MIS Review Vol. 20, No. 2, March (2015), pp. 57-76
DOI: 10.6131/MISR.2015.2002.03

Implementation of N-Cryptographic Multilevel Cryptography Using RSA and Substitution Cryptosystem

Olawale S. Adebayo, Morufu Olalere, Joel N. Ugwu
Department of Cyber Security Science, Federal University of Technology, Nigeria

ABSTRACT: *The purpose of cryptography is to ensure information is made in such a way that an unintended individual will not have access to it or does not understand what it means when intercepted on a communication network. Some people try to defeat this purpose by using an extra ordinary means to harm the algorithmic construct of the system. The effort required for this purpose depends on the complexity of the algorithm and the number of cryptographic-ciphers used. Given that effort required to cryptanalyze ciphertext when one-algorithmic transformation was made is x-effort, then the effort required when n-algorithmic transformation was done is nx-effort. This paper implements multilevel encryption algorithm using two cryptosystem; RSA and Substitution cryptosystem with one transformation per each. It presents an algorithmic paradigm which can be implemented using any programming language. It simplifies the stages used for both encryption and decryption, presenting each stage in a sequential order.*

KEYWORDS: *Cryptography, n-Cryptographic, Ciphertext, Cryptosystem, Cryptanalysis, Multilevel Encryption, RSA Cryptosystem, Substitution Cryptosystem.*

1. Introduction

The effect of information attacks and their exposure on the present day communication media has been alarming. Much research work has been devoted to this area, while people continue to work against the progress of this effort thereby making the field of information security an ever more demanding research area (Ugwu, 2014). Many algorithms have been developed, while many are still in the pipeline, as it is a highly dynamic area. Using multiple transformations could be seen as a better option compared to a single one thereby making the process a multiple transformed output. This multiple transformation could either be of the same algorithm with the same key, or of the same algorithm with different keys, or even different algorithms. As multiple transformed ciphertext will require multiple computational efforts, it is clear that it requires more effort and time to be cryptanalyzed; hence making the ciphertext better secured.

While the security of a multilevel process depends on the security of its component

algorithms, it is better practice to use either the same algorithm with different keys or different algorithms for multilevel processes. The choice of which strategy to use depends on the individual and on the application purpose. The RSA cryptosystem together with Substitution cryptosystem is adopted in this research to realize a given multilevel algorithm, presenting all steps involved in the process, and also simplifying the construct in such a manner that it can easily be implemented using any programming language.

The RSA cryptosystem is made in such a way that it cannot be easily factored; this is done by using a high value prime number during the computation and selection of the exponent. This should be done as if it were to be used alone and after which the substitution operation is used against its output (Bruce, 1996). The substitution table should be an agreement within both ends, if it were not to be implemented within an application interface; both ends should have the knowledge of the substitution table as it will be used for reverse computation by each other when a message is received from another individual that has an RSA public key of the receiver.

Just as algorithms are essential in securing information prior to dispatching the documents on the communication network (Adebayo et al., 2012), the sequence of the algorithmic applications also matters a lot, since it determines how its encryption and decryption transformation will occur. The process that comes first during encryption will often come last during decryption. The transformation process occurs in reverse manner, to realize the original clear text at the other end, but it all depends on the agreement between the parties involved or the implementation used.

2. Literature review

Harn and Lin (1990) proposed a key generation scheme for multilevel data security using bottom-up approach. The term multilevel was used to mean variable securities at different access levels with many users of a single system having different keys at each different access level. This approach was formed modifying the approach proposed by Akl and Taylor (1982) using a top-down model. Usha Devi and Wahida Banu (2012) proposed a multilevel encryption-decryption of text into cipher data in which its characters are encoded uniquely into its corresponding cipher and eliminating the possibility of any pattern as described in their paper titled “Secure Multilevel Cryptography Using Graceful Codes.” It uses more than one level of security by employing many ciphers to disguise any pattern.

Gawande et al. (2012) introduced the culture of securing images using chaotic

mapping and elliptic curve cryptography in a network environment. The dependency of stream ciphers on pseudo-stochastic sequences was noted as it can produce a pseudo-random sequence with good randomness. Hardjono and Sebbery (1989) discovered a system that makes use of hierarchical keys used to encrypt and decrypt data stored in databases using the RSA cryptosystem with additional restriction of encrypted information to the public. The base of the systems security is discrete logarithms and the term "multilevel" used in this context means multiple users with different securities.

"Multi-Level Crypto Disk: A Secondary Storage with Improved Performance" was introduced by Chaitanya et al. (2006). They discussed the issue of hard disks becoming increasingly vulnerable to security attacks as they are now accessed remotely, either with mobile devices or in other unanticipated operating environments. They highlighted the demerits of using single data encryption on storage devices, proposing a secure disk using multiple crypto levels. "Multi-Level Cryptographic Functions for the Functionalities of Open Database System" was designed and implemented by Adio et al. (2011). This is a secure open database system for an organization that can open their information system for access by different users. The implementation does not require input to be hidden from anyone or converted to place holder characters for security reasons, but the user only needs to study the sequence of codes and active boxes that describe his password and uses it in place of his active boxes.

A secure information transmission using Multilevel Steganography and Dynamic Cryptography was proposed by Sikarwar (2012) in his paper titled "An Integrated Synchronized Protocol for Secure Information Transmission Derived from Multilevel Steganography and Dynamic Cryptography." He juxtaposed the use of both simple steganography and cryptography proposing that multiple and dynamic codes give more security. Maruti and Subhash (2009) present a practical implementation of a quasigroup based multilevel encryption for data and speech. It makes use of an indexed scrambling transformation for signal authentication, encryption, and broadcasting applications in secrete-key cryptography. The results presented shows that a quasigroup transformation is very effective in destroying the structure of the input signal, and hence can be a good encryption technique.

2.1 RSA cryptosystem

RSA is a cryptosystem named after the founders' initials; Ronald Rivest, Adi Shamir, and Leonard Adleman in the 1970s (Judy, 2002). Its method of application and the computations of its keys as well as the encryption and decryption procedures are presented with examples below:

2.2 Key generation

2.2.1 Step i. Primes selection

Two numbers that are co-prime are selected randomly.

Say P and Q are chosen and their product **(P × Q)** is computed to be **"n."** This value is kept to be used as a modulus for encryption and decryption of plaintext and ciphertext respectively.

2.2.2 Step ii. Euler-Totient computation, $\phi(n)$

The prime numbers chosen in **2.2.1** above are used for the computation of Totient Function. "1" is subtracted from each of the primes as **(P – 1)** and **(Q – 1)** and their product is determined. This value **((P – 1) × (Q – 1))** is used for the determination of the encryption key and the decryption key as well.

2.2.3 Step iii. Totient co-prime selection

After the Totient function has been computed, we determine the Lowest Common Multiple LCM of the Totient value in order to know the list of values that are relative primes to the Totient value.

From these values, a number is chosen to be used as an encryption key. Let this chosen value be denoted as **e**. The Greatest Common Divisor (gcd) of Totient function, $\mathbf{gcd(e, \phi(n)) = 1}$.

2.2.4 Step iv. Totient co-prime inverse computation

An inverse of **e**, $\mathbf{e^{-1}}$ is computed using Euclidean algorithm as described below. The Extended Euclidean algorithm is used to determine the value $\mathbf{e * d + \phi(n) * y = 1}$.

Where $\mathbf{d = e^{-1}}$ (Ugwu, 2014).

2.3 Encryption and decryption

RSA encryption involves the transformation of plain information using a private key into ciphertext in such a manner that the message on arrival at the destination will be retransformed back to the original plaintext using the public key of the Sender by the Receiver (Ugwu, 2014). Someone can equally write to the owner of an RSA key using his public key that is published to the public. The message will be retransformed to the plain information by the owner using his private key. The plain message has to be transformed to the numerical equivalent, which then allows the enciphering function to be applied in other to get the required numerical equivalent of the ciphertext, and thus substituted with the ciphertext symbol equivalent to get the ciphertext. For instance,

Let **m** be the plaintext message

Let **f** be the enciphering function and $\mathbf{f^{-1}}$ be the deciphering function.

Let **c** be the ciphertext

To encrypt a message using RSA algorithm, the sender computes the following (Simmons, 1979):

$$\mathbf{c = f(m) = m^e} \tag{1}$$

After this computation, the ciphertext, **c** will be sent along the unsecure medium to the destination.

While to decrypt the ciphertext, the receiver will first map the numerical equivalent of the ciphertext before the application of deciphering function, as follows:

$$\mathbf{m = f^{-1}(c) = c^d} \tag{2}$$

3. Implementation of RSA cryptosystem

3.1 Key generation

We shall exemplify the implementation of the RSA cryptosystem using small prime integer values in order to depict what exactly happens within the system. The value shall be small in order to permit the computation of the encrypting and decrypting functions using a scientific calculator, otherwise it will not be computable (Ugwu, 2014).

Let **P = 11**, and **Q = 5**,

Computing the value of **n**,

$\mathbf{n = 11 \times 5}$

$\mathbf{= 55}$

Computing the Euler Totient Function $\boldsymbol{\phi(n)}$,

$\boldsymbol{\phi(n) = (11 - 1) \times (5 - 1)}$

$\mathbf{= 10 \times 4}$

$\mathbf{= 40}$

Selecting the encryption exponent, **e**:

Let **e = 7**;

Checking whether the $\boldsymbol{\phi(n)}$ and **e** are relatively prime

Finding the LCM of **40** and **7**

LCM of $40 = 2^3 \times 5^1$ and LCM of $7 = 7^1$

From the computation above, you could understand that there is no similarity with the components that make up **ϕ(n)** and **e**,

gcd(e, ϕ(n)) = 1

Hence, they are relatively prime to each other.

To compute the value of **d**, we shall invoke the use of extended Euclidian algorithm, which shall accurately give us the value of the inverse function of **e**, **(e^{-1})**

40 = 7 × 5 + 5

7 = 5 × 1 + 2

5 = 2 × 2 + 1

2 = 1 × 2 + 0

Where the computations are in the order of:

Dividend = Divisor × Quotient + Remainder,

For the Iterating values, the former Divisor will become the new Dividend, as the former Remainder will become the new Divisor, this continues until the remainder becomes **0**,

Then, we work back to get the extended Euclidian value,

1 = 5 – (2 × 2)

= 5 – 2 (7 – 5)

= 5 – 2 × 7 + 2 × 5

= 3 × 5 – 7 × 2

= 3 (40 – (7 × 5)) – 7 × 2

= 40 × 3 – 7 × 17, therefore

1 = 40 × 3 + 7 × (-17)

Hence the inverse function of the **e** value is:

-17 (mod**40)**

= 40 – 17

= 23 (mod**40)**

d = 23.

The private keys which are **n**, **ϕ(n)**, **P**, **Q**, **e**, and **d** are:

55, 40, 11, 5, 7, 23

The public keys which are **n** and **e** are:

55 and **7**.

3.2 Encryption and decryption

If one wants to send a message ABA to a friend living at a far distance, but would want to encrypt it manually using the above keys as already computed.

Taking the English alphabetical-number equivalent as written in Table 1:

Table 1 Alphabetical Equivalence Table

,	.	A	B	C	D	E	F	G	H	I	J	K	L
0	1	2	3	4	5	6	7	8	9	10	11	12	13
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25	26	27
+	-	*	/	\	]	[	;	:	}	{	=	_	)
28	29	30	31	32	33	34	35	36	37	38	39	40	41
(	*	&	^	%	$	#	@	!	~	`	<	>	“
42	43	44	45	46	47	48	49	50	51	52	53	54	55

m = A B A, so, from the alphabetical equivalence table above,

m = 2 3 2

The encryption function as stated in Equation (1) above is $\mathbf{c = m^e}$ **(**mod**n)**

Where **c** = the ciphertext message; **m** = the plaintext message; **e** = the encrypting exponent; **n** = the modulus.

$\mathbf{c = 2^7\ 3^7\ 2^7}$ all in mod**55**

c = 128 (mod**55) 2187 (**mod**55) 128 (**mod**55)**

c = 18 42 18

From the alphabetical equivalence table, you could infer that:

18 = Q, and **42 = (**

Then the sender will now send**:**

c = Q (Q to the destination owner of the public key).

The owner of the public key will decrypt the ciphertext using the private keys, **n** and **d** as stated in Equation (2) above.

$\mathbf{m = c^d}$ **(**mod**n)**

$\mathbf{m = 18^{23}\ 42^{23}\ 18^{23}}$ all in mod **55**

m = 74347713614021927913318776832 (mod**55)**

2.16139269415798008294225812728 45e + 37 (mod**55)**

74347713614021927913318776832 (mod**55)**

m = 2 3 2

The receiver will look back from the alphabetical equivalence table to get their values, hence

m = A B A

The receiver, and owner of the key will also understand that the message was meant for him.

Therefore, **c = 8 27 8**

$\mathbf{m = 8^7\ 27^7\ 8^7}$ all in mod**55**

m = 2097152 (mod**55) 10460353203 (**mod**55) 2097152 (**mod**55)**

m = 2 3 2

From the alphabetical table equivalence:

2 = A, and **3 = B**.

Hence the message sent is:

ABA

3.3 Substitution cryptosystem

The substitution cryptosystem involves the use of substitution encryption and decryption table to encrypt and decrypt a message based on its equivalences. We shall exemplify what happens within the Multilevel Offline Cryptography Support System using Table 2 (Simmons, 1979):

Table 2 Substitution Cryptosystem Table

Pi	.	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ci	#	+	-	Z	Y	*	/	!	~	W	X	\	$	%	^	A	C	B	(	{	)	<	}	>	D	F	E
Pi	-	*	/	\	]	[	;	:	}	{	=	_	)	(	*	&	^	%	$	#	@	!	~	`	<	>	“
Ci	G	=	[	@	P	“	:	I	M	H	‘	O	V	U	S	R	J	K	.	]	|	Q	N	&	L	;	_

Where Pi indicates the Plaintext character and Ci indicates the Ciphertext character.

A substitution cipher involves direct changing of the plaintext messages with the ciphertext equivalent.

One might wish to encrypt and send a message joeljupiter@yahoo.com. Before sending this message, he or she must first of all take the direct ciphertext representation to compute the ciphertext from the substitution table before sending it. It is computed as below:

Plaintext = joeljupiter@yahoo.com

From the ciphertext table, the ciphertext equivalence is:

Ciphertext = xa*$x<cw)*(|f+~aa#za%

The ciphertext message above will now be sent to the receiver who then decrypts the message using the substitution cipher table.

4. Multilevel paradigm (RSA-substitution cryptosystem)

N-Cryptography Multilevel ciphers shown in Figure 1 combined the features of RSA and Substitution cipher. In the multilevel cryptographic example here, we will employ the use of RSA and Substitution cryptosystems which produces several layers in the specified prototype algorithm. The encryption layers are:

(1) The upper layer: this layer maps the plaintext letters to their numeric equivalents;

(2) The middle upper layer: this layer involve the transformation of the numeric values using the RSA encryption

(3) The middle lower layer: this layer takes the values from RSA transformation and subsequently maps numeric values to their letter equivalents, and

(4) The terminal layer: this layer involves taking the RSA ciphertext letters' equivalent values from the substitution table.

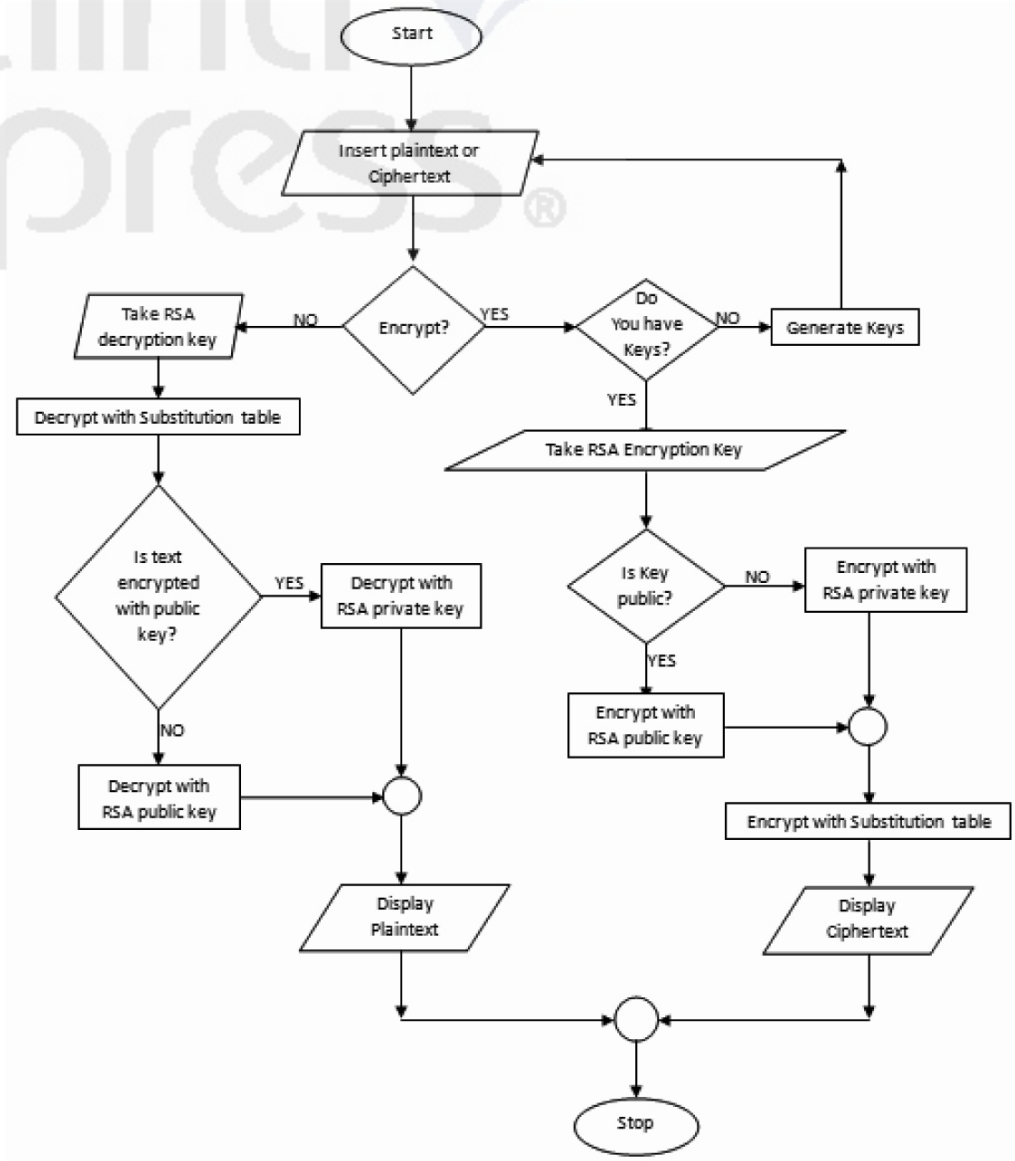

Figure 1 RSA and Substitution Cipher Multilevel Flowchart

For instance using the keys in 2.2 and Tables 1 and 2 specified above, one might wish to send a message "ABA" to a friend using the RSA and Substitution multilevel paradigm. The computation is done as follows:

The plaintext, **m** = **ABA**

From the RSA alphabetical-number equivalent in Table 1 above, the numbers equivalence are:

2 3 2

Encryption function as stated in Equation (1) above is $\mathbf{c = m^e (mod n)}$

Where **c** = the ciphertext message; **m** = the plaintext message; **e** = the encrypting exponent; **n** = the modulus.

m = 2 3 2

n = 55, **e = 7**,

$\mathbf{c = 2^7\ 3^7\ 2^7}$ all in mod**55**

c = 128 (mod**55) 2187 (**mod**55) 128 (**mod**55)**

c = 18 42 18

From the alphabetical-number equivalence table above, you could infer that:

18 = Q, and **42 = (**

Then the sender will now have the RSA ciphertext as:

c = Q(Q

The sender will also go to the substitution-encryption-decryption table above to take the ciphertext equivalence value of the RSA ciphertext as follows:

Q = B

(= U

Then the final ciphertext message will now become **BUB**

The sender will now send **c = BUB** to the destination of the message.

Decryption of ciphertext involves many layers, but is arranged as reverse of the encryption above. These layers are:

(1) The upper layer which involves the decryption of the substitution ciphertext using the substitution encryption-decryption table above to get the RSA ciphertext,

(2) The middle upper layer involves the mapping of the RSA ciphertext with the number equivalent using the letter-number equivalent table.

(3) The middle lower layer which involves the decryption of the numeric equivalence using RSA decryption formula

(4) The lower layer involves subsequent mapping of the numeric values with its letter equivalence from the RSA number-letter equivalence table.

We shall exemplify the steps above using the ciphertext obtained from the computation above.

The ciphertext is **BUB**

Converting the **BUB** to the substitution plaintext to get the RSA ciphertext using Table 2 above,

Pi = **BUB**

Ci = **Q(Q**

Then, the RSA ciphertext value **Q(Q** will be mapped with its alphabet-number equivalence in Table 1 above to get **18 42 18**

Using the formula $\mathbf{m = c^d (\text{mod} n)}$ from Equation (2) and the keys specified in 2.2 above.

$\mathbf{m = 18^{23}\ 42^{23}\ 18^{23}}$ all in mod **55**

$\mathbf{m = 74347713614021927913318776832\ (\text{mod}55)}$

$\mathbf{2.16139269415798008294225812728 45e + 37\ (\text{mod}55)}$

$\mathbf{74347713614021927913318776832\ (\text{mod}55)}$

m = 2 3 2

The receiver will look back from the number-alphabetical equivalence table in Table 1 above to get their values, hence

m = A B A

The receiver, which is the owner of the key, will also understand that the message was meant for him.

5. RSA and substitution ciphers multilevel algorithm

Start

//Input the plaintext message, m or ciphertext, c

//Is it to encrypt

If YES

//Request for key,

//Do you have keys

If YES

//Insert the encryption keys

//Is encryption key public

If yes

//Encrypt with public key

Else

//encrypt with private key

//Encrypt with Substitution table

//Display Ciphertext

Else,

//Generate Keys

//Move back "Insert plaintext or ciphertext"

Else,

//Insert the decryption keys

//Decrypt with Substitution table

//Is text Encrypted with public key?

If yes

//Decrypt with private key

Else

//Decrypt with public key

//Display the Plaintext

Stop

6. Performance evaluation

The researchers adopt ISO/IEC 27004 (Figure 2) (International Organization for Standardization and International Electrotechnical Commission, 2009) and the methodology proposed in NIST SP 800-55 (National Institute of Standards and Technology, 2000) in order to measure the performance of new RSA-Substitution

cryptosystem. The ISO/IEC 27004 identifies four major processes for quality assessment which are listed below:

(1) Measures and Measurement Development.

(2) Measurement Operation.

(3) Data Analysis and Measurement Results Reporting.

(4) ISMP Evaluation and Improvement.

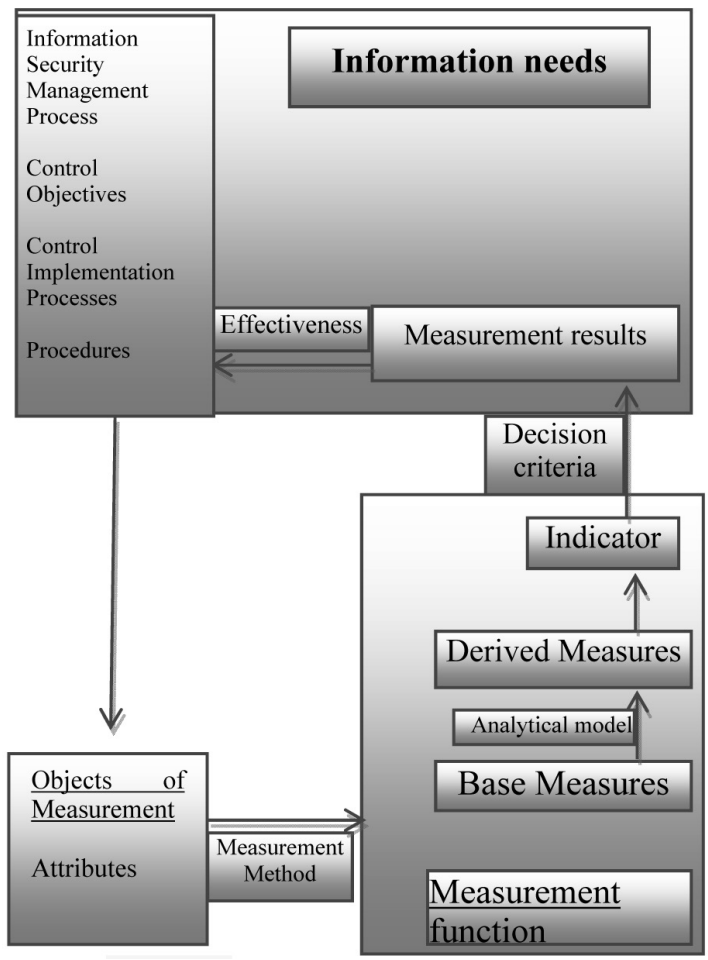

Figure 2 ISO/IEC 27004:2009 Model

6.1 Variable definition

Variable	Value
Robustness	
Very robust	1
Weak	0
Timeliness	
Fast	1
Slow	0
Operation result	
Maximal	1
Minimal	0
Goal & Objectives	
Achieved	1
Not achieved	0
Security standard, policies and procedures	
Observed	1
Not observed	0

6.2 Data analysis

Binary values have been assigned in Table 3 to make a preliminary analysis of the combined effect of the two ciphers. This shows the joint system has the robustness of the RSA system, the speed of a substitution cipher and an optimized operational result. A full ISO/IEC 27004 analysis would require baseline measurements on a practical implementation, followed by comparative measurements using the same methodology over time.

Table 3 Data Analysis of RSA, Substitution, and RSA-Substitution Cryptosystem

Algorithm	Robustness	Timeliness	Operation Result	Goals and Objectives	Implementation Level of Established Security Standard, Policies and Procedures
RSA	1	0	0	1	1
Substitution	0	1	0	1	1
RSA+ Substitution	1	1	1	1	1

7. Benefits of multilevel RSA-substitution cryptosystem

The implementation of RSA-Substitution Cryptosystem shows that it has the followings benefits:

(1) It is robust and highly secure than single cryptosystem.

(2) It is cost effective.

(3) It is efficient in terms of implementation processing time.

8. Future research

The researchers aim to measure and enhance the performance of the RSA-Substitution multilevel cryptosystem in order to improve its robustness and other benefits.

9. Conclusion

This research has successfully implemented a multilevel algorithm using the combination of RSA and Substitution ciphers. Implementing the substitution cipher with RSA encryption realized a high level of information security assurance as the RSA cipher will ensure the authenticity of the message. Implementing this algorithm in the application layer gives a great advantage to the computation of RSA keys because of the factorization of large prime numbers. In the overall implementation of the RSA cipher, using large primes could be seen as a great security additive and hence we are encouraged to use this. The multilevel paradigm presented above is very simple and can be implemented using any programming language. The output of the RSA transformation is taken as the input of the Substitution transformation, after which, the required ciphertext is obtained. The decryption process as indicated by the flowchart above can be seen as the reverse of the encryption process. The substitution transformation that was done last during the encryption process was first to be deciphered, while the RSA step was done last.

Acknowledgements

The researchers wish to acknowledge Dr Andrew Fluck of the Faculty of Education, University of Tasmania, Locked Bag 1307, Launceston, TAS 7250, Australia for editing this work.

References

Adebayo, O.S., Waziri, V.O., Ojeniyi, J.A., Bashir, S.A. and Mishra, A. (2012), 'Information security on the communication network in Nigeria based on digital signature', *International Journal of Computer Science & Information Security (IJCSIS)*, Vol. 10, No. 11, pp. 57-63.

Adio, A.T., Adekoya, A.F. and Oluwafemi, O.E. (2011), 'Multi-level cryptographic functions for the functionalities of open database system', *Computer Technology and Application*, Vol. 2, No. 9, pp. 730-735.

Akl, S.G. and Taylor, P.D. (1982), 'Cryptographic solution to a multilevel security problem', in Chaum, D. (Ed.), *Advances in Cryptology: Proceedings of Crypto 82*, Springer Science & Business Media, Santa Barbara, CA, pp. 237-250.

Bruce, S. (1996), *Applied Cryptography. Protocols, Algorithms, and Source Code in C*, Wiley, New York, NY.

Chaitanya, S., Urgaonkar, B. and Sivasubramaniam, A. (2006), 'Multi-level crypto disk: secondary storage with improved performance vs security trade-offs', Technical Report CSE-09-006, Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA.

Gawande, R.K., Kulkani, P.S. and Ganar, K.A. (2012), 'Multi level image encryption using chaotic mapping and elliptic curve cryptography', *Proceedings of International Conference of Engineering Innovation and Technology*, Nagpur, India, pp. 69-73.

Hardjono, T. and Seberry, J. (1989), 'A multilevel encryption scheme for database security Department of Computer Science', *Proceedings of the 12th Australasian Computer Science Conference (ACSC)*, Wollongong, Australia, pp. 209-218.

Harn, L. and Lin, H. (1990), 'A cryptographic key generation scheme for multilevel data security', *Computers & Security*, Vol. 9, No. 6, pp. 539-546.

International Organization for Standardization and International Electrotechnical Commission. (2009), *ISO/IEC 27004:2009, Information Technology -- Security Techniques -- Information Security Management -- Measurement*. ISO, Geneva, Switzerland.

Judy, W. (2002) 'Notes from her Math 398 course taught in the Spring of 2002 at UNL', *Ericsson AB*. ERLANG Secure Socket Layer 5.1.1.

Mollin, R. A. (2005), *Codes: The Guide to Secrecy from Ancient to Modern Times*, Chapman & Hall/CRC, Boca Raton, FL.

NIST. (2000), 'Federal information technology security assessment framework', available at http://csrc.nist.gov/drivers/documents/Federal-IT-Security-Assessment-Framework.pdf (accessed 28 February 2014).

Satti, M. and Kak, S. (2009), 'Multilevel indexed Quasigroup encryption for data and speech', *IEEE Transactions on Broadcasting*, Vol. 55, No. 2, pp. 270-281.

Sikarwar, N.S. (2012), 'An integrated synchronized protocol for secure information transmission derived from multilevel steganography and dynamic cryptography', *International Journal of Computer Science and Telecommunication*, Vol. 3, No. 4, pp. 31-36.

Simmons, G.J. (1979), 'Symmetric and asymmetric encryption', *Computing Surveys*, Vol. 11, No. 4, pp. 305-330.

Ugwu, J.N. (2014), 'Multilevel offline cryptography support system', Undergraduate project, Federal University of Technology, Minna, Nigeria.

Usha Devi, G. and Wahida Banu, R.S.D. (2012), 'Secure multilevel cryptography using graceful codes', *International Journal of Information and Electronics Engineering*, Vol. 2, No. 5, pp. 840-843.

About the authors

Olawale S. Adebayo is a Lecturer in the Department of Cyber Security Science, Federal University of Technology Minna, Niger State, Nigeria. He bagged Bachelor of Technology in Mathematics and Computer science from Federal University of Technology, Minna in 2004 and MSc. in Computer science from University of Ilorin, Kwara State, Nigeria in 2009. He is presently a PhD student in the Department of Computer Science, International Islamic University Malaysia. His current research interests include: Malware Detection, Information Security, Cryptology, and Data Mining Security. He has published many academic research papers in the above-mentioned research areas. He is a member of Computer Professional Registration Council of Nigeria (CPN), Nigeria Computer Society (NCS), IEEE, Global Development Network, International Association of Engineers (IAENG) and many others. He is a reviewer to many local and international journals and conferences. More at http://www.osadebayo.com.

Corresponding author. Department of Cyber Security Science, School of Information and Communication Technology, Federal University of Technology, Minna, Niger State, Nigeria. Tel: +2348032100361, +60186672915. E-mail address: waleadebayo@futminnna.edu.ng

Morufu Olalere is a Lecturer in the Department of Cyber Security Science, Federal University of Technology Minna, Niger State, Nigeria. He bagged Bachelor of Technology in Industrial Mathematics from Federal University of Technology Akure, Nigeria in 2005 and MSc. in Computer science from University of Ilorin, Kwara State, Nigeria in 2011. He is presently a PhD student in the Department of Computer Science, University of Putra Malaysia. His current research interests include: Information Security and Network Security. E-mail address: lerejide@futminna.edu.ng

Joel N. Ugwu is a graduate of Federal University of Technology Minna, Nigeria. He bagged Bachelor of Technology in Computer Science with Cyber Security option in 2014. E-mail address: joeljupiter@yahoo.com