Organization of Secured Data Transfer in Computers Using Sign-Value Notation

Shamil Magomedov[1, *]

[1] Moscow Technological University (MIREA), 119454 Moscow Russia

Abstract. In this paper, another approach to organization of secured data transfer between processor and random access memory is proposed, which includes two peculiarities: the use of residue number system (RNS) instead of standard positional notation as the basis of the closing data and an approach to organization of transfer process excluding the use of special subsystems within the framework of security system. With such an approach to the transferred data security process, there is no need for the subsystems mentioned above, what deprives potential attackers of opportunities to select the dispatching center as an object of attack and thereby increases data transfer process security.

1 Introduction

Traditional technology of restricted access information protection usually involves the use of encryption techniques. However, the requirements for such systems, are rather strict, and as a consequence, are bulky in the implementation and expensive in operation, what often makes it barely acceptable and cumbersome to use. The following is a procedure for the closure of the information, based on the use of residue number system. In addition, the traditional protection of the data exchange circuit lines as a necessary stage typically involves the distribution of a given task encryption key between the participants of data exchange, and when data transfer occurs, they are encrypted (or subscribed, in the case of an electronic signature) with this key. Control over the use of keys, update, formation and key distribution over existing technology, should be exercised by special control center in the system, in particular a special subsystem (or utility) in the interaction between the processor and RAM. Since all key information, including ciphers, concentrated in the center, the center becomes a potential target for malicious attacks. Due it is necessary to make special efforts to protect this center, what is a quite complex and costly task. The specifics of the problem as applied to data exchange between the CPU and RAM (this problem is the subject of our studies) is, in particular, to use as a basis RNS data closure technology as in [1–3], we consider processors, in which RNS is the basis of computer processor. That is why the use of RNS as a basis for the closure of these technologies in the process of exchange is seen as a natural extension of technologies perform computing operations in the processor.

We did not manage to find any works on the subject of the use of RNS as data closure technology basics. The closest one is our work, [1] where we describe the procedure of closing the data into the RAM, and [2] on the use of RNS in the process of data exchange between remote entities. Close to the procedure in question is the transferred data protection organization while the network sharing on the basis of frequent change of encryption keys is considered in [4].

2 The change frequency of the residue number system base

The implementation of the data transfer protection concept through the use of RNS, which is described in the introduction, at the stage of its formation [1], raises the question of the frequency of updates and even complete replacement of the base of RNS used for closing the data. Frequent change of RNS will require significant expenditure of computing resources of the processor, which is undesirable. In a rare change of RNS base, the risk of opening a base and, as a result, unauthorized entry into the data system significantly grow up. Consequently, there is an optimal value of the interval between consecutive shifts inception. As a result, the following problem arises: how often should I change the base of RNS so that the overall costs are minimized?

Below is a formalized model, in which the solution of a search task of the optimal value of the interval between successive moments of the key changes is offered. We analyze the problem of changing the base of RNS based on the construction of a formalized model of key changes and solutions, within the framework of this model, of the base change frequency task. Here is a formalized description of the problem.

As an optimality criterion we take the chance of theft of keys (i.e. the base of RNS) during a time interval not exceeding the interval between successive moments of

* Corresponding author: petrovich4you@gmail.com

change or update of RNS base (let's call this period an update period), minus the minimum time required to activate this key. We consider first, on what factors the probability of theft of the key within the task depends.

The probability of theft primarily depends on the strength of the closing procedure of the data (encryption) that, in turn, is determined by the number of numbers at the base and by their size: the larger the numbers of the base and the larger their values are, the generally more resistant will be the encryption procedure. Furthermore, among the numbers of the RNS base, there should not be relatively small ones, i. e. all numbers of the base should be approximately the same length. However, the closeness of the individual values may be regarded as a procedural parameter. We note that, typically, herewith, encryption time disproportionately increases.

Thus, the basic parameters of the model are:

a) the quantity of prime numbers included in the base of RNS;

b) the number of numbers in the base of RNS;

c) the minimum time required to restore data if there is the base of RNS;

d) the boundaries of the most preferred areas of possible values of prime numbers in the base of RNS;

e) the period of time of an key upgrade.

One of the basic parameters that must be evaluated in terms of the model, is the degree of resistance of these private keys. In this regard, first we describe the basic concept of the function that evaluates the stability of the system key. Evaluation of resistance should include an assessment of n key length and evaluation of resistance of each number in P_i in the base of RNS. To simplify evaluation, we shall restrict additive estimates according to preset settings of the key. Further, resistance numbers P_i are proposed to assess by means of functions that satisfy the following heuristic conditions conducing to improvement of resistance of RNS.

1. The function should give minimum values for P_i estimates beyond a certain value zone. The lower limit of the zone is determined by the minimum acceptable value of P_i – at lower values the probability of opening P_i greatly increases. The upper limit of the zone is determined by the computing power of the processor and the share of the allowable time for closing the data during processing – this share is determined while designing the processor.

2. Within the area of values, distribution of P_i values at full replacement of one RNS to another one may be determined by linear (horizontal) function, which corresponds to the equiprobable choice of any of the values included in this zone. However, if the numbers at the base of RNS are replaced partially, but not fully, it is desirable (from the perspective of the current data processing) by replacing, wherever possible, to maintain approximately the length of the replaced number and replacing one of the P_i numbers. In this case, a uniform distribution is not appropriate, in connection with what is proposed to use functions with data accumulation area, namely multi-peak functions. The following is proposed for this purpose, namely two-peak function class.

3. You must have parameters that define the degree of blur, data uncertainty *in separate ones* around each peak of multi-peak function. This will be an additional option to increase the resistance, making it difficult, at least acceptable, for an attacker to limit the enumeration area of possible key variants.

4. It is desirable to have a simpler function (to reduce the amount of computations when evaluating resistance) and relatively smooth (for computational stability).

With regard to function, which evaluates the length of a key resistance, it is possible to formulate similar requirements.

In view of the above, we propose the following formula for the initial evaluation of the resistance of the selected base of RNS as an encryption key:

$$S = S(\pi(n, \vec{P})) = \varphi(n) + \sum_{i=1}^{n} f(P_i), \qquad (1)$$

where

$$f(P) = y(P, a_1, b, c, d, \sigma, r) + y(P, a_2, b, c, -d, \sigma, r),$$
$$\varphi(P) = y(P, A, B, C, 0, s, R),$$

$$y(P, a, b, c, d, s, r) = \frac{a}{1 + b(P - (c - c \cdot d))^k}\left(1 - e^{-r \cdot \max(0; P - s)}\right) \qquad (2)$$

and either $k = 2$ or $k = 4$, and all the coefficients are positive numbers.

Let us clarify the content of all the coefficients included in the definition of the functions $f(P)$, $\varphi(n)$ and $y(P, a, b, c, d, \sigma, r)$. Function $y(P, a, b, c, d, \sigma, r)$, and together with it the function $\varphi(P)$ for any positive values of coefficients are one-peak ones. In particular, Fig. 1 shows an example of the function $\varphi(P)$ with the following parameter values: $A = 3$; $B = 0.01$; $C = 7$; $s = 3$; $R = 1$, $k = 4$.

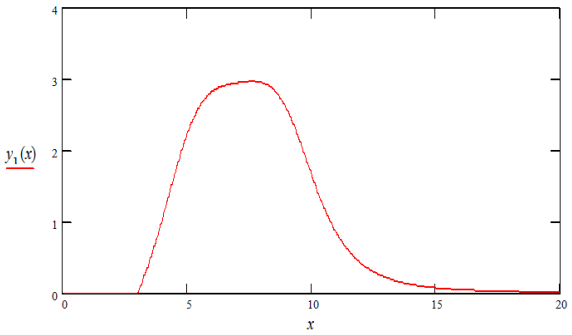

Fig. 1. Example of the function $\varphi(P)$ graph

A is a scale parameter characterizing the maximum value of the function $\varphi(P)$ and is selected at random, being based on reasons of clarity; B is determined by the degree of "stretch" of the function $\varphi(P)$ – the more is B, the more the function $\varphi(P)$ is one-peak, and the narrower is the range of permissible effective (i.e., the most frequent ones) values of the variable x (= n); in the example of Fig. 1, B is selected so that the effective value interval of n key length range from 3 to 15; C defines the central importance of n key length (in this case n = 10), around which the effective range of values

is formed; s is the lower limit of the permissible n key length values; the example imposes limitation – the number of integers in the base of RNS should be less than 3 (corresponding to $s = 3$); R is intended to provide relative smoothness of the point $n = s$ (more precisely, to reduce the size of the jump function at the boundary point $x = s$); with an increase in R at a point $x = s$, the jump of k graphics increases; k firstly determines slope of graph rises to the top (left and right), and secondly, the length of the upper, relatively flat top of the graph – with $k = 4$, the upper part is longer and rises abruptly at $k = 2$, the function is more peaked.

With respect to the function $f(P)$, there is another parameter – d, and the parameter a is replaced into the other two – a_1 and a_2. We explain these parameters. We note that, as can be seen from the examples below showing graphs of the function $f(P)$, this function is two-peak ("two-hump") in the most interesting cases for us. The distance between the peaks is determined by $2cd$, and therefore, d is defined (for fixed c) by the degree of hump blur. Consequently, the parameter d is always less than 1; it is possible to estimate the percentage: how you should dilute the zone of the effective values of the variable x around its center c. Parameters a_1 and a_2 define the maximum value of each hump – of both the left and the right one.

It should be noted that the formula (1) and the function (2) satisfy the conditions listed above 1–3. At the same time, changing the parameter b, you can obtain the function with the desired depth of the pit between the humps, until the case of absence of the pit, and even with the tab instead of the pit (for small values of b). Regarding the choice of the parameter, we specify the following: the value of $k = 4$ is more preferable that provides a greater blur to the data in each hump, but at the same time it requires a larger number of calculation, what is essential while multiply (massively) calculating. To illustrate the positions of things said lower (in Fig. 2), an example of the function $fi(P)$ with $k = 2$ is brought; the rest parameters have the same values. Parameter values: $a_1 = 3.5$; $a_2 = 3$; $b = 10^{-6}$; $c = 400$; $d = 0.3$; $r = 0{,}005$; $s = 500$.

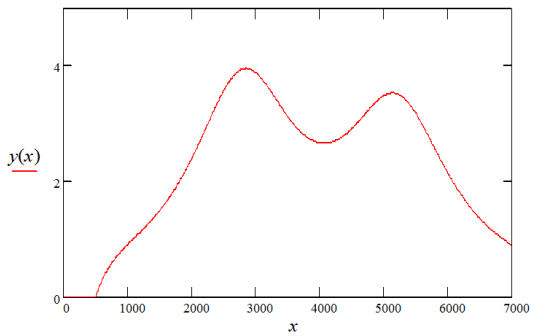

Fig. 2. Example of the evaluation function with $k = 2$

The function $f(P)$ describes the degree of unexpectedness (even unforeseeability) for an attacker to find the number P as part of the base of RNS: small values P are unlikely, because they are less resistant to cracking; larger values P are unlikely as well, because they greatly increase the time required for data processing when used as part of RNS; median values are also increasingly expected by the attacker, since in the median values, values of bases are generally arranged, and they are optimal for compromise requirements simultaneously to the key durability and acceptability of data processing time. The graph in Fig. 2 satisfies the mentioned intuitive reasons, so it is for these reasons, the coefficient S is considered above as the key resistance evaluation.

Based on the selection index S as an indicator of resistance, we can offer a key to select the key by accident, in accordance with its resistance, i.e., the probability of selecting the key $\pi(n, \vec{P})$ is considered equal to normalized value S.

$$q(\pi(n, \vec{P})) = \frac{S(\pi(n, \vec{P}))}{\sum_{\pi(n,\vec{U})} S(\pi(n, \vec{U}))}, \qquad (3)$$

where the sum in the denominator is taken over keys $\pi(n, \vec{U})$ so that values of parameters n and P_i are in effective areas of these parameters. In order to prevent re-use of the key as well as the keys close to it, it is proposed, until the use of formula (3) and before the key update, to multiply the function $f(P)$ on the expression

$$\theta(P) = \prod_{i=1}^{n} (1 - y(P, 1, \Delta, P_i, 0, s, R).$$

The graph of function $1 - y(P, 1, \Delta, P_i, 0, s, R)$ (let's call it the excluding function) at $\Delta = 10^{-7}$, $P_i = 4000$, $s = 500$ and $R = 1$ is shown in Fig. 3.

Fig. 3 shows that the function $y(P, 1, \Delta, P_i, 0,)$ is almost everywhere equal to 1, and only in a narrow band about $1/(P_i \cdot \Delta) \approx 250$ wide it abruptly goes down to zero at $P = P_i$, and then just as abruptly rises. Thus, by multiplying the function $f(P)$ on $y(P, 1, \Delta, P_i, 0,)$ at the point $P = P_i$, the function $f(P)$ falls to zero, thereby excluding the value $P = P_i$ and all close to it values chosen at random while choosing the next value P_j based on the distribution density $q(\pi(n, \vec{P}))$.

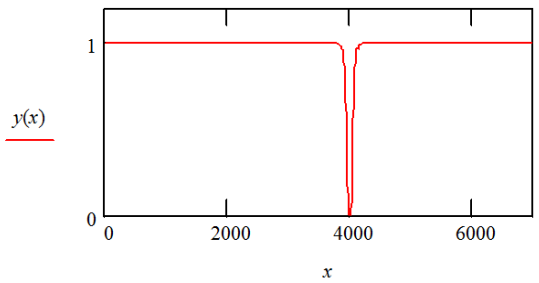

Fig. 3. Graph of excluding function

We should note that the probability $q(\pi(n, \vec{P}))$ can be regarded as the probability of key resistance, i.e. non-disclosure of key for the routine time T. Then, in the first approximation, we can assume that $\Pr(\pi(n, \vec{P}), T) = 1 - q(\pi(n, \vec{P}))$. Assuming that the

dependence of the probability of key non-disclose on time is exponential, i.e., $1-\Pr(\pi(n,\vec{P}),t)=\alpha\cdot e^{-\beta t}$ where α and β are constant, considering $\Pr(\pi(n,\vec{P}),0)=0$ and $\Pr(\pi(n,\vec{P}),T)=1-q(\pi(n,\vec{P}))$, we obtain the following relationship: $\alpha=1$ and $q(\pi(n,\vec{P}))=e^{-\beta\cdot T}$, from where we conclude $e^{-\beta}=\left(q(\pi(n,\vec{P}))^{1/T}\right)$ and

$$\Pr(\pi(n,\vec{P}),t)=1-\left(q(\pi(n,\vec{P}))\right)^{t/T}. \qquad (4)$$

To construct the model, we introduce the following notation: $\Pr(\pi(n,\vec{P}),t)$ is the likelihood of key disclosure, i.e., the base of RNS, at a time *t*; $C(\pi(n,\vec{P}))$ are costs of resources (especially, of time) on the formation of the key; $\tau(\pi(n,\vec{P}))$ is the time required for the activation key; $L(\pi(n,\vec{P}))$ is average loss because of disclosure of the key $\pi(n,\vec{P})$ in the exchange of data; $D(\pi(n,\vec{P}))$ are costs of key $\pi(n,\vec{P})$ update; *T* is routine during operation of the computer while data processing, taking into account the repeatability of data processed (for example, month, week, day); λ is the intensity of the exchange of data between the processor and RAM; δ is the interval between successive moments of the key update; *N* is the maximum acceptable amount of numbers at the base of RNS.

$$x(\pi(n,\vec{P}))=\begin{cases}1, \text{ if key } \pi(n,\vec{P});\\ 0 \text{ otherwise.}\end{cases}$$

We estimate the total loss associated with the use of a given set of encryption keys. During the routine time *T,* the average number of key updates is equal to T/δ. Then the average total costs associated with the upgrade key for the routine period are:

$$\rho_1=\frac{T}{\delta}\sum_{n=s}^{N}\left(\sum_{\pi(n,\vec{P})}(1-q(\pi(n,\vec{P}))C(\pi(n,\vec{P}))x(\pi(n,\vec{P}))\right).$$

The second group of losses is associated with the key opening in the transfer process, resulting in losses. Since the probability of disclosing key is usually a very small value, the probability that during key life more than one key disclosure happens is practically zero. Assuming that the time of the attacker invasion on the network node is a random, it is natural to assume that the average time that a malicious user takes to use the disclosed key is equal to $\delta/2$. Then, for the magnitude of the losses associated with the disclosure of the key, in view of (4), we can write the expression

$$\rho_2=\frac{T}{\delta}\sum_{n=s}^{N}\left(\sum_{\pi(n,\vec{P})}(1-\left(q(\pi(n,\vec{P}))^{\delta/(2T)}\right))L(\pi(n,\vec{P}))x(\pi(n,\vec{P}))\right)$$

Finally, the last group of the losses is related to costs to update the keys. Since the number of updates per routine period is on the average equal to T/δ, the total cost for the routine period of update of keys on the average are:

$$\rho_3=\frac{T}{\delta}\sum_{n=s}^{N}\left(\sum_{\pi(n,\vec{P})}(1-\left(q(\pi(n,\vec{P}))^{\delta/(2T)}\right))D(\pi(n,\vec{P}))x(\pi(n,\vec{P}))\right).$$

Based on the obtained relations, we derive the following expression for the total costs and losses for the routine period of *T*:

$$w(T)=\rho_1+\rho_2+\rho_3=\frac{T}{\delta}\sum_{n=s}^{N}\left(\sum_{\pi(n,\vec{P})}\left[\left(1-\left(q(\pi(n,\vec{P}))^{\delta/(2T)}\right)\right)\times\right.\right.$$
$$\times\left(L(\pi(n,\vec{P}))+D(\pi(n,\vec{P}))\right)+$$
$$\left.\left.+(1-q(\pi(n,\vec{P}))C(\pi(n,\vec{P}))\right]\right)x(\pi(n,\vec{P})).$$

Since the key is always one (single), then the equality is relevant

$$\sum_{n=s}^{N}\left(\sum_{\pi(n,\vec{P})}x(\pi(n,\vec{P}))\right)=1. \qquad (5)$$

On the basis of the obtained relations, a task of selecting the optimum number of numbers in the base of RNS, and the composition of these bases can be formalized: to select key *n* length, the number $P_i(\overline{i=1;n})$ and the period of key update τ so that the total costs *w*(*T*) are minimal, i.e. the equality (5) holds.

$$w(T)\to\min \text{ on } x(\pi(n,\vec{P}))$$
$$\text{such that } x(\pi(n,\vec{P}))=0\vee 1. \qquad (6)$$

In theoretical terms, the task (6) is a classical problem, which developed its own specific methods of solution [7]. Without consideration of this statement, the problem of partial update of RNS base remains.

One of the most difficult problems associated with the implementation of the above-described procedure for updating the key of closing the data using the RNS is the task of choosing the model parameters, in particular, parameters of the functions $\varphi(P)$ and $f(P)$. The solution of this problem is advisable to implement being based on the use of expert procedures.

This work was partially supported by motivational payments system faculty MIREA.

3 Conclusions

The analysis is carried out and a set of parameters that critically affect the efficiency of data security while processing in CPU using RNS is formed.

The model of the task of choosing the optimal interval of updating RNS base, a solution of which will minimize the total costs associated with the malicious theft of keys of closing the data, as well as the cost of resources for key update, is formed.

References

1. S. Magomedov, Vestnik of Astrakhan State Technical University. Series: Management, Computer Science and Informatics, **1**, 22-32 (2015)
2. S. Magomedov, Vestnik of Astrakhan State Technical University. Series: Marine engineering and technology, **2**, 44-46 (2010)

3. S. Magomedov, Caspian magazine. Management and high technology, **4**, 118-125 (2013)
4. V. Laptev, G. Popov, Vestnik of Astrakhan State Technical University. Series: Management, Computer Science and Informatics, **2**, 94-98, (2012)
5. N. Worms, S. Ryadnov et al. *Modular structure neuroprocessor parallel computing systems* (FIZMATLIT, 2003)
6. A. Omondi, B. Premkumar, *Residue number systems: theory and implementation Advances in Computer Science and Engineering* (London, Imperial College Press, 2007)
7. M. Belkin, S. Coogee, A. Sigov, Russian Journal of Technology, 1, 4-21 (2016)