ε,g)MOD(p_i) for each initial prime p_i which divides $\varepsilon-1$. (Since $Z^*/(\varepsilon)$ is cyclic of order $\varepsilon-1$, and $\varepsilon-1$ is a product of initial primes by construction).

7. Thus by the above, to know q it is enough to know for each Euclidean prime ε and each initial prime p with $p|\varepsilon-1$, IND(q,ε,g)MOD(p). These are the pieces of q's identity which the algorithm will find. We will classify the pieces as follows:

quadratic pieces	IND(q,ε,g)MOD(2) for all ε
cubic pieces	IND(q,ε,g)MOD(3) for all ε with $3|\varepsilon-1$
⋮	
p^{th}-ic pieces	IND(q,ε,g)MOD(p) for all ε with $p|\varepsilon-1$

II. We now consider a sequence of identities which establish a connection between the p^{th}-ic pieces we want to find and certain values of $(-)_p$, the p^{th} power residue symbols.

1. Let a,b be natural numbers such that

   a) a,b, a+b $\not\equiv$ 0 MOD(p)

   b) $\tilde{\theta}_p = \sum_{x=1}^{p-1} \left(\frac{\lceil (a+b)x \rceil}{p} - \frac{\lceil ax \rceil}{p} - \frac{\lceil bx \rceil}{p}\right)x^{-1} \not\equiv 0$ MOD(p)

   where x^{-1} is inverse of x in $Z^*/(p)$ (see [A1][APR] for explanation)

For each Euclidean prime ε assume we have associated a generator g. For each Euclidean prime ε with $p|\varepsilon-1$ let J = J(ε,g,p,a,b) denote the Jacobi sum $J = -\sum_{x=2}^{\varepsilon-1} \zeta_p^{-(\text{IND}(x,\varepsilon,g)a+\text{IND}(1-x,\varepsilon,g)b)}$ where ζ_p is a primitive p^{th} root of unity.

Then:

1. $(\frac{J}{q})_p = (\frac{q}{J})_p$ by reciprocity (trivial Hilbert symbol)

2. $(\frac{q}{J})_p^{\tilde{\theta}_p^{-1}} = \left(\frac{q}{(\varepsilon,\zeta_p - g^{\varepsilon-1/p})}\right)_p$ by a Theorem of Stickleberger where $\tilde{\theta}_p^{-1}$ is inverse of $\tilde{\theta}_p$ in $Z^*/(p)$

3. $\left(\frac{q}{(\varepsilon,\zeta_p - g^{\varepsilon-1/p})}\right)_p = (\frac{q}{\varepsilon})_p$ by an isomorphism described in [A1] [APR]

4. $(\frac{q}{\varepsilon})_p = \zeta_p^{IND(p,\varepsilon,g)MOD(p)}$

Thus by I and II, if we can compute all of the $(\frac{J}{q})_p$'s we can reconstruct q and factor n. We next give an algorithm with oracle for problem 3, which accomplishes this.

III. Algorithm

1. For each Euclidean prime ε find a generator g.
2. For each initial prime p find a,b as in II above.
3. For each Euclidean prime ε with $p|\varepsilon-1$ calculate the corresponding Jacobi sum described in II. Denote these $J_1, J_2, \ldots, J_m$.
4. Do Stages 1 through m below.

Stage 1 Set $S_1 = S_1(p) = J_1$

Stage 2 Case I If there is a natural number k_1 less than p such that

$$S_1^{k_1} J_2^{p-1} = X^p MOD(n)$$

has a solution (here the oracle is used) then set

$$S_2 = S_2(p) = 1$$

Case II If Case I does not hold then

$$S_2 = S_2(p) = J_2$$

Stage i Case I If there exists natural numbers $k_1, k_2, \ldots, k_{i-1}$ less than p such that

$$S_1^{k_1} S_2^{k_2} \ldots S_{i-1}^{k_{i-1}} J_i^{p-1} = X^p MOD(n)$$

then set

$$S_i = S_i(p) = 1$$

Case II If Case I does not hold then

$$S_i = S_i(p) = 1$$

Notice that if

$$S_1^{k_1} S_2^{k_2} \ldots S_{i-1}^{k_{i-1}} J_i^{p-1} = X^p \text{MOD}(n)$$

then

$$S_1^{k_1} S_2^{k_2} \ldots S_{i-1}^{k_{i-1}} J_i^{p-1} = X^p \text{MOD}(q)$$

then

$$\left(\frac{S_1^{k_1} S_2^{k_2} \ldots S_{i-1}^{k_{i-1}} J_i^{p-1}}{q} \right)_p = 1$$

from which it follows

$$\left(\frac{S_1^{k_1} S_2^{k_2} \ldots S_{i-1}^{k_{i-1}}}{q} \right)_p = \left(\frac{J_i}{q}\right)_p$$

Therefore the above algorithm provides us with the desired $(\frac{J}{q})_p$'s if we can find the $(\frac{S_i}{q})_p$'s. If $S_i = 1$ then this is trivial; if $S_i \neq 1$ then the algorithm will guess the value of $(\frac{S_i}{q})$ (remember q is unknown).

What remains is to calculate the cost of this guessing. The complexity of the entire reduction algorithm will be a polynomial in the number of guesses which must be made.

Consider the cyclotomic field $Q[\zeta_p]$ where p is an initial prime. Let $n = I_1 \ldots I_\ell$ be a decomposition of n into prime ideals. It is well known that $\ell \leq 2(p-1)$. Consider the map f from the integers of $Q[\zeta_p]$ which are relatively prime to n into the ℓ dimensional vector space over $Z^*/(p)$ given by $f(\alpha) = \langle \beta_1, \beta_2, \ldots, \beta_\ell \rangle$ where $(\frac{\alpha}{I_i})_p = \zeta_p^{\beta_i}$, $i = 1,2,\ldots,\ell$.

It is easy to see that if there are more than ℓ non trivial $S_i(p)$'s then there are more than ℓ independent vectors. From this it follows that there are at most 2(p-1) non trivial S's. Since each $(\frac{S}{q})_p$ can assume any of p values, the number of possible guesses for initial prime p is $p^{2(p-1)}$. Thus using Assumption A above the total number of guesses is

letting $z = d\ell n \ell n(n)$

$$\prod_{i=1}^{z} p_i^{2(p_i-1)} \leq \left(\prod_{i=1}^{z} p_i \right)^{2(p_z)} \leq$$

$$[e^{c\ell n\ell n(n)\ell n\ell n\ell n(n)}]^{2c\ell n\ell n(n)\ell n\ell n\ell n(n)}$$

which gives the result.

III. WORK REMAININIG

The intention of this research is to establish the usefulness of higher reciprocity in algorithms, rather than to prove the most powerful version of the Theorem presented here. Nonetheless, the Theorem could be strengthened in the following ways:

1. Remove assumption A.
2. Factor all composites not just those composed of 2 primes.
3. Make the reduction polynomial time.

1. seems quite feasible (see [APR]), 2 is probably doable, 3 seems quite hard.

As for reciprocity itself, it seems likely that it will play a central role in diophantine complexity.

REFERENCES

[A1] L.M. Adleman, "On Distinguishing Prime Numbers from Composite Numbers" (Abstract), Proc. 21st Annual IEEE Symposium on Foundations of Computer Science (1980), 387-406.

[A2] L.M. Adleman, "Subexponential Algorithm for The Discrete Logarithm Problem," Proc. 20th Annual IEEE Symposium on Foundations of Computer Science (1979).

[AMM] L.M. Adleman, K. Manders and G. Miller, "On Taking Roots in Finite Fields," Proc. 18th Annual IEEE Symposium on Foundations of Computer Science (1977).

[APR] L.M. Adleman, C. Pomerance, and R.S. Rumley, "On Distinguishing Prime Numbers from Composite Numbers," to appear Annals of Mathematics.

[B] E.R. Berlekamp, "Factoring Polynomials Over Large Finite Fields," Math. Comp. 24, pp. 713-735, (1970).

[R1] M.O. Rabin, "Probabilistic Algorithms," in J. Traub, Ed., Algorithms and Complexity, New Directions and Recent Results, Academic Press (New York, 1976), pp. 21-24.

[R2] M.O. Rabin, "Digitalized Signatures and Public Key Functions as Intractible as Factorization," MIT/LCS/TR-212, Technical Memo MIT (1979).

[RSA] R. Rivest, A. Shamir and L.M. Adleman, "A Method for Obtaining Digital Signatures and Public Key Cryptosystems," Communications of the ACM, February 1978.

[S] D. Shanks, "Five Number Theoretic Algorithms," 2nd Manitoba Conference on Numerical Math. and Computing (1972).