High-Speed Unified Elliptic Curve Cryptosystem on FPGAs Using Binary Huff Curves

Ayantika Chatterjee and Indranil Sengupta

School of Information Technology
Indian Institute of Technology Kharagpur, India
cayantika@gmail.com, isg@iitkgp.ac.in

Abstract. Conventional Elliptic Curve (EC) cryptosystems are subjected to side channel attacks because of their lack of unifiedness. On the other hand, unified cryptosystems based on Edwards curves have been found to be slow. The present paper proposes the first VLSI design of binary Huff curves, which also lead to unified scalar multiplication. Several optimized architectural features have been developed to utilize the FPGA resources better, and yet lead to a faster circuit. Experimental results have been presented on the standard NIST curves, and on state-of-the-art $GF(2^{233})$ to show that the design is significantly faster than other unified EC cryptosystems.

Keywords: Binary Huff Curve (BHC), Elliptic Curve Cryptography (ECC), FPGA.

1 Introduction

Elliptic curve cryptography (ECC) is favored in the world of Public Key Cryptography (PKC) due to its smaller key size requirement over RSA. Hence ECC started a new era in the field of public key cryptosystems, once it was introduced by Miller and Koblitz in the year 1985 [9,10]. Considering a point on the curve $(x,y) \in GF(2^n) \times GF(2^n)$, and the curve constants $a_1, a_2, a_3, a_4, a_6 \in GF(2^n)$, the Weierstrass form of Elliptic curve is defined as:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

The security of ECC is mainly the hardness of its ECDLP property. To implement this scalar multiplication, point addition and doubling are necessary operations. But, the addition laws on ECC are not unified, hence the difference of the doubling and addition laws make the system vulnerable against SCA [7].

Due to these reasons, different forms of ECC have been investigated which are capable of having unified addition and doubling laws. Binary Huff Curve [5] is a solution of such problems which is the binary extension of Huff’s model [8]. The main advantage of this curve is the presence of unified addition and doubling law and hence processor based on this curve is expected to be simple side channel
attack preventive. We show in this paper, that the unified addition for BHC can
be efficiently implemented on FPGA resources to lead to a processor which is
faster than other unified curve namely binary Edwards curves.

The remaining part of this paper is organized as follows. In section 2 properties and algorithms related to BHC are discussed. Our contribution about the
detailed design of the processor are described in detail in section 3. At the end,
the time and area requirement of this design are explained and the results are
compared with some previous implementations in section 4.

2 Binary Huff Curves

A binary Huff curve [5] is defined as the set of projective points \((X : Y : Z) \ F_{2^m}\)
satisfying the equation, considering \(a, b \in F^*_{2^m}\) and \(a \neq b\):

\[
E : aX(Y^2 + YZ + Z^2) = bY(X^2 + XZ + Z^2);
\]

- There are three points at infinity satisfying the curve equation, namely \((a : b : 0), (1 : 0 : 0), \text{and} (0 : 1 : 0)\).
- This curve is birationally equivalent [5] to the Weierstrass elliptic curve \(v(v + (a + b)u) = u(u + a^2)(u + b^2)\).
- This curve deals with unified addition laws, but the laws are complete in
certain proper subgroups, which can be used for most cryptographic appli-
cations.

Let, sum of two points \((X_1, Y_1)\) and \((X_2, Y_2)\) be \((X_3, Y_3)\). The resultant point
on the projective curve is defined according to the following addition formulas.
In this paper, we shall concentrate on the design of a ECC processor based on
projective unified addition laws of BHC. This can be evaluated as:

**Table 1. Addition Law**

\[
\begin{align*}
m_1 &= X_1X_2; m_2 = Y_1Y_2; m_3 = Z_1Z_2; \\
m_4 &= (X_1 + Z_1)(X_2 + Z_2) + m_1 + m_3; \\
m_5 &= (Y_1 + Z_1)(Y_2 + Z_2) + m_2 + m_3; \\
m_6 &= m_1m_3; m_7 = m_2m_3; m_8 = m_1m_2 + m_3^2; \\
m_9 &= m_6(m_2 + m_3)^2; m_{10} = m_7(m_1 + m_3)^2; \\
m_{11} &= m_8(m_2 + m_3); m_{12} = m_8(m_1 + m_3); \\
X_3 &= m_4m_{11} + \alpha.m_9; \\
Y_3 &= m_5m_{12} + \beta.m_{10}; \\
Z_3 &= m_{11}(m_1 + m_3);
\end{align*}
\]

Here, \(\alpha = \frac{a+b}{b}\) and \(\beta = \frac{a+b}{a}\) are two constants and the total cost of this projective
addition is \(15M + 2D\) (\(M\) implies multiplication and \(D\) implies multiplication
with constants).