applied sciences

IMPACT FACTOR 2.5

CITESCORE 5.3

Article

A Quantum-Secure Cryptographic Algorithm Integrating Fractals and Prime Numbers

Gerardo Iovane, Elmo Benedetto and Antonio Di Lauro

Topic Collection
Innovation in Information Security

Edited by
Dr. Gianluca Lax and Dr. Antonia Russo

MDPI

https://doi.org/10.3390/app142210138

applied sciences

Article

A Quantum-Secure Cryptographic Algorithm Integrating Fractals and Prime Numbers

Gerardo Iovane *, Elmo Benedetto and Antonio Di Lauro

Department of Computer Science, University of Salerno, Via Giovanni Paolo II, 132, 84084 Fisciano, Italy; ebenedetto@unisa.it (E.B.); a.dilauro14@studenti.unisa.it (A.D.L.)
* Correspondence: giovane@unisa.com

Abstract: The present work introduces a new scheme of data cryptography in the context of emerging trends due to the challenge of defending critical network infrastructure against new exploit systems based on artificial intelligence or defending against quantum threats. In this paper, we will present an innovative cryptographic system composed of keys coming from fractals and prime numbers that are additionally manipulated through mathematical operations using matrices and quantum security. This technique ensures a high level of security, as demonstrated by the NIST p-values calculated on the key. This paper works upon the foundation on the previous work F&NIF (Fractal & Numerical Information Fusion), as we will discuss in the paper. In this work, we take this procedure and expand it with these new added features, using new fractal schemes and, in particular, implementing a novel quantum security procedure. This algorithm creates a security key applicable to cryptography that is resistant to quantum attacks since this procedure is quantum-crypto-agile.

Keywords: fractal cryptography; quantum cryptography; data encryption; prime numbers

Citation: Iovane, G.; Benedetto, E.; Di Lauro, A. A Quantum-Secure Cryptographic Algorithm Integrating Fractals and Prime Numbers. *Appl. Sci.* **2024**, *14*, 10138. https://doi.org/10.3390/app142210138

Academic Editor: Gianluca Lax

Received: 9 October 2024
Revised: 1 November 2024
Accepted: 5 November 2024
Published: 5 November 2024

1. Introduction

In recent years, along with the digitalization of information, the field of cryptography has become more and more important, and the protection and encryption of documents from public and private institutions must guarantee a high level of security. In [1], the authors introduced a new algorithm for extending secure coding based on prime numbers coming from RSA by using Information Fusion techniques and fractals. As we will see below after the introduction, we will extend the results in [1], and we will introduce two quantum properties, that is, the random evolution of the fragments of the key and the recombination of the fragment of the key to obtain a key that is resistant to quantum malicious attacks since the key becomes crypto-agile.

The CSPRNG (Cryptographically Secure Pseudo-Random Number Generator) uses techniques that generate random numbers suitable for cryptographic systems [2]. In this work, we have developed a new algorithm for generating a highly random key based on concepts like fractals, the fractal nature of the sequence of prime numbers [3], the RSA algorithm, and quantum security.

The idea of using fractals in cryptography is a relatively new idea that some researchers are exploring. Mathematical studies on fractals began in 1982 with Benoit Mandelbrot. He formalized this new type of geometry in his book [4]. Mandelbrot sets are widely used for cryptographical purposes. The study in [5] used a modified Mandelbrot set to generate a public key and a modified Julia set to generate a private key. Another study from 2020 used two Mandelbrot sets to create an encryption algorithm [6]. A Mandelbrot set was also utilized in [7] along with the Hilbert transformation to create a random encryption key. In 2019, an algorithm for encrypting images was proposed using two fractals: the Hilbert curve and the H-Tree [8]. These fractals were used in this study to shuffle the pixel positions and their values in an image. There are also types of studies, like ours, that propose the application of different combined techniques to obtaining cryptographic algorithm, with the aim of optimizing their effectiveness. In one study, for example, two algorithms based on fractals and chaos theory are combined to obtain an encoded image [9]. A 2023 work proposed the combination of the RSA algorithm, homomorphic cryptography, and chaotic maps to enhance image encryption [10]. Another work from 2024 combined an E-fractal and a hash function to create a cryptographic key [11]. Another recent article proposes the use of a 3D fractal cube and compressed sensing to create a compression encryption algorithm [12].

The choice of using fractals is motivated by the inherently complex and chaotic structure of these geometric shapes [13]. In fact, as is well known, a fractal F is a set that has the following properties:

- Self-similarity: F is an object that is similar to one or more of its parts. Indeed, F is the union of a certain number of parts that reproduce the entire initial F when sufficiently enlarged.
- Fine structure: F reveals details at every zoom.
- Irregularity: F cannot be described as a set of points that satisfy simple geometrical or analytical rules.
- Fractional dimension: A new dimension concept is defined for fractals, different from the classical conception; this new introduced dimension is called the "Hausdorff dimension" and is defined by a formula that uses the logarithms

$$d = \frac{\ln(N)}{\ln(s)}$$

with N equal to the number of parts into which the object can be divided and s the scale factor.

These properties make the idea of applying fractals in the field of cryptography interesting [14].

Another section of the algorithm concerns the generation of prime numbers. A relevant question in the field of mathematics in recent decades is that of the generation of primes and whether this can be represented by a deterministic scheme. In [15], we showed that the prime numbers follow a multiscale distribution, meaning that they can be classified in terms of tree structures. In [3], we showed that prime numbers can be seen as angles of a multifractal polygon based on a hexagon. All these results indicate that prime numbers follow a multiscale distribution and that starting from two sets (6k + 1 and 6k − 1) that contains all the prime numbers, we can generate other sets that are more likely to contain primes, and as we go down the decomposition hierarchy, the extent of the research decreases, and the computational time is also reduced.

Another feature that is guaranteed in the present work is quantum security, which makes the algorithm quantum-resistant. Quantum-resistant cryptography is cryptography that aims to create functions and protocols that allow the system to remain secure even if there is an attack from a CRQC (Cryptographically Relevant Quantum Computer), a quantum computer with features such that it can break public key cryptography systems, such as RSA [16]. Large quantum computers would theoretically have the ability to break algorithms like RSA using Shor's algorithm, a quantum algorithm that uses mathematical properties to efficiently factorize composite numbers into prime factors. The potential damage that CRQCs could inflict is motivating researchers in the field of cryptography to develop countermeasures, although at the moment, quantum computers are not so diffused [17].

Therefore, this paper uses as a starting point the technique proposed in [1] and extends it, likewise obtaining an algorithm for Information Fusion (IF) that is more efficient compared to that in the previous work. As is well known, Information Fusion is a relatively new research field, which is considered multidisciplinary and consists of taking data from more than one source and joining them together in order to obtain super-information. In the present work, thanks to our Information Fusion algorithm (IFA), we will obtain the following advantages:

- A security key driven by a numeric sub-key obtained via a multiscale algorithm for prime generation;
- A security sub-key based on different fractal algorithms;
- A security key given by fusion of the information from the two sub-keys.

The sub-keys follow a new dynamic process of evolution and recombination that gives them a crypto-agile property in the context of quantum computing. Compared to the state of the art, this work is the first to pave the way for a new approach that simultaneously combines fractals, prime numbers, and a cooperative–competitive quantum mechanism involving both relocation and recomposition in key construction. This new approach endows the solution with intrinsic crypto-agility and thus specific resistance to quantum attacks.

2. Construction of the Fractal Sub-Key

The algorithm randomly chooses between six different fractal sets listed below:

1. A Cantor set;
2. A Sierpinski set;
3. A Mandelbrot set;
4. A Peano set;
5. A Barnsley set;
6. A Vicsek set.

The first four fractals were already used in the previous work [1], while the last two are new additions. The addition of new fractals compared to the previous version aims to increase the randomization of the fractal component of the algorithm; moreover, while the choice of the fractal to use was human-driven in the previous solution, in this solution, the choice is made in a random and rolling mode.

In five of the six fractals listed above, the procedure for creating the number is using Iterated Function Systems (IFSs). An IFS fractal is a fractal generated by iterating a certain number of affine transformations. As is well known, an affine transformation in a plane is a biunivocal application that takes a point P of coordinates (x,y) and returns a point P′ of coordinates (X,Y) according to the following relation:

$$\begin{cases} X = ax + by + e \\ Y = cx + dy + f \end{cases}$$

In an equivalent way, in the matrix representation, we can write

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$$

where

$$det\begin{pmatrix} a & b \\ c & a \end{pmatrix} \neq 0$$

An affine transformation changes the position of a point in the plane. In this way, we can, for example, start from a geometrical figure and apply an affine transformation to every vertex of the figure, thus obtaining a translation, rotation, or distortion of the shape. In the case of the Mandelbrot set, we use a specific formula to generate points in the set. Let us show how the sub-key is created for every fractal.

2.1. Cantor Set

As usual, the construction of the Cantor set has the following procedure: We start from the [0, 1] interval, we divide it into three equal parts, and we delete the open middle third $\left(\frac{1}{3}, \frac{2}{3}\right)$. In this way, we obtain the set

$$C_1 = \left[0, \frac{1}{3}\right] U \left[\frac{2}{3}, 1\right]$$

If we repeat the same procedure for these new two intervals, we obtain

$$C_2 = \left[0, \frac{1}{9}\right] U \left[\frac{2}{9}, \frac{1}{3}\right] U \left[\frac{2}{3}, \frac{7}{9}\right] U \left[\frac{8}{9}, 1\right],$$

and so on.

Cantor set is defined as the intersections of all the obtained values C_n in iterating through the procedure infinitely. The affine transformations that are used to generate the sub-key are the following:

$$A_1 = \begin{cases} X = \frac{1}{3}x \\ Y = y = 0 \end{cases}$$

$$A_2 = \begin{cases} X = \frac{1}{3}x + \frac{2}{3} \\ Y = y = 0 \end{cases}$$

A graphical representation of the process is shown in Figure 1:

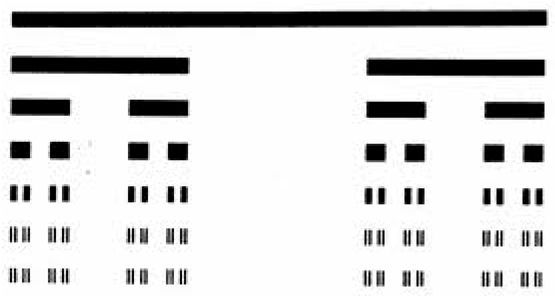


Figure 1. Cantor set iteration.

2.2. The Sierpinski Set

The Sierpinski set has already been explored for cryptographical purposes [18]. It is constructed starting from an equilateral triangle. It is divided into four congruent triangles, and then the central one is eliminated. The procedure is repeated infinitely with the remaining triangles (see Figure 2).

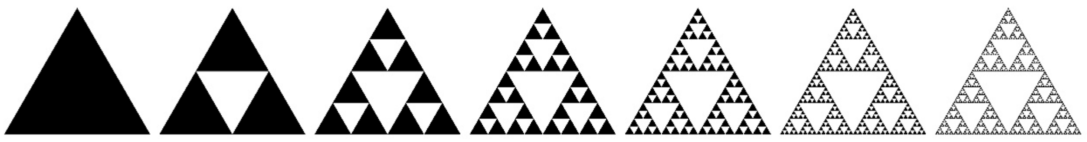


Figure 2. Construction of a Sierpinski triangle.

Sierpinski set is generated from its affine transformations:

$$A_1 = \begin{cases} X = \frac{1}{2}x \\ Y = \frac{1}{2}y \end{cases}$$

$$A_2 = \begin{cases} X = \frac{1}{2}x \\ Y = \frac{1}{2}y + \frac{1}{2} \end{cases}$$

$$A_3 = \begin{cases} X = \frac{1}{2}x + \frac{1}{2} \\ Y = \frac{1}{2}y + \frac{1}{2} \end{cases}$$

2.3. The Mandelbrot Set

Benoit Mandelbrot developed a set that is one of the most known fractals, generated using a recursive formula on the complex plane

$$\begin{cases} z_0 = 0 \\ z_{n+1} = z_n^2 + c \end{cases}$$

where c is a complex number, and it is in the Mandelbrot set if the succession does not tend to infinity. If we represent all the points in the complex plane, coloring all the points that belong to the set in black and the points that do not belong to the set with other colors, we obtain the Mandelbrot fractal, as shown in Figure 3.

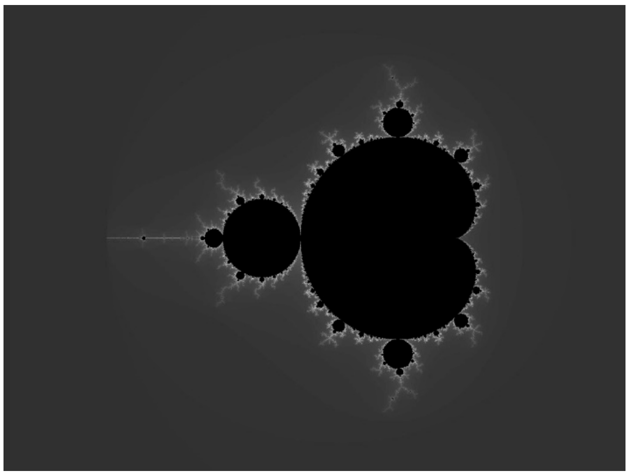


Figure 3. Mandelbrot fractal.

To generate points that belongs to the Mandelbrot fractal, the following procedure is used: We start from the principal formula for the cardioid of the Mandelbrot set. The representation of the cardioid in Cartesian coordinates is

$$x = \frac{1}{2}(cos(\theta) - 1)$$

$$y = \frac{1}{2}sin(\theta)$$

This formula generates the points on the cardioid's surface. Instead, we need to generate the points inside it. For this reason, we introduce a new parameter *r*, which varies from 0 to 1. Thus, we use a linear combination of two points: the origin (0,0) and a point on the surface of the cardioid. In this way, we can generate the points distributed inside the cardioid. The iterative procedure is the following:

- An angle θ between 0 and 2π is chosen randomly;
- A r parameter between 0 and 1 is chosen randomly to generate the points inside the cardioid;
- The following formulas are used to determine the real and imaginary parts:

$$x = r \times \frac{1}{2}(cos(\theta) - 1)$$

$$y = r \times \frac{1}{2}sin(\theta)$$

2.4. The Peano Set

The Peano curve also has already been used to create cryptographical algorithms [19]. This curve was the first example of a space-filling curve ever discovered. It is constructed iteratively in various steps, where at every step, a set of squares and the sequence of the centers of the squares are constructed, as obtained from the precedent step. Visually, this situation is obtained as shown in Figure 4.

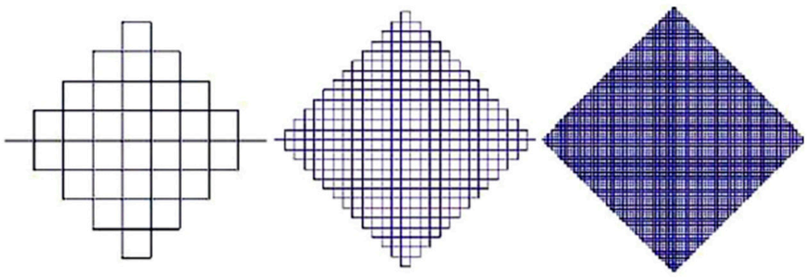
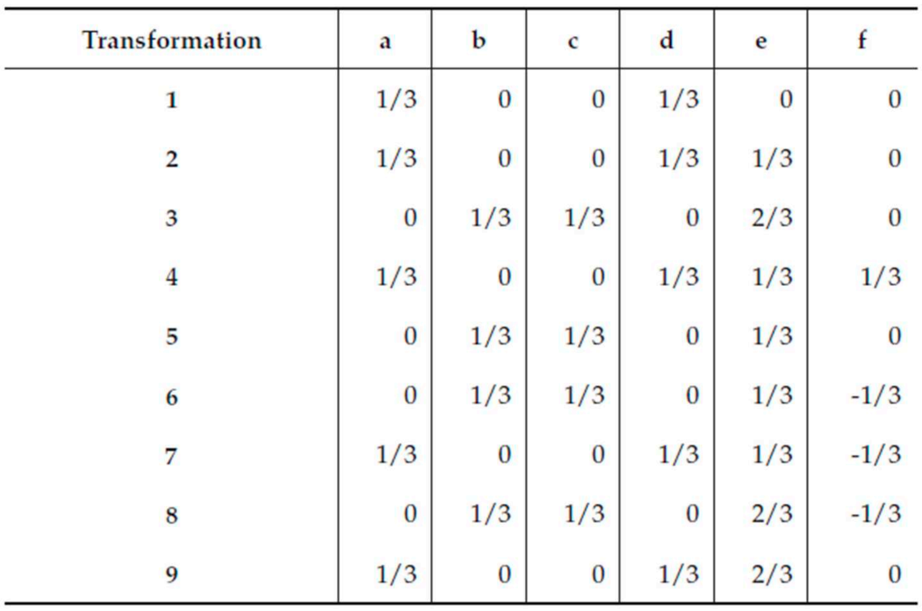


Figure 4. Peano curve construction.

The values of the Peano set are determined by iteratively repeating a set of nine affine transformations whose values are listed below in Figure 5.



Transformation	a	b	c	d	e	f
1	1/3	0	0	1/3	0	0
2	1/3	0	0	1/3	1/3	0
3	0	1/3	1/3	0	2/3	0
4	1/3	0	0	1/3	1/3	1/3
5	0	1/3	1/3	0	1/3	0
6	0	1/3	1/3	0	1/3	-1/3
7	1/3	0	0	1/3	1/3	-1/3
8	0	1/3	1/3	0	2/3	-1/3
9	1/3	0	0	1/3	2/3	0

Figure 5. Peano affine transformations.

2.5. The Barnsley Set

The Barnsley fern is another example of a fractal created by an Iterated Function System. Visually, it is the following shape, as shown in Figure 6.

Fractal confirms its statistical validity in generating random numbers [20]. In a similar way to the majority of the fractals above, it has an affine transformation, which we use to generate random numbers on the plane. The expressions are as follows.

$$\begin{cases} X = \frac{17}{20}x + \frac{1}{25}y \\ Y = \frac{23}{100}x + \frac{11}{50}y + \frac{8}{5} \end{cases}$$

$$\begin{cases} X = \frac{1}{5}x - \frac{13}{50}y \\ Y = \frac{23}{100}x + \frac{11}{50}y + \frac{8}{5} \end{cases}$$

$$\begin{cases} X = -\frac{3}{20}x + \frac{7}{25}y \\ Y = \frac{13}{50}x + \frac{6}{25}y + \frac{11}{25} \end{cases}$$

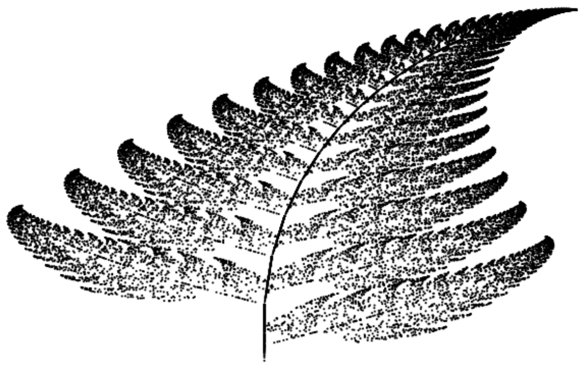


Figure 6. Barnsley fern.

2.6. The Vicsek Set

The last fractal used by our algorithm that can be chosen to generate a fractal sub-key is the Vicsek snowflake, as shown in Figure 7.

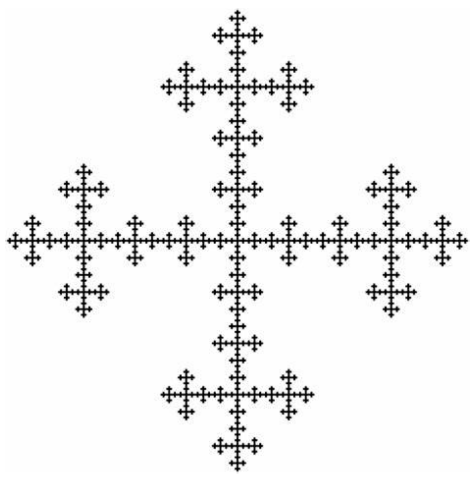


Figure 7. Vicsek fractal.

Affine transformations are the following:

$$\begin{cases} X = \frac{1}{3}x \\ Y = \frac{1}{3}y \end{cases}$$

$$\begin{cases} X = \frac{1}{3}x + \frac{2}{3} \\ Y = \frac{1}{3}y \end{cases}$$

$$\begin{cases} X = \frac{1}{3}x + \frac{1}{3} \\ Y = \frac{1}{3}y + \frac{1}{3} \end{cases}$$

$$\begin{cases} X = \frac{1}{3}x \\ Y = \frac{1}{3}y + \frac{2}{3} \end{cases}$$

$$\begin{cases} X = \frac{1}{3}x + \frac{2}{3} \\ Y = \frac{1}{3}y + \frac{2}{3} \end{cases}$$

3. A Multiscale Prime Generator Sub-Key

The other sub-key that we give as input to the quantum F&NIF algorithm is a product of primes that we obtain through the generation of two prime numbers, with each one generated thanks to a multiscale sieve, using the rules in [21]. We randomly generate a number $n \in$ N, and the sieving algorithm will return either n if it is prime or the prime number closest to n. The sieving algorithm is based on an automated procedure based on a proved theorem in [3], which asserts that each prime number can be written as the difference between the product of the first m primes and a prime obtained in the previous level of gridding. So, using a multiscale analysis, each prime number can be written as

$$p_{ij} = \left(\prod_{j=1}^{m} p_j\right) k - p_i, \qquad \textit{with } k \in N$$

where k is the counting index, p_i is a prime number obtained at the level $m - 1$, j is the level of decomposition, and p_{ij} the prime number obtained. So, the algorithm stores the candidate prime numbers useful for multiscale generation gradually, up to the nearest prime number of the input. Therefore, the key elements to the algorithm are the following:

- $$\left(\prod_{j=1}^{m} p_j\right)$$
  is the product of the first m primes that represents the multiscale level and is the basis for generating the prime candidates.
- At each multiscale level, k can range from 1 to a value k_{max} that changes at every level, and specifically, the last k_{max} takes the value of p_{m+1} that is the prime number that serves to generate the next multiscale level.
- After obtaining the product between the multiscale level and k, we subtract p_i, which is each prime number obtained at the previous multiscale level.

As shown in the recent work [21], the multiscale algorithm proposed here is not only applicable but even more efficient than other sieves.

4. Overview of the Infrastructure

The system for obtaining the cryptographic key is composed of four components:

- A fractal numerical algorithm for generating the fractal sub-key, which randomly chooses one of the fractals listed above;
- The RSA algorithm for creating the numerical code and the private key, which serves in some of the processes in the F&NIF part, as we will see below;
- A multiscale algorithm for generating another number, which is then concatenated to the fractal number;
- The quantum F&NIF algorithm, which includes the mathematical operations on the matrices, transformation at a random time, and quantum security.

A diagram of the quantum F&NIF algorithm is shown in Figure 8.

F&NIF algorithm takes three numerical vectors as its input: the first one is generated by the fractal algorithm; the other is the module, that is, the product of two primes p and q generated by the multiscale algorithm; and the third one is the private RSA key generated through execution of the RSA algorithm. The size of the first two vectors is selected automatically according to the following relation:

$$dimModule = \frac{dim}{2}$$

$$dimFract = dim - dimModule$$

with

dim: The dimension of the new cryptographic key;
dimModule: The dimension of the RSA module's sub-key;
dimFract: The dimension of the fractal sub-key.

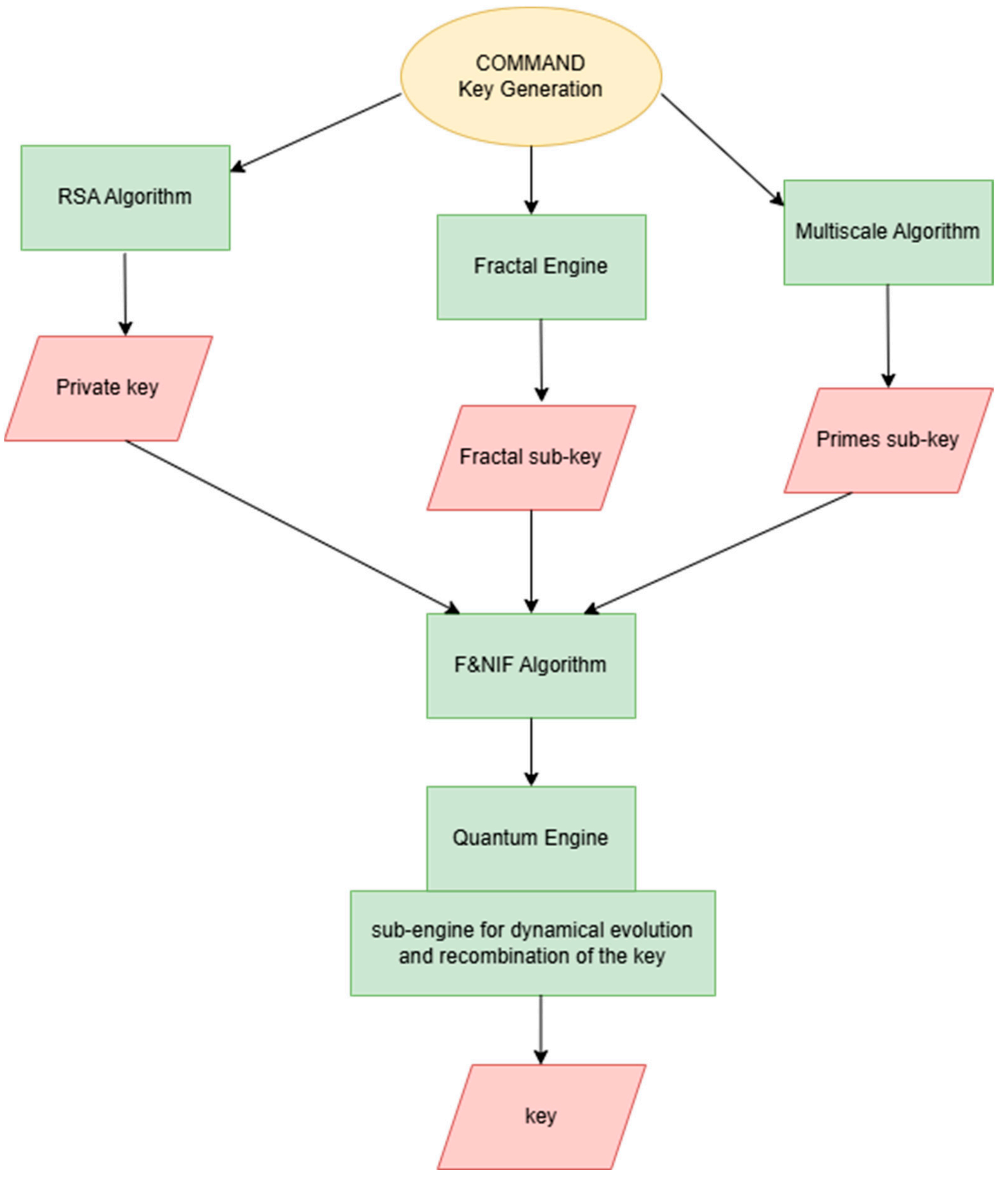


Figure 8. Scheme of the novel algorithm quantum F&NIF.

The multiscale algorithm, as discussed above, consists of generating two prime numbers, the product of which is then concatenated to the fractal number to obtain one single number. So, the algorithm consists of a procedure that is repeated twice, once for each prime number. In the introduction, we explained that this procedure is based on the findings described in the previous section, and the primes generated belong to a particular set, which is determined by the multiscale level, the alpha number, and k. Below is an image of how a multiscale hierarchy is created using the rules in [3], and this was presented in the previous section (see Figure 9).

At this point, we have all the necessary ingredients to implement the new quantum F&NIF algorithm:

- The product of the primes;
- The RSA private key;
- The number generated from the fractals.

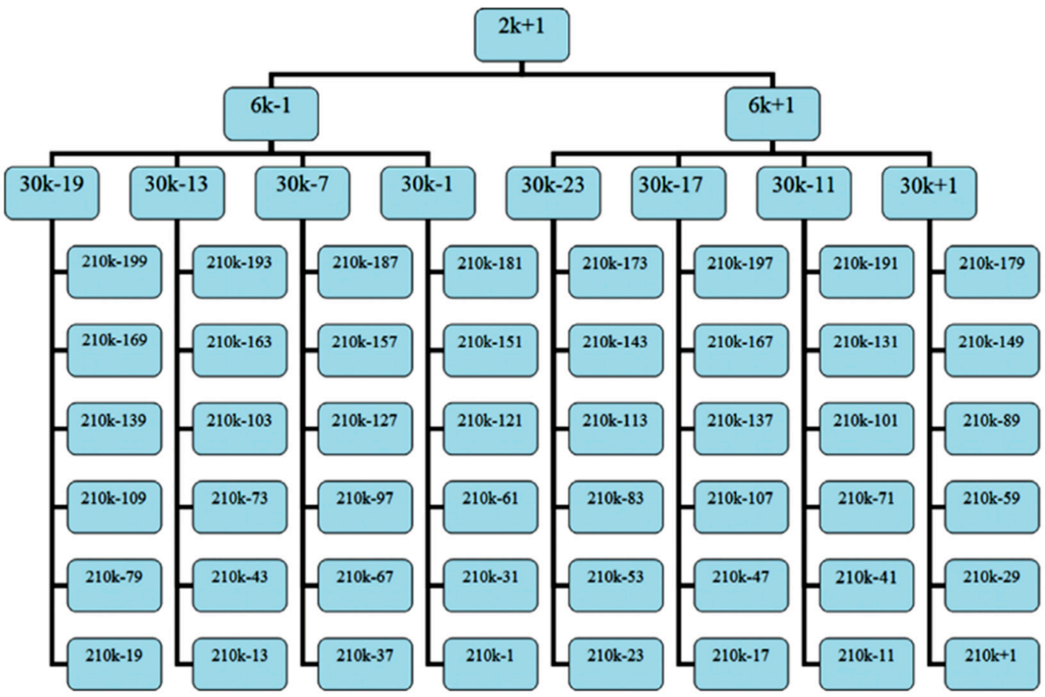


Figure 9. Multiscale hierarchy for generating primes.

The first step is to concatenate the product of the primes and the fractal number. For simplicity, we call this new obtained number "hybrid". Now, we need to arrange the hybrid number into a squared matrix. The dimension of the matrix is determined by finding the next square number from the number of digits of the hybrid. So, if, for example, the hybrid has a dimension 14, the matrix has a dimension 16. Now, we need to add the remaining padding elements at the end of the matrix, and their values are selected in this way: We travel through the RSA private key, and the digits that we find, one at a time, represent the index of the element of the hybrid that we add. So, in a programmatic way, if the first digit of the RSA private key is 2, then, the element hybrid [2], so the second digit of the hybrid, is added in a padding position. This is conducted until the padding elements are exhausted. By considering $a \in \mathrm{Z}^{\mathrm{m}}$ and $b \in \mathrm{Z}^{\mathrm{n}}$ as the two vectors representing the multiscale sub-key and the fractal sub-key, and $\mathrm{l} \in \mathrm{Z}$ as

$$l = m + n$$

the number of padding elements p in the matrix is calculated as

$$p = dim - l$$

where *dim* is the dimension of the matrix, obtained from the next square number mentioned above, and so two cases can be distinguished:

$$\begin{cases} p = 0 & no\ added\ padding \\ p > 0 & addition\ of\ p\ numbers\ of\ padding \end{cases}$$

Consequently, the number of rows of the matrix is

$$rows = \sqrt{dim}$$

Then, we construct another matrix, a permutation matrix. A permutation matrix is a matrix that is obtained by swapping some rows or columns in the identity matrix. The matrix thus must be composed of all zeros and a one in every row in different positions.

In our algorithm, this matrix is constructed from the RSA private key; by scanning one digit at time, this represents the index at which the one is positioned in the row. Since the private key is composed of equal digits, we perform two controls on the private key before constructing the matrix: the first is to make sure that the length of the key is at least equal to the number of rows/columns in the matrix. In the second, we verify whether there are equal elements in the key. This because the ones that we position in the matrix must all have different positions; otherwise, the matrix will no longer be an identity matrix with some rows/columns changed. If we find numbers that are already present in the key, we substitute them with a number generated randomly that is not present. After constructing the permutation matrix, we multiply the hybrid matrix and the permutation matrix.

The algorithm then continues on to the quantum security part, which we anticipated in the introduction. The technique used to guarantee this property regards operations inside the newly obtained product matrix, and it draws inspiration from quantum mechanics, specifically the scattering theory. The technique consists of creating two fragments of matrix digits of a certain length and "colliding" them to create two new fragments, mixing the digits in them. In this case, the crypto-agility consists of dynamically evolving each fragment, which means that at different times, the fragment will be in a different position from its initial allocation. For each fragment created and at every moment the daemon is free from a task, it takes two fragments, *FrA* and *FrB*, which have two contents, *Ca* and *Cb*, and gives as output two new fragments *FrC* and *FrD* with contents *Cc* and *Cd*. Figure 10 shows this interaction.

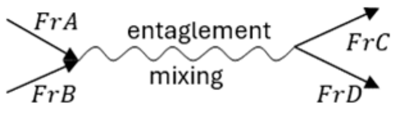


Figure 10. Fragment interaction.

The algorithm we used randomly chooses one of these two operations: the first is swapping rows m and $m + 1$ in the matrix, with m randomly chosen at each specific time; the second is taking rows m and $m + 1$ with m randomly chosen and subdividing them into specific amounts of the total, for example, 30% and the 70% of the digits. So, we obtain four fragments, which we collide two at a time; mix the content; and recombine them with a cross-shifting operation, where the first 70% of the row m is merged with 30% of the row $m + 1$ and 70% of the row $m + 1$ is merged with 30% of the row m. Finally, we unite, row by row, the matrix digits, obtaining the new cryptographic key.

5. Statistical Tests

To demonstrate the randomness of the keys generated, we applied the NIST statistical tests [22]; in particular, we calculated the p-value parameter for four different tests.

- Frequency: This calculates the number of 0-bits and 1-bits of the key, and it is established using a formula if their proportion is compatible with a random number.
- Block frequency: This calculates the proportion of 0-bits and 1-bits that is present in M-bit-size blocks extracted from the sequence.
- Cumulative sum: The parameter of the cumulative sum is calculated in this way: An M-bit size is considered and, replacing the 0-bits with −1-bits, the bit-by-bit sum is executed, obtaining the maximum distance from 0.
- Longest run: This is a parameter that indicates the maximum length of consecutive 1-bits inside an M-bit-size block inside the sequence.

In all the tests run, all the p-values of the parameters were greater than 0.01, and this indicated the cryptographic key is actually random. In Table 1, the p-values obtained from the tests performed for each fractal are reported.

Table 1. *p*-values for each fractal.

	Frequency	Block Frequency	Cumulative Sum	Longest Run
Cantor	0.562347	0.384479	0.512504	0.336289
Sierpinski	0.276303	0.379936	0.359890	0.845236
Mandelbrot	0.425819	0.319567	0.116564	0.617954
Peano	0.606979	0.757654	0.437202	0.782429
Barnsley	0.635501	0.434880	0.218350	0.509799
Vicsek	0.277132	0.381236	0.195538	0.782429

Figure 11 below shows a graph that provides a visual representation of the statistical values found for each fractal:

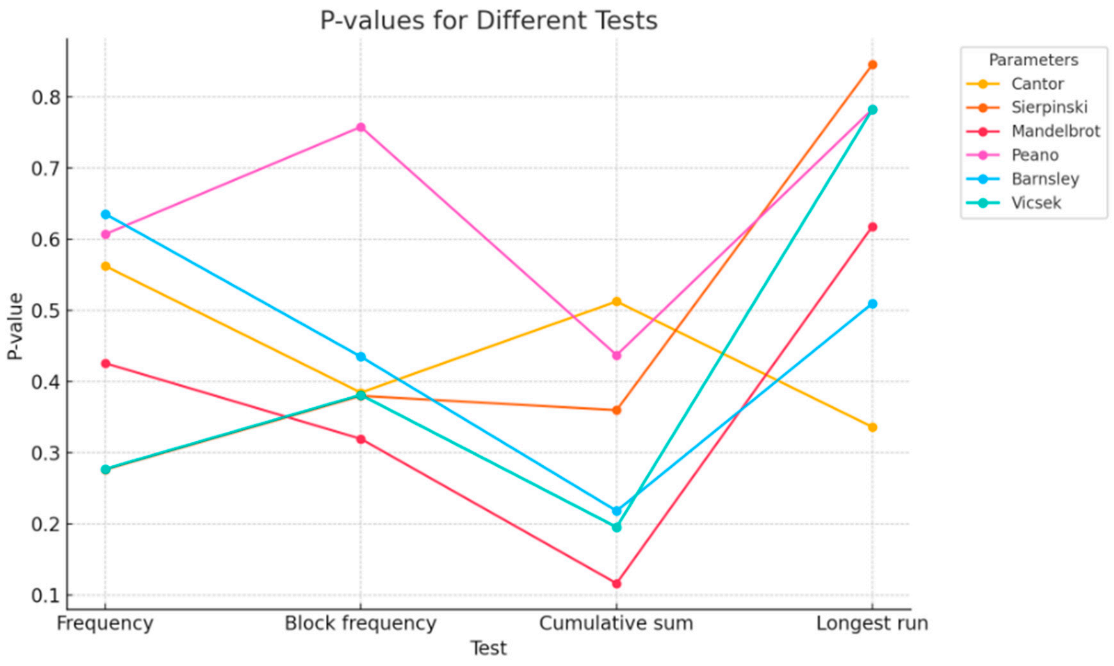


Figure 11. *p*-value comparison chart.

6. Conclusions

The purpose of this work was to develop a new cryptographic algorithm starting from our work in [1]. With respect to the algorithm in [1], we extended the number of fractals, we introduced a sub-algorithm for the choice of fractals that was random and rolling, and then we introduced a new multiscale sieve for prime generation, but the most relevant addition was the introduction of a sub-engine, which introduced two quantum properties useful for its crypto-agility, that is, quantum evolution of the key and quantum recombination. The NIST tests on the cryptographic key confirmed in all cases the randomness of the latter; this was also true for the new fractals added, Barnsley and Vicsek. Consequently, with the quantum security technique, an additional level of security is guaranteed by further varying the position of the components of the key and making it more resistant to the possibility of violation via quantum attack. This makes the security key applicable in the context of cryptography of the next generation.

Author Contributions: Conceptualization, G.I., E.B. and A.D.L.; Writing—review & editing, G.I., E.B. and A.D.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data is contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Iovane, G.; Amorosia, A.; Benedetto, E.; Lamponi, G. An Information Fusion approach based on prime numbers coming from RSA algorithm and Fractals for secure coding. *J. Discret. Math. Sci. Cryptogr.* **2015**, *18*, 455–479. [CrossRef]
2. Ruggeri, N. *Principles of Pseudo-Random Number Generation in Cryptography*; University of Chicago: Chicago, IL, USA, 2006.
3. Iovane, G. The set of prime numbers: Multifractals and multiscale analysis. *Chaos Solitons Fractals* **2009**, *42*, 1945–1958. [CrossRef]
4. Mandelbrot, B.B. *The Fractal Geometry of Nature*; W.H. Freeman and Company: New York, NY, USA, 1982.
5. Agarwal, S. Symmetric key encryption using iterated fractal functions. *Int. J. Comput. Netw. Inf. Secur.* **2017**, *9*, 1–9. [CrossRef]
6. Agarwal, S. A new composite fractal function and its application in image encryption. *J. Imaging* **2020**, *6*, 70. [CrossRef] [PubMed]
7. Sun, Y.Y.; Kong, R.Q.; Wang, X.Y.; Bi, L.C. An image encryption algorithm utilizing Mandelbrot set. In Proceedings of the 2010 International Workshop on Chaos-Fractal Theories and Applications, Kunming, China, 29–31 October 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 170–173. [CrossRef]
8. Zhang, X.; Wang, L.; Zhou, Z.; Niu, Y. A chaos-based image encryption technique utilizing hilbert curves and h-fractals. *IEEE Access* **2019**, *7*, 74734–74746. [CrossRef]
9. Halagowda, S.; Lakshminarayana, S.K.; Lakshminarayana, S. Image encryption method based on hybrid fractal-chaos algorithm. *Int. J. Intell. Eng. Syst.* **2017**, *10*, 221–229. [CrossRef]
10. Mfungo, D.E.; Fu, X. Fractal-based hybrid cryptosystem: Enhancing image encryption with RSA, homomorphic encryption, and chaotic maps. *Entropy* **2023**, *25*, 1478. [CrossRef] [PubMed]
11. Kadhim, O.N.; Najjar, F.H.; Ramadhan, A.J. Secure Image Encryption using E-Fractal-Based Non-Commutative Group and Hash Function. *BIO Web Conf.* **2024**, *97*, 00020. [CrossRef]
12. Lu, Y.; Gong, M.; Cao, L.; Gan, Z.; Chai, X.; Li, A. Exploiting 3D fractal cube and chaos for effective multi-image compression and encryption. *J. King Saud Univ.-Comput. Inf. Sci.* **2023**, *35*, 37–58. [CrossRef]
13. Barnsley, M. *Fractals Everywhere*; Academic Press: Boston, MA, USA, 1988.
14. Huntress, G.B. Encryption Using Fractal Key. U.S. Patent 6,782,101 B1, 24 August 2004.
15. Iovane, G. The set of prime numbers: Multiscale analysis and numeric accelerators. *Chaos Solitons Fractals* **2009**, *41*, 1953–1965. [CrossRef]
16. Dam, D.T.; Tran, T.H.; Hoang, V.P.; Pham, C.K.; Hoang, T.T. A survey of post-quantum cryptography: Start of a new race. *Cryptography* **2023**, *7*, 40. [CrossRef]
17. Mattsson, J.P.; Smeets, B.; Thormarker, E. Quantum-resistant cryptography. *arXiv* **2021**, arXiv:2112.00399. [CrossRef]
18. Jhansi Rani, P.; Durga Bhavani, S. Symmetric Encryption Using Sierpinski Fractal Geometry. In Proceedings of the Computer Networks and Intelligent Computing: 5th International Conference on Information Processing, ICIP 2011, Bangalore, India, 5–7 August 2011; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2011; Volume 157, p. 240. [CrossRef]
19. Ancona, F.; De Gloria, A.; Zunino, R. Parallel VLSI architectures for cryptographic systems. In Proceedings of the Great Lakes Symposium on VLSI, Urbana-Champaign, IL, USA, 13–15 March 1997; IEEE: Piscataway, NJ, USA, 1997; pp. 176–181. [CrossRef]
20. Çiçek, S. Microcontroller-based random number generator implementation by using discrete chaotic maps. *Sak. Univ. J. Sci.* **2020**, *24*, 832–844. [CrossRef]
21. Iovane, G.; Benedetto, E.; Gallo, C. Multiscale Sieve For smart prime generation and application in Info-Security, IOT and Blockchain. *Appl. Sci.* **2024**, *14*, 8983. [CrossRef]
22. Rukhin, A. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*; US Department of Commerce, Technology Administration, National Institute of Standards and Technology: Gaithersburg, MD, USA, 2001; Volume 22.