Quantum-Safe Protocols and Application in Data Security of Medical Records

Adrian-Daniel STEFAN[1], Ionut-Petrisor ANGHEL[2] and Emil SIMION[3]

[1] Advanced Software Services, Faculty of Automatic and Computers, Bucharest, Romania
adrian.stefan0812@stud.acs.upb.ro

[2] Advanced Software Services, Faculty of Automatic and Computers, Bucharest, Romania
ionut.anghel2906@stud.acs.upb.ro

[3] Center for Research and Training in Innovative Applied Mathematical Engineering Techniques „Traian Lalescu", Faculty of Applied Sciences, Politehnica University of Bucharest, Romania
emil.simion@upb.ro

Abstract: The use of traditional cryptography based on symmetric keys has been replaced with the revolutionary idea discovered by Diffie and Hellman in 1976 that fundamentally changed communication systems by ensuring a secure transmission of information over an insecure channel. Nowadays public key cryptography is frequently used for authentication in e-commerce, digital signatures and encrypted communication. Most of the public key cryptosystems used in practice are based on integer factorization (the famous RSA cryptosystem proposed by Rivest, Shamir and Adlemann), respectively on the discrete logarithm (in finite curves or elliptic curves). However these systems suffer from two potential drawbacks like efficiency because they must use large keys to maintain security and of course security breach with the advent of the quantum computer as a result of Peter Shor's discovery in 1999 of the polynomial algorithm for solving problems such factorization of integers and discrete logarithm.

Keywords: PKI, homomorphic encryption, multivariate cryptosystems, PAKE, RLWE, quantum computer, NP complete problems, digital signatures, cloud computing.

1 Introduction

Multivariate cryptography is a class of public key cryptography and is based on the problem of solving nonlinear systems of equations over finite fields, a problem that has been proven to be one of the NP complete class. Therefore, a cryptosystem based on such an approach is a viable option for both conventional and quantum computers since they also have no advantage in solving NP complete problems.

With time several algorithms based on the problem of solving multivariate polynomial systems over finite fields have been proposed and studied. Among them Poly-Dragon stands out, a secure and efficient multivariate public key algorithm that solves certain security problems that the other algorithms in the Dragon family have and due to its bijectivity it can also be used for digital signatures. Accordind to [1], the computation complexity of this algorithm in encryption and signature verification is $O(n^3)$, where n represents the number of bits and for decryption, respectively generating the signature only 4 exponentiations are needed in the finite field $\mathbb{F}_{2^n}$.

Homomorphic cryptography is a new concept with a significant impact in security because it solves the problem of confidentiality and security of data stored in the Cloud ensuring complete anonymization of data and thus avoiding security breaches. Moreover, it is considered a safe method for quantum computers as well which is why its study has attracted many researchers.

Today, Cloud Computing has become a standard method for ease of data access being used both by users to save documents in order to free up storage space on mobile devices and by government organizations, companies for big data processing and analysis, reason for which a convenient way was needed to ensure high confidentiality and intellectual protection of user data.

Homomorphic cryptography based on advanced mathematics allows the processing of encrypted texts which is why it provides a clearly superior performance compared to other cryptographic methods as it offers complete confidentiality and anonymization of data which remain encrypted throughout the processing. Although the idea of homomorphic encryption has been proposed since 1978, it was put into practice by Craig Gentry in 2009 as specified in [2]. From then till now the process has continued to expand, homomorphic cryptography being used primarily in the financial sector, the medical industry, moreover in all Cloud systems where ensuring data protection is paramount.

The study of partially and fully homomorphic cryptosystems has reached a significant stage in the field of computing, especially in Cloud Computing as there was a need for an adequate Big Data infrastructure to ensure data storage and processing using diverse machine learning models while guaranteeing user's data confidentiality [3]. In a world where data is essential and everything is centered around it a major challenge for Cloud Computing providers is to ensure privacy and guarantee better security of private information. With the development of fully homomorphic cryptosystems, the problem of complete data security and anonymization is solved since such an encryption scheme allows the calculation of any function on the encrypted data as illustrated in the article [4].

2 Permutation polynomials

A polynomial $f \in \mathbb{F}_q[x_1, x_2, \dots, x_n]$ is called a permutation polynomial over the field $\mathbb{F}_q$ where $q = p^n, p, n \in \mathbb{N}, p\ prim$ if and only if one of the 4 conditions below is met [1]:

$$\text{The function } c \to f(c) \text{ is surjective} \tag{1}$$
$$\text{The function } c \to f(c) \text{ is injective} \tag{2}$$
$$f(x) = a \text{ has solution in } \mathbb{F}_q, \forall a \in \mathbb{F}_q \tag{3}$$
$$f(x) = a \text{ has unique solution in } \mathbb{F}_q, \forall a \in \mathbb{F}_q \tag{4}$$

Note1: From the above relations a polynomial $f \in \mathbb{F}_q$ is identified as a permutation polynomial if it induces a bijection from$\mathbb{F}_q$ to $\mathbb{F}_q$.

Lemma: $g(x) = \left(x^{2^{k \cdot 2^r}} + x^{2^r} + \alpha\right)^l + x$ is a permutation polynomial [1] over finite field $\mathbb{F}_{2^n}$ if $Tr(\alpha) = 1$ and $l \cdot \left(2^{2^r k} + 2^r\right) \equiv 1\ mod(2^n - 1), l \in \mathbb{N}$, where

$$Tr:\ \mathbb{F}_{2^n} \to \mathbb{F}_2, Tr(x) = \sum_{i=1}^{n} x^{2^{i-1}} \text{ (trace function)} \tag{5}$$

A polynomial $L(x) \in \mathbb{F}_{q^m}[x]$ is called linearized polynomial over field $\mathbb{F}_q$ [1] if relation (6) is satisfied and this polynomial satisfies the property (7)

$$L(x) = \sum_{i=0}^{m-1} \alpha_i x^{q^i}, \alpha \in \mathbb{F}_q \tag{6}$$
$$L(\alpha + \beta) = L(\alpha) + L(\beta), L(x\alpha) = xL(\alpha);\ \alpha, \beta \in \mathbb{F}_{q^m}, x \in \mathbb{F}_q \tag{7}$$

In [1] another family of permutation polynomials over the field $\mathbb{F}_{2^n}$ is given by equation (8) where n is odd number, $\beta \in \mathbb{F}_{2^n}$ such that $\omega(\beta)$ is even and the only roots of the linearized polynomial L_β in $\mathbb{F}_{2^n}$ are 0 and 1, l a positive number that assure relation (9) and $\gamma \in \mathbb{F}_{2^n}, Tr(\gamma) = 1$.

$$f(x) = \left(L_\beta(x) + \gamma\right)^l + Tr(x) \tag{8}$$
$$l(2^{k_1} + 2^{k_2}) \equiv 1\ mod(2^n - 1), (2^{k_1} + 2^{k_2}, 2^n - 1) = 1;\ k_1, k_2 \in \mathbb{N} \tag{9}$$

Note2: Any element from $\mathbb{F}_{2^n}$ is uniquely identifiable as in relation (10) where $B = \{\upsilon_1, \upsilon_2, \dots \upsilon_n\}$ is a basis of the field $\mathbb{F}_{2^n}$ over $\mathbb{F}_2$.

$$x = \sum_{i=1}^{n} x_i\, \upsilon_i \tag{10}$$

Note3: The weight function of an element of $\mathbb{F}_{2^n}$ is defined as the number of occurrences of 1 in the n-tuple representation of x and is denoted by $\omega(x), x = (x_1, x_2, \dots x_n)$.

3 The Poly-Dragon cryptosystem

For the generation of the public key the two families of permutation polynomials defined in equations (5) and (8) will be considered. Given the quadratic size restriction of the public key not all permutation polynomials having the defined forms will be considered but only those for which $l = 2^m - 1, r = k_1 = 0, k = k_2 = m,\ m \in \mathbb{N}$. [1]

The secret key of the cryptosystem consists of the vectors $\alpha, \beta, \gamma \in \mathbb{F}_{2^n}$ and the invertible affine vectors $s, t : \mathbb{F}_{2^n} \to \mathbb{F}_{2^n}$ and the relationship between the encrypted text (y) and the original (x) is described by the relationship below where g and f are the two permutation polynomials that respect the relationships mentioned above [1]:

$$g(s(x) = f(t(y) \tag{11}$$

The encryption and decryption algorithms can be represented schematically in the form below considering the explanations given in [1]:

Encryption (transforming x into y):

- Alice wants to send Bob the message $x = (x_1, x_2, \dots, x_n)$
- Alice substitutes x and $\xi_y = Tr(t(y) \in \{0, 1\}$ in the public key, obtain two systems of n equations, solve them by Gaussian elimination and obtain the unique solution $y' = (y_1, y_2, \dots, y_n)$. Only one of the two systems has a solution because the equation (11) is satisfied (11) and the permutation polynomials f and g are bijections.
- Alice obtains y by inverting the function t and sends the encrypted message to Bob.

Decryption (transforming y into x):

- Bob receives the encrypted message $y' = (y_1, y_2, \dots, y_n)$ and uses the secret key consisting of the quintuple $(\alpha, \beta, \gamma, s, t)$
- Bob calculates $v = t(y')$ and $z_1 = L_\beta(v) + \gamma$
- Bob compute $z_2 = {z_1}^{2^m - 1} + Tr(v)$
- Bob determines $z_3 = {z_2}^{2^m} + z_2 + \alpha + 1$
- Bob calculates $z_4 = {z_3}^{2^m - 1}$
- Bob gets the tuple $X = (X_1,\ X_2);\ X_1 = s^{-1}(z_2 + 1), X_2 = s^{-1}(z_2 + z_4 + 1)$
- Bob identifies the correct message $x = X_1 | X_2$ – there is only one solution.

4 Homomorphic encryption and privacy of data stored in the Cloud

One of the most interesting approaches to cryptography is represented by the homomorphic encryption proposed in 1978 by Rivest, Adleman and Dertouzos based on advanced mathematics that allows the processing of encrypted texts without the need for their initial decryption which is why it ensures both a net performance superior to other cryptographic methods as well as high confidentiality and intellectual protection of user data because they remain encrypted throughout the processing. [5]

Since 1978 when the idea of homomorphic encryption was proposed, respectively after its implementation in 2009 by Craig Gentry this process has continued to expand, finding today many open-source implementations, for instances Microsoft SEAL, HElib, Lattigo, etc. The security of homomorphic encryption is based on the difficulty of solving the RLWE (*Ring Learning With Errors*) problem which is closely related to the lattice problem, thus providing better security than methods based on elliptic curves reason why the homomorphic encryption method is currently considered to be secure for quantum computers. [5]

The RLWE problem is based on the difficulty of solving the matrix equation below even if the initial matrix A is known where e is an error vector and s is the secret key.

$$A \cdot s + e = b; A \in \mathbb{Z}_q[n * m], s \in \mathbb{Z}_q[m * 1], e \in \mathbb{Z}_q[n * 1] \quad (12)$$

An advantage of homomorphic cryptography is performing operations on encrypted data without the need for interactivity between the user and the system, the client being the only owner of the private key which is why there is complete anonymization of the processed information. Therefore, the user can benefit from all the advantages offered by Cloud Computing while maintaining data confidentiality.

Another aspect to mention is mobile devices that have become an essential part of life being efficient and convenient communication tools that support a significant range of applications. However, mobile devices have performance, storage, bandwidth, computing power limitations which is why it was necessary to find a solution to solve these problems. This is where Cloud Computing reach providing adequate infrastructure but the issue of data privacy remains unresolved.

Although homomorphic encryption is one of the solutions that allows the processing of encrypted data but considering the low performance of mobile devices, a new encryption scheme LHE (*Lightweight Homomorphic Encryption*) has been proposed that minimizes the cost of key generation and the use of computing power for encryption. Article [6] presents this approach that offers a balance between security, energy consumption and functionality. Based on Gentry's scheme this method allows homomorphism both in the case of addition and multiplication which is why it represents an effective solution for ensuring the security and confidentiality of data stored in the Cloud.

5 Digital signatures

From a legal point of view, the electronic signature is a way to ensure the identification of the entity that attached data to other data in electronic format and from a practical point of view it provides authenticity, integrity of the signed document and non-repudiation using a standard named PKI (*Public Key Infrastructure*) to provide a high level of security and allow both the actual signing process and the possibility of verifying the signature by other parties.

Digital signatures are based both on asymmetric cryptography to ensure the fundamental principles of authentication and non repudiation and on hash functions that have the role of verifying the integrity of the document, but also provide a means by which the length of the message can be reduced so that the process of signing and verification to be as least expensive as possible given the complex mathematical calculations behind asymmetric algorithms. [7]

From a functional standpoint the digital signature is a technique similar to the MAC (*Message Authentication Code*) algorithm but unlike it, it uses a pair of keys, a private one for the message signing process, respectively a public one needed to validate the process. Thusly, the digital signature can be defined as a triplet (G, S, V) as give out in article [8] where:

- G – probabilistic algorithm for generating a pair of keys (s_k, p_k);
- S – algorithm for applying a digital signature to the message using the private key s_k;
- V – deterministic algorithm; decides whether the digital signature applied to the message is authentic by decrypting the message using the public key p_k.

Effectively hash functions are used to fix the message size problem. In this way a significant reduction of the resources needed to sign the message and the time needed for its transmission is ensured since this time the signature scheme is applied to the message hash function.

Regarding the hash functions, to ensure security they must fulfill the following properties pointed in [8]:

1. ***Irreversibility*** – given the dispersion function (h) and a random value y it is computationally difficult to find the input x such that $h(x) = y$
2. ***Resistance to weak collisions*** – computationally difficult to find y such that $h(x) = h(y)$
3. ***Resistance to strong collisions*** – computationally difficult to find x and y such that $h(x) = h(y)$

Using PKI, digital signatures meet the most demanding security rules which is why they are used in various fields likes medical system, private institutions, the banking system, Blockchain and offer cost reduction and efficiency of the signing process by ensuring easy transmission of the documents.

6 Integration of the PAKE primitive in the TLS protocol

A very important cryptographic primitive used in real scenarios is key exchange (KE) through which at least two entities can establish a session key to encrypt data in an insecure network. The most well-known protocols based on this primitive are Diffie-Hellman (DH) and Elliptic Curve Diffie-Hellman (ECDH) but they are vulnerable to the well-known man-in the-middle attack(MITM) because they don't provide a mechanism to authenticate the user's identity.[9]

Although the problem of user authentication is solved in PKI using digital signatures and trusted certificates, we still have quite a significant cost in key management and applying these concepts to the Web would lead to higher processing times and liability. One of the solutions found in this regard is ***PAKE*** (*Password-Authenticated Key Exchange*), a cryptographic primitive for establishing a secure communication between client and server with advantages such as: simple key management through passwords, resistance against MITM attacks, offline dictionaries, malicious web pages, phishing etc. Taking into account these aspects this primitive has been standardized and implemented in a multitude of real-world scenarios: JPAKE cryptosystems based on elliptic curves and implemented in TLS, Pre Shared Key (PSK) secure authentication for the Internet Key Exchange (IKE) protocol in RFC 6617, SRP (Secure Remote Password) protocol in RFC 2945, etc. [9]

TLS (*Transport Layer Security*) is an authentication protocol implemented to provide secure communication between client and server over an untrusted channel. Consisting of two major components namely *handshake* for negotiation and secure connection establishment, respectively *record* for transmitting data in a secure manner (privacy, authentication). This protocol is widely used (HTTPS, IMAPS, SMTPS) and comprises more than half of the traffic Web. [9]

With the revelation of Shor's algorithms (effective against cryptosystems using public keys based on the problem of factorization, discrete logarithm, elliptic curves) and Grover's (effective against symmetric encryption algorithms) with the advent of the quantum computer the security of many cryptosystems is lost, reason for that there have been many researches of post quantum algorithms. In general these algoritihms are based either on multivariate cryptography, lattices, hash functions or on symmetric ciphers with a much larger key size. In the article [9], the authors focused on the properties of lattices and the advantages offered by these (efficiency, security, simplicity in structure, small size for keys) and implemented the RLWE-PPK algorithm. To solve the security problem of the TLS protocol as well, they added this algorithm to the suite of cryptosystems so that this protocol has the possibility of using postquantum cryptographic primitives and is no longer vulnerable to the advent of the quantum computer. This integration can be done without requiring too many changes to the TLS standards which is why the approach can be easily and safely adopted in practice.

For the implementation of the post-quantum algorithm based on the RLWE problem, in the article [9] the authors used the C++ NFLib library because it contains efficient implementations of various algorithms optimized for lattice-based cryptography. The following specifications were also considered for the cipher suite integrated in TLS:

- ***Key exchange and authentication*** – RLWE-PPK protocol integration for exchange of keys and post-quantum mutual authentication
- ***Authenticated encryption*** – symmetric block algorithm based on Galois fields (AES GCM), algorithm that ensures the CIA triad (Confidentiality, Integrity, Authenticity) on the data
- ***Hash function*** – SHA 384

Following benchmark, an improvement of up to 7.4 times was found in terms of the execution time of the RLWE-PPK algorithm compared to the widely used but vulnerable one in quantum computers, namely J-PAKE. Also, for a realistic scenario a test program was implemented that simulates the establishment of a communication session between a client and server using the TLS protocol based on the encryption provided by RLWE-PPK. Following tests performed on a PC equipped with an Intel Core i7-6820HQ processor with a frequency of 2.7 GHz and 4 GB of RAM obtained on average (1000 executions) the times of 4.83ms and 4.94ms respectively for the handshake between the client and the server. Although the time obtained is an efficient one, in terms of the cost related to the size of the messages it is on average 3.8KB being somewhat higher than in the case of using the DH, RSA, ECDH encryption suite where the value is around 1-2 KB.

7 The proposed model

The idea of the project is to develop a Web platform through which users and doctors can safely keep data and medical history for all visits made by a patient to a public or private clinic.

In order to guarantee a high level of security for both conventional and quantum computers, authentication on the platform will be done using a PAKE-type protocol based on the RLWE problem (RLWE-PPK previously mentioned due to its efficiency) and confidentiality medical data will be secured by encryption using an algorithm based on post-quantum cryptography. As one of the basic functionalities of the platform is represented by the digital signature we will choose an algorithm from the Dragon family and an efficient and secure cryptosystem that could be used is the Poly-Dragon previously described and presented in detail in [1].

As a platform for the medical sector, homomorphic encryption is an indispensable solution as predictive analysis can be done on all the medical data of a patient in an efficient manner by using the methods exposed by a Cloud service. In this way we benefit from the

advantages offered by Cloud Computing and at the same time guarantee the anonymization of the processed data. For instance, one of the main problems present among the population is cardiovascular disease which is why we could implement a service for predicting the risk of developing this condition, a service running in the Cloud. As the data is encrypted, at no point during its processing, during the running of the prediction algorithms the Cloud service will not be able to extract information about the patients' data. When the prediction process is completed, a representative of the medical unit will locally decrypt the returned data and so confidentiality is ensured.

For all above mentioned reasons, the block diagram of this platform can be illustrated as in the figure below:

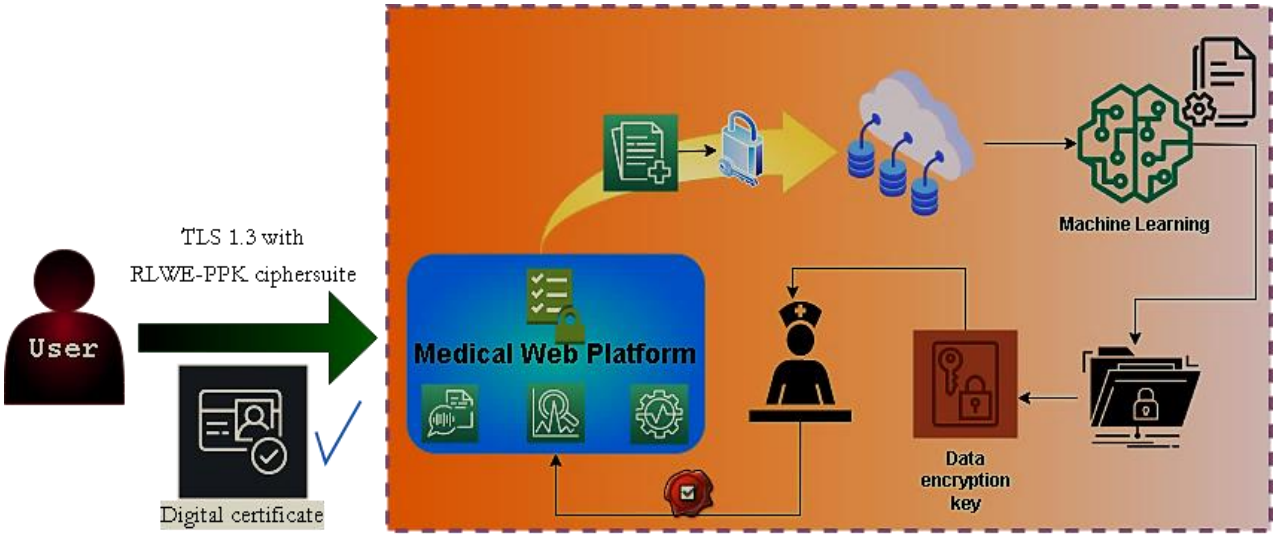

Figure 1 Platform model

8 Implementation details

To implement the Poly-Dragon cryptosystem we need a base of the space $\mathbb{F}_{2^n}$ over field $\mathbb{F}_2$ in order to work with elements from $\mathbb{F}_{2^n}$ in a programming language, for example Python. The most representative such bases are:

$$\Psi \text{ (normal basis)} = \{v, v^2, v^4, \dots, v^{2^{n-1}}\}, v \in \mathbb{F}_{2^n} \quad (13)$$

$$\Theta = \{1, v, v^2, v^3, \dots, v^{n-1}\}, v \in \mathbb{F}_{2^n} \quad (14)$$

Note: In relation (13) we choose $v \in \mathbb{F}_{2^n}$ so that the elements of the set Ψ are linearly independent and in (14) v represents a root of an irreducible polynomial of degree n from $\mathbb{F}_2[X]$.

If we took the normal basis for the implementation, then we would have the disadvantage of simplicity, the efficiency of squaring and implicitly to any power of 2,

respectively the trace determination of an element since the trace actually represents the number of non-zero coefficients in writing the element in relation to Ψ. Although we could benefit from these advantages, still there is a big trouble regarding the cost of multiplying such elements, writing the result in the same base. To overcome this impediment we would need an irreducible polynomial that has a root on υ, but unlike the set Θ this polynomial must also have other special properties and to ease implementation we will use the basis from the relation (14).

To encrypt / decrypt a message of size n bits we need an irreducible polynomial of degree n from $\mathbb{F}_2[X]$. In the article [10] we have given a table containing one example of such a polynomial for $n = \overline{2,10^4}$ but extra, an irreducible polynomial with 2 additional properties:

- The number of non-zero coefficients is minimal;
- Terms with non-zero coefficients have minimum degrees → time-saving process of multiplying two elements

As stated in [1] due to the substitutions performed, encryption/decryption can only be performed on messages having an odd number of bits. However, to be able to transmit any type of message we can use the following idea, namely if the message is of even length we add one bit to the end, otherwise two bits so that the last bit is 0 if we are in the first case and 1 in the second case. Although for an initial message of odd length the penultimate bit is redundant still following this procedure we would achieve the goal of transmitting any message with low losses.

The secret key of the cryptosystem is given by the quintuple $(\alpha, \beta, \gamma, s, t)$ where the first 3 elements represent vectors in the space $\mathbb{F}_{2^n}$ and the last two invertible functions over the same space. In view of [1] it is necessary and sufficient to have the two relations below fulfilled where the trace function and the weight function are those defined in the section P:

$$Tr(\alpha) = Tr(\gamma) = 1 \tag{15}$$

$$\omega(\beta)\ even \tag{16}$$

Note: For the two invertible functions we impose the conditions that they are given by an invertible matrix and a vector of order n. Therefore, the quintuple becomes $(\alpha, \beta, \gamma, As, cs, At, ct)$ and these elements of the secret key are read from a default file where $As, At \in K_{n,n}, cs, ct \in K_{n,1}, K = \mathbb{F}_2$.

Regarding operations on elements over $\mathbb{F}_{2^n}$, elements represented as in the relation (10) to optimize the calculations we take into account the fact that the addition operation in $\mathbb{F}_2$ is equivalent to XOR and multiplication to AND. Also, given the host of squarings for efficiency we precalculate the coefficients of the elements $1, \upsilon^2, \vartheta^4, \dots, \vartheta^{2(n-1)}$ in the considered basis Θ.

Note: For any element $x \in \mathbb{F}_{2^n}$ according to (10) and considering the choice of basis Θ we have the relations below since $x = x^2, 2 = 0$ in $\mathbb{F}_{2^n}$.

$$x = x_0 \cdot 1 + x_1 \cdot v + x_2 \cdot v^2 + \cdots + x_{n-1} \cdot v^{n-1}, x_i \in \mathbb{F}_2, i = \overline{0, n-1} \quad (17)$$

$$x^2 = \sum_{i=0}^{n-1} x_i{}^2 \cdot v^{2i} + 2 \sum_{0 \le i < j \le n-1} x_i x_j \cdot v^{i+j} = \sum_{i=}^{n-1} x_i \cdot v^{2i} \quad (18)$$

To obtain the ciphertext y from the original message x we are take into account the specifications mentioned in the section The Poly-Dragon. Thus, we make the following notations:

$$u = s(x), v = t(y) \quad (19)$$

$$A = \left(u^{2^m} + u + \alpha\right)^{2^m - 1} + u - Tr(v) \quad (20)$$

$$B = L_\beta(v) + \gamma \quad (21)$$

From the above relations and the following equation $\left(u^{2^m} + u + \alpha\right)^{2^m-1} + u = \left(L_\beta(v) + \gamma\right)^{2^m-1} + Tr(v)$ from [1] we conclude that $A = B^{2^m-1}$ şi $A^{2^m+1} = B$ (22) ⇒ knowing A implies easily determining B.

$$A^{2^m+1} = B^{(2^m-1)(2^m+1)} = B^{2^{2m}-1} = B\left(B^{2^{2m-1}-1}\right)^2 = B\left(B^{2^n-1}\right)^2 = B \quad (22)$$

Note: In relation (22) we used the well-known result $x^{2^n-1} = 1, \forall x \in \mathbb{F}_{2^n} \setminus \{0\}$.

According to the encryption algorithm we can say that at this stage we know the u, α, β, γ and $Tr(v)$ giving the values 0 and 1 respectively. Thus, we can determine A and then according to (22) B, then $L_\beta(v)$. Then, we write $v, L_\beta(v)$ in the base Θ and we obtain for each value of $Tr(v)$ a system that has as unknowns the coefficients of v in the base Θ. Each system is solved by Gaussian elimination. The procedure for converting y to x is described in section The Poly-Dragon.

To generate the secret key we consider the following notes:

- Matrices As, At can be sub-diagonal or over-diagonal with value 1 on the main diagonal to ensure invertibility;
- Vectors cs, ct can be chosen randomly;
- β must contain an even number of 1 bits;
- α, γ can be chosen randomly and in the conditions in which the following function does not have the value 1 we modify the first bit of the respective vector to obtain a correct value (this fact is possible because the trace function is additive and $Tr(1) = 1$ for any odd n)

9 Security of the proposed cryptosystem

A first attack attempt on the Poly-Dragon cryptosystem hand out in the article [1] is the method of linearizing equations in variables $x_i, i = \overline{0, n-1}$. Thus the equation $\left(u^{2^m} + u + \alpha\right)^{2^{m-1}} + u = \left(L_\beta(v) + \gamma\right)^{2^{m-1}} + Tr(v)$ can be reduced to a nonlinear system in the variables x_i. To break the encrypted text without knowing the secret key, it is enough to solve this nonlinear system but this attack is completely infeasible according to [1].

A second attack attempt would be the method based on differential cryptanalysis by which the MIC* cryptosystem was broken but according to [1] it fails at the Poly-Dragon cryptosystem. Also, due to the fact that many attacks are based on algorithms of exponential complexity (relinearization, XL, FXL, certain algorithms using the Gröbner basis) these are not feasible.

For all the above, the conclusion reached in the article [1] is that this cryptosystem based on multivariate cryptography is a secure one.

10 Related security solutions

As we mentioned at the beginning there are several post-quantum algorithms based on multivariate cryptography in the Dragon family such as Big Dragon, Little Dragon, Little Dragon Two.

In the case of the Little-Dragon algorithm the relationship between the encrypted text and the plaintext is given by the relation below where q is a power of a prime number, $\theta, \varphi = \overline{1, n-1}$ such that $q^\theta + q^\varphi - 1$ și $q^n - 1$ are coprime and $u = \phi^{-1} \circ L_1(x_1, x_2, \dots, x_n), v = \phi^{-1} \circ L_2(y_1, y_2, \dots, y_n)$, L_1, L_2 invertible linear transformations of ${\mathbb{F}_q}^n$ and $\phi : {\mathbb{F}_q}^n \to {\mathbb{F}_q}^n$ a triangular mapping:

$$v = u^{q^\theta + q^\varphi - 1} \tag{23}$$

Although the above relationship is simpler than the one used in the Poly-Dragon cryptosystem we did not choose this variant because as pointed out in article [11] there is an attack that can break the system.

As for the Big Dragon algorithm it is often vulnerable at least when $\psi(v)$ is public and if the function is secret then no attacks are known for this cryptosystem according to [12]. The relationship between the encrypted text and the plaintext is given by (24) where the differences from the previous method are: $\psi : {\mathbb{F}_q}^n \to {\mathbb{F}_q}^n$ is a nonlinear function such that $\frac{\psi(v)}{v}$ is a permutation of the set ${\mathbb{F}_q}^n \setminus \{0\}$ and $q^{\theta_1} + q^{\theta_2} - q^{\varphi_1} - q^{\varphi_2}$, $q^n - 1$ are coprime.

$$u^{q^{\theta_1} + q^{\theta_2}} v + u^{q^{\varphi_1} + q^{\varphi_2}} \psi(v) = 0 \tag{24}$$

For the Little Dragon Two cryptosystem the notations used are identical to the Poly-Dragon cryptosystem and the relationship between the ciphertext and the plaintext is given by the equation (25):

$$\left(u^{2^m} + u + \alpha\right)^{2^{m-1}} + u = v \tag{25}$$

The security of this cryptosystem is analyzed in the article [13] and the conclusion reached is that it is more effective than the previous version, but it is a safe version. Comparing this solution with Poly-Dragon, the preferred method for encrypting medical data in our platform, we could say that it is a more efficient and simpler solution but due to its complexity the Poly-Dragon cryptosystem is less likely to crack.

11 Conclusion

In this article we have presented a Web platform model through which we can secure and store a patient's complete medical history (visits, tests, letters of recommendation, the interventions in a private or public hospital) so that the doctor can establish a diagnosis as quickly and correctly as possible. In this way we would streamline and digitize the entire arduous process in today's healthcare.

To guarantee the security of access to the platform, respectively the confidentiality of data both now and when the quantum computer appears we considered the modified version of the TLS protocol that accepts a post-quantum algorithm (RLWE-PPK). We also consider homomorphic encryption to be able to process data anonymously in a Cloud environment and a cryptosystem based on multivariate cryptography, namely Poly-Dragon. This cryptosystem can be used for both encryption and signature generation due to the fact that it uses bijective algorithms. Hence all documents issued by a doctor can be digitally signed and recognized by any institution.

References

[1] R. P. Singh, A.Saikia and B.K.Sarma, "Poly-Dragon: An efficient Multivariate Public Key Cryptosystem," [Online]. Available: https://eprint.iacr.org/2009/587.pdf. [Accessed 08 11 2022].

[2] O. Monique, T. Claude and D. Pushkar, "Homomorphic Encryption," [Online]. Available: https://cyberleninka.org/article/n/502733/viewer. [Accessed 08 11 2022].

[3] L. Antonio, F. Tom, J. H. Adam and B. Ayoub, "What is homomorphic encryption?," 13 08 2020. [Online]. Available: https://blog.openmined.org/what-is-homomorphic-encryption/. [Accessed 08 11 2022].

[4] Y. Sophia, G. Vijay, S. Nabil, E. Shen and A. Yerukhimovich, "A Survey of Cryptographic Approaches to Securing Big-Data Analytics in the Cloud," [Online]. Available: https://www2.seas.gwu.edu/~arkady/papers/BigDataSecSurveyHPEC.pdf. [Accessed 08 11 2022].

[5] "Homomorphic Encryption Security Standard," An Open Industry / Government / Academic Consortium to Advance Secure Computation, 11 2018. [Online]. Available: https://homomorphicencryption.org/introduction/. [Accessed 22 11 2022].

[6] R. B. Mohd, S. Qi and L.-J. David, "A New Lightweight Homomorphic Encryption Scheme for Mobile Cloud Computing," [Online]. Available: https://www.flypig.co.uk/papers/MRB-QS-DLJ-2015.pdf. [Accessed 22 11 2022].

[7] M. A. Nia, A. Sajedi and A. Jamshidpey, "An Introduction to Digital Signature Schemes," [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1404/1404.2820.pdf. [Accessed 22 11 2022].

[8] D. Boneh and V. Shoup, "A Graduate Course in Applied Cryptography," [Online]. Available: https://crypto.stanford.edu/~dabo/cryptobook/BonehShoup_0_5.pdf. [Accessed 22 11 2022].

[9] X. Gao, J. Ding, L. Li, S. RV and J. Liu, "Efficient Implementation of Password-Based Authenticated Key Exchange from RLWE and Post-Quantum TLS," 05 07 2012. [Online]. Available: https://eprint.iacr.org/2017/1192.pdf. [Accessed 29 11 2022].

[10] G. Seroussi, "Table of Low-Weight Binary Irreducible Polynomials," 08 1998. [Online]. Available: https://www.hpl.hp.com/techreports/98/HPL-98-135.pdf. [Accessed 05 12 2022].

[11] N. Koblitz, Algebraic Aspects of Cryptography, Springer, 1998.

[12] R. P. Singh, "Permutation polynomials and their applications in cryptography," *Department of Mathematics, Indian Institute of Technology, Guwahati,* 2010.

[13] R. P. Singh, A.Saikia and B.K.Sarma, "Little Dragon Two: An efficient Multivariate Public Key Cryptosystem," 04 10 2009. [Online]. Available: https://eprint.iacr.org/2009/488.pdf. [Accessed 05 12 2022].