An Efficient Method for Finding Square Roots for Elliptic Curves over OEF

Adnan Abu Mahouz[1], **and Gerhard Hancke**[2]
[1]Department of Electrical, Electronic and Computer Engineering, University of Pretoria and Meraka Institute CSIR[*], Pretoria, South-Africa
[2]Department of Electrical, Electronic and Computer Engineering, University of Pretoria, Pretoria, South-Africa

Abstract – *Elliptic curve cryptosystems, like others public-key encryption schemes, require computing a square roots modulo a prime number. The arithmetic operations in elliptic curve schemes over Optimal Extension Fields (OEF) can be efficiently computed by using an irreducible binomial.*

This paper provides an overview of the OEF, Frobenius map and embedding points on an elliptic curve. The focus is on describing an efficient method to find a square root over Optimal Extension Fields. This method benefits from the Itoh-Tsujii Inversion algorithm which is based on the Frobenius map, in order to simplify the problem of finding the square root over OEF.

Keywords: Square root, elliptic curve, optimal extension field, Frobenius map, quadratic residue.

1 Introduction

Computing square roots in the integers modulo a prime number p (i.e. solving the equation $x^2 = a \bmod p$) is not a new operation. Tonelli [1] published what was probably the first efficient algorithm to compute square roots modulo p in 1891. In subsequent years many more algorithms were published; Bach and Shallit [2] give references to several. Shanks [3] improved on Tonelli's algorithm in 1972. Shanks called his algorithm RESSOL, for "Residue Solver", but today it is often called "Shanks's algorithm". Many other authors [2], [4] have described either Tonellis's or Shanks's square root algorithm.

There are public-key encryption schemes and digital signature schemes that require computing square roots modulo a prime number. In Rabin's public-key encryption scheme, the decryption process requires computing two square roots modulo two different prime numbers. Finding random points on an elliptic curve requires computing a square root, which should be very efficient.

Elliptic curves over $GF(p)$ (p is a prime) or $GF(2^n)$ have been implemented by many companies and standardized by several organizations such as IEEE P1363 and ISO/IEC JTC1/SC27. V. Daniel, C. Bailey, and C. Paar [5] recently proposed an elliptic curve scheme based on Optimal Extension fields. Their method represents the elliptic curve points using a polynomial basis. They showed that arithmetic operations could be efficiently computed by introducing a binomial as a minimal polynomial. In their paper they did not mention any method for computing a square root in a finite field.

This paper gives an efficient method for finding a square root in Optimal Extension Fields $GF(p^m)$ where the prime p satisfies $p \equiv 3 \pmod 4$ and m is odd. A similar technique can be used to compute the square root when $p \equiv 5 \pmod 8$. The remaining case is $p \equiv 1 \pmod 8$, which is the most complex case. In the last case Shanks's algorithm can be applied to find the square root without benefit from this technique.

The paper is organized as follows: Section 2 describes the Frobenius map; Section 3 briefly describes the OEF; Section 4 shows how to embed random point on an elliptic curve. Section 5 shows how to use the Frobenius map to simplify solving the square root equation over OEF, and Section 6 uses the same technique to check if an OEF element is a quadratic residue. Finally Section 7 concludes this paper.

[*] This work was supported in part by the Meraka Institute at the CSIR

2 Frobenius Map

This section defines the Frobenius map. Let $GF(p^m)$ be an extension field, with p prime and m a positive integer. Let $A(x) \in GF(p^m)$. Then the mapping $A \to A^p$ is an automorphism known as the Frobenius map, which is defined as:

$$\sigma(A) = A^p \qquad (1)$$

If i is a nonnegative integer, then i-th iterate of the Frobenius map $A \to A^{p^i}$ is also an automorphism. Let us see how to use the Frobenius map to implement the exponentiations of $GF(p^m)$ elements.

Consider the arbitrary element

$$A(x) = \sum_{j=0}^{m-1} a_j x^j \qquad (2)$$

$\in GF(p^m)$, and $a_j \in GF(p)$. The i-th iterate of the Frobenius map is:

$$\sigma^i(A) = A^{p^i} = \left(\sum_{j=0}^{m-1} a_j x^j\right)^{p^i} \qquad (3)$$
$$= \sum_{j=0}^{m-1} \left(a_j x^j\right)^{p^i} = \sum_{j=0}^{m-1} a_j^{p^i} x^{jp^i}.$$

By Fermate's Little Theorem [6] $a_j^p \equiv a_j \bmod p$, it could be rewrite (3) as:

$$A^{p^i} = \sum_{j=0}^{m-1} a_j x^{jp^i}. \qquad (4)$$

This summation can be simplified using the following theorem:

Theorem [7]

Let $P(x)$ be an irreducible polynomial of the form $P(x) = x^m - w$, over $GF(p^m)$, e an integer, $P(\alpha) = 0$, and it is understood that $p \geq 3$, then:

$$\alpha^e \equiv w^t \alpha^s \qquad (5)$$

where $s \equiv e \bmod m$, and $t = \dfrac{e-s}{m}$.

Let e be the exponent, i.e. $e = jp^i$. If $P(x)$ is given, then it is possible to precompute all x^{jp^i}, for $0 < i, j \leq m-1$ in (4), utilizing a table look-up with entries:

$$c_j = w^{\frac{jp^i - j \bmod m}{m}} \bmod p \qquad (6)$$

where $c_j \in GF(p)$. Now (4) can be rewritten as:

$$A^{p^i} = \sum_{j=0}^{m-1} \left(a_j c_j\right) x^j. \qquad (7)$$

3 Optimal Extension Fields

Optimal Extension Fields were introduced by Bailey and Parr in [5]. The Optimal Extension Field is a class of extension field $GF(p^m)$ for p, a prime of special form and m, a positive integer. OEF fully exploits the optimizations of integer arithmetic in modern processors to produce the fastest multiplication results over binary extension and prime fields.

The OEF satisfies the following:

- p is a prime less than but close to the word size of the processor
- p is a pseudo-Mersenne prime given in the form $p = 2^n \pm c$, where $\log_2 c \leq \frac{1}{2} n$
- An irreducible binomial $P(x) = x^m - w$ exists over $GF(p)$.

We use the standard basis representation to represent a field element $A(x) \in GF(p^m)$:

$$A(x) = \sum_{i=0}^{m-1} a_i x^i \qquad (8)$$
$$= a_{m-1} x^{m-1} + \cdots + a_1 x + a_0$$

where $a_i \in GF(p)$. Since p is less than the size of the word size of the processor, A can be represented using m registers.

4 Embedding a point on a curve

Embedding data on a curve does not mean encrypting that data; it simply encodes data as points on a given elliptic curve E defined over a finite field $GF(p^m)$. Consider the following elliptic curve equation:

$$y^2 = x^3 + ax + b. \tag{9}$$

By converting the right side to a simple form $f(x)$

$$y^2 = f(x). \tag{10}$$

Then will get a simple quadratic equation.

There are several approaches to embed data on a curve. One of the relatively straightforward techniques, is the method proposed by Koblitz [8], which makes use of the fact that the density of points for any curve over a finite field is almost uniformly distributed, which means we can look at any subsection of the bits as an integer and increment it using simple arithmetic. For instance, for arbitrary x data, first we check if it is on the curve, i.e. x satisfies (10); if not, increment x and perform the test again until we get the solution of equation (10). Then two values of y are found, and points $P(x, y)$ and $P(x, -y)$ are on the curve.

5 Solving square root equation

To find a point on an elliptic curve $E(GF(p^m))$, for a given x-coordinate, the corresponding y-coordinate must be calculated by substituting the x-coordinate into the elliptic curve equation (10), which returns the value of y^2, so we need to find the square root of y^2.

Cohen in [4] reports that if p is an odd prime number, and a is a quadratic residue, then there exists an x such that $x^2 \equiv a \pmod{p}$. In a finite field $GF(p^m)$ where the prime p satisfies that $p \equiv 3 \pmod{4}$ and m is odd, the solution is given by

$$x = a^{(p^m+1)/4} \pmod{p} \tag{11}$$

In order to find the value of x, we first notice that, if $m = 2k+1$, for some k [9]

$$\frac{p^m+1}{4} = \frac{p+1}{4}\left[p(p-1)\sum_{i=0}^{k-1}(p^2)^i + 1\right] \tag{12}$$

so that

$$a^{(p^m+1)/4} = \left(\left(a^{\sum_{i=0}^{k-1}(p^2)^i}\right)^{p(p-1)} \cdot a\right)^{(p+1)/4} \tag{13}$$

These relations can be verified by straightforward induction.

The quantity

$$a^{\sum_{i=0}^{k-1} u^i}$$

where $u = p^2$ can be efficiently computed in an analogous fashion to Itoh-Tsujii Inversion [7], based on the Frobenius map in characteristic p as shown in the following equation:

$$a^{1+u+\cdots+u^{k-1}} =$$
$$\begin{cases} \left(a^{1+u+\cdots+u^{\lfloor k/2\rfloor-1}}\right)\cdot\left(a^{1+u+\cdots+u^{\lfloor k/2\rfloor-1}}\right)^{u^{\lfloor k/2\rfloor}}, & k\ evven \\ \left(\left(a^{1+u+\cdots+u^{\lfloor k/2\rfloor-1}}\right)\cdot\left(a^{1+u+\cdots+u^{\lfloor k/2\rfloor-1}}\right)^{u^{\lfloor k/2\rfloor}}\right)^{u} \cdot a, & k\ odd \end{cases} \tag{14}$$

Exponentiation to a power of p is a linear operation in characteristic p. After computing this quantity, a multiplication by a, and exponentiation to $p-1$ and $(p+1)/4$ are required to complete the square root evaluation. The conventional exponentiation algorithm may be used to compute these exponentiations; however, if p is large, it may be an advantage to compute z^{p-1} as $z^p \cdot z^{-1}$.

Consider

$$A(x) \in GF(p^m)$$

where $m = 2 \times 8 + 1$, so $k = 8$, assume that A is a quadratic residue.

Using equation (14) to compute $A^{\sum_{i=0}^{k-1} u^i}$ as

$$A^{1+u+\cdots+u^7} = \left(A^{1+u+u^2+u^3}\right) \cdot \left(A^{1+u+u^2+u^3}\right)^{u^4}, \quad k_0 = 8$$

$$A^{1+u+u^2+u^3} = \left(A^{1+u}\right) \cdot \left(A^{1+u}\right)^{u^2}, \quad k_1 = 4$$

$$A^{1+u} = A \,.\, (A)^u, \quad k_2 = 2$$

The following procedure shows how to compute $A^{\sum_{i=0}^{k-1} u^i}$ using the Frobenius map.

```
Computing $A^{\sum_{i=0}^{k-1} u^i}$ in $GF(p^{17})$
Require:  $A(x) \in GF(p^m)$, where
          $m = 2k + 1,\ k = 8$
Ensure:   $C(x) \equiv A^{\sum_{i=0}^{k-1} u^i}$, where
          $C(x) \in GF(p^m),\ u = p^2$
1:  $C_0 \leftarrow A^{p^2} = A^u$                                  $\sigma^2(A)$
2:  $C_1 \leftarrow C_0\ A = A^{1+u}$                               Multi.
3:  $C_2 \leftarrow (C_1)^{p^4} = (A^{1+u})^{u^2}$                   $\sigma^4(C_1)$
4:  $C_3 \leftarrow C_2\ C_1 = A^{1+u+u^2+u^3}$                      Multi.
5:  $C_4 \leftarrow (C_3)^{p^8} = (A^{1+u+u^2+u^3})^{u^4}$           $\sigma^8(C_3)$
6:  $C \leftarrow C_4 C_3 = A^{1+u+u^2+\cdots+u^7} = A^{\sum_{i=0}^{k-1} u^i}$   Multi.
```

Then using equation (13) to complete the evaluation of the square root of A. the following procedure shows the details of computing the square root of an arbitrary element $A(x) \in GF(p^m)$, using equation (13).

```
OEF square root
Require:  $A(x) \in GF(p^m)$, where
          $m = 2k + 1$, $k$ is integer
Ensure:   Find $C(x)$, such that
          $C^2(x) \equiv A(x) \pmod{p}$,
          where $C(x) \in GF(p^m)$
1:  $C_0 \leftarrow A^{\sum_{i=0}^{k-1} u^i}$        Eq. (14)
2:  $C_1 \leftarrow C_0^p$                          $\sigma(C_0)$
3:  $C_2 \leftarrow C_1^p$                          $\sigma(C_1)$
4:  $C_3 \leftarrow (C_1)^{-1}$                     Inv.
5:  $C_4 \leftarrow C_2\ C_3$                       Multi.
6:  $C_5 \leftarrow C_4\ A$                         Multi.
7:  $C \leftarrow (C_5)^{(p+1)/4}$                  Exp.
```

Similar technique can be used to compute the square root in a finite field $GF(p^m)$ when $p \equiv 5 \pmod 8$ and odd m.

6 Quadratic residue

Let p be an odd prime, then the congruence

$$x^2 \equiv a \pmod{p} \qquad (15)$$

for a given a, has three cases; there is no solution, in this case, a is a quadratic non-residue modulo p. There is one solution if $a \equiv 0 \pmod{p}$. In the last case there are two solutions, then a is a quadratic residue modulo p. A simple way of identifying whether or not an integer is a quadratic residue modulo p is the Legendre symbol.

Legendre symbol [8]

Let a be an integer and $p > 2$ a prime. We define the Legendre symbol (a/p) as following:

$$\left(\frac{a}{p}\right) = \begin{cases} 0, & \text{if } p \mid a \\ 1, & \text{if } a \text{ is a quadratic residue modulo } p \\ -1, & \text{if } a \text{ is a quadratic nonresidue modulo } p \end{cases}$$

Then the number of solutions modulo p of the above congruence is $1+(a/p)$.

By Fermat's Little theorem, in $GF(p)$ the square of $a^{(p-1)/2}$ is 1, so $a^{(p-1)/2}$ itself is ± 1, so we have the following congruence:

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p} \qquad (16)$$

In $GF(p^m)$, if $m = 2k+1$, for some k

$$\frac{p^m-1}{2} = \frac{p-1}{2}\left[p(p+1)\sum_{i=0}^{k-1}(p^2)^i + 1\right] \qquad (17)$$

So that

$$a^{(p^m-1)/2} = \left(\left(a^{\sum_{i=0}^{k-1}(p^2)^i}\right)^{p(p+1)} \cdot a\right)^{(p-1)/2} \qquad (18)$$

Again the quantity $a^{\sum_{i=0}^{k-1} u^i}$ can be computed using equation (14) as shown in the previous section. The following procedure computes the value of the Legendre symbol

```
Legendre symbol
Require:  $A(x) \in GF(p^m)$, where $m = 2k+1$,
          and $k$ is an integer
Ensure:   Find $c = (a/p) \equiv A^{(p-1)/2} \pmod{p}$
1:  $C_0 \leftarrow A^{\sum_{i=0}^{k-1} u^i}$      Eq. (14)
2:  $C_1 \leftarrow C_0^p$                          $\sigma(C_0)$
3:  $C_2 \leftarrow C_1^p$                          $\sigma(C_1)$
4:  $C_3 \leftarrow C_1 C_2$                        Multi.
6:  $C_4 \leftarrow C_3 A$                          Multi.
7:  $c \leftarrow (C_4)^{(p-1)/2}$                  Exp.
```

7 Conclusion

Using elliptic curve cryptosystems requires picking random values that satisfy the equation of the curve. On the other hand, plaintext should be embedded as a point on the curve before starting of the encryption process. These two operations require solving a square root equation, and so it is important to find an efficient method to find a square root over a finite field.

In this paper we showed that an analogous fashion to Itoh-Tsujii Inversion algorithm, which is based on the Frobenius map in characteristic p, can be used to simplify the problem of finding the square root over OEF.

8 References

[1] Tonelli, Bemerkung uber die Auflosung Quadratischer Congruenzen, Gottinger Nachrichten, pp. 344—346, 1891.

[2] E. Bach, and J. Shallit, "Algorithmic Number Theory, Efficient Algorithms", Foundations of Computing Series, MIT Press, Cambridge, MA, 1996.

[3] D. Shanks, "Five Number-Theoretic Algorithms", Proceedings of SecondManitoba Conference on Numerical Mathematics, pp. 51—70, 1972.

[4] H. Cohen, "A Course in Computational Algebraic Number Theory", Springer-Verlag, New York, USA, 1995.

[5] V. Daniel, C. Bailey, and C. Paar, "Optimal Extension Fields for Fast Arithmetic in Public-Key Algorithms", Advances in Cryptology – CRYPTO '98, Lecture Notes in Computer Science (LNCS), vol. 1462, Springer-Verlag, pp. 472—485, Aug 1998.

[6] K.H. Rosen, "Elementary Number Theory and its Applications", AT&T Bell Laboratories, New Jersey, USA, 1993.

[7] J. Guajardo, and C. Paar, "Itoh-Tsujii Inversion in Standard Basis and Its Application in Cryptography and Codes, Designs", Codes and Cryptography, vol. 25, no. 2, pp. 107—216, Feb 2002.

[8] N. Koblitz, "A Course in Number Theory and Cryptography", Springer-Verlag, New York, USA, 1994.

[9] P.S Barreto, H.Y. Kim, and M. Scoot, "Efficient Algorithms for Pairing-Based Crytosystems", http://eprint.iacr.org/2002/008.pdf.