Accessible Secure Information Society Applications via the Use of Optimised Cryptographic Calculations

N. Doukas[2] , A. Drigas[1] N.G.Bardis[1,2] and N.V.Karadimas[2]

[1] National Center for Scientific Research "Demokritos",
Institute of Informatics and Telecommunications & Net Media Lab,
Ag.Paraskevi - Athens, 15310, Greece
dr@imm.demokritos.gr, bardis@ieee.org,
[2] Hellenic Army Academy,
Department of Computer Sciences, Vari - 16673,
Greece
doukasn@sse.gr, nkaradimas@sse.gr

Abstract. Information Society aims to promote innovation in the context of governmental and enterprise information systems and participation of the majority of the general population. An important prerequisite for the penetration and widespread use of Information Society technologies is to enhance the perception of security that these technologies offer without making them inaccessible to users with limited computational resources. In this paper a new algorithm for the software implementation of modular multiplication is proposed, which uses pre-computations with a constant modulus to reduce the computational load imposed upon the processor. The developed modular multiplication algorithm provides faster execution on low complexity hardware in comparison with the existing algorithms and is oriented towards the variable value of the modulus, especially with the software implementation on micro controllers and smart cards whose architectures include a small number of bits. The use of the new algorithm in Information Society applications that demand security is investigated. Such applications include e-Government, e-Banking, e-Commerce etc. The algorithm is shown to be adequate both for the applications for which it was originally intended, as well as for applications that are much more demanding in the level of security they require, such as Military data processing and communication systems.

Keywords: Modular multiplication, Montgomery, Barrett, embedded cryptographic software, efficient implementations

1 Introduction

The necessity for fast algorithms to guarantee security and reliability of information systems is a fact that the scientific community has recognised a long time ago (e.g. [1]) and research in this topic is ongoing. The general population, as represented by its governments and institutions such as the European

Union is beginning to appreciate the advantages it can enjoy by the widespread use of information processing systems and by the development of the so called "Information Society" ([11], [12], [13], [14], [15], [16], [17]). Governmental organisations, commercial enterprises, education, military, public order and safety authorities invest in advanced application of information processing systems and aim to provide and use innovative e-services, apart from replacing old procedures with electronic ones. Despite the developments in the area of integrated circuit design and VLSI, portable devices are still limited in the processing power they provide and by the power consumption required for their operation. Additionally, the acquisition costs of the equipment are still an important factor in technology penetration and security components of information processing devices present a significant overhead in the overall cost.

On the technical front, the operations of modular multiplication and modular exponentiation are the computational basis of an important category of the contemporary information security algorithms, which are based on number theory [9]. The computational implementation of the most important of these security algorithms is based on the operations of modular involution and modular multiplication. Contemporary technologies of this sort include Public Key Algorithms for encryption (RSA, ECC), the Diffie-Hellman key exchange algorithm, Digital Signature Algorithms and the Digital Signature Standard. In order to provide an adequate security level against possible attacks, the above algorithms use numbers with word lengths that may in certain cases exceed 10^3 bits. More specifically, current security level requirements dictate that the word lengths to be used for the algorithms based on elliptical curves (ECC) should range from 128 to 256 bits, while the word lengths used for algorithms that are based on exponential transformation should range from 1024 to 2048 bits [6].

Software implementations of modular multiplication for numbers of such long word lengths on general-purpose processors or micro-controllers with much smaller word length architectures (from legacy 8 bit microcontrollers to state-of-the-art 64 bit processors) inevitably involve a large volume of overhead processing and are hence considered extremely computationally complex. Consequently, the software implementations of public key based information security algorithms on general purpose processors become several orders of magnitude slower in comparison with symmetric algorithms (such as DES or AES) if they are required to provide protection against security attacks of similar intensity. Naturally, the speed of computation of modular arithmetic operations on large numbers becomes an especially acute problem in the case of low word length microcontrollers [3].

The basic operation of the modular arithmetic used in algorithms of this class, is the modular involution, i.e., the calculation A^k mod M. In the majority of implementations, modular involution computations are carried out by the method of "squaring and multiplications" [2], which uses a number of multiplications close to the theoretical minimum. Based on this fact, in order to increase the performance of the algorithms implemented in software, it is important to decrease the time needed for performing modular multiplication.

Consequently, research for algorithms to increase the performance of modular multiplications, for use in software implementations of encryption schemes on general purpose processors and micro-controllers, is vitally important. This paper presents an innovative algorithm for performing modular multiplication with a smaller computational effort in conditions where relatively limited computational resources are available. The algorithm is based on exploitation of the fact that encryption keys are not changed very frequently and that hence, parts of the operations that need to be performed are applied to virtually constant parameters. The new algorithm is shown to be capable of improving the speed of calculations related to information security by a factor of 1.5 to 2. By contrast to existing techniques, the proposed technique is targeted at software implementations of modular multiplication on general purpose processors. This implies that the technology is available to anybody using an information processing system and not just to users that enjoy the luxury of using purpose build, dedicated security hardware.

The state of the art in Montgomery Multiplication acceleration is investigated in Section 6. It is noted that almost all proposals are targeted for implementation on special hardware ranging from general purpose FPGAs and reaching to purpose designed VLSI. There has been virtually no research that aimed optimising the speed of execution of the Modular Multiplication for low-end general purpose processors, such as the ones that are commonly used in many portable devices that facilitate the extended use of information processing services required for industrial, military, e-commerce, e-banking, security, e-government and other information society applications. An innovative algorithm is proposed that provides a solution to the particular problem. This algorithm is shown to improve performance by a factor of 1.5 and 2.0. The difference between the proposed algorithm and existing techniques that present certain similarities are clearly explained.

Following the previous analysis, European Union research projects findings are summarised, that investigate Information Society Technology needs and factors that impede the adoption of new technologies by the general public and their penetration in everyday activities. The new algorithm is shown to be capable of answering two important concerns, namely security and cost. The fact that security becomes cheaper and faster to implement implies that it becomes feasible for larger parts of the population to enjoy high levels of security during their everyday electronic transactions. Apart from its other characteristics, that satisfy the requirements set so far, the new algorithm is shown to be robust and capable of being used in very demanding applications such as those concerned with defense and public safety information systems.

2 Principal Notations and Effectiveness Estimation Model of the Modular Multiplication Algorithms

The basic modular arithmetic operation used within the context of information security algorithms is modular multiplication, i.e. the calculation

$$R = A \cdot B mod M. \tag{1}$$

The assumptions required to be made are:

- that the result R, coefficient A, multiplicand B and modulus M are n bit binary numbers,
- that the most significant bit of the modulus is equal to 1, i.e. $2^{n-1} \leq M \cdot 2^n$,
- that the co-factors are lower than the modulus, i.e. $A < M$, $B < M$.

It is also assumed that the operation of modular multiplication is performed on a k bit general purpose processor, microprocessor or micro-controller.

Consequently, each one of the numbers which participate in the operation of modular multiplication can be represented in the form of $s = n/k$ bit words:

$$A = \sum_{j=0}^{s-1} a_j \cdot 2^{j-k}, \quad B = \sum_{j=0}^{s-1} b_j \cdot 2^{j-k}, \quad M = \sum_{j=0}^{s-1} m_j \cdot 2^{j-k} \tag{2}$$

where $a_j,\ b_j,\ m_j$ are k-bit words and $j \in \{0, \ldots, s-1\}$.

In contrast to the classical modular multiplication algorithm outlined in [2], contemporary algorithms [4], [7] do not use the operation of division, which is inefficiently realised on most general purpose processors. Based on this, the usual criterion of computational complexity taken in order to compare software modular multiplication implementations is the total number of multiplications and additions required [6].

Elementary arithmetic considerations show that the product of two k bit numbers requires $2 \cdot k$ bits for its accurate representation. By denoting:

- q_m - the number of multiplications required,
- t_m - the execution time needed for each command,
- q_a - the number of additions required,
- t_a - the execution time of each addition,

the estimate for computational complexity of the particular software implementation of the n bit modular multiplication is:

$$C_n = q_m \cdot t_m + q_a \cdot t_a \tag{3}$$

If the ratio of the execution times for the multiplication and addition commands on the processor is $w = t_m/t_a$, then the execution time of the modular multiplication can be represented as:

$$E_t = t_a(w \times q_m + q_a). \tag{4}$$

3 Brief Analysis of the Contemporary State of the Acceleration Problem: The Software Implementation of the Modular Multiplication

In Algorithm 1, the classical algorithm [2] for a word by word software implementation of modular multiplication is given in the form of C++ pseudocode. In the description given, the implementation of Reduce(X) (which returns the modular reduction of X), has been omitted as it is the main concern of this research and it is going to be extensively analysed in the sections of this paper that follow.

The operation of multiplication is performed on a word-by-word basis. More specifically, the j^{th} $(j = 0, \ldots, s-1)$ of the k bit word of the coefficient a_j is multiplied by shifting each of the s words of the multiplicand in B. The obtained products, which are $2k$ bits long, are added, forming $(n+k)$ bits, a result which is a partial representation of the product shown in Equation 5.

$$a_j \cdot B = \sum_{j=0}^{s-1} a_j \cdot b_i \cdot 2^{i-k} \tag{5}$$

Algorithm 1 Classical scheme for word-by-word Modular Multiplication

```
R=0;
for(i=0; i<s; i++) \{
        Y=0;
        for(j=0; j<s; j++)
               Y+= (a_i*b_j)<<(j*k);
        R += Reduce(Y);
        if (i<s-1) \{
                B<<=k;
                Reduce(B);
        \}
\}
Reduce(R);
```

Following this, the modular reduction of the partial expression is carried out, obtaining j^{th} partial residual $R_j = a_j \cdot BmodM$. The result of the modular multiplication $R = A \cdot BmodM$ is formed as the sum of the modular reductions of the partial expression of the product: $R = (R_0 + R_1 + \ldots + R_{s-1})modM$.

The classical modular reduction algorithm is achieved with the use of the operation of the integer division of $2k$ bits divisible to the k bits divider, obtaining a quotient and a residual. Since the division of the n bit numbers on the k bits processor $(n >> k)$ is carried out very ineffectively, the calculation of the reduction in the classical algorithm requires $s(s+2.5)$ multiplication operations and s integer division operations [4].

Up to now, various algorithms have been proposed ([3], [4], [6]) which improve the performance of the software implementations of the modular multiplication operation. The majority of these algorithms realise the increase in the performance of modular multiplication via the acceleration of modular reduction with the exception of the operation of integer division, which is used in the classical algorithm [2]. Existing algorithms are analysed in greater detail in Section 7. Currently, the most effective method of modular multiplication is the Montgomery algorithm [7], which is well adjusted to the architecture of general purpose processors. The Montgomery algorithm substitutes the operation of division into the random modulus M with divisions into powers of 2, which may be efficiently implemented by shifts. The operation of modular reduction in Montgomery's algorithm requires $s(s+1)$ multiplication operations.

The general computational complexity of the implementation of the Montgomery modular multiplication algorithm on a k bit processor is determined by $2s^2+s$ operations of multiplication and by $4s^2+4s+2$ additions. Accordingly, the calculation time T_M of Montgomery's algorithm on the k bits processor can be calculated approximately as follows:

$$T_M = (2s^2+s)t_m + (4s^2+4s+2)t_a = t_a(s^2(2w+4)+s(w+4)+2) \qquad (6)$$

Known algorithms assume that each calculation of modular multiplication is produced with the new values of co-factors A, B and of the modulus M. However, the analysis of the practical application of information security algorithms, which use modular multiplication, shows that both their keys, and respectively the modulus change relatively rarely. This offers the potential of further decreasing the computational complexity of modular multiplication by simplification in the reduction. The practical implementation of such possibilities requires special research and development. On this basis new modular multiplication algorithms should be developed, which will contain a constant modulus.

The target of this work is the development of an effective modular multiplication algorithm for large numbers, operating with a constant modulus suitable for efficient and fast implementation on general purpose processors that are build on architectures with a small number of bits.

4 Analysis of the Possibilities of Accelerating the Modular Multiplication in the Information Security Systems

Information security algorithms are based on cryptographic properties. The particular cryptographic property of concern to this work has to do with the intractability of a solution of a particular problem based on analytical methods and number theory. This class of algorithms require the special complex procedures for the generation of effective keys.

In particular, the extensively popular RSA algorithm [1] uses a complex procedure to obtain the three numbers d, e and M with lengths n between 1024 and 2048 bits that satisfy the identity:

$$A^{de} \equiv A. \tag{7}$$

The process of the coding of the block A of a certain message consists of the calculation of $C = A^e mod M$ and the decoding of block A is realized with the calculation of $A = C^d mod M$. The pair of numbers $< d, M >$ composes the public key, while the pair $< e, M >$ composes the private key.

One of the above keys, depending on the protocol that the RSA uses, is public while the other one is private. The analysis of the practical use of an RSA algorithm shows that the keys change relatively rarely so that with the use of the same key, tens of thousands of information blocks are processed. This makes it possible to consider that in the process of computational implementation, the RSA key and consequently the modulus are both in effect constant. Analogous reasonings can also be applied to a number of other standardized information security algorithms that are widely applied in practice. A very important algorithm of this sort is the Digital Signature Standard algorithm [6].

The fact that the modulus M is constant, renders it possible to simplify the calculation of modular reduction in the multiplication process via the use of pre-computed results. Such pre-computations depend only on the value of the modulus M and therefore they may be carried out off-line and may be recovered whenever there is a change of the modulus. The results of the pre-computations can be stored in the tabular form in main memory and are used repeatedly with each modular multiplication calculation.

In the modular multiplication implementation, part of the computational resources is strictly used for the calculation of multiplication and the other part for the implementation of modular reduction. In different modular multiplication algorithms ([7], [8]) the specific weight of expenditures for these two procedures varies. Table 1 gives the number of multiplication operations and the word divisions, which are required in the most common modular multiplication algorithms for the calculation of the product $A \cdot B$ and the modular reduction implementation ([4], [5]).

It is trivial to show that the possibilities of decreasing the number of operations for the calculation of the product $A \cdot B$ via pre-computations with a constant modulus are extremely limited, since the modulus itself is not used directly in such calculations. Therefore, the basic technology for increasing the speed of the software implementation of modular multiplication is the use of pre-computations for decreasing the computational complexity of modular reduction. Data analysis, given in Table 1 shows that with the use of pre-computations, a significant decrease in the computational effort required is achieved. This takes place via the reduction of the time expenditures for the modular reduction.

Table 1. Number of multiplications and divisions over k bit words used by different modular multiplication algorithms and calculation of the multiplication of the expression $A \cdot B$ and the residual of modulus M.

Algorithm	**Number of multiplications k-bit word for calculation $A \cdot B$**	**Number of multiplications k-bit word for modular reduction**	**Number of divisions word for modular reduction**
Classical	s^2	$s^2 + 2.5s$	s
Barrett	s^2	$s^2 + 4s$	0
Montgomery	s^2	$s^2 + s$	0

5 Modular Multiplication Organization Based on Pre-Computations with the Fixed Module

In the classical algorithm (Algorithm 1) the modular reduction procedure, *Reduce(X)*, is carried out from the partial products $a_j \cdot B$ and by shifting the code of multiplicand B by k bits to the left. In both cases, the length of the reduced number X is not more than $(s+1)k$ bit words or more than $(s+1) \cdot k = (n+k)$ bits $x_0, x_1, x_2, \ldots, x_{n+k-1}$, as shown in Equation 8.

$$X = \sum_{j=0}^{n+k-1} x_j \cdot 2^j, \quad x_j \in \{0, 1\} \tag{8}$$

The number X can be represented in the form of the sum of two components: an $(n-1)$ bit number X'', which coincides with $(n-1)$ least significant digits of X and an $(n+k)$ bit number X', which consists of the $(k+1)$ most significant digits of X, concatenated with $n-1$ zero bits on its right hand side, as shown in Equation 9.

$$X = \sum_{j=0}^{n+k-1} x_j \cdot 2^j = X' + X'', \quad X' = \sum_{j=n-1}^{n+k-1} x_j \cdot 2^j, \quad X'' = \sum_{i=0}^{n-2} x_i \cdot 2^i \tag{9}$$

In accordance with the property of congruence for the modular reduction, the residual $X mod M$ can be represented in the form of the modular reduction as the sum of the residuals of the components of X i.e. X' and X'', as shown in Equation 10.

$$\begin{aligned} X \, mod M &= \left(\sum_{j=0}^{n+k-1} x_j \cdot 2^j \right) mod M = (X' + X'') mod M \\ &= (X' mod M + X'' mod M) mod M \end{aligned} \tag{10}$$

Since the most significant $(n-1)$ bits of modulus M are equal to one and the X'' is an $(n-1)$ bit number, then $X'' < M$ and consequently $X''modM = X''$. The number X' contains only $k+1$ significant digits. The rest $n-1$ low-order digits are equal to zero. Consequently, X' and accordingly $X'modM$ assume only 2^{k+1} different values. All possible n bit values of $X'modM$ for the appropriate X', can be pre-computed and stored in main memory in the form of tables.

If the symbol Z is designated to represent the binary code which consists of the $(k+1)$ most significant digits of X' then,

$$Z = \sum_{j=n-1}^{n+k-1} x_j \cdot 2^{j-n+1} \tag{11}$$

and with $T(Z)$ being the n bit code of the tabular value $T(Z) = X'modM$, then the modular reduction procedure Reduce(X) is realized in accordance with the following expression in Equation 12.

$$Reduce(X) = X\, modM = (T(Z) + X'')modM \tag{12}$$

In this case, the computational complexity of the modular reduction implementation is determined by maximum two operations of addition between $(n+k)$ bit numbers: the first for the calculation of $T(Z) + X''$ and the second for executing the subtraction $(T(Z) + X'') - M$, if $T(Z) + X'' \geq M$.

Since $0 \leq T(Z) + X'' < 2 \cdot M$, for the modular reduction $(T(Z) + X'')modM$ no more than one subtraction of n bit numbers may ever be required. Therefore, the execution time of the procedure will not exceed $2(s+1)t_a$, with the average value of $1.5 \cdot s \cdot t_a$.

The storage memory, which is required for storing all the pre-computed possible values of $T(Z)$ has a volume of $2^{k+1}n$ bits or $2^{k+1}s$ of k bit words.

The proposed approach is especially effective in the implementation of modular multiplication on the small word length microprocessors, micro-controllers and smart cards. In this case, the memory size for storing the results of pre-computations with a constant modulus proves to be acceptable for the majority of applications. For example, for the accelerated multiplication implementation of the 1024 bit numbers on the 8 bits micro-controller, the capacity of the required storage memory will consist of $(2^9 \cdot 128) = 2^{16}$ bytes (64 KBytes). For the accelerated modular multiplication implementation on the 16-bit processor r, the above capacity requires storage memory which substantially grows and depending on the availability of memory, may decrease the effectiveness of the application of pre-computations.

In order to decrease the capacity of the memory required for storing the results of pre-computations $T(Z)$, multi-section organization is proposed. The essence of the proposed table organization of pre-computations lies in the fact that the value X' is divided into q components, namely

$$X' = X'_1 + X'_2 + \ldots + X'_q, \tag{13}$$

with lengths $n + r_1, n + r_1 + r_2, \ldots, n + r_1 + + r_q$, since $r_1 + r_2 + + r_q = k + 1$.

For every i, $X'_i(i = 1, \ldots, q)$ consists of r_i most significant digits, which coincide with the digits $x_{n+h}, x_{n+h+1}, \ldots, x_{n+h+r_i}$ of the number X ($h = 0$ for $i = 1$ and $h = r_1 + \ldots + r_{i-1}$ for $i > 1$), and the rest of the low-order digits are equal to zero:

$$X'_i = \sum_{h=g_1}^{g_i+r_i} x_{n+h} \cdot 2^{n+h}, \quad g_i = \sum_{t+1}^{i-1} r_t, \quad \forall i \in \{2, \ldots, q\}, \quad g_1 = 0. \tag{14}$$

Following this, in accordance with the property of congruence the modular reduction $X mod M$ can be represented in the form shown in Equation 15.

$$X \, mod M = (X'_1 mod M + X'_2 mod M + \ldots + X'_q mod M + X'') mod M. \tag{15}$$

In order to determine each of the values of $X'_i mod M$, the use of the pre-computed results is proposed, where the results are previously calculated for all possible codes X'_i. Since the number of significant (non zero) bits in the code X'_i is equal to r_i, then the number of possible different values of X'_i will be 2^{r_i} and the memory capacity required for storing all possible values of $X'_i \, mod M$ respectively, will be $(2^{r_i} \cdot n)$ bits. If the r_i bit binary code which contains only r_i high order significant digits X'_i is denoted as Z_i, then

$$Z_i = \sum_{h=g_1}^{g_i+r_i} x_{n+h} \cdot 2^{h-g_i}, \quad g_i = \sum_{t+1}^{i-1} r_t, \quad \forall i \in \{2, \ldots, q\}, \quad g_1 = 0. \tag{16}$$

If the n bit code of the tabular value $T_i(Z_i) = X'_i \, mod M$ is denoted as $T_i(Z_i)$, then the modular reduction procedure is realized in accordance with the following expression:

$$X \, mod M = \left(\sum_{i=1}^{q} T_i(Z_i) + X'' \right) \, mod M. \tag{17}$$

The total volume of the tabular memory required for storing the values

$$T_1(Z_1), T_2(Z_2), \ldots, T_q(Z_q) \tag{18}$$

occupies a space of $n \cdot \sum_{i=1}^{q} 2^{r_i}$ bits. This example of storing $T(Z)$ in a table can be examined as a special case of the results organization within subdivided tables, with pre-computations for $q = 1$.

The use of multi-section tables makes it possible to substantially decrease the memory capacity required for their storage. Following the same assumptions as for the example given above, for the accelerated multiplication implementation of 1024 bit numbers on the 8 bit micro-controller with the two-section memory ($q = 2, r_1 = 5, r_2 = 4$), the required memory capacity will compose of $1024 \cdot (2^5 + 2^4) = 10^{10} \cdot 48$ bits or 6144 bytes or 10.67 times less than during the single-section organization of tabular memory. Such an amount of memory may be considered as easily available in any contemporary information processing device.

From another point of view, the use of multi-section organization of the tabular memory is combined with the increase of the execution time of the modular reduction. The calculation of the sum of Equation 17 requires $q(s+1)$ addition operations for accumulating k bit words. The number of significant digits of the sum code will not exceed in this case $n+q$, so that, if $r_1 \geq q+1$, then for executing the modular reduction of sum with the use of the first table $T_1(Z_1)$, $1.5(s+1)$ addition operations are required, on average. The total number of the additional operations is: $(q+1.5)\cdot(s+1)$.

6 Existing schemes for the acceleration of the Montgomery Multiplication

As it has already been mentioned, virtually all existing proposals for algorithmic improvements to accelerate the execution time required for the modular multiplication, attempt to do so by efficiently using specialised hardware. The work presented in [18] is concerned with supporting the algorithm for Residue Number System division via the subtractive technique. This work uses the Montgomery Algorithm in order to calculate a parity check and could potentially benefit from the new scheme proposed by the results of the present research. In [19], an algorithm is proposed that exploits the particular case of the radix-4 case in order to reduce the computations. The present work is therefore more general in the field of its application. Additionally, the work in [19] is targeted for use and is implemented in an FPGA platform with advanced multiprocessing capabilities. In [20], the various special cases of coefficients for the polynomial multiplication of the NTRU algorithm are distinguished in order to design optimised hardware that can quickly perform the required operations. The work in [21] accelerates the overall processing time by leaving the Montgomery Multiplication unchanged and partially replacing calculations by a modified Barrett Modular Multiplication. In [22] and [25] the concept of low-weight polynomial form integers is introduced in order to design fast implementations on FPGA hardware. In [23], precomputations of just the radix are used with the aim again to facilitate the implementation on FPGAs.

Two of the algorithms encountered in contemporary literature present significant contributions targeted at the particular problem of accelerated, namely purely software implementations of Modular Multiplication to be run on general purpose processors. The work in [24], aims to produce fast software implementations. A series of complex checks is used to avoid unnecessary calculations that result in a 3% to 7% reduction of the computation time required to complete the computations on a statistically significant collection of sample cases, tested on a 1.86 GHz Pentium Processor. This result is not directly comparable to the results of this work, as the Pentium is a 32-bit processor. However [24] may be considered to be serving a different scope of application. The most relevant independent work that may be considered as work parallel to the present study, is the Bipartite Modular Multiplication method presented in [26]. This method recognises the advantages in computational complexity reduction that may re-

sult from splitting the operand in two parts. It however overlooks the advantages of using the precomputed results to speed up the process as it is targeted towards either software implementations that are executed in multiprocessor hardware. Precomputation of results is, as it has already been analysed in previous sections, the principle technique that reduces the overall execution time required for the proposed method. An additional difference of the newly proposed method from the Bipartite Modular Multiplication is the exploitation of the fact that with careful selection of the bit lengths of the two parts of the operand, the calculation for the least significant part of the operand becomes trivial and takes up virtually no computational effort at all. From the above analysis it may be clearly inferred that the proposed algorithms targets an unexplored aspect of the accelerated software modular multiplication, using an innovative approach.

7 Social Expectations from the Information Society and Equal Opportunities Concerns

Having established a new technological proposal, it is interesting to note some conclusions from European Union projects that aimed to investigate a different aspect of Information Society, namely the acceptance of new technologies by European societies and the concerns that are raised relative to the accessibility of new technologies to the less privileged part of the general population. A study was conducted that focused specifically in Hungary and investigated the expansion of use of Information Systems for the purposes of facilitating old and producing new services for governmental agencies, banks, companies, etc [14]. This study found a very small rate of expansion of Information Society until the end of 2002 and a reverse trend in 2003. The study indicated that the stagnation was caused not only by the limited financial and infrastructural possibilities but, in contrast to prior false professional beliefs, also by a specific set of attitudes. It also concluded that internet use increased in recent years to a much smaller extent than anticipated. This was caused by anxiety, suspicion towards novelties and, for various reasons, towards service providers, as well as distrust. In 2003 the lack of interest in innovative internet e-services was reduced by 10%. The prevailing mentality, however, still fails to consider information and knowledge as resources, and information technology as means of production and one of the principle reasons is lack of trust for the technologies involved. The widespread use of modern information and communications technological devices was also found to have a significant effect on the generation of social differences. In other words the difference in means and extent of access to equipment and methods of use of information processing related services, further deepen social differences. This phenomenon, which was singled-out by this study in Hungary [14] but exists in virtually all countries, has been assigned the term *digital divide*. In the case of Hungary it was found that two factors deepening the digital divide are income and locality. Part of the response to the above factors may be the production of secure and reliable, mobile hardware that may, with a low cost of acquisition, provide access to the services already mentioned.

A different study [15] shows that security loopholes in the technology are a major drawback for Information Society Technology penetration. Customers of advanced e-services feel threatened by the perceived possibility for intentional or unintentional abuse of the personal data they submit in order to gain access to such services. Surveys indicate that around 1/3 of the potential customers decided not to participate in the electronic market because they perceive problems (prominent among which are privacy aspects, and concerns with respect to identity fraud). This lack of trust implies that providers of services (companies, banks, the government etc) do not use their full market or society potentials because a significant part of their target audiences are not confident enough to trust their products and services. Lack of trust was found to be particularly caused by privacy and security policies or lack thereof.

The fact that the ability to provide robust information security may provide the distinguishing advantage to an e-service provider is also pointed out in the report [16]. Besides harm that is produced by malicious code, unauthorised access to systems or data, and fraud based on misuse of such data, security and trust concerns for many can be a severe barrier to Information Society participation. It impacts the diffusion of electronic commerce, the success of e-Government and the participation in the Information Society at large. Most importantly, these worries affect participation. For instance, half of all European Internet users said that these concerns had prevented them from shopping online, and similar proportions are expected in respect of e-Government services. Similar concerns affect e-banking and financial e-services [17].

If individuals distrust sending the identifying or financial information over the Internet that is needed to complete transactions, the fraction of commercial and societal activities which can benefit from transition to the electronic medium will be significantly restricted. As a result, insufficient protection (or a perception of insufficient protection) of personal privacy and security in these systems is a potentially serious impediment in the development of the information society and, therefore, is important from the policy perspective. Acknowledging that security and trust are important issues in the development of e-economy and the information society, e-Europe documents [17] state that it is necessary to "determine the adequate amount of security for user needs". Barriers to the implementation of e-government range from the cost to distrust towards e-government.

The findings outlined above support the hypothesis that two major factors that prevent specific groups of people, as well as the general population, of benefiting from advances of Information Society technologies are the lack of sense of security and the costs involved in gaining access to the technology. The proposed algorithm can be viewed as a step in the right direction. It is an enabling technology that may allow the production of low in cost, small and portable in size and low in power consumption secure devices for access to innovative e-services. Such services may originate from all kinds of providers and involve applications related to e-government, e-commerce, data processing within enterprises, defense, public order and safety etc.

8 Scope of application of the proposed modification of Montgomery's diagram and evaluation of its impact on Information Society Applications

Modular multiplication is the enabling computational foundation for contemporary technologies for information protection that are considered basic building blocks in the construction of information protection schemes. Such technologies are public key encryption, digital signatures, authentication, confidentiality, identification based on a zero knowledge scheme etc. The majority of applications that demand the attainment of a high level of security use long word lengths (usually 1024 or 2048 bits), over which modular multiplication is applied. The implementation of such complex calculations demands an excessive computational cost, even for the most powerful modern processors. The problem of computational complexity and speed for the realization of modular multiplication on long word length numbers becomes even more acute in the case of embedded micro-controllers and microprocessors that have by construction an extremely limited processing power.

As computational devices have become increasingly pervasive, they also have become increasingly mobile, making possible new forms of personal learning and challenging our expectations of the environments where learning can happen. The limited power, bandwidth, memory, and display resolution of portable devices pose additional challenges in designing for constructive learning [10]. The new technique is a significant response to these challenges, since it makes the exploitation of bandwidth significantly more efficient. The proposed method renders at least a partial solution of this problem feasible and reduces the required computational effort for calculating the modular product. This reduction is achieved by means of the use of pre-calculated results that depend solely on the modulus, a parameter which is essentially constant for the majority of applications.

The proposed method is extremely effective, especially in the case of embedded micro-controllers and microprocessors. The expansion and the development of information integration, lead to the fact that such controllers are very often encountered inside terminal devices belonging to local and wide area networks. This implies that these devices need to support all the existing network protocols, including the protocols for information security. In this case, it is also important to note that the majority of such devices are required to operate in real time conditions and hence the implementations of network protocols need to be correspondingly fast.

The framework for the collection and management of technical information and control procedures, on which both the concept of Information Society and many Military Applications are based, belong to the basic scope of microcontroller applications that need to support network protocols and use modular multiplication procedures. As society is beginning to appreciate that organisational knowledge is a source of a competitive advantage in the long run [13]. Thus, the creation, deployment, transfer, transformation, renewal and accumu-

lation of knowledge turned out to be key strategic activities for all kinds of firms and organizations, as well as for society in general [13]. The proposed modification of Montgomery's scheme safeguards a significant proportion of this strategic activity which includes both the deployment and the transfer of knowledge.

In [11] it is stated that, for organizations to maintain a competitive edge, employees must be able to ability to externalise and share their knowledge. The sharing of the knowledge is promoted by use of the new scheme. According to [12], an organization is required, among other activities, to be able to organise, store, make accessible and distribute the knowledge it possesses in order to remain competitive.

Remote controlled devices that are monitored and controlled via network connections also belong to the scope of application of modular multiplication implementations. Such systems can potentially be exposed to risks arising from both malicious and unintentional interventions to their operation. It is therefore the case that micro-controllers used in those systems also need to devote a significant portion of their processing power for the implementation of information protection protocols. A fundamental area of effective application of the proposed scheme, are systems for electronic government and electronic commerce that are supported by the use of open networks. The means used in this case, usually smart cards and mobile phones, must be able to implement the entire spectrum of protocols that guarantee the security of all transactions. This means that embedded controllers within these devices must be capable of rapidly calculating complicated modular multiplication expressions on numbers of long word lengths.

9 Conclusions

The problem of increasing the performance of the modular multiplication software implementation, which is the basic computational operation used in a wide circle of information security algorithms was analysed. It was noted that during the practical application of information security algorithms, that are based on analytically intractable number theory problems the keys and consequently the modulus, change relatively rarely. Based on the above finding, a new algorithm was proposed for the modular multiplication that differs from the classical organization of the modular reduction execution. Reduction in the computational complexity of the software implementation was achieved by the use of pre-computations results, which depend only on the modulus and which are stored in tabular form in main memory. The performance estimate of the proposed algorithm and memory usage for storing the precomputed tables were computed.ted tables were computed.

The analysis carried out showed that the speed of the software implementation of modular multiplication on the micro-controllers with the use of the proposed algorithm was increased by a factor between 1.5-2, in comparison with the most effective existing algorithm, the Montgomery algorithm.

A study of Information Society and the new social objectives that it represents showed that serious impediments against widespread diffusion of innovative information processing technologies and the innovative services that these technologies enable are the lack of technological information security perceived by users. This lack of security was found to be caused by cost and mobility considerations that prevent manufacturers from including encryption components in a wide range of extensively used products, or make such products too expensive for significant parts of the population. The analysis showed that the proposed algorithm is a step in the right direction, since it is an enabling technology for fast and accessible implementations of security algorithms for use in information processing devices. Such devices can then be used to promote the goal of providing a wide range of innovative e-services by governmental, enterprise, commercial and military organisations.

References

1. R.L. Rivest, A. Shamir, and L. Adleman; A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, Comm. ACM, vol. 21, no. 2, pp. 120-126, Feb. 1978.
2. Bosselaers A., Govaerts R., Vandewalle J. Comparison of three modular reduction functions. Proceeding of Advances in Cryptology CRYPTO'93, LNCS-773, Springer-Verlag, 1993, pp. 175-186.
3. Dhem J.-F., Quisquater J.-J. Resent results on modular multiplications for smart cards. Proceeding of GARDIS 1998. LNCS-1820, Springer-Verlag, 2000, pp. 350-366.
4. Laszlo Hars. Long Modular multiplication for Cryptographic Applications. Cryptographic Hardware and Embedded System- CHES'2004. LNCS-3156, Springer-Verlag, 2004, pp.45-61.
5. Hong S.M., Oh S.Y., Yoon H. New modular multiplication algorithms for fast modular exponentiation. Proceeding of Advances in Cryptology Eurocrypt'96, LNCS-1070, Springer-Verlag, 1996, pp. 166-177.
6. Menezer A.J., Van Oorschot P.C., Vanstone S.A. Handbook of Applied Cryptography. CRC-Press, 1997.
7. Montgomery P.L. Modular multiplication without trial division. Mathematics of Computation, Vol. 44, 1985, pp. 519-521.
8. Barrett, P.: Implementing the River Shamir and Adleman public key encryption algorithm on a standard digital signal processor. In Odlyzko, A, ed.: Advances in Cryptology - CRYPTO '86, Santa Barbara, California. Volume 263 of Lecture Notes in Computer Science., Springer - Verlag (1987) pp. 311-323.
9. Cohen, H.: A Course in Computational Algebraic Number Theory 2^{nd} edn. Graduate Texts in Mathematics. Springer (1995).
10. Carol Strohecker. Designing for sensing, sensibilities, and sense-making. Int. J. Knowledge and Learning, Vol. 1, No. 3, 2005
11. Choudrie, J. and Selamat, M.H., 'Managing organisational learning through continuous information systems development: tacit knowledge diffusion and meta-abilities perspectives', Int. J. Knowledge and Learning, Vol. 1, No. 4, 2005,pp.342-356.
12. Coakes, E. and Bradburn, A., 'Knowledge management practices in some UK service organisations', Int. J. Knowledge and Learning, Vol. 1, No. 4, 2005, pp.305-317.
13. Ordonez de Pablos, P., 'Intellectual capital statements: what pioneering firms from Asia and Europe are doing now', Int. J. Knowledge and Learning, Vol. 1, No. 3, 2005, pp.249-268.

14. Hungarian Informatikai es Hirkozlesi Miniszterium. Hungarian Information Society Strategy. Hungary 2008.
15. Ronald Leenes, Simone Fischer-Hubner (Ed). Privacy and Identity Management for Europe Framework document V2. EU PRIME Project, Contract No 507591, 2006.
16. European Commission Information Society DG Final Report for no 2004/s 3-001615. Inventory and best practices guide on Member states, EEA Countries and Accession states activities in the field of network and Information security, 1 March 2005.
17. EU SIBIS Project Workpackage 1. eEurope Benchmarking Framework Deliverable 1.2: Update of eEurope Benchmarking Framework, October 2002.
18. Jung-Hee Suk, Jin-Seon Youn, Hui-Gon Kim, Taek-Won Kwon and Jun-hm Choi: A Parity Checker for a Large Residue Numbers. Signals, Circuits and Systems, 2005 (ISSCS 2005). International Symposium on. Volume 1, 14-15 July 2005 Page(s):355 - 358 Vol. 1.
19. Tawalbeh, L.A.; Tenca, A.F.; Koc, C.K.; A radix-4 scalable design Potentials, IEEE Volume 24, Issue 2, April-May 2005 Page(s):16 - 18.
20. Colleen ORourke and Berk Sunar: Achieving NTRU with Montgomery Multiplication. IEEE Transactions on Computers, Vol. 52, No. 4, April 2003.
21. Hasenplaugh, W.; Gaubatz, G.; Gopal, V.; Fast Modular Reduction Computer Arithmetic, 2007. ARITH '07. 18th IEEE Symposium on 25-27 June 2007 Page(s):225 - 229.
22. Jaewook Chung and M. Anwar Hasan. Low-Weight Polynomial Form Integers for Efficient Modular Multiplication. IEEE Transactions on Computers, Vol. 56, No. 1, January 2007.
23. Thapliyal, H.; Ramasahayam, A.; Kotha, V.K.; Gottimukkula, K.; Srinivas, M.B.; Modified Montgomery modular multiplication using 4:2 compressor and CSA adder Electronic Design, Test and Applications, 2006. DELTA 2006. Third IEEE International Workshop on 17-19 Jan. 2006 Page(s):4 pp.
24. Tang, P.T.P.; Modular Multiplication using Redundant Digit Division. Computer Arithmetic, 2007. ARITH '07. 18th IEEE Symposium on 25-27 June 2007 Page(s):217 - 224.
25. Jaewook Chung; Anwar Hasan, M.; Montgomery Reduction Algorithm for Modular Multiplication Using Low-Weight Polynomial Form Integers. Computer Arithmetic, 2007. ARITH '07. 18th IEEE Symposium on 25-27 June 2007 Page(s):230 - 239
26. Marcelo E. Kaihara and Naofumi Takagi; Bipartite Modular Multiplication Method; IEEE Transactions on Computers, Vol. 57, NO. 2, February 2008.