PISANC Encryption Algorithm

Lazuardi Firdaus
School of *Electrical Engineering and Informatics*
Bandung Institute of Technology
Bandung, Indonesia
lazuardifirdaus369@yahoo.com

Richard Matthew
School of *Electrical Engineering and Informatics*
Bandung Institute of Technology
Bandung, Indonesia
richard.matthew20@gmail.com

***Abstract*—PISANC is an encryption algorithm designed based on the concept of confusion, diffusion, and feistel network. PISANC implements what is named as NRZI encodings and triple feistel using inspiration from NRZI on physical layer of OSI and regular feistel network.**

Keywords—encryption, confusion, diffusion, feistel

I. Introduction

Eventhough cryptography may seems like an otherworldly science for the general masses, we actually depend our daily lives on the works of cryptography more than we think. Cryptography protects our sensitive information that may be stored in databases so that strangers won't misuse it. Cryptography also hides our messages across the internet so that we may communicate with our relatives easily without fear of others listening.

In spite of its importance in our daily lives, the world of cryptography is an ever-changing one. What seems like an unbreakable method of encryption today may be found unusable tomorrow. An example of this is DES which is an encryption algorithm deemed safe enough to be the standard until 1998. At that time a company called Electronic Frontier Foundation had managed to build a DES cracking machine that can do its job within a few days[1]. Since then, DES are deemed unsafe.

Research on how people can crack existing encryptions are always ongoing. This way hopefully we can determine if they are no longer safe to use before any exploit can be made. New encryption methods are also appearing to replace the old ones that are deemed unusable.

In participation of this, this paper proposes the concept of a new encryption method called PISANC (which recursively stands for: Pisanc IS A new eNCryption). PISANC is an block cipher encryption that seeks the characteristics of a good encryption namely confusion and diffusion. PISANC is inspired from the techniques used in informatics, specifically NRZI encoding and the extension of feistel network called triple feistel.

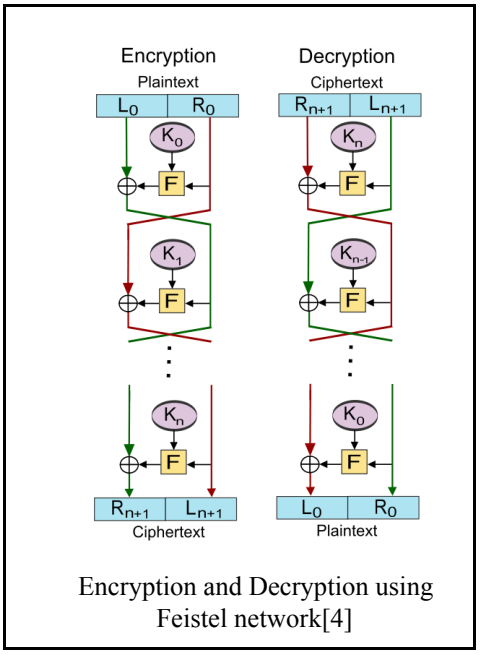
II. Basic Theory

A. Characteristics of a good encryption

A good encryption won't be easily cracked by cryptanalysts trying to extract the plaintext from the ciphertext, instead it will frustrate them. This can be achieved using the properties of confusion and diffusion[2]. Confusion means that the statistical correlation between ciphertext and plaintext are hard to find. This can be achieved by using an algorithm that is complex and includes some kind of substitution.

Diffusion means that the statistical structure of plaintext are "dissipated" into a large part of ciphertext. An encryption with a good diffusion has a property that a slight change in plaintext will affect many parts of the ciphertext. Cryptanalysts must consider the whole ciphertext to deduce the content of even a very small part of plaintext.

B. Feistel Network



Encryption and Decryption using Feistel network[4]

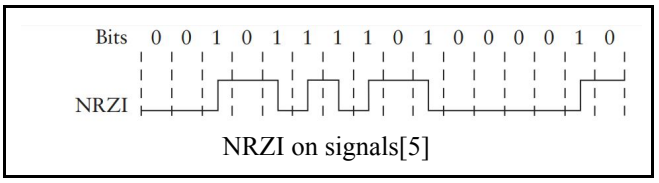
Feistel network is a technique invented by Horst Feistel for Lucifer encryption algorithm[3]. Feistel network can be found in many well-known block encryption algorithms which includes DES, LOKI, GOST, FEAL, Lucifer, and Blowfish.

Feistel works by first separating the plaintext into two blocks. Both of these blocks will then be repeatedly go in a function (usually called F-function) using a transforming key, and undergoes "XOR" operation.

C. NRZI

NRZI is an encoding protocol commonly used in the physical layer of OSI network. The idea behind NRZI network is to pass bit information by "whether there is a change" of the signal instead of just the "value" of the signal. A 1 bit is passed by changing the signal from high to

low (or vice versa), and 0 bit is passed by not changing the signal (high remains high and vice versa).



NRZI on signals[5]

III. PISANC Block Cipher

A. Basic Idea

The idea behind PISANC is to seek the characteristic of a good encryption which is confusion, diffusion, and feistel network. These characteristics are met using the idea of using NRZI encoding commonly found in the physical layer of OSI layers and the extension of feistel layer called triple feistel. These 2 basic ideas combined can already fulfil the characteristics of confusion.

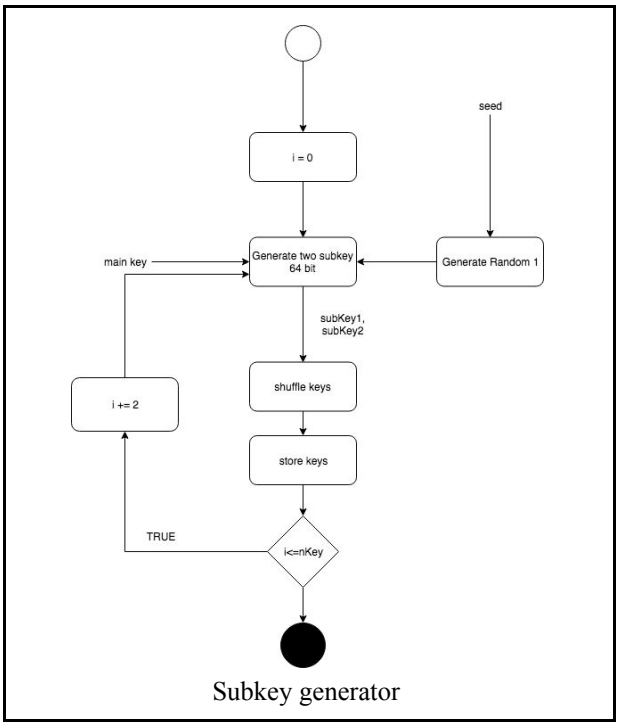
PISANC encryption as a whole work by first encrypting the plaintext using the NRZI encoding and then further encrypting the result with 5 rounds of triple feistel. PISANC encryption use 192 bits main key and a seed for random generator with a size ranging from 1 bit to 1 GB.

B. Subkey & Random Number Generator

For both the NRZI encoding and triple feistel to work they need to be supplied with a variety of keys which is derived from PISANC main key. NRZI encoding needs one key. Triple feistel needs keys as much as two times of the amout of rounds it will take. For this occasion, the amount of key needed for triple feistel is 10.

The subkey generator will generate 10 subkeys. First, it takes input of the main key and random number from seed. The random number range from seed length x 10 to seed length x 1000. Second, it generate two subkeys from main key with index equal to random number modulo by main key length. Third, those two subkeys are shuffled and then stored. The process is repeated 5 times.

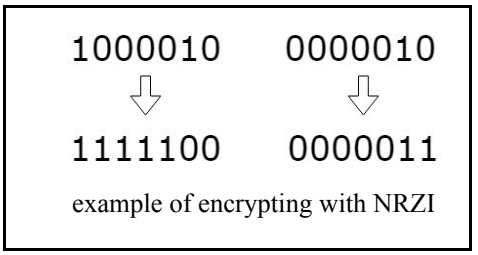
The random number generator will generate 10 random number ranging from 30 to 100. For this implementation the random number will be stored in an array called nArray.



Subkey generator

C. *NRZI encoding*

NRZI techniques can be used to encrypt plaintext to ciphertext the same way it encodes binary to the high low signal. The first bit of plaintext translates to the first bit of ciphertext. The following bits however translates this way: if the following plaintext bit is 0 the following ciphertext bit follows the bit before it, if the plaintext bit is 1 the ciphertext bit is the negation of the bit before it.

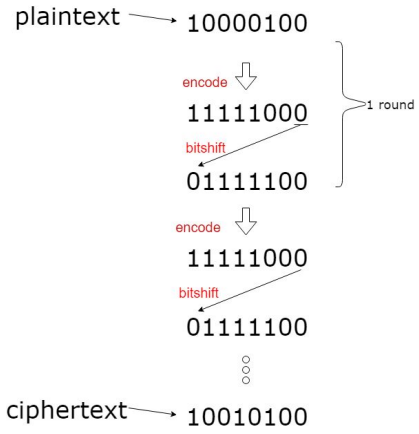


example of encrypting with NRZI

Now, NRZI encoding alone only receives 1 input which is the plaintext bit without any key. To make it "accept" a key as input, we can define the key as "how many rounds of encoding will be done". NRZI encoding with a key of 25 will means that the plaintext bits will be encoded 25 times.

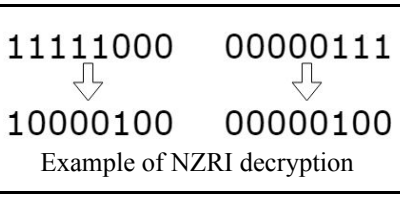
The use of NRZI encoding is to achieve as high diffusion as possible. Using NRZI encoding, changes in first bit will change the whole block, Changes in second bit will change the whole block except the first bit, and so on.

To distribute the diffusion property uniformly to the whole block, the last bit will be pushed to the front of the block each round. This way, the bit with the least diffusion in a round will have the most diffusion on the next round.



NRZI encoding complete with bitshift and rounds

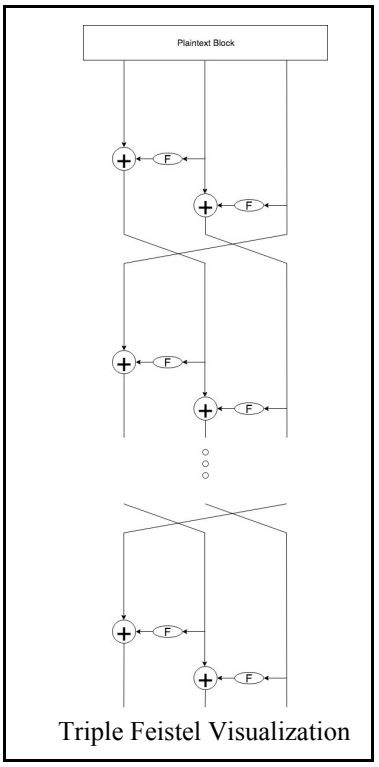
Decryption on NRZI encoding are as straightforward as reversing the rounds. The first bit is pushed to the back of the block and the NRZI translation is undone. The translation can be undone as follows: the first bit of ciphertext is translated as it is to the plaintext. For each of the following ciphertext bit, if the ciphertext bit is the same as the previous ciphertext bit the plaintext bit translates as 0, if the ciphertext bits are different the plaintext translates as 1.



Example of NZRI decryption

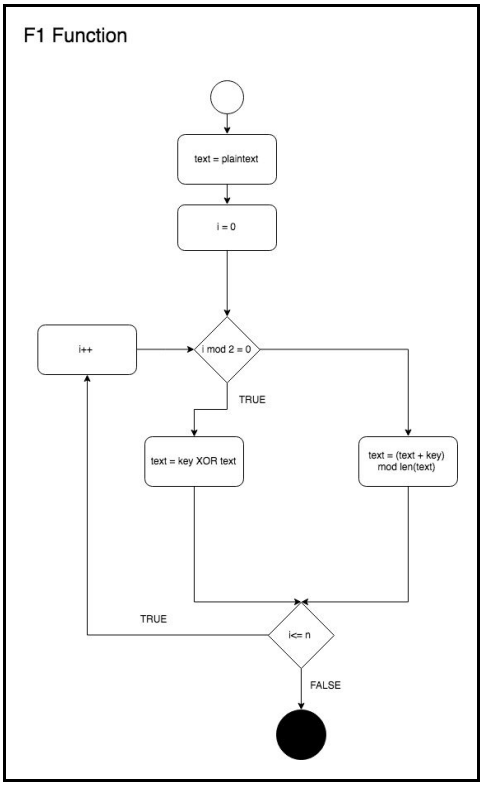
D. Triple Feistel

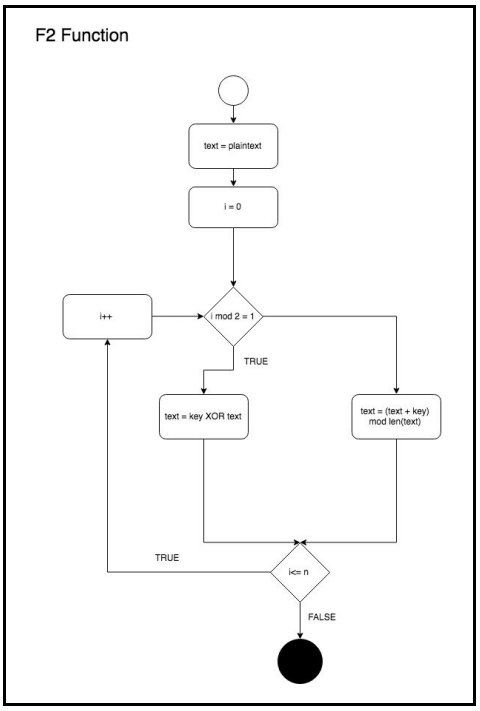
After encryption using the NRZI encoding, the resulting bits will then be encrypted using the extension of feistel network named triple feistel. Triple feistel works the same way with regular feistel network except that the input bits will be divided into 3 blocks.Because of this division into 3 blocks, triple feistel must receive 3n amount of bits for example 96, 192, and 288.



Triple Feistel Visualization

Triple feistel also uses 2 F-functions instead of one. The First F-function will first do xor then bitwise add for n times. The n is taken from nArray. The second F-function is the opposite of First F-function





IV. SIMULATION

An implementation of PISANC using 3 methods has been made. Those 3 methods are EBC, CBC, and Counter mode.

Below is an example of a plaintext that has been encrypted using the 3 modes and the resulting decryption.

Plaintext : PISANC (which recursively stands for: Pisanc IS A new eNCryption). PISANC is an block cipher encryption that seeks the characteristics of a good encryption namely confusion and diffusion. PISANC is inspired from the techniques used in informatics, specifically NRZI encoding and the extension of feistel network called triple feistel.
Ciphertext (ECB): îÕÎWèÆÐÊEÈ¶EL«ØA¡Ì4aeØÑ¼ óÕW;QqX§Aã*¬iYS-+çe[¯S°¿þ£b#÷iG=Í)áªy8.JÑò}²:ð'ª<½ùÓ)Q0u5K ß]xðG_L8ú>ÂT³é¼×2\¯àhf¬á?zeQûv)4ösßI çQçñ~^¡ý°s[_³*ñTv¦Q±V¶ºÈ+Yëf8ÃêXR«P E¿{ëèÆÜÜïóY=e5ËW+gZmei=JìBKl{Þë++ 1 Õa¼³5§äéÇÂ«§Ç=ç" =ËÁ{·pE¨>F ÍHê~åAÛ
Ciphertext(CBC): oR÷Dö½iñÈÐ'5=ÊlÌM¦Ù<±%/½-±t²I3êòJ>× ÕÏáÊd;è¿,*±+â¸÷T°6@GY¨×õDßTÅ^òüôª}Î³ ò Ó<JªÜÄÂ©5Ø3ø*@"vdlÅ°X¯&Lç©Úöû°½ fûÕx É)Jy¦1+²¼¡+Ö¿Z¡©PµÍNÇEó(

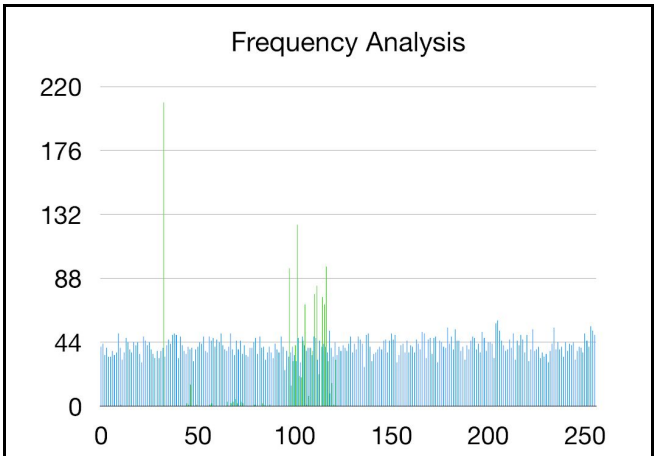
ììÒf= DGho&WBZû#§Èw({jn¦¾aH§É7ÅIÅG=û ·Gò ìI¿+Û
Ciphertext(Counter): °î4á@¦zÚ³?5k®g'C®4Ê8v>âóJ·°oã(çýègùÄ÷ ñ_*¸éÇ# ñÁßÐplB²ñÉÝ2ú:¶p iÑF&8_\QM^ñ©)!ÂXà ,/©X¾$u?²gAÎ(ËìÉKÀ iù®ÊµMÛ¾aYßãÌÌâ·Æ:â>¢ëu;ÊRWI©ÿ Í øÊ(mEÊØøÔÄ_(&ünÌøñhÓ"Ö_¨Üà×óÎ&-Àx bânëKBG.Zß?®4Á8jÑ§}ÿ /e%Dô¦ðÆ,æ¬¤:d_®õõc³ÞôÖ?µô÷*u_<Ò êqG!qGr=!i£,;8cïÊÕ#ÄÅãNB»w
Decrypted ciphertext: PISANC (which recursively stands for: Pisanc IS A new eNCryption). PISANC is an block cipher encryption that seeks the characteristics of a good encryption namely confusion and diffusion. PISANC is inspired from the techniques used in informatics, specifically NRZI encoding and the extension of feistel network called triple feistel.

Through the implementation and testing of PISANC encryption, the execution time of both the encryption and decryption using the PISANC encryption was had as follows.

Method	**Size**	**Encrypt Time**	**Decrypt Time**
EBC	100 bytes	0.1071 s	0.0902 s
CBC	100 bytes	0.0977 s	0.0886 s
Counter	100 bytes	0.0846 s	0.0977 s
EBC	1318 bytes	1.0424 s	0.9504 s
CBC	1318 bytes	0.8776 s	0.9573 s
Counter	1318 bytes	1.0004 s	1.0055 s
EBC	10.544 bytes	7.6389 s	8.2056 s

CBC	10.544 bytes	8.0687 s	7.9579 s
Counter	10.544 bytes	8.5059 s	8.5865 s

A frequency analysis on the plaintext and ciphertext of the 10.544 bytes PISANC encryption have also be done using the EBC mode.



Frequency analysis on the plaintext (green) and ciphertext (blue) of an encryption using PISANC

V. Security

Using key size of 192 bits and seed size range of 1 bit to 1GB there can be $2^{192 + 8e+9}$ combinations. Potential attackers need to use two keys size of 64 bits and 2 random number from 30 to 100 in each round. In 5 rounds there can be $(2^{(2x64)}x70x70)^5$ combinations.

Based on the calculation above we can conclude that cracking the encryption using bruteforce alone will take a very long time.

VI. Conclusion

PISANC encryption is a brand new encryption that takes its inspiration from NRZI and feistel network. Based on the simulation that has been carried out, PISANC has been proven to be quite secure.

References

[1] Electronic Frontier Foundation, '*Cracking DES – Secrets of Encryption Research, Wiretap Politics & Chip Design*', 1998 Published. [Online]. Available: http://cryptome.org/jya/cracking-des/cracking-des.htm. [Accessed: 13- March- 2019]

[2] S. Claude, "A Mathematical Theory of Cryptography", Alcatel-Lucent: Paris, 1945, pp. 92.

[3] F. Horst, "Cryptography and Computer Privacy", Scientific American:New York, 1973

[4] https://upload.wikimedia.org/wikipedia/commons/thumb/f/fa/Feistel_cipher_diagram_en.svg/511px-Feistel_cipher_diagram_en.svg.png

[5] L. P. Larry, S. D. Bruce, "Computer Networks: A System Approach 3rd Edition", Morgan Kaufmann: Amsterdam, 2003