

# Proposing the Use of Non-Conventional Basis of Finite Fields

CryptoLab<sup>1</sup>  
Department of Mathematics  
Korea University  
and  
Chang Han Kim  
Department of Computational Mathematics  
Semyung University

*Background of Proposal:* Elliptic Curve Cryptosystem(ECC) is well-suitable for use in constrained environments<sup>2</sup> such as mobile devices and smart cards, since it provides more efficient, high-strength security with smaller key sizes than any known Public-key cryptosystem. We focus on the interoperability of a hardware-based system(e.g. cryptographic VLSI chip-embedded mobile devices) and a software-based system (e.g. smart cards) for ECC. The implementation of ECC is accomplished essentially by arithmetic in finite fields, in particular  $GF(2^n)$  of characteristic two or  $GF(p)$  of odd prime  $p$ . In VLSI implementation or smart card with no additional coprocessor for modular exponentiation the common choice for the underlying finite field is a class of fields  $GF(2^n)$ . The efficient computation of field arithmetic depends greatly on the particular ways in which the field elements are represented. These most efficient ways are a polynomial basis representation and a normal basis representation. The difficulty of communication between these two systems for ECC results from the choice of basis; the most common choices of basis for software implementation and hardware implementation are a polynomial basis and a normal basis respectively. Interoperability between these systems using the two different types of field representation needs a conversion of basis by a appropriate change-of-basis matrix. Since the size of key for ECC is becoming increasingly large in proportion to the growth of computing power, this method seems to be a significant burden for the space-sensitive systems such as mentioned above. To enhance usability of the space-sensitive systems Kaliski Jr.*et al* presented algorithm for the storage-efficient finite field basis

---

<sup>1</sup>It consists of Jong In Lim, Ok Yeon Yi, Joong Chul Yoon, Sang Ho Oh, Seok Hie Hong, Dong Hyun Cheon, and Sung Jae Lee.

<sup>2</sup>limits in memory usage and processing power.

conversion[2]. The time complexity of this algorithm for a basis conversion amounts to 10%  $\sim$  20% of one elliptic curve scalar multiplication according to a choice of basis. Such a fact gives us some motive for the communication with no basis conversion. Hereafter we consider finite fields  $GF(2^n)$ .

*Description of Non-Conventional Basis:*

**Definition 1** *If a subset  $B = \{\alpha, \alpha^2, \alpha^3, \dots, \alpha^n\}$  of  $GF(2^n)$  is linearly independent over  $GF(2)$ , then we'll call a set  $B$  an anomalous basis of  $GF(2^n)$  over  $GF(2)$ .*

Let  $f$  be a monic irreducible polynomial of degree  $n$  over  $GF(2)$  where  $\alpha$  is a root of  $f$ . Then the set  $\{\alpha, \alpha^2, \alpha^3, \dots, \alpha^n\}$  is linearly independent over  $GF(2)$ . Note that an anomalous basis  $\{\alpha, \alpha^2, \alpha^3, \dots, \alpha^n\}$  is not equal to a polynomial basis  $\{1, \alpha, \alpha^2, \dots, \alpha^{n-1}\}$  over  $GF(2)$ . We'll pay attention to a special case of a generating polynomial  $f$ , that is all-one-polynomials(AOP). The finite field arithmetic operations derived from such polynomials are performed very efficiently in both software implementation and hardware implementation.

*Claimed Attributes and Advantages:* Let  $B = \{\alpha, \alpha^2, \alpha^3, \dots, \alpha^n\}$  be an anomalous basis of  $GF(2^n)$  over  $GF(2)$  where  $\alpha$  is a root of AOP of the degree  $n$  and let  $\mathbf{a}$  be an element of  $GF(2^n)$ . Then they can be represented by a basis  $B$  as follows:

$$\mathbf{a} = \sum_{i=1}^n a_{i-1} \alpha^i$$

where  $a_i \in GF(2)$  for each  $i \in \{0, 1, 2, \dots, n-1\}$ .

In this section we'll introduce software and hardware attributes of an anomalous basis constructed using AOP and claim its advantages over conventional bases.

*a. Software Implementation* A field multiplication consists of two step, that is, the product of two polynomials and the reduction by a generating polynomial. First step is independent of a choice of basis. In case of second step, the common irreducible polynomial for the efficient reduction is a trinomial or a pentanomial. The identity of AOP,  $\alpha^{n+1} = 1$ , tells us that the reduction by

AOP is performed faster than that by a trinomial or a pentanomial. A squaring operation can be represented by a simple vector-form<sup>3</sup>. For example, if  $n = 10$ , then  $\mathbf{a}^2$  can be described as  $(a_5, a_0, a_6, a_1, a_7, a_2, a_8, a_3, a_9, a_4)$ .

**Note 1**

$$\begin{aligned}
 \mathbf{a}^2 &= \left( \sum_{i=1}^n a_{i-1} \alpha^i \right)^2 \\
 &= \left( \sum_{i=1}^{2m} a_{i-1} \alpha^i \right)^2 \\
 &= \sum_{i=1}^{2m} a_{i-1} \alpha^{2i} \\
 &= \sum_{i=1}^m a_{i-1} \alpha^{2i} + \sum_{i=m+1}^{2m} a_{i-1} \alpha^{2i} \\
 &= \sum_{i=1}^m a_{i-1} \alpha^{2i} + \sum_{i=1}^m a_{m+i-1} \alpha^{2i-1}.
 \end{aligned}$$

where  $n = 2m$  for some positive integer  $m$ .

Almost Inverse algorithm[5] has been known as the most improved algorithm for a multiplicative inversion. it is applicable to computing a multiplicative inverse of non-zero field elements represented by an anomalous basis.

Arithmetic in the fields  $GF(2^{180})$  derived from an anomalous basis, was implemented in the C-language on a Pentium 120. In Table 1 we give the running times for the operations of multiplication, squaring, and inversion in  $\mu$  seconds and compare our results with those of [6]<sup>4</sup>.

	anomalous basis using AOP	standard basis using trinomial
multiplication	51.4	71.8
squaring	1.5	2.7
inversion	161.4	225

Table 1. Time for field operations

<sup>3</sup>The coordinate vector  $(a_0, a_1, \dots, a_{n-1})$  with the ordered anomalous basis  $B$  can be interpreted as  $\sum_{i=1}^n a_{i-1} \alpha^i$ .

<sup>4</sup>The work implemented arithmetic in  $GF(2^{177})$  using Pentium 133.

*b. Hardware Implementation* Hardware implementation of field arithmetic using an anomalous basis representation was introduced already in [3]. As compared to the previous parallel multipliers, the parallel multiplier using an anomalous basis representation gives the lower or same complexity for time and space. Here we'll mention only the result. The reader is referred to [3] for a full detail of hardware implementation.

	multiplication (squaring)
anomalous basis (AOP)[3]	$n^2$ AND, $n^2 - 1$ XOR gates, $D_A + (1 + \lceil \log_2(n - 1) \rceil)D_X$ Delays (Rewiring)
standard basis (AOP)[4]	$n^2$ AND, $n^2 - 1$ XOR gates, $D_A + (2 + \lceil \log_2(n - 1) \rceil)D_X$ Delays ( $n - 1$ XOR gates, $1D_X$ Delay)
normal basis (type I)[1]	$n^2$ AND, $n^2 - 1$ XOR gates, $D_A + (1 + \lceil \log_2(n - 1) \rceil)D_X$ Delays (Rewiring)

Table 2. Space and time complexities for field operations<sup>5</sup>

Notice that squaring an element takes one clock cycle, like a normal basis representation. With the simple squaring property of the proposed basis representation used together with this multiplier, an efficient hardware architecture for computing a multiplicative inverse in  $GF(2^n)$  can be constructed using optimal inverse algorithm[7], which needs 9 to 12 multiplications and 149 to 199 squarings for  $150 \leq n \leq 200$ .

We could gain great advantages in both of software implementation and hardware implementation from the use of the proposed basis representation. In other words, it contains all benefits of a polynomial basis representation and a normal basis representation. In conclusion, we suggest to IEEE Standard P1363 or P1363a that an anomalous basis representation should be used as a new alternative for implementation of ECC.

**cf.**

[3] and [7] can be found as preprints in "<http://bora.dacom.co.kr/~gausmath>".

---

<sup>5</sup> $D_A$  and  $D_X$  are time delays for AND gates and XOR gates respectively.

*References:*

- [1] M. A. Hasan, M. Z. Wang, and V. K. Bhargava, "*A Modified Massey-Omura parallel multiplier for a class of finite fields*", IEEE Transactions on Computers, v42, no.10, pp.1278-1280, October 1993.
- [2] B. S. Kaliski Jr and Y. L. Yin, "*Storage-Efficient Finite Field Basis Conversion*", Contribution to IEEE Standard P1363, 1998.
- [3] C. H. Kim, S. H. Oh, and J. I. Lim, "*A New Hardware Architecture of Operations in  $GF(2^n)$* ", Submitted to IEEE Transactions on Computers, 1998.
- [4] C. K. Koc and B. Sunar, "*Low-complexity Bit-parallel Canonical and Normal Basis Multiplier for a Class of Finite Fields*", IEEE Transactions on Computers, v47, no.3, pp.353-356, March 1998.
- [5] R. Schroepfel, H. Orman, S. O'Malley, and O Spatscheck, "*Fast Key Exchange with Elliptic Curve Systems*", Proc. Crypto'95, Springer-Verlag, 1995, pp.43-56.
- [6] E. De Win, A. Bosselaers, S. Vandenberghe, P De Gersem, and J. Vandewalle, "*A Fast Software Implementation for Arithmetic Operations in  $GF(2^t)$* ", Asia Crypto'96, Springer-Verlag, 1996, pp.65-76.
- [7] S. H. Oh and C. H. Kim, "*Algorithm of Inverse Operation in  $GF(2^n)$* ", Submitted to IEEE Transactions on Information Theory, 1998.