August 2019

“Chaos is order to be deciphered”.

José Saramago

A radically new conceptualization is presented for the development of cryptographic models, based on the introduction of new transposition and substitution techniques of the basic unit of information: The bit. The cryptographic universes, where each universe generates a gigantic set of encoders (versions or connection points) and each encoder generates, in turn, an equally gigantic number of encryption algorithms. In addition, the concepts of DNA, Three Dimensional Bolt, Necessary Noise, Recombinant Algorithm, Pivot, Cycle are introduced. And a new generator of random numbers: Cylinder(). Likewise, the use of information structures with geometries of 1, 2 and 3 dimensions should be noted. Throughout this document calculations of complexity are presented about the current universe (as sample) of Cyphertop, to give the reader an idea of the impossibility of any computational attempt to violate CYPHERTOP. The figures presented here correspond to a sample universe of Cyphertop.

1. Introduction.

The cryptographic models have evolved throughout the history of mankind because of the obvious need for safekeeping highly sensitive information.

In ancient times techniques were used that while they seem simple today, at their time they were effective. It was the transposition of symbols from one position to another, according to a predefined pattern or substitution of symbols. The first attempt to formalize a mathematically complex model was done during the Second World War:

The famous machine Enigma, built by the Germans posed a challenge of huge proportions to British intelligence. The British possessed the Typex machine and the Americans the M-135-C machine. As for the Japanese, they counted with the transpositions machine of Ito. Cryptography makes a spectacular leap with the appearance of the integrated circuit computers.

Models like the DES, AES, RSA, BLOWFISH appear and the CCE (Elliptic Curves Cryptography) is implemented, only to mention a few of them. Such is the current scene. Quantum cryptography begins to develop its first implementations, basically, it takes charge of using the quantum properties of photons for the safe sending of the key to be used by a pair of users for the encrypting of a file using symmetric cryptography. But one concept has remained invariant throughout human history regarding cryptography:

Formerly it was called method or procedure of concealment of information. At the dawn of the century, the cryptographic machines already described appeared and when the computer appeared it was called an algorithm. Today the encryption algorithms dominate the scenario, be it as software implementations or as integrated circuits that contain specialized processors for applying one algorithm or another.

2. The Cyphertop model: Breaking the paradigm.

Years back, when it was decided to create a cryptographic model, among other goals set, there were the following:

- Design an algorithm whose complexity was such, that it would render all brute force attacks useless.
- It had to be resistant to the already common eventuality that its source code fell in the hands of the enemy.
- It should be of easy implementation and replacement.
- Resistant to incipient quantum computing.

Thus, as the advance was made in the construction of techniques of fast movement of information in the memory of the computer (in the way of bits) and models to generate lists of random numbers so that with them, dispersion tables could be built, new ideas appeared to improve the proposal. Cyphertop, as the model was called, made no concessions with the initial goals. Furthermore, the improvements introduced made it possible to establish more ambitious goals.

3. Cyphertop: The destruction of the Encryption Algorithm concept.

When there is talk about encrypting, the immediate and obligated figure is the encryption algorithm.

A tool with which we encrypt a file and assuming that that tool could be generalized as a function, its inverse, f^{-1} serves to decrypt the corresponding encryption. At an earlier stage to the design of Cyphertop, a figure called recombinant mini algorithm was implemented, which consisted of several small algorithms of transposition of bits on structures of 1, 2 and 3 dimensions.

When we say recombinant we are referring to its position within a general algorithm. They are called to work on the data in a random way following one another without a predefined order, within the source code. It was precisely this figure that allowed the formulation of a daring hypothesis at its time:

Is it possible to develop an encryption algorithm that is not automorphic in time, that is, that it does not remain the same through time?

The known encryption algorithms are automorphic in time. That is, they perform in the same way every time they are called to work on a set of data. AES for example, provides a key for every pair of users, if user A sends the same file n times to his partner user B, the n encryptions of the file will be equal. The challenge of AES lies in the fact that currently there are not enough computational resources to cover the space of solutions of the key (128 or 256) bits in a reasonable time.

Escalating the challenge, the Cyphertop model makes the following assumption:

There are enough computational resources to attack algorithms of the AES type, with keys as long as 2^{15} = 32,768 bits!!

A key of this size would be impractical in the best of cases, but however, as an illustration, we refer to the fact that the National Security Agency (NSA) of the United States of America, has set itself as a challenge at the present time to count with the computational and algorithmic

infrastructure to break the AES 256, that is with power of 2^8, in times as reasonably short as to make useless encrypted information.

Cyphertop by far exceeds the previous assumption:

Its space for the key, if we want to use the terminology of the AES type algorithms, is superior to the 200,000 bits and if we want to be more rigorous, the previous figure can only be taken as a reference since the notion of block of data that is handled by the AES type algorithms in Cyphertop is impossible to apply.

The concept of the Cryptographic universe has then been developed. Where every universe can generate an enormous number of versions or connection points that are in reality, each one of them, generators of encryption algorithms to provide secure communication between two users of a communications network. Each version or connection point can generate a huge number of Encryption Algorithms or **AC** (Spanish acronym for algoritmo de cifrado).

For the present Sample universe that was developed for demonstration purposes:

AC = $2.085924830 \times 10^{93}$

This number is obtained from the possible combinations of 10 random numbers that act as initial parameters (modifying numbers) and they are generated by the Cyphertop code every time that we want to encrypt a set of data. These numbers modify the DNA of the respective connection point that is being used in the encryption.

4. Cyphertop: Generator of algorithms.

Cyphertop does not have a defined morphology. Every encryption will be done by fixed routines that are common to all encryptions and also, by a set of recombinant Mini algorithms. This set is different for each particular encryption.

A Recombinant mini algorithm is a small software routine that takes care of transposing bits within a data structure. These mini algorithms follow one after the other in different combinations for each occasion.

A **k** number of these mini algorithms is established each one of them in charge of generating dispersion (transposition) of bits. These **k** mini algorithms establish their order of appearance according to the initial parameters of encryption (the 10 numbers already mentioned, modifying numbers) and their sequence of appearance, therefore, will vary for each encryption. Hence the name of Recombinants.

The order of appearance of the recombinant mini algorithms and the addressing of the data each recombinant mini algorithms makes depends on the Cylinder() function, which is the generator of pseudo random numbers implemented for Cyphertop.

Establishing a comparison, the AES 256 Algorithm, for example, is composed of 14 rounds or stages and for that, it uses structures of 1 o 2 dimensions: Vectors and matrices. All the encodings under AES rigorously repeat that scheme. In AES, the operations that are carried out within the rounds we have: rotations of bits on vectors, substitution of bits on matrices, XOR operations and linear transformations (multiplications).

The way in which these operations are done depends on an initial key, which is used to generate subkeys. But what is relevant here is that, given an initial key (which serves to communicate at least two nodes) and any set of data, if that set is encrypted **n** times, the **n** encryptions will be equal between them.

Back to Cyphertop: How many combinations are possible for these recombinant mini algorithms?

Depending on the Cyphertop universe in particular and for the present case, with the sample universe set here, we have that k = 1024 therefore, the Possible Combinations of the appearance of these recombinant mini algorithms are**:** **CP = 1024!** Which is:

$5{,}41852879605885728307692 \times 10^{2.639}$

Possible combinations. The probability of the repetition of sequences of recombinant algorithms will be quite small in practice. And if they are repeated in the order of occurrence they would still have to repeat the form of working of each recombinant mini algorithm. Each one of these mini algorithms can use information structures of 1 to 3 dimensions where the data is moved within these structures.

In each encryption, these recombinant mini algorithms, therefore, conform along with the "Xored" routines (XOR operations), complement and movement of bytes (dispersion of bytes) what is defined as a particular encryption algorithm for that occasion and defined by the initial parameters. It should be noted that the **AC** algorithms generated by a version or connection point of Cyphertop, based on the initial parameters (the ten modifying numbers) are different to the **AC** algorithms that may be generated by another version or connection point of a given universe of Cyphertop with those same parameters (modifying numbers).

Therefore, it is not viable talking about Algorithm in the Cyphertop system, like we do with other encryption systems. Cyphertop is not intrinsically an encryption algorithm but a generator of encryption algorithms since it produces the same results that would be seen if multiple encryption algorithms were applied on the same file.

11. The pseudo random machine Cylinder.

In C++ there exists an instruction: rand() which enables generating pseudo random numbers and a random() function for generating random numbers. Cyphertop has its own system for generation of pseudo random numbers:

Cylinder, which consists of a three-dimensional structure (cylinders) particular for each version, and based on the DNA of that version. The Cylinder is built using the DNA content and the initial parameters (modifying numbers) that are generated for each encryption, as it name indicates, modify the content of that Cylinder. The Cylinder is used to generate the pseudo random numbers series throughout the encryption process.

The numbers produced by Cylinder are pseudo random, but Cylinder is of such complexity that we can assure the name "random" for them. Besides, in each encryption process, its operation varies according to the initial parameters (the ten numbers).

Given the complexity of algorithms and data involved in Cylinder, the pseudo random numbers that it generates can be considered as "Random", since the process to generate them through the Cylinder() function is highly complex and does not obey any mathematical formula or procedure,

instead it is the product of the movement of the rotational mechanism (cylinders) that has been constructed with the modifying numbers and the DNA, which are both really random and are different every time that the rotational mechanism is built. As an additional note, an interesting fact is that the randomness tests to which the random numbers produced by Cylinder were subjected to, all yielded a high randomness index.

To invoke this procedure the function Cylinder() is used and just like any randomic function, the range within which the requested random number is desired must be specified.

Cyphertop encrypts two consecutive times the same data set (Double cycle, which is explained later) and for each cycle, it uses five (5) of the ten modifying numbers. Therefore, it has at its disposal **CylC** = $4.567192606 \times 10^{46}$ random lists or ways of configuring the Cylinder mechanism, by cycle.

Every time that a set of data is going to be coded, Cylinder is initiated in **CylC** different possible ways, according to five (5) of the 10 initial parameters for that encoding. Being

CylC = $4.567192606 \times 10^{46}$.

Since there are two cycles, for every encryption the Cylinder() function provides **AC** random lists or ways of configuring the mechanism of cylinders:

AC = $\mathbf{(CylC)^2} = (4.567192606 \times 10^{46})^2 = 2.085924830 \times 10^{93}$

The Cylinder() function then has **AC** series of pseudo-random numbers at its disposal for every connection point, while rand() of C++ only has $2^{15} = 32{,}768$ different and invariant series for all its users. In addition, the implementation of rand() varies from compiler to compiler, which makes it ineligible for encryption purposes.

5. The DNA concept in Cyphertop

For the construction of the pseudo random machine Cylinder and for other labors of Cyphertop, a set of numbers (DNA) is required which are created at the installation of each version or connection point.

The lists of numbers and its extension vary from universe to universe and are called the generic DNA of each Cyphertop Universe. And as already stated, the content of the DNA varies from one connection point to another. The capture of the DNA is done taking into account data like the system's clock, the cycles of the processor, the UUID or the GUID numbers produced by the operational system, the identifier of processes and randomic variables that the C++ language provides. On mobile devices, you can go to the accelerometers of the device. The tables of data thus generated are unique and unrepeatable for each version of Cyphertop. The present universe of Cyphertop contains 25,009 numbers distributed in three lists that comprise its DNA.

6. A Universe of Cyphertop: A generator of Versions or Connection Points.

Just as each version of Cyphertop is a generator en encryption algorithms, each universe of Cyphertop is a generator of versions or connection points. For the sample universe, it is possible to generate:

PC = 8.21479634 x $10^{60.224}$

Versions or Connection Points (**PC**). And what differentiates one version from another within the same Cyphertop universe is the internal data that are stored when that version is generated. This internal data is nothing more than the "DNA" of that version or connection point that as it was said before consists of collections of numbers that have been captured by the Cyphertop system at the moment of installing the new version or connection point.

In biology, the DNA differentiates each living being from the others. In Cyphertop the DNA differentiates one connection point from others. A universe of Cyphertop produces **AC x PC** different encryptions of one same file or set of data. That is, all the encryptions of the same file that all the connection points of that universe can produce. The present sample universe has:

AC x PC = 1.7135447663 x $10^{60.318}$

Forms of encrypting a data set.

7. Cyphertop Universes: How many are there?

George Cantor, when working with infinite sets was able to prove that larger infinites than others exist. Cantor discovered that those do not always have the same size, that is, the same cardinal: for example, the set of the not enumerable rationals is of the same size than the set of the natural ones, while that of the real is not.

Therefore, there exist several infinites, some larger than others. It is possible to create as many universes as multidimensional geometrical structures and algorithms of manipulating data between them and they can combine among them. The portions of Cyphertop code that have to do with the manipulation of these structures vary from one universe to another. Likewise, the set of recombinant mini algorithms is different for each universe of Cyphertop. Finally, the prototype of DNA is unique for each universe.

The initial parameters used in each Cyphertop encryption vary from one universe to another. The model of generation of random numbers selected by the Cyphertop system, the Cylinder() function also varies from universe to universe.

8. The necessary noise.

The majority of the encryption algorithms do not introduce noise into the encrypted contents. Cyphertop, on the other hand, does. Not only does it. Cyphertop introduce a random quantity of noise (garbage bits) every time it executes an encryption, but also those bits can be found in the encrypted contents, virtually in any place.

With it, an additional uncertainty level is generated which will help to prevent all attempts to formalize patterns trying to collect enough information to achieve decoding. This noise or garbage allows masking the true length of the set of encrypted data.

9. Pivots.

Pivot is defined here as a fixed value that could be found in a determined place of an encrypted file.

In order for the decoder code of Cyphertop to find the form in which the encryptor code proceeded over a set of data, it is indispensable that the encrypter "transmits", within the encrypted file, the parameters that it used on that occasion in particular to encrypt that set of data (modifying numbers).

Such parameters are constituted by a set of numbers taken randomly by the encrypting code, in reality, numerical variables that are generated by the interaction of the user with the computer at the moment of encrypting.

This set of data (modifying numbers) is hidden inside the Cyphertop encryption, so that the decoding code "works", in an inverse symmetric way (f^{-1}) over the encrypted data and so it can obtain the original text sent.

There are no fixed positions within an ecryption of Cyphertop where the bits that conform this set of numbers can be placed, that is, there are no pivots. In Cyphertop, the encryption of a set of data has three types of information:

1. Bits corresponding to the random numbers that constitute the parameters of the encryption. (Modifying numbers).

2. Garbage bits that are generated in a random way.

3. Bits corresponding to the encryption per se, of the set of data.

The bits corresponding to the three types that have just been stated experience throughout the encryption process, several movements through information structures of 1 to 3 dimensions. In addition, they undergo several XOR processes with text likewise elaborated with figures provided by the Cylinder() function and finally complement processes. Of the above it turns out:

- One bit of any of the three types can be located in any place within an extension greater than 320 million bits. If the file to encrypt is equal to or greater than this figure.

- Throughout the encrypting processes, this same bit has a 50 percent probability of mutating its original value, that is, if it is 0 of mutating to 1 and if it is 1 of mutating to 0.

- It is impossible, because of the way the Cyphertop system works, to divide the encrypted file into "Blocks" for its analysis.

Given the premises above, no significant bit of one Cyphertop encryption has a fixed position. Inside one Cyphertop encryption there are no "reference positions", that is, there are no Pivots.

10. The Cycles of Cyphertop.

Every text processed by AES 256 experiences 14 rounds of encryption. Each round is equal to the preceding one except the initial and final round, which differs from the other intermediate ones.

In Cyphertop the data are subjected to movements of bits through geometrical structures, using a sequence of recombinant algorithms, a giant table of dispersion of bytes and finally, XOR and complement operations.

All the preceding set of movements and transformations of the information is called cycle. Cyphertop has two cycles:

In the first cycle, the movements of bits through the geometrical structures follow the sequence of recombinant algorithms that Cyphertop arms on each occasion according to the content of the initial parameters for that cycle (the first five modifying numbers). Substitution also operates that way (XOR operations and complement). But the bits transposed and transformed (substitution) correspond to the original file, to garbage bits and to the bits of the five initial parameters used to encrypt the first cycle.

In the second cycle as in the first one, the same thing operates, with the difference that the bits to be transposed and substituted belong to the file already encrypted in the first cycle, to the five modifying numbers for that cycle and to garbage bits created during the second cycle.

In each cycle, garbage is added in amounts calculated by Cylinder(). These garbage bits add noise, that is, uncertainty. The distortion generated in the final Cyphertop encryption is high, without the length of the encrypted file increasing significantly.

Each cycle has a header that contains the data (five initial modifying numbers) that modify the structure of Cylinder and other procedures within Cyphertop.

When the data is processed in the second cycle, the first “block” will contain data of the header of the first cycle, plus garbage bits and bits of the encrypted original file. Then a displacement of blocks occurs which makes it impossible for cryptanalysts to perform an analysis of the data by blocks. This is the figure that was intentionally sought when encryption at two cycles was conceptualized. And in addition, the two cycles guarantee that any text obtained when trying to break the ciphering of the second cycle will give as a result a file likewise encrypted (in the first cycle) and never the plain original text.

11. The impossible reading: The three-dimensional bolts.

In the process of decoding, for the correct functioning of each one of the cycles that have just been described, the decoding code of Cyphertop must be instructed on how to proceed.

The initial parameters, (modifying numbers), as it is obvious, must be transmitted within the encrypted data so that the decoding code of Cyphertop arms the inverse procedure to the encryption. These data as it has been stated already, are transmitted in the Header of the encrypted file. Five parameters (Modifying numbers) per cycle.

It is imperative that this information be safe from any attempt to attack by brute force. Such was the requirement of design for storing these types of data within the encryption of Cyphertop. The solution was to generate a three-dimensional data structure, within which, the bits generated randomly are inserted. Later, inside this three-dimensional structure of bits, the figure corresponding to the parameter (Modifying numbers) that we wish to send is “carved”, so to speak. Finally, we go on to stir that bits of this three-dimensional structure and to transform its bits through XOR and complement operations and they are moved to a vector whose bits are inserted one by one in positions of the Header, selected by Cylinder().

For the decoding code of Cyphertop to find again the parameters (modifying numbers), it is necessary to repeat the process inversely and finally, read in a “visual” way the figures corresponding to the parameters (modifying numbers) sent for the encrypting code.

That is why this structure is called a three-dimensional Bolt since its subtle mechanism hides access to information of great importance for the crypto analyst.

12. A computational curiosity.

If a quadrillion computers could be created, each one of them one quadrillion times more powerful than the most powerful one of the present, the Summit computer of the National Oak Ridge Laboratory with a yield of = 200 petaflops = 2×10^{17} flops

(Floating Point Operations per second). The combined power of this quadrillion "improved" Summit computers would be: $10^{24} \times 10^{24} \times 2.7 \times 10^{17} = 2.7 \times 10^{65}$ flops

Now, from now until the end of the universe (according to cosmologists), there are 15,000 million years, that is 4.7304×10^{17} seconds

Then all the computers proposed above, in the time of existence remaining to our universe would have carried out 1.277208×10^{83} flops

Assuming that in each flop a configuration of a Cyphertop encryption could be examined, the computational power already described would not be enough to examine all the possible forms in which one connection point or version of Cyphertop can encrypt a set of data: $2.085924830 \times 10^{93}$

Such is the difficulty to proceed with an attack of brute force against one encryption made with a particular version of Cyphertop. Likewise, an attack of this kind on one encryption done with an unspecific version of a universe of Cyphertop in particular, is discarded. And finally, what can be expected if it is an attack to a Cyphertop encryption done by an unspecific version, belonging to an unspecific universe of the Cyphertop system?

13. Surjective relations, isomorphism, polymorphism, and polysomorphism.

We have already seen that each connection point can encrypt a data set in AC different ways where **AC** = $2.085924830 \times 10^{93}$.

From the point of view of a curious third party, he can see several encryptions, like the fact of belonging likewise to different files. But that is not so. Therefore there is a tautological relation here: Different sets of symbols (coded), originated by the same file.

Furthermore, precisely the fact that a file can be:

- Coded in different ways by the same installed version of Cyphertop: Polymorphism in the result.

- Coded in AC different ways by each version of Cyphertop and that these encodings differ from the other versions of the same universe.

Likewise, different files coded by the same version of Cyphertop or by another version of the same universe, can give an equal encoding as a result, establishing on this occasion an inverse surjective relation: Isomorphism in the result.

Several different files with the same encoding. Although this type of event is not frequent, what is remarkable here is that under the Cyphertop system, it is possible.

Finally, if any encryption corresponds with several encodings of the same file by part of one or more connection points, but at the same time corresponds to encodings of different files by part of one or more connection points, we will have at the same time (in reference to the final result, in other words, to the encryption) the two figures stated before: Polymorphism and isomorphism.

This figure we call "Polyisomorphism" since it shows both figures in one same result: Polymorphism and isomorphism.

14. The magnificent geometry.

Cyphertop is due in great part to the geometrical disposition of its information structures. However, it is mandatory to admit that geometry has always been present in cryptography, ever since man found himself in the need to safeguard information. Look at the following historical relationship:

In ancient times, romans and other peoples sent encrypted files or messages, transposing letters or substituting them, but all the labor was carried out over the file or message, in a linear way. (One dimension, that is, one vector). The encryption of Alberti appears, developed by Giovani Battista Belasso and cryptography advances to two dimensions: The matrix. Methods based on that ciphering were developed to give it more robustness: The Vigénere cipher in France.

In the second half of the XIX century, the Confederated States of America, for example, used a ciphering disk to implement the Alberti ciphering during the Civil War in the United States. But the encryption method by disks had its most famous exponent in the Enigma machine: used by the Germans from the 1920s till the Second World War.

Currently, there are cryptographic methods that use data cubes and other structures in three dimensions. And in fact, there are encryption systems that use more than three dimensions. In the Enigma machine in a certain way, given the necessary rotation of coupled disks we can talk of a third dimension: But a detailed mathematical analysis, allows the simulation of an Enigma machine through a computer program, replacing the disks with circular vectors.

Cyphertop can adopt information structures from 1 to 3 or more dimensions for any of its universes. The current sample universe of Cyphertop contemplates information structures of up to three dimensions: Vectors, Matrices, and Cylinders (Cubes in rotation).

Cyphertop is a cryptographic system in evolution and the geometry of its information structures intends to escalate in complexity. The data movement within this geometry, assisted by the random addressing that Cylinder can provide, is so to speak, the down payment of the complexity of Cyphertop.

15. Randomness: From philosophy to physics.

Any event whose occurrence is totally unpredictable in the conditions of time and place is defined as random. For the purposes of everyday reality it suffices with the following statement:

"Any event whose occurrence is determined by models of complexity greater than the understanding on the part of contemporary human knowledge can be considered as random".

In that order of ideas, when human knowledge advances and the understanding of the physical variables of such behavior (the occurrence of the event) become possible, we change the category of this event: It no longer belongs to the category of random event.

16. The second premise.

At the beginning of this document, the two basic premises for the design of Cyphertop were illustrated:

- The impossibility of an attack by brute force: It was achieved, although, by the general nature of the description that has been made here, it is not formally proven, except for the figures that are exhibited. The final test, code in hand, is beyond the scope of this document and requires a different technical level and conditions of confidentiality that is not the case to be illustrated here.

- Even with the source code in enemy hands, Cyphertop is resistant to all attempts of decryption. Let's look at this point next:

The history of Cryptography is full of stories about how many encryption processes were violated or as it is more common to call it, broken by a persevering enemy or, very clever who resorted not only to the ingenious spark of inverse reasoning but also to buying or stealing the encrypting procedure.

During the Second World War, the Enigma machines obtained as a "loan" from the Germans, plus the previous knowledge already obtained by the Polish intelligence, were decisive so that Alan M. Turing and other English crytographers put the "Bomb" on the verge of completion, an electromechanical device with which the German code was finally broken towards halfway through the war.

In Japan, the transpositions machine of Ito (Purple Ciphering) suffered "an unauthorized loan" that allowed, among other things, to break the Japanese code, intercept the plane in which Admiral Yamamoto was travelling and destroy it.

In this respect, there are many legends about the reliability of ciphering systems sold by private companies, which are susceptible to huge external political and economic pressure.

In the event of an attack on Cyphertop through non-computational ways, it should be noted that it is precisely the thousands of data that the user generates interacting with the computer at the moment of installing a connection point or version of Cyphertop which makes each version possess its own identity (DNA), that is, a different encryption mechanism from other versions of Cyphertop.

Finally, the internal data of each version of Cyphertop are not contained in a common everyday file. They are hidden through an internal cryptographic process using Cyphertop.

To do a decryption attempt on an undetermined version of Cyphertop, it is necessary to make a brute force attack on its internal data which requires a labor of computational testing of

PC = $8.21479634 \times 10^{60.224}$ possible configurations of the internal data (DNA) of that version or connection point as long as the source code of Cyphertop is at hand! Any attack without this resource (the code) is more than discarded.

That is because the challenge would be even greater, that means analyzing the content of two consecutive blocks of ciphering when they are files sufficiently large: That is, an analysis of 40 million bytes, in other words, complexity analysis over: **320 million bits.**

And of course, the impossibility of proceeding with an attack by brute force with code in hand has been illustrated, therefore its meaningless to make computational calculations vs the complexity of 320 million bits.

Assuming that the code is acquired, obtaining the internal data by computational effort, as was demonstrated, would be an almost impossible mission.

But obtaining the data (DNA) of a point specifically of Cyphertop, via “loan”, only violates the compatible points of access of that connection point or version That is, the node in question.

As will be seen later in the following section (17), even so, the security of Cyphertop is reinforced given the mutation capacity that the DNA has in the Cyphertop system.

On the other hand, it is a work of seconds to carry out the transformation of those nodes into a new version of Cyphertop with different internal data (DNA). This places Cyphertop in a prominent place regarding control of emerging damage in what is related to the information that is handled through it, for the situations of illegal appropriation of its internal platform of operation (DNA).

17. Mutation in time. How and when.

It was mentioned before that each connection point of the Cyphertop model encrypts the same file in a different way on each occasion. This is partially true due to the fact that the parameters (modifying numbers), for each encryption allow generating: **AC** = $2.085924830 \times 10^{93}$ different ways of encrypting such file, or to say it in Cyphertop language: Different encryption algorithms.

But they are not infinite. And either way the same $2.085924830 \times 10^{93}$ algorithms for each connection point will continue to be generated, therefore, the promise that the Cyphertop model would be not automorphic in time is in terms of everyday reality.

The time of existence of a human being is not enough to carry out **AC** encryptions of a set of data. Concern also arose regarding the security of the Cyphertop model in front of thefts of codes or what is really worrying, the theft of the very DNA of the connection point.

In this regard, it is good to clarify that protection mechanisms have been designed to adequately safeguard the DNA residing in a Cyphertop connection point.

The intelligence services have as a rule that they keep the same passwords or keys for no more than six months maximum. The world today is composed of new challenges: Hackers, cyberspace bandits, intelligence services and all kinds of intruders who steal the intimacy of the internauts every day.

Payment platforms have been constantly the target of attacks and thefts which show that for now, there is no information protection system completely invulnerable. In virtue of which Cyphertop implements the figure of cyclic regeneration of DNA of each connection point (DNA mutation) adapting itself to this trend.

This event will be designed so that it occurs with a certain frequency. Requesting the users of the connection point to renew their common DNA wich has been used long enough and it’s time for its renewal. This process requires an agreement between users in order to safeguard the encrypted files with the DNA to be replaced. As it is logical, the new DNA will be transmitted by the user generator to the receiver, encrypted with the old DNA already used for that couple of users.

The figure of the regeneration of the DNA (by the decision of the parties), answers affirmatively the question of whether Cyphertop is an automorphic system in time or not.

Therefore, now there can be talk in strict terms of a mutating system in time, not automorphic, which will give the user better protection of his information since the new set of DNA will provide the user with some other $2.085924830 \times 10^{93}$ different algorithms from the ones he had been using. Anyway, would a human being live the necessary time to send during his lifetime $2.085924830 \times 10^{93}$ encryptions of the same file? And thus exhaust all the possible algorithms that Cyphertop can provide him to encrypt the same file, with a determined DNA?

18. The architecture of a Cyphertop network.

A universe of Cyphertop is constituted by a swarm of networks of connection points.

Thus, it is possible to have:

- Dedicated connections point to point that serve to connect exclusively two points on the network.
- Star topology with central point communicating with the satellites, but without a connection between them.
- Star Topology but with communication between all the points in the star.

19. Conclusions:

Cryptography has evolved from the transposition and substitution of symbols to the handling of the minimum unit of information:

The bit.

When we speak or we listen, we do so in a vectorial way. Television gives us information in a two-dimensional way, with three-dimension projections. With the new holographic techniques, the capture of information in a three-dimensional way becomes a reality.

The future of cryptography belongs to the multidimensional geometry and to the complexity of the generation of pseudo random numbers. The author considers that the escalation in dimensions with contributions of new geometries for the transposition of bits is the way of the future.

Likewise, new and sophisticated techniques for capturing events of real life may be used through the I/O mechanisms of the computer or of the cell and also mechanisms related with the internal functioning of its processors, providing levels of randomness, that in its time exceed by a wide security margin, the frontier of the random with the predictable, which will result in an advance in the generators of pseudo random numbers in such a way that its results can well be cataloged as random, due to their high complexity at the moment of their generation and use.

Cyphertop opens up a new approach for high security cryptography. The paradigm of the length of the key is no more. Neither is the massive calculations to break prime numbers that are larger every time.

There are no limits other than those imposed by human knowledge. And of course, there is your imagination and creativity. May this be the occasion to quote an illustrious North American writer and poet, Edgar Allan Poe:

“It is doubtful that the human race will create an enigma that the same human genius will not solve”.

Therefore,

The challenge of Cyphertop is open...