High Performance Reconfigurable Elliptic Curve Cipher Processor Implementation

Jian Zhang*, Liting Niu

School of Physical Science and Technology, Wuhan University, Wuhan 430072, China

*Corresponding author’s e-mail: 2018302020165@whu.edu.cn

Abstract. Elliptic Curve Encryption (ECC) has been widely used in the field of digital signatures in communication security. ECC standards and the diversification of application scenarios put forward higher requirements for the flexibility of ECC processors. Therefore, it is necessary to design a flexible and reconfigurable processor to adapt to changing standards. The cryptographic processor chip designed in this paper supports the choice of prime and binary fields, supports the maximum key length of 576 bits, uses microcode programming to achieve reconfigurable function, and significantly improves the flexibility of the dedicated cryptographic processor. At the same time, the speed of modular multiplication and modular division can be greatly improved under the condition of keeping the low level of hardware resources through a carefully designed modular unit of operation. After using FPGA for hardware implementation, it is configured into a 256-bit key length. The highest clock frequency of this design can reach 55.7MHz, occupying 12425LUTS. Compared with a similar design, the performance is also greatly improved. After MALU module optimization design, modular multiplication module division also has significant advantages in computing time consumption.

1. Introduction

With the vigorous development of the Internet of Things (IoT), especially the wide application of radio frequency identification (RFID) and near field communication (NFC) technology in the field of payment, people have higher and higher requirements for communication security. Public key cryptography, namely asymmetric encryption, is an effective encryption method to ensure communication security. RSA and Elliptic Curve Encryption (ECC) are two widely used public-key encryption algorithms. Under the same security level, the key length of the elliptic curve encryption algorithm is much lower than that of RSA, and it has the advantages of smaller hardware resources, lower power consumption, and faster computation. Therefore, ECC is more suitable for application in the field of the Internet of Things with strict requirements on resources [1].

The main operation of the elliptic curve encryption algorithm is elliptic curve scalar multiplication (ECSM), that is, a point on an elliptic curve is multiplied by a scalar to get the target...
point. The hardware implementation of elliptic curve encryption algorithm can be carried out in binary domain or prime domain, and ECSM can be calculated by the binary algorithm, Montgomery algorithm or non-adjacent form scalar multiplication algorithm (NAF). ECC has various specifications and standards, such as FIPS 186-2. IEEE P1363 and ANSI X9.62. And so on. At present, there have been a lot of hardware implementation schemes to calculate the elliptic curve scalar multiplication, some solutions to accelerate the operation speed of scalar multiplication [2], the other solution is to solve the problem of hardware resource consumption [3], but these solutions are limited in the fixed, fixed length of the key, fixed standard of reconfigurable design, It cannot meet the high flexibility requirements of the current development of the Internet of Things for the Elliptic Curve Crypto processor.

In this paper, a high-performance and high flexibility hardware implementation scheme of the ECC processor is proposed. It can select binary fields over prime fields, and it supports key lengths up to 576 bits. In addition, it can support a variety of ECSM algorithms through the software-hardware cooperation scheme, and then improve the performance of hardware computing by carefully designing the modular multiplication unit Malu. Power Analysis Attack is one of the threats to cryptographic algorithm hardware [4], which analyzes and obtains key information through statistical Power curve. The algorithm implemented in this scheme can resist simple power analysis attacks (SPA) at the same time, and consider performance, flexibility, and security.

2. Basic knowledge

2.1. Elliptic curve cryptography

Prime fields \(GF(p)\) is defined as (1).

\[ y^2 = x^3 + ax + b \]  

Among them \(x, y, a, b \in GF(p)\) and satisfy \(4a^2 + 27b^2 \neq 0\).

Binary domain \(GF(2^m)\). The elliptic curve of is defined as (2).

\[ y^2 + xy = x^3 + ax^2 + b \]  

Which meet the \(b \neq 0\).

The ECSM calculation process includes point addition and multiple point operation on the elliptic curve. Point addition and point doubling operations in affine coordinates are shown in Table 1 and Table 2 respectively.

| Table 1. Point addition operation in affine coordinates |
|---|---|
| **domain** | **Point to add** \(P_3 = P_1 + P_2\) |
| \(GF(p)\) | \(\lambda = (y_2 - y_1) / (x_2 - x_1)\) |
| | \(x_3 = \lambda^2 - x_1 - x_2\) |
| | \(y_3 = \lambda(x_1 - x_3) - y_1\) |
| \(GF(2^m)\) | \(\lambda = (y_2 + y_1) / (x_2 + x_1)\) |
| | \(x_3 = \lambda^2 + \lambda + x_1 + x_2 + a\) |
| | \(y_3 = \lambda(x_1 + x_3) + x_3 + y_1\) |
Table 2. Multiple point operation in affine coordinates

| domain | Times the point($P_1 = 2 \cdot P_2$) |
|--------|---------------------------------------|
| $GF(p)$ | $\lambda = (3x_1^2 + a) / 2y_1$       |
|     | $x_3 = \lambda^2 - 2x_1$              |
|     | $y_3 = \lambda(x_1 - x_3) - y_1$      |
| $GF(2^m)$ | $\lambda = x_1 + y_1 / x_1$          |
|     | $x_3 = \lambda^2 + \lambda + a$      |
|     | $y_3 = x_3^2 + \lambda x_3 + x_3$    |

There are many schemes to perform ECSM computation, each with its advantages and disadvantages. Algorithm 1 is a binary algorithm, which has the advantage of easy design and simple structure but is vulnerable to simple power analysis attacks. Algorithm 2 is the Montgomery gradient algorithm, which can resist SPA attacks in one iteration after another [5], with relatively high security.

Algorithm 1 Binary algorithm applicable to ECSM:

Input: $k, P, \ P_1$ Among them $k = (k_{m-1}, k_{m-2}, \ldots, k_0), k_i \in \{0,1\}$

Output: $Q = k \cdot P$

Step1: $Q \leftarrow 0$

Step2: for $i$ from $m - 1$ to 0 do
   $Q \leftarrow 2Q$
   if $k_i = 1$ then $Q \leftarrow Q + P$

Step3: return $Q$

Algorithm 2 Montgomery gradient algorithm applicable to ECSM:

Input: $k, P, \ P_1$ Among them $k = (k_{m-1}, k_{m-2}, \ldots, k_0), k_i \in \{0,1\}$

Output: $Q = k \cdot P$

Step1: $P_1 \leftarrow P, \ P_2 \leftarrow 2P$

Step2: for $i$ from $m - 2$ to 0 do
   if $k_i = 1$ then $P_1 \leftarrow P_1 + P_2, \ P_2 \leftarrow 2P_2$
   else $P_2 \leftarrow P_1 + P_2, \ P_1 \leftarrow 2P_1$

Step3: return $Q = P_1$

2.2. Carry Retaining Adder (CSA)

In hardware computing schemes, a carry reservation adder (CSA) is usually used to compute the sum of a tree or multiple numbers, which outputs the XOR result and the carry result. Given three m-bit widths, the output calculated by CSA is:

$$\text{sum} = a_i \oplus b_i \oplus c_i$$

$$\text{carry} = (a_i \& b_i)| (a_i \& c_i)| (b_i \& c_i)$$
The &, | and ^ respectively and, or and xor. a\(^m+1\) bit operations can be performed using a carry propagation adder (CPA), which, when used in conjunction with CSA, reduces the computation latency of the data path.

3. Modular Multiplication Unit (MALU)

3.1. Modular multiplication operation

In prime fields \(GF(p)\) and the binary field \(GF(2^m)\), the modular multiplication operation is a very key link. Li Yanmei proposed a scalar multiplication algorithm of the elliptic curve based on multi-basis representation [6,7], which proved that multi-basis operation could significantly improve the operation speed of modular multiplication at the cost of increasing the hardware realization area. In this design, the tradeoff is to choose the optimal base 4-mode multiplication to greatly improve the speed of the modular multiplication and ensure the realizability of the hardware. Given \(m\) bits \(A = \sum_{i=0}^{m-1} a_i 2^i\) and \(B = \sum_{i=0}^{m-1} b_i 2^i\). In prime fields \(GF(p)\) is multiplied by (5):

\[
C = a \cdot b \pmod{p} = b_0 \cdot A + b_1 \cdot A \cdot 2 + \cdots + b_{m-1} \cdot (A \cdot 2^{m-2}) \cdot 2 \pmod{p}
\]

From Equation (5), the modular multiplication operation of base 4 realized by algorithm 3 requires \(m^2\) clock cycles. In contrast, base 4 operation can significantly reduce time consumption.

Algorithm 4 basis 4 cross modulus multiplication operation implementation:

Input:
Among them \(a_i, b_i, p_i \in \{0,1\}\)

Output:
Among them \(c_i \in \{0,1\}\)

Step1: \(C \leftarrow 0\)

Step2: for \(i\) from 0 to \(\left\lceil \frac{m}{2} \right\rceil - 1\) do

\[
C = C + b_i \cdot A + b_i \cdot 2A \pmod{p}
\]

\[
A = 4 \cdot A \pmod{p}, B = B/4
\]

Step3: return \(C\)

3.2. Modular Inverse and Modular Division

In the elliptic curve encryption algorithm, the first modulus division algorithm can be regarded as the joint action of the first modulus inverse algorithm and the modulus multiplication algorithm [8]. An efficient modular division operation can be extended from the Euclidean algorithm (GCD) to the symbol bit base 4 GCD algorithm [9]. Algorithm 5 gives a pair of prime fields \(GF(p)\) binary domain Binary domain \(GF(2^m)\). It’s similar.

Algorithm 5 base 4 prime field GCD module division algorithm implementation:

Input:
Among them \(X, Y \in [1, p - 1]\)

Output:
\(Z = A/B \pmod{p}\)

Step1: \(A \leftarrow P, B \leftarrow Y, U \leftarrow X, V \leftarrow 0, \rho \leftarrow n, \delta \leftarrow 0\)

Step2: while \(\rho > 0\) do

if \(B \pmod{4} = 0\) then

\(B \leftarrow \frac{B}{4}, U \leftarrow \frac{U}{4} \pmod{p}\)

else

\(A \leftarrow A - V, B \leftarrow B - \rho \cdot U, U \leftarrow -U, V \leftarrow -V, \rho \leftarrow \rho - 1\)

end if
if $\delta \leq 0$ then $p \leftarrow p - 2$
else if $\delta = 1$ then $p \leftarrow p - 1$
else $\delta \leftarrow \delta - 2$

else
$A_{\text{next}} \leftarrow B$, $V_{\text{next}} \leftarrow U$
if $(A + B) \mod 4 = 0$ then
$B \leftarrow \frac{A + B}{4}$, $U \leftarrow \frac{U + V}{4} \mod p$
else
$B \leftarrow \frac{A - B}{4}$, $U \leftarrow \frac{U - V}{4} \mod p$
if $\delta < 0$ then $A \leftarrow A_{\text{next}}$, $V \leftarrow V_{\text{next}}$
else if $\delta = 0$ then $p \leftarrow p - 1$

else $\delta \leftarrow \delta - 1$
end while

Step3: if $Y < 0$ then $V \leftarrow V + p$
if $A = 1$ then $Z \leftarrow V$ else $Z \leftarrow p - V$

Step4: return $Z$

4. ECC processor architecture design

4.1. Overall architecture and module functions
The ECC special processor consists of ECC Control Unit (ECC), Modular Alu (Modular Alu), Modular ROM, Register List, and AMBA Bus Interface) and other modules. The processor starts to work, the load is stored in ROM in advance of the elliptic curve parameters and operation instruction, instruction based on the more fundamental point multiplication, instead, die inverse (ADD/SUB, the MUL, DIV) arithmetic operations, such as when the processor functions change only changes in programmable command is stored in ROM, greatly improve the flexibility of the processor. Of course, the difficulty with this approach is that the implementation of the algorithm is more complex to program.

The Register List is 128x 64-bits, limited by a 4-bit address width, and can hold up to 14 temporary variables. After the LOAD instruction is issued, the control unit decodes and loads the elliptic curve parameters in the ROM and stores them in the register. The area cost of the chip can be effectively reduced by setting the ellipse parameters and loading the pre-processed parameters into registers.

In addition to storing parameters, ROM also divides a part of the space of storing instructions for the core control unit to receive instructions.

The Modular operation unit executes operation instructions of ADD/SUB, MUL, and DIV establishes a 64-bit data path between the register list and is the core operating unit of the coprocessor. The overall architecture is shown in Figure 1.
In our design, the AMBA bus is adopted to mount the ECC processor to the bus structure [10][11], to realize the collaborative work of multiple systems. Adding module AMBA Bus allows ECC processors to play a greater role as slave devices to connect to other main processors [12-15].

4.2. Instruction architecture design

The core Control Unit of the processor mainly completes finger fetching, decoding, and execution from ROM. The state transition diagram is shown in Figure 2, and the 16-bit RISC instruction description is shown in Table 3.

![Figure 2. Schematic diagram of controller state transition](image)

| instruction | Functional description |
|-------------|------------------------|
| INIT        | Select prime field, binary field |
| LOAD        | Initializes the load of elliptic curve parameters |
| JCMP        | Conditional jump instruction |
| JUMP        | Unconditional jump instruction |
| STOP        | Terminates the current operation and returns the end flag |
| ADD         | Mode and operation |
| SUB         | Die cut operation |
| MUL         | Die by computing |
| DIV         | Die divide |

5. Hardware verification and discussion of ECC processor
5.1. FPGA hardware verification of the processor

The FPGA hardware verification of the ECC processor is divided into two steps: first, logic simulation is completed with ModelSim, and appropriate excitation is added to verify the logic correctness of each module through the simulation waveform. After the verification of the first step, the surface module design meets our logical requirements. Next, the Vivado tool is used for integrated wiring, and the generated bitstream file is downloaded to the FPGA. We can use the ILA core provided by Xilinx to observe and debug the correctness of each signal online and verify whether the processor correctly executes the microcode written in advance in ROM and returns the correct results. Table 4 shows the combined results of the FPGA hardware validation[21-24].

| The name                  | ECC processor                  |
|---------------------------|--------------------------------|
| The FPGA model            | VIRTEX-II                      |
| LUTs                      | 12425/51840 (24 %)             |
| Pin number                | 88 / 493 (17 %)                |
| fmax                      | 55.7MHz                        |
| Die by time-consuming     | 2.30 us                        |
| Die except the time-consuming | 6.20 us                      |
| The key length            | A 256-bit                     |

5.2. Performance analysis and comparison

The computing speed of the ECC processor is mainly determined by the efficiency of the MALU module and the delay of the data path. To make a convincing comparison with some similar works, we set our design to the same key length and verify it on the same FPGA platform. The comparison results are shown in Table 5.

| area/Slices | Fmax/MHz | Die by time-consuming/us | Die except the time-consuming/us |
|-------------|----------|--------------------------|----------------------------------|
| [13]        | 5477     | 14                       | 18.28                            | 43.89                            |
| [14]        | 5379     | 34                       | 7.53                             | 14.6                             |
| [15]        | 9213     | 37                       | 3.46                             | 4.98                             |
| [16]        | 4836     | 110.4                    | 1.25                             | 637                              |
| This design | 5863     | 55.7                     | 2.30                             | 6.20                             |

By comparison, we can see that under the same key length, our processor runs at 55.7MHz frequency and occupies 12425LUTS, which has a high processing efficiency. In addition, other similar works are fixed bit width operations in each field, which cannot be programmed to reconstruct, greatly limiting the scope of application. Our design not only does not use too many additional hardware resources but also features higher performance computing efficiency and
more flexible and reconfigurable configuration, which can greatly broaden our application scenarios [23]. In fact, it’s hard to say which design is the best. These similar works are implemented using different technologies and are designed specifically for specific applications. Our design is aimed at non-specific application scenarios, and flexible implementations such as multiple algorithm standards and configurable key length can serve the requirements of cryptographic processors for non-single scenarios.

6. Conclusion
To ensure communication security, the design of an elliptic curve cryptographic processor for asymmetric encryption is very important. In this paper, a high-performance reconfigurable ECC hardware implementation scheme is proposed, which meets the flexible and configurable requirements of current ECC multi-standard and diverse algorithms. Relying on the function of microcode programming configuration, the processor can select prime number field and binary field for calculation, the key length can be configured, and the algorithm structure can be reconfigured. Compared with similar designs, the design can deal with the same 256-bit wide elliptic curve modular operation in the prime number field. While maintaining a low level of hardware resource consumption, the design can significantly improve the computational efficiency and reduce the operation time of module multiplication and module division through the well-designed modular operation unit MALU. Such systems can be widely used on the Internet of Things, where flexibility and performance are highly required.

References
[1] Li, C. H. (2010) Computer and Modernization, 2010, 1(8): 191-193. ZHANG Wei. Research on Application of ECC Signcryption Algorithm in RFID [J]. Computer and Modernization, 1(8): 191-193.
[2] Liu, Z. H. (2014) Research and improvement of scalar multiplication algorithm for elliptic curve encryption [D]. Beijing University of Chemical Technology.
[3] Wang, J. M. (2019) Design and Hardware Simulation of Low-cost RFID Authentication Protocol Based on ECC [D]. Xidian University.
[4] Tang, Z., Lin, L. (2016) A Secure and Efficient ECC Anti-Power Attack Scheme [J]. Measurement and Control Technology, 35(09):149-152.
[5] Chen, L., Ge, W. C. (2014) Edge Channel Attack for Elliptic Curve Encryption Algorithm [J]. Communications Technology,2014,47(09):1062-1065.
[6] Li, Y. M., Yin, X. C., Shao, M. L. (2019) Multi-scalar Multiplication Algorithm for Sliding Window Elliptic Curves Based on Multi-base Representation [J]. Computer and Modernization, (1):11-16.
[7] Guo, X. J., Liu, Y. M., Li, N. (2009) FPGA design of reconfigurable dot product [J]. Information Security and Communication Security, 4(06):86-87+90.
[8] Qin, F., Dai, Z. B. (2008) Computer Engineering and Applications, 4(23):117-119+129.
[9] Ding, Y., Gui, F. (2008) High Speed Band Mode Division over GF(2^m) Based on GCD Algorithm [J]. Journal of Communications, 4(10):199-204.
[10] Chai, S. J. (2020) Research on Encryption IP Core of AHB-Lite Bus Transmission Data Based on FPGA [D]. Lanzhou Jiaotong University.
[11] Wang, Y., Lin, T., Yu, N. M. (2019) Design of AHB Bus Matrix Based on AMBA [J]. Microelectronics & Computer, 36(02):73-77.
[12] Bao, R. H. (2019) Design of Several IP Modules in SoC Design Platform [D]. Hangzhou Dianzi University.
[13] Huang, R. M. (2017) Information Security System Based on Elliptic Curve Encryption Algorithm [J]. Journal of Neijiang Normal University, 32(02):72-76.
[14] Pan, J. Y. (2016) Application of Elliptic Curve Encryption Algorithm in Hybrid Cryptographic System [J]. Information and Communications, 4(08):27-28.
[15] Qiu, Y. Q. (2020) Design of High-Performance Processor Interconnection Verification System Based on FPGA [D]. Xidian University.
[16] Zhu, Y., Ge, H. B. (2020) High Performance Implementation of Elliptic Curve Cryptography ECC on Binary Field Based on FPGA [J]. Computer Science, 47(08):127-131.
[17] Zhan, T. M., Wu, W. T. (2017) Implementation of High-Performance Elliptic Curve Cryptography Algorithm on Zynq Platform [J]. Communications Technology, 50(05):1074-1078.
[18] Cheng, Y. S., Cheng Y. S. (2018) Prototype Verification of IP Subsystem of SoC Chip Based on FPGA [D]. Chongqing University of Posts and Telecommunications,2018. Verification of IP Subsystem Prototype of SoC Chip Based on FPGA [D]. Chongqing University of Posts and Telecommunications.
[19] Daly, A., Marnane, W., Kerins, T., Popovici, E. (2004) An FPGA implementation of a GF(p) ALU for encryption processors[J]// Microprocessors and Microsystems, 28(5): 253-260.
[20] Ghosh, S., Mukhopadhyay, D., Roychowdhury, D. (2011) Petrel: Power and Timing Attack Resistant Elliptic Curve Scalar Multiplier Based on Programmable GF(p) Arithmetic Unit[J]// IEEE Transactions on Circuits and Systems I, pp. 1798 - 1812.
[21] Lee, J. W., Chung, S. C., Chang, H. C., Lee, C. Y. (2014) Efficient Power-Analysis-Resistant Elliptic Curve Cryptographic Processor Using Heterogeneous Dual-Processing-Element Architecture[J]// IEEE Transactions on Very Large-Scale Integration (VLSI) Systems, pp. 49 - 61.
[22] Sakivam,a K., Preneel, B., Verbauwhede, I. (2006) A fast dual-field modular arithmetic logic unit and its hardware implementation[J]// IEEE International Symposium on Circuits and Systems, pp. 21-24.
[23] Yuan, H. (2019) Research on Key Technologies of Dynamic Reconfigurable Cryptochip [D]. Tsinghua University.
[24] Yang, C. H. (2021) Optimization Method of Network Information Private Key Transform Based on Improved ECC Algorithm [J]. Network Security Technology and Application, (07):34-35.