ISSN 2278-3091

Volume 9, No.1.1, 2020

International Journal of Advanced Trends in Computer Science and Engineering

Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse1691.12020.pdf

https://doi.org/10.30534/ijatcse/2020/1691.12020

Secured Private Key Handling using Transposition Cipher Technique

Maricel Grace Z. Fernando[1], Ariel M. Sison[2], Ruji P. Medina[3]

[1]Technological Institute of the Philippines, Philippines, maricelgracefernando@gmail.com
[2]Emilio Aguinaldo College, Philippines, ariel.sison@eac.edu.ph
[3] Technological Institute of the Philippines, Philippines, ruji.medina@tip.edu.ph

ABSTRACT

Developments in communications technology motivate most businesses and academic institutions to move towards the use of digital documents. Digital signature is an alternative way in signing a document using a pen. It is used for verifying the authenticity of digital messages or documents. A private key is required to sign a digital document. It is a vulnerable component of the public key cryptosystem. Once unauthorized user gains knowledge of the private key, a valid digital signature can be produced on behalf of the registered user.

This paper proposes a transposition cipher technique to protect the private key from unauthorized users. It rearranges the sequence of private key characters. The mapping of characters' position shifting is based on the defined Fibonacci series and created equations. This technique makes the private key unusable, thus prevent unauthorized access in case of loss or theft.

The degree of security performance of the proposed technique is evaluated using the correlation coefficient cryptographic test. The correlation values determine the confusion effect of the block cipher. The results obtained from the test showed that the proposed algorithm provides a strong non-linear relationship between the plaintext and ciphertext.

Key words : Correlation Coefficient, Cryptography, Private Key, Transposition Cipher,

1. INTRODUCTION

Many business institutions adapt the use of recent technology in communicating with its clients and employees. Data are prone to manipulation and misuse [1] when transmitted or while in storage. It significantly demands strong and effective means to keep it secure and private.

Digital signature is used to sign and authenticate a digital document. It is equivalent to signing a printed document using a pen. RSA is the most popular public key cryptosystem for digital signature [2][3] that uses asymmetric or public key cryptography. The public key is disclosed to everyone while the private key should be kept securely by the owner. The security of the system relies on the private key [4] and is used to create a digital signature. Anyone who has a copy of a private key can create an authenticated digital signature on the owner's behalf. It is essential to ensure the confidentiality of the key [5].

However, private key is long [5] and difficult to remember. [6] Keeping it in different places increases the risk of being compromised [7]. Hence, an effective way to secure and maintain confidentiality and integrity should be applied. A transposition cipher rearranges characters to produce a ciphertext. This technique is used to ensure that the private key is safe while it is in removable storage.

Biometrics [8] and Smart Cards [9][10] can be used to protect the private keys but these methods need a special reader that entails an additional cost to the organization or business.

This paper proposes a technique to protect the private key by rearranging the sequence of characters prior to keeping in a storage device. Furthermore, the confusion effect of the 128-character block is evaluated by calculating the correlation values.

2. METHODOLOGY

The proposed technique aims to provide security on the user's private key. It utilizes RSA algorithm-generated private key plaintext to produce the ciphertext. This technique shuffles and recovers the characters based on the defined Fibonacci series and created equations.

2.1 Encryption Algorithm

A. Private Key Block

The plaintext is divided into number of blocks. Each block contains 120 characters from the RSA private key. A total of 8 (eight) random characters are padded to the block before the encryption process. If the size of the last block is less than the initial required number of characters, additional characters are padded to complete the 128-character block.

B. Key Selection

The algorithm uses the Fibonacci series (FS) values as encryption keys. FS value 1, 2, 3, 5, 8, 13, 21, 34 and 55 dictates the character position shifting. This is used as the round key on the encryption process. It causes an irregular position shifting of characters.

C. Character Position Mapping

Character position is planned to change unevenly. The adjustment of private key character location is caused by the equations (1) – (4). Values for the equations are startVal = 0, midVal = 64, endVal = 128 and locNo = 0. CTxt for ciphertext, PTxt for plaintext and locNo for the character location in an array.

$$CTxt[locNo{+}{+}] = PTxt[midVal - FS] \tag{1}$$

$$CTxt[locNo{+}{+}] = PTxt[midVal + FS] \tag{2}$$

$$CTxt[locNo{+}{+}] = PTxt[endVal - FS] \tag{3}$$

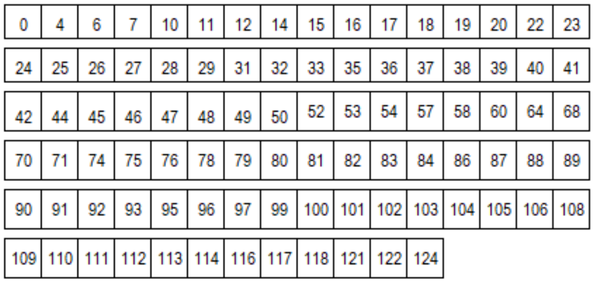
$$CTxt[locNo{+}{+}] = PTxt[startVal - FS] \tag{4}$$

The encryption key is the initial FS value to be used in the equation. The technique produces a different sequence of characters depending on the key. Shown in Table 1 are the FS values and the calculated values from created equations (1) – (4). The value is wrapped around until fully employed in the mapping of character position shifting.

Table 1: Derived Values Based on FS Value

FS Value	**Derived Values from Equation**			
	(1)	***(2)***	***(3)***	***(4)***
1	63	65	127	1
2	62	66	126	2
3	61	67	125	3
5	59	69	123	5
8	56	72	120	8
13	51	77	115	13
21	43	85	107	21
34	30	98	94	34
55	9	119	73	55

After the 36 characters are pulled-out and moved to a new location, remaining characters are taken and arranged in ascending order based on location number in an array. Shown in Figure 1 is the sequence of the remaining 92 characters.



0	4	6	7	10	11	12	14	15	16	17	18	19	20	22	23
24	25	26	27	28	29	31	32	33	35	36	37	38	39	40	41
42	44	45	46	47	48	49	50	52	53	54	57	58	60	64	68
70	71	74	75	76	78	79	80	81	82	83	84	86	87	88	89
90	91	92	93	95	96	97	99	100	101	102	103	104	105	106	108
109	110	111	112	113	114	116	117	118	121	122	124				

Figure 1: Series of characters for location 36 – 127

D. Encrypted Private Key Block

Position shifting of 128 characters is done until 5 (five) rounds are completed. Ciphertext blocks are then oncatenated to produce the ciphertext of each block.

2.1 Decryption Algorithm

The ciphertext is divided into blocks of 128 characters. The process needs a similar key used during the encryption process. Private key plaintext is recovered through the character position shifting that is based on the created equations (5) – (8). Values to be used for the equations are startVal = 0, midVal = 64, endVal = 128 and locNo = 0. CTxt for ciphertext, PTxt for plaintext and locNo for the character location in an array.

$$PTxt[midVal - FS] = CTxt[locNo{+}{+}] \tag{5}$$

$$PTxt[midVal + FS] = CTxt[locNo{+}{+}] \tag{6}$$

$$PTxt[endVal - FS] = CTxt[locNo{+}{+}] \tag{7}$$

$$PTxt[startVal + FS] = CTxt[locNo{+}{+}] \tag{8}$$

After proposed technique is performed on each block, 8 (eight) characters occupying location number 120 – 127 on each block are removed. Putting the blocks in correct order completes the process of recovering the plaintext.

3. ILLUSTRATION OF THE PROPOSED TECHNIQUE

3.1 Encryption Algorithm

A. Private Key Plaintext

An RSA 1024-bit generated private key is shown in Figure 2. It is composed of 812 characters. The plaintext is divided into 120 characters per block, 6 full blocks are produced. For the last block with only 92 characters, random characters are added before performing the encryption process.

```
MIICWwIBAAKBgQCqQBOlSu4sWo5YQH7MrqNRanac
WLwbMRgoIJ4iyaUi5FAJlKoxgIM7MngPAhZ2oBl0OY
YR5VL/slspjveYALJBDIvl1eXjU116IGOlTnzZRjPny2m
faCLqU4lK8IMCAIgYOMRzIa5WUukLmCT1XJj8PPN+
EsZLRBM8QKfD4oCfpwIDAQABAoGAGVa0FdPTIvLP
YL8ooBoV4JYx1Q833+wspjxEjPTqCQlmCCcCNZ/k6Di
YudMbuMgXQb8V/y/cbIYFH2hR3WOtI9be1rztFSSKBb
3UlrTM6IgEmE9p9Fisk9ii6GSIhSY4AUvU1j982LBJIuq
eIErmYhPGr15k6ByzUS+H0NJ18iECQQDm3ZVTGmyH
bJoRPeDixikomsEQ3nf+B84xZD2ttx6PcUZYpYNJVGN
X/BT8HrmzohfMP1dMVM5mbXujA/H/1KdZAkEAvMk
YZ2YncJNUEZsfqqkpqT980Gzs3sUQIi1sY8ocsLsT6e7n
wMEJkv9bMHE9J8QArSdW/ICKGprWUy5Vye2e/wJAZ
Hot4kRO1cXbbAOdW7VY9nYb7uaUHLhy8iYpffCD2X
Q4ZJ5sfD7BGer5ix0oP1kduGQc73i9d/DQbYtrCKgZyQJ
AXGjTWUnX5HM5o6SbS56ilgZNwniPOjycOR6i//XfN
WNdzBT1vJTBawM0T/IeTWjUK+411lPVRlCkuPL1a4K1
YQJAfDweyxcyP2+fUTCkjii8U4gk/kxiJmf5hCNdX3HG
KbEfqSzMXmnA5iTPfQRcuZzl6ncnT2Y3lNffTN7MzSI
Qcw==
```

Figure 2: Sample RSA 1024-bit generated private key

B. Padded Characters

Before the encryption process, 8 (eight) characters are padded to occupy the location number 120 – 127 in an array. Shown in Figure 3 are the colored boxes with an asterisk.

LN	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PTC	M	I	I	C	W	w	I	B	A	A	K	B	g	Q	C	q
LN	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
PTC	B	O	l	S	u	4	s	W	o	5	Y	Q	H	7	M	r
LN	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
PTC	q	N	R	a	n	a	c	W	L	w	b	M	R	g	o	I
LN	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
PTC	J	4	i	y	a	U	i	5	F	A	J	l	K	o	x	g
LN	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
PTC	I	M	7	M	n	g	P	A	h	Z	2	o	B	l	0	O
LN	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
PTC	Y	Y	R	5	V	L	/	s	l	s	p	j	v	e	Y	A
LN	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
PTC	L	J	B	D	I	v	l	1	e	X	j	U	1	1	6	I
LN	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
PTC	G	O	l	T	n	z	Z	R	*	*	*	*	*	*	*	*

LN - Location in an array

PTC - Plaintext character

Figure 3: Private Key Block

C. Encryption Key

The key determines the initial FS value to be used in the created equations. Using FS value 8 (eight) in round 1 of the proposed technique, location 0 – 35 are occupied by the characters from the following locations as shown in Figure 4.

56	72	120	8	51	77	115	13	43	85	107	21	30	98	94	34
9	119	73	55	63	65	127	1	62	66	126	2	61	67	125	3
59	69	123	5												

Figure 4: Series of characters based on location using FS value 8

The series of locations identified produces the sequence of characters shown in Figure 5.

F	h	*	A	y	I	T	Q	M	L	U	4	M	B	Y	R
A	R	Z	5	g	M	*	I	x	7	*	l	o	M	*	C
l	g	*	W												

Figure 5: Series of characters in location 0 – 35 in Round 1

The sequence of characters for locations 36 – 127 is based on the arrangement shown in Figure 6. These characters are not moved from an array after performing the previous operations. Locations are arranged in ascending order.

w	M	W	I	B	K	B	g	C	q	B	O	I	S	u	s	W	o	5	Y
Q	H	7	r	q	N	a	n	a	c	W	L	w	b	R	g	o	I	J	4
i	a	U	i	A	J	K	I	n	P	A	2	o	B	0	O	Y	Y	R	5
V	/	s	l	s	p	j	v	e	A	L	J	D	I	v	l	1	e	X	j
1	1	6	I	G	O	n	z	Z	*	*	*								

Figure 6: Series of characters in location 36 – 127 in Round 1

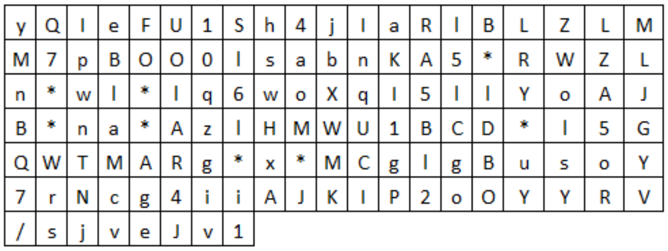
The sequence of characters for the subsequent 4 (four) rounds using the FS values 2, 55, 1 and 21 as key are shown in Figure 7 - Figure 10.

n	L	*	*	a	w	*	A	q	R	z	l	H	I	O	M	W	U	1	B
C	A	D	M	*	l	5	*	L	G	J	Q	a	W	*	h	F	y	T	Q
U	4	M	Y	R	A	R	Z	5	g	*	l	x	7	*	I	o	M	C	l
g	w	M	W	I	B	K	B	g	q	B	O	I	S	u	s	o	5	Y	7
r	N	c	b	g	o	4	i	a	i	A	J	K	I	n	P	2	o	B	0
O	Y	Y	R	V	/	s	s	p	j	v	e	A	L	J	l	v	I	1	e
X	j	1	6	I	l	n	Z												

Figure 7: Series Characters after Round 2

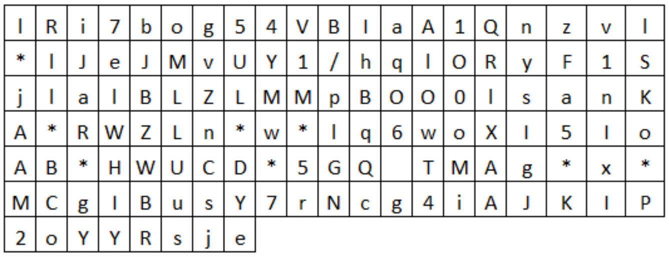
R	e	S	I	W	B	Z	L	M	K	n	*	w	B	I	*	l	q	6	w
o	l	X	q	I	5	l	l	Y	o	s	A	J	B	n	*	n	a	*	A
z	I	H	O	M	W	U	1	B	C	D	M	*	I	5	*	L	G	Q	a
W	h	F	y	T	Q	U	4	M	R	A	R	Z	5	g	*	x	7	*	M
C	g	I	g	B	O	u	s	o	Y	7	r	N	c	b	g	4	i	a	i
A	J	K	I	P	2	o	0	O	Y	Y	R	V	/	s	p	j	v	e	A
L	J	v	I	1	j	1	l												

Figure 8: Series Characters after performing Round 3



y	Q	I	e	F	U	1	S	h	4	j	I	a	R	l	B	L	Z	L	M
M	7	p	B	O	O	0	l	s	a	b	n	K	A	5	*	R	W	Z	L
n	*	w	l	*	l	q	6	w	o	X	q	I	5	l	l	Y	o	A	J
B	*	n	a	*	A	z	l	H	M	W	U	1	B	C	D	*	l	5	G
Q	W	T	M	A	R	g	*	x	*	M	C	g	l	g	B	u	s	o	Y
7	r	N	c	g	4	i	i	A	J	K	I	P	2	o	O	Y	Y	R	V
/	s	j	v	e	J	v	1												

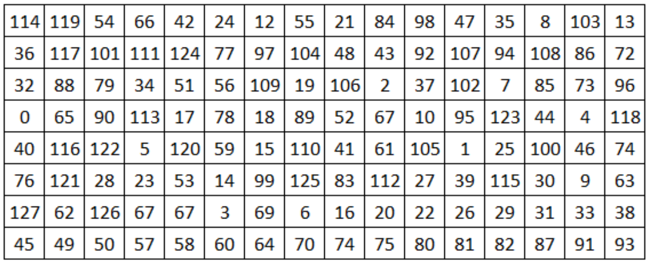
Figure 9: Series Characters after performing Round 4



I	R	i	7	b	o	g	5	4	V	B	I	a	A	1	Q	n	z	v	l
*	l	J	e	J	M	v	U	Y	1	/	h	q	l	O	R	y	F	1	S
j	l	a	l	B	L	Z	L	M	M	p	B	O	O	0	l	s	a	n	K
A	*	R	W	Z	L	n	*	w	*	l	q	6	w	o	X	I	5	I	o
A	B	*	H	W	U	C	D	*	5	G	Q		T	M	A	g	*	x	*
M	C	g	I	B	u	s	Y	7	r	N	c	g	4	i	A	J	K	I	P
2	o	Y	Y	R	s	j	e												

Figure 10: Series Characters after performing Round 5

Figure 11 shows the position of the characters after performing Round 5. Numbers based on the initial character location before the position shifting process. This shows that the character changed its position irregularly.



114	119	54	66	42	24	12	55	21	84	98	47	35	8	103	13
36	117	101	111	124	77	97	104	48	43	92	107	94	108	86	72
32	88	79	34	51	56	109	19	106	2	37	102	7	85	73	96
0	65	90	113	17	78	18	89	52	67	10	95	123	44	4	118
40	116	122	5	120	59	15	110	41	61	105	1	25	100	46	74
76	121	28	23	53	14	99	125	83	112	27	39	115	30	9	63
127	62	126	67	67	3	69	6	16	20	22	26	29	31	33	38
45	49	50	57	58	60	64	70	74	75	80	81	82	87	91	93

Figure 11: The new position occupied by characters after Round 5

A. *Decryption Process*

The decryption process recovers the plaintext private key block. The process is performed in reversed order using the same key which used during encryption.

4. TESTING

Confusion is designed to hide the relationship between the plaintext and the ciphertext. The confusion effect of the block cipher can be determined by the correlation values.

To determine the security of the proposed algorithm, the Pearson correlation coefficient was used:

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\ \sqrt{n\sum y_i^2 - (\sum y_i)^2}} \qquad (9)$$

where

r_{xy} = Pearson r correlation coefficient between x and y

n = number of observations

x_i = value of x (for ith observation)

y_i = value of y (for ith observation)

The correlation coefficient measures the degree of the linear relationship between two variables [11] [12]. It is a value between -1 and 1. The correlation is 1 in an increasing linear relationship, -1 in a decreasing linear relationship and some values between all other cases, indicating the degree of linear dependence between the variables. If the variables are independent, the correlation value is 0. Determining the strength of the correlation relationship are based on the following ranges:

Table II: Correlation Value and Correlative Degree

Correlation Coefficient	Correlative Degree
0	Non-linear relationship
0.7 – 1.0	Strong
0.3 – 0.7	Moderate
0.1 – 0.3	Weak
0.0	None

5. RESULT AND DISCUSSION

The proposed algorithm rearranges the plaintext private key characters based on the Fibonacci series values and the defined equation. As shown in the final character sequence, the 8 padded key characters mixed well with other characters after the last round.

The security performance of the proposed technique in terms of correlation coefficient was done to determine the dependency of the plaintext and ciphertext. The original location of the plaintext is identified then compared with its location after the last round of transposition process. Then, the correlation values were computed using the Pearson Correlation Coefficient.

A total of 200 randomly selected keys were utilized to get the correlation values for the 128-character block. Based on the results shown in Figure 12, 118 values of correlation coefficient values are less than 0.1 and 82 values between 0.1 and 0.3. It shows that the proposed technique provides weak linear relationship between the plaintext and the ciphertext. This exhibits that changes in a character position will not always bring the same change on the other characters.

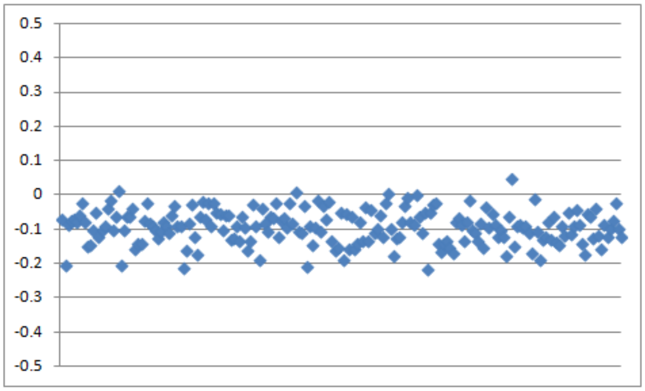


Figure 12: Results of Correlation Coefficient Test

6. RESULT AND DISCUSSION

This paper presents a practical and effective way of handling the private key in maintaining its confidentiality and integrity. The technique makes the key unusable by rearranging the sequence of private key characters while stored on a storage device. The correlation coefficient values were calculated to evaluate the security performance of the proposed transposition cipher technique. Pearson correlation coefficient was used to determine the coefficient values. The results showed that there is a weak linear relationship between the plaintext and the ciphertext.

For further studies, the proposed technique can be implemented in a system application to evaluate users' experience.

REFERENCES

1. A. M. Sison, B. T. Tanguilig, B. D. Gerardo, and Y. C. Byun. An improved data encryption standard to secure data using smart cards, in *Proc. - 2011 9th Int. Conf. Softw. Eng. Res. Manag. Appl. SERA 2011*, 2011, pp. 113–118. https://doi.org/10.1109/SERA.2011.27
2. L. K. Galla, V. S. Koganti, and N. Nuthalapati. Implementation of RSA, in *2016 Int. Conf. Control Instrum. Commun. Comput. Technol. ICCICCT 2016*, 2017, pp. 81–87. https://doi.org/10.1109/ICCICCT.2016.7987922
3. Z. J. Xiao, Z. T. Jiang, Y. Bin Wang, and H. Chen. Improved RSA Algorithm and Application in Digital Signature, in *Appl. Mech. Mater.*, 2015, vol. 713–715, pp. 1741–1745.
4. W. Yu *et al.* Protecting your own private key in cloud: Security, scalability and performance, in *2018 IEEE Conf. Commun. Netw. Secur. CNS 2018*, 2018.
5. V. Andrianova. Electronic signature key storage, in *Procedia Comput. Sci.*, vol. 145, 2018, pp. 471–480. https://doi.org/10.1016/j.procs.2018.11.010
6. C. Adams and G. V. Jourdan. Digital signatures for mobile users, in *Canadian Conference on Electrical and Computer Engineering*, 2014, pp. 1–5.
7. S. G. Chernyi, A. A. Ali, V. V Veselkov, I. L. Titov, and V. Y. Budnik. Security of Electronic Digital Signature in Maritime Industry, 2018, pp. 29–32.
8. T. H. L. Nguyen, Q. D. Tran, and T. H. Nguyen. A biometrics encryption key algorithm to protect private key in bioPKI based security system, in *ICICS 2009 - Conference Proceedings of the 7th International Conference on Information, Communications and Signal Processing*, 2009, pp. 5–9.
9. K. O. Elish, Y. Deng, D. D. Yao, and D. Kafura. Device-based isolation for securing cryptographic keys, in *Procedia Computer Science*, 2013, vol. 19, pp. 1130–1135. https://doi.org/10.1016/j.procs.2013.06.160
10. H. Rezaeighaleh, R. Laurens, and C. C. Zou. Secure Smart Card Signing with Time-based Digital Signature, in *2018 International Conference on Computing, Networking and Communications, ICNC 2018*, 2018, pp. 182–187. https://doi.org/10.1109/ICCNC.2018.8390321
11. V. Iranmanesh, S. M. S. Ahmad, W. A. Wan Adnan, F. L. Malallah, and S. Yussof. Online signature verification using neural network and pearson correlation features, in *2013 IEEE Conference on Open Systems, ICOS 2013*, 2013, pp. 18–21. https://doi.org/10.1109/ICOS.2013.6735040
12. A. Alabaichi, F. Ahmad, and R. Mahmod. Security analysis of blowfish algorithm, in *2nd International Conference on Informatics and Applications, ICIA 2013*, 2013, pp. 12–18. https://doi.org/10.1109/ICoIA.2013.6650222