Chapter 1 Introduction

1.1 Motivation

Neal Koblitz and Victor Miller independently proposed elliptic curve cryptography (ECC) in 1985 [1, 2]. ECC is considered a serious alternative to many public key encryption algorithms. With key sizes of 128 to 256 bits, ECC offers security equal to that of RSA [3], which has key sizes of 1000 to 2000 bits [4, 5]. No significant weaknesses have yet been identified in the ECC algorithm, as it depends on the discrete logarithm problem over points on an elliptic curve [6]. The difficulty of the problem allows ECC key sizes to be reduced considerably [3]. This advantage of ECC has recently gained remarkable recognition and has been incorporated in many standards, such as IEEE, ANSI, NIST, SEC and WTLS.

Scalar multiplication is the basic operation of ECC. The scalar multiplication of a group of points on an elliptic curve is comparable to the exponentiation of a multiplicative group of integers modulo a fixed integer m. The scalar multiplication operation is denoted as kP, where k is an integer and P is a point on the elliptic curve. The kP operation represents the addition of k copies of point P. Scalar multiplication is then conducted according to a series of point doubling and point addition operations of the point P, which depend on the bit sequence representing the scalar multiplier k. Several scalar multiplication methods have been proposed [6].

Identifying efficient scalar multiplication methods for high-performance end servers is crucial. Sequential scalar multiplication methods are too slow to meet the demands of the increasing number of customers for such servers. Scalar multiplication methods that can be parallelised are often used for high-speed implementations [7–12].

1.2 The contributions of this thesis

- We analysed all possible scenarios which may accelerate the scalar multiplication.
- We proposed a new method that shows the best result when compared with the others.
- We implemented the previous methods and the new using C++.
- We have analysed and discussed the results of the execution time for each method by comparing it with the others.

1.3 Thesis Organisation

Chapter 2 briefly addresses finite-field arithmetic and elliptic curves in general. Chapter 3 illustrates the work related to this thesis. Chapter 4 explains the methodology of this thesis and the proposed method. Chapter 5 presents and discusses the results of all methods tested in different cases. Finally, Chapter 6 concludes the thesis and discusses future work.