MINISTRY OF DEFENSE
DEFENSE INSTITUTE
„PROFESSOR TSVETAN LAZAROV"

ABSTRACT

of dissertation for awarding educational and scientific degree **"Doctor of Philosophy"**

on the topic: **"The Resilience Analysis of Public Key Cryptographic Systems"**

field of higher education:	5. Technical sciences
professional field:	5.2 Electrical engineering, electronics and automation
in PhD program:	Automated systems for processing information and management
PhD candidate:	**eng. Andrey Georgiev Ivanov**
thesis supervisor:	**col. assoc. prof. PhD Nikolai Todorov Stoianov**

Sofia 2022 r.

The dissertation consists of 144 pages, of which 106 pages are the main part and 21 pages are the appendix. The main part of the dissertation includes: 10 figures, 25 described algorithms, 114 literary sources and scientific publications.

The numbering of chapters, figures, tables, formulas and cited literature in the abstract corresponds to that in the dissertation. The numbering of the literature used in the abstract corresponds to the numbering of the literature used in the dissertation.

The defense of the dissertation thesis will be held on _____ at _____ hours in room _____ at the Defense Institute "Professor Tsvetan Lazarov", at an open meeting of the scientific jury:

1. _____
2. _____
3. _____
4. _____
5. _____

Reserve members of the jury:

1. _____
2. _____

Author:
eng. Andrey Georgiev Ivanov

Thesis supervisor:

col. assoc. prof. PhD Nikolai Todorov Stoianov

Topic:

"The Resilience Analysis of Public Key Cryptographic Systems"

INTRODUCTION

Throughout history, humanity has distinguished itself from other living organisms on the planet by constantly improving its specific skills. These skills are a part of all aspects of life and activity. Humans create and develop their tools for work, their ways of thinking, and their methods of communication. In the beginning, human individuals used mainly verbal methods of communication and relatively simple, monochromatic pictures. As their skills and knowledge developed, they created better and more advanced mechanisms for gathering information and communicating with each other. So, in our modern day, we use electronic means of communication, information storage, and processing on a daily basis.

Throughout the centuries, people have realized that the information they possess, its reliable storage and preservation, plays an extremely important role in their individual and social advantage compared to others. For this reason, means have been and are being created to provide reliable and secure protection of information with different volumes and structures. The most widely used method of protecting information is encryption. Cryptography (from Greek kryptos - "hidden" and graphein "write") studies the principles, means, and methods of transforming data to conceal its semantics or to protect against unauthorized access. It is also a tool used to protect against changes in the content or structure of information by individuals whose access to the actual content should be restricted.

This is achieved through a mathematical algorithm where the data undergoes a change based on at least one secret parameter called a "key" (cryptographic key), known only to the participants in the communication or the people to whom access to it is delegated.

The development of technology, particularly electronics, in the last century has necessitated the use of cryptography at the individual level. Today, people use technical electronic devices in their daily activities, where the implementation of cryptographic algorithms and methods for protecting information is mandatory and already considered a natural occurrence for these protective means to be used in almost all areas of human activity.

Two main groups of algorithms exist in cryptography, based on their mathematical foundations and implementation. Their distinction is based on the size, characteristics of the key, and the way it is used to protect data. These two groups of algorithms are:

- **Symmetric Key Cryptographic Algorithms.**
- **Public Key Cryptographic Algorithms.**

The widespread use of public key cryptography in the everyday life of a modern person, with the goal of securely and reliably protecting personal, public, social, and state data, transforms it into a significant factor for the functioning of society, individual countries, and the world as a whole.

All that has been stated so far proves the great importance of cryptography systems for protecting information, through the use of a public key encryption algorithms. The development of the dissertation work is motivated by the existing issue of determining the level of resilience in the functioning of public key cryptography systems.

Proofs of vulnerabilities in public key cryptography systems used and implemented are increasingly being published in the internet. Here, the question arises as to how many of the established vulnerabilities are officially reported so they can be remediated. With the growth of information technology, large companies, government organizations, and hacker groups (which may pose as **ill-wisher**) aim to secure easier unauthorized access to information that is not their own. The most unobtrusive and easiest way to achieve this is through compromising cryptography systems. Ill-wishers strive to achieve this by using previously created back doors (algorithmic, mathematical, etc.) to access private keys (used in public key cryptography), which can lead to the revelation of explicit information (the original form of protected information).

These reasons for the existence of the possibility of unauthorized access to restore private keys in a public key cryptography system assume that their mathematical foundations and functioning principles should be evaluated. It is necessary to summarize and analyze existing hardware tools and software libraries used for cryptographic purposes in practice. The indicators and parameters of the studied cryptography tools must be analyzed and evaluated, and based on the results, it must be determined if there is a hidden possibility of them being compromised through the rapid and easy recovery of the private key, which directly affects the resilience of the functioning of cryptography systems using public key cryptography.

The objective of this work is to examine and analyze the most widely applied algorithms and approaches used in public key cryptography for the purpose of protecting information. To assess the degree of their resilience against attacks related to the presence of weaknesses that may exist as a result of deliberately created vulnerabilities in the core cryptographic primitives of the respective algorithm. To develop models that serve as a basis for demonstrating the probable possibility of the existence or non-existence of vulnerabilities of this kind.

The dissertation work is structured: introduction, three chapters, conclusion, literature, list of publications related to the dissertation, list of terms used, mathematical symbols and basic functions, list of figures.

In the first chapter, the mathematical and algorithmic foundations of public key cryptography are examined, its application and areas of use are described, the standards and requirements applied in practice are systematized for systems using public key cryptography. An assessment is made of the problems associated with the resilience of the functioning of cryptographic systems with public key cryptography and the following are formulated: the aim and objectives in carrying out the scientific research work being developed.

In the second chapter, two solution models are proposed. The first is implemented to achieve higher efficiency and reliability in determining the divisibility of a number by using the Miller-Rabin probabilistic algorithm. The second solution model presents the mathematical possibility for a more efficient implementation of the Silver-Pohlig-Hellman algorithm, which is used to attack the most widely used cryptographic algorithm in public key cryptography, RSA, at the present time.

In the third chapter, the mathematical foundations and model of a new approach for implementing a Kleptographic algorithm are described. The following is described: its practical implementation, generation of public key domains, assessment of its

technical implementation parameters, and a comparative analysis between the proposed algorithm and other existing Kleptographic algorithms. In this chapter, an analysis and evaluation of the possibility of the existence and use of Kleptography in practical applied systems based on public key cryptography are made. Conclusions are drawn and a mechanism is proposed to avoid vulnerabilities based on Kleptographic attacks on RSA-based systems.

In the conclusion, the contributions of the dissertation are described, as well as directions for future work.

The following limitations were adopted to fulfill the main tasks in the dissertation work:

- Only the RSA algorithm used in public key cryptography has been examined and analyzed for resilience.
- Only the possibilities of influencing the resilience of public key cryptographic systems related to RSA key generation and the attack possibilities for breaking it have been examined, analyzed and evaluated.

CHAPTER ONE

PUBLIC KEY CRYPTOGRAPHY: FUNDAMENTALS, ALGORITHMS, APPLICATIONS AND AREAS OF USE. RESILIENCE OF OPERATION.

Public key cryptography is a means of protecting information that is based on extremely difficult mathematical problems, the difficulty of which remains high even in the presence of rapidly functioning and well-organized modern computing technology in large clusters. To achieve this goal, mathematical operations that are part of the "Number Theory" section are used in the implementation of cryptographic algorithms that fall within this area of cryptography [8, 41].

1.1 The foundation of public key cryptography. Encryption algorithms. Areas of use. Resilience during operation.

The foundation for creating algorithms in public key cryptography is the following hard-to-solve mathematical problems: "Factorization of large numbers into prime factors" (Number Factorization), "Computing the discrete logarithm in finite fields" (Discrete Logarithm in Finite Field DLP), and "Computing the elliptic curve discrete logarithm" (Elliptic Curve Discrete Logarithm Problem - ECDLP). The mathematical primitives used for the purposes of public key cryptography are related to calculations and operations in mathematical structures of the multiplicative and additive Abelian groups type [74, 95].

1.1.1 Foundation of public key cryptography. Encryption algorithms.

Computing the discrete logarithm in finite fields is a problem where if a is a primitive element (generator) of the finite field F_p and d is an arbitrary element of F_p ($d \in F_p$), it is difficult to calculate x given d, a and p, such that: $d = a^x \bmod p$. This problem is computationally difficult because there is a periodicity and there are a large number of values of $x \in Z$ for which the equation: $d = a^x \bmod p$ holds true. This is due to the fact that the equation: $a^x = a^{x+k(p-1)} \bmod p$ holds true for all $k \in Z$.

The following algorithms are presented in this subsection of the dissertation:

Algorithm of Diffie-Helman …
Algorithm of ElGamal …
RSA …
DSA …
Elliptic Curve Cryptography (ECC) …

1.1.2 Use areas of public key cryptography.

Biggest boost for the adoption of public key cryptography is the development of technologies used in the Internet. This is logical and due to the fact that there is an increasing need for secure information exchange between unknown parties in communication. Security that reaches desired levels of safety is achieved through the use of a combined approach that uses public key cryptography for digital signing or protection of cryptographic keys which are used in symmetric key cryptography.

The storage of cryptographic keys used in public key cryptography is achieved by building a suitable organization and infrastructure. Such a form of organization is called a "Public Key Infrastructure" (PKI) [100, 101, 112]. In it, the keys are stored in the form of digital certificates.

The goal of PKI is to facilitate secure electronic transfer of information for a range of network activities such as e-commerce, online banking, confidential email and many others. PKI is required for activities where passwords are an inadequate method of authentication and more stringent proof of identity of the parties involved in the communication and validation of the transferred information is sought.

The PKI cryptography is an agreement that links public keys to their respective identities (such as people and organizations) and serves as a secure and reliable storage of issued digital certificates. The binding is established through the process of registration and certificate issuance by a certificate authority (CA). Every registration authority is responsible for one or more of the following functions:

- Identification and authentication of certificate candidates.
- Approval or rejection of certificate applications
- Initiating the revocation or suspension of certificates under certain circumstances
- Processing requests from subscribers for revocation or suspension of their certificates
- approve or reject requests from subscribers to renew their certificates

The content and structure of a digital certificate is standardized using X.509 (the "International Telecommunication Union" - ITU standard). Each certificate must be signed only by one entity (the CA). Each CA has its own unique root certificate. There may be a hierarchical relationship between different CAs.

One of the most common uses of digital certificates is to authenticate an electronic page. This principle arose as a means of combating fraud in the internet space, implemented through fake internet pages created for the purpose of gathering personal or sensitive information.

In order to ensure the best security in this area, some CA's have created and joined the "Certificate Transparency organization" (CTo) [87, 102, 114]. The CTo brings transparency to the SSL/TLS system that maintains the network. The SSL/TLS protocols implement cryptographic operations through the use of public cryptography.

Using CTo statistics (Figure 1), we can gain a real understanding of the number of certificates used in the internet space.

As of December 2021, there are over hundreds of millions of certificates. The following two figures from the global statistics are noteworthy:

- Number of newly generated certificates - 256 788 certs/hr
- Number of expired certificates - 239 840 certs/hr

The difference in the quantity is 16,948 certificates per hour in favor of the newly issued certificates [7, 89].

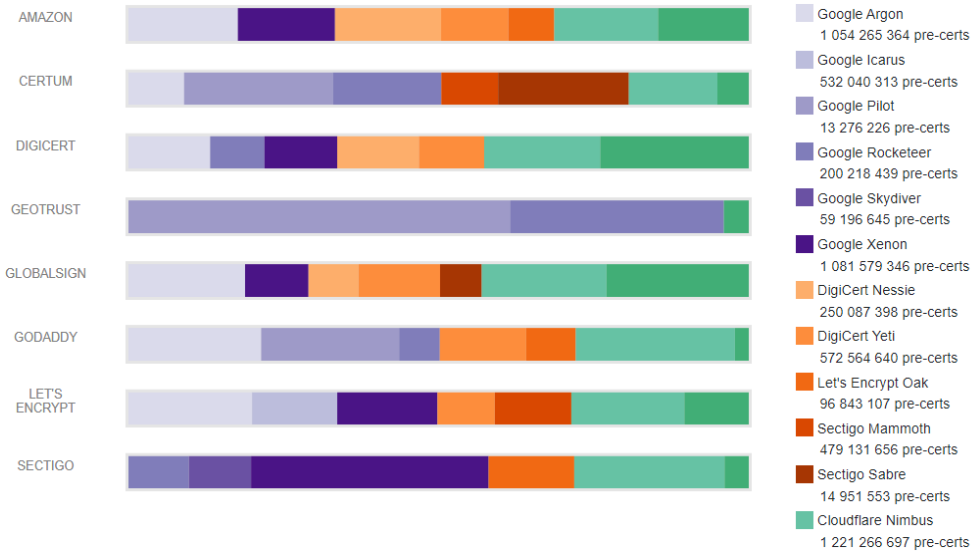

Fig. 1

Number of digital certificates in real use

https://ct.cloudflare.com/

One aspect of security in the internet space is the use of digital certificates to sign software code. Another important use is the safe provision of updates and corrections to existing software. Operating systems like Windows, Mac OS X, and most Linux distributions provide updates by signing the code to ensure that third parties cannot deliberately spread code that would compromise the security of various electronic systems. Digitally signed code is a means of verification, even if it is delivered by a third party.

Certification based on digital certificates is the answer to preventing fraud in machine-to-machine communication. Therefore, organizations such as banks and other financial institutions can communicate with each other internally and build trusted communications between them, ensuring that no fraudulent systems or malicious software can take control over their or shared infrastructure.

The number of governments around the world that are introducing electronic systems for registering and digitally identifying their citizens in national identification systems is increasing. The reasons for this are to improve access to services, national security, combat corruption and others.

The mutual recognition of digital signatures in the European Union allows for the conclusion of agreements and trade deals between governments and/or between companies. The Regulation of Electronic Identification and Trusted Services (eIDAS 910/2014/EC) simplifies and standardizes the identification data, including digital

identities and digital and electronic signatures. It creates a "single digital market" to ensure secure digital transactions between EU member states and countries with which they have international relations.

In general, the fields of application and use of public cryptography can include: secure and safe communication in the internet environment, protection of personal identity, software security, secure communication in the growing machine-to-machine communication, digital identity of the population and many others.

As of December 2021, the adoption rate of RSA is approximately 93%. This result is based on statistics obtained through the verification of root certificates in the distributions of major operating systems (Windows, Linux/Android, and MAC OS), as well as statistics provided by CTOs, as shown in Figure 2.

Fig. 2

Ratio between RSA and ECDSA based digital certificates in use
https://ct.cloudflare.com/

1.1.3 Keys used in public cryptography. Key generation, standards and the risks for the cryptographic system resilience.

For every cryptographic system, one of the most important elements of security is the user key that is used to encrypt information. This is why the key generation process is always an important stage in data protection. The influence this process has on the security and resilience of public-key cryptographic systems is huge. The questions related to the resilience research of cryptographic systems using public cryptography are related to the processes of generating and storing the private keys used by the users [10, 24, 30, 31, 37, 67]. If a cryptographic system allows for the generation and use of cryptographic keys that can easily be compromised, it is considered unstable [75, 76, 78].

The large extent of the use of RSA in practice, highlighted as a result in the previous subheading, is the reason for the research process on the resilience of public-key cryptography systems to be focused on evaluating the resilience of RSA.

It is important for the proper and stable functioning of public-key cryptography-based cryptographic systems to apply standards in the processes related to key generation [92, 110]. The generation of RSA cryptographic system keys is regulated by standards that are applied in the process of creating the two parts of the key [84, 90, 96].

The two standards most widely used in practice are developed by the National Institute of Standards and Technology (NIST) in the US and the European Telecommunications Standards Institute (ETSI), which specify a series of steps and conditions that numbers composing the key used in RSA cryptographic systems must comply with.

In practice, key generation is implemented through software using computer configuration applications or a specialized hardware module with a communication interface. The checks for compliance with the above-mentioned standards are performed in specialized laboratories and each key generation tool has a validity period for use.

Despite the requirements of the standards and the strict control over the work of hardware devices and software applications used for encryption, there is a risk that the key generation process may be compromised.

In RSA-based cryptographic systems, it is a crucial component that p and q are chosen with sufficient randomness to guarantee the security of the public key. Decomposing a large N modulo to obtain p and q is not possible under normal circumstances. However, if the keys are generated with poor randomness (not fully random factors generated according to a completely random principle), then it is possible that two RSA public keys will share a factor. And if enough keys are generated, the resulting modular numbers N_i can be decomposed by searching for their common divisor $(GCD(N_i, N_J)\ i \neq j\ \ N_i \neq N_J)$ [46, 108].

In 2012, a group of researchers including Lenstra [46] conducted a study in which they analyzed approximately 6.2 million digital certificates that are widely used in the internet space and found that about 4.3% of them had shared factors. During the analysis of 45 million modulus values, elements of RSA keys, scanned between 2015-2017 (from actually functioning digital certificates), using the first million small primes, 192,709 keys were broken. This number corresponds to 344,055 different certificates in the original data set, or 0.56%.

This statistics and analysis show a real problem in the principle and approach for generating factors for modular numbers used in certificates intended for RSA cryptographic systems.

An example of a problem with a standardized and certified as functional hardware device or software product is the "Trusted Platform Module" (TPM) produced by Infineon. The problem with Infineon chips is discovered during operation, not certification. TPM modules are electronic chips that generate and store keys for cryptographic algorithms. In the specific example of chips produced by Infineon, affected chips are those with software versions: 4.0-4.33; 4.4-4.42; 5.0-5.61; 6.0-6.42; 7.0-7.61; 133.0-133.32; 149.0-149.32. They are mainly used in computer configurations and affect the security of Windows and Linux operating systems. The most important aspect of the example is the manifestation of the vulnerability: **"The vulnerability allows the recovery of a private key when only the public key is available"**. The National Security Agency (NSA) publication states that this vulnerability affects to a significant extent device used by the Department of Defense. (Department of Defense - DoD) [81, 88, 109].

There is another group of approaches for attacking RSA-based cryptographic systems that are most difficult to establish and it is difficult to assign them to any of the other classical categories of attacks. These are attacks that are based on pre-laid mathematical constructions, which only in a specific combination between the mathematical model of the algorithm and the generated key create a vulnerability [40, 58]. The group of such types of attack methods is named "Kleptographic Attack Methods". The goal of kleptography is to create the conditions for an attack, through

which the realization of a vulnerability is possible only from the knowledge of the way the weakness functions in the mathematical and cryptographic primitives of the algorithm. The term "kleptography" was first proposed by Adam Young and Moti Yung in 1996 [3, 18, 19, 26]. One of their articles is entitled "Kleptography: Using Cryptography Against Cryptography" [20, 25, 39, 77].

At present, there is no strict classification of kleptographic attacks. This is due to the fact that there is no clear formal model of kleptographic mechanisms, and a wide spectrum of attacks against them is used. The general description of kleptography is a method that includes methods for constructing channels for secretly transmitting sensitive cryptographic information or developing cryptographic primitives with partial violations of cryptographic properties under certain conditions.

One of the striking examples known to the world community in 2021 is the secret agreement between the United States and Germany, which began after the end of World War II. The countries agreed and gained ownership of a Swiss company (Crypto AG), which developed and commercially offered cryptographic means to governments worldwide (over 60 countries). Part of the functionality of Crypto AG's cryptographic means allowed the familiar countries to quickly extract cryptographic keys and thus access the explicit data [80].

In this part of the dissertation work, the algorithm by Yung and Young is presented.

A Kleptographic Attack algorithm by Young and Yung Against RSA …

1.2 Goals and objectives of the dissertation work

The widespread use of the RSA cryptographic algorithm, as well as the presence of assumptions and examples demonstrating the potential for risk affecting its resilience, led to the development of a doctoral thesis aimed at exploring potential ways to create models and solutions that could affect its resilience and security of encrypted data. The existence of models and solutions that weaken or strengthen the resilience of RSA will greatly affect the security of encrypted data that is exchanged daily by millions of correspondents.

As a result of the analysis carried out, the goal of the doctoral thesis is as follows: to propose solution models that allow for the checking of existing objective opportunities that affect the resilience of cryptographic systems based on the RSA cryptographic algorithm.

The following tasks must be solved in order to achieve the set goal:

1) Analyze the algorithms used to evaluate the divisibility of a number, examine the principles of operation of the most widely used algorithms in this field. Evaluate their weaknesses. Analyze the possibility of solutions that would lead to increased efficiency and reliability of the results obtained and, if possible, to increase the speed of the divisibility evaluation process.
2) Existing RSA attack algorithms should be analyzed and the possibility of creating a formal model to increase the efficiency of prime factorization should be evaluated.
3) By analyzing the mathematical constructions of encryption algorithms and those for their attack, to confirm or reject the possibility of a real threat to the stability of the RSA function by creating a weakness in the process of generating the private and public key.

4) Create a functional model of a practical software implementation, based on which to analyze the efficiency of the results.

1.3 Conclusions.

- The application of algorithms for solving problems in number theory increases the complexity of implementing attacks against cryptographic primitives used in public cryptography.
- The most widely used encryption algorithm in practice is RSA, which uses public key cryptography. The existence of a vulnerability that could compromise the stability of a cryptographic system using RSA would threaten the security of a large number of communication and information systems and, as a result, would negatively impact large social groups as a whole.
- The two main approaches that provide a real chance of achieving an effective attack against RSA are related to the principles and methods of creating cryptographic keys. This includes the degree of reliability that the number factorization algorithms provide. The second approach is related to the presence of hidden mechanisms by the RSA user that can easily compromise the private key, thus having a negative impact on the resilience of RSA.
- Based on everything discussed so far, including the theory on which public-key cryptography algorithms are based, the ways they are used in practice, the principles and methods for generating and evaluating prime numbers as part of the keys intended for encryption in public cryptography, it can be concluded that there are risks to the resilience of public cryptography that can affect the security of encrypted data.
- The technological development of electronic devices is not the only factor contributing to the risks of unstable functioning of RSA-based cryptographic systems. The potential existence of the possibility of introducing kleptographic crypto-mechanisms may have a serious negative impact on all spheres that use encryption algorithms based on problems related to the decomposition of large prime numbers.

CHAPTER TWO

MODELS OF SOLUTIONS IMPACTING THE RESILIENCE OF RSA.

Two solution models affecting the stability of RSA-based cryptographic systems are considered in this chapter. The first one is related to creating a divisibility assessment model for numbers, aimed at improving the results when using the Miller-Rabin algorithm. The second solution model implements the idea of a more efficient attack on RSA-based cryptographic systems by using the attack algorithm proposed by Silver-Pohlig-Hellman [83, 86].

2.1 Solution Model Allowing for Improving the Estimation of Number Divisibility when Using the Miller-Rabin Algorithm

In public key cryptography, prime numbers are of utmost importance in achieving high security and reliable protection of encrypted information. They are the cornerstone of every public key used in the implementation of the RSA algorithm. That

is why this part of the dissertation work will focus on some important algorithmic aspects in the field of number theory ("Primality Tests"). This is done with the aim of gaining greater clarity on the relationship between the proposed model, which is meant to complement one of the most widely used algorithms in practice, the Miller-Rabin.

The ability to reliably determine the primality of a number has great and important significance for the security of encrypted data and the stability of cryptographic systems that use public keys [47, 48, 49]. To achieve the goals of cryptography, the size of the prime numbers used is very large, in the order of hundreds of digits in decimal form, which is equivalent to thousands of bits in binary form.

The determination of the divisibility of a number is a question that has occupied mathematicians since antiquity. In the beginning, attempts were made to synthesize and formulate formulas for calculating prime numbers as a function of input parameters, but with the development of number theory, it was established that there is no unique mathematical apparatus of functions that definitively applies to all existing prime numbers [44, 63]. As a result of this fact, the only possible approach is the application of an algorithm to check whether a number is divisible by composite factors or not (factors other than 1 and the number itself).

In this field, there are practically two main groups of algorithms: deterministic and probabilistic. Which group a given algorithm for determining the divisibility of a number belongs to depends on the degree of certainty of the answer.

Probabilistic algorithms

The probabilistic algorithms created over the years are numerous, and each one is characterized by an evaluation speed for divisibility that is multiple times faster than any deterministic algorithm. The probability of one number being evaluated as composite, even if it is actually prime, will not affect the security of protected data, as it will be disregarded for key participation. Unfortunately, the characteristic error of these algorithms is that composite numbers are determined as prime. If a large number p with a dimension of several thousand bits is evaluated as prime and is actually composite, a false assumption will be made about the size of the field F_p, in which all arithmetic, used in the encryption process, was implemented [9]. This in turn creates a scenario where if an adversary establishes divisibility of p and decomposes the number into prime factors, they will successfully carry out an attack against RSA.

The most widely used probabilistic algorithm in all modern systems that implement public cryptography is the "Miller-Rabin Primality Test" [82, 97].

The Miller-Rabin algorithm is based on the third approach, to find the number x for which the following are satisfied:

$$x^2 \equiv 1 \; mod \; p \text{ и}$$
$$x \not\equiv \pm 1 \; mod \; p \qquad (5)$$

The Miller-Rabin algorithm is based on a third approach, to find a number x such that it is called a witness for divisibility.

The principle of the idea for a solution model that allows for the improvement of results in the most widely used algorithm in practice for determining the divisibility of a number, that of Miller-Rabin, has been developed and presented at the DIGILIENCE 2020 conference (Varna) and published in the international publication "Information & Security: An International Journal" [71].

The solution model considers the possibility of a transition between individual multiplicative subgroups G_i, formed by their own generator g_i and the tested number p. The idea for this implementation of the proposed solution model is based on a heuristic algorithm that uses a linear Diophantine equation:

$$d_x \,.\, d_y - p.k = \, d_z \tag{6}$$

where $d_i = g^i \, mod \, p$ and every $d_i \, \in Z_p$ is an element of a ring with order $\#O_{(g,p)}$, obtained through a generator g. In the special case when $d_x = {d_y}^{-1} \, mod \, p$ the value of d_z is 1.

A heuristic approach has established that, with $d_z = 1$ and the order of the group $\#O_{(g,p)} < p - 1$, a new value of ${d_x}^*$ can be obtained, for which $\left(\frac{d_x}{p}\right) \neq \left(\frac{{d_x}^*}{p}\right)$. This new value of ${d_x}^*$, used as a generator, often forms a group with a set of elements that are completely or partially different from the set of elements in the group formed by the generator d_x. This can be achieved through the following iterations to find a suitable value of ${d_x}^*$:

Input:	$\boldsymbol{d_x, p}$
Output:	$\boldsymbol{{d_x}^*}$
(1)	${d_x}^* \leftarrow \, d_x$
(2)	$m \leftarrow \left(\frac{d_x}{p}\right)$
(3)	**do**
(4)	**solve:** ${d_x}^* \,.\, d_y - p.k = \, d_z$
(5)	${d_x}^* \leftarrow k$
(6)	**while** $m \neq \left(\frac{{d_x}^*}{p}\right)$

In practice, in order to quickly determine the divisibility of a number through the use of Miller-Rabin, not all x in the interval $1 < x < 4.log^2 p$, are tested, but random values are selected. When working with large numbers, the probability of selecting a number from the same set generated by a specific generator g is huge, which is a result of working with numbers that are very large in value and used for the modulus. This creates the possibility of permuting the set and the result of the test, through the Miller-Rabin algorithm, will be the same without contributing anything to the divisibility check. This is graphically shown in Figure 3.

If a mechanism is used where, instead of random generated values (after the first generator), derivatives of the first or any subsequent generator are used, we can obtain a new ring which elements set does not completely match or has a partial intersection with the set of elements of the previous ones. On that way the sets of numbers which can be tested could even completely cover Z_p. The number divisibility check can be carryout for a significantly smaller number of iterations by the Miller-Rabin method. This approach can greatly reduce the number of iterations required for the Miller-Rabin test and can greatly improve the efficiency of the algorithm. This model of iterative approach is illustrated graphically in Figure 4.

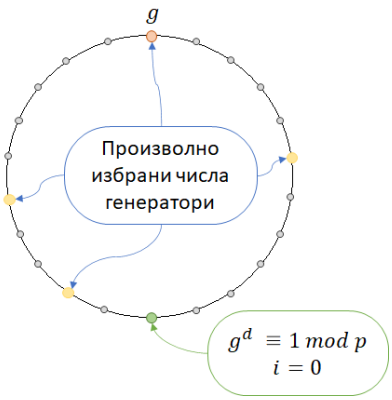

Fig. 3

Depicts the result that the selected numbers to be checked (in a standard Miller-Rabin test) are part of the same ring, which does not actually increase the credibility of the test

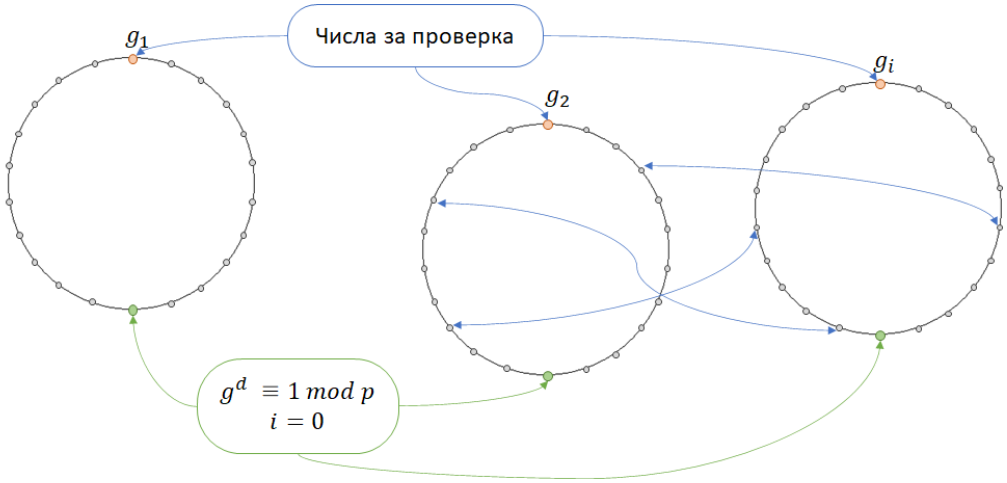

Fig. 4

Depicts the idea of using such numbers for verification using the Miller-Rabin test, which are part of rings with different composition elements

The steps of an algorithm implementing the proposed solution model (whose idea is graphically depicted in Figure 4) are the following:

Input:	number $\boldsymbol{p}$
Output:	COMPOSIT, PRIME
(1)	pick up small prime number q such that $a = p \, mod \, q > 1$
(2)	$b \leftarrow a^{p-1} \, mod \, p$
(3)	if $b > 1$f
(4)	output: COMPOSIT
(5)	$JacobiSymbol = 1$
(6)	$f \leftarrow (p-1)$
(7)	for each $i \in 0..4$
(7.1)	$b \leftarrow f \, / \, 2$
(7.2)	if $JacobiSymbol = 1$
(7.3)	$g \leftarrow \boldsymbol{GetPRU} \, (a, p)$
(7.4)	if $JacobiSymbol = -1$
(7.5)	$g \leftarrow \boldsymbol{getNotPRU} \, (a, p)$

(7.6)	$JacobiSymbol \leftarrow \left(\frac{g}{p}\right)$
(7.7)	if $JacobiSymbol = 0$
(7.8)	output: COMPOSIT
(7.9)	$LegendreSymbol \leftarrow g^b \bmod p$
(7.10)	if $JacobiSymbol <> LegendreSymbol$
(7.11)	output: COMPOSIT
(7.12)	while $(LegendreSymbol = 1)\ and\ (b \bmod 2 = 0)$
(7.12.1)	$b \leftarrow b\ /\ 2$
(7.12.2)	$LegendreSymbol \leftarrow g^b \bmod p$
(7.12.3)	if $(LegendreSymbol > 1)\ and\ (LegendreSymbol < f)$
(7.12.4)	output: COMPOSIT
(7.13)	$a \leftarrow [g * modfK(g,p)] \bmod p$
(7.14)	If $a = 1$
(7.15)	$a \leftarrow g$
(8)	output: PRIME

The pseudo code of the functions used in the algorithm is as follows:

	function $modK$ (in $generator$, in $modulonumber$)
(1)	$a \leftarrow generator^{-1} \bmod modulonumber$
(2)	return $\lfloor((a\,.generator) - 1)/modulonumber\rfloor$

	function $getNotPRU$ (in $generator$, in $modulonumber$)
(1)	$g \leftarrow generator$
(2)	$U \leftarrow modulonumber$
(3)	if $\left(\frac{g}{U}\right) = -1$
(4)	$g \leftarrow g^2 \bmod modulonumber$
(5)	$i \leftarrow 0$
(6)	$m \leftarrow g$
(7)	do
(8)	$i \leftarrow i + 1$
(9)	$m \leftarrow U - m$
(10)	$m \leftarrow modK(m,U)$
(11)	if $m < 2$
(12)	$i \leftarrow 0$
(13)	$m \leftarrow modK(g,U)$
(14)	$g \leftarrow (m+g)^g \bmod U$
(15)	while $\left(\frac{g}{U}\right) \in 0..1$
(16)	return m

	function $GetPRU$ (in $generator$, in $modulonumber$)
(1)	$g \leftarrow generator$
(2)	$U \leftarrow modulonumber$
(3)	if $\left(\frac{g}{U}\right) = -1$
(4)	$g \leftarrow g^2 \bmod modulonumber$
(5)	$e \leftarrow U \bmod 5$
(6)	if $e \in \{2..3\}$
(7)	return $5\,.g \bmod U$
(8)	$e \leftarrow U \bmod 6$

```
(9)   if e = 5
(10)      return −3 . g  mod U
(11)  e ← U mod 8
(12)  if e ∈ {3,5}
(13)      return 2 . g  mod U
(14)  if e = 7
(15)      return −2 . g  mod U
(16)  i ← 0
(17)  m ← g
(18)  do
(19)      i ← i + 1
(20)      m ←  U − m
(21)      m ← modK(m, U)
(22)      if m < 2
(23)          i ← 0
(24)          m ← modK(g, U)
(25)          g ← (m + g)^g mod U
(26)  while  (g/U) ∈ 0..1
(27)  return m
```

2.2 Solution model allowing application of the Silver-Pohlig-Hellman algorithm without limiting conditions.

Each type of attack against a cryptographic protection system is designed to provide itself with the necessary amount of information and through additional analysis and in most cases in combination with a serious mathematical apparatus to realize the extraction of explicit information. In very rare cases, this is achieved without extracting the encryption key.

Public key algorithms are theoretically easier to attack than algorithms operating with symmetric keys because the ill-wisher easily receives a copy of the public key used to encrypt the information. Part of the ill-wisher's efforts are facilitated because the message largely gives information about the encryption algorithm used and the size of the key is known in advance. Public key algorithm attacks are categorized into two groups: key search attacks and analytical attacks [68].

Key search attacks are the more popular type of attack to use against public-key cryptographic systems because they are the easiest to understand. These attacks are realized by extracting the private key from the freely available public key [42, 56].

Attacks on the public key system built on an RSA basis are carried out by trying to decompose the modular number N into simple multipliers. This modular number is available to the attacker because it is the domain of both parts of the key (private and public). Once decomposed into multipliers the modular number N , allows to easily calculate the private key.

In this part of the dissertation, a model of solution of a system of congruent equations is proposed, which used in algorithms to restore the private key by efficiently decomposing a modular number N allows the recovery of the private key d. This is possible for implementation in algorithms such as that of Silver-Pohlig-Hellman in which the solution of a system of congruent equations is needed.

In this part of the dissertation, a presentation of the algorithm proposed by Silver-Pohlig-Hellman is made.

In the realization of the Pohlig-Hellman algorithm, the following classical algorithms are used: calculation of the largest total multiple, elevation of a degree in modular arithmetic $(z = x^n \ mod \ p)$, Calculation of an "inverse module" $(z = x^{-1} \ mod \ p)$ and solution of a system of congruent equations.

Solving a system of congruent equations

To find a solution to systems of this kind, an algorithm known as the "Chinese theorem" has been applied for centuries, which is believed to have been created in the 2nd century BC by Sun Tzu.

The limitation of applying the "Chinese Theorem" is that modular numbers must be coprime, which requires a complete decomposition of the order of the group into simple multipliers. This constraint can be successfully overcome if an algorithm is applied which allows solving a system of congruent equations without this limiting condition. An algorithm offering the ability to solve systems of congruent equations without the presence of limiting conditions was developed and proposed at the scientific conference "Multimedia Communications, Services and Security. MCSS 2020" [35].

The motivation for this research and development is related to two reasons: 1) it is possible to solve systems of congruence equations in cases where modular numbers are not coprime; 2) it is possible to implement a solution in systems for parallel computing of large numbers.

In order to clarify the idea of this approach, we will derive a mathematical equation that will be the basis of this algorithm.

If we are given a system of congruent equations:

$$\begin{aligned} x &\equiv r_1 \ mod \ u_1 \\ x &\equiv r_2 \ mod \ u_2 \\ x &\equiv r_3 \ mod \ u_3 \\ &\cdot \\ &\cdot \\ &\cdot \\ x &\equiv r_i \ mod \ u_i \end{aligned} \tag{7}$$

and we express the first two equations as: $x_0 = r_1 + k_1.u_1$ и $x_0 = r_2 + k_2 \, .u_2$ we will have a solution of the smallest value. x_0. This is achievable if we know the values of k_1 and k_2, where $1 \leq k_1 < u_2$ and $1 \leq k_2 < u_1$. The value of x_0 will be a solution to the first two equations of the system.

If we multiply both sides of the equations we get:

$$x_0{}^2 = (r_1 + k_1.u_1)(r_2 + k_2.u_2) \tag{8}$$

$$x_0{}^2 = \ u_1.u_2.k_1.k_2 + k_1.u_1.r_2 + k_2.u_2.r_1 + r_1.r_2 \tag{9}$$

If we calculate by modulus A the both sides of equation (9), where $A = u_1.u_2$, we will obtain the next congruence:

$$x_0{}^2 \equiv \ (u_1.u_2.k_1.k_2 + k_1.u_1.r_2 + k_2.u_2.r_1 + r_1.r_2) \ mod \ A \tag{10}$$

In this equation, the value of $u_1.u_2.k_1.k_2 \ mod \ A$ is equal to 0, hence:

$$x_0{}^2 \equiv \ (k_1.u_1.r_2 + k_2.u_2.r_1 + r_1.r_2) \ mod \ A \tag{11}$$

By task definition $r_1 + k_1.u_1 = r_2 + k_2\,.u_2$, from where we can express $k_1 = \frac{k_2.u_2+r_2-r_1}{u_1}$ and substituted in the equation (11), from which it follows:

$$x_0{}^2 \equiv \left(\frac{k_2.u_2 + r_2 - r_1}{u_1}\,.u_1.r_2 + k_2.u_2.r_1 + r_1.r_2\right) mod\ A \quad (12)$$

$$x_0{}^2 \equiv (k_2.u_2.r_2 + r_2{}^2 - r_1.r_2 + k_2.u_2.r_1 + r_1.r_2)\ mod\ A \quad (13)$$

$$x^2 \equiv [k_2.u_2.(r_2 + r_1) + r_2{}^2]\ mod\ A \quad (14)$$

From the congruence (14) we can derive:

$$x^2 \equiv [k_2.u_2.(r_2 + r_1) + r_2{}^2]\ mod\ u_1 \quad (15)$$

Since $x_0 \equiv r_1\ mod\ u_1$, therefore $x_0{}^2 \equiv r_1{}^2\ mod\ u_1$ we substitute $z = x^2 = r_1{}^2\ mod\ u_1$ in congruence (15) and we obtain:

$$z \equiv [k_2.u_2.(r_2 + r_1) + r_2{}^2]\ mod\ u_1 \quad (16)$$

Therefore, to calculate k_2 the following Diophantine equation must be solved:

$$b.k_2 - u_1.y = z - r_2{}^2 \quad (17)$$

where $b = u_2.(r_2 + r_1)$.

We use the smallest positive solution for k_2 and put it in$x_0 = r_2 + k_2\,.u_2$ to obtain a solution for x_0, which is the solution of the first two equations. Equation (17) always has a solution, because $z - r_2{}^2$ is always divisible by $GCD(b, u_1)$. Proof of this is the derived congruent equation (16).

Using equation (17) in the following algorithm, we can solve the whole system congruent equations (7). The algorithm is applicable to all cases regardless of whether the modular numbers u_i, u_j $(i \neq j)$ are coprime. As a consequence, it is not necessary to decompose the u_i into simple multipliers. This allows acceleration of the process for solving systems of congruent equations.

Steps of the algorithm using the presented mathematical apparatus:

Input:	$Reminders\ [r_0, r_1, r_2, \dots, r_n]$ $modNumbers\ [u_0, u_1, u_2, \dots, u_n]$
Output:	x_0 - **minimal value of** x
(1)	$uniqLCM \leftarrow u_0$
(2)	$bValue \leftarrow r_0$
(3)	**for each** $i \in 1..n$
(4)	$m_1 \leftarrow uniqLCM$
(5)	$uniqLCM = (uniqLCM\ .\ u_i)$
(6)	$z \leftarrow (bValue^2\ mod\ m_1) - r_i{}^2$
(7)	$b \leftarrow (bValue + r_i)\,.u_i$
(8)	**solve:** $b\,.k_2 - m_1\,.y = z$
(9)	$bValue \leftarrow m_2\ .\ minpositive(k_2) + a_2$
(10)	**output:** $x_0 \leftarrow bValue$

The loop iterations of the proposed algorithm are equal to the number of equations in the system minus one. In this proposed algorithm, the values of numbers increase with each iteration. In the classical algorithm, all mathematical operations

must be performed with a binary number size close to the product of all modular numbers m_i. The only drawback of the algorithm described above is that if any pair of u_i и u_j $(i \neq j)$ are not coprime numbers, then x will not be the minimum solution value of the system. To overcome this shortcoming, the next algorithm was developed.

Input:	$Reminders\ [r_0, r_1, r_2, \dots, r_n]$ $modNumbers\ [u_0, u_1, u_2, \dots, u_n]$
Output:	x_0 - **minimal value of the solution** x
(1)	$uniqLCM \leftarrow u_0$
(2)	$bValue \leftarrow r_0$
(3)	**for each** $i \in 1..n$
(4)	$m_1 \leftarrow uniqLCM$
(5)	$g \leftarrow GCD(uniqLCM, u_i)$
(6)	$uniqLCM \leftarrow (uniqLCM\ .\ u_i)\ div\ g$
(7)	**if** $(u_i > m_1)$
(8)	$m_2 \leftarrow m_1$
(9)	$a_2 \leftarrow bValue$
(10)	$m_1 \leftarrow u_i$
(11)	$a_1 \leftarrow r_i$
(12)	**else**
(13)	$a_1 \leftarrow bValue$
(14)	$m_2 \leftarrow u_i$
(15)	$a_2 \leftarrow r_i$
(16)	$z \leftarrow (a_1{}^2\ mod\ m_1) - a_2{}^2$
(17)	$b \leftarrow [(a_1 + a_2)\ .m_2]$
(18)	**solve:** $b\ .k_2 - m_1\ .y = z$
(19)	$bValue \leftarrow m_2\ .\ minpositive(k_2) + a_2$
(20)	**if** $(bValue\ mod\ m_1 <> a_1)$
(21)	$t = a_1 - bValue$
(22)	$w = m_2\ .\ minpositive(k_2)$
(23)	**solve:** $w\ .n - m_1\ .y = t$
(24)	$bValue = bValue + w\ .minpositive(n)$
(25)	**output:** $x_0 \leftarrow bValue$

To demonstrate the functioning of the algorithm, an example is given in "Appendix 2". In this example, modular numbers are not coprime two by two.

The proposed solution model is applicable in the realization of the Silver-Pohlig-Hellman algorithm without limiting conditions that must be complied with if the "Chinese Theorem" is applied to solve a system of congruent equations.

2.3 Conclusions.

- A decision model allowing improvement of the divisibility score of a number when using the Miller Rabin algorithm in the process of generating prime numbers for cryptographic keys leads to an increase in the robustness of functioning of RSA-based cryptographic systems.
- A solution model allowing the application of Silver's Pohlig Hellman algorithm without limiting conditions can be used to reduce the number of iterations needed to achieve faster and more effective attack against RSA-based cryptographic systems and creates prerequisites for reducing the resilience of RSA-based systems.
- Attacks on a single key do not allow the range of the attack. Attacks using an analytical approach are wide-ranging and allow compromising a specific cryptographic algorithm or a whole group of algorithms.

CHAPTER THREE

METHODOLOGY FOR THE USE OF KLEPTOGRAPHY IN RSA BASED CRYPTOGRAPHIC SYSTEM.

According to the rules of public cryptography, the necessary conditions to achieve sufficient security are compliance with the standards when generating keys and the reliable and secure storage of the personal parts of each of them. The implementation of all this may not be the guarantor of the security of an RSA-based cryptographic system. Security can be compromised easily, provided that the creator of the means by which keys are generated has information allowing him to decompose the modular number N with the speed approximately equal to real time operation, by applying kleptography [38].

In this chapter, mathematical foundations and solution model of a new idea of a kleptographic attack algorithm are laid out. An assessment of the feasibility and benchmarking with existing algorithms has been made. The degree of robustness from disclosure was analyzed and the technical parameters that characterize the proposed algorithm were determined.

3.1 Mathematical foundations and model.

The idea of using exact squares related to a composite number was first invented and proposed by Pierre de Fermat in 1643. Since then, it has been known that exact squares have an effective application in the process of decomposing compound numbers into multipliers. Using this knowledge, we will represent with formulas and graphically, some new points of view related to composite numbers and the relationship of exact squares with them.

Let $N = p.q = t^2 + r$ or $N \bmod t = r$, where $t = \lfloor \sqrt{N} \rfloor$ и $N + R = (t+1)^2$. From this representation we can derive the following equations:

$$(t+1)^2 = t^2 + 2t + 1 = N + R \quad (18)$$

$$2t - r = R - 1 \quad (19)$$

The presentation of the number N and the formulas derived so far can be visualized graphically in the following figures, where the number N is depicted in gray

color and is represented as a rectangle whose face is a product of both sides: number p (in light yellow color) and number q (in pale blue color). This graphical representation fully corresponds to the representation of a composite number $N = p.q$.

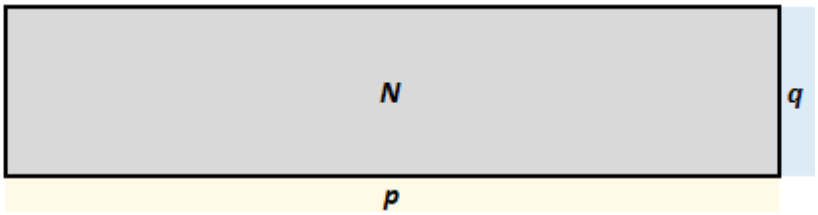

Fig. 6

Graphical representation of the number N as the product of p and q

Figure 7 illustrates graphically the relationship of the numbers t (in pale green) forming a square with side $t + 1$ and the numbers R (pale orange) and r (in dark blue).

We will make a second representation of the composite number N, $p = t + x$ and $q = t - y$, where $p > q \rightarrow x > y$ (figure 7), from which it follows that:

$$N = (t + x)(t - y) \quad (20)$$

where: $p = (t + x)$, $q = (t - y)$

$$N = t^2 + tx - ty - xy \quad (21)$$

$$N - t^2 = t(x - y) - xy \quad (22)$$

$$t(x - y) - xy = r \quad (23)$$

$$t = \frac{r + xy}{(x - y)} \quad (24)$$

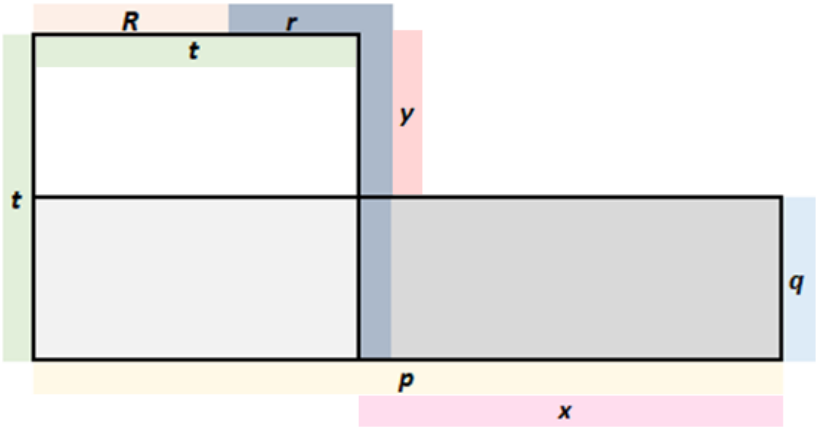

Fig. 7

Graphical representation of the mathematical dependence of numbers N, p, q and t, r, R, x, y

From the equations presented so far, we can derive the following formulas:

$$p + q = 2t + x - y \quad (25)$$

$$p - q = x + y \quad (26)$$

If we form a square with a side $\frac{p+q}{2}$ and from its aria subtract the aria of square with a length of side t, we will get a difference, which we will denote by v : $\left(\frac{p+q}{2}\right)^2 - t^2 = v$, therefore $\left(\frac{p+q}{2}\right)^2 = t^2 + v$.

By definition, N is a modular key number for RSA, then this number is composite obtained by multiplying two primes p и q, which in value are large numbers much greater than 2. Since they are prime numbers and greater than 2, they are odd in value, therefore their sum is divisible by 2 without residue and $\frac{p+q}{2}$ is an integer.

The area of the square with side $\frac{p+q}{2}$ is larger than that of the square with side t, because by definition $x > y$ (see equation 20 and by condition $p > q$). Since $\frac{p+q}{2}$ is larger in value than t we can write the equality:

$$\frac{p+q}{2} = t + m \tag{27}$$

Based on equations (25) and (27), the equation applies:

$$\frac{2t + x - y}{2} = t + m \tag{28}$$

Simplifying (28) we obtain:

$$x - y = 2m \tag{29}$$

For a better idea we will use the next two figures (8 and 9), which illustrate the relationships between the formulas. On these graphs, the number m used in equations (27, 28, 29) is depicted in dark blue-gray, the number y in pale orange and the number x in pale pink.

If we denote by $i = y + m$ and consider Figure 9, where 3 areas with equal size are marked with blue color (3 rectangles colored with light red color and each with sides q and $i = y + m$), we can derive equation (30), which is a derivative of all described so far:

$$(t + m)^2 - i^2 = N \tag{30}$$

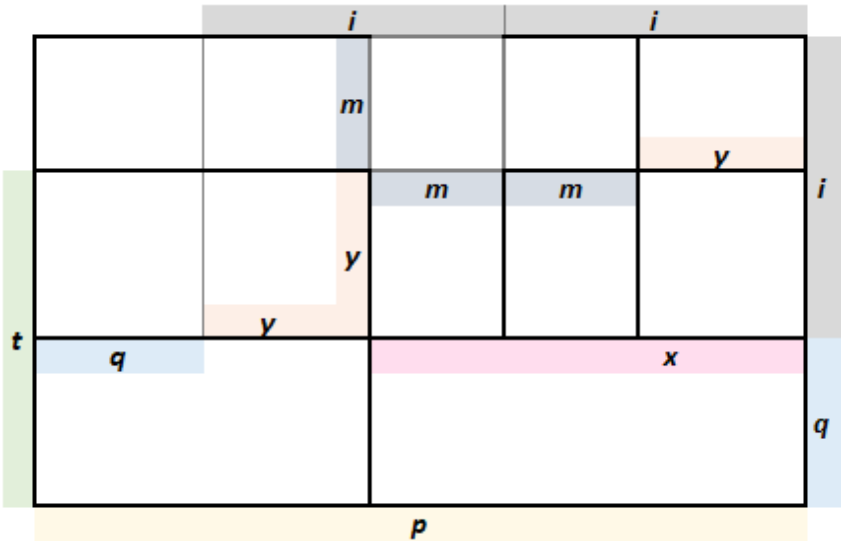

Fig. 8

Graphical representation of the mathematical dependence of numbers N, t, x, y and m, i

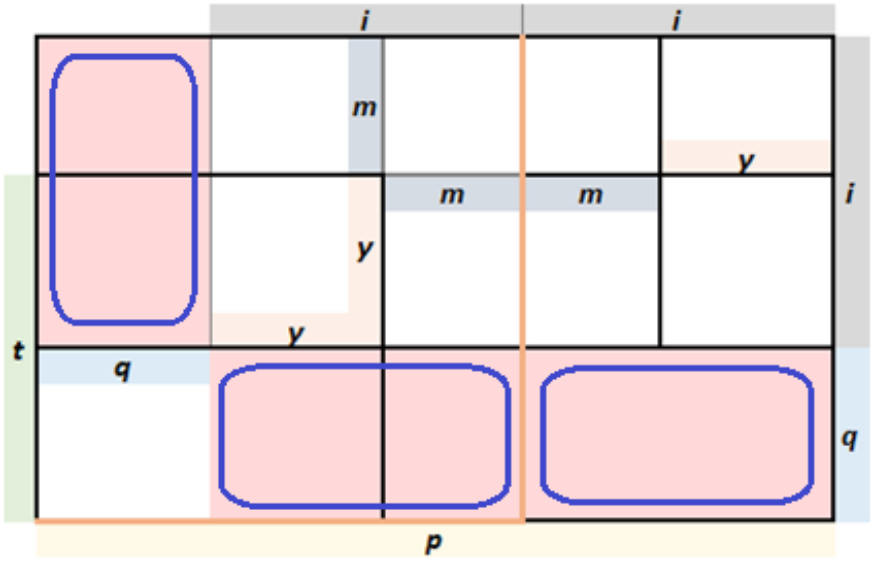

Fig. 9

Graphical representation of the mathematical dependence between $(t+m)^2$, i^2 and number N

If we represent $m = g + 1$ and substitute in equation (29) we get:

$$g = \frac{x - y - 2}{2} \tag{31}$$

Substituting m in (30) we will derive:

$$(t + g + 1)^2 - i^2 = N \tag{32}$$

If we express i^2 by $i^2 = (t + g + 1)^2 - N$ and simplify we get an equation that expresses i by g:

$$i^2 = (t + 1)^2 + 2 \,.\, g \,.\, (t + 1) + g^2 - N \tag{33}$$

From (18) we know that $N + R = (t + 1)^2$, substituting$(t + 1)^2 - N$ into (33) we obtain:

$$i^2 = g^2 + 2 \,.\, g \,.\, (t + 1) + R \tag{34}$$

The conclusions we can draw from (32) and (34) are two: 1) at firmly set values of t and R, the value of N depends on the magnitude of g; 2) we have a minimum value of g when the difference between p and q is 2, then $g = 0$.

By definition, if we have a number N, we easily calculate t and R, because we know from the formulas that the value of $t = \lfloor \sqrt{N} \rfloor$ or $N \; mod \; t = r$ and $R = (t^2 + 1) - N$. From what has been said, it follows that, if t and R are known quantities, then the decomposition of a complex number N depends on finding an integer g that satisfies both equations (32) and (34) at the same time. From the mathematical apparatus derived so far, it is true that the value of g From the mathematical apparatus derived so far, it is true that the value of p and q.

In the approach proposed by Fermat, two numbers are sought the difference of squares to which it is equal to the number N. In the equations derived so far, these two numbers are equivalent to i and sum $t + g + 1$ (see equation 32). From the approach thus used and the derived equations it is evident that these two numbers are directly dependent on the value of a number g (see equations 32 and 34) and are significantly larger in value, which means that searching for a value of the number g is a significantly faster process and would lead to decomposition of N for a much smaller number of iterations. The number of numbers that can satisfy equations (32) and (34) depends on

the number of pairs of multipliers whose product is N. If N is the product of two simple multipliers, then the value of g satisfying these equations is only one. From the mathematical apparatus can be drawn another correct conclusion: with numbers N obtained by multiplying more than two simple multipliers, the smallest value of g_i we will have if the difference $p_i - q_i$ is the smallest one of all pairs p_i and q_i in to the set of integers which as a product are equal to N.

Based on the created mathematical apparatus, an answer to the question "Is there a way for the value of g that satisfies both equations to be reduced?" can be sought. The answer to this question is "YES". This can be achieved by multiplying N by $n = a.b$ and if $\frac{a}{b} \approx \frac{p}{q}$, then for $M = N.n = p.q.a.b$ we will have $g_M < g_N$. Even if the values of the ratios do not fully match ($\left\lfloor\frac{a}{b}\right\rfloor = \left\lfloor\frac{p}{q}\right\rfloor$), the value of g_M can be zero. In other words, we will have a new larger composite number M with factors $a.p \approx b.q$, for which $\left\lfloor\frac{a.p}{b.q}\right\rfloor \approx c$, where $c \ll q < p$ and $g_M < g_N$.

Given g_M we can factorize M into prime factors and thus calculate the values of p and q by finding the greatest common divisor (GCD) between f and N where f is a factor of M. The possible combinations for this factor are: $f = p.a$, $f = p.b$, $f = q.a$ or $f = q.b$. The conclusion is that if the factors p and q of N generate a lower value of g, it is highly likely that by exhaustively searching for g, the number N can be factored.

From everything stated thus far, it can be concluded that ensuring the security of RSA-based cryptographic systems is not enough by just complying with the NIST's "SP 800-56B" standard from 2020 and the ETSI's "TS 102 176-1 V2.0.0" standard from 2007. To remember, according to these standards:

- $p - q > 2^{nBits\,-\,100}$ by NIST
- $0.1 < |log_2 p - log_2 q| < 30$ by ETSI

Even if these conditions are met, if we obtain $M = n.N$ with small values of a and b, and $g_M \ll q < p$, we can quickly factor N by using full exhaustion of n combined with full exhaustion of g_M if $g_M \,.\, n \ll q < p$. In other words, the "ratio" of p and q is extremely important for the security of RSA-based systems. The process of finding a balance between n and the two factors of N is called "Fusion of Number Balance" (FNB). The significance of the relationship between the two factors p and q of the modulus N involved in forming an RSA key and the practical use of the "Fusion of Number Balance" algorithm is demonstrated in the example provided in "Appendix 3."

In conclusion of this point, it can be stated that "Synthesis of Numeric Balance" is not a universal algorithm for factoring numbers into prime factors. It can only be used to fully exhaust values for n for composite numbers, where the relationship between the factors p and q allows it. However, it is a vivid example that the conditions for checking randomly generated prime numbers, as elements of RSA-based cryptographic system keys, must be revisited and an additional condition checking the magnitude of g must be included.

3.2 Practical Implementation of an RSA-based Kleptographic Algorithm.

In order to clarify the degree of potential risk to the stability of public key based (RSA) cryptography systems, a variant of a Kleptography algorithm has been developed through the impact on cryptography primitives, as described in the

previous mathematical basis. Based on the investigation of its telemetric parameters, potential hidden functioning capabilities, etc., assessments of the level of security vulnerability are made that can be implemented in the event of its use.

To eliminate the assumptions of weaknesses in the generation of prime numbers, an algorithm for their generation has been created. The result of the algorithm creates a random number that is tested for divisibility. The input parameter of the algorithm is the length in bits that the number should have. The pseudo code of the algorithm has the following form:

Input:	$\boldsymbol{l}$ **– needed bit length**
Output:	$\boldsymbol{p}$ **– prime number**
(1)	$t \leftarrow 2^l$
(2)	$z \leftarrow 0$
(3)	**do**
(4)	$\quad z \leftarrow z\ shl\ 8 + randomByte$
(5)	**while** $z > t$
(6)	$b \leftarrow z\ shr\ (bitlengt(z) - l)$
(7)	**if** $b\ mod\ 2 = 0$
(8)	$\quad b \leftarrow b + 1$
(9)	**while** $b\ not\ prime$
(10)	$\quad b \leftarrow b + 2$
(11)	**output:** $p \leftarrow b$

The algorithm for generating prime numbers doesn't rely on fast prime number generation principles and methods because most of them use a set of prime numbers that are smaller in value. This limits the set in which the final generated number falls. The proposed algorithm is iterative and the number of iterations required to find a prime number will be equal to $\frac{(p-b)}{2}$ (according to the definition of variables in the pseudo code). This means that the execution complexity of this algorithm is the product of $\frac{(p-b)}{2}$ and the complexity of the algorithm used to check divisibility (such as the complexity of the Miller-Rabin primality test)

For greater clarity and convenience, the idea for the kleptographical algorithm will be described schematically and algorithmically, which will facilitate subsequent evaluations and analyses. For brevity, the proposed algorithm is named "gBaseKleptoRSA". The meanings of the numbers in the descriptions will correspond in meaning to the meanings in the previous point.

For the implementation of the gBaseKleptoRSA attack, the attacker possesses a secret key n. This key has a value $n = a.b$, where a and b are mutually prime numbers $(GCD(a\ ,\ b) = 1)$ and with a size greater than 2^{32}. The size of a and b is assumed to be grater then 2^{32} because it is logical to use such pairs $[a, b]$ such that $n = a.b$ has a size in bits that is relatively large enough to enable full exploitation for its revelation. The numbers a and b will serve to form a solid hidden relationship between the generated N factors.

The length of the key will be represented by k. The binary length of n will be represented by l. To obtain the values of the factors for each kleptographic key we generate, we will use a randomly generated prime number u with a binary size of $\frac{k-l}{2}$. For each key, the value of u is generated separately, not repeated, and not stored. We obtain the factors through the following calculations:

$$p = NextPrime(u.a) = Prime(u.a + r_p)$$
$$q = NextPrime(u.b) = Prime(u.b + r_q)$$

This means that the value of p is the next prime number greater than $u \cdot a$, and for q, it's the next prime number greater than $u \cdot b$. We can calculate the maximum approximate size of the values of r_p and r_q using the formula for calculating the approximate number of primes less than x: $x/\ln x$.

Despite the fact that this formula does not give an exact result, but an approximate one, and years of research on the problem of determining the number of prime numbers up to the value of x have not produced a result for computing with absolute accuracy, for our purposes we can use it. With its help, we obtain: $r_p \approx \frac{p}{\ln p} - \frac{u.a}{\ln u.a}$ и $r_q \approx \frac{q}{\ln q} - \frac{u.b}{\ln u.b}$. The values of r_p and r_q affect the rate of generation of p and q. The approximate number of required checks for the divisibility of p and q will be $r_p \,.\, r_q =$ number of primality tests.

In this way, through the secret value of the number n, we will get the value of $q_M = 0$ for the number $M = n.N$. Thus, the attacker must perform the following calculations:

$$M = n.N$$
$$t_M = \lfloor \sqrt{M} \rfloor$$
$$i_M = \sqrt{(t_M + 1)^2 - M}$$
$$f = t_M - i_M + 1$$
$$p = GCD(f, N) \quad q = N \; div \; p$$

The used public exponent in the generated key is an input value, because by standard it should be a small prime number with the minimum number of bits equal to 1. In practice, the most commonly used value for the public exponent of an RSA key is:

$$e = 65537_{dec} = 100001_{bin}.$$

The pseudo code of the proposed algorithm is:

Input:	$\boldsymbol{k}$ – bit length of $\boldsymbol{N}$ $\boldsymbol{e}$ – public key exponent $(\boldsymbol{a}, \boldsymbol{b}: \boldsymbol{n} = \boldsymbol{a}.\boldsymbol{b})$ – attacker's private key
Output:	$\boldsymbol{p}, \boldsymbol{q}$ – prime number $\boldsymbol{N}$ – RSA modulo number $\boldsymbol{d}$ – user's private key
(1)	$n \leftarrow a \,.\, b$
(2)	$l \leftarrow BitSize(n)$
(3)	$u \leftarrow RandomPrime\left(\frac{k-l}{2}\right)$
(4)	$p \leftarrow NextPrime(u.a)$
(5)	$q \leftarrow NextPrime(u.b)$
(6)	if $(p-1) \,.\, (q-1) \; mod \; e = 0$
(7)	goto: (3)

(8)	if $p < q$
(9)	$z \leftarrow p$
(10)	$p \leftarrow q$
(11)	$q \leftarrow z$
(12)	$N \leftarrow p \,.\, q$
(13)	$d \leftarrow e^{-1} \, mod \, \varphi(N)$
(14)	output: p, q, N, d

In order for the proposed algorithm to function properly, a divisibility check has been included, in which the values of $\varphi(p)$ and $\varphi(q)$ are tested for divisibility by the public exponent e. This is a mandatory requirement because if they are not mutually prime, it is impossible to calculate the private key, thus the Diophantine equation:

$$e \,.\, d - \varphi(N).\, t = 1$$

will not have solution because $GCD(\varphi(N), e) > 1$.

Based on the described mathematical apparatus, we will use pseudo code to describe the algorithm by which the attacker will decompose the modular number N:

Input:	$\boldsymbol{N}$ – RSA modulo number $\boldsymbol{e}$ – public key exponent $(\boldsymbol{a}, \boldsymbol{b}: \boldsymbol{n} = \boldsymbol{a}.\boldsymbol{b})$ – attacker's private key
Output:	$\boldsymbol{p}, \boldsymbol{q}$ – prime number $\boldsymbol{d}$ – user's private key
(1)	$M \leftarrow n.N$
(2)	$t \leftarrow \lfloor \sqrt{M} \rfloor$
(3)	$i \leftarrow \sqrt{(t+1)^2 - M}$
(4)	$p \leftarrow GCD(t - i + 1, N)$
(5)	$q \leftarrow {}^{N}/_{p}$
(6)	$d \leftarrow e^{-1} \, mod \, \varphi(N)$
(7)	output: p, q, d

The algorithm was implemented in a program code and through the created software application, over 2000 keys were generated with a size of 2048 bits. Based on logical analysis and evaluation of the results, it was determined that in a not small part of the generated keys, 74.8%, the value of the most significant bit is 0. In these cases, the actual key size does not meet the requirements of the standards and is not 2048 bits. This weakness of the algorithm was the reason to create a new algorithm with a different principle of using generated prime numbers, but with the mathematical and algorithmic foundation of the cryptographic attack method retained. This version of the algorithm is referred to as **"gBaseKleptoRSA2"**.

The new method does not rely on just one pair of a and b multipliers for one value of n, but generates a set of pairs a_i and b_i. Thus, the process is divided into two stages. In the first stage, a basic table is generated, which has constant values for its elements. This allows it to be part of a hardware program code or pre-defined constants in a

software program. It represents a set of options for the attacker's keys and allows for rapid modular number N decomposition for every key generated through "gBaseKleptoRSA2". The description of the semantics of the two stages is as follows:

- In the first stage, using the input parameters $[a_0, b_0]$, a set $S = \{[a_i, b_i],\ i \in 0..80\}$ is generated, which is done using a pre-defined 9×9 matrix $R_{x,y} = [l, h]$.

$$\begin{bmatrix} R_{0,0} = [0,0] & \cdots [0,4][0,-1] \cdots & R_{0,8} = [0,-4] \\ \vdots & \ddots & \vdots \\ R_{8,0} = [-4,0] & \cdots [-4,4][-4,-1] \cdots & R_{8,8} = [-4,-4] \end{bmatrix}$$

  The matrix specifies the law through which the remaining elements of the set S are generated using binary operations, left shifts (shl), and right shifts (shr), starting from $S[a_0, b_0]$ and the following operations:

$$x = i\ div\ 9$$
$$y = i\ mod\ 9$$
$$a_i = a_0\ shl\ R_{x,y\rightarrow}\ l\ \ if\ R_{x,y\rightarrow}l \geq 0$$
$$a_i = a_0\ shr\ R_{x,y\rightarrow}\ l\ \ if\ R_{x,y\rightarrow}l < 0$$
$$b_i = a_0\ shl\ R_{x,y\rightarrow}\ h\ \ if\ R_{x,y\rightarrow}h \geq 0$$
$$b_i = a_0\ shr\ R_{x,y\rightarrow}\ h\ \ if\ R_{x,y\rightarrow}h < 0$$

  This creates the possibility to access at least one element from S, such that if we have the following calculation for a modular number: $N = a_i\ .\ b_i\ .u^2 = (a_i\ .\ u)\ .(b_i\ .\ u) = p\ .\ q$, we obtain one for which the most significant bit of the specified key length is equal to 1, satisfying the standards.
- In the second stage, a simple number u for a specific RSA key is generated. This u is randomly generated, used only once and not stored. We start searching and calculate:

$$p_i = u.a_i$$
$$q_i = u.b_i$$
$$N = p_i\ .\ q_i$$

  The search starts and the following calculations are made: the first index i for which we get 1 as the most significant bit of N breaks the search and we perform the following calculations.

$$D_a = random(1..97)\ shl\ d_a$$
$$p = NextPrime(u.(a_i + D_a))$$
$$D_b = random(1..97)\ shl\ d_b$$
$$q = NextPrime(u.(b_i + D_b))$$
$$N = p\ .\ q$$
$$d = e^{-1}\ mod\ \varphi(N)$$

The algorithm implementing the steps of the first stage for generating a table of pairs $[a_i, b_i]$ is described with the following pseudocode:

Input:	$\boldsymbol{R_{x,y}}$ – matrix $[\boldsymbol{a_0, b_0}]$ – initial base value of S
Output:	$\boldsymbol{S} = \{[\boldsymbol{a_i, b_i}],\quad \boldsymbol{i} \in \boldsymbol{0..80}\}$
(1)	$S[0].a \leftarrow a_0$,
(2)	$S[0].b \leftarrow b_0$,
(3)	for each $i\ \in 1..80$

(4)	$x \leftarrow i \; div \; 9$
(5)	$y \leftarrow i \; mod \; 9$
(6)	**if** $R[x,y].l \geq 0$
(7)	$S[i].a \leftarrow S[0].a \;\; shl \; R[x,y].l$
(8)	**else**
(9)	$S[i].a \leftarrow S[0].a \;\; shr \; R[x,y].l$
(10)	**if** $R[x,y].h \geq 0$
(11)	$S[i].b \leftarrow S[0].b \;\; shl \; R[x,y].h$
(12)	**else**
(13)	$S[i].b \leftarrow S[0].b \;\; shr \; R[x,y].h$
(14)	**output:** S

The steps described in the pseudo code of the following algorithm are executed to generate the key:

Input:	$\boldsymbol{k}$ – bit length of $\boldsymbol{N}$ $\boldsymbol{e}$ – public key exponent $\boldsymbol{S} = \{[\boldsymbol{a_i}, \boldsymbol{b_i}], \; \boldsymbol{i} \in \boldsymbol{0..80}\}; d_a, d_b \in 1..24$
Output:	$\boldsymbol{p}, \boldsymbol{q}$ – prime number $\boldsymbol{N}$ – RSA modulo number $\boldsymbol{d}$ – user's private key
(1)	$l \leftarrow BitSize(S[0].b)$
(2)	$u \leftarrow RandomPrime\big((k/2) - l\big)$
(3)	$m \leftarrow random(1..80)$
(4)	**for each** $i \in 1..80$
(5)	$f \leftarrow (m + i) \; mod \; 80$
(6)	$d \leftarrow u^2 \; . \; S[i].a \; . S[i].b$
(7)	**if** $bitlength(d) = k$
(8)	$D_a \leftarrow random(1..97) \; shl \; d_a$
(9)	$p \leftarrow NextPrime\big(u.(S[i].a + D_a)\big)$
(10)	$D_b \leftarrow random(1..97) \; shl \;\; d_b$
(11)	$q \leftarrow NextPrime\big(u.(S[i].b + D_b)\big)$
(12)	**if** $p < q$
(13)	$z \leftarrow p$
(14)	$p \leftarrow q$
(15)	$q \leftarrow z$
(16)	$N \leftarrow \; p \; . \; q$
(17)	$d \leftarrow \; e^{-1} \; mod \; \varphi(N)$
(18)	**output:** p, q, N, d

The examples of the result from executing the key generation algorithm using gBaseKleptoRSA2 are given in tabular form in "Appendix 4".

The algorithm used by the attacker to recover the private key generated by gBaseKleptoRSA2 has the following iterations:

Input:	$\boldsymbol{N}$ – RSA modulo number $\boldsymbol{e}$ – public key exponent $\boldsymbol{S} = \{[\boldsymbol{a_i}, \boldsymbol{b_i}],\ \boldsymbol{i} \in \boldsymbol{0..80}\}$; $d_a, d_a \in 1..24$
Output:	$\boldsymbol{p}, \boldsymbol{q}$ – prime number $\boldsymbol{d}$ – user's private key
(1)	for each $i \in 0..80$
(2)	for each $wL \in 1..97$
(3)	$aV \leftarrow S[i].a + wL\ shl\ d_a$
(4)	for each $wH \in 1..97$
(5)	$bV \leftarrow S[i].b + wH\ shl\ d_b$
(6)	$n \leftarrow aV\ .\ bV$
(7)	$M \leftarrow n.N$
(8)	$t \leftarrow \lfloor \sqrt{M} \rfloor$
(9)	$i \leftarrow \left\lfloor \sqrt{(t+1)^2 - M} \right\rfloor$
(10)	if $i^2 = (t+1)^2 - M$
(11)	$f \leftarrow t - i + 1$
(12)	$q \leftarrow GCD(f, N)$
(13)	$p \leftarrow {}^{N}/_{q}$
(14)	$d \leftarrow e^{-1}\ mod\ \varphi(N)$
(15)	output: p, q, d

The algorithm only outputs a result if the input is a number N generated using gBaseKleptoRSA2 with a table S as the input parameter during generation.

3.3 Benchmarking and evaluation of detectability

Information about topics related to kleptography is limited compared to other areas in cryptography. This issue is still not deeply explored compared to the rest of the theory regarding the protection of information through encryption. For this reason, the comparative analysis of the topic of kleptography presented in this chapter is focused on publicly described kleptographic algorithms attacking RSA. The analysis of existing kleptographic algorithms is focused on those that are claimed to have properties and functionality that make them practically applicable.

In order to make the tests necessary for analysis comparable, they are performed on the same hardware computer configuration: CPU i5 10th generation (i5-1035G4) with 8 cores, maximum frequency of 3.7GHz, and 16GB of RAM. During the execution of the tests, the processor load was in the range of 60% to 70%.

All kleptographic algorithms that have been proposed and created with the purpose of attacking RSA are based on the Young and Yung variant and in most publications, they refer to it or their code execution is very similar to their algorithm. For

this reason, in the dissertation work, only the Young and Yung algorithm is described and it is analyzed for practical-applicability evaluation. The following analysis can be taken as covering all modifications and developments of kleptographic algorithms for RSA, because the lines of pseudo-code, which the analysis is directed to, in the Young and Yung algorithm, are part of most proposed algorithms in the free internet space, even those created with the goal of improving specific elements of the algorithm.

Let's examine and analyze the following lines from Young and Yung's algorithm.:

(1)	**random** $s \in z_{p-1}^{\times}$, $bitsize(s) \approx k/2$
(2)	$p \leftarrow hash\ (s)$
(3)	**if** $p\ is\ not\ prime$ **or** $GCD(e, p-1) \neq 1$
(4)	**goto (1)**

1) In order to implement it, a hash function with a result whose binary length must be equal to or greater than half the length of the key (line (1): $k/2$) is used. If a 2048-bit key needs to be generated, according to this requirement, we must use a hash function with a result of at least 1024 bits in size. There are not many algorithms that offer such a length of result. We can only mention two SHA 3 and Skein, which, in addition to covering the requirement for the length of the hash result, are also robust and reliable hash algorithms. This is the other requirement that is specified as a condition in the Young and Yung algorithm that we are analyzing. If it is necessary to generate a larger key, with 4096 bits for example or larger, in such a case, the proposed kleptographic algorithm cannot function because there are no such hash functions that return results with such dimensions (2048 bits and larger).

2) According to rule (3), if the result of the hash of arbitrary data is not a prime number or $GCD(e, p-1) \neq 1$, we return to step (1). The probability of the process seriously slowing down in this part of the algorithm is high. Two reasons justify this statement: the slow execution of hash algorithms that need to calculate the hash of data with the desired length, and secondly, the probability of hitting a prime number in the range of these large numbers is not high enough and this will take several iterations. The time to generate a key is an important indicator. If the necessary time to generate a key deviate significantly from the average-statistical time in the standard algorithms for generating RSA keys (those without Kleptography), it would be a noticeable sign of malfunction. In the presence of such suspicions, the presence of Kleptography can be assumed.

Let's look at another part of Young and Yung's proposed kleptographic algorithm:

(7)	**solve:** $concat(c, RND)) = p.q + r$, $bitsize(q) \approx k/2$
(8)	**if** $q\ is\ not\ prime$ **or** $GCD(e, q-1) \neq 1$
(9)	**go to (1)**

On line (7) it is necessary to solve an equation where on the left is a number formed by a binary concatenation of c and random data obtained on line (5):

(5)	**random** $RND, bitsize(RND) \approx k/2$

The value of c is extremely important because it represents the result of encrypting s (obtained in step (1)), with the attacker's public key. As we have already noted, through s, the multiplier p is obtained. The main idea of the attack is to obtain p and then q by recovering s. If we look again at step (7), the only parameter that can be changed is RND, and the goal of solving the equation is to find a simple number q with an approximate length of $k/2$. For this, iterations are necessary to obtain such a value of r, so that: $q = \frac{concat(c,RND))-r}{p}$

To avoid overloading the algorithm, it is expected that the value of r is reasonably small. If due to a high value of r the process continues, as we have already explained, this will be a noticeable sign of malfunctioning. If r is of low value and we get a satisfactory result for q, it would mean that it is very likely that p is very close in value to q. If this is the case, it means that the difference $x - y$ (see Chapter 3, equation (29)) will be small, which in turn means that we will have a composite number with a small value of g, and as already demonstrated in the first point of this chapter, such N will be easily factorizable.

The second metric for the analysis of the Young and Yung algorithm is security. It includes an assessment of the possibility of information extraction from the secret channel by those who do not have the attacker's key. The analysis results show that the algorithm provides a secure channel for the protection of information passing through it. The difficulty of determining the secret parameter used to form p is equivalent to solving a problem for computing the discrete logarithm.

The main conclusion from the presented is that the variants of the Young and Yung algorithms and all their modifications are practically not applicable for implementation in hardware modules (HSM, smart cards, etc.) or software libraries. The main reason for this will be the illogically longer period of generating an RSA key compared to classical cryptographic algorithms, in which a kleptographic attack is not included.

To assess the threat to the stability of RSA-based cryptographic systems from the use of gBaseKleptoRSA2, an analysis was performed that included a series of tests similar to the analysis for the Young and Yung algorithm. To perform the tests, a software application was created that generates RSA keys using the gBaseKleptoRSA2 kleptographic algorithm. This was executed on the computer configuration considered to be the base in order to achieve an objective analysis.

In the first part of the analysis, tests were performed in order to evaluate the generation time for RSA keys with a size of 2048 bits. The average key generation time measured was ≈1.98 seconds. This is a commensurate key generation time of this kind compared to algorithms that do not use a kleptographic attack.

The second metric by which gBaseKleptoRSA2 was tested is the security of the leakage channel to the attacker so that he can implement a kleptographic attack to restore the private key. In order to be able to determine the unambiguous presence of an attack, it is necessary to restore in whole or in part the table of secret parameters (keys) of the attacking implementation gBaseKleptoRSA2.

The process of this analysis requires an estimate of pairs of modular number N multipliers that are involved in forming a different key. For example, examine p_i from one key with p_j from another or a pair of p_i with q_k $(k \neq i \neq j)$. If, as a result of such a study of pairs of multipliers, a number w is found for which:

$$w = GCD(x - r_x, y - r_y), \qquad w > 2^{32} \tag{35}$$

it can be assumed that such w is a potential member of S.

Using the specified basic constants for implementing gBaseKleptoRSA2 in the previous point, a full combination will require us 877,972,608 possible number combinations for research. The number of keys that need to be generated to provide this amount of numbers is 438,986,304. With an average key generation time of 1.98 seconds, this makes 10,060.10 days (≈27.56 years). If a supercomputer with 500 Intel(R) Xeon(R) CPU E5-2660 v2 2.20GHz processors is used in the key generation process, the key generation process will take 1,100 days. Expressed as a financial investment at an average of $1,250 per cloud HSM (price for IBM or AWS per month) or $4.85 per hour (Microsoft Cloud HSM), it is an investment of over $413,000. In order to achieve results from the analysis of discovering kleptography gBaseKleptoRSA2, time is also needed to calculate using the equation (35) for every number from all 877,972,608 numbers (Y). During the analysis for implementation of the search, we use one reference number x, with which we perform calculations with each number from the set of Y, in order to discover $w \geq 2^{32}$. The average calculation time using the specified configuration with the i5 processor is 7.39 minutes, which equals approximately 12,344.40 years. If a parallel calculation form is used with the help of the Intel(R) Xeon(R) CPU E5-2660 (500 CPU), it would take approximately 6.17 years.

The increase in the interval of values in which D_a and D_b are selected leads to an increase in the difficulty of detecting kleptography of this type. Tests have shown that changing the possible set of values for D_a and D_b from $1..97$ to $1..171$ and $d_a, d_b \in 64$ increases the time to break N (for a size of 2048 bits) by less than 10 seconds, but the increase in the minimum number of numbers to be studied increased by $9{,}815{,}261{,}184$, which requires over 11 times more time (~69 years) for key generation and searching for a possible w that proves the existence of gBaseKleptoRSA2.

Another important feature of using gBaseKleptoRSA2 is that when generating keys with different lengths, the same table S can be used.

If we summarize it about the proposed attack gBaseKleptoRSA2 we can say:

- ***Applicable at different key lengths***
- ***Difficult to be detected***

The generalizations made may be the basis of the claim that gBaseKleptoRSA2 is an extremely dangerous attack tool that can have a serious impact on the resilience of RSA-based cryptographic systems.

From what has been said so far, an important question arises: is it possible that there is an attack with a similar mechanism that is currently operating in hardware tools or software libraries that are used in practice. This requires an analysis of already existing cryptographic systems aimed at the RSA key generation process.

At the end of this dissertation work, attention was drawn to a study by a group of Czech researchers from Masaryk University [57]. The results published by them led to the conclusion that some of the technical tools and software libraries have weaknesses in the key generation process or there are forms of kleptography. Their study covers 38 key generation tools, including 19 open-source software libraries, 3 commercial software products, and 16 hardware tools.

In this final part of the dissertation work, it is clearly demonstrated the reasons that lead to the need to create a tool for studying and evaluating the weaknesses in the generation of RSA keys. This tool should have functionality for evaluating and identifying generated weak keys and for searching for signs of embedded kleptography.

3.4 Conclusions.

From the content of the third chapter of the dissertation, the following conclusions can be formulated:

- The standards in the field of cryptography related to the creation of RSA cryptographic keys need to be expanded. The conditions that need to be met during the generation of the pairs of prime numbers that will be used to form the key must be expanded. It is necessary to include a condition for evaluating the size of the ratio between the prime numbers p and q that form the modulo N.
- The mathematical foundations that serve for the functioning of RSA give the possibility, under certain conditions, to create a mechanism for implementing a cryptographic attack that is difficult to detect. This directly affects the negative aspect of the degree of resilience of the most widely used algorithm in the field of public cryptography, RSA.
- In order to achieve sufficient levels of security in RSA-based cryptographic systems, it is of utmost importance to develop technical and software tools that allow for fast, inexpensive, and accessible assessment of the presence of kleptography.

CONCLUSION

The development of public cryptography is a process where the pace of appearance of new mathematical foundations and encryption algorithms is not high and the speed of implementation in practical use is extremely low. This is mainly due to the fact that it takes time to test and study possible and underestimated assumptions for creating mathematical, logical and algorithmic vulnerabilities. But with even slower pace, developments are emerging that aid in the deeper testing and checking for the stability of cryptographic systems. Only such developments are capable of providing a toolset for creating a higher-quality product in the form of cryptographic protection tools.

The development of technology is a high-speed growth process that creates a basis for increasing communication opportunities of machine-to-machine and human-to-human types, leading to the need for the existence of larger and larger amounts of sensitive information. This requires the use and application of reliable and functional secure cryptographic protection systems. The increasing amounts of information, digitization, and technological progress will attract more and more those who want to exploit the weaknesses of the systems for their own benefit.

The creation of a solution model that allows for improvement of the quality assessment of generated keys for RSA-based cryptographic systems through fast and deterministic algorithms for evaluating the primality of numbers leads to an increase in the stability of the operation of such systems.

The use of the basic mathematical foundations on which modern public cryptography is built is a prerequisite for it to be secure in the rapidly evolving computational and quantum technologies. The risk to the security of RSA-based cryptographic systems exists, and the emergence of solution models that offer the possibility of increasing RSA attack performance exacerbates this risk. Optimizing existing attack algorithms using such models increases the speed of attack, which reduces the stability of public key cryptographic systems.

The model of the kleptographic attack described in the dissertation demonstrates the real danger to the resilience of cryptographic systems using RSA. Based on the results of the analysis and tests conducted, it can be assumed that current standards regulating the generation of RSA keys, which are widely used in practice, probably need to include additional verification conditions to reduce the likelihood of kleptography. This necessity stems from the fact that this cryptographic algorithm is the most widespread and used in practice. According to the statistics of the organization for transparency of digital certificates as of 2022, its use exceeds 75%. This necessity is further exacerbated by numerous incidents of vulnerabilities discovered in both hardware solutions for generating RSA keys and a significant number of software solutions used for such purposes.

The main conclusion that can be drawn from the dissertation work is that public cryptography will continue to develop and be widely applied, but in order to be most beneficial to society, the possibilities of its compromise need to be reduced to a minimum through the use of methods for full-scale attacks against the public key generation processes.

The efforts, knowledge, and data gathered during the development of the dissertation will be used for the purposes of future work related to creating solutions models in the following areas:

- Enhancement of the stability and reliability of key generation used in public cryptography.
- Qualitative and rapid assessment of the presence of implanted kleptography in systems that use public cryptography.
- Creation of models that will serve as a foundation for the creation of an algorithmic base for future post-quantum era work.

Achieved results

The following scientific and practical results were achieved in order to achieve the stated goal of the dissertation and carry out the related tasks:

1. In the first point of the second chapter, a solution model is proposed that allows for increasing the reliability of the divisibility assessment result as an addition to the widely used Miller-Rabin algorithm. This model creates the possibility of achieving a deterministic result without reducing the execution speed possessed by the Miller-Rabin algorithm.

2. In the second point of a second chapter a new model of solving systems of congruent equations is proposed, in which no limiting conditions exist for the model to be executable. The lack of limiting conditions distinguishes it from the algorithms known in practice, such as that of the "Chinese Theorem" and allows solutions to be achieved without the need for the numbers with which it is

calculated by module to be coprime. The use of this model in addition to the RSA attack algorithm proposed by Pohlig-Hellman enhances the performance of its implementation.

3. In the third chapter of the dissertation, a mathematical apparatus and a practical solution is presented, which allows the creation of a kleptographic algorithm for attacking RSA-based cryptographic systems. Comparative analyses and assessment of how kleptographic algorithms of this type threaten the resilience to functioning of systems using RSA. These results raise a question for consideration related to the sufficiency of the conditions in the composition of the applied standards relating to the assessment of the quality of a generated RSA key.

List of dissertation-related publications

[1] Stoianov, Nikolai, and Andrey Ivanov. "Public Key Generation Principles Impact Cybersecurity." Information & Security: An International Journal 47, no. 2 (2020): 249-260.

https://doi.org/10.11610/isij.4717

[2] Ivanov A., Stoyanov N., "Analysis of Approaches for Evaluating the Robustness of Public Key Cryptographic Algorithms", Tenth International Scientific Conference "Scientific Research and Investments in Technological Innovations - a Decisive Factor for Defense and Security", HEMUS 2020, Plovdiv.

[3] Ivanov A.G., Stoianov N.T. (2020) New Approach of Solving Congruence Equation System. In: Dziech A., Mees W., Czyzewski A. (eds) Multimedia Communications, Services and Security. MCSS 2020. Communications in Computer and Information Science, vol 1284. Springer, Cham. https://doi.org/10.1007/978-3-030-59000-0_2