Efficient Arithmetic in Finite Field Extensions with Application in Elliptic Curve Cryptography

Daniel V. Bailey[1] and Christof Paar[2]

[1] CS Department, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609 USA
Email: bailey@cs.wpi.edu
[2] ECE and CS Departments, Worcester Polytechnic Institute, 100 Institute Road, Worcester, MA 01609 USA
Email: christof@ece.wpi.edu

To appear in Journal of Cryptology

Abstract. This contribution focuses on a class of Galois field used to achieve fast finite field arithmetic which we call an Optimal Extension Field (OEF), first introduced in [3]. We extend this work by presenting an adaptation of Itoh and Tsujii's algorithm for finite field inversion applied to OEFs. In particular, we use the facts that the action of the Frobenius map in $GF(p^m)$ can be computed with only $m-1$ subfield multiplications and that inverses in $GF(p)$ may be computed cheaply using known techniques. As a result, we show that one extension field inversion can be computed with a logarithmic number of extension field multiplications. In addition, we provide new extension field multiplication formulas which give a performance increase. Further, we provide an OEF construction algorithm together with tables of Type I and Type II OEFs along with statistics on the number of pseudo-Mersenne primes and OEFs. We apply this new work to provide implementation results using these methods to construct elliptic curve cryptosystems on both DEC Alpha workstations and Pentium-class PCs. These results show that OEFs when used with our new inversion and multiplication algorithms provide a substantial performance increase over other reported methods.

Keywords

finite fields, fast arithmetic, binomials, modular reduction, elliptic curves, inversion

1 Introduction

Since their introduction by Victor Miller [19] and Neil Koblitz [13], elliptic curve cryptosystems (ECCs) have been shown to be a secure and computationally efficient method of performing public-key operations. Our focus in the present paper is the efficient realization of ECCs in software. Our approach focuses on the finite field arithmetic required for ECCs. Finite fields are identified with the notation $GF(p^m)$, where p is a prime and m is a positive integer. It is well known that finite fields exist for any choice of prime p and integer m.

A standard technique in the development of symmetric-key systems has been to design a cipher to be efficient on a particular type of platform. For example, the International Data Encryption Algorithm [15] and RC5 [23] are designed to use operations that are efficient on desktop-class microprocessors. In addition, the NIST/ANSI Data Encryption Algorithm has been designed so that hardware realizations are particularly efficient [20] [1].