computing@tanet.edu.te.ua
www.computingonline.net

ISSN 1727-6209
International Journal
of Computing

# THE GENERAL CHINESE REMAINDER THEOREM

**Sorin Iftene [1], Florin Chelaru [2]**

[1] Faculty of Computer Science, "Al. I. Cuza" University, Iasi, Romania
e-mail: siftene@infoiasi.ro
[2] Faculty of Computer Science, "Al. I. Cuza" University, Iasi, Romania
e-mail: florin.chelaru@infoiasi.ro

**Abstract:** *The Chinese remainder theorem deals with systems of modular equations. The classical variant requires the modules to be pairwise coprime. In this paper we discuss the general variant, which does not require this restriction on modules. We have selected and implemented several algorithms for the general Chinese remainder theorem. Moreover, we point out some interesting applications of this variant in secret sharing and threshold cryptography.*

**Keywords:** *The Chinese remainder theorem, secret sharing, threshold cryptography.*

## 1. INTRODUCTION AND PRELIMINARIES

The Chinese remainder theorem deals with systems of modular equations. It has many applications in computer science (see, for example, [1]). We only mention the *RSA* decryption algorithm proposed by Quisquater and Couvreur [2], the discrete logarithm algorithm proposed by Pohlig and Hellman [3] and the algorithm of recovering the secret in the Mignotte's threshold secret sharing scheme [4] or in the Asmuth-Bloom threshold secret sharing scheme [5]. The classical variant requires that modules be pairwise coprime. In this paper we discuss the general variant, which does not require this restriction on modules.

The paper is organized as follows. The rest of this section is dedicated to some preliminaries on number theory, focusing on the Chinese remainder theorem. We survey the most important algorithms for the general Chinese remainder theorem in Section 2. Implementation details and test results are discussed in the next section. Some interesting applications of the general variant of the Chinese remainder theorem are presented in Section 4. Our conclusions are presented in the last section.

We present next some basic facts on number theory. For more details, the reader is referred to [6] and computational aspects can be found in [7].

Let $a, b \in \mathbf{Z}, b \neq 0$. The *quotient* of the integer division of $a$ by $b$ will be denoted by $a \operatorname{div} b$ and the *remainder* will be denoted by $a \bmod b$. Moreover, in case that $a \bmod b = 0$ we shall say that $b$ is a *divisor* of $a$ or $b$ *divides* $a$ and we shall use the notation $b \mid a$.

Let $a_1,\ldots,a_n \in Z, a_1^2 + \cdots + a_n^2 \neq 0$. The *greatest common divisor (gcd)* of $a_1,\ldots,a_n$ will be denoted by $(a_1,\ldots,a_n)$. We say that $a_1,\ldots,a_n$ are *coprime* if $(a_1,\ldots,a_n) = 1$. It is well-known that there exist $\alpha_1,\ldots\alpha_n \in \mathbf{Z}$ which satisfy

$$\alpha_1 a_1 + \cdots + \alpha_n a_n = (a_1,\ldots,a_n)$$

(the linear form of the *gcd*).

Let $a_1,\ldots,a_n \in \mathbf{Z}$ such that $a_1 \cdots a_n \neq 0$. The *least common multiple (lcm)* of $a_1,\ldots,a_n$ will be denoted by $[a_1,\ldots,a_n]$.

Let $a,b,m \in \mathbf{Z}$. We say that $a$ and $b$ are *congruent modulo* $m$, and we use the notation $a \equiv b \bmod m$ or $a \equiv_m b$, if $m \mid (a-b)$. $\mathbf{Z}_m$ denotes the set $\{0,\ldots,m-1\}$ for any $m \geq 2$.

We shall present next the general variant of the Chinese remainder theorem:

**Theorem 1.** ([8]) *The system of equations*

$$\begin{cases} x \equiv b_1 \bmod m_1 \\ \quad \vdots \\ x \equiv b_k \bmod m_k \end{cases} \qquad (1)$$

*has solutions in* $\mathbf{Z}$ *if and only if*

$$b_i \equiv b_j \bmod (m_i, m_j)$$

*for all* $1 \leq i, j \leq k$. *Moreover, if the above system of equations has solutions in* $\mathbf{Z}$, *then it has an*

*unique solution in* $Z_{[m_1,\ldots,m_k]}$.

The direct implication can be easily proven as follows. Assume first that $x_0$ is an integer solution of the system (1) and let $1 \le i, j \le k$. The relation $x \equiv b_i \bmod m_i$ implies that $x \equiv b_i \bmod (m_i, m_j)$. Moreover, by the same argument, we have also $x \equiv b_j \bmod (m_i, m_j)$ which leads, by transitivity, to $b_i \equiv b_j \bmod (m_i, m_j)$.

The uniqueness modulo $[m_1,\ldots,m_k]$ can also be easily proven. Let $x_0$ and $y_0$ be two solutions for the system (1). By transitivity we obtain that $x_0 \equiv y_0 \bmod m_i$, for all $1 \le i \le k$, which implies that $(x_0 - y_0)$ is a multiple of $m_1,\ldots, m_k$, which leads to $[m_1,\ldots,m_k] \mid (x_0 - y_0)$.

The converse implication will be discussed in Section 2.

An important particular case is the case $(m_i, m_j) = 1$ for all $1 \le i < j \le k$. In this case, for any integers $b_1,\ldots,b_k$, the system of equations

$$\begin{cases} x \equiv b_1 \bmod m_1 \\ \quad \vdots \\ x \equiv b_k \bmod m_k \end{cases}$$

has an unique solution in $\mathbf{Z}_{m_1 \cdots m_k}$.

## 2. ALGORITHMS FOR THE GENERAL CHINESE REMAINDER THEOREM

In this section we will present the most important algorithms for the general variant of the Chinese remainder theorem.

### 2.1 ORE'S ALGORITHM

Ore [8] has proposed an interesting proof for the general Chinese remainder theorem.

Let $m = [m_1,\ldots,m_k]$ and $c_i = \dfrac{m}{m_i}$, for $1 \le i \le k$.

*Claim 1*: $(c_1,\ldots,c_k) = 1$.

We first choose

$$D = \{p \ prime \mid (\exists 1 \le i \le k)(p \mid m_i)\}.$$

Thus, every element $m_i$ can be written as

$$m_i = \prod_{p \in D} p^{e_{p,i}}$$

(with $e_{p,i} = 0$ if $p$ does not divide $m_i$), and

$$[m_1,\ldots,m_k] = \prod_{p \in D} p^{\max(e_{p,1},\ldots,e_{p,k})},$$

$$c_i = \prod_{p \in D} p^{\max(e_{p,1},\ldots,e_{p,k}) - e_{p,i}}.$$

For every $p \in D$, there is an $i$, $1 \le i \le k$ such that $e_{p,i} = \max(e_{p,1},\ldots,e_{p,k})$. In this case, $p$ does not divide $c_i$ which implies that for every possible common prime divisor $p$ there is an element $i, 1 \le i \le k$, such that $p$ does not divide $c_i$ which eventually leads to $(c_1,\ldots,c_k) = 1$.

Thus, there are $\alpha_1,\ldots,\alpha_k \in \mathbf{Z}$ that satisfy $\alpha_1 c_1 + \cdots + \alpha_k c_k = 1$.

*Claim 2*: $\dfrac{m_i}{(m_i, m_j)} \mid c_j$, for all $1 \le i, j \le k$.

We have that

$$\frac{m_i}{(m_i, m_j)} = \prod_{p \in D} p^{e_{p,i} - \min(e_{p,i}, e_{p,j})}$$

and so, the relation $\dfrac{m_i}{(m_i, m_j)} \mid c_j$ is equivalent to

$$e_{p,i} - \min(e_{p,i}, e_{p,j}) \le \max(e_{p,1},\ldots,e_{p,k}) - e_{p,j}$$

for every $p \in D$, which clearly holds true.

*Claim 3*: The value

$$x_0 = (\alpha_1 c_1 b_1 + \cdots + \alpha_k c_k b_k) \bmod m$$

is a solution of the system of equations (1).

Let *i* be an arbitrary element, $1 \le i \le k$. For every $1 \le j \le k$ we have

$$b_j \equiv_{(m_i, m_j)} b_i \Rightarrow c_j b_j \equiv_{c_j (m_i, m_j)} c_j b_i$$
$$\Rightarrow c_j b_j \equiv_{m_i} c_j b_i \ (\text{by Claim 2})$$
$$\Rightarrow \alpha_j c_j b_j \equiv_{m_i} \alpha_j c_j b_i$$

If we sum the last congruencies, for $1 \le j \le k$, we obtain

$$\alpha_1 c_1 b_1 + \cdots + \alpha_k c_k b_k \equiv_{m_i} b_i (\alpha_1 c_1 + \cdots + \alpha_k c_k)$$

which finally leads to $x_0 \equiv b_i \bmod m_i$.

This demonstration leads to Ore's algorithm:

**CRT_Ore($b_1, \ldots, b_k, m_1, \ldots, m_k$)**

**Input**: $b_1,\ldots,b_k, m_1,\ldots,m_k \in \mathbf{Z}$ such that $b_i \equiv b_j \bmod (m_i, m_j), \forall 1 \le i < j \le k$;

**Output**: $x_0$, the unique solution

modulo $[m_1,\ldots,m_k]$ for the system (1);

**begin**

1. $m = [m_1,...,m_k]$;
2. for $i := 1$ to $k$ do $c_i = \frac{m}{m_i}$;
3. find $\alpha_1,\ldots,\alpha_k \in Z$ such that $\alpha_1 c_1 + \cdots + \alpha_k c_k = 1$;
4. $x_0 = \sum_{i=1}^{n} \alpha_i \cdot c_i \cdot b_i \bmod m$;

**end**.

Gauss [9] has described a similar algorithm for the case of pairwise coprime modules.

## 2.2 FRAENKEL'S ALGORITHM

We have also rediscovered a more interesting algorithm for the general Chinese remainder theorem, algorithm presented by Fraenkel in [10]. The idea is to consider $x_i$, the unique solution modulo $[m_1,...,m_i]$ of the system of equations

$$\begin{cases} x \equiv b_1 \bmod m_1 \\ \vdots \\ x \equiv b_i \bmod m_i \end{cases}$$

for all $1 \le i \le k$. Thus, we have that $x_1 = b_1 \bmod m_1$ and that $x_k$ is the required solution.

Consider now $1 \le i \le k-1$. The system of equations

$$\begin{cases} x \equiv x_i \bmod [m_1,\ldots,m_i] \\ x \equiv b_{i+1} \bmod m_{i+1} \end{cases}$$

has an unique solution in $Z_{[[m_1,\ldots,m_i],m_{i+1}]} = Z_{[m_1,\ldots,m_{i+1}]}$, namely $x_{i+1}$. From the first equation, $x_{i+1}$ must be of form $x_i + y \cdot [m_1,\ldots,m_i]$, for some $y \in \mathbf{Z}$. By replacing $x$ using this form in the second equation, we get the equation

$$x_i + y \cdot [m_1,\ldots,m_i] \equiv b_{i+1} \bmod m_{i+1}.$$

From this equation, we have

$$y = \frac{b_{i+1} - x}{(c_i, m_{i+1})} \cdot \left( \frac{c_i}{(c_i, m_{i+1})} \right)^{-1} \bmod \frac{m_{i+1}}{(c_i, m_{i+1})},$$

where $c_i = [m_1,...,m_i]$, for all $1 \le i \le k$.

We obtain the next algorithm:

**CRT_Fraenkel($b_1, \ldots, b_k, m_1, \ldots, m_k$)**

**Input**: $b_1,\ldots,b_k, m_1,\ldots,m_k \in \mathbf{Z}$ such that $b_i \equiv b_j \bmod (m_i, m_j), \forall 1 \le i < j \le k$;

**Output**: $x_0$, the unique solution modulo $[m_1,\ldots,m_k]$ for the system (1);

**begin**

1. for $i := 1$ to $k-1$ do $c_i := [m_1,...,m_i]$;
2. $x_0 := b_1 \bmod m_1$;
3. for $i := 1$ to $k-1$ do
   begin
4. $y := \frac{b_{i+1} - x}{(c_i, m_{i+1})} \cdot \left( \frac{c_i}{(c_i, m_{i+1})} \right)^{-1} \bmod \frac{m_{i+1}}{(c_i, m_{i+1})}$;
5. $x := x + y \cdot c_i$;

   end

**end.**

When $(m_i, m_j) = 1$, for all $1 \le i < j \le k$, we obtain Garner's algorithm [11].

## 2.3 ALGORITHMS BASED ON THE PRIME FACTORIZATIONS OF THE MODULES

Suppose that we know all the prime factors of the involved modules and that these are $p_1,\ldots,p_l$. In this case, every module can be written as

$$m_i = \prod_{j=1}^{l} p_j^{e_{i,j}}$$

for all $1 \le i \le k$, with $e_{i,j} = 0$ in case that $p_j$ does not divide $m_i$. In this case, the equation $x \equiv b_i \bmod m_i$ is equivalent to the system of modular equations $x \equiv b_i \bmod p_j^{e_{i,j}}, 1 \le j \le l$.

Consider $e_j = \max(\{e_{i,j} \mid 1 \le i \le k\})$, for all $1 \le j \le l$, and an index $i_j$, $1 \le i_j \le k$, such that $e_j = e_{i_j,j}$. In this case, the system (1) can be rewritten as

$$\begin{cases} x \equiv b_{i_1} \bmod p_1^{e_1} \\ \vdots \\ x \equiv b_{i_l} \bmod p_l^{e_l} \end{cases} \quad (2)$$

Indeed, because the knowledge of the remainder of integer division of a positive integer $a$ by a prime

power $p^e$ implies the knowledge of the remainder of integer division of $a$ by any $p^{e'}$ with $1 \le e' \le e-1$, it is sufficient to consider only the equations with the greatest prime powers. In this way, any system of modular equations can be reduced to one with pairwise coprime modules. Garner's algorithm or Gauss' algorithm can be used for solving the system (2).

## 3. IMPLEMENTATION DETAILS AND TEST RESULTS

### 3.1 IMPLEMENTATION DETAILS

The large number operations and algorithms are inspired from [7] and have been considered for the only purpose of comparing the two algorithms (Ore's and Fraenkel's). Thus, only few improvements have been made in order to achieve certain performance.

**Addition and subtraction**. We have used the classical algorithms for these two operations. There is no need of getting into details at this part.

**Multiplication**. For few digits numbers we have used the classical *School Multiplication*, while for large amounts of digits, we chose *Karatsuba Multiplication*, which provided a certain speed improvement.

**Division and modular reduction**. Speed related reasons led us to choose *Recursive Division* for numbers with many digits, leaving the smaller ones to the classical *School Division*.

**The (extended) greatest common divisor**. We stopped upon the (extended) *Binary Algorithm*.

**The least common multiple**. We used a simple algorithm for this particular operation: for two numbers, we made use of the fact that

$$[a,b] = \frac{a \cdot b}{(a,b)}$$

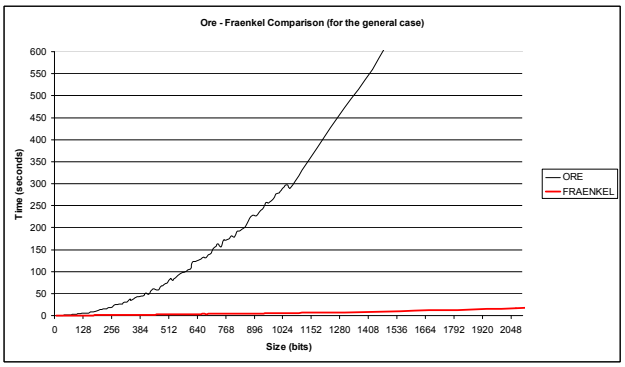
and for more numbers, we split them into two groups, for which we computed recursively the *lcm*. Then, with the results, we applied the above formula.

We designed the library in Microsoft® Visual C++® 6.0 and it is only compatible with Windows® operating systems. The test system was an AMD Athlon™ XP 2700+ with 1024 MB RAM.

### 3.2 COMPARISONS BETWEEN ORE'S AND FRAENKEL'S ALGORITHM

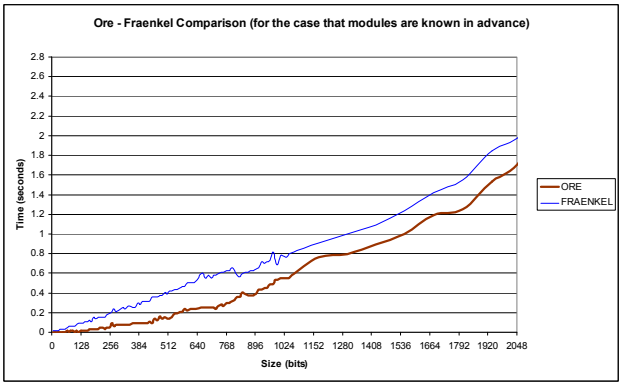
We have considered systems with 5 equations and tested the two algorithms for modules with sizes up to 2048 bits. We divided the comparisons into two sections, which we considered to be distinct, and yet, each very important.

First of them is the comparison for the case that modules are not known in advance.




It is clear that Fraenkel's algorithm behaves much better than Ore's algorithm. This can be motivated by the fact that finding $\alpha_1, \ldots, \alpha_k \in Z$ such that $\alpha_1 c_1 + \cdots + \alpha_k c_k = 1$ (Step 3 from Ore's algorithm) requires a lot of time.

The second one assumes that modules are known in advance and, thus, some pre-computations can be performed.




We have excluded the time for these pre-computations (Steps 1, 2, 3 for Ore's algorithm and the computations of values $c_i$, $(c_i, m_{i+1})$ and the other related values in Fraenkel's algorithm).

In this case, Ore's algorithm behaves better than Fraenkel's algorithm. This can be motivated by the fact that Fraenkel's algorithm has two steps in which pre-computation can not be entirely done (Steps 4, 5) as opposed to Ore's algorithm.

## 4. APPLICATIONS IN CRYPTOGRAPHY

In this section we point out some interesting applications of the general variant of the Chinese remainder theorem. More exactly, we will discuss a generalization of Mignotte's threshold secret sharing scheme and its application to threshold

cryptography.

## 4.1 SECRET SHARING

A secret sharing scheme starts with a secret and then derives from it certain shares (or shadows) such that the secret may be recovered only in the case of possessing a certain predetermined set of shares. The initial applications of secret sharing were safeguarding cryptographic keys and providing shared access to strategical resources. Threshold cryptography (see, for example, [12]) and some e-voting schemes (see, for example, [13]) are some examples of more recent applications of the secret sharing schemes.

In the first secret sharing schemes only the cardinality of the sets of shares was important for recovering the secret. Such schemes have been referred to as threshold secret sharing schemes. We mention Shamir's threshold secret sharing scheme [14] based on polynomial interpolation, Blakley's geometric threshold secret sharing scheme [15], Mignotte's threshold secret sharing scheme [4] and Asmuth-Bloom threshold secret sharing scheme [5], both based on the Chinese remainder theorem. Ito, Saito, and Nishizeki [16], Benaloh and Leichter [17] give constructions for more general secret sharing schemes.

Let $n$ be an integer, $2 \le n$ and $\Pi$ a set of subsets of {1, 2, . . . , $n$}. Informally, a $\Pi$-*secret sharing scheme* is a method of generating the elements $(S,(I_1,\ldots,I_n))$ such that

(1) for all $A \in \Pi$, the problem of finding the element *S*, given the set $\{I_i \,|\, i \in A\}$, is "easy";

(2) for all $A \notin \Pi$, the problem of finding the element *S*, given the set $\{I_i \,|\, i \in A\}$, is intractable.

The set $\Pi$ will be referred to as the *authorized access structure* or simply as the *access structure*, *S* will be referred to as the *secret* and $I_1,\ldots,I_n$ will be referred to as the *shares* (or the *shadows*) of *S*. The elements of the set $\Pi$ will be referred to as the *authorized access sets* of the scheme.

A natural condition is that an access structure $\Pi$ is *monotone*, i.e.,

$(\forall B \subseteq \{1,2,\ldots,n\})((\exists A \in \Pi)(A \subseteq B) \Rightarrow B \in \Pi)$.

In this case, the access structure $\Pi$ is well specified by the set of the minimal authorized access sets, i.e., the set

$$\Pi_{\min} = \{A \in \Pi \mid (\forall B \in \Pi - \{A\})(\neg B \subseteq A)\}.$$

In case $\Pi_{\min} = \{A \subseteq \{1,2,\ldots n\} \,||\, \mathrm{A} \,|\, = \mathrm{k}\}$, for some positive integer $k$, $2 \le k \le n$, a $\Pi$-secret sharing scheme will be referred to as an $(k, n)$-*threshold secret sharing scheme*.

In [18] we have proposed a generalization of Mignotte's threshold secret sharing scheme. Our scheme is based on some particular sequences of integers, namely the *generalized Mignotte sequences*. More exactly, a *generalized* ($k$, $n$)-*Mignotte sequence* is a sequence of $n$ positive integers such that

$$\max_{1 \le i_1 < \cdots < i_{k-1} \le n} ([m_{i_1},\ldots,m_{i_{k-1}}]) < \min_{1 \le i_1 < \cdots < i_k \le n} ([m_{i_1},\ldots,m_{i_k}])$$

Having a publicly known generalized ($k$, $n$)-Mignotte sequence $m_1,\ldots,m_n$, the scheme continues as follows:

- The secret $S$ is chosen as a random integer with $\beta < S < \alpha$, $\beta = \max_{1 \le i_1 < \cdots < i_{k-1} \le n} ([m_{i_1},\ldots,m_{i_{k-1}}])$ and $\alpha = \min_{1 \le i_1 < \cdots < i_k \le n} ([m_{i_1},\ldots,m_{i_k}])$;

- The shares $I_1,\ldots,I_n$ are chosen as follows:

$$I_i = S \bmod m_i,$$

for all $1 \le i \le n$;

- Having the shares $I_{i_1},\ldots,I_{i_k}$, the secret $S$ can be obtained, using the general variant of the Chinese remainder theorem, as the unique solution modulo $[m_{i_1},\ldots,m_{i_k}]$ of the system

$$\begin{cases} x \equiv I_{i_1} \bmod m_{i_1} \\ \quad \vdots \\ x \equiv I_{i_k} \bmod m_{i_k} \end{cases}.$$

For $(m_i, m_j) = 1$, for all $1 \le i < j \le n$, we obtain Mignotte's threshold secret sharing scheme [4].

## 4.2 THRESHOLD CRYPTOGRAPHY

In *threshold* (or *group-oriented*) *cryptography* (see, for example, [12]), the capacity of performing cryptographic operations such as decryption or digital signature generation is shared among members of a certain group. This can be achieved by combining *multiplicative* secret sharing schemes [19] with homomorphic cryptographic operations. The majority of the threshold cryptographic schemes are based on Shamir's secret sharing scheme. In [20] we have indicated how to combine the threshold secret sharing schemes based on the general Chinese remainder theorem with *RSA* cryptosystem [21] in order to get threshold decryption or signature generation. To be precise, the value $x^d \bmod N$ must be collectively computed by any $k$ of $n$ users, where $N = pq$, with

$p$ and $q$ primes, and $(d,[p-1,q-1])=1$.

We will briefly present this scheme (for more details, the reader is referred to [20]). The dealer chooses a generalized Mignotte sequence $m_1,\ldots,m_n$ such that $[p-1,q-1] \mid m_i$, for all $1 \le i \le n$ and $\beta < d < \alpha$. The dealer computes the shares $I_1,\ldots,I_n$ as above and distributes them securely to users. By Ore's construction, the secret key $d$ can be expressed as

$$d = (\alpha_{i_1} c_{i_1} I_{i_1} + \cdots + \alpha_{i_k} c_{i_k} I_{i_k}) \bmod m_{i_1} \cdots m_{i_k},$$

for any set $\{i_1,\ldots,i_k\} \subseteq \{1,2,\ldots n\}$, where $c_{i_j} = \dfrac{m_{i_1} \cdots m_{i_k}}{m_{i_j}}$, for all $1 \le j \le k$ and $\alpha_{i_1},\ldots,\alpha_{i_k}$ are some positive integers such that

$$\alpha_{i_1} c_{i_1} + \cdots + \alpha_{i_k} c_{i_k} \equiv 1 \bmod m_{i_1} \cdots m_{i_k}.$$

In this case we obtain

$$x^d \bmod N = \prod_{j=1}^{k} x^{\alpha_{i_j} c_{i_j} I_{i_j}} \bmod N.$$

Thus, if an authorized group of users want to cooperate in computing $x^d \bmod N$, they individually compute the partial results of form $y_{i_j} = x^{\alpha_{i_j} c_{i_j} I_{i_j}} \bmod N$ and send them to a combiner who will compute the final result as

$$\prod_{j=1}^{k} y_{i_j} \bmod N.$$

In this way the private parameter will not be revealed to the members of the group or to the combiner.

## 5. CONCLUSIONS

We have surveyed the most important algorithms for the general Chinese remainder theorem. In our knowledge, such a survey does not exist in the literature. We have implemented and compared Ore's and Fraenkel's algorithm. Our tests showed that Fraenkel's algorithm behaves better in general, but in the case of modules known in advance, Ore's algorithm proves to be faster, thanks to pre-computation enhancements. Moreover, we pointed out some interesting applications of this variant in secret sharing and threshold cryptography.

## 6. REFERENCES


[1] C. Ding, D. Pei, and A. Salomaa. *Chinese remainder theorem: applications in computing, coding, cryptography*. World Scientific Publishing Co., Inc., 1996.

[2] J.-J. Quisquater and C. Couvreur. *Fast decipherment algorithm for the RSA public-key cryptosystem*. IEE Electronics Letters 8(21) (1982), pp. 905-907.

[3] S. C. Pohlig and M. E. Hellman. *An improved algorithm for computing logarithms over GF(p) and its cryptographic significance*. *IEEE* Transactions on Information Theory, 24:106–110, 1978.

[4] M. Mignotte. *How to share a secret*. In T. Beth, editor, Proceedings of the Workshop on Cryptography, Burg Feuerstein, 1982, volume 149 of Lecture Notes in Computer Science, pages 371–375. Springer-Verlag, 1983.

[5] C. A. Asmuth and J. Bloom. *A modular approach to key safeguarding*. IEEE Transactions on Information Theory, IT-29(2):208–210, 1983.

[6] H. Cohen. *A Course in Computational Algebraic Number Theory*. Graduate Texts in Mathematics. Springer-Verlag, 4th edition, 2000.

[7] F.L. Tiplea, S. Iftene, C. Hritcu, I. Goriac, R.M. Gordan and E. Erbiceanu. MpNT: *A Multi-Precision Number Theory Package. Number-Theoretic Algorithms* (I). Technical Report TR03-02 (2003), Faculty of Computer Science "Al.I.Cuza" University of Iasi. (http://www.infoiasi.ro/~tr/tr.pl.cgi)

[8] O. Ore. *The general Chinese remainder theorem*. American Mathematical Monthly, 59:365–370, 1952.

[9] C.F. Gauss, *Disquisitiones Arithmeticae*, 1801. English translation by Arthur A. Clarke, Springer-Verlag, New York, 1986.

[10] *A. S. Fraenkel. New proof of the generalized Chinese remainder theorem*. Proceedings of American Mathematical Society, 14:790–791, 1963.

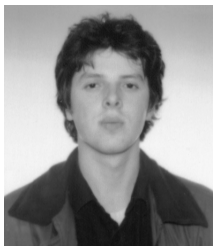
[11] H. Garner. *The residue number system*. *IRE* Transactions on Electronic Computers EC-8 (1959), pp. 140-147.

[12] Y. Desmedt. *Some recent research aspects of threshold cryptography*. In E. Okamoto, G. I. Davida, and M. Mambo, editors, *ISW '97:* Proceedings of the First International Workshop on Information Security, volume 1396 of Lecture Notes in Computer Science, pages 158–173. Springer-Verlag, 1998.

[13] R. Cramer, M. K. Franklin, B. Schoenmakers, and M. Yung. *Multi-authority secret-ballot elections with linear work*. In U. Maurer, editor, Advances in Cryptology - EuroCrypt '96, volume 1070 of Lecture Notes in Computer Science, pages 72–83. Springer-Verlag, 1996.

[14] A. Shamir. *How to share a secret*. Communications of the ACM, 22(11):612–613,

1979.
[15]G. R. Blakley. *Safeguarding cryptographic keys*. In National Computer Conference, 1979, volume 48 of American Federation of Information Processing Societies Proceedings, pages 313–317, 1979.
[16]M. Ito, A. Saito, and T. Nishizeki. *Secret sharing scheme realizing general access structure*. In Proceedings of the IEEE Global Telecommunications Conference, Globecom '87, pages 99–102. IEEE Press, 1987
[17]J. Benaloh and J. Leichter. *Generalized secret sharing and monotone functions*. In S. Goldwasser, editor, Advanced in Cryptology-CRYPTO '88, volume 403 of Lecture Notes in Computer Science, pages 27–35. Springer-Verlag, 1989
[18]S. Iftene. *A generalization of Mignotte's secret sharing scheme*. In T. Jebelean, V. Negru, D. Petcu, and D. Zaharie, editors, Proceedings of the 6th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, Timisoara, Romania, September, 2004, pages 196–201, 2004.
[19]Y. Desmedt, G. Di Crescenzo, and M. Burmester. *Multiplicative non-abelian sharing schemes and their applications to threshold cryptography*. In J. Pieprzyk and R. Safavi-Naini, editors, Advances in Cryptology - Asiacrypt '94, volume 917 of Lecture Notes in Computer Science, pages 21–32. Springer-Verlag, 1995.
[20]S. Iftene. *Threshold RSA Based on the General Chinese Remainder Theorem*. Technical Report TR05-05 (2005), Faculty of Computer Science "Al.I.Cuza" University of Iasi. (http://www.infoiasi.ro/~tr/tr.pl.cgi)
[21]R. L. Rivest, A. Shamir and L. M. Adelman. *A Method for Obtaining Digital Signatures and Public-key Cryptosystems*. Communications of the ACM 2 (21) (1978), pp. 120-126.



***Sorin Iftene*** *has received his M.Sc., B.Sc., and Ph.D. degrees at Faculty of Computer Science, "Al. I. Cuza" University, Iasi, Romania, and is currently lecturer at the same faculty. He is interested in algebraic foundations of computer science, cryptography and computer security. His research was partially supported by the National University Research Council of Romania under the grant CNCSIS632/2006.*

***Florin Chelaru*** *is a third-year student at Faculty of Computer Science, "Al. I. Cuza" University, Iasi, Romania. He is interested in algebraic foundations of computer science, cryptography, data structures, algorithms and computer programming.*