Arithmetic Operations in $GF(2^m)$

G. B. Agnew
Department of Electrical and Computer Engineering, University of Waterloo,
Waterloo, Ontario, Canada N2L 3G1

T. Beth
Universität Karlsruhe, Postfach 6980,
D-7500 Karlsruhe 1, Federal Republic of Germany

R. C. Mullin and S. A. Vanstone
Department of Combinatorics and Optimization and Department of Computer Science,
University of Waterloo, Waterloo, Ontario, Canada N2L 3G1

Communicated by Ernest F. Brickell
Received 26 January 1988 and revised 25 October 1991

Abstract. This article is concerned with various arithmetic operations in $GF(2^m)$. In particular we discuss techniques for computing multiplicative inverses and doing exponentiation. The method used for exponentiation is highly suited to parallel computation. All methods achieve much of their efficiency from exploiting a normal basis representation in the field.

Key words. Public key cryptography, Normal basis, Discrete exponentiation.

1. Introduction

In this article we are concerned with arithmetic operations in the finite field $GF(2^m)$. In particular we discuss the computation of multiplicative inverses and exponentiation.

We can think of the elements in $GF(2^m)$ as being $m$-tuples which form an $m$-dimensional vector space over $GF(2)$. If

$$\beta, \beta^2, \beta^4, \ldots, \beta^{2^m-1}$$

is a basis for this space, then we call it a normal basis and we call $\beta$ a generator of the normal basis. It is well known [5] that $GF(2^m)$ contains a normal basis for every $m \geq 1$. It is of interest to point out that recently it was shown [4] that a normal basis exists in $GF(2^m)$ with the additional property that a generator of the normal basis is also a generator for the entire multiplicative cyclic group of the field.
For \( a \in GF(2^m) \) let \((a_0, a_1, \ldots, a_{m-1})\) be the coordinate vector of \( a \) relative to the ordered normal basis \( N \) generated by \( \beta \). It follows that \( a^2 \) then has coordinate vector \((a_m - 1, a_0, a_1, \ldots, a_m - 2)\), so squaring is simply a cyclic shift of the vector representation of \( a \). In a hardware implementation squaring an element takes one clock cycle and so is negligible. For the remainder of this article we assume that squaring an element is "free". In Section 2 we address the problem of computing inverses in \( GF(2^m) \) and in Section 3 we discuss methods for speeding up exponentiation in such fields.

2. Computing Inverses

Let \( \alpha \) be any nonzero element of \( GF(2^m) \). Suppose we want to compute \( \alpha^{-1} \). We observe that

\[
\alpha^{-1} = \alpha^{2^{m-2}} = \alpha^{\sum_{i=1}^{m-1} 2^i} = \prod_{i=1}^{m-1} \alpha^{2^i}.
\]

This can be computed in \( m - 2 \) multiplications. We now describe several techniques for improving the situation.

The most efficient technique in terms of minimizing the number of multiplications to compute an inverse was proposed by Itoh et al. \cite{3}. Since this reference is not easily accessible, we describe the technique and give a proof of its operation count below.

Consider \( GF(2^m) \) and the factorizations

\[
2^{m-1} - 1 = \begin{cases} 
(2^{(m-1)/2} - 1)(2^{(m-1)/2} + 1), & m \text{ odd,} \\
2^{m-2} + (2^{(m-2)/2} - 1)(2^{(m-2)/2} + 1), & m \text{ even.}
\end{cases}
\]

For \( m \) odd, we require one multiplication to compute \( \alpha^{2^{m-1}} \) assuming \( \alpha^{2^{(m-1)/2}-1} \) has been evaluated. For \( m \) even, we require two multiplications to calculate \( \alpha^{2^{m-1}} \) assuming \( \alpha^{2^{(m-1)/2}-1} \) has been evaluated. Hence, to compute \( \alpha^{-1} \) in \( GF(2^m) \) we show that using this procedure recursively the number of multiplications required is

\[
 nb(m - 1) + \omega(m - 1) - 2, \tag{*}
\]

where \( nb(x) \) is defined as the minimum number of bits required to represent the integer \( x \) and \( \omega(m - 1) \) is the Hamming weight of the binary representation of \( m - 1 \). We note that there are precisely \( \omega(m - 1) \) values of \( t \) in the recursion where \( \alpha^{2^t} \) must be evaluated and \( t \) is odd. Before giving the proof we consider an example.

\footnote{In our comparison of techniques based on the number of clock cycles required, the overhead for squaring will be taken into account.}