A FPGA Implementation of High Security Hybrid Reconfigurable Cryptographic Processor with RSA and SEA

A. Chitra , K. Sangeethalakshmi, P.Sivalakshmi, R.Shimila

Abstract- *Data security is in Demand in everyday life of Digital World, since Digital data's can be reproduced much easily. To achieve the maximum security required a Parallel Processing, User Reconfigurable Cryptographic RISC Microprocessor is proposed in our paper. Rather than protecting the data using tools and external codes, a microprocessor is specially designed in our project to offer maximum digital security. Cryptographic processor can be classified either as asymmetric cryptography or a symmetric cryptography processor. Asymmetric cryptography has the advantage of Reception security but has the limitation of High resource Utilization. And a symmetric cryptography processor has the limitation of single key security but comparatively has the advantages of low area, resource and power consumption. Thus in this project we are proposing Hybrid architecture in which both the advantage of asymmetric and symmetric cryptographies are combined. For implementation, Asymmetric RSA cryptography and a symmetric lightweight SEA encryption is combined to mutate a reconfigurable Cryptographic processor.*

Index words- Data security, Reduced instruction set Computer (RISC), Reconfigurable Architecture, Cryptographic Processor, Scalable Encryption Algorithm (SEA), Rivest-Shamir-Adelman (RSA) cryptosystem.

---

1 INTRODUCTION

Today more and more sensitive data is stored digitally. Bank accounts, medical records and personal emails are some categories that data must keep secure; this has made cryptography an important research topic. Cryptography is used for confidentiality, authentication, data integrity, and non-repudiation, which can be divided into two families: secret-key cryptography [1]-[3] and public-key cryptography [4]-[6]. Secret-key cryptography, which usually has a relatively compact architecture and smaller key size than public-key cryptography, is often used to encrypt/decrypt sensitive information or documents. Public-key cryptography offers fundamental technology for key agreement, encryption/decryption (two keys), and digital signatures. The use of systems with increasing complexity, which usually are more secure, has a result low throughput rate and more energy consumption. However the evolution of cipher has no practical impact, if it has only theoretical background. Every encryption algorithm should exploit as much as possible the conditions of the specific system without omitting the physical, area and timing limitations. This fact requires new ways in design architectures for secure and reliable crypto systems. After the advent of the internet and especially nowadays, security of data and protection of privacy have become a major concern for everyone's life, although mathematicians and researchers have been trying to address this problem since the end of the World war II, when cryptology science came out from the ambit of the army to enter the Bell laboratories. DES, RSA and PGP are only the tip of the iceberg of a vast amount of cryptographic algorithms developed by those scientists. The software implementations are 10-100 slower than the hardware.

The prevalence of microprocessors in all aspects of everyday life has led to information being easily and widely distributable in digital format. However, this information can be confidential or copyrighted material, or only intended for access by selected individuals. Therefore it is necessary that microprocessors are available which are capable of efficiently encrypting this information, without detriment to performance. RISC microprocessors are used abundantly in numerous communication systems, such as smart cards, mobile handsets, set-top boxes (STBs) and Personal Digital Assistants (PDAs). Implementing public-key cryptosystems on a general-purpose processor (GPP) is flexible because a variety of cryptosystems can be used at runtime. A drawback of GPP realizations that it generally results in a lower throughput rate and larger power consumption. Considerable effort has been directed toward a fast realization of cryptography algorithms consisting of very large integer operands (up to 4096 bits). For real-time applications, a dedicated hardware implementation is required to speed up the computation of cryptosystems. In particular, an application-specific integrated circuit (ASIC) solution generally leads to a higher throughput rate at a lower cost, but it is inflexible; therefore, it is only applied to a limited subset of cryptosystems. To mitigate the gap between GPP and ASIC realization, application-specific cryptographic processors have been proposed [7]-[9], each with its own instruction-set architecture, data path design, and target applications.

For example [8] have proposed their Reconfigurable cryptographic processor on Elliptic curve cryptography (ECC) and RSA [6] cryptosystems. Goodman [7] presented a reconfigurable cryptographic processor aimed at energy-constrained applications. Eslami's cryptographic processor [9] can support the two private key algorithms, AES [1] and DES [2], and ECC over for smart card applications.

2 PROPOSED CRYPTOGRAPHIC

PROCESSOR

Protecting the digital data through encryption using tools and external codes are highly cost effective and also results in performance degradation. To achieve much efficiency in encryption a reconfigurable cryptographic microprocessor is designed in this project to offer maximum digital security. With the conventional design of SEA [10] and RSA [6] standards as supporting co-processors, a logic module is also been implemented in the design to ensure the robustness of this processor design to serve all kind of encryption and decryption needs. A typical CPU unit with RAM, ALU, PC, Register bank and Buses are included as prioritized units for utilizing the Cryptographic co processers, which consists of Parallel Processing Unit, Bit permutation unit, sequencing cache and Byte permutation units. A sophisticated instruction sets will be derived to issue control signals to the main processor to initiate and control cryptographic operations. The performance evaluation of this processing design will be analyzed through a programmable FPGA kit.

A cryptographic processor can be classified either as asymmetric cryptography [8] or a symmetric cryptography processor [16]. Thus Asymmetric cryptography has the advantage of Reception security but has the limitation of High resource Utilization. And a symmetric cryptography processor has the limitation of single key security but comparatively has the advantages of low area, resource and power consumption.

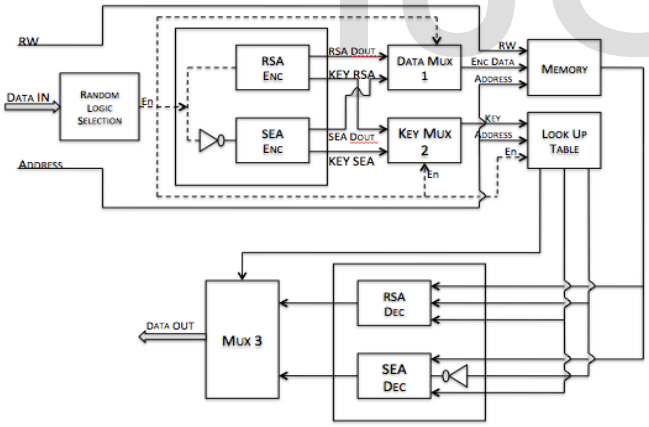

Fig 1: Block Diagram of our Proposed Architecture

Thus in this project a Hybrid architecture is proposed in which both the advantage of asymmetric and symmetric cryptographies are combined. For implementation a Asymmetric RSA cryptography and a symmetric light weight SEA encryption is combined to mutate a reconfigurable instruction driven processor.

3 ALGORITHM AND FORMULATIONS

In this section a brief introduction about RSA and SEA algorithms are provided.

3.1 RSA ALGORITHM

In 1977 Rivest and Adelman [] introduced what has become one of the most successful and widely used public key cryptosystem based on computing modular exponentiations. The algorithm has been implemented in many commercial applications It is based on a very simple number-theoretical idea, and yet it has been able to resist all cryptanalytic attacks. The idea is a clever use of the fact that, while it is easy to multiply two large primes, it is extremely difficult to factorize their product. Thus, the product can be publicized and used as the encryption key. The primes themselves cannot be recovered from the product and are used for decryption. As it was proposed [8].

3.1.1 STEPS IN RSA ALGORITHM

- Generate two large random primes, p and q. Compute n = pq and (φ) phi = (p-1)(q-1). n is known as the *modulus*
- Choose an integer e, 1 < e < phi, such that gcd (e, phi), e is the encryption exponent.
- Compute the decryption exponent d, 1 < d < phi, such that d = e-1 Mod phi.
- The public key is (n, e) and the private key is (n, d). Keep all the values d, p, q and phi secret.
- Send the public key to transmitter and secret key to receiver.
- Transmitter encrypt the original message, c = me Mod n, and send the cipher text to receiver.
- Receiver decrypt cipher text by, m = cd Mod n, and retrieve the original message.

3.1.2 MONTGOMERY MULTIPLICATION ALGORITHM

The two operations the RSA encryption and decryption are mutually inverse. This indicates that the original data can be recovered through the RSA encrypt/decrypt process. The operation of modular exponentiation is carried out iteratively by repeating a modular multiplication and modular squaring operation. The modular multiplication and squaring algorithm are based on Montgomery algorithm. The Montgomery multiplication [14]-[15] requires two multipliers first one is used for modular squaring and second using for modular multiplication. Registers are used to store the result from the parallel and serial way. The parallel register associated with the squarer is used to feed the parallel operand to the squarer while the serial register is used to feed the serial input to the squarer and multiplier. The parallel register associated with the multiplier is used to store the result from the multiplier, which is used as a parallel input to the multiplier.

IJSER

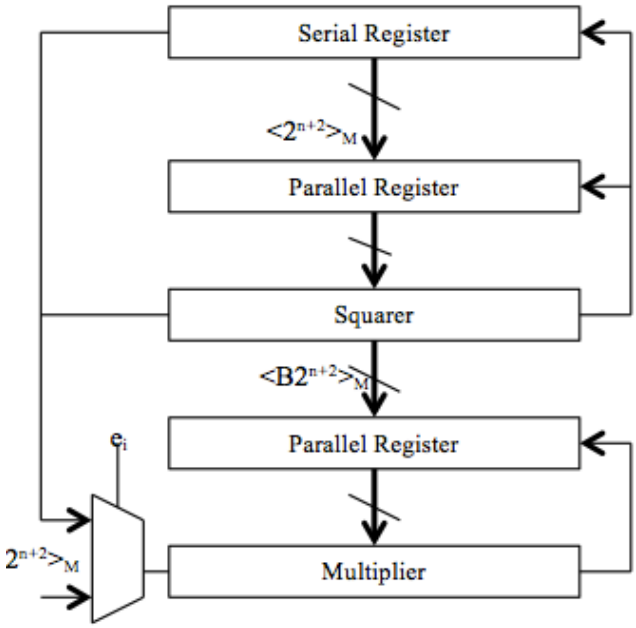

Fig a.CHITRA2: Montgomery multiplier algorithm

The Montgomery multiplication is carried out as follows, C=ME Mod N; $E \sum_{i=0}^{n} ei2^{!}$; P0 =2n+2Mod N; C0=2n+2Mod M, C=Pn For i= 0 to n-1; Ci+1=Ci2 Mod N; if ei=1 then Pi+1=PiCi Mod N, else Pi+1 =Pi ,
Let message M=10, Public key exponent e=5, n=15. Without Montgomery multiplication the equation is calculated as, Me mod n => 105 mod 5 => (10*10*10*10*10) mod N => 100000 mod 15, which is very difficult to calculate and large buffer size is needed to calculate the modular exponentiation. But with Montgomery multiplication algorithm, using squaring and repeated multiplication reduces the memory size. (102) Mod 15=I1; (I1*10) Mod 15=I2; (I2*10) Mod 15=I3 (I3*10) Mod 15=cipher text. Thus the memory required to store the intermediate results will not exceed more than N, Which will reduces the overall chip size by reducing the length of the register size. But it will increase the number of operation to get the cipher as well as decrypted text.

3.2. SEA ALGORITHM

Scalable encryption algorithm (SEA) is a parametric block cipher for resource-constrained systems (example sensor networks RFIDs). It was Initially designed as a low-cost encryption/authentication routine (i.e., with small code size and memory) targeted for processors with a limited instruction set (i.e., AND, OR, XOR gates, word rotation, and modular addition). This algorithm takes the plaintext, key, and the bus sizes as parameters and, therefore, can be straightforwardly adapted to various implementation contexts and/or security requirements.

3.2.1 PARAMETERS AND DEFINITIONS

SEAn, b operates on various text, key and word sizes. It is based on a Feistel structure with a variable number of rounds, and is defined with respect to the following parameters:

- n: plaintext size, key size.
- b: processor (or word) size.
- nb = 2nb: number of words per Feistel branch.
- nr: number of block cipher rounds.

Let x be an n/2 bit. In the following we will consider two representations,

- Bit representation: xb = x(n2 − 1) x(n2 − 2) . . . , x(2) x(1) x(0).
- Word representation: xW = xnb−1 xnb−2 . . . x2 x1 x0.

3.2.2 BASIC OPERATIONS

Due to simplicity constraints, SEA is based on a limited number of elementary operations denoted as follows:

3.2.2.1 BITWISE XOR ⊕ :

The bitwise XOR is defined on n/2 vectors,
⊕ :Z n/22 ×Z n/22 → Z n/22:x,
y → z = x⊕ y⇔z(i) = x (i)⊕ y (i).

3.2.2.2 SUBSTITUTION BOX S:

For efficiency purposes, it is applied bitwise to any set of three words of data using the following recursive definition:
S: Znb2b →Znb2b: x→x= S (x)⇔x3i = (x3i+2 ∧x3i+1)⊕ x3i,
x3i+1 = (x3i+2 ∧ x3i)⊕ x3i+1, x3i+2 =(x3i ∨x3i+1)⊕ x3i+2,
0≤i≤nb/3−1

3.2.2.3 WORD ROTATION R:

The word rotation is defined on nb-word vectors:
r: Znb2b→Znb2b :x→y=R(x)⇔yi+1 =xi , 0≤i≤nb −2,
y0=xnb-1.

3.2.2.4 BIT ROTATION R:

The bit rotation is defined on nb-word vectors:
r: Znb2b →Znb2b :x→y=r(x)⇔y3i =x3i ≫1, y3i+1 =x3i+,
y3i+2 =x3i+2 ≪1ρ≤i≤nb/3, where and represent the cyclic right and left shifts inside a word.

3.2.2.5 ADDITION MOD2B □:

The mod 2b addition is defined on nb-word vectors:
□: Znb2b ×Znb2b →Znb2b: x, y→z=x□y ⇔ z =x□y, 0≤i≤n −1.

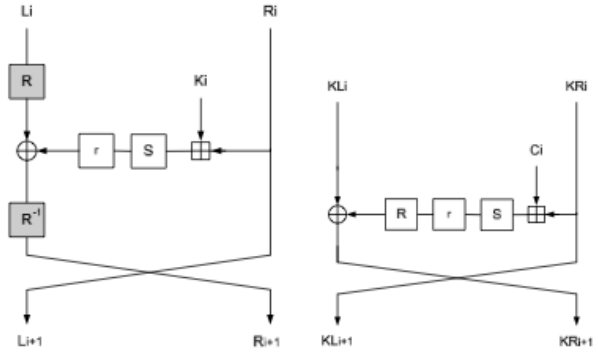

Fig 3: Encrypt/decrypt round and key round

3.2.2.6 THE ROUND AND KEY ROUND: Based on the previous definitions, the encrypt round FE, decrypt round FD and key round FK are pictured in Figure 3 and defined as the functions
F: Z22n/2× Z 2n/2 → Z 22n/2 such that:

[Li+1,Ri+1]=FE (Li, Ri, Ki)⇔ Ri+1 =R (Li) ⊕ r (S (Ri □Ki))
Li+1=Ri

[Li+1,Ri+1]=FD (Li, Ri, Ki) ⇔ Ri+1 =R1 (Li⊕ r (S (Ri□ Ki))),
Li+1=Ri

[KLi+1,KRi+1]=FK (KLi, KRi, Ci) ⇔ KRi+1 =K (Li) ⊕ R(r (S (KRi □Ci))), KLi+1=KRi .

4 HARDWARE DESCRIPTION

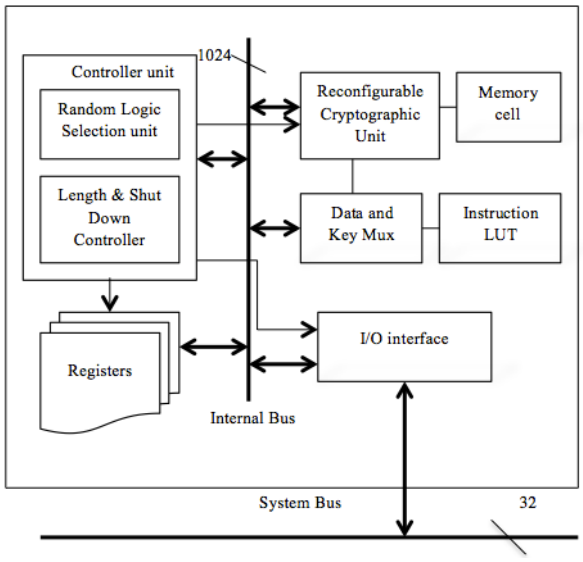

Fig 4: Reconfigurable Cryptographic Processor architecture

Fig-4 depicts the block diagram of the proposed reconfigurable cryptographic processor (RCP). The core of our RCP includes 1024 reconfigurable data path, which is used to carry out the required cryptographic operations. That is the functionality of the data path can be reconfigured for performing both symmetric SEA and asymmetric RSA.

4.1 RCP Architecture

As shown in Fig 4, the RCP consists of four main blocks controller unit, registers, I/O interface, and reconfigurable cryptographic unit. The choice of 32-bit system bus is to be compatible with most popular buses. The operand size and internal bus used in the RCP allows users to handle up to 1024-bit operations for cryptography applications. Note that buffers are required to translate data to/from the processor. The controller is responsible for controlling data path and for implementing cryptography algorithms. There are two units in the controller Random logic selection unit and Length and shut down controller. The random selection unit is to select the required cryptographic operation and the shut down controller is used to shut down the unused data path to save power. The registers are designed to store the initial values of algorithmic variables and intermediate results. The reconfigurable data path consists of six local registers (P, Q, N, Z, Pub, Pri), a parallel register, full adder and 1024 reconfigurable cells. The RCs can reconfigure to perform specified cryptographic operations and to set their interconnection with registers P, Q, N, Z, Pub, and Pri. The I/O interface is used for two purposes.

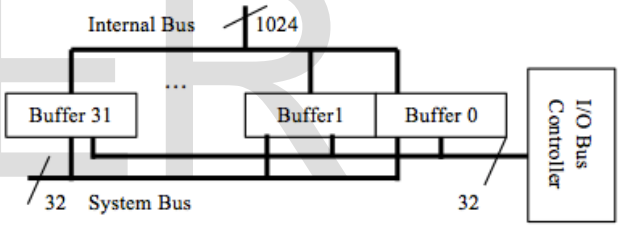

Fig 5. I/O interface

F

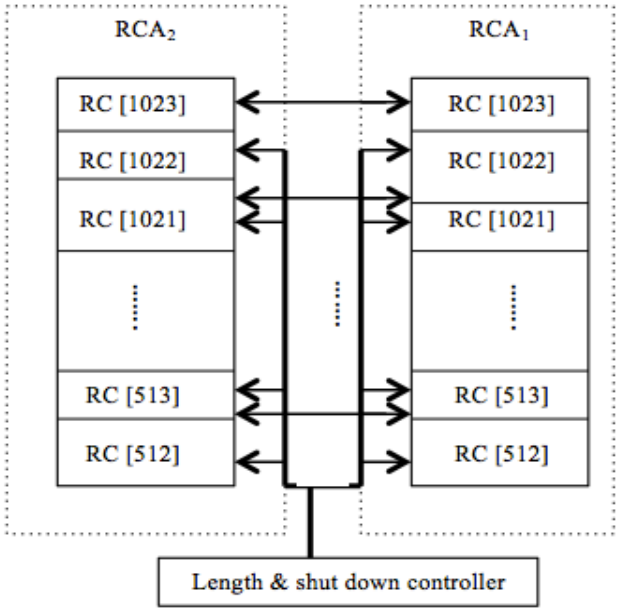

Fig 6. Shut Down controller

IJSER

The first one is to transfer data between the system bus and internal bus. The second one aligns data I/O data depending on the predetermined precision, i.e., operand size. Fig 5 shows the I/O structure, which consists of 32-32 buffers and an I/O bus controller for bus selection. Moving 1024 bits of data to/from the RCP thus requires 32 cycles.

5. RESULTS AND PERFORMANCE

The proposed work is implemented in the Quartus II version 10.0 and family cyclone II. The following tables table I, II and III show the synthesis details of RSA.

TABLE 1
RSA Mapping summary

Total Logic Elements	56
Dedicated logic Registers	38
Combinational functions	56
Total Pins	17

TABLE 2
RSA Power Analysis summary

Total thermal Power dissipation	70.41 mW
Core Static power Dissipation	47.36 mW
I/O thermal power dissipation	23.06mW

TABLE 3
RSA Timing Analysis

Frequency	177 MHZ
Worst case tco	8.410 ns
Clock set up	5.649 ns

The following tables show the synthesis summary of SEA algorithm implemented in Quartus II version 10.0 and family Stratix II.

TABLE 4
SEA Mapping Summary

Combinational ALUTs	368
Total pins	196

TABLE 5
SEA Power Analysis Summary

Total thermal Power dissipation	330.84 mW
Core Static power Dissipation	303.05 mW
I/O thermal power dissipation	27.80 mW

TABLE 6
SEA Timing Analysis Summary

Frequency	339.56 MHZ
Worst case tsu	6.091 ns
Worst case tco	15.969 ns
Worst case tpd	9.333 ns
Worst case th	2.636 ns
Clock set up	2.945 ns

6 CONCLUSIONS

A Hybrid Reconfigurable cryptographic processor that supports both the RSA and SEA cryptographic operations using reconfigurable data path was presented. The reconfigurable data path merged the RSA and SEA to obtain double security to meet the high security demands. Experimental results show that the developed cryptographic processor exhibits obvious speed and performance advantages in comparison with related works and offers more security.

REFERENCE

[1] Federal Information Processing Standards Publication 197, "Advanced Encryption Standard (AES)," 2001.

[2] U. S. Department of Commerce, Washington, DC, "National Encryption Standard," 1988.

[3] B. Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed. New York: Wiley, 1996.

[4] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, Handbook of Applied Cryptography. Boca Raton, FL: CRC Press, 1997.

[5] IEEE Standard Specifications for Public-Key Cryptography, IEEE STD 1363–2000, 2000.

[6] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Common. ACM, vol. 21, pp.120–126, Feb. 1978.

[7] J. Goodman and A. P. Chandrakasan, "An energy-efficient reconfigurable public-key cryptography processor," IEEE J. Solid-State Circuits, vol. 36, no. 11, pp. 1808–1820, Nov. 2001.

[8] Jun-Hong Chen, Ming-Der Shieh, and Wen-Ching Lin "High-Performance Unified-Field Reconfigurable Cryptographic Processor" IEEE Transactions On Very Large Scale Integration (VLSI) Systems, Vol. 18, No. 8, August 2010.

[9] Y. Eslami, A. Sheikholeslami, P. G. Gulak, S. Masui, and K. Mukaida, "An area-efficient universal cryptography processor for smart cards, "IEEE Trans. Very Large Scale Integer. (VLSI) Syst., vol. 14, no. 1, pp. 43–56, Jan. 2006.

[10] F. Mace, F. -X. Standaert, and J. -J Quisquater "FPGA implementation(s) of a Scalable Encryption Algorithm," in IEEE Transaction on very large scale integration (VLSI) systems, VOL.16, NO. 2, FEBRUARY 2008.

[11] F. -X. Standaert, G. Piret, N. Gershenfeld, and J. -J. Quisquater, "SEA: A Scalable Encryption Algorithm for Small Embedded Applications," in the Proceedings of CARDIS 2006, ser. LNCS, vol. 3928, Taragona, Spain, pp. 222–236 2006.

IJSER

[12] M. Feldhofer, J. Wolkerstorfer, and V. Rijmen, "AES Implementation on a Grain of Sand," in IEEE Proceedings on Information Security, vol. 152. IEE, pp. 13–20 Oct. 2005.

[13] K. Jarvinen, M. Tommiska, J. Skytta, "Comparative Survey of High- Performance Cryptographic Algorithm Implementations on FPGAs," IEEE Proceedings on Information Security, vol. 152, pp. 3–12 Oct. 2005.

[14] o. Nibouche, M.Nibouche and A.Bouridane, " High speed FPGA implementation of RSA Encryption algorithm", IEEE Transactions 2003.

[15] Montgomery P L. Mathematics of Computation, 40,170, (519) 1985

[16] Ricardo Chaves, Georgi Kuzmanov, "Reconfigurable memory based AES co-processor", In Proceedings of the 13th Reconfigurable Architectures Workshop IEEE Transactions 2006.

[17] P. L. Montgomery, "Modular multiplication without trial division," Math. Comp., vol. 44, no. 170, pp. 519-521, 1985.

[18] J. Guo and C. Wang, "A novel digit-serial systolic array for modular multiplication" in Proc. IEEE ISCAS '98-01.2,pp. 177-180, 1998.

[19] B. Mohammadian, M. Lotfizad, M. E. Nick, M. R. Mali. FPGA Implementation of 1024-bit Modular Processor for RSA Cryptosystem.

[20] Peter L Montgomery. Modular Multiplication Without Trial Division. Mathematics of Computation, 44 (170), April 1985·

IJSER