An On-chip Data Storage with ElGamal Elliptic Curve Cryptosystems

S. Choomchuay, S. Pongyupinpanich and K.Hadkhuntod
Department of Electronics, Faculty of Engineering
Microelectronic Devices R&D Lab, Research Centre for Communications and Information Technology
King Mongkut's Institute of Technology Ladkrabang (KMITL)
Chalongkrung Road, Ladkrabang, Bangkok 10520, Thailand.
Tel: (66) 2 326 4222 Ext. 114, Fax: (66) 2 739 2398, E-mail: kchsomsa@kmitl.ac.th

Abstract

An On-chip Data Storage with cryptosystem proposed in this paper was designed for modern smart card system and "on the fly data encryption/decryption". To ensure the security, the ElGamal Elliptic curve cryptosystems with the key size of 176 bits was embeded. The design can be configured to be either the encrypter or the decryptor with simple control signals. To minimize the hardware cost, size and complexity, the slight modified fix-coefficient bit-serial finite field multiplier has been evoked. The FPGA-based prototype implementation had proved the concept idea and mathematics behind.

1. Introduction

One most important application of a portable on-chip data storage and corresponding processor is smart card. A class of contact smart card can incorporate a tiny microprocessor and a set of memory. Data stored are divided into zones according to their accessibility, i.e., open zone, secret zone and user record. Firstly developed in 1950 by Diners Club, the nowadays smart card is much more smart. It can accommodate more information and covers most area of applications; ticket, phone card, credit card, medical card and many other type of ID cards. Despite the operation of the smart cards that quite vary, the common issue is their data security. For such a guarantee, data stored as well as data transmitted or received are manipulated to be in the form that are not easy to be guessed or understood. Generally those texts are ciphered or encrypted by using a special class of mathematics. There comes the cryptosystems.

The history of cryptography perhaps began when DES (Data Encryption Standard) was first introduced. Such a research began in 1970 and brought into practice in 1977. Since DES is a symmetric key system (shared key, symmetric key), its security level is quite limited. As an alternative approach, the concept of public key cryptosystems (asymmetric key system) was introduced in 1976 by Diffie Hellman. The system is based on Discrete Logarithm Problem (DLP). In 1978, Rivest Shamir and Adleman introduced the RSA public key cryptosystem based on a large integer factorization. Subsequently, a more complicate one which still based on DLP was proposed by ElGamal in 1985. Yielding better security level with shorter key, the ElGamal cryptosystem based on Elliptic curve (ECC) is more intricate compared to its firstly-developed version.

In this paper we propose an implementation of data encryption/decryption scheme that could be either applied to smart card system or transmitting and receiving data with ciphering and deciphering capability. Mathematical background of cryptosystem, in particular ElGamal Elliptic Curve Cryptosystems as well as the corresponding example are given in the following section 2. A discussion of hardware for implementing such an algorithm is also discussed. In section 3, the architecture that emphasizes the working in a pair of the encrypter and the decrypter is elaborated. Several mathematic operations can be computed off line. As a result, this piece of hardware just have to compute only a few simple multiplications. Bit serial design has further reduced the multiplier size. The architecture given in section 2 is then implemented with FPGA as detailed in section 3 before the conclusion of our works.

2. Cryptosystems

2.1 ElGamal Public Key Cryptography

A standard ElGamal public-Key cryptography is based on the large integer prime field Z_P of which α is a root, for instant $P = 2357$ and $\alpha = 2$. A private key a, $a \in Z_P$, can be chosen arbitrarily. The public key R is defined such that $R(P, \alpha, \alpha^a)$. The sender chooses k as an arbitrary to compute $\gamma = \alpha^k$. A message $m \in Z_P$ is encrypted as $\delta = m\left(\alpha^a\right)^k$. Both γ and δ are then sent through the communication channel. Once the receiver has received the messages, he computes $\beta = \gamma^{(P-1-a)}$. Message can be decrypted as $m = \beta \cdot \delta$. A point to be noted here, $\alpha^{kP} = \alpha^k$.

2.2 Elliptic Curve Mathematics

Elliptic curves cryptography is preferably implemented using non-supersingular because of its better security. The underline field of characteristic 2, GF(2^m) is then the set of solution to the equation

$$y^2 + xy = x^3 + Ax^2 + B \qquad (1)$$

where $A, B \in GF(2^m)$ and $B \neq 0$. Number of points in an elliptic curve, order of the curve, is denoted by #E.

Let $P=(x_1,y_1)$ be a member of those points, then $-P=(x_1,y_1+x_1)$. For all $Q=(x_2,y_2)$ with $P,Q\neq \mathcal{O}$ (where $\mathcal{O}$ denotes the point at infinity) and $Q\neq -P$, the resulted $P+Q=(x_3,y_3)$ are as follows:

$$x_3=\begin{cases}\left(\dfrac{y_1+y_2}{x_1+x_2}\right)^2+\dfrac{y_1+y_2}{x_1+x_2}+(x_1+x_2)+A, & P\neq Q\\ x_1^2+\dfrac{B}{x_1^2}, & P=Q\end{cases}$$

and

$$y_3=\begin{cases}\left(\dfrac{y_1+y_2}{x_1+x_2}\right)(x_1+x_2)+x_3+y_1, & P\neq Q\\ x_1^2+\left(x_1+\dfrac{y_1}{x_1}\right)x_3+x_3, & P=Q\end{cases}$$

2.3 Elliptic Curve Public-Key Encryption

Let *P* be a fix point and be a member of #E. The sender A can choose randomly an integer *a* as a private key where $a\in GF(2^m)$. With this private key *a*, the public key *R* can be computed from *R* = *aP*. With another randomly chosen *k*, $k\in GF(2^m)$, the fix point *Q* also can be computed from *Q* = *kP*. The sender A encrypts his massage *M* by making use of $kR=akP=(x',y')$ as $m=MkR$, preferably, $m_1=M_1x'$ and $m_2=M_2y'$, where $M=M_1+M_2$. The encrypted words then sent along with *Q*.

Once the encrypted words received by the receiver B, with A's private key, he computes $aQ=akP=(x',y')$. Subsequently $m_1(x')^{-1}=M_1x'(x')^{-1}$ and $m_2(y')^{-1}=M_2y'(y')^{-1}$ are computed and the original messages M_1 and M_2 are obtained respectively. It should be noted that, firstly, the private key *a*, must be known to both the sender and the receiver in prior. Secondly, in obtaining the original message, two inversion multiplications are required.

2.4 Design Example

The Elliptic Cryptosystems (ECC) with the key size of 155 to 201 bits can yield the similar level security as the Discrete Logarithm Cryptosystems (DLC) with the key size of 512 to 1024 bits. Thus, ECC is superior to DLC in terms of block length and storage requirement. Our implementation is based on the key size of 176 bits where GF(2^{176}) is the underline field.

1) To minimize the hardware complexity (wiring complexity according to the feedback and number of XOR gates), the irreducible polynomial $I(x)=x^{176}+x^{11}+x^3+x^2+1$ was chosen.

2) *A* and *B*, $A,B\in GF(2^{176})$ are chosen as arbitraries:

A = 6f4e a37c cb7c 6443 44de 2368 dba3 c524 bd4a e585 bb6c
B = 8a38 e3e4 1595 f58f a35e c0a2 dd61 d80a 8984 bc08 4f91

3) With such *A* and *B* in 2), there could be a possible solution to the elliptic curves equation;

$$y^2+xy=x^3+Ax^2+B$$

We yield a fix point *P(x,y)* of the curve

x = 9e15 2cda 8c22 e6e8 23d7 674e 72e5 a195 44b2 35e7 c3d1
y = 8bf3 e7ed dbe0 3681 aa7b c488 1815 dba6 c633 569a 43b3.

4) Regardless of the computing complexity at this step, A private key *a* is freely chosen and similar as *k*.

a = 75 Private key
k = 100 Any field element

5) The encrypter has to compute

$kR=akP=(x',y')$, *R* denotes the public key

and yields

x' = f874 8697 a7d1 73d5 da23 9933 615f 89d1 aef7 93af c1e8
y' = 34ab ecbb a95f 327f ffb2 50f2 c7a8 65d5 dd5a 471e 8422

6) The encrypted messages are computed as

$m_1=M_1x'$ and $m_2=M_2y'$ where M_1 and M_2 are consecutive blocks of which the data length is *m* bits (176 bits).

7) Encrypted messages are fed and stored in the array of Flash ROMs which can be on chip or can be a separate module.

Operation 6) and 7) can repeat many times until all the data is encrypted and stored (or sent). To read out the encrypted messages, Operations are as follows:

1) *Q = kP* is pre-computed and *aQ = (x',y')*

2) Step 1) could be omitted since the encrypter and the decrypter are in the same module. From 5) above $(x')^{-1}$ and $(y')^{-1}$ can be computed.

$(x')^{-1}$ = 78c4 f832 64fe 4305 7a4c 2f3e 9cd2 88bd 8d1c e7ee
$(y')^{-1}$ = f64c b612 cb58 5156 292d f8d1 0e60 6c8a a4c9 677a

3) Subsequently;

$M_1=m_1(x')^{-1}$ and $M_2=m_2(y')^{-1}$

Operation in 3) is to be repeated until all the data are decrypted.

2.5 Galois Field Multiplication

Let GF(2^m) be the extension field of GF(2) and I(x) be an irreducible polynomial of degree *m*, the multiplication of 2

field elements, $W = UV$, where $U(x) = \sum_{i=0}^{m-1} u_i x^i$, $V(x) = \sum_{i=0}^{m-1} v_i x^i$ and $W(x) = \sum_{i=0}^{m-1} w_i x^i$ can be defined as:

$$W(x) = U(x)V(x) \bmod I(x) \qquad (2)$$

Implementation of equation (2) can be carried out either in bit-serial multiplication or bit-parallel multiplication manner. The later can be fast as a single clock cycle but at the complexity of m^2. A bit-serial, on the other hand, is less complex but slower (m modules, m clock cycles). In both approaches, a fix-coefficient multiplier is much simpler than a general purpose one [1]. In the composite field environment, $GF(2^k)^n$, a special arrangement can be also made [2,3,4]. In some applications, representation of field element with other basis (such as normal basis and dual basis) [5,6], the operation can be made simpler.

2.6 Hardware Aspects

Cryptosystems based on Elliptic curve can be implemented under various constrains and applications. Sutikno [7] have implemented the ElGamal ECC processor in 32 bi-wide bus with 155 bit-wide register. Paar [3] had more focused to the implementation of the multiplier which is applicable to a composite field. Our application is aiming toward the minimum hardware at the PC serial port communication speed. At this fairly low speed, i.e. 9600 bps, there are not much constraints in hardware design and operation. We employed "MSB-First multiplier" with m processors. The operation of fix coefficient multiplication is fairly obvious such that the processor is simple as a latch or delay flip-flop. With some additional control circuitry, the same multiplier can be used for encryption and decryption.

3. Architecture

ElGamal Elliptic Curve Public-Key Encryption discussed in section 2 above can be more simplified when encrypter circuit and the decrypter circuits are sitting aside. That is, x', y', $(x')^{-1}$ and $(y')^{-1}$ can be pre-computed and kept in the registers. The encrypter is then just to compute

$$m_1 = M_1 x' \text{ and } m_2 = M_2 y' \qquad (3)$$

At the receiving end, the decrypter has to compute

$$m_1 (x')^{-1} = M_1 x' (x')^{-1} \qquad (4.1)$$

and

$$m_2 (y')^{-1} = M_2 y' (y')^{-1} \qquad (4.2)$$

The operation is illustrated in Fig.1. The advantage of this architecture is the minimum hardware requirement. Only one merely fix-coefficient multiplier can be used when the chip is programmed to be just only for encryption or decryption. Such a multiplier can be also used alternately for encryption and decryption in a half duplex operation. In this case, the fix-coefficient multiplier has to be modified to cover 2-4 coefficients. Even so, with careful selection of the multiply end, the complexity of the multiplier is simpler than a general purpose one. The architecture is depicted in Fig 2.

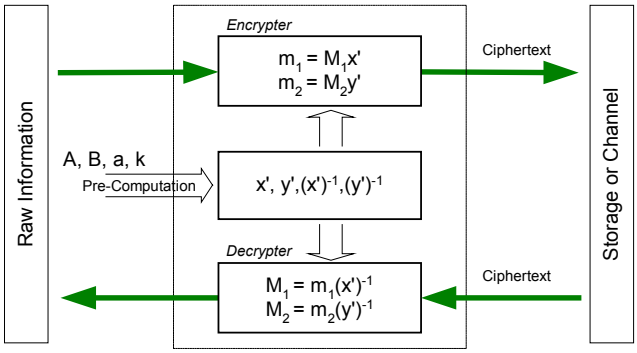

Fig. 1 Encryption and decryption scheme when the encrypter and the decrypter are sitting aside

Note that, the multiplication is performed over the Galois field, selection of the irreducible polynomial can effect the wiring complexity. In practical, select one that has small number of terms (i.e. 5 terms). The proposed architecture also has some drawbacks, for instants, it is not flexible to change the private key and/or public key. If one wants to do so, this piece of hardware then requires the capability of elliptic equation solving, inversion and multiplication. The circuit size then cannot be kept minimum. For current chip design and VLSI technology, the hard-wire program as our approach, it is not very difficult.

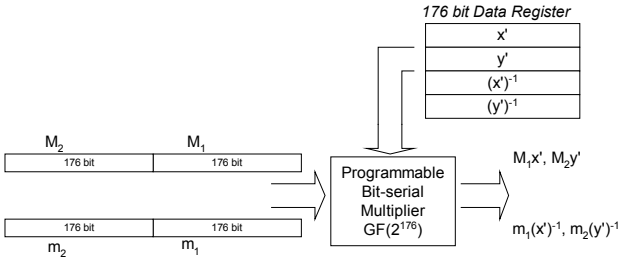

Fig. 2 An architecture when a single fix-coefficient bit-serial multiplier is used

4. Implementation

Our target is to design a tiny piece of hardware with the following features:

- Can be configured to work as an encrypter or a decrypter. The ElGamal ECC has to be incorporated.
- Encrypted data are kept on chip (smart card application)
- RS-232 interface standard (Baud rate =9600)

To verify the above concept, the prototype design was implemented with FPGA since the design time is fairly low and good degree of flexibility. We first focused on text data, however, with memory expansion image data can be also accommodated. The functional blocks are shown below in Fig. 3.

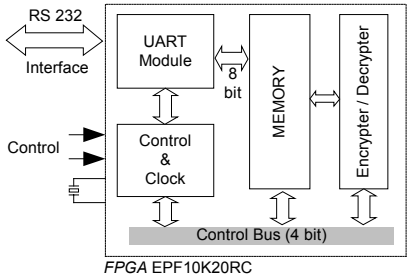

Fig. 3 Prototype design implemented with FPGA

Control lines control 4 functions of the chip which are: IN_RAM – Read data from the port and write to the memory, OUT_RAM – Read stored data and write to the port, ENCRY – Data in the memory are encrypted and write back, and DECRY – Data in the memory are decrypted and write back. These features may not quite necessary in the real application but at the state of verification they are quite important since the contents of memory can be simply dumped. We employed FPGA *EPF10K20RC* (Altera, 20,000 gates) in our design. The chip operates at 6.00 MHz clock frequency. Gates consuming can be detailed as:

UART Module	1600 gates
Control & Clock	2752 gates
Encrypter/Decrypter	5000 gates
Memory (256 Bytes)	2048 gates

It should be note that just about 60% of the available gates are utilized. In fact in the final design, the capacity of the memory has been increased to cover wider need of data storing. However for such a FPGA, the maximum number of gates that could be allowed for memory is 12288 gates (about 1500 bytes). In practical, on-chip flash ROM (smart card) or external Flash ROM modules (other applications) can be used.

5. Conclusions

In this work we have proposed the data encrypter and decrypter that have to work in a pair. ElGamal elliptic curve cryptosystems provided good security while the key size can be kept small, 176 bits in our case. Several computations, including inversion have been computed of line. This enabled the small hardware size. Carefully selection of x', y', (1/x') and (1/y') can further make the multiplier slightly more compact. With the proposed concept, on the fly encryption/decryption could be future developed to enhance data security during its transmission without the requirement of computer or additional software. Real time application may be also possible provided that the multiplication speed can be made to synchronize the data speed. The concept of on-chip storage can be extended for smart card application. The prototype design finished in FPGA has demonstrated very well, the concept as well as the future promise.

References

[1] S. Choomchuay, “On the Implementation of Finite Field Operations”, Ladkrabang Engineering Journal, Vol. 11, No. 1, June 1994, pp. 7-16.

[2] Christof Paar, "A New Architecture for a Parallel Finite Field Multiplier With Low Complexity Based on Composite Field", IEEE Trans. On Computers, Vol. 45, No. 7, July 1996.

[3] G. Orlando and C. Paar, "A Super-serial Galois Field Multiplier for FPGAs and its Application to Public-Key Algorithms", 7th Annual IEEE Symp. on Field-Programmable Custom Computing Machines, Napa, CA, Apr. 1999.

[4] S. Pongyupinpanich and S.Choomchuay, "Composite Field Bit Serial-Parallel Multiplier, Proc. of Information and Computer Engineering Workshop 2002 (ICEP2002), Prince of Songkla University, Jan. 2002, pp.65-69.

[5] E.R. Berlekamp, "Bit-serial Reed-Solomon Encoder", IEEE Trans. on Information Theory, Vol. IT-28, Nov. 1982, pp. 869-874.

[6] S. Choomchuay, “On the Implementation of Finite Field Basis Conversions”, EECON-17, King Mongkut's Institute of Technology, North Bangkok, 1994, pp. 482-486.

[7] S. Sutikno, A. Surya and R. Effendi, "An Implementation of ElGamal Elliptic Curves Crytosystems", Int. Symposium on Intelligent Signal Processing and Communication Systems (ISPACS ‘99), Phuket, Thailand, Dec.1999.

ISCIT 2002

Proceedings
The Second International Symposium on Communications and Information Technology

23-25 October 2002
Central Hotels&Resorts, Pattaya
Chonburi, Thailand

EIC

IEEE THAILAND SECTION

IEEE COMMUNICATIONS SOCIETY THAILAND CHAPTER