SPKP (Split Plaintext Key Pair) Algorithm – A Novel Method for Symmetric Encryption

Renjith PR
MTech (Information System Security) Student
IGNOU India

Anita John
Department of Computer Science
Rajagiri School of Engineering & Technology
Cochin, Kerala, India

Praseeda K Gopinadhan
MTech (Information System Security) Student
IGNOU India

ABSTRACT
Cryptography, defined as the science and study of secret writing, concerns the ways in which communications and data can be encoded to prevent disclosure of their contents through eavesdropping or message interception, using codes, ciphers and other methods, so that only intended people can see the real message. The design and implementation of a new symmetric key algorithm SPKP (Split Plaintext Key Pair)[1] is proposed . The algorithm encrypts the plaintext file by using the password of the file as the key. The plaintext and key are split in equal numbers and shift cipher is applied to each block of the plaintext. This new algorithm can be considered as a hybrid approach to its precursors.

General Terms
Information security, cryptography

Keywords
Symmetric cryptosystem, information security, split-plaintext-key pair

1. INTRODUCTION
The ideology of cryptography was introduced earlier, where confidential messages were sent as cipher text [2]. Cryptography is still growing and research is still alive for new cryptography algorithms. There are several cryptographic algorithms that can be as simple as shift cipher or as complex as DES etc. As per Kerckhoffs's Desiderata of Cryptography [3], the security of the system should depend only on the secrecy of the keys and not on the secrecy of the encryption or decryption algorithm [4]. In almost all the cryptographic algorithms, either the plain text is encrypted using key or it is divided into small chunks and applies the same key. In this paper, we propose a novel symmetric cryptographic algorithm that splits the plaintext as well as the key in equal numbers and applies the split key on the corresponding split plaintext by using shift cipher. Here the number of splits is determined by a random number generator algorithm. The input to the crypto algorithm is a text file (the plaintext) and its password (the key).

1.1. Cryptographic goals of the algorithm
The major objectives of cryptography are

- Confidentiality which deals with the secrecy of the message
- Integrity which deals with the correctness of the content message
- Authentication which deals with the identity of the sender
- Non-repudiation deals with non-denial of sending of the message

This algorithm preserves the basic three properties of cryptography namely confidentiality, integrity and authentication.

1.1.1 Confidentiality
The plaintext is encrypted by a symmetric key (i.e. the password of the file). The key is kept secret and no one is able to decrypt the cipher text by using another key. This serves confidentiality.

1.1.2 Integrity
The encrypted file can be decrypted only by the same key to enforce integrity.

1.1.3 Authentication
Symmetric key makes the sender and receiver share a common key. The authenticated parties who know the key will only be able to decrypt and see the contents of the message [5].

2. SPLIT PLAINTEXT KEY PAIR (Pi,Ki) ALGORITHM

The algorithm uses the password to encrypt the file with a unique number that creates the unique encrypted text file. The same password is use to decrypt the file thus enabling maximum security of the file.

The plaintext for the algorithm is a file that contains some text information and the key is the password of the file. The algorithm computes a random number from the password to generate the key. The number of letters and ASCII value of key determines the number of splits (say 'n') the key and the plaintext are splitted. The key and plaintext are split equally. Let the plaintext P be split into p1,p2, p3…pn & let the key K be split intok1,k2,k3…kn. The pairs (p1,k1), (p2,k2), (p3,k3)…(pn,kn) undergoes encryption. For a pair (pi,ki) called as split plaintext key pair, pi is encrypted by the key ki.

The algorithm uses shift cipher[6] is used to create the cipher text. The shift cipher is applied on each split plaintext key pair to get the corresponding cipher text. The split cipher is applied in a alternative way to reduce the chances of frequency analysis [7] attacks. These cipher texts are combined to get the final cipher text. As per Kerchoff's principle, the security of the system should depend only on the secrecy of the keys & private randomizer and not on the secrecy of the encryption/decryption transformations. These cipher texts are combined to get the final cipher text. Figure 1 shows the flowchart of the encryption process.

2.1. Algorithm

The encryption is done through the following steps

Step 1 : Start.

Step 2 : Accept file (as plaintext P) and password (as key K).

Step 3 : Generate unique random number 'r' from the password, which serves as the key K.

Step 4 : Split the plaintext P and the key K into 'n' splits.

Step 5 : Encrypt the first split of the plaintext (p1) with the first split of the key(k1) by shift cipher to get corresponding split cipher text(c1)

Step 6 : Repeat step 6 for all 'n' splits

Step 7 : Combine the splits to get the cipher text C

Step 8 : Stop

2.2 Flowchart – encryption

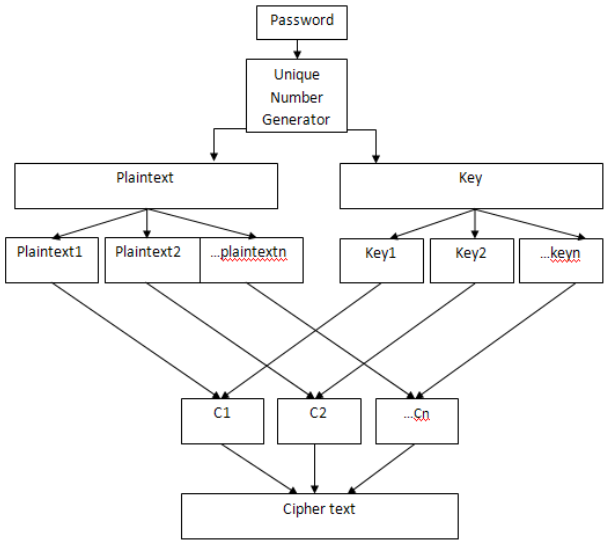

Figure 1: Split-plaintext-key pair algorithm –Encryption

2.3. PRNG (Pseudo Random Number Generation)

The unique random number 'r' is generated from the password i.e. the key by the PRNG method. In this method, each letter of the password is converted to ASCII. Depending on the number of characters of the key as well as the ASCII value of each character a unique number 'n' is generated. This unique number 'n' represents the number of splits on the plaintext and key. Figure 2 illustrates the randomness of the unique random number generator for 50 trials. For each trial a password is entered and a random number is generated. Even though all the passwords are almost likely the same, the PRNG result preserves its uniqueness and randomness. Table1 shows how unique and random the PRNG result is for 50 trials for almost similar inputs for passwords.

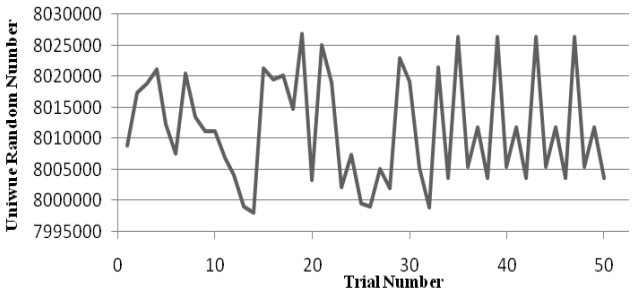

Figure 2: Number of trials v/s randomness generator

Table 1: PRNG Uniqueness & Randomness

Trial No:	PRNG Result	Trial No:	PRNG Result
1	8008883	26	7998975
2	8017370	27	8005184
3	8018888	28	8002016
4	8021248	29	8023010
5	8012364	30	8019196
6	8007658	31	8005331
7	8020503	32	7998848
8	8013520	33	8021514
9	8011162	34	8003583
10	8011115	35	8026464
11	8006929	36	8005392
12	8004176	37	8011766
13	7999021	38	8003583
14	7998063	39	8026464
15	8021314	40	8005392
16	8019530	41	8011766
17	8020184	42	8003583
18	8014784	43	8026464
19	8026943	44	8005392
20	8003319	45	8011766
21	8025065	46	8003583
22	8019115	47	8026464
23	8002164	48	8005392
24	8007438	49	8011766
25	7999476	50	8003583

2.4 Implementation

The implementation of the algorithm is done by using Visual Studio.NET 2010 using C#. The files are used for storage of data.

A sample plaintext (see Figure 3) and its corresponding cipher text (see Figure 4) are shown below.

2.4.1 Sample plaintext

On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document. You can use these galleries to insert tables, headers, footers, lists, cover pages, and other document building blocks. When you create pictures, charts, or diagrams, they also coordinate with your current document look. You can easily change the formatting of selected text in the document text by choosing a look for the selected text from the Quick Styles gallery on the Home tab. You can also format text directly by using the other controls on the Home tab. Most controls offer a choice of using the look from the current theme or using a format that you specify directly. To change the overall look of your document, choose new Theme elements on the Page Layout tab. To change the looks available in the Quick Style gallery, use the Change Current Quick Style Set command. Both the Themes gallery and the Quick Styles gallery provide reset commands so that you can always restore the look of your document to the original contained in your current template. On the Insert tab, the galleries include items that are designed to coordinate with the overall look of your document.

Plaintext Size =1201

Key length = 8

PRNG Result (r) = 756

Number of Splits (n) = 7

Size of Splits =171

Number of padded characters =4

2.4.2Cipher text

Vu'{ol'Puzly{'{hi3'{ol'nhsslyplz'pujs|kl'p{ltz'{oh{'hyl'klzpnulk'{ v'jvvykpuh{l'~p{o'{ol'v}lyhss'svvr'vm'€ v|y'kvj|tlu{5`v|'jhu'|zl'{ olzl'nhsslyplz'{v'puzly{'{hislz3'olhklyz3'mvv{lyz3'spz{z3'jv}ly' whnlz3'huk'v{oly'kvj|tlu{'i|pskpun'isvjrz5'^olu'€ v|'jylh{l'wpj{|yl z3'johy{z3'vy'kphnyhtz3'{ol€ 'hszv'jvvykpuh{l'~p{o'€ v|y'j|yylu {'kvj|tlu{'svvr5`v|'jhu'lhzps€ 'johunl'{ol'mvyth{{pun'vm'zlslj{lk' {l□{'pu'{ol'kvj|tlu{'{l□{'i€ 'jovvzpun'h'svvr'mvy'{ol'zlslj{lk'{l□{' myvt'{ol'X|pjr'Z{€ slz'nhssly€ 'vu'{ol'Ovtl'{hi5`v|'jhu'hszv'mvyt h{'{l□{'kpylj{s€ 'i€ '|zpun'{ol'v{oly'jvu{yvsz'vu'{ol'Ovtl'{hi5'Tv z{'jvu{yvsz'vmmly'h'jovpjl'vm'|zpun'{ol'svvr'myvt'{ol'j|yylu{'{o ltl'vy'|zpun'h'mvyth{'{oh{'€ v|'zwljpm€ 'kpylj{s€ 5'[v'johunl'{ol' v}lyhss'svvr'vm'€ v|y'kvj|tlu{3'jovvzl'ul~'[oltl'lsltlu{z'vu'{ol'Wh nl'Sh€ v|{'{hi5'[v'johunl'{ol'svvrz'h}hpshisl'pu'{ol'X|pjr'Z{€ sl'n hssly€ 3'|zl'{ol'Johunl'J|yylu{'X|pjr'Z{€ sl'Zl{'jvtthuk5'Iv{o'{ol'[oltlz'nhssly€ 'huk'{ol'X|pjr'Z{€ slz'nhssly€ 'wyv}pkl'ylzl{'jvtthu kz'zv'{oh{'€ v|'jhu'hs~h€ z'ylz{vyl'{ol'svvr'vm'€ v|y'kvj|tlu{'{v'{ ol'vypnpuhs'jvu{hpulk'pu'€ v|y'j|yylu{'{ltwsh{l5'Vu'{ol'Puzly{'{ hi3'{ol'nhsslyplz'pujs|kl'p{ltz'{oh{'hyl'klzpnulk'{v'jvvykpuh{l'~ p{o'{ol'v}lyhss'svvr'vm'€ v|y'kvj|tlu{5""

2.5 Decryption

By applying the unique number generation, the password of the file (symmetric key) is divided to get the number of splits. The key and cipher text are split accordingly. The split key is applied on the corresponding split cipher text (i.e ith key split is applied on ith cipher text) for decryption. Combine the plaintext splits to complete the decryption process as shown in Figure 5.

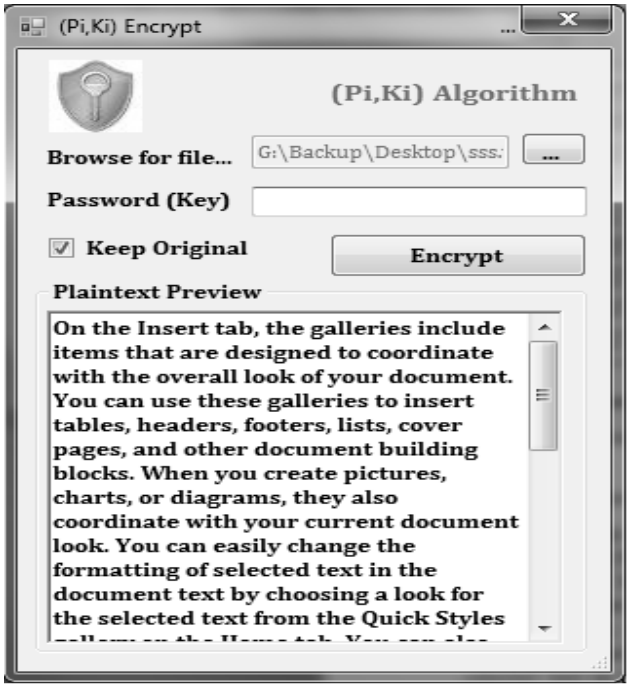

Figure 3: Screenshot of Encryption – plain text view

Figure 4: Screenshot of Encryption – cipher text view

2.5 Flowchart – decryption

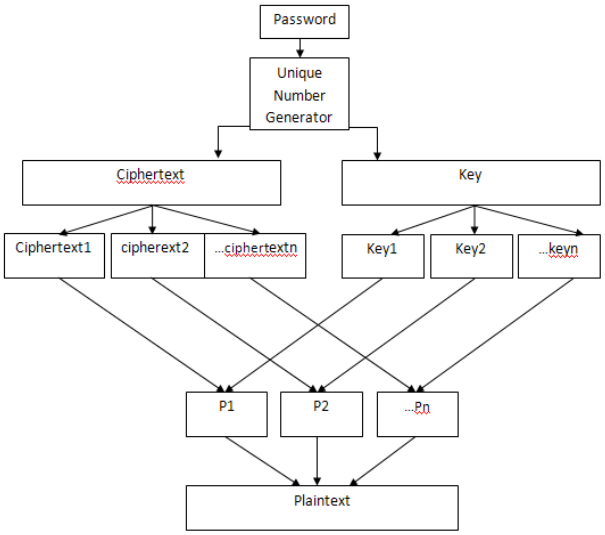

Figure 5: Split-plaintext-key pair algorithm –Decryption

2.6 Results

The execution time taken for the SPKP algorithm versus different symmetric algorithms for plaintexts of varying size is also tested. The algorithms considered are AES, DES and 3DES. It is also mandatory to consider the system in which you are testing the complexity of the algorithm. The system with the following configuration is used for the algorithm testing.

OS : Windows 7 Professional

Processor : Intel(R) Core (TM) i5 2004 CPU @ 3.10 GHz 3.10 GHz

RAM : 2GB

Type : 64 bits OS

Table 2: Plaintext sizes

Plaintext Type	Number of characters
Very Large	63544
Large	37719
Medium	4191
Small	177

Table 3: Time Complexity v/s Plaintext sizes

Algorithm	Execution Time for different plaintext sizes(mS)			
	Very Large	Large	Medium	Small
SPKP	6.64	4.59	2.41	2.41
AES	58.29	58.88	53.47	58.48
DES	3.42	2.45	0.68	0.53
3DES	7.25	5.03	0.76	0.34

Table 4: Change in time complexity w.r.t varying plaintext sizes

Algorithm	Change in time complexity
Scenario 1: Large v/s very large plaintext	
SPKP	2.05
AES	-0.59
DES	0.97
3DES	2.22
Scenario 2: Medium v/s large plaintext	
SPKP	2.18
AES	5.41
DES	1.77
3DES	4.27
Scenario 3: Small v/s medium plaintext	
SPKP	0
AES	-5.01
DES	0.15
3DES	0.42

Having glance through the execution time of the algorithms with respect to plaintexts of different sizes, it is very clear that the SPKP algorithm maintains a steady execution time irrespective of the size of plaintext. SPKP lags behind DES & 3DES for small plaintext, but as size increases the margin for change for SPKP is less and it is steady as well.

3. CONCLUSION

In this paper, a new symmetric encryption algorithm is presented. The algorithm uses the concept of splitting the plaintext and key equally and applies shift cipher for encryption. It needs a secured channel to exchange the key [8] between the sender and receiver. It follows a new hybrid approach of splitting the plain text and the key. Non-repudiation [9] factor can be incorporated in the algorithm by including a watermark key [10] which is derived from the properties of each split plaintext. Further, the algorithm may be redesigned for random pairing between splits of plaintext & key. The results prove that the proposed algorithm is competent with similar algorithms [11] in the case of time complexity. It can also be altered in such a way that a matrix of plaintext & key [12] can be maintained. The coding behind SPKP algorithm is not optimized and hence the time complexity shown in the results can be fine tuned to better values to get improved results.

4. REFERENCES

[1] Renjith PR, Arun Sojan, Praseeda K Gopindhan, "A Novel Method for Symmetric Encryption using Split Plaintext Key Pair (Pi,Ki) Algorithm", International Journal of Computer Applications, Special Issue on Network Security and Cryptography November 2011

[2] William Stallings, "Cryptography and Network Security" 4th Edition. Pearson Education Inc, Upper Saddle River, New Jersey, 2006.

[3] Kerckhoffs's principle or Kerckhoffs's Desiderata of Cryptography Online : http://en.wikipedia.org/wiki/Kerckhoffs's_principle accessed 11 June 2011

[4] Kenneth W. Dam, National Research Council (U.S.). Committee to Study National Cryptography Policy "Cryptography's Role in Securing the Information Society" pp 202

[5] Christopher Paar, Jan Pelzl, Bart Preneel,"Understanding Cryptography- A Textbook for Students and Practitioners", Springer, Berlin Heidelberg 2010.

[6] Shift cipher Online: en.wikipedia.org/wiki/Caesar_cipher

[7] Frequency analysis Online: http://en.wikipedia.org/wiki/Frequency_analysis accessed 11 June 2011

[8] Diffie Hellman key exchange online: en.wikipedia.org/wiki/Diffie–Hellman_key_exchange accessed 11 June 2011

[9] Schneier. Applied Cryptography. Wiley and Sons, 1996.

[10] Ding Huang and Hong Yan, "Interword Distance Changes Represented by Sine Waves for Watermarking Text Images" Online: www.sydney.edu.au/engineering/it/~vip2000/papers/ accessed 8 June 2012

[11] Monika Agrawal and Pradeep Mishra, "A Comparative Survey on Symmetric Key Encryption Techniques",

International Journal on Computer Science and Engineering May 2012

[12] V. U. K. Sastry,D. S. R. Murthy,S. Durga Bhavani,Block Cipher Having a Key on One Side of the Plain Text Matrix and its Inverse on the Other Side, International Journal of Computer Theory and Engineering, Vol. 2, No. 5, October, 2010