ON TAKING ROOTS IN FINITE FIELDS

Leonard Adleman[1]
Massachusetts Institute of Technology

Kenneth Manders[2]
University of California, Berkeley

Gary Miller[3]
University of Rochester

INTRODUCTION

Among the most important concepts in number theory is that of quadratic residue.

> For all relatively prime a, m $\in \mathbb{N}$ a is a *quadratic residue modulo* m if and only if $x^2 \equiv a$ MOD(m) has a solution (if not, a is a *quadratic non-residue*).

From purely computational considerations the concept is crucial, being of central importance in all recent primality algorithms [12], [16], [18] and in methods for factoring polynomials over finite fields [5], [6].

How hard is it to decide if a is a quadratic residue modulo m? Few problems have received more attention [7]. When m is prime, the Legendre symbol, the Jacobi symbol and the Gaussian Law of Quadratic Reciprocity yield a polynomial time algorithm [14]. When m is composite, then the above result for primes together with the Chinese Remainder Theorem yields a polynomial time algorithm assuming m can be factored (thus this problem is probably not γ- or NP-complete). Finding x's sucb that $x^2 \equiv a$ MOD(m) when a is a quadratic residue is a far more complex problem. In [11] it was shown that finding the least x such that $x^2 \equiv a$ MOD(m) is NP-complete (even if m is factored). The main result of this paper is:

THEOREM I [Assuming Extended Riemann Hypothesis]. *There is a deterministic polynomial time algorithm which on inputs* a, p $\in \mathbb{N}$, *where* p *is prime, outputs the least* x $\in \mathbb{N}$ *such that* $x^2 \equiv a$ MOD(p) *(or "no" if* a *is a quadratic non-residue modulo* p*).*

The history of this problem is quite interesting [7]. The first reference to it is by Bhascara Acharya (1150 AD) who considered the very special case $x^2 \equiv 30$ MOD(7). General methods have been discussed by many great mathematiciains including Lagrange, Legendre, Gauss, Dirichlet and Lebesgue. However, among recent algorithms it appears that those of Berlekamp [5], [6] (as part of a general method to factor polynomials) and Lehmer [10] are best. Both algorithms run in random polynomial time.[†]

Both are based on a clever trick, where, roughly speaking, the problem $x^2 \equiv a$ MOD(p) is transformed into a new problem $(x+c)^2 \equiv a$ MOD(p) which because of the quadratic character of the roots is easy to solve. This trick seems to rely on very subtle features of the additive structure of numbers modulo p. As a consequence there seems to be no direct way to apply existing number theoretic results to determine if these algorithms run in deterministic polynomial time. Our algorithm also runs in random polynomial time but since it relies directly on features of the multiplicative structure of numbers modulo p, deep number theoretic results due to Ankeny [4] apply. (See below.)

The next theorem contrasts a consequence of Theorem I with a result from [11].

THEOREM II. *Let* P_1 *be the problem of finding an* x $\in \mathbb{N}$ *such that* $x^2 \equiv a$ MOD(m) *from inputs* a, m $\in \mathbb{N}$, *where* a *is a quadratic residue modulo* m, *and* m *is presented fully factored. Let* P_2 *be the problem of finding the least* x $\in \mathbb{N}$ *such that* $x^2 \equiv a$ MOD(m) *from inputs* a, m $\in \mathbb{N}$, *where* a *is a quadratic residue modulo* m, *and* m *is presented fully factored.*

a) [*Assuming Extended Riemann Hypothesis*] P_1 *is in deterministic polynomial time.*

b) P_2 *is* NP-*complete.*

To solve P_1, use Theorem I to solve $x^2 \equiv a$ MOD(p) for each prime divisor of m. If p^α, $\alpha > 1$ is the largest power of p dividing m, solve $x_k^2 \equiv a$ MOD(p^k), $k = 2,3,\ldots,\alpha$ as

$$x_k = x_{k-1} + v_k p^{k-1} , \qquad 2x_{k-1} v_k p^{k-1} + (x_{k-1}^2 - a) \equiv 0 \text{ MOD}(p)$$

(see [14], Ch. 2.6), and use the Chinese Remainder Theorem to find the solution MOD(m). The difficulty of P_2 seems to be that we have no way of predicting the size of the solution obtained from the Chinese Remainder Theorem in relation to the choice of solution to $x^2 \equiv a$ MOD(p^α). (This phenomenon is reminiscent of the difficulty of comparing the sizes of numbers represented in a residue system [9].)

Our methods extend to give the following results:

THEOREM III [Assuming Extended Riemann Hypothesis]. *For all* n, *there is a deterministic polynomial time algorithm which on inputs* a, p $\in \mathbb{N}$, *where* p *is prime, outputs the least* x $\in \mathbb{N}$ *such that* $x^n \equiv a$ MOD(p) *(or "no" if no such* x *exists).*

THEOREM IV [Assuming Extended Riemann Hypothesis]. *There is a deterministic algorithm running in time* $O(n \log^c(p+a))$ *for some* $c > 0$ *such that on inputs* a, p, n $\in \mathbb{N}$ *where* p *is prime outputs the least* x $\in \mathbb{N}$ *such that* $x^n \equiv a$ MOD(p) *(or "no" if no such* x *exists).*

Extensions of Theorem IV to include finding roots of arbitrary polynomials of degree n would be of considerable interest. Recent results due to Plaisted [15] seem to indicate that these algorithms must be exponential in n. In this direction, it is straightforward to use Theorem I and the quadratic formula to get:

THEOREM V [Assuming Extended Riemann Hypothesis]. *There is a deterministic polynomial time algorithm which on inputs* a,b,c,p $\in \mathbb{N}$, *where* p *is prime, outputs both roots of the equation* $ax^2 + bx + c = 0$ MOD(p) *(or "no" if the equation has no roots).*

[1] Supported in part by Office of Naval Research contract N00014-67-0204-0063.

[2] Supported by National Science Foundation grant DCR72-03725-A02.

[3] Supported by National Research Council grant NRC-A5549.

[†] Intuitively, we mean that the algorithm, when modified to use a random number generator, will always output the correct answer and that the algorithm has a high probability (independent of the input) of termination within polynomial time. This is a seemingly stronger notion than that of Strassen and Solovay [18], who only require that the output be correct with a high probability. For a more precise definition see [3]; for a discussion of the relationship between these two types of definitions see [13].

Because of the explicit formulas for solutions to polynomials of degree up to four it is likely that a similar result can be proved for these cases by applying Theorem III. Unfortunately further generally valid extensions by this technique are impossible as a consequence of Galois' famous results (in cases in which solution formulas for equations of higher degree of special forms do exist, the algorithm might again be extended).

THE ALGORITHM

The following three easy lemmas are central to the algorithm:

LEMMA I. *For all* $N, K, p, a \in \mathbb{N}$ *with* $p-1 = 2^K(2N+1)$ *and* p *prime:*

If $a^{2N+1} \equiv 1 \text{ MOD}(p)$ *then* a^{N+1} *is a solution to* $x^2 \equiv a \text{ MOD}(p)$.

PROOF. $(a^{N+1})^2 \equiv a^{2N+2} \equiv (a^{2N+1})(a) \equiv (1)(a) \equiv$ $\equiv a \text{ MOD}(p)$.

LEMMA II. *For all* $N, K, p, a, g, J \in \mathbb{N}$ *with* $p-1 = 2^K(2N+1)$, p *prime, and* g *a quadratic non-residue modulo* p:

If $a^{2^J(2N+1)} \equiv 1 \text{ MOD}(p)$ *and* $a^{2^{J-1}(2N+1)} \not\equiv 1 \text{ MOD}(p)$ *where* $1 \leq J < K$ *then*

$$(ag^{2^{K-J}})^{2^{J-1}(2N+1)} \equiv 1 \text{ MOD}(p) .$$

PROOF. $(ag^{2^{K-J}})^{2^{J-1}(2N+1)} \equiv (a^{2^{J-1}(2N+1)}) \cdot (g^{2^{K-1}(2N+1)}) \equiv (-1)(-1) \equiv 1 \text{ MOD}(p)$.

LEMMA III. *For all* $m, g, a, b, y \in \mathbb{N}$, $y \equiv 0 \text{ MOD}(2)$:

If $b^2 \equiv ag^y \text{ MOD}(m)$ *then* $(b(g^{y/2})^{-1})^2 \equiv a \text{ MOD}(m)$.

PROOF. $(b(g^{y/2})^{-1})^2 \equiv b^2(g^{y/2})^{-1}(g^{y/2})^{-1} \equiv$ $\equiv ag^y(g^{y/2})^{-1}(g^{y/2})^{-1} \equiv a(g^{y/2})(g^{y/2})(g^{y/2})^{-1}(g^{y/2})^{-1}$ $\equiv a \text{ MOD}(m)$.

The algorithm begins by finding the least number J_1 such that $a^{2^{J_1}(2N+1)} \equiv 1 \text{ MOD}(p)$. If $J_1 = 0$ then the algorithm finds a square root by application of Lemma I. If not it uses Lemma II to construct a new number a_2 for which the least number J_2 such that $a_2^{2^{J_2}(2N+1)} \equiv 1 \text{ MOD}(p)$ is guaranteed to be such that $J_2 < J_1$. It continues in this fashion until for some a_i, $J_i = 0$. It then uses Lemma I to find a square root b_i of a_i. It finally uses Lemma III to extract a square root of a from b_i.

To carry out this process the algorithm needs a quadratic non-residue modulo p. The following deep result due to Ankeny [4] guarantees that this can be found in polynomial time.

THEOREM (Ankeny) [Assuming Extended Riemann Hypothesis]. *There exists a* $c > 0$ *such that for all primes* p *the least quadratic non-residue modulo* p *is less than* $c(\log p)^2$. *(Recent results obtained by P. Weinberger* [20] *suggest that* $c < 4$.

Below is a more precise version of the procedure. "On input a,p

I Use the Euclidean Algorithm and Miller's Algorithm [12,13] to reject the input if either a and p are not relatively prime or if p is not prime.

II If $a^{\frac{p-1}{2}} \not\equiv 1 \text{ MOD}(p)$ then output "a is not a quadratic residue modulo p" (see [14]) and halt.

III Find the least g such that $g^{\frac{p-1}{2}} \not\equiv 1 \text{ MOD}(p)$ (this g is a quadratic non-residue and by Ankeny's results this step is carried out in polynomial time).

IV Compute K, N such that $p-1 = 2^K(2N+1)$.

V Set $L = 1$.

VI Find the least J such that $a^{2^J(2N+1)} \equiv 1 \text{ MOD}(p)$.

VII If $J = 0$ (Lemma I applies) then set $D = a^{N+1}$ and go to VIII. If $J \neq 0$ (Lemma II applies) then set $a = ag^{2^{K-J}}$, set $L = L \cdot g^{2^{K-J-1}}$ and go to VI. (Note: Step II and Lemma II assure $J < K$).

VIII Compute (using Euclidean Algorithm) the inverse L^{-1} of L MOD (p), output $\text{MIN}\{D \cdot L^{-1} \text{ MOD}(p), -(DL^{-1}) \text{ MOD}(p)\}$.

We outline a second proof of Theorem I, which provides further motivation for Lemma II, and for the more general construction of Theorems III and IV. First we note a principle which allows a reduction of the general problem of finding a q-th root modulo p to the problem of finding a q-th root modulo p of a number whose order is a power of q, i.e. $b^{2^\alpha} \equiv 1 \text{ MOD}(p)$ and 2^α is the smallest exponent for which this holds.

LEMMA IV. *If* $a^{st} \equiv 1 \text{ MOD } p$; $(s,t) = 1$, *then for any solution* $v, w \in \mathbb{Z}$ *of* $vs + wt = 1$, *we have for* $b \equiv a^{wt} \text{ MOD}(p)$, $c \equiv a^{vs} \text{ MOD}(p)$:

$$bc \equiv a \text{ MOD}(p)$$
$$b^s \equiv 1 \text{ MOD}(p) \textit{ and } (O(a),s) = (O(b),s)$$
$$c^t \equiv 1 \text{ MOD}(p) \textit{ and } (O(a),t) = (O(c),t)$$

where $(x,y) = \gcd\{x,y\}$, *and* $O(x)$ *is the order of* x *in the multiplicative group of integers modulo* p *and prime to* p.

The proof is obvious. As small v, w exist, b and c can be obtained quickly.

Let us take (without loss of generality) $q = 2$ in the following. With any prime number p, we can associate the binary tree with root -1 and such that the immediate descendants of any node b, $1 < b < p$, b a quadratic residue mod p, are the two solutions of $x^2 \equiv b \text{ MOD}(p)$, $0 < x < p$. Thus the leaves of the tree are quadratic nonresidues mod p. All numbers occurring as nodes in the tree have order 2^α for some $\alpha > 0$.

By Lemmas IV and I we can now reduce the problem of solving $x^2 \equiv a \text{ MOD}(p)$ to the case where a occurs in the tree associated with p; this problem can in turn be solved by finding a leaf x in the subtree with root a and squaring x until we obtain one of the immediate descendants of a, which is the desired solution. Ankeny's result shows that we can quickly find a quadratic non-residue modulo p, which by Lemma IV can again be reduced to a leaf ℓ of the tree. If ℓ does not lie in the subtree with root a, let b be the root of the minimal subtree containing ℓ and a; say $\ell^{2^L} \equiv b \text{ MOD}(p)$. By Lemma II we can now find a c such that $c^{2^{L-1}} \equiv -1 \text{ MOD}(p)$; ℓc is still a quadratic nonresidue modulo p and ℓc and a are contained in a strictly smaller subtree than ℓ and a. By repeating this process, we find a leaf in the subtree with root a, as desired.

For Theorems III and IV we modify the algorithm as follows: To extract an n-th root, we factor gcd(n,p-1) and repeatedly extract a q-th root, where q is a prime dividing gcd(n,p-1). Finally we extract an n/gcd(n,p-1)-th root. This last problem can be solved immediately, using a generalization of Lemma I:

LEMMA I'. *For any* $n,N,N',a,p,h,K \in \mathbb{N}$ *with* p *prime,* $p-1 = n^K(nN+N')$, $0 < N' < n$, $(n,nN+N') = 1$ *and* $h = n^{-1}$ MOD(nN+N')*:*

If $a^{nN+N'} \equiv$ MOD(p)*, then*

a^h *is a solution to* $x^n \equiv a$ MOD(p) .

In this case we have K = 0 and clearly the conditions of Lemma I' can be satisfied. Thus we can in the following restrict ourselves to consideration of the problem:

$$x^q \equiv a \text{ MOD}(p), \quad q,p \text{ primes}, \quad q|p-1 .$$

To solve this problem, we apply the original algorithm; steps I-VI are modified in the obvious fashion replacing "2" by "q" and "quadratic (non-)residue" by "q-th (non-)residue" and "2N+1" by "qN+N', 0 < N' < q". Ankeny's theorem applies to the size of the least q-th nonresidue equally well [4], so step III runs in polynomial time.

In step VII, if J = 0 we apply Lemma I' instead of I; if J > 0 then we use a similar generalization of Lemma II, with "$g^{2^{K-J}}$" replaced by "$g^{2^{K-j}\cdot\lambda}$", where λ is determined by trial and error to satisfy

$$(\lambda,n) = 1, \quad 0 < \lambda < n, \quad \lambda \in \mathbb{N}$$
$$a^{2^{J-1}(2^{N+N'})} \cdot g^{\lambda\cdot 2^{K-1}(2N+N')} \equiv 1 \text{ MOD}(p)$$

Finally, step VIII is carried out as before, with "q" replacing "2" in Lemma III.

Remark. We indicate a different method for finding roots. The algorithms produced by this method are less efficient than those given; however they exemplify a strategy (using primitive roots) which may be more generally applicable in obtaining extensions of these results.

In the algorithms above, take g to be a primitive root modulo p rather than an arbitrary q-th nonresidue. This can be done, using

THEOREM (Wang [19]) [Extended Riemann Hypothesis] *For some* c > 0, *and any prime* p*: The least primitive root modulo* p *is less than* $c(\log p)^2\#^6$*, where # is the number of prime factors of* p-1.

Unfortunately, there is no known deterministic polynomial time algorithm to decide whether g is in fact a primitive root modulo p, so step III of the algorithm will not work. Instead, steps IV-VIII (as modified for q-th roots) are run for each number less than $c(\log p)^8$ until a correct result is obtained; this can be verified by substitution in the congruence.

OPEN PROBLEMS

1. Are the problems of
   a) Primality testing
   b) Solving $x^2 \equiv a$ MOD(p), p a prime (inputs: a,p)

   deterministic polynomial time interreducible (without ERH, of course)? This type of question might lead to more insight into the role played by ERH: Is ERH being put to a different use in the two problems?

2. Is the problem of finding a solution of $x^2 \equiv a$ MOD(m), where m is an arbitrary natural number and the prime factorization of m is not given, in deterministic polynomial time? Compare Theorem II. If not, is factorization "required" to solve this problem? That is, can the problem be solved in deterministic polynomial time using an oracle for a problem Q such that factorization cannot be done in deterministic polynomial time with an oracle for Q. (Only a plausibility argument can be expected at present for the latter claim.)

3. Consider the problem

   "on input <p,y,g>, p prime, 0 < y < p, g primitive root modulo p, find x: $0 \leq x \leq p-1$, $g^x \equiv y$ MOD(p)."

   Is there a deterministic polynomial time algorithm to do this? Assume ERH.

The last two problems have practical as well as theoretical interest because of recent developments in cryptology. See [8] and [17].

ACKNOWLEDGMENT

The authors thank Dana Angluin for helpful conversations and encouragement.

REFERENCES

1. Adleman, L., Number Theoretic Aspects of Computational Complexity, Ph.D. Thesis, Berkeley (1976).
2. Adleman, L. and Manders, K., Diophantine Complexity, Conf. Rec. 17th Annual IEEE Symp. Foundations of Computer Science (1976), 81-88.
3. Adleman, L. and Manders, K., Reducibility, Randomness, and Intractability, Proc. 9th Annual ACM Symp. on Theory of Computing (1977), 151-163.
4. Ankeny, N., The Least Quadratic Non-Residue, Annuals of Mathematics 55 (1952), 65-72.
5. Berlekamp, E.R., Algebraic Coding Theory, McGraw-Hill, New York (1968).
6. Berlekamp, E.R., Factoring Polynomials Over Large Finite Fields, Mathematics of Computation 24 (1970), 713-735.
7. Dickson, L., History of the Theory of Numbers, Chelsea Publishing Co., New York (1952, 1966).
8. Diffie, W. and Hellman, M., New Directions in Cryptography, IEEE Trans. Information Theory (Nov. 1976).
9 Knuth, D., Seminumerical Algorithms, Vol. 2 of The Art of Computer Programming, Addison-Wesley (1969).
10. Lehmer, D., Computer Technology Applied to the Theory of Numbers, Studies in Number Theory, Mathematics Association of America (1969), 117-151.
11. Manders, K. and Adleman, L., NP-Complete Decision Problems for Quadratic Polynomials, Proc. 8th Annual ACM Symp. on Theory of Computing (1976), 23-29.
12. Miller, G.L., Riemann's Hypothesis and Tests for Primality, Ph.D. Thesis, Berkeley (1975).
13. Miller, G.L., Riemann's Hypothesis and Tests for Primality, J. Computer and System Sciences 13 (1976), 300-317.
14. Niven, I. and Zuckerman, H., An Introduction to the Theory of Numbers, John Wiley & Sons (1972).
15. Plaisted, D., New NP-hard and NP-complete Polynomial and Integer Divisibility Problems, Conf. Rec. 18th Annual IEEE Symp. Foundations of Computer Science (1977).
16. Rabin, M.O., Probabilistic Algorithms, in Algorithms and Complexity, New Directions and Recent Results, J. Traub (ed.), Academic Press, 21-40.

17. Rivest, R.L., Shamir, A. and Adleman, L., On Digital Signatures and Public-key Cryptosystems, Laboratory for Computer Science, M.I.T. Report MIT/LCS/TM-82 (April 1977).

18. Strassen, V. and Solovay, R., Fast Monte-Carlo Tests for Primality, SIAM Journal on Computing (1977), 84-85.

19. Wang, Y., On the Least Primitive Root of a Prime, Sci. Sinica 10 (1961), 1-14.

20. Weinberger, P. (Private Communication), Also informal presentation at the Asilomar Number Theory Conference (Dec. 1975).