# Construction and Design Estimation of an Elliptic Curve Cryptograph Hardware

Masaaki Shirase (110062)

School of Information Science,
Japan Advanced Institute of Science and Technology

February 14, 2003

**Keywords:** Elliptic curve cryptography, remainder algorithm, cryptograph operation, Wallace tree multiplier.

## 1 Background and Purpose

This paper proposes a design for elliptic curve cryptograph hardware and estimates its performance. Elliptic curve cryptography is one of the public key cryptography as well as RSA. Elliptic curve cryptography has a shorter key length than RSA's key and on the condition of same safety its cryptographic operation by software is faster than RSA. However elliptic curve cryptography is not fast enough.

Cryptographic tecnique is important to maintain secure communication. Public key cryptography is suitable for communication among the general public as Internet, since the cryptosystem does not restrict participant of the system.

## 2 Elliptic curve cryptography

An elliptic curve $E$ is a cubic curve that is represented as $E : y^2 = x^3+ax+b$ (Weierstrass form) or $By^2 = x^3 + Ax^2 + x$ (Montgomery form) and so on. Let $p$ is a prime number and $F_p$ is a finite field has $p$ elements. The set

$E(F_p) = \{(x,\ y) \in F_p\ :\ x$ and $y$ satisfy the elliptic curve equation$\} \cup \{\mathcal{O}\}$,

is called $F_p$ rational point set of $E$, where $\mathcal{O}$ is a special point at infinity. $E(F_p)$ has a group structure which zero element is $\mathcal{O}$. We define the element $P+Q$ on $E(F_p)$ from any two elements $P$ and $Q$ on $E(F_p)$. We write $tP$ for $t$ summands of $P$.

We know the fact that finding an integer $t$ from two elements of $E(F_p)$, $P$ and $tP$, is difficult. Elliptic curve cryptography is based on this difficulty.

In an elliptic curve cryptosystem, $E(F_p)$ and a base point $B$ on $E(F_p)$ are disclosed to the public. The cryptosystem participant selects a integer $s$ at random and calaculates $P_k = sB$. The person releases $P_k$ as publik key and conceals $s$ as secret key.

A ciphertext of massage $m$ to the person who discloses $P_k$ is $(C_1,\ c_2) = (rB,\ x(rP_k \oplus m))$, where $r$ is a random number, $X(rP_k)$ is $x$–coordinate of $rP_x$ and $\oplus$ is bit–wise exclusive or operation. The decryption is calacuration $x(sC_1) \oplus c_2$.

This paper propose two remainder algorithms with a prime $p$ which satisfies $p = 2^n - k,\ k^2 < 2^n$ and $k > 0$. These algorithm are suitable for hardware implementation. We use these algorithms to accelarate elliptic curve cryptography operation.

Calaculating $tP$ from an element of $E(F_p)$, $P$, and an integer $t$ is main operation of elliptic curve cryptography. We use the addition formula for $P+Q$ and the duplication formula for $2P$ to calaculate $tP$ from $P$. We reduce divisions on $F_p$ to multiplications since $x/y = x \cdot y^{p-2}$ on $F_p$.

These formulas consist of four atithmetic operations on $F_p$. For we calaculate addition, substraction and multiplication on $F_p$, we operate them normally and take the remainder of them with $p$.

Weierstrass form of elliptic curves is general and popular form of them. But because of computational complexity, the cryptograph hardware uses Montgomery form rather than Weierstrass form.

## 3 Increase in efficiency of cryptographic operation

Division on $F_p$ is high cost operation since we must operate many multiplication to calaculate $x/y = x \cdot y^{p-2}$. Two divisions $a/b$ and $c/d$ can be calaculated as $a/b = ad(bd)^{-1},\ c/d = bc(bd)^{-1}$, then we calaculate only one division $(bd)^{-1}$. Adopting this method make the computational complexity

of the cryptographic operation reduce about 8%.

We define a new operation $ecc(X_0, X_1, X_2, X_3, i, j)$ as follows

1. $Y_0 = X_0 + X_1,\ Y_1 = X_2 - X_3$,
2. Return $Y_i \cdot Y_j$ and $Y_1$,

for $X_0, X_1, X_2, X_3 \in F_p,\ i, j \in \{0, 1\}$. We only use *ecc* to operate elliptic curve cryptographic operation and *ecc* make the operation control be easy.

## 4 Instruction set

The instsuction form for the cryptograph hardware consists of *opcode*, *operand0*, *operand1*, *operand2*, *next address*. Operand0 and 1 specifies addresses of data used by operation *ecc*. The hardware selects operand0 if a value is 0 and selects operand1 if it is 1. Opecode2 contains an initial value of a counter or a jump address. Next address is an instruction address of next clock.

The instruction length is 80 bits and it consists of 882 words in the case of the cryptographic hardware is pipelined with 14 steps.

## 5 Outline of cryptograph hardware

The Cryptograph hardware consists *ecc* functional unit, an output control unit, two queues, a memory and other control units. Output control unit decides whether *ecc*'s result needs exclusive or operation with other data or not, and operates the operation if it needs. We do not need instructions to control this output control unit. Two queues, Queue1 and Queue2, save input data temporarily to control timing of them. Queue1 sends data to *ecc* functional unit and Queue2 to the output control unit. The memory saves an halfway value of cryptograph operation. Each control unit uses opcodes to control the cryptographic hardware. The ctyptograph hardware needs about 300K gates, a 70K-bit RAM and a 90K-bit ROM.

## 6 Design of cryptograph hardware

The *ecc* functional unit consists of $F_p$ adder, $F_p$ substractor and $F_p$ multiplier. Further $F_p$ adder consists of a 162-bit normal adder and a remainder functional unit. $F_p$ substructor and $F_p$ multiplier also consist of a 162-bit normal substructor or multiplier and a remainder functional unit. These remainder functional unit use algorithms proposed in this paper to calaculate. The auther adopts a carry-lookaheder adder as the 162-bit adder to make the cryptograph hardware has high-speed. the 162-bit substructor's sutructure is like the 162-bit adder's structure.

The auther adopts a Wallace tree multiplier as the 162-bit multiplier. Wallace tree multipliers' CSA part has a little logic depth. Logic depth of 162 bit Wallace tree multiplier's CSA part is 36. Wallace tree multiplier's layout is complicated in general. This paper considers how a Wallace tree multiplier's layout is simple.

Components of the cryptograph hardware except *ecc* functional unit have simple structure.

## 7 Peformance estimation

The auther uses SFL to write the cryptograph hardware and uses PARTHENON to measure its performance and its scale.