Research Journal of Recent Sciences ISSN 2277-2502
Vol. **3(3),** 89-92, March **(2014)** *Res.J.Recent Sci.*

Review Paper

Fault-Tolerant and Information security in Networks using Multi-level Redundant Residue Number System

Pazahr Ali[1],Mizanian Kambiz [2],Safi Seyyed Mohammad[1], Taghipour Eivazi Shiva [3] and Rezaei Mehdi[4]
[1]Department of Computer Engineering, Ahvaz branch, Islamic Azad University, Ahvaz, IRAN
[2]Department of Computer Engineering, Sharif University of Technology, Tehran, IRAN
[3]Department of Computer Engineering, Science and Research branch, Islamic Azad University, Tehran, IRAN
[4]Institute for Fundamental Sciences (IPM), School of Computer Science,Tehran, IRAN

Available online at: www.isca.in, www.isca.me
Received 22nd January 2013, revised 25th August 2013, accepted 8th October 2013

Abstract

Residue Number System (RNS) is a non-weighted integer system that can speed up arithmetic operations, Since the operations are done in parallel with no carry. Redundant RNS (RRNS) is a trait of RNS provides reliable communication in networks which can detect the errors and correct them. Also this system can be used in cryptography algorithms because it is fast. Here, it is the suggested RRNS to be used in a Multi-Level system to achieve both reliable and secure connections.

Keywords: Computer Arithmetic, RNS, Error detection, Error Correction, RSA

Introduction

Information security and reliable communications has been more important during the last decades. It would be necessary to use cryptography algorithms providing secure data communications. Nowadays, asymmetric key cryptography named Public key such as RSA is used [1]. To encrypt the data in most communications at least a 1024-bit key length is needed [2]. Also so many factors include noise effect on the data in communications, which can be overcome preferably by adding some fault tolerance tools such as error detection and possibly correction. These two properties require high speed processing. Thus, to provide a suitable hardware for RSA, a secure communication Multi-Level Redundant Residue Number System is suggested, which can perform arithmetic operations in parallel as well as detect and correct the errors.

This paper is organized as follows: In section 2, the original Residue number system with its specifications is introduced. A brief review of RRNS is defined in section 3. The multi-level RNS, which is used to speed up the arithmetic operations in RNS, is presented in section 4. The common cryptography algorithm (RSA) is introduced in section 5 which can be handled by RNS. Section 6 consists of the implementation of RSA by multi-level RRNS, which enables the system to have a secure and reliable communication.

Residue Number System

RNS is a non-weighted, carry-free system that uses the reminder of a number to a set of positive integers named module $\{P_1, P_2, ..., P_n\}$. The system will have the largest possible dynamic range (M) where $M = P_1P_2...P_n$, and $GCD(P_i, P_j) = 1,\ i \neq j$.

Any integer X in dynamic range of $0 \leq X < M$ can be represented as $X = (x_1, x_2, ..., x_n)$ where x_i denotes X mod P_i.

The residues are smaller than the original number and as a result, the arithmetic operations such as addition, subtraction and multiplication could be carried out independently and parallel between different modules[3]. Therefore, RNS is suitable in an application including addition, subtraction and multiplication widely, such as RSA[4-6], digital signal processing[7], image processing[8], fault tolerant[9].

Redundant RNS

To design a circuit, being able to detect error and correct it in RNS, redundant modules should be used. This system is named as Redundant Residue Number System (RRNS).

RRNS is a kind of RNS which detects and corrects the errors, due to adding some other modules to the original modules named redundant modules. The detection and correction feature of RRNS is due to the number of redundant modules which is used[10].

The module set $\{P_1, P_2, ..., P_n\}$ is non-redundant module set, known as RNS. By adding the k redundant modules, the system is denoted by $\{P_1, P_2, ..., P_n, P_{n+1}, P_{n+2}, ..., P_{n+k}\}$. As it can be seen, the modules consist of two parts, the part one comprises n original modules implying dynamic range; the second part includes k redundant modules to enable the system to detect and correct the errors. By using the k redundant module, the system can detect at most k residue errors as well as correct up to $\lfloor k/2 \rfloor$ errors [10].

Multi-Level RNS

In Residue number system, if the forward converter is applied on each residue with a new module set, the multi-level RNS is obtained which can be repeated until very small moduli is obtained (see figure 1). As it can be seen the small moduli set leads to applied small remainders. It is important to note that the arithmetic operations are in the last level of the system. Therefore, these operations will be on very small integers leading faster parallel operations along with decreasing power consumption. Also, this system can be used widely to increase the security and fault tolerance.

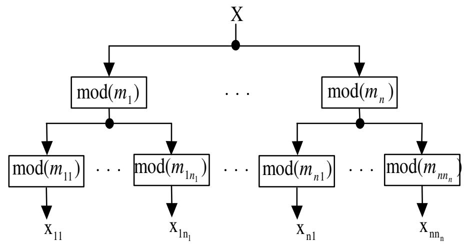

Figure-1
Multi-Level RNS

Clearly, the module set of first level of Residue Number System is $\{P_1, P_2, P_3, \dots, P_n\}$. Then the remainder of the first level is converted into the corresponding module set of second level as $\{p_{i1}, p_{i2}, p_{i3}, \dots, p_{in_i}\}$ for modulo m_i, $i = 1, 2, 3, \dots, n$.

This calculation can be carried out until the kth level when a very small module is obtained. However, it is important to note that the dynamic range of *i*th level in this system must be greater than the previous one, (*i*-1)th level [11].

Also, the remainder of each level could be presented by a vector of corresponding residues. The remainder of first level is denoted by $(r_1, r_2, r_3, \dots, r_n)$ and the residues of second level for r_i can be denoted by $(r_{i1}, r_{i2}, r_{i3}, \dots, r_{in_i})$ $i = 1, 2, 3, \dots, n$.

By considering two-Level RNS, the arithmetic operations "$\circ$" such as addition, subtraction and multiplication can be applied to the second level residues as follow.

$$\{z_{i1}, z_{i2}, z_{i3}, \dots, z_{in_i}\} = \{x_{i1}, x_{i2}, x_{i3}, \dots, x_{in_i}\} \circ \{y_{i1}, y_{i2}, y_{i3}, \dots, y_{in_i}\}$$

Where $z_{ij} = (x_{ij} \circ y_{ij}) \bmod m_{ij}$, $i = 1, 2, 3, \dots, n$, $j = 1, 2, 3, \dots, n_i$ [11].
Therefore, the arithmetic operations can be carried out on very small remainders parallely where the calculations of RSA could be applied to small numbers in the second level, which speed up execution of the algorithm.

Cryptosystem RSA

RSA is one of the asymmetric cryptography algorithms which exploits both public and private keys. There are two private keys in sender and receiver side in RSA which are primes and the multiplication of both keys generates the public key. Each of the private key should have a length about 1024 bit at least. The main idea of RSA is the easy multiplication of two numbers which generates the public one, but factorizing a public key into two prime operands is very difficult.

The algorithm can be implemented in 3 steps[1]:

First step is choosing the keys as follow: i. Both sender and receiver select a prime number which denoted by p and q as private keys. ii. Calculate n=p × q and φ= (p-1) × (q-1). iii. Choose random number e as public key in the sender side where 0 < e < φ are wise prime with φ. iv. Select a private key named d where $e \times d \overset{\Phi}{\equiv} 1$.

Second step is the encryption in sender, which involves in the following steps: i. Get the message M that should be transmitted from sender, Where M is an integer less than n. ii. M is encrypted by computing $C = M^e \bmod n$. iii. Sender transmits C to receiver.

The last step is applied by the receiver decrypting the cipher text to plain text by using its own private key. Following steps are applied by the receiver: i. Consider the message C encrypted in sender. ii. Receiver calculates $M = C^d \bmod n$. iii. M is the plain text encrypted by the sender side which is decrypted by receiver now.

Since C and M are less than n and also this algorithm uses multiplication widely, thus it is suitable to be implemented in RNS to speed up computation.

Designing a secure and reliable communication by using Two-Level RNS

In this paper, to achieve both secure and reliable communications, using Multi-Level RRNS is proposed leading speeding up RSA calculations. Also since redundant module is used in RRNS, it can detect and correct the errors in the modules.

The scheme is shown in figure 2. As it is seen, the redundant modules are used in both level of the system. It is worth to note that only the data in the second level are transmitted in network communication. Therefore, it is sufficient to apply RSA calculation only on the remainder of the second level. On the other hand, to achieve fault tolerant in the system, redundant modules are used.

$$\begin{tikzcd}[column sep=tiny]
& \{\underbrace{r_1,r_2,\dots,r_n}_{Original\ Residue},\underbrace{r_{n+1},r_{n+2},\dots,r_{n+k}}_{Redundant}\} \arrow[dl] \arrow[dr] & \\
\{\underbrace{r_{11},r_{12},\dots,r_{1m}}_{Original\ Residue},\underbrace{r_{1m+1},r_{1m+2},\dots,r_{1m+j}}_{Redundant\ of\ second\ level}\} & & \{\underbrace{r_{(n+k)1},r_{(n+k)2},\dots,r_{(n+k)m'}}_{Original\ Residue},\underbrace{r_{(n+k)m+1},r_{(n+k)m+2},\dots,r_{(n+k)m+j'}}_{Redundant\ of\ second\ level}\}
\end{tikzcd}$$

Figure-2
Two-Level RRNS

As mentioned before, data in the second level are transmitted. Therefore, the cipher text is decrypted to plain text by using the decryption algorithm first. Then, in the next step, error detector is used to determine modules with error. Error detection is applied just for the data in the second level residues. If there was an error, this remainder marked as faulty in the first level and thus it is not valid. Afterward, if the numbers of faulty remainders are not greater than K (number of redundant modules in the first level) error corrector could be used to correct the errors.

Example: Consider the main module in the first level as {35, 36} with redundant module {42}. Therefore, the dynamic range of this module set is 1260.

The module set in the second level is used according to figure 3.

$$\begin{tikzcd}[column sep=small]
& \{\underbrace{35,36}_{Original} \arrow[dl] \arrow[d] & \mid \underbrace{42}_{Redundant}\} \arrow[d] \\
\{5,7\mid 8\} & \{4,9\mid 10\} & \{6,7\mid 8\}
\end{tikzcd}$$

Figure-3
Modules of Two-Level RRNS (for example)

Let the sender wants to send m=1000. Then as it is shown in figure 4, the residues are computed.

$$\begin{tikzcd}[column sep=small]
& \{20,28 \arrow[dl] \arrow[d] & \mid 34\} \arrow[d] \\
\{0,6\mid 0\} & \{0,1\mid 8\} & \{4,6\mid 2\}
\end{tikzcd}$$

Figure-4
Residue for 1000 in Two-Level RRNS

To have a secure communications, the sender should encrypt the data in the second level by using RSA.

The requirements are listed as below: i. P=3, q=11. ii. $n = p \times q = 3 \times 11 = 33$ iii. $\Phi(n) = (p-1) \times (q-1) = 20$
iv. $k = 7, k^{-1} = 3$

Therefore, the sender computes

$m = 0 \Rightarrow C = 0^7 \bmod 33 = 0$
$m = 1 \Rightarrow C = 1^7 \bmod 33 = 1$
$m = 2 \Rightarrow C = 2^7 \bmod 33 = 29$
$m = 4 \Rightarrow C = 4^7 \bmod 33 = 16$
$m = 6 \Rightarrow C = 6^7 \bmod 33 = 30$
$m = 8 \Rightarrow C = 8^7 \bmod 33 = 2$.

Thus, the sender sends { 0,30,16,0,1,2,16,30,29} to the receiver.

Suppose that there is an error which changes the data in the network to {0, 30, 16, 1, 1, 2, 16, 30, 29}.
Receiver receives this data and begins to decrypt it with its own private key as follow:

$C = 0 \Rightarrow m = 0^3 \bmod 33 = 0$
$C = 1 \Rightarrow m = 1^3 \bmod 33 = 1$
$C = 2 \Rightarrow m = 2^3 \bmod 33 = 8$
$C = 16 \Rightarrow m = 16^3 \bmod 33 = 4$
$C = 29 \Rightarrow m = 29^3 \bmod 33 = 2$
$C = 30 \Rightarrow m = 30^3 \bmod 33 = 6$

Therefore, data which the receiver gets, will be: { 0,6,4,1,1,8,4,6,2}.

In the next step, receiver should try to detect the errors in each module. Compute each module by using remainders of main module as follow:

$$\{0_5, 6_7\} \Rightarrow re\min der1 = 20$$
$$\{1_4, 1_9\} \Rightarrow re\min der2 = 1$$
$$\{4_6, 6_7\} \Rightarrow \mathrm{Re}\,dundant = 34$$

Then, the remainder of each module set is computed with respect to the corresponding redundant module which is compared with transmitted number.

$$20 \bmod 8 = 4, 4 = 4 \checkmark$$
$$1 \bmod 10 = 1, 1 \neq 8 \times$$
$$34 \bmod 8 = 2, 2 = 2 \checkmark$$

The receiver discovers error in the second module by these computations. Therefore just the main numbers resulted from first and second modules are correct and thus are transmitted to higher level. Since there is a redundant module in the first level of module, it can be used to correct the data by knowing the location of the error.

Thus, it is sufficient to compute:

$X = \{20_{35}, 34_{42}, 1_4\}$ There is not such X.

$$X = \{20_{35}, 34_{42}, 1_{9_{36}}\} \Rightarrow X = 370{,}1000$$
$$X = \{20_{35}, 34_{42}, 8_{10_{36}}\} \Rightarrow X = 1000$$

By using the first and third module, in the first level and thus each modules with error in the second level is computed resulting.

Comparison: Previous and common methods for detection and correction of the data along with encrypting them are compared in this section. To be able to detect and correct data in numerical computations, the bit redundant should be added to the main number which is greater than the main obtained number. Thus, the computation time in cryptography action could be high.

The proposed method is a new version of the method appeared in Residue Number System Code[11] which is faster in correcting the data.

Conclusion

In this paper to achieve both reliability and security in communications, RRNS is used by RSA cryptography system. To decrease the complexity of RSA calculations, RRNS is applied in Two-Level RNS; therefore the numbers are so much smaller than the original keys, so that the speed of calculations increases extraordinary and the security guaranteed.

References

1. Shankar Dhakar R., Kumar Gupta A. and Sharma P., Modified RSA Encryption Algorithm (MREA), Second International Conference on Advanced Computing & Communication Technologies **(2012)**
2. Bajard J. and Imbert L., A Full RNS Implementation of RSA, *IEEE Transactions on Computers*, **53(6)** **(2004)**
3. Piestrak J., A high-speed realization of a residue to binary number system converter, IEEE Transactions on Circuits and Systems – II Analog and Digital, *Signal Processing*, **42(10)**, 10, **(1995)**
4. Ramirez J., Garcia A., Lopez-Buedo S. and Lloris A. RNS-enabled digital signal processor design, *Electronics Letters*, **38(6)** **(2002)**
5. Soderstrand M.A. and Eds A., Residue Number System arithmetic: modern applications in digital signal processing, New York, IEEE Press **(1986)**
6. Safi S.M., Rashno M., Abedi P., Kaboli M. and Fatemeh S.S, An Efficient Residue to Binary Converter for the New Two-Level Moduli Set{22n {2n ,2n+1 -1}, 2n -1, 2n +1}, *Journal of Research Journal of Recent Sciences,* **1(7),** 83-86 **(2012)**
7. Cardarilli G.C., Nannarelli A. and Re M., Residue number system for low-power DSP applications, in Proc. 41st IEEE Asilomar Conference Signals, System, Computer **(2007)**
8. Wei W. and et al., RNS application for digital image processing, in Proc. 4th IEEE International Workshop System on-Chip Real-Time Application **(2004)**
9. Pham D.M., Premkumar A.B. and Madhukumar A.S, Error Detection and Correction in Communication Channels Using Inverse Gray RSNS Codes, *IEEE Transactions on Communications*, **59(4)**, **(2011)**
10. Navi K., Molahosseini A.S., and Esmaeildoust M., How to Teach Residue Number System to Computer Scientists and Engineers, *IEEE Transactions on Education*, **54(1)**, **(2011)**
11. Hosseinzadeh M., Reshadi M., Khademzadeh A. and Navi K., Reliable and Secure Chip Level Communication By Residue Number System Code, *Journal of integrated design & process science*, **12** **(2008)**