Itoh-Tsujii Inversion in Standard Basis and Its Application in Cryptography and Codes

JORGE GUAJARDO guajardo@ece.wpi.edu
ECE Department, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609 USA

CHRISTOF PAAR christof@ece.wpi.edu
ECE Department, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609 USA

Abstract. This contribution is concerned with a generalization of Itoh and Tsujii's algorithm for inversion in extension fields $GF(q^m)$. Unlike the original algorithm, the method introduced here uses a standard (or polynomial) basis representation. The inversion method is generalized for standard basis representation and relevant complexity expressions are established, consisting of the number of extension field multiplications and exponentiations. As the main contribution, for three important classes of fields we show that the Frobenius map can be explored to perform the exponentiations required for the inversion algorithm efficiently. As an important consequence, Itoh and Tsujii's inversion method shows almost the same practical complexity for standard basis as for normal basis representation for the field classes considered.

Keywords: Cryptography, codes, Galois fields, inversion, polynomial basis, AOP, ESP

In **Design, Codes and Cryptography**, *25(2): 207-216, Feb 2002.*

1. Introduction

Galois field arithmetic has received considerable attention in recent years due to their application in public-key cryptography schemes and error correcting codes. In particular, two public-key cryptosystems based on finite fields stand out: elliptic curve cryptosystems, introduced by Miller and Koblitz [17, 11], and hyperelliptic cryptosystems, a generalization of elliptic curves introduced by Koblitz in [12]. Both prime fields and extension fields have been proposed for use in such cryptographic systems. In order to performance-optimize elliptic curve schemes, it is desirable to improve the underlying arithmetic operations. Besides multiplication, the other field operation which is performance critical is inversion.

The two most popular methods for large finite field inversion are either based on the Euclidean algorithm or one of its derivatives (e.g., the almost inverse algorithm [19]), or on Fermat's little theorem. For extension fields $GF(q^m)$, the Itoh and Tsujii inversion (ITI) algorithm [8] offers an alternative. Although it does not perform a complete inversion, it reduces extension field inversion to inversion in $GF(q)$. It is assumed that subfield inversion can be done relatively easily, e.g., through table look-up or with one of the general methods mentioned earlier. The ITI algorithm is applicable to finite fields $GF(2^m)$ given in a normal basis representation. In particular, the original reference deals with composite fields $GF((2^n)^m)$. Our contribution applies the idea of Itoh and Tsujii to fields $GF(q^m)$ given in standard basis (or polynomial or canonical) basis representation. Although the exponentiations required in the algorithm make it rather inefficient for general fields in a standard basis representation, it can be shown that for certain classes of finite fields, the exponentiations can be computed with a very low complexity. The field classes for which efficient inversion algorithms are possible include composite fields $GF(q^m)$, $q = 2^n$, with a binary extension field polynomial; fields $GF(q^m)$ where $q = p^n$, p is an odd prime, and the field polynomial is a binomial; and fields $GF(q^m)$ where q is a prime power and the field polynomial is an equally spaced polynomial with binary coefficients.

There has been a great deal of research done on the arithmetic advantages of irreducible polynomials of special form over $GF(2)$. In [9], All One Polynomials (AOPs) and Evenly Spaced Polynomials (ESPs) over $GF(2)$ are introduced. The authors show necessary and sufficient conditions for ESPs to be irreducible over $GF(2)$ and propose a new configuration of parallel multipliers for fields $GF(2^m)$, based on irreducible AOPs and ESPs over $GF(2)$. In [7], necessary and sufficient conditions for a family of infinitely many ESPs to be irreducible over $GF(2)$ are introduced. In addition, a uniqueness criteria which characterizes all irreducible ESPs over $GF(2)$ in a strict sense is presented. Both [6] and [22] use irreducible AOPs and ESPs over $GF(2)$ to implement efficient parallel multipliers. The complexity of the proposed multipliers in both of these contributions is less than that of the parallel multiplier proposed in [9]. Furthermore, since inversion can be achieved by repeated multiplications, [6] applies parallel multipliers to efficiently compute the inverse of elements in $GF(2^m)$ using a variant of Fermat's Little Theorem. Finally, [4] describes a version of the ITI algorithm applied to fields $GF((2^n)^m)$ in a polynomial basis representation. The authors show a major computational advantage when choosing the extension field polynomial with only binary coefficients. However, the upper bound on the number of operations needed to compute an inverse

in $GF((2^n)^m)$ is left unchanged from the one showed in the original Itoh and Tsujii's paper.

The reminder of the contribution is structured as follows. Section 2 revisits the Itoh and Tsujii (ITI) algorithm for inversion and generalizes it for fields of any characteristic. Unlike the original algorithm, we generalize the ITI inversion algorithm to standard (or polynomial) basis representation in Section 3. In addition, the number of extension field multiplications and exponentiations required to perform an inverse operation is established. Finally, in Section 4 we show that for three important classes of fields, the Frobenius map can be explored to perform the exponentiations required for the inversion algorithm efficiently.

2. Preliminaries: The ITI Algorithm over $GF(q^m)$

Itoh and Tsujii proposed in [8] two algorithms. In the first part of the paper, they present a method for computing multiplicative inverses in $GF(2^n)$ through a clever construction of addition chains. In the second part of [8], the authors propose a method for reducing inversion in $GF(q^m)$ to inversion in $GF(q)$, where $q = 2^n$. Both algorithms were studied in the context of a normal basis representation of the fields $GF(2^n)$ and $GF((2^n)^m)$. The basic property of the second algorithm is that inversion in $GF((2^n)^m)$, is reduced to inversion in the subfield $GF(2^n)$. It is assumed that subfield inversion can be done efficiently. In the following we will review the second algorithm with a new notation. Our presentation will be slightly more general as we do not require a subfield of the form $GF(2^n)$ but allow general subfields $GF(q)$.

THEOREM 1. *[8] Let $A \in GF(q^m)^*$ and $r = (q^m - 1)/(q - 1)$. Then, the multiplicative inverse of an element A can be computed as*

$$A^{-1} = (A^r)^{-1} A^{r-1}. \tag{1}$$

Computing the inverse through Theorem 1 requires four steps:

Step 1 Exponentiation in $GF(q^m)$, yielding A^{r-1}.

Step 2 Multiplication of A and A^{r-1}, yielding $A^r \in GF(q)$.

Step 3 Inversion in $GF(q)$, yielding $(A^r)^{-1}$.

Step 4 Multiplication of $(A^r)^{-1} A^{r-1}$.

Steps 2 and 4 are trivial since both A^r, in Step 2, and $(A^r)^{-1}$, in Step 4, are in the subfield. Both operations can, in most cases, be

done with a complexity that is well below that of one single extension field multiplication. The complexity of Step 3, subfield inversion, depends heavily on the type and order of the subfield $GF(q)$ and will not be discussed here. However, in many practical scenarios, e.g., in cryptographic applications, the subfield can be small enough to perform inversion very efficiently, in other words, through table look-up [4, 21], or by using the Euclidean algorithm which can be applied with relatively low processing times for small subfield orders [1]. What remains is Step 1, exponentiation to the $(r-1)$th power in the extension field $GF(q^m)$. The rest of this contribution will deal with Step 1.

First, we notice that the exponent can be expressed in q-adic representation as

$$r - 1 = q^{m-1} + \cdots + q^2 + q = (1 \cdots 110)_q$$

This exponentiation can be computed through repeated raising of intermediate results to the q-th power and multiplications. The number of multiplications in $GF(q^m)$ can be minimized by using the addition chain proposed by Itoh and Tsujii [8]. Thus:

$$\#\text{MUL} = \lfloor \log_2(m-1) \rfloor + HW(m-1) - 1 \tag{2}$$

where $HW(.)$ denotes the Hamming weight of its operand. The number of exponentiations to the q-th power is given by

$$\#q\text{-EXP} = m - 1$$

The original paper assumes a normal basis representation of the field elements of $GF(q^m)$, $q = 2^n$, in which the exponentiations to the q-th power are simply cyclic shifts of the m coefficients that represent an individual field element. In standard basis, however, these exponentiations are in general considerably more expensive. Standard basis q-th power exponentiation will be considered in detail in Section 3 below.

The algorithm performs alternatingly exponentiations and multiplications with previous results. For the treatment in Section 3 it is important to observe that in general several exponentiations to the qth power are performed between two multiplications.

3. Itoh-Tsujii Inversion in Standard Basis

In the following, we will consider the field $GF(q^m)$ generated by an irreducible polynomial $P(x) = x^m + \sum_{i=0}^{m-1} p_i x^i$ over $GF(q)$ of degree m. Let α be a root of $P(x)$, then we represent $A \in GF(q^m)^\star$ as $A(\alpha) = \sum_{i=0}^{m-1} a_i \alpha^i$, $a_i \in GF(q)$.

We will now establish the complexity of raising A to the q^e-th power, where e is a positive integer. This is the eth iterate of the Frobenius automorphism:

$$A^{q^e}(\alpha) = \left(\sum_{i=0}^{m-1} a_i \alpha^i \right)^{q^e} = \sum_{i=0}^{m-1} a_i \alpha^{i\, q^e}$$

This exponentiation is a $GF(q)$-linear mapping for all integers $e > 0$. We can explicitly construct the matrix which describes the mapping by considering the standard basis representations of $\alpha^{i\, q^e}$, $i = 1, 2, \ldots, m-1$:

$$\alpha^{i\, q^e} \equiv s_{0,i}^{(e)} + s_{1,i}^{(e)} \alpha + \cdots + s_{m-1,i}^{(e)} \alpha^{m-1}, \;\; i = 1, 2, \ldots, m-1 \qquad (3)$$

together with the identity $P(\alpha) = 0$. Notice that the superscripts "(e)" are mere indices. The matrix follows now as

$$A^{q^e}(\alpha) = \begin{pmatrix} 1 & s_{0,1}^{(e)} & s_{0,2}^{(e)} & \cdots & s_{0,m-1}^{(e)} \\ 0 & s_{1,1}^{(e)} & s_{1,2}^{(e)} & \cdots & s_{1,m-1}^{(e)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & s_{m-1,1}^{(e)} & s_{m-1,2}^{(e)} & \cdots & s_{m-1,m-1}^{(e)} \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_{m-1} \end{pmatrix} \qquad (4)$$

In general, the eth iterate of the Frobenius map has a complexity of $m(m-1)$ multiplications and $m(m-2)+1 = (m-1)^2$ additions in $GF(q)$. We note that this complexity is roughly the same as one $GF(q^m)$ multiplication, which requires m^2 subfield multiplications if we do not assume fast convolution techniques (e.g., Karatsuba's algorithm or methods based on the Chinese Remainder theorem.)

An adaptation of the ITI algorithm to standard basis is straightforward. In fact, the description above is independent of the basis representation. We observe, however, that in standard basis the eth iterate of the Frobenius map, where $e > 1$, is as costly as a single exponentiation to the qth power. Thus, we change the algorithm slightly by performing as many subsequent exponentiations to the qth power in one step between multiplications. This yields the same multiplication complexity as given in (2), but we perform now eth iterates of the Frobenius map with the following complexity:

THEOREM 2. *Let $A \in GF(q^m)$. One can compute A^{r-1} where $r-1 = q + q^2 + \cdots + q^{(m-1)}$ with no more than*

$$\begin{aligned} \#MUL &= \lfloor \log_2(m-1) \rfloor + HW(m-1) - 1 \\ \#q^e\text{-}EXP &= \lfloor \log_2(m-1) \rfloor + HW(m-1) \end{aligned}$$

operations, where $HW(.)$ denotes the Hamming weight of its operand and $\#MUL$ and $\#q^e$-EXP refer to multiplications and exponentiations to the q^eth power in $GF(q^m)$, respectively.

Proof. First, consider the computation of A^{s_k} where $s_k = \sum_{i=1}^{2^k} q^i = q + q^2 + \cdots + q^{2^k}$. Notice that $A^{s_k} = (A^{s_{k-1}})^{q^{2^{k-1}}} A^{s_{k-1}}$. If we denote by $M(k)$ the number of multiplications and by $E(k)$ the number of exponentiations to the q^eth power required to compute A^{s_k}, then it is easy to see that $M(k) = M(k-1) + 1$ and $E(k) = E(k-1) + 1$. Notice also that $A^{s_0} = A^q$, thus $M(k=0) = 0$ and $E(k=0) = 1$. It follows that $M(k) = k$ and $E(k) = k + 1$. Furthermore, in computing A^{s_k}, we have also computed A^{s_i} for $s_i < s_k$. We now apply a similar procedure as in the proof of Theorem 2 in [8]. Let $m - 1 = \sum_{u=1}^{t} 2^{k_u}$ with $k_1 > k_2 > \cdots > k_t$. Then, one can re-write A^{r-1} as follows:

$$A^{r-1} = A^{q^{m-1}+\cdots+q^2+q} = (A^{s_{k_t}}) \left(\cdots (A^{s_{k_3}}) \left[(A^{s_{k_2}})(A^{s_{k_1}})^{q^{2^{k_2}}} \right]^{q^{2^{k_3}}} \cdots \right)^{q^{2^{k_t}}}$$

Since $k_1 > k_i$ for $i = 2, \cdots, t$ then if we compute $A^{s_{k_1}}$ as above, all the $A^{s_{k_i}}$ for $i = 2, \cdots, t$ will also be computed. From our previous results we see that $M(k_1) = k_1 = \lfloor \log_2(m-1) \rfloor$ and $E(k_1) = k_1 + 1 = \lfloor \log_2(m-1) \rfloor + 1$. Also notice that we go through $t - 1 = HW(m-1) - 1$ multiplications and eth iterates of the Frobenius after computing $A^{s_{k_1}}$. Adding up the partial complexities, one obtains the result in Theorem 2. □

We would like to stress that Theorem 2 is just an upper bound on the complexity of this exponentiation. Thus, it is possible to find addition chains which yield better complexity as shown in [3].

In addition, we see from Theorem 2 that Step 1 of the ITI algorithm requires about as many exponentiations to the q^eth power as multiplications in $GF(q^m)$ if a standard basis representation is being used. In the discussion earlier in this section it was established that eth iterates of the Frobenius map are roughly as costly as multiplications. Hence, if it is possible to make exponentiations to the q^eth power more efficient, considerable speed-ups of the algorithm can be expected. In the remainder of the paper we will introduce three classes of finite fields for which the complexity of the eth iterates of the Frobenius map is in fact substantially lower than that of a general multiplication in $GF(q^m)$.

4. Field Types with Low Complexity Inversion

This section introduces three types of finite fields for which eth iterates of the Frobenius map are substantially less costly than general field multiplications. All three field families have been proposed for use in public-key cryptosystems, mainly in the context of elliptic curve cryptosystems.

4.1. Fields $GF((2^n)^m)$ with Binary Field Polynomials

Fields of characteristic two with two field extensions $GF(q^m)$, $q = 2^n$, sometimes referred to as *composite fields*, have been proposed repeatedly for applications in elliptic curve cryptosystems [5, 21, 4]. They are also attractive for smaller fields such as those needed for error correcting codes [18].

Let the field polynomial $P(x)$ be irreducible over $GF(2)$ and of degree m. Then, it is a well known fact that $P(x)$ will also be irreducible over $GF(2^n)$ if and only if $\gcd(n, m) = 1$ [15, Corollary 3.47]. Notice that since $P(x)$ is binary, all the powers α^{iq^e} in (3) can also be represented as binary polynomials. Hence, the matrix coefficients $s_{i,j}$ are elements of $GF(2)$ and no general multiplications are required in the matrix multiplication shown in (4). Assuming on average an equal number of ones and zeros in the matrix, an eth iterate of the Frobenius map can be computed with an average complexity of

$$\left(\frac{m-1}{2} - 1\right) m + 1 = \frac{(m-1)(m-2)}{2} \leq \frac{m^2}{2}$$

additions in $GF(2^n)$ and no $GF(2^n)$ multiplications. Since $GF((2^n)^m)$ multiplications require (in a straight forward realization) m^2 subfield multiplications and $(m-1)^2$ subfield additions, the dominant complexity for computing A^{r-1} in $GF((2^n)^m)$ is now determined by the number of extension field multiplications as given in (2).

Example 1. As an example we consider the special case where $n = 16$ and $m = 11$ which is of interest for cryptographic systems that are based on the discrete logarithm problem for elliptic curves [21, 4]. We chose as field polynomial the trinomial $P(x) = x^{11} + x^2 + 1$. We can now apply Theorem 2 to compute $A^{r-1} = A^{2^{16}+2^{2\cdot 16}+\cdots+2^{10\cdot 16}}$. Note that $m - 1 = 10 = 2^3 + 2 = 2^{k_1} + 2^{k_2}$ and that $q = 2^n$. Then

$$A^{r-1} = (A^{s_{k_2}})\,(A^{s_{k_1}})^{q^{2^{k_2}}} = (A^{2^n+2^{2n}})\left(A^{2^n+2^{2n}+\cdots+2^{8n}}\right)^{2^{2n}} = A^{2^n+2^{2n}+\cdots+2^{10n}}$$

$A^{s_{k_1}}$ can be computed using the following addition chain:

$$\begin{aligned}
A & \\
A^{2^n} & \\
(A^{2^n})^{2^n} &= A^{2^{2n}} \\
A^{2^n} A^{2^{2n}} &= A^{2^n+2^{2n}} \\
(A^{2^n+2^{2n}})^{2^{2n}} &= A^{2^{3n}+2^{4n}} \\
A^{2^n+2^{2n}} A^{2^{3n}+2^{4n}} &= A^{2^n+2^{2n}+2^{3n}+2^{4n}} \\
(A^{2^n+2^{2n}+2^{3n}+2^{4n}})^{2^{4n}} &= A^{2^{5n}+2^{6n}+2^{7n}+2^{8n}} \\
A^{2^n+2^{2n}+2^{3n}+2^{4n}} A^{2^{5n}+2^{6n}+2^{7n}+2^{8n}} &= A^{s_{k_1}}
\end{aligned}$$

Notice that in computing $A^{s_{k_1}}$, we also computed $A^{s_{k_2}} = A^{2^n+2^{2n}}$ and that in the overall process we performed $\lfloor \log_2(10) \rfloor + HW(10) - 1 = 3 + 2 - 1 = 4$ multiplications and 5 exponentiations to a power 2^{ne} as predicted by Theorem 2. Furthermore, each exponentiation to a power 2^{ne} will only require $(m-1)(m-2)/2 = 10 \times 9/2 = 45$ additions in $GF(2^n)$ on average.

4.2. Fields $GF(q^m)$ with Binomials as Field Polynomials

For extension fields with odd prime characteristic it is often possible to choose irreducible binomials $P(x) = x^m - \omega$, $\omega \in GF(q)$. A specific sub-class of these fields where q is a prime of the form $q = p = 2^n - c$, c "small", has recently been proposed for cryptographic applications in [2, 14, 10] (Also see [1] for tabulated tables with values for n, c, m, and ω). We will show that for the general case of fields $GF(q^m)$ with binomials as field polynomials, the eth iterates of the Frobenius map in the ITI algorithm are computationally inexpensive. [15, Theorem 3.75] describes the conditions necessary for irreducible binomials to exist. The computational savings are due to the following theorem:

THEOREM 3. *Let $P(x)$ be an irreducible polynomial of the form $P(x) = x^m - \omega$ over $GF(q)$, e an integer, $P(\alpha) = 0$, and it is understood that $q = p^n$, $p \geq 3$. Then:*

$$\alpha^e \equiv \omega^t \alpha^s$$

where $s \equiv e \bmod m$ and $t = \frac{e-s}{m}$

Proof. First, notice that since $P(\alpha) = 0$, then $\alpha^m \equiv \omega$. Now $\alpha^e = \alpha^{tm+s}$, where t and s are as defined above. Then, $\alpha^e = \alpha^{tm}\alpha^s \equiv \omega^t \alpha^s$ □

It follows immediately from Theorem 3 that the exponentiation matrix in (4) has only one non-zero entry per row. Moreover, the theorem also

provides an efficient method for computing these entries. Again, the dominant complexity of Step 1 of the ITI algorithms is determined by the number of extension field multiplications as given in (2).

Example 2. In [1], we find that $p = 2^7 - 1$ is prime and that $P(x) = x^{21} - 3$ is irreducible over $GF(p)$. Then, $P(x)$ is also irreducible over $GF(p^k)$ for $\gcd(k, m) = 1$. Let $k = 2$ and $q = p^2$, then using the construction shown in the proof of Theorem 2, it is easy to see that, for any $A \in GF(q^{21})$, computing $A^{r-1} = A^{q+q^2+\cdots+q^{20}}$ requires 5 multiplications in $GF(q^{21})$ and 6 exponentiations in $GF(q^{21})$: two to the qth power, one to the q^2th power, two to the q^4th power, and one to the q^8th power. This is in complete agreement with Theorem 2. Finally notice that for α a root of $P(x)$ and $\omega = 3 \in GF(p)$, we have the following identities: $\alpha^{iq} \equiv (73\alpha)^i$, $\alpha^{iq^2} \equiv (122\alpha)^i$, $\alpha^{iq^3} \equiv (25\alpha)^i$, and $\alpha^{iq} \equiv (117\alpha)^i$, for $i = 1, 2, \cdots m - 1 = 20$. This implies that in computing an exponentiation to the q^eth power, one will perform at most $m - 1$ multiplications by an element of $GF(q)$ as mentioned above.

4.3. Fields $GF(q^m)$ with Binary s-ESP Field Polynomials

Irreducible All One Polynomials and Equally Spaced Polynomials have been proposed in [9, 7, 6, 22] to optimized the arithmetic in fields of characteristic 2. Nevertheless, these types of polynomials have not been treated in the literature for the case of odd characteristic extension fields. This section considers fields with binary irreducible s-ESPs as their field polynomial. In the following, we present some definitions and theorems necessary for the subsequent discussion.

DEFINITION 1. *[20] A polynomial $f(x) = x^m + x^{m-1} + \cdots + x + 1$ over $GF(q)$ is called an All One Polynomial (AOP) of degree m.*

DEFINITION 2. *[9] A polynomial $g(x) = x^{sm} + x^{s(m-1)} + \cdots + x^s + 1 = f(x^s)$ over $GF(q)$, where $f(x)$ is an AOP of degree m over $GF(q)$ is called a binary s-Equally Spaced Polynomial (s-ESP) of degree sm.*

We have abused the original definitions which were for the case $q = 2$ and generalized them to $q = p^n$, p an odd prime. Notice that AOPs are just a special case of binary s-ESPs in which $s = 1$. Appendix A shows a method for constructing irreducible binary s-ESPs over fields of any characteristic.

In the following we show the computational advantages derived from choosing a s-ESP as our field polynomial. We consider fields $GF(q^{sm})$

with a binary irreducible s-ESP as their field polynomial. Again, we will show that raising elements of these fields to the q^e-th power is computationally inexpensive. We look again at the representatives of the residue classes which contain $\alpha^{i\,q^e}$ in (3).

THEOREM 4. *Let $P(x) = x^{sm} + x^{s(m-1)} + \cdots + x^s + 1$ be a binary irreducible s-ESP over $GF(q)$ and $P(\alpha) = 0$. Then an element $\alpha^l \in GF(q^{sm})^*$, $l > 0$, has the polynomial representation:*

$$\alpha^l \equiv \begin{cases} \alpha^r, & \text{if } 0 \le r < sm \\ \sum_{i=0}^{m-1} -\alpha^{is+(r-m)}, & \text{if } sm \le r < sm+s \end{cases}$$

where $l \equiv r \bmod (sm+s)$ and $ord(\alpha) = sm + s$.

Proof. Let $P(x)$ and α be as defined in the theorem. Then, all α^l with $0 \le l < sm$ are distinct monomials, elements of $GF(q^{sm})^*$. For $l \ge sm$, we have the following equivalences:

$$\begin{aligned} \alpha^{sm} &\equiv -1 - \alpha^s - \cdots - \alpha^{sm-s} \\ \alpha^{sm+1} &\equiv -\alpha - \alpha^{s+1} - \cdots - \alpha^{sm-s+1} \\ &\vdots \\ \alpha^{sm+s-1} &\equiv -\alpha^{s-1} - \alpha^{2s-1} - \cdots - \alpha^{sm-1} \\ \alpha^{sm+s} &\equiv 1 \end{aligned}$$

It follows from the congruences above that $\mathrm{ord}(\alpha) = s(m+1)$ and therefore $\alpha^l \equiv \alpha^{l \bmod (sm+s)}$. The upper part of the congruence, where $0 \le (l \bmod (sm+s)) < sm$, is now clear. For the other case, where $sm \le (l \bmod (sm+s)) < sm+s$, α^l is a polynomial with equally space coefficients of the form shown above. Finally, notice that since α is a root of $P(x)$, the above holds also true for fields of the form $GF((q^k)^{sm})$ where $\gcd(k, sm) = 1$. □

It follows from Theorem 4 that the matrix in (4), which describes the eth iterates of the Frobenius map contains entries equal to -1, 0, or 1. Hence, multiplication by the matrix does not require any subfield multiplications but only additions and subtractions.

Example 3. As an example, we consider the 5-ESP $P(x) = x^{20}+x^{15}+x^{10}+x^5+1$ which is irreducible over $GF(7)$ (fields of characteristic 7 have been suggested for cryptographic applications in [13]) and over $GF(7^3)$ since $\gcd(3, 20) = 1$ (Notice that $f(x^s) = P(x)$ where $f(x) = x^4+x^3+x^2+x+1$ is an irreducible AOP and $s = 5$). Then, following the construction method outlined in the proof of Theorem 2 it is easy to see that, for any $A \in GF(q^{20})$, $q = 7^3$, computing $A^{r-1} = A^{q+q^2+\cdots+q^{19}}$ requires 6 multiplications in $GF(q^{20})$ and 7 exponentiations in $GF(q^{20})$:

three to the qth power, two to the q^2th power, one to the q^4th power, and one to the q^8th power. Once again, this confirms Theorem 2. Now for α satisfying $P(\alpha) = 0$ with $\mathrm{ord}(\alpha) = s(m+1) = 25$, we obtain the following relations for the case $i = 1$ in (3): $\alpha^q \equiv \alpha^{18}$, $\alpha^{q^2} \equiv -\alpha^{19} - \alpha^{14} - \alpha^9 - \alpha^4$, $\alpha^{q^4} \equiv \alpha$, and $\alpha^{q^8} \equiv \alpha^2$.

5. Conclusions

This contribution adapted the algorithm by Itoh and Tsujii for inversion in extension fields to a standard basis representation. We showed that by considering eth iterates of the Frobenius map rather than single exponentiations to the qth power, the number of extension field exponentiations is logarithmic (rather than linear) in the field extension degree m. Secondly, we showed that the required exponentiations can be performed very efficiently for three families of finite fields. As an important consequence, the cost for inversion for these fields is essentially determined by the number of field multiplications only, which is given by Itoh and Tsujii as $\lfloor \log_2(m-1) \rfloor + HW(m-1) - 1$. In particular for small extension degrees such as $m = 3$, which are attractive for some cryptographic applications [2], the ITI algorithm offers an efficient alternative to variants of the Euclidean algorithm.

Appendix

A. Construction of Binary Irreducible s-ESPs

The following theorem describes the cases when irreducible AOPs exist.

THEOREM 5. *[16, Chapter 5] The polynomial* $f(x) = x^m + x^{m-1} + \cdots + x + 1$ *is irreducible over* $GF(q)$ *if and only if* $m+1$ *is prime and* q *is primitive in* $GF(m+1)$

The following theorem gives us a way of constructing binary irreducible s-ESPs from irreducible AOPs.

THEOREM 6. *[15, Theorem 3.35] Let* $f_1(x), f_2(x), \cdots, f_N(x)$ *be all the distinct monic irreducible polynomials in* $GF(q)[x]$ *of degree* m *and order* e*, and let* $s \geq 2$ *be an integer whose prime factors divide* e *but not* $q^m - 1/e$*. Assume also that* $q^m \equiv 1 \bmod 4$ *if* $s \equiv 0 \bmod 4$*. Then* $f_1(x^s), f_2(x^s), \cdots, f_N(x^s)$ *are all the distinct monic irreducible polynomials in* $GF(q)[x]$ *of degree* ms *and order* es*.*

It is important to point out that by choosing $GF(q)$ and m such that any of the $f_i(x)$ in Theorem 6 is a binary AOP, we immediately obtain a binary irreducible s-ESP, where s satisfies the conditions in Theorem 6. Notice, also, that if we construct an irreducible s-ESP of degree sm over $GF(q)$ using Theorem 6, call it $P(x)$, then $P(x)$ is also irreducible over $GF(q^k)$, for k satisfying $\gcd(k, sm) = 1$ [15].

References

1. Bailey, D. V. and C. Paar, 'Efficient Arithmetic in Finite Field Extensions with Application in Elliptic Curve Cryptography'. To appear in the Journal of Cryptology.
2. Bailey, D. V. and C. Paar: 1998, 'Optimal Extension Fields for Fast Arithmetic in Public-Key Algorithms'. In: H. Krawczyk (ed.): *Advances in Cryptology — CRYPTO '98*. Berlin, pp. 472–485. Lecture Notes in Computer Science Volume 1462.
3. Chung, J. W., S. G. Sim, and P. J. Lee: 2000, 'Fast Implementation of Elliptic Curve Defined over $GF(p^m)$ on CalmRISC with MAC2424 Coprocessor'. In: Çetin K. Koç and C. Paar (eds.): *Workshop on Cryptographic Hardware and Embedded Systems — CHES 2000*. Berlin, pp. 57–70.
4. Guajardo, J. and C. Paar: 1997, 'Efficient Algorithms for Elliptic Curve Cryptosystems'. In: B. Kaliski (ed.): *Advances in Cryptology — CRYPTO '97*. Berlin, pp. 342–356. Lecture Notes in Computer Science Volume 1294.
5. Harper, G., A. Menezes, and S. Vanstone: 1992, 'Public-Key Cryptosystems with very small key lengths'. In: R. A. Rueppel (ed.): *Advances in Cryptology — EUROCRYPT '92*. Berlin, pp. 163–173. Lecture Notes in Computer Science Volume 658.
6. Hasan, M., M. Wang, and V. Bhargava: 1992, 'Modular Construction of Low Complexity Parallel Multipliers for a Class of Finite Fields $GF(2^m)$'. *IEEE Transactions on Computers* **41**(8), 962–971.
7. Itoh, T.: 1991, 'Characterization for a family of infinitely many irreducible equally spaced polynomials'. *Information Processing Letters* **37**(5), 273–277.
8. 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.
9. Itoh, T. and S. Tsujii: 1989, 'Structure of Parallel Multipliers for a Class of Fields $GF(2^k)$'. *Information and Computation* **83**, 21–40.
10. Kobayashi, T., H. Morita, K. Kobayashi, and F. Hoshino: 1999, 'Fast Elliptic Curve Algorithm Combining Frobenius Map and Table Reference to Adapt to Higher Characteristic'. In: Jacques Stern (ed.): *Advances in Cryptology — EUROCRYPT '99*. Berlin, pp. 176–189. Lecture Notes in Computer Science Volume 1592.
11. Koblitz, N.: 1987, 'Elliptic Curve Cryptosystems'. *Mathematics of Computation* **48**, 203–209.
12. Koblitz, N.: 1989, 'Hyperelliptic Cryptosystems'. *Journal of Cryptology* **1**(3), 129–150.
13. Koblitz, N.: 1998, 'An Elliptic Curve Implementation of the Finite Field Digital Signature Algorithm'. In: Hugo Krawczyk (ed.): *Advances in Cryptology —*

CRYPTO '98. Berlin, pp. 327–337. Lecture Notes in Computer Science Volume 1462.

14. Lee, E. J., D. S. Kim, and P. J. Lee: 1998, 'Speed-up of F_{p^m} Arithmetic for Elliptic Curve Cryptosystems'. In: *ICICS '98*. Berlin.
15. Lidl, R. and H. Niederreiter: 1983, *Finite Fields*, Vol. 20 of *Encyclopedia of Mathematics and its Applications*. Reading, Massachusetts: Addison-Wesley.
16. Menezes, A. J.: 1993, *Application of Finite Fields*. Boston: Kluwer Academic Publishers.
17. Miller, V.: 1986, 'Use of Elliptic Curves in Cryptography'. In: H. C. Williams (ed.): *Advances in Cryptology — CRYPTO '85*. Berlin, pp. 417–428. Lecture Notes in Computer Science Volume 218.
18. Paar, C.: 1996, 'A New Architecture for a Parallel Finite Field Multiplier with Low Complexity Based on Composite Fields'. *IEEE Transactions on Computers* **45**(7), 856–861.
19. Schroeppel, R., H. Orman, S. O'Malley, and O. Spatscheck: 1995, 'Fast Key Exchange with Elliptic Curve Systems'. In: D. Coppersmith (ed.): *Advances in Cryptology — CRYPTO '95*. Berlin, pp. 43–56. Lecture Notes in Computer Science Volume 963.
20. Wah, P. and M. Wang: 1984, 'Realization and application of the Massey-Omura lock'. In: *Proc. International Zurich Seminar*. Switzerland.
21. Win, E. D., A. Bosselaers, S. Vandenberghe, P. D. Gersem, and J. Vandewalle: 1996, 'A Fast Software Implementation for Arithmetic Operations in $GF(2^n)$'. In: K. Kim and T. Matsumoto (eds.): *Advances in Cryptology — ASIACRYPT '96*. Berlin, pp. 65–76. Lecture Notes in Computer Science Volume 1233.
22. Wu, H. and M. Hasan: 1998, 'Low Complexity Bit-Parallel Multipliers for a Class of Finite Fields'. *IEEE Transactions on Computers* **47**(8), 883–887.