Securing Private Key using New Transposition Cipher Technique

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

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

Abstract

The security of any public key cryptosystem depends on the private key thus, it is important that only an authorized person can have access to the private key. The paper presents a new algorithm that protects the private key using the transposition cipher technique. The performance of the proposed technique is evaluated by applying it in the RSA algorithm's generated private keys using 512-bit, 1024-bit, and 2048-bit, respectively. The result shows that the technique is practical and efficient in securing private keys while in storage as it produced high avalanche effect.

Key words: cryptography, transposition cipher, private key

Introduction

The Internet allows major transactions that cause significant increase in various data being processed and managed online. Transmitting sensitive data on the internet poses security threats thus, it is essential for an individual and businesses to search for ways to maintain the confidentiality and integrity of data during communication and those data in storage as well. Likewise, it is necessary to encrypt data before using it in communication [1].

Cryptography is one of the popular approaches that provide data security that involves encryption and decryption processes[2]. Encryption is the process of converting the information into an unrecognizable form, the ciphertext [3] while decryption is the reverse process that will recover the original message. The plaintext is the original message and the coded message is the ciphertext. [4] Only the intended recipient will be able to read or decrypt the file.

There are two types of cryptography: the symmetric and asymmetric key cryptography. Symmetric Cryptography performs encryption and decryption using the same key [5][6]. Asymmetric or public key cryptography involves two mathematically linked keys: the public key and the private key [7]. The sender uses the receiver's public key for encryption and uses his private key for decryption. The public key can be disclosed to the public or maybe advertised as widely as the user wants while private key has to be kept secret.

The security of any public key cryptosystem depends on the private key [8] which is the most vulnerable component of the system. However, the private key is long and difficult for a human to remember [9] and keeping it in different places increases the possibility that it may be compromised [10].

RSA is an asymmetric cryptographic algorithm named after Rivest, Shamir and Adleman. It is considered as the most popular public cryptosystem for digital signature [11][12].

Once the private key is compromised, it is possible for an adversary to create an authenticated digital signature on any electronic document on a user's behalf [10] or decrypt confidential data. Thus, it is important to ensure that only an authorized person can have access to the private key.

Biometrics [13] and Smart cards [14] are used to protect keys. Such methods require a specialized reader that incurs additional cost on the existing infrastructure of the organization or business. Essentially, using cryptography is a practical solution to protect the private key while in storage. The transposition cipher is a classical cryptographic algorithm that rearranges the order of characters [15] to some fixed permutation.

This paper presents a new transposition cipher technique to secure the private key by encrypting it before storing it in the storage device.

Methodology

The design of the proposed transposition cipher technique aims to protect one of the most sensitive data, the private key. It involves the encryption and decryption process.

Encryption Process

This process converts the private key plaintext to its unusable format.

1. Plaintext Block

The private key plaintext is divided into blocks with 120 characters each. An additional 8 (eight) random characters are padded to create a full block that serves as an initial input in the encryption process.

Fig.1 shows the sample block with 128 characters. A series of 120 characters are taken from the generated private key using the RSA algorithm.

2. Encryption Key

The technique uses the Fibonacci Series (FS) to plan the position shifting of characters in each round. The algorithm uses the FS value: 1, 2, 3, 5, 8, 13, 21, 34, and 55. This serves as the key in every round of the encryption process. The key determines the initial value for character position shifting on the block.

Shown in Fig. 2 are the steps involve in encrypting the private key blocks. The full block, composed of 128 characters, is transformed into the ciphertext by performing the transposition cipher technique. To make the position shifting more irregular, the technique is performed for 5 (five) rounds. In each round, as shown in the figure, the key is required.

ISBN: 978-1-7281-2501-5

*	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
^	M	I	I	C	X	Q	I	B	A	A	K	B	g	Q	D	E

*	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
^	i	b	Z	x	3	s	S	P	2	j	k	f	a	w	x	H

	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
	i	I	6	M	C	K	Q	e	9	i	Q	4	9	w	K	/

*	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
^	5	o	a	R	+	u	k	V	S	V	q	R	3	n	J	1

*	64	65	66	67	68	69	70	71	108	109	110	111	112	113	114	115
^	F	4	F	W	i	G	5	3	I	R	O	V	V	N	V	L

*	116	117	118	119	120	121	122	123	124	125	126	127
^	f	t	v	k	*	*	*	*	*	*	*	*

*character location
^plaintext characters of the private key

Fig. 1 Sample Block with 128 characters

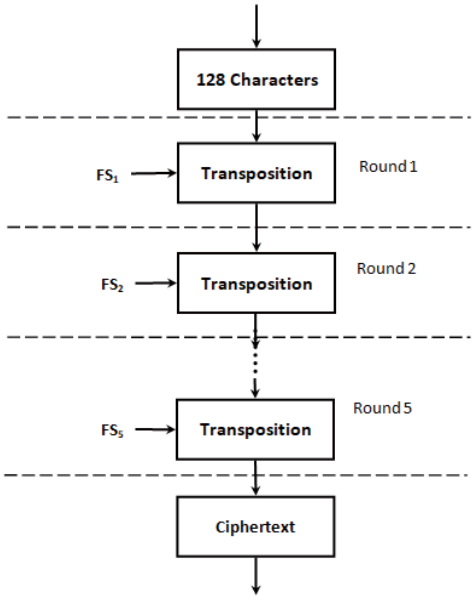

Fig. 2 Encryption process for each block

2. Position Shifting Map

The (1) – (4) use the following values: midValue = 64, endValue = 128 and startValue = 0. CT for ciphertext, PT for plaintext. The loc is initialized to 0; the first location on the array to be occupied by the characters during the position shifting process.

For every FS value, the following equation is applied:

$$CT[loc{+}{+}] = PT[midValue - FS] \quad (1)$$

$$CT[loc{+}{+}] = PT[midValue + FS] \quad (2)$$

$$CT[loc{+}{+}] = PT[endValue - FS] \quad (3)$$

$$CT[loc{+}{+}] = PT[startValue - FS] \quad (4)$$

Shown in Table 1 are the allowed FS value in the algorithm and the derived value from the equations. The FS value is wrapped around until all values have been employed in the position mapping.

TABLE 1
VALUES DERIVED BASED ON FS VALUE

FS Value	Values based on equations			
	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

Based on the location numbered 0-127, using key value 5 (five) produces the series of characters as shown in Fig. 3. After the values were derived from FS Value 55, it uses the values from FS values 1, 2 and 3 to complete the position shifting of the 36 characters.

59	69	123	5	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				

Fig. 3 Series of characters using FS value 5

The character locations 36 – 127 are arranged as shown in Fig. 4. These are the remaining character locations where the contents are not utilized. Numbers are arranged in ascending order.

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													

Fig.4 Character sequence after using the identified FS value series

Shown in Figure 5 is the result of the entire encryption process, encrypted blocks are concatenated to produce the final ciphertext. The private key stored in the storage media remains unusable unless decrypted using the user's valid key.

ISBN: 978-1-7281-2501-5

Fig. 5 Sample Encrypted Private Key

Decryption Process

The decryption process recovers the private key plaintext before using it to sign a document or decrypt a file.

1. Ciphertext Block

The encrypted private key is divided into 128 characters per block.

2. Decryption Key

The key used on the encryption key has to match with the decryption key. The key is used to perform the transposition cipher technique to be able to come up with the original private key plaintext.

2. Position Shifting Map

Position shifting is based on (5) – (8) that use the following values: midValue = 64, endValue = 128, and startValue =0. CT for ciphertext, PT for plaintext. The loc, which refers to the character location, is initialized to 0; the location on the array being occupied by the private key ciphertext characters.

Character position is identified by performing the following equations to recover the private key block plaintext:

$$PT[midValue - FS] = CT[loc{+}{+}] \quad (5)$$

$$PT[midValue + FS] = CT[loc{+}{+}] \quad (6)$$

$$PT[endValue - FS] = CT[loc{+}{+}] \quad (7)$$

$$PT[startValue + FS] = CT[loc{+}{+}] \quad (8)$$

3. Extracting the padded characters

After rearranging of 128 characters for 5 (five) rounds, the last 8 characters on each block is removed. Blocks are then placed together to recover the user's private key plaintext.

Results and Discussion

Applying the algorithm to rearrange the series of 128 characters is shown in Fig. 1. The FS value 21, 2, 13, 5, and 34 is used as the encryption key. Fig. 6 – Fig. 10 show the result of character position shifting based on the created equations. The * (asterisk) character, which are presented in red font, represents the 8 padded characters.

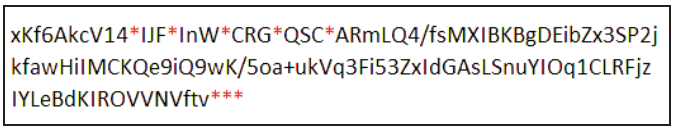

Fig. 6 Series of characters using FS value 21 as the key

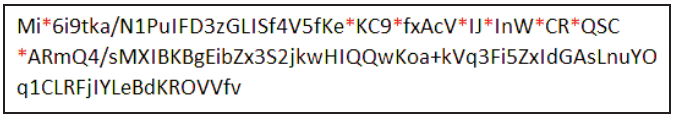

Fig. 7 Series of characters using FS value 2 as the key

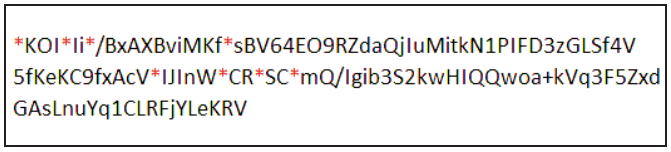

Fig. 8 Series of characters using FS value 13 as the key

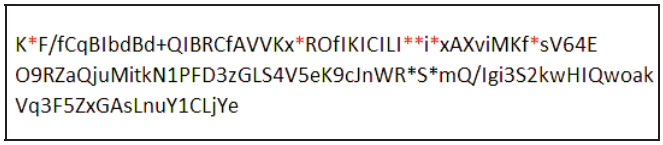

Fig. 9 Series of characters using FS value 5 as the key

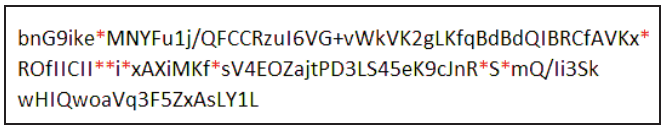

Fig. 10 Series of characters using FS value 34 as the key

The final result of the encryption process using the transposition cipher technique is shown in Fig. 10. The padded characters, represented by an asterisk mixed well with the plaintext characters. After performing the decryption technique, the result showed that the original block of the private key plaintext was recovered.

The test was done to determine the accuracy and security of the proposed algorithm in terms of the avalanche effect. The test covers the encryption and decryption process applied in 512-bit, 1024-bit and 2048-bit generated RSA private keys.

Table 2 shows the count of private key characters, the number of blocks, and the padded characters required to perform the encryption process. The result of the test conducted proved the accuracy of the algorithm. The encrypted private keys were reverted to its original form.

.

TABLE 2
TESTING RESULTS

RSA Algorithm	Total No. of characters	No. of blocks	No. of chars padded to the last block	Total Padded characters	Result
512-bit	512	4	60	88	Accurate
512-bit	512	4	54	84	Accurate
1024-bit	896	7	40	88	Accurate
1024-bit	896	7	36	84	Accurate
1024-bit	896	7	32	80	Accurate
2048-bit	1792	14	96	200	Accurate
2048-bit	1792	14	92	200	Accurate

ISBN: 978-1-7281-2501-5

The security of the proposed algorithm was determined by calculating the Avalanche Effect. It is the desirable property of an encryption algorithm in which a small change in either the plaintext or key should produce a significant change in the ciphertext. Table 3 shows that the proposed algorithm exhibits strong avalanche effect.

TABLE 3
RESULT OF AVALANCHE EFFECT

Private Key	Key	Avalanche Effect
512-bit	75824 56241 13652 64735 1	74.80%
	75824 56241 13652 64735 2	
1024-bit	24652 25613 76832 56734 71422 83561 76425 1	84.59%
	24652 25613 76832 56734 71422 83561 76425 2	
2048-bit	22417 52624 43567 56183 42675 61325 83267 64257 65138 42712 73465 83762 65213 52642 1	70.03%
	22417 52624 43567 56183 42675 61325 83267 64257 65138 42712 73465 83762 65213 52642 2	

Conclusion and Recommendation

This paper presented a new transposition cipher technique to secure the private key while in removable storage. It will be practical and economical to implement such since no additional hardware is required and it can help to maintain the integrity and confidentiality of the private keys.

The evaluation showed that the algorithm produced the expected results; the recovered plaintext of the private key was the same with the original plaintext before encryption and the avalanche effect was high.

Other technique to assess the algorithm is recommended to further determine its security.

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," *Proc. - 2011 9th Int. Conf. Softw. Eng. Res. Manag. Appl. SERA 2011*, pp. 113–118, 2011.

[2] M. Thangavel, P. Varalakshmi, M. Murrali, and K. Nithya, "An Enhanced and Secured RSA Key Generation Scheme (ESRKGS)," vol. 0, pp. 3–10, 2014.

[3] U. Bhargava, A. Sharma, R. Chawla, and P. Thakral, "A new algorithm combining substitution & transposition cipher techniques for secure communication," *Proc. - Int. Conf. Trends Electron. Informatics, ICEI 2017*, vol. 2018-Janua, pp. 619–624, 2018.

[4] S. S. Gupta, G. Dhanjal, S. Bambardekar, and P. Vartak, "Data Security using Compressed Classical Technique," *2018 Int. Conf. Smart City Emerg. Technol. ICSCET 2018*, pp. 1–5, 2018.

[5] R. Kaur and A. Kaur, "Digital Signature," *Int. Conf. Comput. Sci.*, 2012.

[6] S. Chandra, S. Bhattacharyya, S. Paira, and S. S. Alam, "A study and analysis on symmetric cryptography," *2014 Int. Conf. Sci. Eng. Manag. Res. ICSEMR 2014*, 2015.

[7] S. A. Jaju and S. S. Chowhan, "A Modified RSA algorithm to enhance security for digital signature," *2015 Int. Conf. Work. Comput. Commun. IEMCON 2015*, 2015.

[8] W. Yu *et al.*, "Protecting Your Own Private Key in Cloud : Security , Scalability and Performance," 2018.

[9] C. Adams and G. Jourdan, "Digital Signatures for Mobile Users," pp. 1–5, 2014.

[10] S. G. Chernyi, A. A. Ali, V. V Veselkov, I. L. Titov, and V. Y. Budnik, "Security of Electronic Digital Signature in Maritime Industry," pp. 29–32, 2018.

[11] L. K. Galla, V. S. Koganti, and N. Nuthalapati, "Implementation of RSA," *2016 Int. Conf. Control Instrum. Commun. Comput. Technol. ICCICCT 2016*, pp. 81–87, 2017.

[12] Z. J. Xiao, Z. T. Jiang, Y. Bin Wang, and H. Chen, "Improved RSA Algorithm and Application in Digital Signature," *Appl. Mech. Mater.*, vol. 713–715, pp. 1741–1745, 2015.

[13] N. Thi and H. Lan, "A Biometrics Encryption Key Algorithm to Protect Private Key in BioPKI Based Security System," pp. 5–9, 2009.

[14] K. O. Elish, Y. Deng, D. D. Yao, and D. Kafura, "Device-Based Isolation for Securing Cryptographic Keys," vol. 19, pp. 1130–1135, 2013.

[15] M. Sokouti, B. Sokouti, and S. Pashazadeh, "An approach in improving transposition cipher system," *Indian J. Sci. Technol.*, vol. 2, no. 8, pp. 8–15, 2009.

ISBN: 978-1-7281-2501-5