DECOMPOSITION OF AN INTEGER FOR EFFICIENT IMPLEMENTATION OF ELLIPTIC CURVE CRYPTOSYSTEM

YOUNG-HO PARK

Abstract. This paper presents the Gallant-Lambert-Vanstone method for speeding up scalar multiplication of elliptic curves and an alternate decomposition method using the theory of $\mu$-Euclidean algorithm. Also the extended method to hyperelliptic curves over finite fields that have efficiently-computable endomorphisms is presented.

1. Introduction

Public key cryptosystems based on the discrete log problem on elliptic curves over finite fields (ECC) have gained much attention as a popular and practical scheme for resource-constrained devices. The dominant cost operation in ECC is scalar multiplication, that is, computing $kP$ for a point $P$ on an elliptic curve. So various methods for faster scalar multiplication have been devised by selecting relevant objects involving base fields and elliptic curves [1, 3, 4, 6, 7, 12, 13].

In Crypto 2001, Gallant, Lambert and Vanstone [3] introduced a new method for faster scalar multiplication on elliptic curves over (large) prime fields that have an efficiently-computable endomorphism. The key idea of their method is to decompose an arbitrary scalar $k$ into components such that the size of each component is a half of that of $k$. They gave an algorithm for decomposing $k$ into the desired form using the extended Euclidean algorithm but did not derive explicit bounds for decomposition components. In PKC 2002, Park, et al. [8] presented an alternate algorithm for decomposing an integer $k$ using the theory of $\mu$-Euclidean algorithm. This algorithm runs a little bit faster than that of Gallant et al.’s and gives explicit bounds for the components. In Eurocrypt 2002, they also extended this algorithm to hyperelliptic curves that have efficiently computable endomorphisms [9].

In this paper, we survey the Gallant-Lambert-Vanstone method and an alternate algorithm for decomposing an integer $k$ using the theory of $\mu$-Euclidean algorithm. Also the extended method to hyperelliptic curves over finite fields that have efficiently-computable endomorphisms is presented.

Key words and phrases. Elliptic Curve Cryptosystem, Hyperelliptic Curve, Scalar Multiplication, Decomposition of integer.

This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

©2005 Information Center for Mathematical Sciences
2. Gallant-Lambert-Vanstone method

Let $E$ be an elliptic curve over a finite field $\mathbb{F}_q$ and $\phi$ be an efficiently-computable endomorphism in $\text{End}(E)$. For cryptographic purposes, the order of $E(\mathbb{F}_q)$ must have a large prime factor $n$. Let $P \in E(\mathbb{F}_q)$ be a point of prime order $n$. Then the map $\phi$ acts on the subgroup of $E(\mathbb{F}_q)$ generated by $P$ as a multiplication by $\lambda$, where $\lambda$ is a root of the characteristic polynomial of $\phi$ modulo $n$. In place of the Frobenius, Gallant et al. exploited $\phi$ to speed up the scalar multiplication by decomposing an integer $k$ into a sum of the form $k = k_1 + k_2\lambda \pmod{n}$, where $k \in [1, n-1]$ and $k_1, k_2 \approx \sqrt{n}$. Now we compute

$$kP = (k_1 + k_2\lambda)P = k_1P + k_2\lambda P = k_1P + k_2\phi(P).$$

Since $\phi(P)$ can be easily computed, a windowed simultaneous multiple exponentiation applies to $k_1P + k_2\phi(P)$ for additional speedup. It is analyzed in [3] that this method improves a running time up to 66% compared with the general method, thus it is roughly 50% faster than the best general methods for 160-bit scalar multiplication.

We will now describe the algorithm in [3] for decomposing $k$ out of given integers $n$ and $\lambda$. It is composed of two steps. By considering the homomorphism $f : \mathbb{Z} \times \mathbb{Z} \to \mathbb{Z}_n$ defined by $(i, j) \mapsto (i + j\lambda) \pmod{n}$ we first find linearly independent short vectors $v_1, v_2 \in \mathbb{Z} \times \mathbb{Z}$ such that $f(v_1) = f(v_2) = 0$. As a stage of precomputations this process can be done by the Extended Euclidean algorithm, independently of $k$.

Secondly, one needs to find a vector in $\mathbb{Z}v_1 + \mathbb{Z}v_2$ that is close to $(k, 0)$ using linear algebra. Then $(k_1, k_2)$ is determined by the equation:

$$(k_1, k_2) = (k, 0) - ([b_1]v_1 + [b_2]v_2),$$

where $(k, 0) = b_1v_1 + b_2v_2$ is represented as an element in $\mathbb{Q} \times \mathbb{Q}$ and $[b]$ denotes the nearest integer to $b$.

In the procedure of finding two independent short vectors $v_1, v_2$ such that $f(v_1) = f(v_2) = 0$, Gallant, et al. showed $\| v_1 \| \leq 2\sqrt{n}$ but could not estimate $\| v_2 \|$ explicitly. However they expected heuristically that $v_2$ would also be short. For this reason, they could not give explicit upper bounds of $k_1$ and $k_2$ although the lengths of components prove to be near to $\sqrt{n}$ through numerous computational experiments.

3. An alternate decomposition method

We are now describing a new method for decomposing $k$ from a viewpoint of algebraic number theory. Recall that $\text{End}(E)$ is a quadratic order of $K = \mathbb{Q}(\sqrt{-D})$ ($D > 0$), which is contained in the maximal order of $K$, denoted $\mathcal{O}_K$. Let $\phi$ be an efficiently-computable endomorphism in $\text{End}(E)$. Then we have $\mathbb{Z}[\phi] \subset \text{End}(E) \subset \mathcal{O}_K$. Since $\phi$ is in general not a rational integer, it satisfies a quadratic relation

$$\phi^2 - t_{\phi}\phi + n_{\phi} = 0.$$
The existence of such \( a \) is guaranteed from Lemma 1 in [8].

We then want to decompose a scalar \( k \) using a division by \( \alpha \) in the \( \mu \)-Euclidean ring \( \mathbb{Z}[\phi] \), where \( \mu \) is some positive real (see [8] or [11]).

Viewing \( k \) as an element of \( \mathbb{Z}[\phi] \) we divide \( k \) by \( \alpha \) satisfying (2) in \( \mathbb{Z}[\phi] \) and write

\[
 k = \beta \alpha + \rho
\]

with \( N_{\mathbb{Z}[\phi]/\mathbb{Z}}(\alpha) < \mu N_{\mathbb{Z}[\phi]/\mathbb{Z}}(\alpha) \) for some \( \beta \) and \( \rho \in \mathbb{Z}[\phi] \). We then compute

\[
 kP = (\beta \alpha + \rho)(P) = \beta(\alpha(P)) + \rho(P) = \rho(P).
\]

From a representation of \( \rho \), that is, \( \rho = k_1 + k_2 \phi \), it turns out that

\[
 kP = \rho P = k_1 P + k_2 \phi(P).
\]

Since \( \phi(P) \) is easily computed we can apply a (windowed) simultaneous multiple exponentiation to yield the same running time improvement as in [3]. Unlike [3] our method gives rigorous bounds for the components \( k_1, k_2 \) in terms of \( n_\phi \). To see this, we give the following theorem estimating \( N_{\mathbb{Z}[\phi]/\mathbb{Z}}(\rho) \).

**Theorem 2.** Let \( \alpha = a + b\phi \neq 0 \in \mathbb{Z}[\phi] \). If \( \beta \in \mathbb{Z}[\phi] \) then there exist \( \delta, \rho \in \mathbb{Z}[\phi] \) such that \( \beta = \delta \alpha + \rho \) and \( N_{\mathbb{Z}[\phi]/\mathbb{Z}}(\rho) < \mu N_{\mathbb{Z}[\phi]/\mathbb{Z}}(\alpha) \) with

\[
 0 < \mu \leq \begin{cases} 
 9 + 4n_\phi & \text{if } \phi \text{ is odd,} \\
 1 + n_\phi & \text{if } \phi \text{ is even.}
\end{cases}
\]

**Proof.** See [8] \( \square \)

The result of Theorem 2 shows that the upper bound of \( \mu \) is better than that of N. Smart. In fact, he has an upper bound of \( \mu \) in [11] as follows:

\[
 N_{\mathbb{Z}[\phi]/\mathbb{Z}}(\rho) < \mu N_{\mathbb{Z}[\phi]/\mathbb{Z}}(\alpha) \text{ with } 0 < \mu \leq (9 + 4n_\phi)/4.
\]

### 3.1. Decomposition Algorithm

Now we have an efficient algorithm to compute a remainder \( \rho = k_1 + k_2 \phi \) from \( k \) and \( \alpha = a + b\phi \). It is also composed of two steps as in [3].

**Precomputations**

1. \( N_\alpha = N_{\mathbb{Z}[\phi]/\mathbb{Z}}(\alpha) = s_n n, \ t_\phi = Tr_{\mathbb{Z}[\phi]/\mathbb{Z}}(\phi) \) and \( c = -[t_\phi/2] \).

2. Set \( \phi' = \phi + c \). \( N = N_{\mathbb{Z}[\phi]/\mathbb{Z}}(\phi') \) and \( T = Tr_{\mathbb{Z}[\phi]/\mathbb{Z}}(\phi') = \begin{cases} 
 1 & \text{if } t_\phi \text{ is odd} \\
 0 & \text{otherwise.}
\end{cases} \)

3. \( a_1 = a - bc, \ b_1 = b \) (to represent \( \alpha = a_1 + b_1 \phi' \)).
Now we restrict ourselves to See [8].

Lemma 3.

\[ \text{If } t_\phi = 0 \text{ and } n_\phi = 1, \text{ then } N_{\mathbb{Z}[\phi]/\mathbb{Z}}(\alpha) \leq n/2 \text{ for } E_1, \]
\[ \text{if } t_\phi = -1 \text{ and } n_\phi = 1, \text{ then } N_{\mathbb{Z}[\phi]/\mathbb{Z}}(\alpha) \leq 3n/4 \text{ for } E_2, \]
\[ \text{if } t_\phi = 1 \text{ and } n_\phi = 2, \text{ then } N_{\mathbb{Z}[\phi]/\mathbb{Z}}(\alpha) \leq n \text{ for } E_3, \]
\[ \text{if } t_\phi = 0 \text{ and } n_\phi = 2, \text{ then } N_{\mathbb{Z}[\phi]/\mathbb{Z}}(\alpha) \leq 3n/4 \text{ for } E_4. \]

Proof. See [8] \square

\[ \square \]
Finally, Lemma 3 gives explicit upper bounds on the components of \( k \).

**Theorem 4.** For any \( k \), let \( \rho \) be a remainder of \( k \) divided by \( \alpha \) using Algorithm 2 and write \( \rho = k_1 + k_2 \phi \). Then we have

\[
\max\{|k_1|, |k_2|\} \leq \begin{cases} \sqrt{n/2} & \text{for } E_1, \\ \sqrt{n} & \text{for } E_2, \\ \sqrt{8n/7} & \text{for } E_3, \\ \sqrt{3n/2} & \text{for } E_4. \end{cases}
\]

**Proof.** See [8] □ □

4. **Extended method to hyperelliptic curves**

4.1. **GLV method using an efficient endomorphism on Jacobian.** Let \( X \) be a hyperelliptic curve over a finite field \( \mathbb{F}_q \) having an efficiently-computable endomorphism \( \phi \) on the Jacobian, \( J_X(\mathbb{F}_q) \). Let \( D = [a(x), b(x)] \in J_X(\mathbb{F}_q) \) be a reduced divisor of a large prime order \( n \). The endomorphism \( \phi \) acts as a multiplication map by \( \lambda \) on the subgroup \( < D > \) of \( J_X(\mathbb{F}_q) \) where \( \lambda \) is a root of the characteristic polynomial \( P(t) \) of \( \phi \) modulo \( n \). In what follows, let \( d \) denote the degree of the characteristic polynomial \( P(t) \).

The problem we consider now is that of computing \( kD \) for \( k \) selected randomly from the range \([1, n - 1]\). Suppose that one can write

\[
k = k_0 + k_1 \lambda + \cdots + k_{d-1} \lambda^{d-1} \quad (\text{mod } n),
\]

where \( k_i \approx n^{1/d} \). Then we compute

\[
kD = (k_0 + k_1 \lambda + \cdots + k_{d-1} \lambda^{d-1})D = k_0 D + k_1 \lambda D + \cdots + k_{d-1} \lambda^{d-1} D
\]

(4)

Since \( \phi(D) \) can be easily computed and the bitlengths of components are approximately \( \frac{1}{2} \) that of \( k \), various known methods for simultaneous multiple exponentiation can be applied to (4) to yield faster point multiplication. Thus we might expect to achieve a significant speedup because a great number of point doublings are eliminated at the expense of a few addition on the Jacobian.

4.2. **Decomposition of an integer \( k \).** We now introduce the extended method decomposing an integer \( k \) into a sum of the form given by (3) using a division in the ring \( \mathbb{Z}[\phi] \) generated by an efficiently-computable endomorphism \( \phi \).

Let us consider the map

\[
h : \mathbb{Z}[\phi] \to \mathbb{Z}_n, \quad \sum_{i=0}^{d-1} a_i \phi^i \mapsto \sum_{i=0}^{d-1} a_i \lambda^i \quad (\text{mod } n).
\]

Firstly, we need to find \( \alpha \in \mathbb{Z}[\phi] \) with short components such that \( h(\alpha) = 0 \). Secondly, viewing an integer \( k \) as an element in \( \mathbb{Z}[\phi] \) we divide \( k \) by \( \alpha \) using Algorithm below and write

\[
k = \beta \alpha + \rho
\]
with \( \beta, \rho \in \mathbb{Z}[\phi] \). Since \( h(\alpha) = 0 \) and \( \alpha D = O \) for \( D \in J_X(\mathbb{F}_q) \), we compute
\[
kD = (\beta \alpha + \rho) D = \beta \alpha D + \rho D = \rho D.
\]
Writing \( \rho = \sum_{i=0}^{d-1} k_i \phi^i \in \mathbb{Z}[\phi] \), the preceding equation alternately gives an desired decomposition of an integer \( k \) as in Eqn.(4). This decomposition makes use of the division process in the ring \( \mathbb{Z}[\phi] \), so we now describe an efficient and practical algorithm to compute a remainder \( \rho \) of a given integer \( k \) divided by \( \alpha \).

\[
\hat{\alpha} = -h(\alpha) \in \mathbb{Z}[\phi].
\]

**Algorithm (Divide \( k \) by \( \alpha = \sum_{i=0}^{d-1} a_i \phi^i \))**

**Input:** \( k \approx n \).

**Output:** \( \rho = \sum_{i=0}^{d-1} k_i \phi^i \).

1) Precompute \( \hat{\alpha} = N/\alpha \) in \( \mathbb{Z}[\phi] \) and put \( \hat{\alpha} = \sum_{i=0}^{d-1} b_i \phi^i \).
2) \( x_i = k \cdot b_i \) (for \( i = 0, \ldots, d-1 \)).
3) \( y_i = \lfloor \frac{N}{x_i} \rfloor \) (for \( i = 0, \ldots, d-1 \)).
4) \( \rho = k - \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} a_i y_j \phi^{i+j} \).

**Return:** \( \rho = \sum_{i=0}^{d-1} k_i \phi^i \).

5. Conclusion

We introduced the Gallant-Lambert-Vanstone method and an alternate algorithm for decomposing an integer \( k \) using the theory of \( \mu \)-Euclidean algorithm. The alternate method gives not only a different decomposition of a scalar \( k \) but also produces explicit upper bounds for the components by computing norms in the complex quadratic orders. The reader can find the relative works in [5] and [14]. For hyperelliptic curves, the extended method decomposing an integer \( k \) using a division in the ring \( \mathbb{Z}[\phi] \) generated by an efficiently-computable endomorphism \( \phi \) was presented.

**References**

[2] G. Cornacchia, "Su di un metodo per la risoluzione in numeri interi dell’ equazione \( \sum_{k=0}^{n} C_a x^{n-k} y^k = P^n \), Giornale di Matematiche di Battaglini, 46, (1908), 33-90.


Sejong Cyber University, Seoul, KOREA
E-mail address: youngho@cybersejong.ac.kr