International Journal on Data Science and Technology
2016; 2(4): 46-50
http://www.sciencepublishinggroup.com/j/ijdst
doi: 10.11648/j.ijdst.20160204.12
ISSN: 2472-2200 (Print); ISSN: 2472-2235 (Online)

Methodology Article

A Highly Secured Mathematical Model for Data Encryption Using Fingerprint Data

Arindam Kumar Paul, Liton Devnath, Md. Rafiqul Islam

Mathematics Discipline, Khulna University, Khulna, Bangladesh

Email address:

arindam017@gmail.com (A. K. Paul), litonmathku09@gmail.com (L. Devnath), mrislam_66@yahoo.com (Md. R. Islam)

To cite this article:

Arindam Kumar Paul, Liton Devnath, Md. Rafiqul Islam. A Highly Secured Mathematical Model for Data Encryption Using Fingerprint Data. *International Journal on Data Science and Technology.* Vol. 2, No. 4, 2016, pp. 46-50. doi: 10.11648/j.ijdst.20160204.12

Received: June 2, 2016; **Accepted**: July 9, 2016; **Published**: August 16, 2016

Abstract: In this research we tried to find out a new, unique and efficient way for converting plain-text to numbers or other values which can be used both in Symmetric and Public Key Cryptography, Stenography and others related with data security. It is just a developed and highly secured model for just converting plain-text to cipher text. We proposed a new data encryption and decryption method here which can help anyone for encrypting and decrypting data more securely.

Keywords: Cryptography**,** Decryption, Hybrid Encryption Method, Image Processing, Plain-Text to the Cipher Text

1. Introduction

Here we have developed a highly secured model for data encryption. That is increasing security for both symmetric and asymmetric algorithms for converting the plain-text to the cipher text. But, our aim is to do this work more securely and without any doubt that a hacker can easily decrypt the cipher text to the original text. For these purposes, we have proposed a new model for doing this work perfectly and securely. We used fingerprint data for modeling the whole process.

We know that fingerprint data [1] is unique for every individual person. So, if we encrypt plain text using the fingerprint data, it should be more secure encryption.

2. Literature Survey

Cryptography has been used for data encryption, decryption and secure communication from very ancient ages. Before the modern era, cryptography was concerned solely with message confidentiality conversion of messages from a comprehensible form into an incomprehensible one and back again at the other end, rendering it unreadable by interceptors or eavesdroppers without secret knowledge (namely the key needed for decryption of that message). Encryption attempted to ensure secrecy in communications, such as those of spies, military leaders, and diplomats. In recent decades, the field has expanded beyond confidentiality concerns to include techniques for message integrity checking, sender/receiver identity authentication, digital signatures, and interactive proofs and secure computation, among others.

Cryptanalysis of the new mechanical devices proved to be both difficult and laborious. In the United Kingdom, cryptanalytic efforts at Bletchley Park during WWII spurred the development of more efficient means for carrying out repetitious tasks. This culminated in the development of the Colossus, the world's first fully electronic, digital, programmable computer, which assisted in the decryption of ciphers generated by the German Army's Lorenz SZ40/42 machine.

Just as the development of digital computers and electronics helped in cryptanalysis, it made possible much more complex ciphers. Furthermore, computers allowed for the encryption of any kind of data representable in any binary format, unlike classical ciphers which only encrypted written language texts; this was new and significant. Computer use has thus supplanted linguistic cryptography, both for cipher design and cryptanalysis. Many computer ciphers can be characterized by their operation on binary bit sequences, unlike classical and mechanical schemes, which generally manipulate traditional characters directly. However, computers have also assisted cryptanalysis, which has compensated to some extent for increased cipher complexity. Nonetheless, good modern

ciphers have stayed ahead of cryptanalysis; it is typically the case that use of a quality cipher is very efficient (i.e., fast and requiring few resources, such as memory or CPU capability), while breaking it requires an effort many orders of magnitude larger, and vastly larger than that required for any classical cipher, making cryptanalysis so inefficient and impractical as to be effectively impossible.

Extensive open academic research into cryptography is relatively recent; it began only in the mid-1970s. In recent times, IBM personnel designed the algorithm that became the Federal Data Encryption Standard; Whitfield Diffie and Martin Hellman published their key agreement algorithm; and the RSA algorithm was published in Martin Gardner's Scientific American column. Since then, cryptography has become a widely used tool in communications, computer networks, and computer security generally. Some modern cryptographic techniques can only keep their keys secret if certain mathematical problems are intractable, such as the integer factorization or the discrete logarithm problems, so there are deep connections with abstract mathematics. There are very few cryptosystems that are proven to be unconditionally secure. The one-time pad is one. There are a few important ones that are proven secure under certain unproven assumptions. For example, the infeasibility of factoring extremely large integers is the basis for believing that RSA is secure, and some other systems, but even there, the proof is usually lost due to practical considerations. There are systems similar to RSA, such as one by Michael O. Rabin that is provably secure provided factoring $n = pq$ is impossible, but the more practical system RSA has never been proved secure in this sense. The discrete logarithm problem is the basis for believing some other cryptosystems are secure, and again, there are related, less practical systems that are provably secure relative to the discrete log problem.

As well as being aware of cryptographic history, cryptographic algorithm and system designers must also sensibly consider probable future developments while working on their designs. For instance, continuous improvements in computer processing power have increased the scope of brute-force attacks, so when specifying key lengths, the required key lengths are similarly advancing. The potential effects of quantum computing are already being considered by some cryptographic system designers; the announced imminence of small implementations of these machines may be making the need for this preemptive caution rather more than merely speculative.

But now-a-days, Cryptography has developed for very secure communication by inventing various methods and technologies. For security purposes, various models and algorithms have been developed by us. Now, Cryptography can be differentiated into two main techniques. They are:

- Secret Key Cryptography: It uses a secret key for both encryption and decryption.
- Public Key Cryptography: It uses two different key known as public key and private key for encryption and decryption process. Public key is known to all and has a mathematical relationship with the receiver's private key.

The second one is considered as more secure and reliable technique [2], [3], [4].

3. Working Process

At first, let me explain what kind of information can I get from a fingerprint image and what is the output.. So, if we encrypt plain text using the fingerprint data, it should be more secure encryption [7], [8]. This mathematical model can be implemented very perfectly by developing a system using image processing. By analyzing a fingerprint image of any person and using the obtained data for encryption and communication will be a very secure way for this purposes. In our work, we have built a mathematical model by using the pixels determination method of a fingerprint image.

I. Here, we want to use a quite different method for our encryption key generation. We have considered a fingerprint image of any dimensions, let 320*256 is the base dimension. This image is receiver's sharable key. By converting the image to a different dimension we can get as many keys as we wish for encryption. But, since the image is same but of different dimensions, so there is always a common ratio between the receivers's shared key. When receiver's key is declared by the receiver, sender will use that key means black pixels number of that image for encrypting data.
II. Here, we only know the black pixels number of a fingerprint and then we extended the black pixels number by a certain function.
III. The function type depends on user's requirement. User can declare such a function which may output a very big number or any kind of number as you wish. But someone must declare such a function that is hard enough to inverse. The properties of this function are explained bellow.
IV. Since, our aim is to represent each alphabet by 2 or 3 or 4 or 5 numbers or as you can. So, if you represent an alphabet by 3 digits of the numbers, user must extends the black pixels number with such a function that the extended number must be divisible by 26. Because, there are 26 alphabets in English language. User can take this number according to your requirements.
V. Here are 2 ways to define every alphabet by a certain number. User can only once generate a number of 130 digits if user wants to define every alphabet by a fixed number. As an example, If user take a 130 digit number, Then you can define last or first 5 integers of that number as "A" or "Z". And then it'll follow the sequence for finding the corresponding integers for every alphabet.
VI. User can generate a very big number instead of 130 digits, because if user will use a fixed number for each alphabet then each alphabet can be recognized easily by any hacker.
VII. So, instead of using a fixed number for each alphabet, user can use different integers for representing every

alphabet. If an alphabet repeats more than one time, then this method is very efficient and secured.

VIII. For making it more secure, user can generate a very large number with any function, that number must be divisible by 26 as in English languages has 26 letters excluding uppercase.

IX. Then, by the first 3 or 5 integers of that number, as you wish, we can represent the first alphabet "a", and by the second 3 integers of that number we can represent the alphabet" and we can proceed up to the alphabet "z".

X. After finding "z", we'll take the next 3 integers of that number for representing the second" a" included in the plain text. And then we'll repeat this operation since the whole plain text is converted into cipher text.

XI. But, According to our model, we know that there are total 26 alphabets in English literature. So, we must take some number always which number is divided by the number 26.

XII. We must not send our cipher text directly because, if any hacker can see the cipher text, there is always a probability of finding out the original text though our encryption method is secure enough. After getting the number representation of our message, we'll again use a function; it may be any function and may be same as the function we defined above. But if we use different function here, it'll be more secure. This function will again extend our cipher text into a very big number. And we'll send that big number in exponential form. The receiver must know the function. Receiver will get the original cipher text means our number representation of plain text by inversing the function. Anyone should use such a function that is hard to inverse.

XIII. After getting the message from the sender, at first the Receiver will find the original cipher text. Then, he can find the original plain text simply by taking the modulo of that number. If anyone uses 1 number for each alphabet, then receiver will take modulo 10 of that number. If 2 digits represent an alphabet, then receiver will find the encrypted message by taking modulo 100 and so on for more digits. Receiver can use a program for this work. A very simple program can do all the decryption process very easy. But, extracting plain text from sender's number will take time a bit long. Because that program will at first find the cipher text and then find the original message. But, here we use very long text, so that the program will take much more time for calculating the original result means original plain text.

You can choose the first alphabet representing number from any digit you wish. Then you can proceed by defining any other alphabets. You can also start defining alphabet from the end of that number. For a big number, it is simply hard to predict it.

Anyone can use 2-10 digits for representing a single alphabet. Longer digit means more secured. A very long digit for every alphabet increases the length of the root number. [10], [11].

Since, we are using he black pixels numbers of the fingerprint image, so the key will be different for each communication. We have only one image for getting the black pixels quantity. But we can resize that image as many times as we wish. So, for every communication, we'll use different key as our encryption key. No two key will be same or similar. And any 2 key will not contains the same digits because of having another dimension.

We can use this process for a hybrid encryption designing and for many more purposes [5], [6].

4. Properties of the Function and Key

The used function for encrypting data can be shared for the senders. We can take a very larger dimension's image for obtaining a very big number of black pixels. Or user can define any function which is difficult to inverse. Or user can send the function into the cipher text anyhow. User should be more creative for selecting the function. The function, key and how many digits represents an alphabet must be known by both user. Bot user also know from what digit to start means permutation type, how many alphabets or symbol are defined and the others additional if used.

5. Mathematical Model

I. Suppose Our Key is N.

II. M Is any number that is a multiple of 26 or the number of alphabets $\&N$.

III. M Must be very big or it may be of a certain length according to users demand.

IV. Define the all alphabets a-z, or A-Z, or both by taking as many digits you want to use for a single alphabet. As an example, for $A = 335, B = 645, c = 345$ etc.

V. If you consider only lowercase or uppercase, you the length of M must be of 26 digits for representing each alphabet by only one digit of that number.

VI. If you consider both the lowercase and uppercase, then the length of M must be a number containing 52 digits for representing each alphabet by only one digit from the M.

VII. If any message or text contains a single alphabet more than one time, then it would be better to choose different digits for each repetition of that alphabet.

VIII. For this, produce a very big number M, find the total number of digits by calculating:

- The number of digits you want to use for representing a single alphabet let D.
- The number of repetition of an alphabet which repeats more than any other alphabet let R.
- The total number of alphabet, for English language it may be 26 or 52. Let define it by A.
- Then we can get the number of digits of M, we are denoting it by NOD, by the following calculation: $NOD = A \times R \times D$
- Let us consider A=52, D=3, R=5 then we must produce a number which contains at least $\mathbf{52 \times 3 \times 5 = 780}$ digits.

Then, you can choice first 3 digits for representing the alphabet "a", Second 3 digits for representing "b" and so on. You have finished defining each 52 (including lowercase and uppercase) alphabets by total 156 digits. But, as your highest repeating alphabet repeats 5 time, So, you can consider the next 3 digits after 156th digit as new value of a, means the value of second "a" used in your message will be represented by the next 3 digits after 152th digit. You can also represent "a" by from any cycles which contain 52 digits. Otherwise you can inverse the whole number for selecting alphabet's value or you can use any other way you think good for you.

The Whole Procedure can be easy for all by this figure:

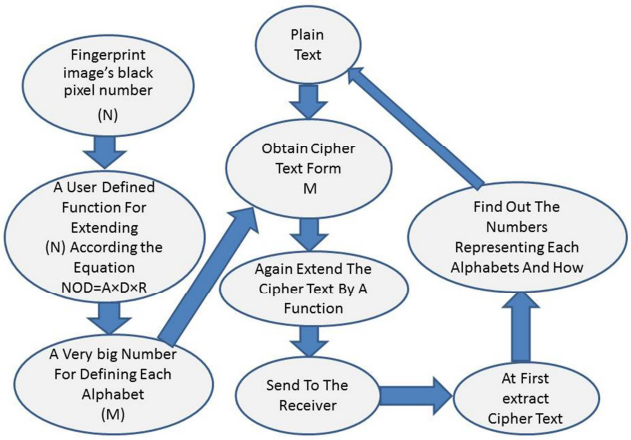

Figure 1. *The Encryption and Decryption Process.*

6. Strengths and Weaknesses

Generally speaking everyone feel the minimum key size for a block cipher should be 128 bits. The minimum for the block size depends on the precise application but in many applications (for example construction of MAC functions) a 128-bit block size should now be considered the minimum in many applications. But for maximizing the security one should use a standard key for his work purpose. It may be very big, medium or small. So here we used a very big cipher and block size for ensuring the maximum security. This Model is secure enough to protect almost all kinds of attacks. Because, each individual can uses this model by changing various properties, adding a new part and subtracting any prone or more part. A User can customize this model according to his work purpose. So, it is almost impossible to use or guess another's algorithms. [7], [8], [9].

Our Proposed model may be a little bit hard to understand to someone. Encryption and decryption process is too large and hard to implement. Working with very big number for encryption can be hard for user.

7. Future Work

This is just a mathematical model of data encryption. Nothing else. But, it can be used at all types of symmetric and public key algorithms. In future, we'll try to develop a mathematical model of both symmetric and public key algorithm using this encryption method. This model can be used to develop other kinds of data security models. Many Cryptographic and Stenographic model can be developed from this model and methods. It can help any Cryptographic algorithm for developing security. Our model can be used as a symmetric key cryptography algorithm too.

8. Conclusions

Cryptography plays a vital role now-a-days in all sectors related with data security. For the better security, the design of any algorithm must be complex and enough secured. But the security concern is not same everywhere. It varies person to person, field to field. We have just designed a mathematical model of a Cryptography algorithm which can be used in various ways by various persons in various fields based on their requirement of security.

References

[1] L. Devnath, A. K Paul, M. R Islam, (2016). "A Study on Binary Number of Gender Identification Based on Fingerprints", "International Journal of Scientific & Engineering Research, Volume 7, Issue 2, Pages 338-342.

[2] Liddell, H. George, Scott, Robert, Jones, H. Stuart; McKenzie, Roderick. (1984). A Greek-English Lexicon. Oxford University Press.

[3] Bellare, Mihir, Rogaway, Phillip. (2005). "Introduction". Introduction to Modern Cryptography Page 10, https://en.wikipedia.org/wiki/Cryptography.

[4] P. Kuppuswamy, Saeed Q. Y. Al-Khalidi. (2014). "Hybrid Encryption/Decryption Technique Using New Public Key and Symmetric Key Algorithm", National Chengchi University & Airiti Press Inc, Vol. 19, No. 2, Pages 1-13.

[5] B. Adida, M. Bond, J. Clulow, A. Lin, Ross J. Anderson, and Ronald L. Rivest. (2007). On the security of the EMV secure messaging API (extended abstract). In Bruce Christianson, Bruno Crispo, James A. Malcolm, and Michael Roe, editors, Security Protocols Workshop, volume 5964 of Lecture Notes in Computer Science, Pages 147–149. Springer.

[6] A. Agrawal, G. Patankar. (2016). "Design of Hybrid Cryptography Algorithm for Secure Communication", International Research Journal of Engineering and Technology, Volume: 03 Issue: 01.

[7] D. Harinath, M V Ramana Murthy, B Chitra. (2015). "Cryptographic Methods and Performance Analysis of Data Encryption Algorithms in Network Security", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 5, Issue 7.

[8] S. Bhati, A. Bhati, S. K. Sharma. (2012). "A New Approach towards Encryption Schemes: Byte – Rotation Encryption Algorithm", Proceedings of the World Congress on Engineering and Computer Science, Vol II .

[9] D. Pointcheval, (2002). "Asymmetric Cryptography and Practical Security", Journal of Telecommunications and Information Technology. Volume 4, Pages 41-56.

[10] J. M. Pollard. (July 1978). "Monte Carlo Methods for Index Computation (mod p)", Mathematics of Computation, 32 (143), Pages 918-924.