ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

A Framework on Residue Architectures for Cryptography

Kotte Narasimharao
M.Tech (ES & VLSI)
Chintalapudi Engineering College.

S.Vidya Rani
Assistant Professor
Chintalapudi Engineering College.

ABSTRACT

Residue systems of representation, like Residue Number Systems (RNS) for primary field (GF (p)) or Trinomial Residue Arithmetic for binary field (GF (2^k)), are characterized by efficient multiplication and costly modular reduction. On the other hand, conventional representations allow in some cases very efficient reductions but require costly multiplications.

The main purpose of this paper is to analyze the complexity of those two different approaches in the summations of products. As a matter of fact, the complexities of the reduction in residue systems and of the multiplication in classical representations are similar. One of the main features of this reduction is that it doesn't depend on the field. Moreover, the cost of multiplication in residue systems is equivalent to the cost of reduction in classical representations for special well-chosen fields.

Taking those properties into account, we remark that an expression like $A * B + C * D$, which requires two products, one addition and one reduction, evaluates faster in a residue system than in a classical one. So we propose to study types of expressions to offer a guide for choosing a most appropriate representation. One of the best domain of application is the Elliptic Curves Cryptography where addition and doubling points formulas are composed of products summation. The different kinds of coordinates like affine, projective, and Jacobean, offer a good choice of expressions for our study.

Keywords: ***Elliptic Curve Cryptography (ECC), modular addition, modular multiplication, modular reduction, Residue Number System (RNS), hardware implementation.***

INTRODUCTION

The computation of the Montgomery exponentiation (ME) in the Residue Number System (RNS) sanctions constraining the delay due to carry propagation and reaching a high degree of parallelism. This approach mainly requires the execution of a set of Montgomery multiplications (MMs). However, in RNS, some operations (e.g. division, comparison, modulo) are natively arduous to execute. Hence, several approaches have been proposed in order to planarity exploit the potential of RNS for modular exponentiation, by minimizing the impact of cognate drawbacks. A key element of these approaches is the Base Extension (BE), which calculates a number on a different RNS base.

A paramount number of applications including cryptography, error rectification coding, computer algebra, DSP, etc., rely on the efficient realization of arithmetic over finite fields of the form GF (2n), where n € Z and n ≥1, or the GF(p) form ,where p a prime. Cryptographic applications form a special case, since, for security reasons, they require immensely colossal integer operands efficient field multiplication with astronomically immense operands is crucial for achieving a satiating cryptosystem performance, since multiplication is the most time- and area-consuming operation. Therefore, there is a desideratum for incrementing the speed of cryptosystems employing modular arithmetic with the least possible area penalty. A flamboyantly discernible approach to achieve this would be through parallelization of their operations. In recent years, RNS and PRNS have relished renewed scientific interest due to their ability to perform expeditious and parallel modular arithmetic. Utilizing RNS/PRNS, a given path accommodating an astronomically immense data range is superseded by parallel paths of more diminutive dynamic ranges,

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

with no desideratum for exchanging information between paths. As a result, the utilization of residue systems can offer reduced involution and power consumption of arithmetic units with sesquipedalian word lengths. On the other hand, RNS/PRNS implementations bear the extra cost of input converters to translate numbers from a standard binary format into residues and output converters to translate from RNS/PRNS to binary representations.

An incipient methodology for embedding residue arithmetic in a dual-field Montgomery modular multiplication algorithm for integers in GF(p) and for polynomials in GF (2n) is presented in this paper. The mathematical conditions that need to be satiated for a valid RNS/PRNS incorporation are examined. The derived architecture is highly parallelizable and multifarious, as it fortifies binary-to-RNS/PRNS and RNS/PRNS-to-binary conversions, Commixed Radix Conversion (MRC) for integers and polynomials, dual-field Montgomery multiplication, and dual-field modular exponentiation and inversion in the same hardware residue number system.

The Residue Number System (RNS) is a non-weighted number system that can map astronomically immense numbers to more diminutive residues, without any desideratum for carry propagations .Its most paramount property is that integrations, subtractions, and multiplications are inherently carry-free. These arithmetic operations can be performed on residue digits concurrently and independently. Thus, utilizing residue arithmetic, would in principle, increment the speed of computations RNS has shown high efficiency in realizing special purport applications such as digital filters , image processing , RSA cryptography and concrete applications for which only integrations, subtractions and multiplications are utilized and the number dynamic range is concrete. Special moduli sets have been used extensively to reduce the hardware involution in the implementation of converters and arithmetic operations. Among which the triple moduli set {2n+1,2n,2n-1} have some benefits. Since the operation of multiplication is of major paramountcy for virtually all kinds of processors, efficient implementation of multiplication modulo 2n-1 is consequential for the application of RNS.

RNS DEFINITION:

A residue number system is characterized by a predicate that is not a single radix but an N-tuple of integers (mN,mN-1 … m1). Each of these mi (i = 1, 2, … N) is called a modulus. An integer "X" is represented in the residue number system by N-tuple (xN, xN-1… x1) where xI is a nonnegative integer gratifying

$$X = m_I * q_I + x_I \quad , \ldots\ldots\ldots\ldots (1)$$

Where qI is the most sizably voluminous integer such that 0<=xI<= (mI – 1). xi is kenned as the residue of X modulo mi, and notations X mod mi and |X| mi are commonly utilized.

Example:

Consider a two-moduli system with moduli $m_2 = 3$ and $m_1 = 2$. The representation of X = 5 in this residue number system is (x_2, x_1) where

$$X_2 = X \bmod m_2 = 5 \bmod 3 = 2 \text{ or, } |5|_3 = 2$$

$$X_1 = 5 \bmod 2 = 3$$

Now,

If X = -2 then

$$X_2 = -2 \bmod 3 = 1 \ (-2 = 3^* -1 + 1)$$

$$X_1 = -2 \bmod 2 = 0$$

Therefore, the residue representation of 5 is (2, 1). The number "X" can be of any integer.

In Residue Number Systems (RNS), an integer X is represented by its residues {x0…..xn-1} modulo a base of relatively prime numbers {m0….mn-1}. Thus an astronomically immense number can be represented as a set of diminutive integers. Advisement and multiplication can be facilely parallelized, there is no carry propagation. The time is reduced to the evaluation of these operations with diminutive numbers. This representation is utilizable in cryptography and digital signal processing. Furthermore, in these two domains, modular multiplication (A X B mod N) is frequently utilized.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Advantages:

The RNS system provide a unique feature of parallelism that make arithmetic operations such as integration, subtraction and modulation very facile to handle and perform incrementing speed and reducing chip area.

- Carry free High-Speed
- Parallel Operation
- Low Power Circuits
- Medium Security
- Error Detection and Correction Capability
- Fault Tolerant

RNS has its disadvantages additionally. Operations such as division, sign-detection, magnitude comparison and overflow detection are intricate and hard to implement. This has constrained the application of RNS to certain fields where integration/multiplication operations are utilized extensively and the result is kenned to be within a predetermined range. RNS work only for integer values therefore integrating extra cost for conversion from binary-to-RNS and vice versa.

Applications:

The residue number system is very alluring solution to many researchers especially during the last decade. Extensive research have been put on the theory of amending the RNS system and applying it in some application areas such as, digital signal processing, digital filters, expeditious Fourier transform (FFT), and image processing.

The RNS is inherently parallel, modular and fault tolerant. Performing operations such as advisement, subtraction, and multiplication is inherently carry-free, thus reducing a substantial amount of circuit integration area where carry-detection circuitry had to be implemented afore.

- RSA Algorithm
- Digital Signal Processing
- Digital Filtering
- Image Processing
- Error Detection and Correction

BASIC RSA ALGORITHM:

RSA is a popular cryptography algorithm widely utilized in signing and encrypting operations for security systems.

The RSA algorithm defines a mechanism to secure the message exchanges in communication systems by providing two types of accommodations: authentication and data integrity [9]. Authentication consists of signing and verifying operations to assure the identities of the message sender. In the signing operation the Sender takes the message M, his private signing key D and N from the public key (E, N) to compute the signature S by: S = MD mod N (1) The signing key D is much more astronomically immense than the verifying key E. Thus the performance of RSA relies on expeditious implementation of the signing operation S = MD mod N.

Data that can be read and understood without any special measures is called plaintext or clear text. The method of dissimulating plaintext in such a way as to obnubilate its substance is called encryption. Encrypting plaintext results in unreadable gibberish called cipher text. You utilize encryption to ascertain that information is obnubilated from anyone for whom it is not intended, even those who can visually perceive the encrypted data. The process of reverting cipher text to its pristine plaintext is called decryption. Figure 1.1 illustrates this process.

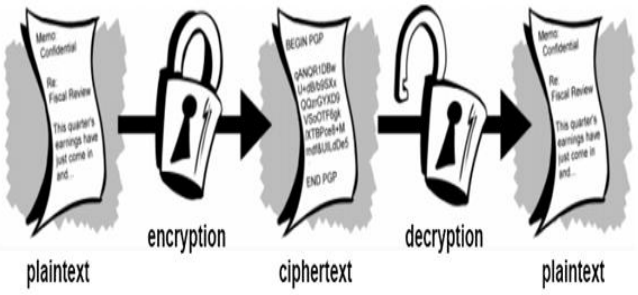


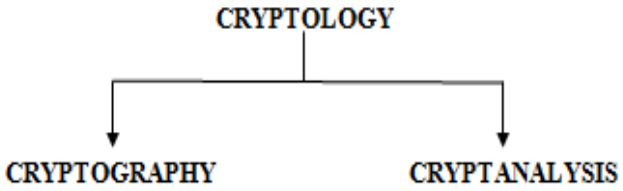
Figure 1.1 Encryption and decryption

CRYPTOGRAPHY

To enhance the security of the data, code language for indicting messages were utilized. The branch of

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

mathematics that investigates the code languages and methods is called cryptology. Cryptology consists of two streams namely cryptography and cryptanalysis. Cryptography is a science of coding message furtively while cryptanalysis is a science of breaking codes.



Our project is concerned with cryptography. Cryptography is a science of utilizing mathematics to encrypt and decrypt data. Cryptography enables to store sensitive information or transmit it across insecure networks so that it cannot be read by any one except the intended recipient.

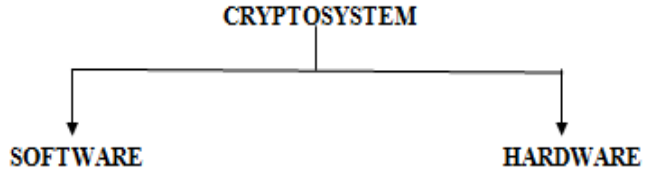
Cryptography or Cryptology is derived from Greek kryptos "hidden" and the verb grafo "write" or legion "to speak" is the practice and study of obnubilating information. In modern times, Cryptology is considered to be a branch of both mathematics and computer science, and is afflicted proximately with information theory, computer security and engineering. Cryptography is utilized in applications present in technology advanced in societies; examples include the security of the ATM cards, computer pass words and electronic commerce which all depend upon Cryptography.

HOW DOES CRYPTOGRAPHY WORK

A cryptographic algorithm, or cipher, is a mathematical function utilized in the encryption and decryption process. A cryptographic algorithm works in cumulating with a key—a word, number, or phrase—to encrypt the plaintext. The same plaintext encrypts to different cipher text with different keys.

The security of encrypted data is entirely dependent on two things: the vigor of the cryptographic algorithm and the secrecy of the key.

A cryptographic algorithm, plus all possible keys and all the protocols that make it work comprise a cryptosystem. PGP is a cryptosystem. Cryptosystem can be divided in to Software and Hardware.



THE PURPOSE OF CRYPTOGRAPHY

In data and telecommunications, cryptography is compulsory when communicating over any un-trusted medium, which includes just about any network, particularly the Internet.

Within the context of any application-to-application communication, there are some categorical security requisites including:

Authentication: The process of proving one's identity. (The primary forms of host-to-host authentication on the Internet today are name-predicated or address-predicated, both of which are notoriously impotent.)

Privacy/confidentiality: Ascertaining that no one can read the message except the intended receiver.

Integrity: Assuring the receiver that the received message has not been altered in any way from the pristine.

Non-repudiation: A mechanism to prove that the sender genuinely sent this message.

Cryptography, then, not only forefends data from larceny or alteration, but can withal be utilized for utilize authentication. There are, in general, three types of cryptographic schemes typically used to accomplish these goals: secret key (or symmetric) cryptography, public-key (or asymmetric) cryptography, and hash functions, each of which is described below. In all cases, the initial unencrypted data is referred to as plaintext. It is encrypted into cipher text, which will in

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

turn (customarily) be decrypted into utilizable plaintext.

SYMMETRIC CRYPTOGRAPHY

In a cryptosystem that utilizes symmetric cryptography, both parties will be utilizing the same key for encryption and decryption, as shown in Figure 1.2. This provides dual functionality. As we verbally expressed, symmetric keys are additionally called secret keys because this type of encryption relies on each utilizer to keep the key a secret and felicitously bulwarked. If this key got into an intruder's hand, that intruder would have the ability to decrypt any intercepted message encrypted with this key.

Each pair of users who want to exchange data utilizing symmetric key encryption must have their own set of keys. This denotes if Dan and Iqqi want to communicate, both need to obtain a facsimile of the same key. If Dan withal wants to communicate utilizing symmetric encryption with Norm and Dave, he now needs to have three separate keys, one for each friend.

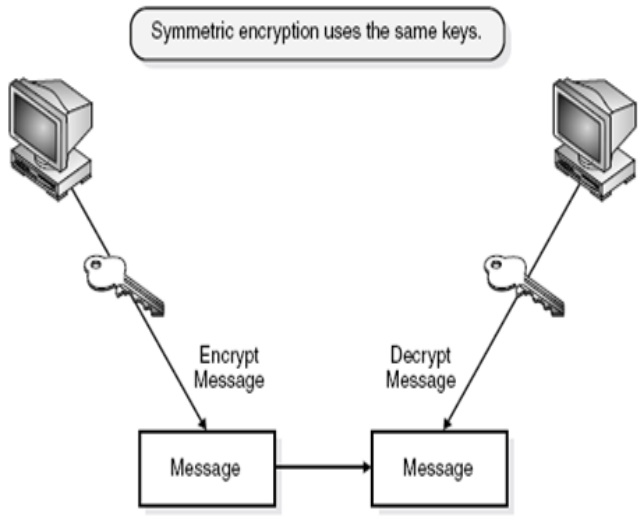


Figure 3.2 Using symmetric algorithms, the sender and receiver use the same key for encryption and decryption functions.

This might not sound like an immensely colossal deal until Dan realizes that he may communicate with hundreds of people over a period of several months, and keeping track and utilizing the correct key that corresponds to each concrete receiver can become a very daunting task. If Dan were going to communicate with 10 other people, then he would require to keep track of 45 different keys. If Dan were going to communicate with 100 other people, then he would have to maintain and keep up with 4,950 symmetric keys. Dan is a pretty effulgent guy, but does not indispensably want to spend his days probing for the right key to be able to communicate with Dave.

The security of the symmetric encryption method is plenarily dependent on how well users bulwark the key. This should raise red flags to you if you have ever had to depend on a whole staff of people to keep a secret. If a key is compromised, then all messages encrypted with that key can be decrypted and read by an intruder.

Examples

- Data Encryption Standard (DES)
- Triple DES (3DES)
- Advanced Encryption Standard (AES)

ASYMMETRIC CRYPTOGRAPHY

Some things you can tell the public, but some things you just want to keep private. In symmetric key cryptography, a single secret key is utilized between entities, whereas in public key systems, each entity has different keys, or asymmetric keys. The two different asymmetric keys are mathematically cognate. If a message is encrypted by one key, the other key is required to decrypt the message.

In a public key system, the dyad of keys is composed of one public key and one private key. The public key can be kenned to everyone, and the private key must only be kenned to the owner. Many times, public keys are listed in directories and databases of e-mail addresses so they are available to anyone who wants to utilize these keys to encrypt or decrypt data when communicating with a particular person. Figure 1.3 illustrates an asymmetric cryptosystem.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

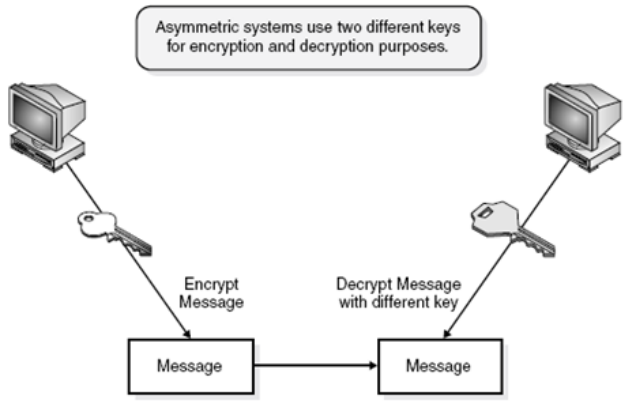


Figure 3.3 Asymmetric cryptosystem

The public and private keys are mathematically cognate, but cannot be derived from each other. This designates that if an evildoer gets a replica of Bob's public key, it does not mean he can now utilize some mathematical magic and ascertain Bob's private key.

If Bob encrypts a message with his private key, the receiver must have a facsimile of Bob's public key to decrypt it. The receiver can decrypt Bob's message and decide to reply back to Bob in an encrypted form. All she requires to do is encrypt her replication with Bob's public key, and then Bob can decrypt the message with his private key. It is not possible to encrypt and decrypt utilizing the exact same key when utilizing an asymmetric key encryption technology.

Bob can encrypt a message with his private key and the receiver can then decrypt it with Bob's public key. By decrypting the message with Bob's public key, the receiver can be sure that the message genuinely emanated from Bob. A message can only be decrypted with a public key if the message was encrypted with the corresponding private key. This provides authentication, because Bob is the only one who is supposed to have his private key. When the receiver wants to ascertain Bob is the only one that can read her replication, she will encrypt the replication with his public key. Only Bob will be able to decrypt the message because he is the only one who has the obligatory private key.

Now the receiver can withal encrypt her replication with her private key in lieu of utilizing Bob's public key. Why would she do that? She wants Bob to ken that the message emanated from her and no one else. If she encrypted the replication with Bob's public key, it does not provide authenticity because anyone can get a hold of Bob's public key. If she utilizes her private key to encrypt the message, then Bob can be sure that the message emanated from her and no one else. Symmetric keys do not provide authenticity because the same key is utilized on both ends. Utilizing one of the secret keys does not ascertain that the message originated from a categorical entity.

Examples

- RSA
- Elliptic Curve Cryptosystem (ECC)
- Diffie-Hellman
- El Gamal
- Digital Signature Standard (DSS)

TYPES OF CRYPTOGRAPHIC ALGORITHMS

There are several ways of relegating cryptographic algorithms. For purposes of this paper, they will be categorized predicated on the number of keys that are employed for encryption and decryption, and further defined by their application and use. The three types of algorithms those are discussed in Figure 1.5.

Secret Key Cryptography (SKC): Utilizes a single key for both encryption and decryption

Public Key Cryptography (PKC): Uses one key for encryption and another for decryption

Hash Functions: Utilizes a mathematical transformation to irreversibly "encrypt" formation

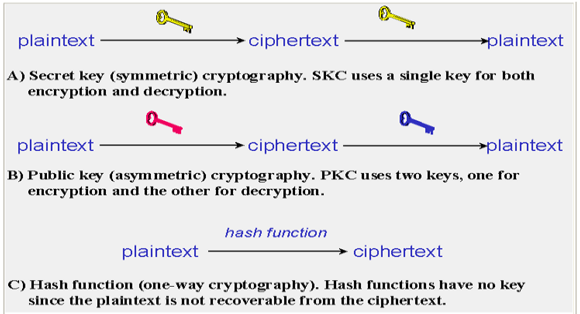


Figure 3.5 Three types of cryptographic algorithms

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

RESIDUE ARITHMETIC

RNS consists of a set of ,L pair-sapient relatively prime integers A= (m1.m2,……,mL) (called the base) and the range of the RNS is computed as . Any integer Z €[0,A-1] has a unique RNS representation ZA given by ZA=(Z1,Z2,….ZL) = ((Z)m1, (Z)m2, (Z)m3, (Z)m4…….. (Z)mL), where (Z)mi denotes the operation Z mod mi. Postulating two integers a, b in RNS format, i.e., aA=(a1,a2,….aL) and, bA=(b1,b2,…bL) then one can perform the operations in parallel by

$$a_{\mathcal{A}} \otimes b_{\mathcal{A}} = \left(\langle a_1 \otimes b_1 \rangle_{m_1}, \langle a_2 \otimes b_2 \rangle_{m_2}, \ldots, \langle a_L \otimes b_L \rangle_{m_L}\right)$$

To reconstruct the integer from its residues, two methods maybe employed [14]. The first is through the CRT according to

$$z = \sum_{i=1}^{L} \left\langle z_i \cdot A_i^{-1} \right\rangle_{m_i} \cdot A_i - \gamma A$$

where Ai =A/ai and A^{1}iis the multiplicative inverse of Aionai . The result of ∑ ki=1((xi A1i) mod ai) Aiis equal tox + A, where < k.

APPLICATIONS

1. Secrecy in transmission
2. Secrecy in storage
3. Integrity in transmission
4. Integrity in storage
5. Authentication of identity
6. Electronic signatures.

CONCLUSION

The mathematical framework and a flexible, dual-field, residue arithmetic architecture for Montgomery multiplication in and is developed and the necessary conditions for the system parameters (number of modulus word length) are derived. The proposed DRAMM architecture supports all operations of Montgomery multiplication in and , residue-to-binary and binary-to-residue conversions, MRC for integers and polynomials ,dual-field modular exponentiation and inversion, in the same hardware. Generic complexity and real performance comparisons with state-of-the-art works prove the potential of residue arithmetic exploitation in Montgomery multiplication.

REFERENCES

[1] I. Blake, G. Seroussi, andN.Smart, Elliptic Curves in Cryptography. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[2] D. Hankerson, A. Menezes, and S. Vanstone, Guide to Elliptic Curves Cryptography. New York, NY, USA: Springer-Verlag& Hall/CRC, 2004.

[3] J.-P. Deschamps, Hardware Implementation of Finite-Field Arithmetic. New York, NY, USA: McGraw-Hill, 2009.

[4] R. Lidl and H. Niederreiter, Introduction to Finite Fields and Their Applications. New York, NY, USA: Cambridge Univ. Press, 1986.

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

[6] S. Kawamura, M. Koike, F. Sano, and A. Shimbo, "Cox-Rower architecture for fast parallel Montgomery multiplication," in EUROCRYPT' 00: Proc. 19th Int. Conf. Theory and Application of Cryptographic Techniques, 2000, pp. 523–538.

[7] J.-C. Bajard and L. Imbert, "Brief contributions: A full RNS implementation of RSA," IEEE Trans. Comput., vol. 53, no. 6, pp. 769–774, Jun. 2004.

[8] D. Schinianakis, A. Fournaris, H. Michail, A. Kakarountas, and T. Stouraitis, "An RNS implementation of an elliptic curve point multiplier," IEEE Trans. Circuits Syst.I, vol. 56, no. 6, pp. 1202–1213, Jun. 2009.

[9] A. Halbutoğullari and Ç. K. Koç, "Parallel multiplication in using polynomial residue arithmetic," Design, Codes and Cryptography, vol. 20, no. 2, pp. 155–173, Jun. 2000.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

[10] M. G. Parker and M. Benaissa, " multiplication using polynomial residue number systems," IEEE Trans. Circuits Syst. II, vol. 42, no. 11, pp. 718–721, Nov. 1995.

[11] H. Nozaki, M. Motoyama, A. Shimbo, and S.-I. Kawamura, "Implementation of RSA algorithm based on RNS Montgomery multiplication," inProc. 3rd Int.Workshop on Cryptographic Hardware and Embedded Systems (CHES '01), 2001.

[12] J.-C. Bajard, L. Imbert, and G. A. Jullien, "Parallel Montgomery multiplication in using Trinomial Residue Arithmetic," in IEEE Symp. Computer Arithmetic, 2005, vol. 0, pp. 164–171.

[13] N. Guillermin, "A high speed coprocessor for elliptic curve scalar multiplications over ," in Cryptographic Hardware and Embedded Systems, CHES 2010, 2010, pp. 48–64, Lecture Notes in Computer Science 6225.

[14] F. J. Taylor, "Residue arithmetic: A tutorial with examples," IEEE Computer, vol. 17, pp. 50–62, May 1988.

[15] J. Omura and J. Massey, "Computational method and apparatus for finite field arithmetic," U.S. Patent 4,587,627, May 6, 1986.

[16] R. Mullin, I. Onyszchuk, S. Vanstone, and R. Wilson, "Optimal normal bases in ," Discrete Applied Mathematics, vol. 22, pp. 149–161, 1988.

[17] G. Agnew, R. Mullin, and S. Vanstone, "An implementation of elliptic curve cryptosystems over ," IEEE J. Select. Areas Commun., vol. 11, no. 5, pp. 804–813, Jun 1993.

[18] R. Schroeppel, H. Orman, S. W. O'Malley, and O. Spatscheck, "Fast key exchange with elliptic curve systems," in CRYPTO '95: Proc. 15th Annu. Int. Cryptology Conf. Advances in Cryptology, 1995, pp. 43–56.

[19] Ç. K. Koç and T. Acar, "Montgomery multiplication in ," Design, Codes and Cryptography, vol. 14, no. 1, pp. 57–69, Apr. 1998.

[20] D. Schinianakis,A.Kakarountas, T. Stouraitis, and A. Skavantzos, "Elliptic curve point multiplication in using Polynomial Residue Arithmetic," in Proc. IEEE Int. Conf. Electronics, Circuits, and Systems (ICECS 2009), 2009, pp. 980–983.

[21] A. Skavantzos and F. J. Taylor, "On the polynomial residue number system," IEEE Trans. Signal Process., vol. 39, no. 2, pp. 376–382, Feb. 1991.

[22] C. McIvor, M. McLoone, and J. McCanny, "Modified Montgomery modular multiplication and RSA exponentiation techniques," IEE Proc.—Computers and Digital Techniques, vol. 151, no. 6, pp. 402–408, Nov. 2004.

[23] M. Sudhakar, R. Kamala, and M. Srinivas, "A bit-sliced, scalable and unified Montgomery multiplier architecture for RSA and ECC," in IFIP Int. Conf. Very Large Scale Integration, 2007, VLSI—SoC 2007, Oct. 2007, pp. 252–257.

[24] M.-D. Shieh, J.-H.Chen, H.-H.Wu, andW.-C. Lin, "A new modular exponentiation architecture for efficient design of RSA cryptosystem," IEEE Trans. VLSI Syst., vol. 16, pp. 1151–1161, Sep. 2008.

[25] M. Huang, K. Gaj, S. Kwon, and T. El-Ghazawi, "An optimized hardware architecture for the Montgomery multiplication algorithm," in Proc. Practice and Theory in Public Key Cryptography, PKC'08, 2008, pp. 214–228.

[26] W. Freking and K. Parhi, "Performance-scalable array architectures for modular multiplication," in Proc. IEEE Int. Conf. Application-Specific Systems, Architectures, and Processors, 2000, pp. 149–160.

[27] P. Amberg, N. Pinckney, and D. Harris, "Parallel high-radix Montgomery multipliers," in Proc. 42nd

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Asilomar Conf. Signals, Systems and Computers, Oct. 2008, pp. 772–776.

[28] N. R. Pinckney and D. M. Harris, "Parallelized radix-4 scalable Montgomery multipliers," Integr. Circuits Syst., vol. 3, no. 1, pp. 39–45, Nov. 2008.

[29] M.-D. Shieh and W.-C. Lin, "Word-based Montgomery modular multiplication algorithm for low-latency scalable architectures," IEEE Trans. Comput., vol. 59, no. 8, pp. 1145–1151, aug. 2010.

[30] A. Tenca and Ç. K. Koç, "A scalable architecture for modular multiplication based on Montgomery's algorithm," IEEE Trans. Comput., vol. 52, no. 9, pp. 1215–1221, sep. 2003.

[31] D. Harris, R. Krishnamurthy, M. Anders, S. Mathew, and S. Hsu, "An improved unified scalable radix-2 Montgomery multiplier," in Proc. IEEE Symp. Computer Arithmetic, 2005, vol. 0, pp. 172–178.

[32] A. Satoh and K. Takano, "A scalable dual-field elliptic curve cryptographic processor," IEEE Trans. Comput., vol. 52, pp. 449–460, Apr. 2003.