VOL. 9, NO. 12, DECEMBER 2014 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

PERFECT SHUFFLE ALGORITHM FOR CRYPTOGRAPHY

Ernastuti
Jalan Margonda 100, Depok, Indonesia
E-Mail: ernas@staff.gunadarma.ac.id

ABSTRACT

The problem of protecting the messages from the hackers is usually solved by cryptographic methods. This led to create various algorithms of encryption and decryption. This paper proposes the new algorithm based on perfect shuffle technique. It is called Perfect Shuffle Crypto Algorithm (PSCA) which is classified as a transposition or permutation technique in the crypto system. The PSCA is an asymmetry key encryption, uses a pair of keys, that are a public key for encrypting data, and a corresponding private secret key for decrypting. PSCA is very fast and simple for technical realization. For the linear plaintext length of $N=2^n$, it will take O (N log N) to complete both encrypting plaintext and decrypting cipher text. The PSCA is reasonably secure, especially for chipertext-only attack. It is enable to apply PSCA as an basic algorithm alternative to develop or create an crypto algorithm which employ the layered scheme.

Keywords: encryption, ciphertext, decryption, perfect shuffle, plaintext, public key.

1 INTRODUCTION

The problem of protecting the messages from the hackers is solved usually with application of cryptographic methods [2]. By progressively more information is stored and transmitted in electronic form, then security in the data transmitting process is urgently required. The study of cryptography has become important. This led to the creation of many encryption algorithms and standards [1] [9]. Cryptography is science or art to maintain the security of the message [2]. The cryptographic system is illustrated as Figure-1 as follows:

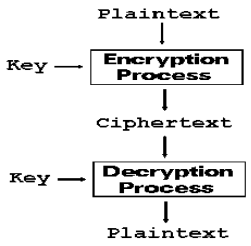

Figure-1. Cryptographic System.

A cipher is an algorithm for performing encryption (and the reverse, decryption). The original information is known as plaintext, and the encrypted form as cipher text. The cipher text message contains all the information of the plaintext message, but is not in a format readable by a human or computer without the proper mechanism to decrypt it [2]. A cryptographic algorithm works incombination with a key-a word, number, or phrase-to encrypt the plaintext.

The same plaintext encrypts to different ciphertext with different keys. The security of encrypted data is entirely dependent on two things: the strength ofmthe cryptographic algorithm and the secrecy of the key [9].

In conventional cryptography, also called secret-key or symmetric-key encryption, one key is used both for encryption and decryption. The Data Encryption Standard (DES) is an example of a conventional cryptosystem that is widely employed by the Federal Government. Cryptographic strength is measured in the time and resources it would require to recover the plaintext. The result of strong cryptography is ciphertext that is very difficult to decipher without possession of the appropriate decoding tool. How difficult? Given all of today's computing power and available time, even a billion computers doing a billion checks a second, it is not possible to decipher the result of strong cryptography before the end of the universe [9].

The fundamental objective of cryptography is to enable two people, to communicate over an insecure channel in such a way that an opponent cannot understand what is being said.

In modern cryptography, also called public key or asymmetric key encryption, uses a pair of keys for encryption: a public key, which encrypts data, and a corresponding private, or secret key for decryption. We publish our public key to the world while keeping our private key secret. Anyone with a copy of our public key can then encrypt information that only we can read. Even people we have never met.

The primary benefit of public key cryptography is that it allows people who have no preexisting security arrangement to exchange messages securely. The need for sender and receiver to share secret keys via some secure channel is eliminated; all communications involve only public keys, and no private key is ever transmitted or shared. Some examples of public-key cryptosystems are Elgamal (named for its inventor, Taher Elgamal), RSA (named for its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman), Diffie-Hellman Algorithm, and DSA, the Digital Signature Algorithm (invented by David Kravitz).

All cryptographic functions must have the properties of reversibility, which is able to restore the encrypted cipher text back to plain text through decryption process.

In this paper, the new algorithm based on perfect shuffle technique is proposed and implemented. It is called Perfect Shuffle Crypto Algorithm (PSCA). The perfect shuffle has many interesting mathematical properties and applications in computer science [3]. The PSCA is asymmetry scheme, uses a pair of keys: a public key for encryption, and a secret key for decryption. In this paper, is presented procedure of PSEA, as well algorithm and security analysis. The illustration of executing PSCA is also presented.

The rest of this paper is organized in the following sections: Section 2 shows the materials and methods that are the basis of the idea of building PSEA. Section 3 describes procedures of encryption and description in PSEA, while Section 4 discusses about cryptosystems analysis, and finally Section 5 lays down the conclusion.

2 MATERIAL AND METHODS

The idea of building the PSCA came from the shuffle-exchange interconnection network model for parallel computation. The model is described in section 2.1, and properties of the model are presented in section 2.2.

2.1 SHUFFLE EXCHANGE NETWORK MODEL

Shuffle exchange network model is based on two routing functions, shuffle and exchange [3] [4]. A perfect shuffle of N = 8 processors is presented in Figure-2.1 (a). Perfect shuffle cuts the deck into two halves from the center and then inter mixes them evenly. Inverse perfect shuffle does the opposite to restore the original ordering as shown in Figure-2.2 (b).

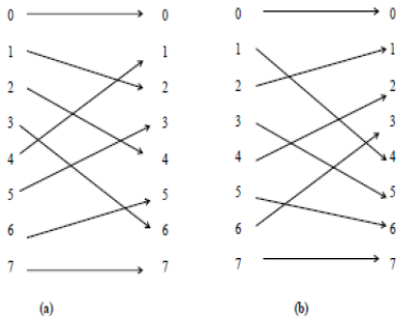

Figure 2.1: (a) Perfect Shuffle (b) Inverse Perfect Shuffle

In other word, perfect Shuffle connection link node i to node j in the following way:

$j = 2*i$	, $0 \le i \le 2^n/2-1$
$j = 2*i + 1 - 2^n$	, other

These shuffle exchange functions can be implemented as either recirculating network or a multistage network. Figure-2.3 produces a single stage recirculating shuffle exchange network.

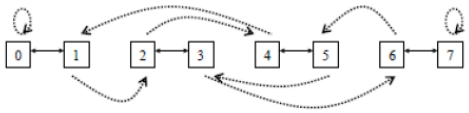

Figure 2.2: Shuffle Exchange Recirculating Network for N=8

For more obvious, a shuffle exchange network consists of $n = 2^k$ nodes, numbered 0, 1, …, n-1, and two kinds of connection, called *shuffle* and *exchange*. Exchange connections link pairs of nodes whose numbers differ in their least significant bit. The perfect shuffle connection links node i with node 2i modulo n-1, with the exception that node n-1 is connected to itself [5]. See the Figure-2.3, dashed arrows denote shuffle connections, and solid arrows denote exchange connections.

To understand the derivation of the name perfect shuffle, consider shuffling a deck of eight cards, numbered 0, 1, 2, 3, 4, 5, 6, 7. If the deck is divided into two exact halves and shuffled perfectly, then the result is the following order: 0, 4, 1, 5, 2, 6, 3, 7.

2.2 PROPERTIES SHUFFLE EXCHANGE

A multistage Shuffle Exchange Network has $N = 2^n$ inputs, termed sources (S), and 2^n outputs, termed destinations (D). It has n stages and each stage has N/2 switching elements [3]. There is a unique path between each source-destination pair [7]. Figure-2.3 exhibits a Shuffle Exchange Network of size 8 x 8. The Shuffle Exchange Network is a self-routing network [8].

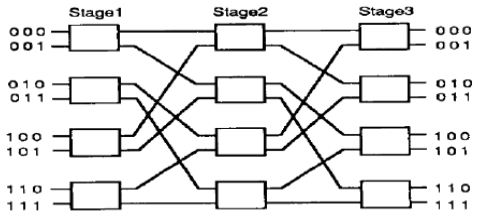

Figure 2.3: a Shuffle Exchange Network of size 8 x 8

Let $a_n a_{n-1} a_{n-2} \ldots a_1$ be address of a node in perfect shuffle network, expressed in binary. A datum at this address will be at address $a_{n-1} a_{n-2} \ldots a_1 a_n$ following a shuffle operation. In other words, the change in the address of a piece of data after a shuffle operation corresponds to left cyclic rotation of the address by 1 bit. If $N = 2^n$, then *n* shuffling operations move a datum back to its original location [5]. Figure-2.3 show the illustration of $\log_2 8$ shuffling operation, the data back to its original position. Further more, if $N = 2^n$, then *2n* shuffle-exchange operations move a datum back to its original location [6].

3. PERFECT SHUFFLE CRYPTO

"How many shuffle exchanges will it take to come back to the starting point?" It will take *2n* times of shuffle exchange operations definitely to back to the original data position. Based on the interesting properties of shuffle exchange network model, then it is enable

obtaining a mind to build a new encryption and decryption algorithm. In this paper proposes a crypto algorithm, it is called the perfect shuffle crypto algorithm (PSCA).

3.1 PRINCIPLE OF CRYPTOGRAPHY

Standardization of writing on cryptography can be written in the language of mathematics. The fundamental function in cryptography is encryption and decryption. Encryption is the process of converting an original message (plaintext) into a message in language code (ciphertext), denoted by C = E (P), (where P = the original message; E = encryption process; C = the coded message), and decryption is the process of changing the message in a coded language back into the original message. P=D(C), where D = the decryption process. Generally, in addition to using certain functions in performing encryption and decryption, the function is often given an additional parameter called the key.

3.1 ENCRYPTION

The input is a linear list ($p_1p_2…p_2^n$) of plaintext with a public key consisting of only one positive integer *k*, where $k < n$. The encryption works as follows. First, it performs the shuffle operation on $p_1p_2…p_2^n$. Then on the result is performed the exchange operation. Use the two operations until k rounds, where k < n. The following algorithm describes the procedure of encryption.

PSEUDOCODE

SHUFFLE EXCHANGE ENCRYPTION

Initial condition: List of N ≥ 1 element of plaintext stored in P [0…2^n – 1] and a public key k.

Final condition: Each element chipertext C[i] contains the result of shuffle-exchange operation on P[i] at k rounds.

Global variables: N, n, P [0…2^n – 1], C [0…2^n – 1], k

```
begin
 n ← log2 N                          //* N: length of plaintext file*//
 if n ≠ ⌈ log2 N⌉ then                          //* for N < 2^n *//
                        for i ← N+1 to 2^n – 1 do
                            P[i] ← $
                         endfor
 for i ← 0 to 2^n – 1 do
       T[i] ← P[i]
 endfor
 Input k                 //* k is a public key, and 1 ≤ k < n*//
 for i ← 1 to k do
   for j ← 0 to 2^n –1 do       //*Shuffle operation*//
        if (j mod 2 = 0) and (j ≠ 0 or j ≠ 2^n –1)
         then B[j] ← T[j/2]
         else
           If (j mod 2 ≠ 0) and (j ≠ 0 or j ≠ 2^n –1)
              then B[j] ← T[⌊j/2+(2^n –1)/2⌋]
    endfor
    for j ← 0 to 2^n –1 do    //*Exchange operation*//
          If j mod 2 = 0    then   C[j] ← B[j+1]
                            else    C[j] ← B[j–1]
     endfor
     for j ← 0 to 2^n – 1 do
          T[j] ← C[j]
     endfor
endfor
end
```

Encryption using PSCA is illustrated by this example. The Table-1 shows execution of the encryption algorithm for N = 8 elements plaintext, and the key k = 2.

Table 1. An illustration of encryption, with N = 8, k = 2.

PlainText	A	B	C	D	E	F	G	H
	0	1	2	3	4	5	6	7
Encryption								
i =1								
Shuffle	A	E	B	F	C	G	D	H
Exchange	E	A	F	B	G	C	H	D
i=2								
Shuffle	E	G	A	C	F	H	B	D
Exchange	G	E	C	A	H	F	D	B
Chipertext	G	E	C	A	H	F	D	B

3.2 DECRYPTION

The decryption algorithm is similar to the encryption algorithm, but using the two operations until *2n* – k rounds, where k < n. The input of decryption is a linear list ($c_1c_2…c_2^n$) of chipertext and a private key consisting of only one positive integer *2n–k*, where k < n. The following algorithm is the procedure of decryption.

PSEUDOCODE

SHUFFLE EXCHANGE DECRYPTION

Initial condition: List of N ≥ 1 element of chipertext stored in C [0…2^n - 1] and a private key v = (*2n – k*).

Final condition: Each element P[i] contains the result of shuffle-exchange operation on C[i] at n – k rounds

Global variables: N, n, P [0…2^n – 1], C [0…2^n – 1], k, v

```
begin
 for i ← 0 to 2^n – 1 do
    T[i] ← C[i]
 endfor
 v = 2n – k            //* v is a private key, and 1 ≤ k < n*//
 for i ← 1 to v do
   for j ← 0 to 2^n –1 do              //*Shuffle operation*//
        if (j mod 2 = 0) and (j ≠ 0 or j ≠ 2^n –1)
         then B[j] ← T[j/2]
         else
           If (j mod 2 ≠ 0) and (j ≠ 0 or j ≠ 2^n –1)
              then B[j] ← T[⌊j/2+(2^n –1)/2⌋]
    endfor
    for j ← 0 to 2^n –1 do          //*Exchange operation*//
          If j mod 2 = 0    then   P[j] ← B[j+1]
                            else    P[j] ← B[j–1]
     endfor
     for j ← 0 to 2^n – 1 do
          T[j] ← P[j]
     endfor
endfor
end
```

The following Table-2 shows an example of decrypction using PSCA for N = 8 elements in ciphertext, and the key v = 4.

Table 2. An illustration of decryption, with N = 8, v = 2.

Chipertext	G	E	C	A	H	F	D	B
	0	1	2	3	4	5	6	7
Decryption								
i =1								
Shuffle	G	H	E	F	C	D	A	B
Exchange	H	G	F	E	D	C	B	A
i=2								
Shuffle	H	D	G	C	F	B	E	A
Exchange	D	H	C	G	B	F	A	E
i =3								
Shuffle	D	B	H	F	C	A	G	E
Exchange	B	D	F	H	A	C	E	G
i=4								
Shuffle	B	A	D	C	F	E	H	G
Exchange	A	B	C	D	E	F	G	H
Plaitext	A	B	C	D	E	F	G	H

4 CRYPTOSYSTEM ANALYSES

4.1 TIME COMPLEXITY

The PSCA takes the time of O (N log N) to complete encryption and description, where $N = 2^n$. It's obtained from a proof as follow. In Encryption, there is a looping *for* $i \leftarrow 1$ *to* k *do* that followed by a looping for $j \leftarrow 0$ *to* $2^n - 1$ do. It shows that each iteration i performs $2^n - 1$ shuffle operations, where $1 \le i \le k$, and $k < n$. Thus, the encryption takes $O(k*2^n)$. There is also a looping *for* $i \leftarrow 1$ *to* v *do* (in decryption) that followed by a looping *for* $j \leftarrow 0$ *to* $2^n - 1$ *do*. It shows that each iteration i requires $2^n - 1$ shuffle operations, with $1 \le i \le v$, and $v = n-k$. The decryption takes O $((n-k)*2^n)$. Therefore, it is obvious that PSCA takes O $(k*2^n)$ + O $((n-k)*2^n)$. In other word PSCA complexity is O (log N)*O (N) or can say O (N log N). So that it is considered as a fast cryptosystem

4.2 SECURITY ANALYSIS

The PSCA is a transposition method which at least require N! possibility to discover plaintext from its chipertext. PSCA builds cryptosystem in asymmetry scheme. The security of PSEA comes from the shuffle and exchange operations, and also its private key. It may be difficult to discover the private key which is in range of logarithm of basis 2. The cryptalist must be able to create an invers of shuffle-exchange function to back to original position, if they are not able to crack the private key. The chipertext is a random sequence that is built through shuffle and exchange operations in k iterations, where $k \in \{1, 2,\ldots, n-1\}$ with $n = \log_2 N$, k is its public key. The index order of chipertext sequence is tightly influenced by previous sequence. Thus, to recognize the pattern of chipertext is going to become difficult, especially for the chipertext-only attack. The technique used to perform cryptanalysis of the PSEA Cipher is known-plaintext attack. If cryptanalysis has plaintext and ciphertext fractions which mutually correspondence, the Cipher can be solved.

5. CONCLUSIONS

PSCA is a fast cryptosystem which is simple for realization. Thus, it can be used for cryptographic protection of any text file in O (N log N) of time, N is length of text file (plaintext). PSEA is reasonably secure in facing the ciphertext-only attack, but weak if attacked by known-plaintext attack. It is good to apply PSCA as an alternative in developing or creating a new algorithm which employ the layered scheme.

REFERENCES

[1] Abdelfatah A. Y. and Ayman M. A. 2008. A Shuffle Image-Encryption Algorithm. Journal of Computer Science. 4(12): 999-1002. ISSN 1549-3636.

[2] Ernastuti and Ravi A.S. 2009. The Application of ELC Numbers to Golden Cryptography. In Proceedings of the 5th International Conference on Information and Communication Technology and Systems (ICTS) 2009, pp. 329-334. ISSN: 2085-1944.

[3] John E. and Hongbing. F. 2002. The Cycles of Multiway Perfect Shuffle. Discrete Mathematics and Theoretical Computer Science 5. pp. 169-180.

[4] Graham L.R. and William M.K. 1983. The Mathematics of Perfect Shuffles. Advances in Applied Mathematics. 4. pp. 175-196.

[5] Quinn J.M. 1987. Designing Efficient Algorithms for Parallel Computers. McGraw Hill International Editions. pp. 26-28. ISBN 0-07-051071-7.

[6] Das N. and Bhattacharya. B. 1998. Permutation admisilibility in shuffle-exchange networks with arbitrary number of stages. High Performance Computing. pp. 270-276. ISBN: 0-8186-9194-8.

[7] Schiano. L. and Lombardi .F. 2003. On the test and Diagnosis of Perfect Shuffle. In Proceedings. 18th IEEE International Symposium. pp. 97-104. ISSN: 1550-5774.

[8] Arzilawati N. and Yunus M.D. 2011. Shuffle Exchange Network in Multistage Interconnection Network: A Review and Challenges. International Journal of Computer and Electrical Engineering. 3(5): 724-728.

[9] Markovic M. and Dordevic G. 2006. On Implementation aspects of Standard Asymetric and Symetric Cryptographic Algorithms on TI Signal Processors. Security, Privacy and Trust in Pervasive and Ubiquitous Computing, Second International Workshop on. pp. 57-62