Efficient Computation of Roots in Finite Fields

PAULO S. L. M. BARRETO* (pbarreto@larc.usp.br)
Laboratório de Arquitetura e Redes de Computadores (LARC), Escola Politécnica, Universidade de São Paulo, Brazil.

JOSÉ FELIPE VOLOCH† (voloch@math.utexas.edu)
Department of Mathematics, University of Texas, Austin TX 78712, USA.

Abstract. We present an algorithm to compute r-th roots in $\mathbb{F}_{q^m}$ with complexity $O((\log m + r \log q)m^2 \log^2 q)$ for certain choices of m and q. This compares well to previously known algorithms, which need $O(rm^3 \log^3 q)$ steps.

Keywords: finite fields, root computation, efficient algorithms

1. Introduction

Calculation of roots in finite fields $\mathbb{F}_{q^m}$ (where $q = p^d$ for some prime p and some $d > 0$) is a classical problem in computational algebra and number theory. Besides its intrinsic interest, it plays an essential role in cryptosystems based on elliptic curves and other Abelian varieties, where plain exponentiation (fundamental for more conventional cryptosystems) is not relevant.

A typical application of root extraction is point compression in elliptic curves. In general, a point (x, y) on a curve $E(\mathbb{F}_{q^m})$ is compressed as (x, β) where $\beta \in \mathbb{Z}_2$ is a single bit from y; the full y value is recovered from (x, β) by solving for y the curve equation $y^2 = P(x)$, which involves computing a square root $\sqrt{P(x)}$. Under certain circumstances it may be more convenient to compress (x, y) as (α, y) where $\alpha \in \mathbb{Z}_3$ is a single trit from x; the full x value is recovered by solving the curve equation for x rather than y. This is the case when taking a cubic root is more efficient than taking a square root, for instance, when the curve equation is $y^2 = x^3 + c$ over $\mathbb{F}_{q^m}$ for some c (in this case, our new algorithm is most suitable for odd m and a Fermat prime q). A similar situation arises in the operation of hashing onto elliptic curves, as needed by a large number of cryptosystems, notably pairing-based schemes [4, 5, 10, 12]. The method usually consists of mapping messages onto (x, β) or (α, y) pairs as above by applying a conventional hash function, then solving the curve equation to get a complete point.

* Supported by Scopus Tecnologia S. A.
† Supported by NSA grant MDA904-03-1-0117

Higher order roots appear in analogous settings over hyperelliptic and superelliptic curves [6].

Taking r-th roots in a finite field $\mathbb{F}_{q^m}$ is most commonly computed by means of the Adleman-Manders-Miller algorithm [1] (see also [2, section 7.3]), which extends Tonelli's square root algorithm. The complexity of the Adleman-Manders-Miller algorithm is $O(rm^4 \log^4 q)$ steps in general, but for certain special fields $\mathbb{F}_q$ this drops to $O(rm^3 \log^3 q)$ steps if r is fixed and small. Cipolla's square-root algorithm attains complexity $O(m^3 \log^3 q)$ for any finite field $\mathbb{F}_{q^m}$, but does not seem to admit of a simple generalization for higher order roots.

Computing generic powers, on the other hand, is a problem for which many efficient algorithms are known. Particularly efficient algorithms are described in [7], with complexity $O(m^2 \log\log m \log q)$ if q is a small prime. Other specialized improved algorithms for exponentiation are given in [13]. It is natural to ask if similar methods could be used for root calculation since one can compute $\sqrt[r]{x}$ as x^v with $v \equiv r^{-1} \pmod{q^m - 1}$ when $(r, q^m - 1) = 1$, and what can be done when r is not invertible modulo $(q^m - 1)$ as in this case not all field elements have r-th roots.

Our contribution in this paper is to show, for a large family of prime powers q and extension degrees m, how to take advantage of the periodic structure of v written in base q to compute r-th roots in $\mathbb{F}_{q^m}$. The complexity of the resulting scheme is $O((\log m + r \log q) m^2 \log^2 q)$ due to a divide-and-conquer strategy; we conjecture that algorithms even more efficient are possible using more complex addition chains. Furthermore, we explore extensions of this strategy that work, under certain circumstances, when $(r, q^m - 1) \neq 1$.

The basic idea of the new algorithm stems from the Itoh-Tsujii [9] algorithm for computing multiplicative inverses and was already used in [3] to efficiently compute square roots. The Itoh-Tsujii algorithm is described in depth in [8], where the reader will also find a discussion of implementation aspects; that discussion applies equally well to our algorithm.

2. The New Algorithm

Our new algorithm computes r-th roots in $\mathbb{F}_{q^m}$ provided that q, m, r satisfy certain constraints. Our fundamental strategy is to seek a simple expression for $v \equiv r^{-1} \pmod{q^m - 1}$ so as to reduce the complexity of computing $\sqrt[r]{x}$ as x^v in $\mathbb{F}_q$.

Efficiently taking roots that are powers of the characteristic p in $\mathbb{F}_{q^m}$ is straightforward. Notice that, since raising to a power of p is a

linear bijection in characteristic p, the complexity of such operation is no larger than that of multiplication, namely, $O(m^2 \log^2 q)$, and under certain circumstances can be much smaller [11, chapter 6].

For certain primes $q > 2$ and odd m, efficient computation of square roots has been described in a previous work [3], and generalizing that method to roots that are higher powers of 2 is immediate. It is possible to compute roots by using exponent periodicity for more general q, m, r. This is established by the results below. First we state as a separate lemma the divide-and-conquer strategy to using periodicity in the algorithm.

LEMMA 1. *Let $\mathbb{F}_{q^m}$ be a finite field of characteristic p and let s be a power of p. Define the map $\phi_n : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$, $y \mapsto y^{1+s+\cdots+s^n}$ for $n \in \mathbb{N}^*$. We can compute $\phi_n(y)$ with $O(\log n)$ multiplications and raisings to powers of p.*

Proof. If $n = 2k+1$, then $y^{1+s+\cdots+s^n} = y^{1+s+\cdots+s^k}(y^{1+s+\cdots+s^k})^{s^{k+1}}$ and if $n = 2k$ then $y^{1+s+\cdots+s^n} = y^{1+s+\cdots+s^{k-1}}(y(y^{1+s+\cdots+s^{k-1}})^s)^{s^k}$. The result follows by iterating this procedure, halving the value of n each time; this can be done at most $\max(\lfloor \lg n \rfloor, 1)$ times, taking no more than 2 multiplications and 2 raisings to powers of s at each step. □

We will show that we can reduce root extraction to an operation of the form $x \mapsto x^a \phi_n(x^b)$ for small a, b and apply the above lemma.

2.1. Taking roots by inverting the exponent

We first tackle root extraction in the case $(r, q-1) = 1$. The periodic structure of $r^{-1} \pmod{q^m - 1}$ leads to an efficient r-th root algorithm, as established by the following considerations.

LEMMA 2. *Given q and r with $(q(q-1), r) = 1$, let $k > 1$ be the order of q modulo r. For any $m > 0$, $(m, k) = 1$, let u, $1 \leqslant u < r$ satisfy $u(q^m - 1) \equiv -1 \pmod{r}$ and $v = \lfloor q^m u / r \rfloor$. Then $rv \equiv 1 \pmod{q^m - 1}$. In addition, $v = a + b\sum_{j=0}^{n-1} q^{jk}, a, b < q^{2k}, n = \lfloor m/k \rfloor$.*

Proof. Note that $(q^m - 1, q^k - 1) = q - 1$ since m and k are coprime. As r divides $q^k - 1$ but is coprime to $q - 1$ we conclude that $q^m - 1$ and r are coprime, and u therefore is well-defined. Thus $u(q^m - 1) = vr - 1$ for some integer v and since $q^m u / r = v + (u-1)/r$ we have that $v = \lfloor q^m u / r \rfloor$ and from the equation $u(q^m - 1) = vr - 1$ it follows that $rv \equiv 1 \pmod{q^m - 1}$.

Finally, let $z = u(q^k - 1)/r$. Then z is an integer and $z < q^k - 1$. Now,

$$q^m u/r = q^m z/(q^k - 1) = q^{m-k} z \sum_{j=0}^{\infty} q^{-jk} = q^{m-k} z \sum_{j=0}^{n-1} q^{-jk} + \alpha,$$

say. Thus $v = zq^{m-nk} \sum_{j=0}^{n-1} q^{jk} + \lfloor \alpha \rfloor$ and we take $a = \lfloor \alpha \rfloor, b = zq^{m-nk}$, which completes the proof. □

Remark 1. To compute u one can replace m by its least residue modulo k since $q^k \equiv 1 \pmod{r}$. Also u/r has a periodic expansion in base q of period k and therefore $v = \lfloor q^m u/r \rfloor$ also has a periodic expansion in base q with the same period for all m in a fixed residue class modulo k.

THEOREM 1. *Let q be a prime power, let $r > 1$ be such that $(q(q-1), r) = 1$ and let $k > 1$ be the order of q modulo r. For any $m > 0$, $(m, k) = 1$, the complexity of taking r-th roots in $\mathbb{F}_{q^m}$ is $O((\log m + r \log q) m^2 \log^2 q)$.*

Proof. By Lemma 2, $r^{-1} \equiv a + b \sum_{j=0}^{n-1} q^{jk} \pmod{q^m - 1}$ for some $a, b < q^{2k}$ depending only on $m \pmod{k}$ and $n = \lfloor m/k \rfloor$. By Lemma 1, raising to the power $\sum_{j=0}^{n-1} q^{jk}$ takes $O(\log n)$ multiplications and raisings to powers of p. The remaining work essentially consists of raising to the powers a and b, each operation taking $O(k \log q)$ multiplications due to the bound on the exponents. The total computation cost is therefore $O(\log m + r \log q)$ operations of complexity $O(m^2 \log^2 q)$, from which the claimed overall complexity follows. □

Remark 2. The complexity simplifies to $O(rm^2 \log^3 q)$ if $r \log q \gtrsim \log m$, and to $O(m^2 \log m \log^2 q)$ if $r \log q \lesssim \log m$. In either case it compares well to the $O(rm^3 \log^3 q)$ complexity of the Adleman-Manders-Miller algorithm at its best.

2.2. Taking r-th roots when r is not invertible

We now deal with root extraction when $r \mid (q-1)$. The analogous result to Lemma 2 above is the following. Note that not every element of $\mathbb{F}_{q^m}$ is a r-th power, so we need to work in the subgroup of order $(q^m - 1)/r$ of $\mathbb{F}_{q^m}^*$.

LEMMA 3. *Given q and r with $r \mid (q-1)$ and $((q-1)/r, r) = 1$, for any $m > 0$, $(m, r) = 1$, let u, $1 \leqslant u < r$ satisfy $u(q^m - 1)/r \equiv -1 \pmod{r}$ and $v = \lceil q^m u/r^2 \rceil$. Then $rv \equiv 1 \pmod{(q^m - 1)/r}$. In addition, $v = a + b \sum_{j=0}^{n-1} q^{jr}, a, b < q^{2r}, n = \lfloor m/r \rfloor$.*

Proof. Note that, since $(q^m-1)/r \equiv m(q-1)/r \pmod r$, $(q^m-1)/r$ and r are coprime and u therefore is well-defined. Thus $u(q^m-1)/r = vr-1$ for some integer v and since $q^m u/r^2 = v-(r-u)/r^2$ we have that $v = \lceil q^m u/r^2 \rceil$. From the equation $u(q^m-1)/r = vr-1$ it follows that $rv \equiv 1 \pmod{(q^m-1)/r}$. The rest of the proof is identical to that of Lemma 2. □

Remark 3. To compute u one can replace m by its least residue modulo r since $(q^m-1)/r \equiv m(q-1)/r \pmod r$. Also u/r^2 has a periodic expansion in base q of period r and therefore $v = \lceil q^m u/r^2 \rceil$ also has a periodic expansion in base q with the same period for all m in a fixed residue class modulo r.

THEOREM 2. *Let q be a prime power and let $r > 1$ be such that $r \mid (q-1)$ and $((q-1)/r, r) = 1$. For any $m > 0$, $(m, r) = 1$, given $x \in \mathbb{F}_{q^m}$ one can compute the r-th root of x in $\mathbb{F}_{q^m}$, or show it does not exist, in $O(r(\log m + \log\log q)m^2 \log^2 q)$ steps.*

Proof. By Lemma 3, $r^{-1} \equiv v = a + b\sum_{j=0}^{n-1} q^{jr} \pmod{(q^m-1)/r}$ for some $a, b < q^{2r}$ depending only on $m \pmod r$ and $n = \lfloor m/r \rfloor$. By Lemma 1, raising to the power $\sum_{j=0}^{n-1} q^{jr}$ takes $O(\log n)$ multiplications and raisings to powers of p. The remaining work essentially consists of raising to the powers a and b, each operation taking $O(r \log q)$ multiplications due to the bound on the exponents. The cost of raising to v is therefore $O(\log m + r \log q)$ operations of complexity $O(m^2 \log^2 q)$. But given $x \in \mathbb{F}_{q^m}$ we must still check that $y = x^v$ is a correct root, and to this end we compute y^r with cost $O(\log r m^2 \log^2 q)$. If x is an r-th power, then necessarily $y^r = x$ and we are done, otherwise y^r is not equal to x and we are done too. The total computation cost is therefore $O(r(\log m + \log\log q)m^2 \log^2 q)$ as claimed. □

2.3. An example

As an example consider cube roots in characteristic two. Let $\mathbb{F}_{2^m}$ be a finite field. If m is odd then we are in the situation of Lemma 2 with $q = 2$ and $1/3 \equiv \sum_{j=0}^{(m-1)/2} 4^j \pmod{2^m-1}$.

If m is even and not divisible by 3, then we are in the situation of Lemma 3 with $q = 4$ and we need to distinguish two cases. For $m \equiv 2 \pmod 6$ we have $1/3 \equiv 1 + 56\sum_{j=0}^{(m-8)/6} 64^j \pmod{(2^m-1)/3}$ and for $m \equiv 4 \pmod 6$ we have $1/3 \equiv 2 + 112\sum_{j=0}^{(m-10)/6} 64^j \pmod{(2^m-1)/3}$. For $6 \mid m$ we can apply Theorem 2 with $q = 64$ and $r = 9$ to obtain a method for extracting 9-th roots if m is not divisible by 9. To convert this to a method of extracting cube roots we need to

introduce randomization as in the Adleman-Manders-Miller algorithm. Given x choose a random z and try to extract the 9-th root y of xz^3. If successful output y^3/z as a cube root of x, otherwise try a different z. If a number of these steps fail, declare x not to be a cube.

3. Conclusion

This contribution described an efficient algorithm to compute r-roots in certain finite fields $\mathbb{F}_{q^m}$. Whenever applicable, our technique benefits from the periodic structure of either $r^{-1} \pmod{q^m - 1}$ or $r^{-1} \pmod{(q^m - 1)/r}$, which is handled by a divide-and-conquer technique. The computational complexity of new algorithm improves upon previously known methods like the Adleman-Manders-Miller algorithm.

References

1. Adleman, L. M., K. Manders, and G. Miller: 1977, 'On taking roots in finite fields'. In: *18th IEEE Symposium on Foundations of Computer Science*. pp. 175–177.
2. Bach, E. and J. Shallit: 1996, *Algorithmic Number Theory*, Vol. 1. MIT Press.
3. Barreto, P. S. L. M., H. Y. Kim, B. Lynn, and M. Scott: 2002, 'Efficient Algorithms for Pairing-Based Cryptosystems'. In: *Advances in Cryptology – Crypto'2002*, Vol. 2442 of *Lecture Notes in Computer Science*. pp. 354–368.
4. Boneh, D. and M. Franklin: 2003, 'Identity-based encryption from the Weil pairing'. *SIAM Journal of Computing* **32**(3), 586–615.
5. Boneh, D., B. Lynn, and H. Shacham: 2002, 'Short signatures from the Weil pairing'. In: *Advances in Cryptology – Asiacrypt'2001*, Vol. 2248 of *Lecture Notes in Computer Science*. pp. 514–532.
6. Galbraith, S., S. Paulus, and N. Smart: 2002, 'Arithmetic on superelliptic curves'. *Mathematics of Computation* **71**, 393–405.
7. Gao, S., J. von zur Gathen, D. Panario, and V. Shoup: 2000, 'Algorithms for Exponentiation in Finite Fields'. *Journal of Symbolic Computation* **29**, 879–889.
8. Guajardo, J. and C. Paar: 2002, 'Itoh-Tsujii Inversion in Standard Basis and its Application in Cryptography and Codes'. *Designs, Codes and Cryptography* **25**, 207–216.
9. Itoh, T. and S. Tsujii: 1988, 'A fast algorithm for computing multiplicative inverses in GF(2^m) using normal bases'. *Information and Computation* **78**, 171–177.
10. Libert, B. and J.-J. Quisquater: 2003, 'New identity based signcryption schemes based on pairings'. In: *2003 IEEE Information Theory Workshop*. Paris, France.
11. Menezes, A.: 1993, *Elliptic Curve Public Key Cryptosystems*. Kluwer Academic Publishers.
12. Smart, N. P.: 2002, 'An Identity Based Authenticated Key Agreement Protocol Based on the Weil Pairing'. *Electronics Letters* **38**, 630–632.

13. von zur Gathen, J. and M. Noecker: 2003, 'Computing special powers in finite fields'. *Mathematics of Computation*. Article electronically published on September 26, 2003; see `http://www.ams.org/jourcgi/jour-getitem?pii=S0025-5718-03-01599-0`.