UDC 004.056.5

FAST SECURE CALCULATION OF THE OPEN KEY CRYPTOGRAPHY PROCEDURES FOR IOT IN CLOUDS

A. Mirataei, M. Haidukevych, O. Markovskyi

The article proposes a method for accelerating the implementation of cryptographic data protection mechanisms on embedded IoT terminal microcontrollers, the basic operation of which is the modular exponentiation of high-capacity numbers. The method is based on the use of remote computer systems to speed up calculations and provides protection against the reconstruction of the secret keys of cryptosystems from data transmitted to the cloud. It has been theoretically and experimentally proven that the method allows, on average, 50 times, to speed up the implementation of cryptographic data protection protocols in IoT while providing a level of security sufficient for most practical applications.

***Key words*:** *Modular exponentiation, secure cloud computing, IoT security, RSA cryptosystems.*

Introduction

The dynamic development of Internet technologies led to the emergence and rapid spread of remote-control systems for real-world objects. Such systems are called the Internet of Things. Real-world objects are directly controlled using portable embedded microcontrollers that are equipped with radio modems for Internet communication with a central computer. According to [1], today the number of such microcontrollers significantly exceeds the number of personal computers.

At the same time, there is a steady trend towards expanding the scope of the Internet of Things. For most areas of application of the Internet of Things, the issue of ensuring reliable protection against external influence is critical. The potential threat of such influence is due to the fact that data exchange is carried out directly through the open Internet. Accordingly, to ensure reliable protection of the Internet of Things, it is necessary to realize the possibility of fully using cryptographic protocols for information protection [2].

The vast majority of existing information protection protocols involve the use of public key cryptography. The basic computational operation of such cryptography is modular exponentiation, which is performed with large numbers. In particular, currently this operation uses 2048-bit numbers with the prospect of growth to 4096 in the coming years. Performing a modular exponentiation operation on numbers of such a digit size requires a significant amount of computing and time resources. Embedded portable low-bit microcontrollers of the Internet of Things do not have enough computing power to implement such complex calculations in real time. In recent years, the most acceptable way to solve the problem of the shortage of computing power is to use cloud technologies [3]. Such technologies provide access to significant computing resources of remote computer systems, including those that have modular exponentiation hardware.

Direct use of these resources by microcontrollers of the Internet of Things to implement cryptographic protocols is impossible, since these calculations contain secret keys. Accordingly, there is a need for such an organization of modular exponentiation, in which the majority of the volume of calculations is performed on a remote computer system, while it is practically impossible to recover the secret keys based on the data provided to it.

Thus, the scientific task of developing a method of secure modular exponentiation in the cloud to accelerate the implementation of cryptographic protection protocols in the Internet of Things is relevant for the current stage of information technology development.

Problem statement and review of methods for its solution

In recent years, the use of cloud technologies has become the dominant approach to solving the problem of the shortage of computing power in solving applied problems [3]. These technologies allow using the Internet to provide a wide range of users with significant computing power of modern multiprocessor computer systems to quickly solve their applied problems. In fact, within the

framework of cloud technologies, the effect of virtualization of the availability of significant computing power to users is achieved. This approach not only increases the capabilities of users by orders of magnitude, but also ensures the economic efficiency of creating high-capacity computer systems [4].

One of the main disadvantages of cloud technologies, which significantly limits their practical application, is the possibility of unauthorized access to user data during their processing on remote computer systems [5].

Therefore, in recent years, work on the creation of homomorphic data encryption methods has been carried out on a broad front [6,7]. Unlike traditional data encryption, which is widely used in data transmission and storage, homomorphic encryption allows you to perform certain types of processing of encrypted data with the possibility of restoring the correct processing results through decryption [8]. Until now, schemes of the so-called full homomorphic encryption have been proposed, in which multiplication and addition operations can be performed on the encrypted data [9]. An important point here is that full homomorphic data encryption actually has the properties of universality, as it can be applied to a sufficiently wide range of data processing tasks, which are reduced to addition and multiplication operations. However, the existing fully homomorphic encryption schemes have significant computational complexity and are not yet in practical use, although quite intensive research is being conducted to create fully homomorphic encryption methods acceptable for practical use [10].

In practice, specialized homomorphic encryption schemes are used to protect data and the process of their processing on remote computer systems, which are focused on a specific procedure for remote data processing. Accordingly, for the operation of modular exponentiation $A^E mod M$ - the basic operation of modern cryptography, specialized methods of protection of secret elements, A and E, have been developed [11].

The vast majority of known methods of protected modular exponent calculation are based on the additive decomposition of the exponent code E into components, part of which is used for exponentiation on remote computing power, and part of which is used to perform this operation on the terminal microcontroller. This allows you to protect the code of the exponent E from reconstruction attempts based on the data that is transmitted to the cloud. Other mechanisms are used to protect against attempts to reproduce the number A [12].

In [12], a method of remote calculation of the modular exponent based on the random division of the exponent code E into groups of digits is proposed. This allows the computation of $A^E mod M$ to be organized as a modular product of modular exponents that can be computed independently on multiple remote computer systems. To protect the number A, which is raised to a power, its multiplication by a secret number is used, which is selected in such a way that the result of its modular raising to the power E is equal to one.

A significant advantage of the considered method is that the capabilities of multiprocessor systems can be fully used for the parallel calculation of partial modular exponents. In [13], based on theoretical and experimental data, it is shown that the described method can actually increase the performance of modular exponentiation by approximately three times.

The work [14] describes the method of secure calculation of the modular exponent based on the logical decomposition of the code of the exponent A. This approach allows you to speed up the calculation of partial exponents by reducing the number of modular multiplication operations by a constant number. At the same time, it is possible to transfer intermediate data from remote computer systems, which allow to significantly speed up the calculation of the components of modular exponentiation on the terminal processor. To protect the number A, its multiplication by a special number is applied, for which a modular inversion is determined in advance.

It is known about a group of methods for the secure calculation of the modular exponent, which is based on the fact that the multiplicative components of the module M are known, which makes it possible to organize a secure calculation by adding to the code of the exponent a number that is a multiple of the Euler secret period [15].

Another interesting method is described in [16]. The main emphasis in these studies is on the fact that the user can not only remotely perform modular exponentiation in closed mode, but also indirectly control the correctness of the performed operations.

The analysis of modern technologies for secure computation of the modular exponent on remote computer systems showed that their significant drawback is that different cryptographic mechanisms are used to protect both secret components of this operation.

Purpose and objectives of research

The aim of the study is to increase the speed of implementation of cryptographic information protection mechanisms on terminal microcontrollers of computer control systems in real time by organizing the secure execution of the basic operation of these mechanisms - modular exponentiation on remote computer systems.

To achieve the set goal, the following tasks are solved in the framework of this work:

- a review of existing methods for the secure calculation of the modular exponent on remote computing systems, an analysis of the possibilities for improving their efficiency;
- development of a method for accelerating the calculation of the basic operation of a wide range of cryptographic algorithms with a public key - modular exponentiation on terminal microcontrollers through the use of remote powerful computer systems based on the multiplicative-additive decomposition of the exponent code, which is the secret key of cryptosystems;
- theoretical and experimental evaluation of the effectiveness of the developed method based on the criterion of the level of protection from the reconstruction of secret components to data transmitted to the cloud, as well as the criterion of the achieved acceleration of the implementation of cryptographic data protection protocols on low-power portable terminal microcontrollers of computer control systems for real world objects.

The method of protected modular exponentiation in the cloud based on multiplicative-additive exponential decomposition

To achieve the goal, the method of protected modular exponentiation $A^E \bmod M$ on remote computing capacities has been developed, which is based on the multiplicative-additive decomposition of secret code of the exponent E, that is its representation in the form $E=F\cdot a+k$. At the same time, the F code is transmitted to the remote system, and the components k and a are used only on the terminal microcontroller. The main advantage of the proposed approach is the possibility of simultaneously solving two problems: the practical impossibility of illegal reconstruction of the secret codes of the exponent E and the number A by the code F and the open value of the module M.

The expansion of the exponent is carried out by selecting two random parameters a and k. Moreover, the number a is chosen so that its bit rate m_a does not exceed 5, and the value k must satisfy the condition $(E-k) \bmod a = 0$. Since the exponent code E is part of the private key, in real public-key information security protocols, the numbers E and M rarely change, so they can be considered constant. This allows you to perform the selection of parameters a and k described above only once during key generation.

The proposed method of secure remote calculation of the $A^E \bmod M$ modular exponent involves the following sequence of actions:

1. The modular exponentiation operation $G=A^a \bmod M$ is performed on the terminal microcontroller.
2. The result of the calculation of G, as well as the numbers F and M are sent to a remote computer system.
3. The calculation of the modular exponent $W= G^F \bmod M$ is implemented in the cloud and the result is returned via the Internet to the terminal microcontroller.
4. At the same time, the modular exponent $Y=A^k \bmod M$ is calculated on the terminal microcontroller.
5. After obtaining the W value from the cloud, the modular product of the W and Y values is calculated on the terminal microcontroller: $W\cdot Y \bmod M$.

The operation of the proposed method of protected modular exponentiation in the cloud based on the multiplicative-additive expansion of the exponent can be illustrated by the following numerical example. Let the module M=143, as well as the public key D=7 and the private key E=103 be formed at the stage of generating a cryptosystem with a public key. The latter is used as a secret key of the terminal microcontroller of the system of remote control of objects in the real world. The next step is to expand the exponent: $E=F\cdot a+k$. Let the chosen value of a be equal to 3. According to the above, the value of variable k should be selected in such a way that (E-k) mod a=0. One of the possible options satisfying the condition can be k = 4, since (103-4) mod 3 =0.

As part of the current example, the modular exponent 80^{103} mod 143=115 is calculated, respectively A=80, E=103, M=143. According to clause 1, the auxiliary quantity $G=A^a$ mod M = 80^3 mod 143=60 is calculated. In accordance with clause 2, the obtained number G=60 and the values F=33 and M=143 are sent to a remote computer system to calculate the modular exponent $W=G^F$ mod $M=60^{33}$ mod 143=125. The result of W=125 is returned to the terminal microcontroller. At the same time, using the terminal microcontroller, the operation of modular elevation of the number A to the k power is implemented: $Y=A^k$ mod $M=80^4$ mod 143=81. According to clause 5, the modular product of the values of W and Y is calculated: $R=W\cdot Y$ mod M =125·81mod143 =115. The final result of the calculations is the code R=115.

As follows from the above, the proposed method allows you to quickly calculate the basic operation of public key cryptography on a low-power portable terminal microcontroller through the use of powerful remote computer systems. At the same time, no codes are transmitted to these systems that allow recovering the values of the secret codes of the modular exponentiation operation. An important element of the proposed method is that the operation of modular exponentiation is also implemented on remote computer systems. This makes it possible to use cryptoprocessors for the fast implementation of this operation, which allow speeding up the execution of modular exponentiation by 2-3 orders of magnitude.

The constructiveness of the proposed method can be proved as follows. Considering that the value calculated in the cloud is $W=G^a$ mod M = $((A^F)^a$ mod M = $A^{F\cdot a}$ mod M. Operations performed on the terminal microcontroller can be represented as: $Y=A^k$ mod M, $R=W\cdot Y$ mod M = $A^{F\cdot a}\cdot A^k$ mod M = $A^{F\cdot a+k}$ mod M. Since $E=F\cdot a+k$, that $R=A^E$ mod M, which was to be proved.

The developed method differs in that a single mechanism is used to encrypt the secret code of the exponent and the number A raised to a power - the multiplicative-additive decomposition of the exponent code. The use of a single cryptographic mechanism for encrypting the two secret components of the modular exponentiation operation makes it possible to increase the proportion of operations performed on remote computer systems. As a result, a greater acceleration of this operation, which is important for cryptographic applications, is achieved in comparison with known approaches. At the same time, a high level of security is maintained, based on the analytically unsolvable discrete logarithm problem.

The above feature determines the originality of the completed development. Another important feature of the proposed method is that it allows you to simultaneously perform the remote exposure operation on encrypted data, which is carried out on remote computer systems, and the exposure operation using an additive component on the terminal microcontroller. This solution makes it possible, on the one hand, to increase the speed of implementation of cryptographic protection mechanisms with a public key, and, on the other hand, allows choosing the value of the additive component of the exponent code within a wide range. The latter provides a higher level of protection provided against brute force attempts to select the value of the additive component. In known methods, expanding the range of possible values of the additive component of the exponential code necessarily entails a decrease in the computational speed. In the proposed scheme, the expansion of the range of possible values of the additive component due to the organization of parallel computing has practically no effect on the time of computing the modular exponent.

Evaluation of the developed method effectiveness

It is advisable to evaluate the effectiveness of the proposed method according to two criteria: determining the performance indicators and the level of protection against the attempts of the villain

to reconstruct the value of the exponent E and the number A based on the data transmitted to the cloud.

The level of protection is determined by the amount of time resources that the villain must spend on trying to recover the secret code of the exponent and the number A. Since the public key D is known, the attacker can determine the values of X and U such that $X^E \bmod M=U$ by calculating $U^D \bmod M = X$. Accordingly, to find the secret code of the exponent, the most appropriate tactic of the attacker is the selection of unknown parameters a and k. It is quite obvious that the T_{CR} time for implementing such a selection depends on the number of possible options for the values of parameters a and k, and the time T_{EXPcl} – the calculation of the modular exponent in the cloud. The numerical value of T_{CR} is determined by the formula:

$$T_{CR} = 2^{m_a+m_k} \cdot T_{EXP\_CL} \tag{1}$$

It follows from formula (1) that the required level of protection, that is, the amount of time resources for its violation, can be ensured by the appropriate selection of the value of the parameter k.

The technology of such flexible management of the security level when using the proposed method can be illustrated by the following example. Suppose that gaining access to the private key E of the terminal microcontroller of the remote object control system allows the villain to obtain V_{SRC} benefits valued at \$1,000,000. Accordingly, to ensure reliable protection, it is necessary that the C_{CS} cost of the amount of resources spent on selecting parameter k exceeds the benefits, that is, it should be at least \$1,000,000. Assuming that the cost of renting the C_{CS} time of the remote computer system is \$100/hour, the specified condition is fulfilled for T_{CR} values determined from the equation:

$$T_{CR} = \frac{V_{SRC}}{C_{CS}} = 10^4 \ hours \ . \tag{2}$$

Determining the numerical value of the T_{EXP_CL} time for calculating the modular exponent can be done based on the fact that nowadays almost all computer systems, including laptops, have hardware to quickly perform this important operation in the form of a built-in cryptoprocessor. In particular, the Hi/fn 7955 cryptoprocessor provides modular exponentiation of 2048-bit numbers in the time T_{EXP_CL}=0.083 s.= $2.3 \cdot 10^{-5}$ hours. Taking into account the determined numerical values of T_{CR} and T_{EXP_CL} included in formula (1), it is possible to obtain:

$$m_k + m_a = \log_2 \frac{T_{CR}}{T_{EXP\_CL}} = \log_2 \frac{10^4}{2.3 \cdot 10^{-5}} = 29 \tag{3}$$

Hence, the number m_k of binary digits of the number k is determined as 29-2 =27. That is, to ensure the level of protection defined in the current example, the bit number k must be at least 27.

As an indicator of speed, the acceleration factor β is most often used, which is determined by the ratio of the time T_M of calculating the modular exponent on the terminal microcontroller to the time T_C of performing this operation with the involvement of remote computing power according to the proposed method:

$$\beta = \frac{T_M}{T_C} \ . \tag{4}$$

When calculating the modular exponent by the classic method, the calculation time is T_M=$1.5 \cdot n \cdot T'$, where n is the number of bits of the exponent, and T' is the time of modular multiplication of n-bit numbers [7]. Under the condition of using the Montgomery scheme for modular multiplication, the time T' is $2 \cdot n^2 \cdot t/r$, where r is the bit rate of the processor, and t is the time the processor executes one instruction of the addition type. Thus, the formula for calculating the modular exponent has the following form:

$$T_M = \frac{3 \cdot n^3 \cdot t}{r}. \quad (5)$$

It is clear from this formula (4) that there is a cubic dependence of T_M time on bit rate n.

The time T_C of calculating the modular exponent according to the proposed method depends on three components: the duration of the operation $A^a \bmod M$ on the terminal microcontroller, the maximum value between the time of calculating the modular exponent in the cloud and the time of implementing the modular raising of the number A to the k power, as well as the time T' of the modular multiplication $R=W \cdot Y \bmod M$. Accordingly, the value of T_C can be represented in the form of a formula:

$$T_C = T_{EXP\_a} + \max(T_{CL}, T_{EXP\_k}) + T'. \quad (6)$$

The duration of T_{EXPa} according to the classical method of calculating the modular exponent is $T_{EXPa}=1.5 \cdot m_a \cdot T'$. Similarly, the execution time of the $A^k \bmod M$ operation is $T_{EXPk}=1.5 \cdot m_k \cdot T'$.

When implementing the modular exponentiation operation on a remote computer system, the T_{CL} calculation time is:

$$T_{CL} = 2 \cdot T_{DT} + T_{EXP\_CL}, \quad (7)$$

where $2 \cdot T_{DT}$ is the time of transferring data to the cloud and returning results, T_{EXP_CL} is the time of $G^F \bmod M$ execution. In fact, the duration of calculations in the T_{expF} cloud can be neglected due to the fast execution of calculations, therefore $T_{CL}=2 \cdot T_{DT}$.

Let $\max(T_{CL}, T_{EXPk})=T_{EXPk}$, then the T_C calculation formula can be written as follows:

$$T_C = 1.5 \cdot (m_k + m_a) \cdot T'. \quad (8)$$

Then the acceleration coefficient β can be represented as:

$$\beta = \frac{1.5 \cdot n \cdot T'}{1.5 \cdot (m_k + m_a) \cdot T'} = \frac{n}{m_a + m_k}. \quad (9)$$

In the framework of the above example, the secret code E of the exponent has a bit size of 2048: $n=2048$. As described above, the bits m_k and m_a are 24 and 5, respectively. Therefore, β = 2048 / 29 = 70.6. That is, within the framework of the considered example, the application of the developed method provides acceleration of modular exponentiation by 70.6 times.

The conducted experimental studies showed that the real speedup achieved by using the proposed method ranges from 50 to 100, depending on the requirements for the level of security.

6.Conclusion

As a result of the research aimed at improving the efficiency of information protection in computer control systems operating in real time and using the Internet as a data exchange medium, the following results were obtained.

A method for accelerating the calculation of the basic operation of a wide range of cryptographic algorithms with a public key - modular exponentiation on terminal microcontrollers through the use of remote powerful computer systems is proposed and investigated. The method provides for the organization of protection of data transmitted to the cloud due to the multiplicative-additive decomposition of the exponent code, which is the secret key of cryptosystems.

It has been theoretically and experimentally proved that the developed method makes it possible to reliably protect the secret components of modular exponentiation when this operation is remotely implemented on potentially open computer systems. At the same time, the developed method makes it possible to speed up the implementation of cryptographic data protection protocols on low-power portable terminal microcontrollers of computer control systems for real world objects by almost two orders of magnitude.

The developed methods are focused on use in real-time computer control systems and can significantly speed up the implementation of cryptographic data protection protocols on low-power terminal microcontrollers of such systems.

References

1. Noot M.M. Current research on Internet of Things (IoT) / M.M. Noot, W.H. Hassan. // Compute Network. -Vol.148. -No.15. -2019. -pp.283-294.
2. Khan M.A. IOT security: Review, blockchain solution and open challenges / M.A. Khan, K. Salah // Future Generation Computer Systems. -Vol.82. -No.5. -2018. -pp.395-411.
3. Fox G. Using Clouds for Technical Computing / G. Fox, D. Gannon // Cloud Computing and Big Data. Amsterdam: IOS Press. -2018. -pp.81-102.
4. Boroujerdi N. Cloud Computing: Changing Cogitation about Computing / N. Boroujerdi, S. Nazem // IJCSI International Journal of Computer Science Issues, -Vol. 9, -Issue 4. -2012. -No.3. -pp.169-180.
5. Zhang L. Improving security and privacy attribute based data sharing in cloud computing / L. Zhang, Y. Cui, Y. Mu // IEEE Systems Journal. -2019. -Vol.14, -No.1, -pp.387–397.
6. Kaur C., Mourad H.,M., Banu S.S. Security and Challenges using Clouds Computing in Healthcare Management System. / C. Kaur, H.M. Mourad, S.S. Banu // International Jounal of Trend in Scientific Research and Development.- 2019.- Vol.3.- No. 6. –pp.44-52
7. Getov V. Security as a Service in Smart Clouds – Opportunities and Concerns / V.Getov // IEEE 36-th Annual Computer Software and Application Conference. Izmir, 16-20 July 2012.- pp.16-22.
8. Van Dijk M. Fully homomorphic encryption over the integers / M. Van Dijk, C. Gentry, S. Halevi, V. Vaikuntanathan // In Annual International Conference on the Theory and Applications of Cryptographic Techniques. -Springer, -2010, -pp.24–43.
9. Brakerski Z. Fully homomorphic encryption without modulus switching from classical GapSVP / Z. Brakerski // Annual Cryptology Conference. -Springer, -2012, -pp.868–886.
10. Cheon J.H. Homomorphic encryption for arithmetic of approximate numbers./ J. H. Cheon, A. Kim, M. Kim, and Y. Song // In International Conference on the Theory and Application of Cryptology and Information Security. Springer, 2017, - pp. 409–437.
11. Lui Z. Efficient ring-LWE encryption on 8-bit AVR processors / Z. Liu, H. Seo, S. S. Roy, J. Großschadl, H. Kim, and I. Verbauwhede // In International Workshop on Cryptographic Hardware and Embedded Systems. Springer, 2015, pp. 663–682.
12. Bardis N. Secure Implementation of Modular Exponentiation on Cloud Computing Resources / N. Bardis, O. Markovskyi // Proceeding of International Conference Applied Mathematics, Computational Science and Systems Engineering. Athens, Greece, October 6-8, -2017, -pp.90-96.
13. Kostenko J. V. Method protected modular exponentiation on remote computers systems / J.V. Kostenko, O.V. Rusanova // Bulletin of the National Technical University of Ukraine "KPI". Informatics, management and computer techniques. Kyiw.: „VEK+”. -№ 64. -2016. - pp. 51-54.
14. Bardis N. Secure, Green Implementation of Modular Arithmetic Operations for IoT and Cloud Applications / N. Bardis // In book Green IT Engineering: Components, Networks and System Implementation. -2017. -pp.43-64.
15. Markovskyi O. Secure Modular Exponentiation in Cloud Systems / O. Markovskyi, N. Bardis, N. Doukas, S. Kirilenko // Proceedings of The Congress on Information Technology, Computational and Experimental Physics (CITCEP 2015), 18-20 December 2015, Krakow, Poland, -2015. -pp.266-269.
16. Xiaofeng Chen New Algorithms for Secure Outsourcing of Modular Exponentiations / Chen Xiaofeng, Li Jin, Ma Jianfeng, Tang Qiang, Lou Wenjing // ESORICS 2012, LNCS 7459, -2012. -pp.541–556.