Abstract. In the 1990s and early 2000s several papers investigated the relative merits of polynomial-basis and normal-basis computations for $F_{2^n}$. Even for particularly squaring-friendly applications, such as implementations of Koblitz curves, normal bases fell behind in performance unless a type-I normal basis existed for $F_{2^n}$.

In 2007 Shokrollahi proposed a new method of multiplying in a type-II normal basis. Shokrollahi’s method efficiently transforms the normal-basis multiplication into a single multiplication of two size-$(n+1)$ polynomials.

This paper speeds up Shokrollahi’s method in several ways. It first presents a simpler algorithm that uses only size-$n$ polynomials. It then explains how to reduce the transformation cost by dynamically switching to a ‘type-II optimal polynomial basis’ and by using a new reduction strategy for multiplications that produce output in type-II polynomial basis.

As an illustration of its improvements, this paper explains in detail how the multiplication overhead in Shokrollahi’s original method has been reduced by a factor of $1.4$ in a major cryptanalytic computation, the ongoing attack on the ECC2K-130 Certicom challenge. The resulting overhead is also considerably smaller than the overhead in a traditional low-weight-polynomial-basis approach. This is the first state-of-the-art binary-elliptic-curve computation in which type-II bases have been shown to outperform traditional low-weight polynomial bases.

Keywords: Optimal normal basis, ONB, polynomial basis, transformation, elliptic-curve cryptography.

1 Introduction

If $n+1$ is prime and $2$ has order $n$ modulo $n+1$ then the field $F_{2^n} = F_2[\zeta]/(\zeta^n + \cdots + \zeta + 1)$ has a “type-I optimal normal basis” $\zeta, \zeta^2, \zeta^4, \ldots$. It has been known
for many years that this basis allows not only fast repeated squarings but also surprisingly fast multiplications, costing only $M(n) + 2n - 2$ bit operations where $M(n)$ is the minimum cost of multiplying $n$-coefficient polynomials. The idea is to permute the basis into $\zeta, \zeta^2, \zeta^3, \ldots, \zeta^n$, and to decompose multiplication into the following operations:

- $M(n)$ bit operations: multiply the polynomials $f_1\zeta + \cdots + f_n\zeta^n$ and $g_1\zeta + \cdots + g_n\zeta^n$ in $\mathbb{F}_2[\zeta]$.
- $n - 2$ bit operations: eliminate the coefficients of $\zeta^{n+2}, \ldots, \zeta^{2n}$ using the identities $\zeta^{n+2} = \zeta, \ldots, \zeta^{2n} = \zeta^{n-1}$; this requires additions to the existing coefficients of $\zeta^2, \ldots, \zeta^{n-1}$.
- $n$ bit operations: eliminate the coefficient of $\zeta^{n+1}$ using the identity $\zeta^{n+1} = \zeta + \zeta^2 + \cdots + \zeta^n$.

An alternative introduced in [IT89] is to use a redundant representation, specifically coefficients of $1, \zeta, \ldots, \zeta^n$, with arithmetic modulo $\zeta^{n+1} + 1$. Multiplication then costs $M(n + 1) + n$ bit operations; this is worse than $M(n) + 2n - 2$ for small $n$, but it becomes better for large $n$, since $M(n)$ is subquadratic.

However, most integers $n$ do not have type-I optimal normal bases. In particular, an odd prime $n$ cannot have a type-I optimal normal basis. This poses severe problems for cryptographic applications that, for security reasons, prohibit composite values of $n$.

The conventional wisdom for many years was that type-I normal bases were a unique exception. For all other normal bases the best multiplication methods in the literature were quite slow. In particular, multiplication in a “type-II optimal normal basis” of $\mathbb{F}_{2^n}$ was asymptotically at least twice as expensive as multiplication in traditional low-weight polynomial bases (trinomial bases and pentanomial bases):

- Traditional normal-basis multipliers use $\Theta(n^2)$ bit operations.
- The type-II multiplier in [FH07] uses approximately $13 \cdot 3^{\lceil \log_2 n \rceil}$ bit operations.
- The type-II multiplier in [BG01, Section 4.1] uses approximately $3M(n)$ bit operations.
- The type-II multiplier in [GvzGP95] (see also [GvzGPS00]) uses approximately $2M(n)$ bit operations.

Normal bases were competitive only in extreme situations: applications where $n$ was very small; applications having many repeated squarings and very few multiplications; and applications that imposed extremely small hardware-area requirements, effectively punishing polynomial bases by prohibiting fast-multiplication techniques.

The picture changed a few years ago when Shokrollahi introduced a new type-II multiplier using only $M(n) + O(n \log_2 n)$ operations. See Shokrollahi’s thesis [Sho07, Chapter 4] and the subsequent WAIFI 2007 publication [vzGSS07] by von zur Gathen, Shokrollahi, and Shokrollahi. This new multiplier makes type-II normal bases competitive with traditional low-weight polynomial bases for a