Microprocessors and Microsystems 000 (2016) 1–12

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

High performance hardware support for elliptic curve cryptography over general prime field

Khalid Javeed[a,*], Xiaojun Wang[b,c], Mike Scott[d]

[a] Department of Electrical Engineering, CIIT, Abbottabad, Pakistan
[b] School of Electronics Engineering, Dublin City University, Dublin, Ireland
[c] School of Computer & Software, Nanjing University of Information Science and Technology, Nanjing, Jiangsu, China
[d] Cetivox, 81 Rivington St, London EC2A 3AY, UK

ARTICLE INFO

Article history:
Received 11 June 2015
Revised 31 October 2016
Accepted 9 December 2016
Available online xxx

Keywords:
Modular multiplier
Finite field arithmetic
FPGA
Elliptic curve scalar multiplier

ABSTRACT

Secure information exchange in resource constrained devices can be accomplished efficiently through elliptic curve cryptography (ECC). Due to the high computational complexity of ECC arithmetic, a high performance dedicated hardware architecture is essential to provide sufficient performance in a computation of elliptic curve scalar multiplication. This paper presents a high performance hardware support for elliptic curve cryptography over a prime field $GF(p)$. It exploited a best available possible parallelism of elliptic curve points in projective representation. The proposed hardware for ECC is implemented on Xilinx Virtex-4, Virtex-5 and Virtex-6 FPGAs. A 256-bit scalar multiplication is completed in 2.01 ms, 2.62 ms and 3.91 ms on Virtex-6, Virtex-5 and Virtex-4 FPGA platforms, respectively. The results show that the proposed design is 1.96 times faster with insignificant increase in area consumption as compared to the other reported designs. Therefore, it is a good choice to be used in many ECC based schemes.

1. Introduction

Security protocols based on elliptic curve cryptography (ECC) proposed by Koblitz [1] and Miller [2], have been widely accepted. It is now considered as one of the best Public Key Cryptography (PKC) algorithms and provides much higher security strength per bit than RSA [3]. For example, a 256-bit ECC can provide an equivalent security level as in comparison to 3072-bit RSA. Due to this lower bit sizes, it is considered as the most suitable choice for resource constrained devices. Therefore, to accomplish the speed requirements in real time applications, its efficient hardware implementation is of prime importance. Field programmable gate arrays (FPGAs) is considered as the most suitable hardware platform for implementation of computational intensive security algorithms including ECC.

In all elliptic curve EC cryptosystems, elliptic curve scalar multiplication is a primary and the most computational intensive operation. Mathematically, it is denoted as $Q = sP$, where P, a point on the EC is multiplied by a scalar s to produce a resultant point Q. It can be computed by adding a point P to itself $(s-1)$ times. The point P and Q are the public parameters, while scalar s is a secret (private value) that is required to be shared among users involved in the communication. The computational hardness of finding s given Q and P is known as EC discrete logarithm problem (ECDLP), which is the basis for all EC based schemes. The EC scalar multiplication operation is computed through a series of EC point addition and EC point doubling operations. Further, these group operations rely on finite field arithmetic primitives including addition, subtraction, multiplication, inversion and division. Efficient hardware implementation of these finite field primitives can significantly reduce the computation time for EC scalar multiplication operation. Side channel attacks are another way to find s by avoiding the ECDLP. These attacks are threat to the cryptographic hardware devices [4] and are based on exploiting timing and power leakage information to gain insight of the scalar s [5]. In this regard, detailed survey papers are reported in [6,7]. However, these attacks are not considered in this work. Moreover, there are many different elliptic curves representations offering different trade-offs between computational performance and security, therefore a flexible implementation that can adopt different values for curve parameters and the prime p is also desirable.

Since its proposal, extensive research work in underlying mathematics, security, and efficient implementation has been carried out. To date, numerous hardware accelerators and cryptographic processors for ECC have been proposed [8–19]. In this aspect, see [20,21] for a detailed review. EC based crypto-systems can be

* Corresponding author.
E-mail address: khalidjaveed@ciit.net.pk (K. Javeed).

http://dx.doi.org/10.1016/j.micpro.2016.12.005
0141-9331/

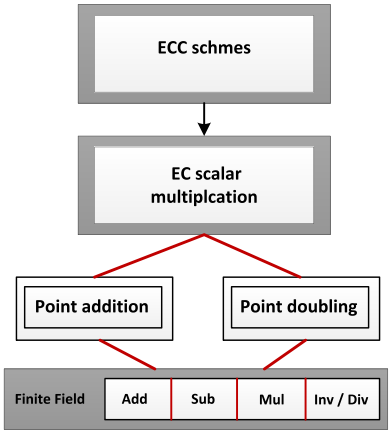

Fig. 1. ECA and ECD in affine coordinates.

partitioned into two categories : systems over special prime fields and systems over the general prime field. Typically, systems designed over special prime fields offer higher performance due to simpler and fast reduction steps in modular multiplication operation. In this regard, the National Institute Of Science and Technology (NIST) recommended five ECs defined over special prime fields [22]. However, designs over these NIST prime fields are not very flexible and lacks generality. Therefore, the main focus of this work is to provide both flexibility and performance to the users. Flexibility is provided in a way to select own prime number and curve parameters, while performance is offered in terms of computation time, area, clock cycles, and throughout.

Typical hierarchy of EC based cryptographic schemes is shown in Fig. 1. It is subdivided into four layers. Top layer consists of ECC based cryptographic protocols such as ECDSA, ECMQV [23]. The next layer is the EC scalar operation which is comprised of EC group operations subdivided into two parts: EC point addition and EC point doubling operation. Further down, these EC point operations consists of finite field arithmetic operations such modular addition, subtraction, multiplication and division. These finite field primitives are the bottleneck of any EC based cryptographic scheme, therefore they have a strong impact on the overall performance.

1.1. Contribution

This work presents a novel high performance hardware support for ECC capable of working for any prime number $p \leq$ 256-bit. Therefore, the proposed design is suitable to be integrated in many EC based crypto-systems to speed up the elliptic curve scalar multiplication operation. This is the most important characteristic of any design because it extends its adaptability to a range of server applications. The proposed hardware accelerator is based on parallel arithmetic unit, which consists of four modular multipliers and a single modular adder/subtractor units. As, modular multiplier is the most critical component in the design of elliptic curve scalar multiplication architecture especially over projective coordinates, so we used a radix-4 Booth encoding based interleaved modular multiplier [24]. This technique requires $\lceil k/2 \rceil + 2$ clock cycles to perform a single k-bit modular multiplication. Therefore, our parallel arithmetic unit can execute four modular multiplication instructions concurrently in $\lceil k/2 \rceil + 2$ clock cycles.

Our experimental results are based on two elliptic curve scalar multiplication algorithms: double-and-add and Non-adjacent-form recoding. In double-and-add method see Algorithm 1, a scalar s is encoded as a simple binary number of length k, where on average the Hamming weight of k is $k/2$ and hence, this technique requires $k-1$ point doubling and $k/2$ point addition operations. The NAF method (see Algorithm 2) is slightly efficient in a sense that it can reduce the Hamming weight of k form $k/2$ to $k/3$, therefore elliptic curve scalar multiplication based on this method requires $k-1$ point doubling and $k/3$ point addition/subtraction operations. Finally, we show performance comparison among the proposed hardware architecture with its contemporary designs. The experimental results reveal that our strategy is 1.96 times faster, consumes 1.94 times fewer clock cycles as compared to the existing designs.

Algorithm 1 Double and add method for scalar multiplication.

Require: *PointP* and scalar S represented as **NAF**
Ensure: $Q = s \times P$
1: $Q \leftarrow \infty$;
{k= binary representation of s}
2: **for** i from $k-1 \rightarrow 0$ **do**
3: $Q \leftarrow 2Q$ ||{Point doubling}
4: **if** $(s_i = 1)$ **then**
5: $Q \leftarrow Q + P$; ||{Point addition}
6: **end if**
7: **end for**
8: **return** Q

Algorithm 2 NAF scalar multiplication.

Require: Point P and scalar s
Ensure: $Q = s \times P$
1: $Q \leftarrow \infty$; {k= NAF representation of s}
2: **for** i from $k-1 \rightarrow 0$ **do**
3: $Q \leftarrow 2Q$ ||{Point doubling}
4: **if** $(s_i = 1)$ **then**
5: $Q \leftarrow Q + P$; ||{Point addition}
6: **else if** $(s_i = -1)$ **then**
7: $Q \leftarrow Q - P$; ||{Point subtraction}
8: **end if**
9: **end for**
10: **return** Q

The rest of the paper is organized as follows: Section 2 briefly explains the mathematical background of ECC. Section 3 presents hardware architectures for modular arithmetic primitives. The EC scalar multiplier architecture is presented in Section 4. Section 5 deals with the implementation results and performance comparison. Finally, the paper is concluded in Section 6.

2. Preliminaries

This work focuses on elliptic curve $\mathbb{K}$ defined over prime field $GF(p)$, where p is a large prime characteristic number greater than 3. Field elements are represented as integers in the range $[0 \rightarrow p-1]$. An elliptic curve $\mathbb{E}$ over $GF(p)$ in short Weierstrass form is represented as

$$\mathbb{K} : y^2 = x^3 + ax + b \tag{1}$$

where, a, b, x, and $y \in GF(p)$ and $4a^3 + 27b^2 \neq 0 \pmod{p}$. A point $P(x, y)$ in affine coordinate is said to be a point on the curve if it satisfies Eq. (1). The set of all such points plus the point at ∞ form an abelian group. The EC point addition and EC point doubling operations over these such groups are used to construct many elliptic curve crypto-systems. The *EC* point addition and *EC* point doubling operations in affine coordinates can be represented as follows: let $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$ are two points on the elliptic curve.

Please cite this article as: K. Javeed et al., High performance hardware support for elliptic curve cryptography over general prime field, Microprocessors and Microsystems (2016), http://dx.doi.org/10.1016/j.micpro.2016.12.005