УДК 623.647, 681.325.5-181.4

Rodrigue Elias

DESIGN OF AN ELLIPTIC CURVE CRYPTOGRAPHY USING A FINITE FIELD MULTIPLIER IN GF(2^{521})

Криптографія на основі еліптичних кривих забезпечує найбільший захист серед відомих систем з відкритим ключем. Переваги використання маленького ключа робить криптографію на основі еліптичних кривих привабливою, оскільки вона вимагає меншої пам’яті і менших обчислювальних ресурсів. Пропонується помножувач елементів скінченного поля, який є найголовнішим і найбільш споживаючим елементом криптопроцесора, пропонується нова структура помножувача із змінними розрядністю вихідних результатів і кількістю операційних циклів. Кількість вихідних бітів може бути довільно обрана в новій архітектурі залежно від співвідношення «апаратні ресурси – продуктивність». Розглядаються арифметичні пристрої, що використовують помножувачі з різною розрядністю, порівнюється їх робота, апаратні витрати і ефективність реалізації на кристалі. Переваги нової структури ілюструються на прикладі 521-розрядного криптопроцесора, який використовує нормальний базис для представлення елементів поля GF(2^{521}).

ECC (elliptic curve cryptography) offers the highest security per bit among the known public key systems. The benefit of smaller key size makes ECC particularly attractive for embedded applications since its implementation requires less memory and processing power consumption. For a finite field multiplier which is the most important and the most area-consuming unit, a new multiplier structure with scalable output sizes and operation cycles is proposed. The number of output bits can be freely chosen in the new architecture with the performance-area trade-off depending on the application. Arithmetic units using multipliers with various operation bits will be synthesized, and their performance, area, and implementation efficiency will be compared. Through the use of an optimal arithmetic unit, a 521-bit ECC processor based on the normal basis representation will be designed and synthesized in GF(2^{521}).

Introduction. Electronic security has been of considerable interest in recent years because of the increase in electronic transactions – especially money and bank transactions. The developing technology requires a longer key length to satisfy higher levels of security. However, as the key length becomes longer, the operation time increases even more, as does the design complexity and area. Hence, a cryptography algorithm with a short key length and a satisfactory security level is desired.

The operation of ECC is based on the arithmetic of a finite field. The most frequently used finite-field arithmetic operations in ECC are addition and multiplication. In Galois field of 2^m, addition and subtraction of a finite field are bitwise XOR operations, regardless of the representation. The calculation process and the complexity of implementation of the finite-field multiplication depend on the basis representation. The field inversion operation will be performed by iterative multiplications.

The finite-field operation can be performed based on a polynomial-basis representation or a normal-basis representation. The structures of multipliers based on a polynomial-basis representation are easy to extend while the inversion operation is quite complex. Arithmetic units based on a normal-basis representation are adequate because the square and the inverse operation are quite easy to implement.

References observation. Elliptic curve cryptography (ECC) offers the highest security per bit among the known public key cryptosystems [1]. As a proof, the RSA system with a 1024-bit key has a security level similar to that of an ECC system with a 160-bit key [2]. The benefit of smaller key sizes makes ECC particularly attractive for embedded, mobile applications since its implementation requires less memory and processing power [1]. The elliptic curve cryptosystem with a 160-bit modulus is expected to be secure for, at least, the next 5 years [3]. We expect the elliptic curve cryptosystem with a 521-bit modulus to be secure for the next 25 to 30 years.

Arithmetic units based on a normal-basis representation are adequate because the square and the inverse operation are quite easy to implement [4].

Three types of multipliers exist: the parallel input and serial output of Massey and Omura [5], the serial input and parallel output of Feng [6], and the parallel input and parallel output of Wang [7].

Problem. The first and the second multiplier types take m cycles to obtain the result for an m-bit operation. The last one takes only one clock cycle to obtain the result, but it needs more than m-times the area than the others. Therefore, the choice depends on either the cost or the clock cycles. Since a field multiplier is a crucial unit in an ECC processor (ECCP), a new multiplier structure whose performance can be chosen freely is necessary.

Purpose of the work. In this paper, we want to propose a new scalable multiplier structure based on Massey-Omura (M-O) multiplier. For large m, the serial multiplier and the parallel multiplier may not meet either the timing or the area constraints. If the problem is to be solved, a new multiplier structure is needed so that the performance and the area can be adjusted to meet the constraints.

Scalable Multiplier Based on a Normal Basis. Structure of scalable multiplier with n = 16 in $GF(2^{521})$ is shown in Fig. 1.

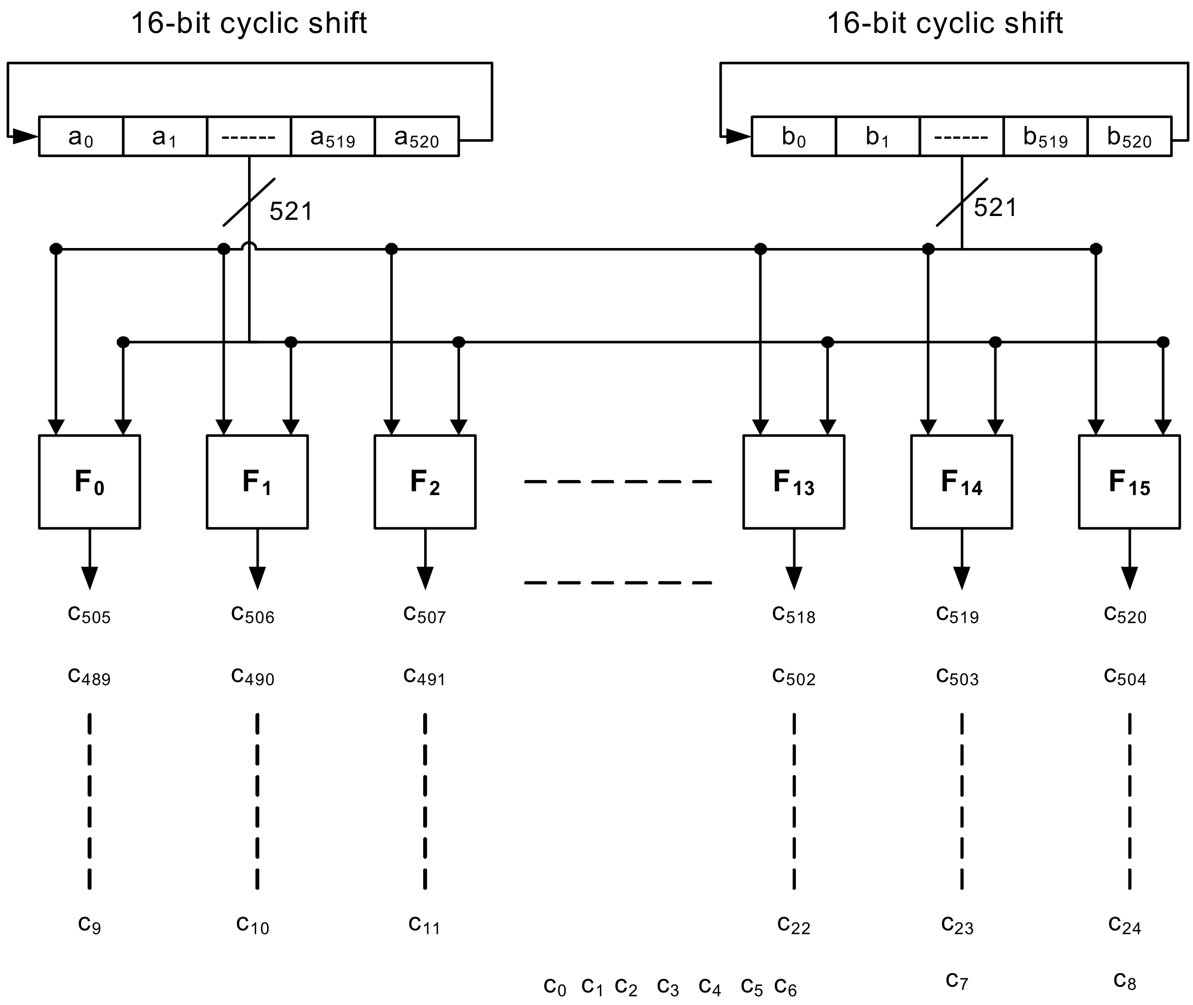
16-bit cyclic shift
16-bit cyclic shift
a_0
a_1
------
a_{519}
a_{520}
b_0
b_1
------
b_{519}
b_{520}
521
521
F_0
F_1
F_2
F_{13}
F_{14}
F_{15}
c_{505} c_{506} c_{507} c_{518} c_{519} c_{520}
c_{489} c_{490} c_{491} c_{502} c_{503} c_{504}
c_9 c_{10} c_{11} c_{22} c_{23} c_{24}
c_0 c_1 c_2 c_3 c_4 c_5 c_6 c_7 c_8

Fig. 1. Structure of Scalable Multiplier with n = 16 in $GF(2^{521})$

A normal basis for GF(2m) is a set of the form:

$$\{\alpha^{2^0},,, \ldots, \alpha^{2^{(m-2)}},\}. \qquad (1)$$

The representation of GF(2m) via the normal basis is carried out by interpreting the bit string as the element:

$$A = a_0\alpha^{2^0} + a_1\alpha^{2^1} + a_2\alpha^{2^2} + \ldots + a_{m-1}\alpha^{2^{m-1}}. \qquad (2)$$

Equation (2) can be rewritten in the bit string as:

$$A = (a_0a_1a_2\ldots a_{m-1}). \qquad (3)$$

In the normal basis representation, A^2 is a cyclic shift of A:

$$A^2 = a_{m-1}\alpha^{2^0} + a_0\alpha^{2^1} + a_1\alpha^{2^2} + \ldots + a_{m-2}\alpha^{2^{m-1}} \qquad (4)$$

Let A and B be two elements of GF(2^m) in a normal basis representation and C be their product.

$$A = (a_0a_1\ldots a_{m-1}),\ B = (b_0b_1\ldots b_{m-1}),\ C = (c_0c_1\ldots c_{m-1}). \qquad (5)$$

The last term c_{m-1} of the product C is a binary function (f-function) of the coefficients of A and B.

$$\left.\begin{array}{l} c_{m-1} = f(a_0, a_1, a_2, \ldots, a_{m-1}; b_0, b_1, b_2, \ldots, b_{m-1}) \\ c_{m-2} = f(a_{m-1}, a_0, a_1, \ldots, a_{m-2}; b_{m-1}, b_0, b_1, \ldots, b_{m-2}) \\ \quad . \quad . \quad . \quad . \\ c_0 = f(a_1, a_2, \ldots, a_{m-1}, a_0; b_1, b_2, \ldots, b_{m-1}, b_0) \end{array}\right\} \qquad (6)$$

Equation (6) can be rewritten by using an integer n, such that: $1 \le n \le m$.

$$(c_{m-1}, \ldots, c_{m-n}) = (f(a_0, \ldots, a_{m-1}; b_0, \ldots, b_{m-1}), , f(a_{m-n+1}, \ldots, a_{m-n}; b_{m-n+1}, \ldots, b_{m-n}))$$

$$(c_{m-n-1}, \ldots, c_{m-2n}) = (f(a_{m-n}, \ldots, a_{m-n-1}; b_{m-n}, \ldots, b_{m-n-1}), \ldots, f(a_{m-2n+1}, \ldots, a_{m-2n}; b_{m-2n+1}, \ldots, b_{m-2n}))$$

$$. \ . \quad . \quad . \quad .$$

$$(c_{m-kn-1}, \ldots, c_0, c_{m-1}, \ldots, c_{m-n+r}) = (f(a_{m-kn}, \ldots, a_{m-1}, a_0, \ldots, a_{m-kn-1}; b_{m-kn}, \ldots, b_{m-1}, b_0, \ldots, b_{m-kn-1}), \ldots, f(a_{m-n+r+1}, \ldots, a_{m-1}, a_0, \ldots, a_{m-n+r}; b_{m-n+r+1}, \ldots, b_{m-1}, b_0, \ldots, b_{m-n+r})) \qquad (7)$$

Design of Arithmetic Unit Based on The Normal Basis in GF(2^{521}). An arithmetic unit performs finite-field operations such as addition, multiplication, squaring and inversion and greatly affects the performance of a crypto-processor. The scalable multiplier can be used for finite-field multiplication. The trinomial that is recommended in IEEE 1363-2000 [8] for a generator polynomial is given by:

$$f(x) = x^{521} + x^{15} + 1 \qquad (8)$$

An arithmetic unit can be constructed by adding an adder and a few multiplexers and by modifying the control unit of an inversion unit. A block diagram of the proposed arithmetic unit is shown in fig. 2.

Elliptic Curve Cryptography Operation and Design. The final operation performed in an ECCP is the elliptic curve point multiplication. Let a and b be the coefficients of a given elliptic curve E, $y^2 + xy = x^3 + ax^2 + b$, and $P_0 = (x_0, y_0)$, $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$ be points of that given elliptic curve. The point addition ($P_2 = P_0 + P_1$) and the point doubling ($P_2 = 2P_1$) are performed as follow:

Point Addition: $z = (y_0 + y_1)/(x_0 + x_1)$;

$x_2 = z^2 + z + x_0 + x_1 + a$; $y_2 = z(x_1 + x_2) + x_2 + y_1$.

Point Doubling: $z = x_1 + (y_1/x_1)$;

$x_2 = z^2 + z + a$; $y_2 = z(x_1 + x_2) + x_2 + y_1$;

The hierarchy of ECC operations is shown in the Fig. 3.

The ECCP is organized with four parts: an input and output processing unit (IOPU), a control unit (CU), a register file (RF), and an arithmetic unit (AU).

The IOPU controls the input and the output. The multiplication factor “k” is transferred to CU upon request for the point multiplication operation, and other data are transferred to the RF, which can receive input data while it transmits a calculated result. The IOPU also generates status signals, “Ready”, “Done”, and “TxReq”. The meaning of the status signals are summarized in the table 1.

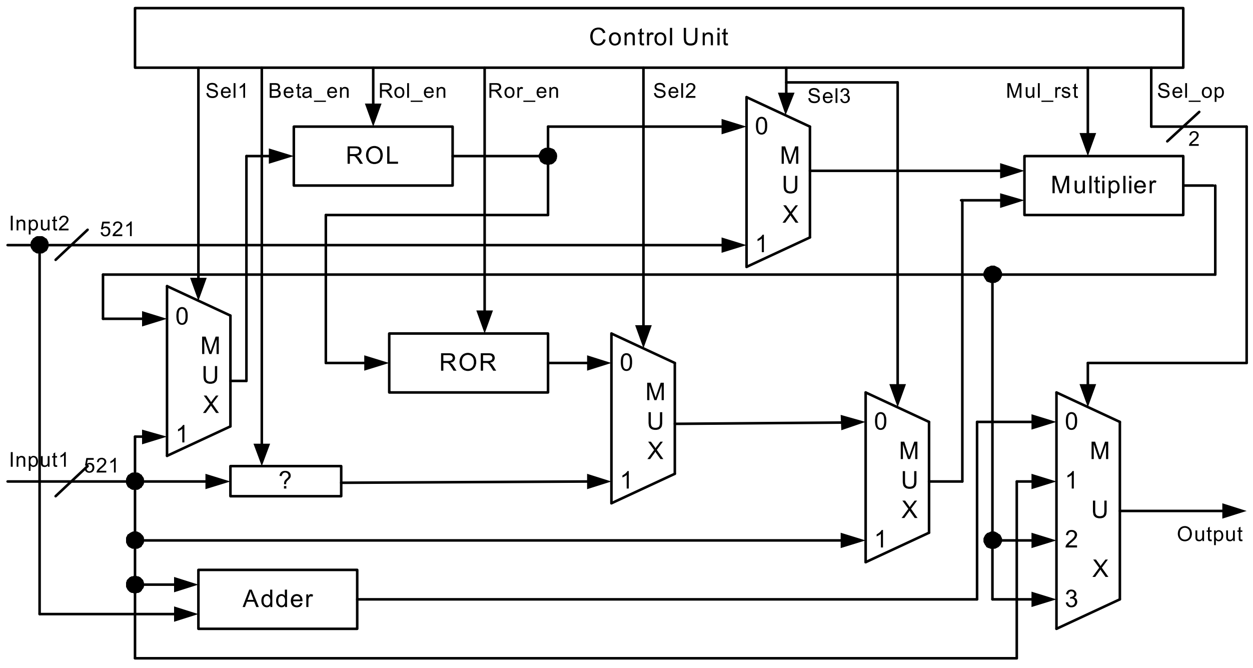

Fig. 2. Block diagram of a 521-bit arithmetic unit in a normal basis representation

Table 1

Summary of status/request signals of the proposed ECCP.

Signal	Meaning
Ready (out)	The processor is ready to receive input data when the signal is active. The input data are stored in the input buffer and wait for calculation.
Done (out)	The processor has a calculated result in the output buffer and is ready to transmit data when the signal is active.
TxReq (in)	The processor transmits data to a host when the signal is active. It is valid only when the "Done" signal is active.

The CU decodes the instructions and controls the other units according to the sequence of finite-field operations. The CU polls the IOPU if the data are ready and the output buffer is empty. It controls the data transfer between the IOPU and the RF and between the RF and the AU. It also issues operation commands to the AU to perform field operations. The instructions are summarized in table 2.

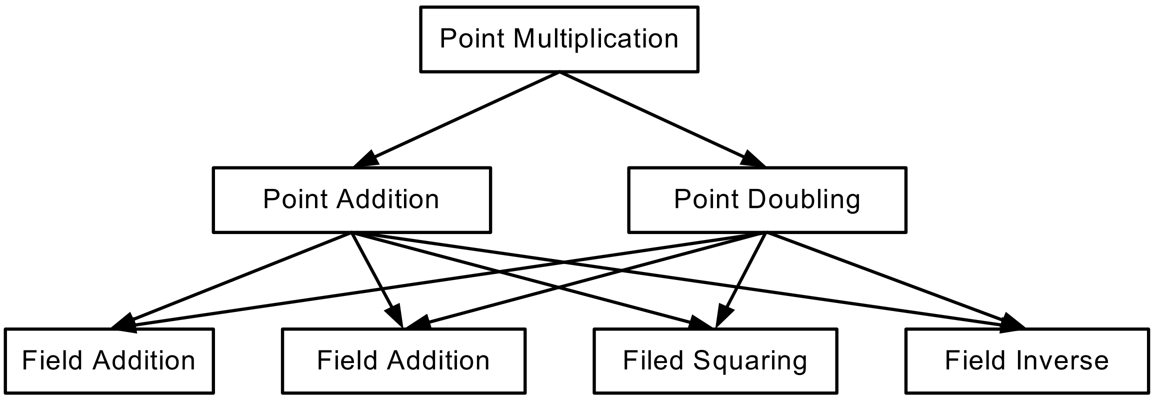

Fig. 3. Hierarchical diagram of ECC operations

Table 2

List of the instructions performed by the ECCP

Instruction	Action
FADD	Field Addition
FMUL	Field Multiplication
FINV	Field Inversion
FSQR	Field Squaring
ECA	Point Addition
ECD	Point Doubling
ECM	Point Multiplication

The RF consists of eight 521-bit registers which store internal data which are used by the AU. The AU performs finite-field operation such as addition, multiplication, squaring, and inversion. A block diagram of the proposed ECCP is shown in Fig 4.

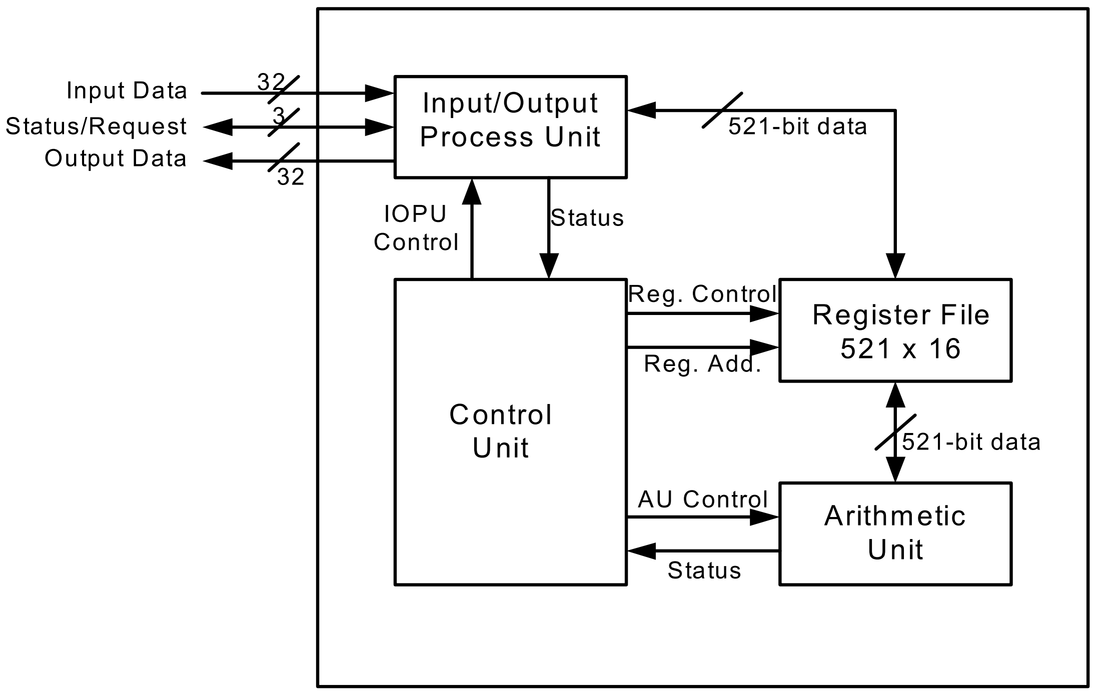

Fig. 4:Block diagram of the proposed ECCP structure

Conclusion. In this paper a scalable finite-field multiplier structure for the ECC operation based on the normal-basis representation over GF(2^{521}) is proposed. The number of output bits of the multiplier can be freely chosen in the new architecture with the performance area trade-off depending on the application.

1. Leung K.H., Ma K.W., Wong W.K. and Leong P.H.W. 2000 IEEE Symposium on Field Programmable Custom Computing Machines. 2. Standards for Effcient Cryptography Group. SEC 1:

Elliptic Curve Cryptography, September 2000. Version 1.0. Available at www.secg.org. *3. Daniel R.L. Brown. Certicom Proposal to Revise SEC 1: Elliptic Curve Cryptography, Version 1.0 January 14, 2005. Available at* www.secg.org. *4. Sutikno S., Effendi R. and Surya A. IEEE Asia Pacific Conference on Circuit and Systems 1998, p.647. 5. Reyhani-Masoleh A., Hasan M.A. New Construction of Massey-Omura Parallel Multiplier over GF(2^m). – Vol. 51, Issue 5 (May 2002), ISSN: 0018-9340. 6. Song L., Parhi K.K. Efficient Finite Field Serial / Parallel Multiplication. – 1996. 7. Stallings W. Cryptography and Network Security. Prentice Hall, 2003. 8. IEEE Std 1363-2000 IEEE Standard Specifications for Public-Key Cryptography Sponsor Microprocessor and Microcomputer Standards Committee of the IEEE Computer Society. Approved 30 January 2000.*