Information Sciences 245 (2013) 304–312

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Extended elliptic curve Montgomery ladder algorithm over binary fields with resistance to simple power analysis

Sung Min Cho [a], Seog Chung Seo [b], Tae Hyun Kim [c], Young-Ho Park [d], Seokhie Hong [a,*]

[a] Center for Information Security Technologies (CIST), Korea University, Seoul, Republic of Korea
[b] Samsung Advanced Institute of Technology (SAIT), Yongin, Republic of Korea
[c] Telecommunication Technology Association (TTA), Seongnam, Republic of Korea
[d] School of Computer Engineering, Sejong Cyber University, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 4 November 2010
Received in revised form 23 April 2013
Accepted 13 May 2013
Available online 18 May 2013

Keywords:
Composite formulas
Elliptic curve
Montgomery ladder algorithm
Side-channel atomicity
Simple power analysis

ABSTRACT

In this paper, we propose a scalar multiplication algorithm on elliptic curves over $GF(2^m)$. The proposed algorithm is an extended version of the Montgomery ladder algorithm with the quaternary representation of the scalar. In addition, in order to improve performance, we have developed new composite operation formulas and apply them to the proposed scalar multiplication algorithm. The proposed composite formulas are $2P_1 + 2P_2$, $3P_1 + P_2$, and $4P_1$, where P_1 and P_2 are points on an elliptic curve. They can be computed using only the x-coordinate of a point $P = (x,y)$ in the affine coordinate system. However, the proposed scalar multiplication algorithm is vulnerable to simple power analysis attacks, because different operations are performed depending on the bits of the scalar unlike the original Montgomery ladder algorithm. Therefore, we combine the concept of the side-channel atomicity with the proposed composite operation formulas to prevent simple power analysis. Furthermore, to optimize the computational cost, we use the Montgomery trick which can reduce the number of finite field inversion operations used in the affine coordinate system. As the result, the proposed scalar multiplication algorithm saves at least 26% of running time with small storage compared to the previous algorithms such as window-based methods and comb-based methods.

1. Introduction

Elliptic Curve Cryptography (ECC) was introduced in 1985 independently by Koblitz and Miller [14,18]. Since there are no known sub-exponential time algorithms to solve the Elliptic Curve Discrete Logarithm Problem (ECDLP), ECC provides an equivalent level of security with even smaller key size than other public key cryptosystems based on the Discrete Logarithm Problem (DLP) and the Integer Factoring Problem (IFP) over finite fields. The small key size of ECC is suited for low-powered mobile devices such as smart-cards, mobile phones, and PDAs. As the demands for low-powered devices have been increasing, many researches have been conducted for improving efficiency and security in ECC.

Scalar multiplication, defined as dP ($P + \cdots + P$, d times), where d is an integer and P is an elliptic curve point, is a major operation in ECC. It is computed with elliptic curve basic operations, elliptic curve point addition, i.e., $ECADD(P_1,P_2) = P_1 + P_2$, and elliptic curve point doubling, i.e., $ECDBL(P_1) = 2P_1$, where P_1 and P_2 are points on the elliptic curve, that are composed of several field operations. Many researches have been conducted to raise the efficiency of the scalar multiplication. There are two main

[*] Corresponding author. Tel.: +82 1062016348.
E-mail address: shhong@korea.ac.kr (S. Hong).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.05.009

approaches to reduce the running time of the scalar multiplication. The first approach is to reduce the number of elliptic curve basic operations that appear in the scalar multiplication algorithm. For example, window-based methods and comb-based methods use signed representations of the scalar and table look-up [2,20,22]. The second approach is to optimize the elliptic curve basic operations themselves. For example, elliptic curve operations can be computed by some efficient coordinate systems, such as projective coordinate and Jacobian projective coordinate, and composite operations, such as $2P_1 + P_2$ and $3P_1$ [5,9]. In addition, in the affine coordinate system, the use of the Montgomery trick can reduce the number of field inversions [6].

However, if the implementation of ECC is careless, an adversary can break the secret key by observing side channel informations [15,16]. This attack is called the Side Channel Analysis (SCA). Namely, the adversary obtains the secret key without solving ECDLP and decapsulating the physical devices. After the introduction of SCA, the implementation of ECC has to ensure resistance against SCA. In particular, Simple Power Analysis (SPA) is one of the powerful attacks for the scalar multiplication algorithms. Since elliptic curve basic operations, *ECADD* and *ECDBL*, have different costs and patterns, they can be distinguished from their power consumptions or computing timings. In addition, since the basic operation which is computed depends on the value of the secret key, most scalar multiplication algorithms are vulnerable to SPA. Thus, many countermeasures have been proposed to make the scalar multiplication resistant against SPA. The previous SPA countermeasures can be classified into two types. The first type makes the scalar multiplication a fixed pattern, independent of the secret key, such as the Montgomery ladder [13] and the double-and-add always method [7]. However, they require many dummy operations. And so, they can be slower than unprotected methods against SPA. The second type makes the basic operations indistinguishable, such as indistinguishable operation method [3] and side-channel atomicity [4]. They impose *ECADD* and *ECDBL* to have the same patterns and the number of field operations. Then, since power consumption of *ECADD* can be remarkably similar to that of *ECDBL*, an adversary cannot distinguish between *ECADD* and *ECDBL* by SPA. However, the second type can leak information for the Hamming weight of the secret key by counting elliptic curve point operations from a power consumption trace. For example, for the binary scalar multiplication algorithm, we can identify elliptic curve point operations from a power consumption trace, even if *ECADD* and *ECDBL* are not distinguished. The total number of elliptic curve point operations is the sum of the Hamming weight and the bit size of the scalar, i.e., the secret key. Since the parameters of cryptosystems are public, we know the bit size of the key. As the result, the Hamming weight of the key is the number of elliptic curve point operations minus the bit size of the key.

As mentioned above, the demerit of the Montgomery ladder algorithm is slow performance. Therefore, our aim is to improve the performance of the Montgomery ladder algorithm. The original Montgomery ladder algorithm uses the binary expression of the scalar. In this paper, we propose a new efficient and secure scalar multiplication algorithm based on the Montgomery ladder algorithm with the quaternary expression of the scalar. We also propose composite formulas with SPA resistance by applying the concept of side-channel atomicity. Here, the quaternary expression is useful to reduce the length of the scalar representation. For example, let the key be an n-bit long positive integer. Then, the length of the quaternary expression is $0.5n$. In conjunction with the quaternary expression, we propose new composite operations, such as elliptic curve point double-add-double, i.e., $ECDAD(P_1,P_2) = 2P_1 + 2P_2$, elliptic curve point triple-and-add, i.e., $ECTA(P_1,P_2) = 3P_1 + P_2$, and elliptic curve point quadrupling, i.e., $ECQPL(P_1) = 4P_1$. The proposed new composite operation formulas can utilize only the x-coordinate in the affine coordinate system. However, the proposed algorithm is vulnerable to SPA, because different operations are performed depending on the bits of the scalar unlike the original Montgomery ladder algorithm. In order to resist SPA, we apply side-channel atomicity to our composite operations. Furthermore, we use the Montgomery trick to speed up the processing. The proposed quaternary Montgomery ladder algorithm saves at least 26% of running time using only two storages compared to existing algorithms, such as window-based methods and comb-based methods using 8 storages when applying window size 4 [20,22].

2. Elliptic curve cryptosystem and side channel analysis

2.1. Elliptic curve cryptosystem

A non-supersingular elliptic curve over $GF(2^m)$ is given by the following Weierstrass equation [2].

$$E/GF(2^m) : y^2 + xy = x^3 + ax^2 + b \tag{1}$$

with $a, b \in GF(2^m)$, $b \neq 0$. The set of all the points in E together with the point at infinity O is denoted by $E(GF(2^m))$. It forms an abelian group with the point at infinity O as the identity element. Assume $P_1 = (x_1,y_1)$ and $P_2 = (x_2,y_2)$ are two points on E, where $P_2 \neq -P_1$. Then $P_3 = P_1 + P_2 = (x_3,y_3)$ on the affine coordinate system is computed as follows:

$$\begin{aligned} &x_3 = \lambda^2 + \lambda + x_1 + x_2 + a, \quad y_3 = (x_1 + x_3)\lambda + x_3 + y_1, \\ &\text{where } \lambda = \frac{(y_1 + y_2)}{(x_1 + x_2)} \text{ if } P_1 \neq P_2, \text{ and } \lambda = x_1 + \frac{y_1}{x_1} \text{ if } P_1 = P_2. \end{aligned} \tag{2}$$