PRINT ISSN 1119-8362
Electronic ISSN 2659-1502

Full-text Available Online at
https://www.ajol.info/index.php/jasem
http://www.bioline.org.br/ja

J. Appl. Sci. Environ. Manage.
Vol. 27 (1) 155-160 January 2023

Enriching Information Security via Hybrid of New Expand Rivest Shamir Adleman and Data Encryption Standard Cryptosystem

[1]HAMMAWA, MB; [1]BISALLAH, H; *[2]ABDULRAHMAN, A

*[*1]Department of Computer Science, University of Abuja, Abuja, Nieria*
[2]Department of Computer Science, Ibrahim Badamasi Babangida University, Lapai, Niger State, Nigeria

**Corresponding Author Email: abdulg2009@ibbu.edu.ng*
Co-Authors Email: Hashim.bisallahA@unibuja.ude.ng; mbhammawa@gmail.com

ABSTRACT: Cryptography system gives security services ability to protect information from people who are authorized to use it. This research introduces elevated information security system via hybrid of new expand Rivest Shamir Adleman (RSA) and data encryption standard (DES) cryptosystem using appropriate standard lgorithms. An Expand Algorithm (ERSA) algorithm Key cryptosystem based on nth large prime number system was introduced. Then a hybrid between the Expand Algorithm (ERSA) with Data Encryption Standard (DES) was applied using four randomly selected variables, each generated from large factor of "N" prime numbers. The information undergoes a cypher text-decryption process which provides strong security and uphold high information confidentiality and integrity of data. Results showed that the use of ERSA and DES algorithm reduced the key generation period and its complexity analysis of encryption and decryption are stronger, unlike the application of traditional RSA algorithm.

DOI: **https://dx.doi.org/10.4314/jasem.v27i1.22**

Cite this paper as: HAMMAWA, M. B; BISALLAH, H; ABDULRAHMAN, A. (2023). Enriching Information Security via Hybrid of New Expand Rivest Shamir Adleman and Data Encryption Standard Cryptosystem. *J. Appl. Sci. Environ. Manage.* 27 (1) 155-160

Dates: Received: 01 January 2023; Revised: 25 January 2023; Accepted: 26 January 2023; Published: 31st January 2023

Keywords: cryptography; Private Key; plain text; Cryptosystem; Public Key Security

It is known that the best way to have effective, secure and low-cost storage of data files is to store our information in the cloud. There is need to have a strong encryption and decryption algorithms that will help secure our database, as keeping information on the internet is getting popularity daily. Most private organizations and government are interested in using cloud services to store their confidential information, in which, if stored in the cloud it is less cost. Companies like Amazon and others cloud companies are organizations providing these services. The major challenges on cloud computing are the inheritance of internet insecurity problems. Hackers invading stored information on cloud and steal the data. Though cloud provides some method of protecting information by encrypt and decrypt information (AbdElminaam, 2018).These methods applied are not good enough to rely on. Cloud computing are servicing both public and private organizations which store good quantity of information on their cloud. The information can be reachable or access by the organizations from any part of the world through the internet which is not secure. The major security challenges is the information stored on cloud can be access through user authentication at various layer of security (Budiman & Rachmawati, 2018). The Cryptography system apply on various layers security is reliant on the level of confidentiality required. Though, we cannot fully rely on these securities provided as hackers still access information on cloud without authority. The emergences of cloud computing system brought many benefits to government and organization using it.

**Corresponding Author Email: abdulg2009@ibbu.edu.ng*

Likewise, it has lot of weaknesses that come along with cloud computing and must be solved by using an enhance cryptosystem. The major drawback is inadequate security provision to information in the cloud from unapproved entree (Bhandari, Gupta, & Das, 2016). These bugs are challenges to both cloud service providers and the customers who are interested in putting their data on the cloud. In order to secure the sensitive data from unlawful access, a suitable improve encryption algorithm must be applied to cloud data storage facilities. There are numerous security processes which have been futured for securing the cloud computing, some of these proposed security systems use encryption procedure.

RSA algorithm is one of the security measures used by researcher to secure database centre, this has been working fine to protect unauthorized users (Denis & Madhubala, 2021). RSA algorithm has two main processes to follow: Key generation, then encryption with the decryption system which is applied to secure data stored on cloud. The intention is to curtail the time complexity and space analysis throughout these progressions. There is need to give more to this RSA algorithm, expand manipulate encryption method which is now introduced known as Expand RSA (ERSA). This system of algorithm is to prevent attackers against hacking. Mathematical attacks with brute forces are taking care of with the help of using ERSA algorithm. The study hybrid the new algorithm to farther strengthen the system using DES. In traditional RSA, there is a high probability of conjecturing the combination until there exponent size scope are made complex and above 2048 bits. In the anticipated algorithm, the likelihood is reduced, if the exponent scope size is 1024 bits or more. (Meneses et al., 2016), (Budiman & Rachmawati, 2018). Our own enhancement which is Expand RSA utilizes four prime numbers or more rather than two primes with the encryption controlled. The decoding succession is equivalent to unique traditional RSA and hybrids it with DES to make it more powerful.

MATERIALS AND METHODS

An Expand RSA cryptosystem has the prospect to protect our data by means of "n" prime numbers. Purposive selected large prime numbers with modular multiplicative reverse are used to strengthen the security of normal RSA algorithm. The secret key which is used to encrypt by converting plain text gives Expand RSA (ERSA) strength to deny documents which are of interest to the hackers. The hackers are not unauthorized to access the data. Introducing hybrid of two different cryptosystems really help the improvement of our research. DES is first used to encrypt password authentication and the use of ERSA to re-encrypt the password for the second time makes it stronger. We analysis this research by using JAVA programming language, as Expand RSA (ERSA) and DES are executed utilizing JAVA. We randomly introduced large four different prime numbers which might be enormous number to protected arbitrary capacity. Different primes numbers are applied by prime system creator which works in Java Bulky Whole number library to delivered secret key cipher text time. The Expand RSA (ERSA) classic does examine the key production coding and decoding period amid Conventional RSA and Extend RSA (ERSA) in the light of these structured periods for unequivocal piece interval.

RSA (Rivest Shamir Adleman) Cryptosystem: RSA (Rivest Shamir Adleman) rationally uses two prime numbers to generate it keys for public encryption and private decryption keys. These keys are used to manipulate text messages in a way that unauthorized users cannot understand the text messages. The e which is call encryption key is used to turn the message to a cipher text, and the cipher text is been sent to the end receiver which he/she will convert it to plain text that is readable format using d decryption. Below show the three steps in which keys are been generated and possible calculations on traditional RSA algorithm (Al Mamun, Mahmood, and Amin, 2021).

Generation OF RSA Key:
RSA_key_gen() Input:
Input prime A and B
*output the product of A*B= N*
Euler phi of N = (A-1) (B-1)
code key: (Chaudhary et al., 2018)
Procedure:

RSA Encryption: In RSA algorithm, text messages are converted to meaningless text messages to unauthorized user by using *e* to encrypt our information before sending it to the end users (Murad and Rahouma, 2022). We apply decryption method which have the necessary formular to turn it to readable text message. The following steps are taken to covert plain text to cipher text:

1) Obtains the recipient public key (n, e)
2) Represent the plain text message as positive integer.
3) Compute the cipher text.
4) Send the cipher text.

RSA Decryption: RSA decryption is done with the help of private key to get the plain text message. The steps for decryption are given as:

1) Compute by using private key.

2) Extracts the plain text from integer representing.

Data Encryption Standard (DES): Most widely used encryption since the 1970s is the DES data encryption standards (Lalitha and Srinivasu, 2017). Till today, this encryption standard is still in use. Some of the legacy applications of this encryption system are using the DES encryption even though it is considered as insecure as the asymmetric algorithm. Nowadays, due to the key size DES encryption, people are still using DES which is an algorithm that can be broken easily using modern computing systems (Hammawa MB, et al, 2019). The hybrid of ERSA -DES will help in strengthen date that are been secure with. 1974 IBM introduced cryptographers working encryption based on Horst Feistel Lucifer encryption method (Lalitha and Srinivasu, 2017). The proposed encryption was given to NBS. It was sent to NSA for review which was approved. In 1977, DES standard encryption was approved to use 56-bit key length and data block size of 64 as the Data Encryption Standard. Since then, it has been used for the common public use and commercial use (Lalitha and Srinivasu, 2017). DES is configured to secure data against cryptanalysis attack till 1990s. DES has stand against attack which led to U.S. government approving it to be used in different sectors for a period 10 years which lasted till 1999. AES advance encryption standards was introduced to replace DES in the year 2001. The first DES introduced, has 56-bit length key size, which process a 64-bit data block size within a period of 16 round block encryption. The text message of data block size of 64 bit will be cyphered 16 number of times, each time different sub-key will be introduced which was generated from the innovative of 56-bit secret key. Few ideas are introduced to make the transformation of data block to encryption stronger so as to make it more Confusion and Diffusion to read. Whereas the diffusion techniques depend on manipulating the main message in bit or symbols with multiple encryption or symbols making it more difficulty for interpretation of the real after encryption (Hamdan.O.Alanazi, 2010).

Proposed Model: An Expand RSA algorithm is introduced to enhance traditional RSA algorithm with series of attempts to hacks on the traditional RSA. These unauthorized users have gotten to know that RSA algorithms use two prime numbers as a traditional method, the hackers have made many attempts to hack it. Especially, often times, when sending the encryption keys hackers make attempt to break it to the encrypted text messages that is been send. The study proposes Nth prime numbers to strengthen and make it harder to predict any possible attack on this new ERSA algorithm and it integrates it with DES. Expanding the RSA algorithm helps to make it difficult to hack into the encrypted data message which is to be sent to a receiver. The end user who has the decryption key can now decrypt it for better understanding of the text message. With the help of this ERSA and DES hybrid, it will help in protecting data from hackers who are making attempt to crack the information send which is confidential.

ERSA key: This proposed algorithm has space of N^{th} prime numbers that are involved in strengthening the proposed ERSA scheme. Our conversion to ciphers and reverting it to deciphers follow four randomly picked large prime numbers. We use indiscriminately ways to choose enormous prime numbers. This means that we randomly generate prime numbers. In this case, we generate large four primes to test run our new algorithm, which the variables are as; "w", "x", "y", and "z". We found the product of these prime numbers which give us the n. After that, we calculated phi number which is the subtraction of one from each selected prime and multiply the remainder to have our phi number, from this, we got our encryption key called e key. These keys advocate contains different parts (e, f, N) where "f" is made from encryption key "e". It gives greater unpredictability while the considering of "N". The "N" is covered up into a segment "g" and g are sent. Consequently, a trespasser with the realities of "g" got a bogus of "N" which is expected to calculate and get the estimation of four prime numbers and afterward "e".

Our new research is introduced underneath.

```
ERSA_ key_gen()
   */
   public void generatePublicPrivateKeys()
   {
   // N = w * x*y*z
   N = p.multiply( q ) ;

// r = (w – 1) *(x – 1)*(y – 1)*( z – 1)
   r = r.multiply( q.subtract( BigInteger.valueOf( 1 ) )
) ; //(w-1)(x-1)
```

ERSA Encryption with Decryption: The cipher text message which is made with encryption and decryption keys are made with public and private key exponent. This "N" which is a product of randomized large prime number giving it strong to factor four arbitrary "e", "f", "d", "g". This is part of what built encryption to become problematic for hackers to hack the system. So, the procedures in which we operate this algorithm goes as follows:

```
   * Generate Public and Private Keys.
   // Choose E, coprime to and less than r
```

```
do
{
E = new BigInteger( 2 * primeSize, new Random()
) ;
}
While ( ( E.compareTo( r ) != -1 ) || ( E.gcd( r
).compareTo( BigInteger.valueOf( 1 ) ) != 0 ) ) ;
```

Figure 1 shows how the data flow diagram fully explaining the interconnectivities of Expand RSA (ERSA) system. Four large randomly distinctive prime integer are chaotically chosen as an input to process, their product is N while the phi Φ(N). The (e, f) is thoroughly looked over the unveil $1 < e < \Phi(N)$. The (d, g) is determined to apply the secret key one cipher text message and plain text message are applied smearing secret and open public key types.

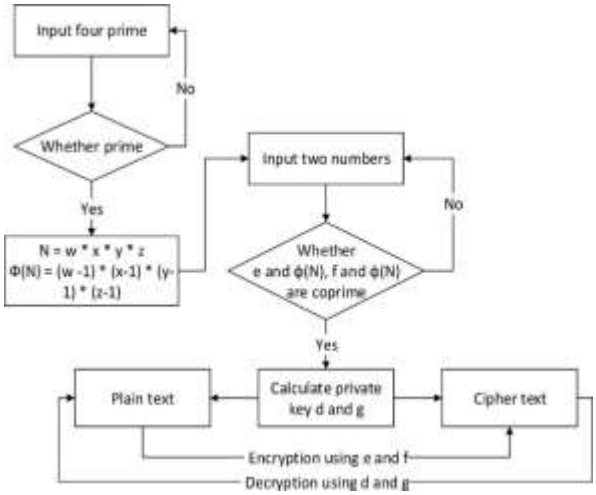

Fig 1. Flow chart of Expand RSA (ERSA) Algorithm.

RESULT AND DISCUSSION

ERSA example

Key Generation

$w = 199$, $x = 211$, $y = 223$, $z = 227$

$N = w * x * y * z$

$= 199 * 211 * 223 * 227$

$= 2, 125, 525, 169$

$Ø\ (N) = (w - 1)\ (x - 1)\ (y - 1)\ zt - 1)$

$= 198 * 210 * 222 * 226$

$= 2, 086, 151, 760$

Public Key

$e = 563$

$f = (563 * 2) + 1$

$f = 1127$

Private Key

$563d \pmod{2, 086, 151, 760} = 1$

$3, 705, 420.533$

$0.533 \approx 1$

$d = 1{,}111, 918, 888$

$g = d - 1$

$g = 1{,}111{,}918{,}887$

Encryption

$C = M^{(f-1)/2} Mod\ (g+1)$

$C = 1234^{((1127 - 1)/2)} Mod\ (1111918887 + 1)$

$C = 1406540448$

Decryption

$M = C^{d} Mod\ (g+1)$

$C = 1406540448^{1111918887} Mod\ (1111918887 + 1)$

$C = 1234$

Result Simulation: We achieve our proposed hybrid of Expand RSA (ERSA) and DES algorithm by simulating it while utilizing JAVA running on an Intel Center (TM) i3-3540M CPU @ 4.60 GHz and 6.00 GB Slam. The algorithm (Customary RSA, proposed Expand RSA (ERSA) and DES) have influencing the degree of security quality and speed. Expanding the modulus length raised multifaceted nature of factorizing it. Thus, additionally increases the length of the mystery key. The conventional RSA and proposed Expand RSA (ERSA) boundary changes with time and others near noticeable quality that show the new algorithm serves better. Analysis on RSA performance: Expand of the new algorithm (ERSA), was research to factor size bit of information. This compares between new algorithm and the conventional Algorithm are trendy on Table 1.

Table 1. Performance of RSA

Length of w,x	**(Time analysis for RSA algorithm)**		
(in bits) Key generation (in ms)		**Encryption (in ms)**	**Decryption (in ms)**
64	67.53	0.15	0.23
128	86.45	0.18	0.26
256	88.86	0.29	0.87
512	168.57	0.48	4.3
1024	530.8	1.56	23.16
2024	4189.37	3.29	126.72
4096	53,456	10.15	1100.2

Likewise, the proposed Expand algorithm (ERSA) framework of coding period and decoding are shown on table 2.

Table 2. Performance of Expand RSA (ERSA)

Length of w,x,y	**Analyzing time for Expand RSA (ERSA) algorithm**		
and z (in bits) Key generation time		Encryption (in ms)	Decryption time (in ms)
(in ms)			
100	236	0.19	1.6
128	249.29	0.58	2.78
256	246.9	1.4	13.3
512	234.6	3.88	85.87
1024	1249.4	6.76	439.38
2048	7067.9	20.77	2362.77
4096	1,241,900	55.79	18,999.46

The proposed Expand (ERSA) is more mind-boggling compared to traditional algorithm. More drawn-out

secret keys period of proposed Expand (ERSA) can be viewed as merit on the ERSA by way that they stretch to hack the cipher text message is harder as a result of the unnecessary intricacy included. Figure 2 portrays the encryption stretch assessment among RSA and proposed Extended (ERSA) structure. It shows that, for the subordinate piece length of prime numbers, two algorithms take nearly a similar total of time. In any case, with the expansion of bit length, the hole between bends rises quickly.

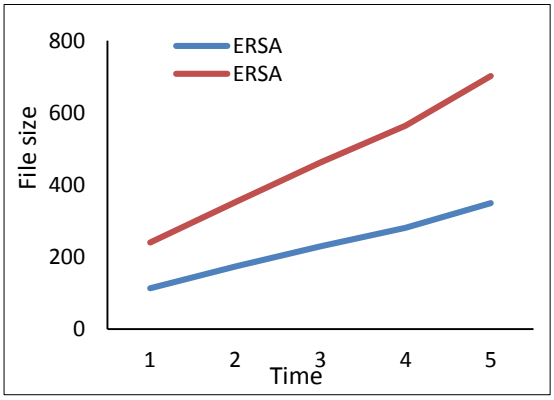

Fig 2. Encryption time comparison

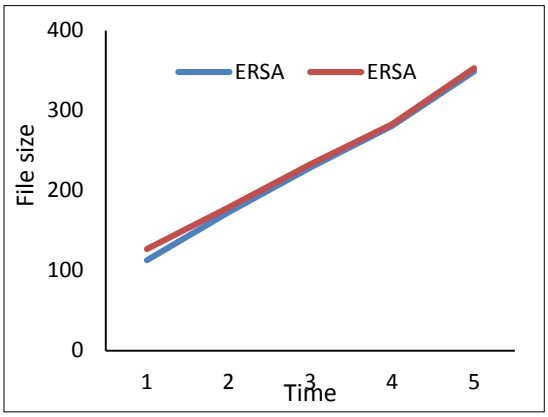

Fig 3. Decryption time comparison

Figure 3 demonstration shows decryption period comparison between traditional RSA and proposed ERSA system. It proves looks almost the same amount of time expended by traditional and ERSA for the minor bit length of primes. With the rise of bit length, the variance between curves raises slowly. Without difficulty, the above graphs show that encryption and decryption period are of the high side than traditional RSA. The increase in stretch is flexible because it strengthens the security to a great degree in the proposed ERSA technique.

Complex Analysis: Analyzing the complexity of Traditional RSA algorithm with ERSA algorithm is analyze beneath.

Complexity of RSA Algorithm: In Arbitrary Complexity selected four primes for the algorithm.

The Complexity using Expand RSA (ERSA) Algorithm: The complexity helps to raise the graphical representation based on the number of the prime accepted for the proposed improve algorithm. When randomly picked prime numbers act like this:

- (1^{st}) prime number is $O\ s(*(\log 2w)4 *\ln w)$.
- 2^{nd} prime number is $O\ s(*(\log 2x)4 *\ln x)$.
- 3^{rd} prime number is $O\ s(*(\log 2y)4 *\ln y)$.
- 4^{th} prime number is $O\ s(*(\log 2z)4 *\ln z)$.

Big of O N:

- With the arithmetic calculation of this complexity of N $O(\log 2w*\log 2x*\log 2y*\log 2z)$.

using Euler phi value of N

- Using Euler phi value of N remains:

$O((\log 2(w-1) * - * - * (x\ 1)\ (y\ 1)\ (z-1)_)^4 *((w-1) * - * - * (x\ 1)\ (y\ 1)\ (z- -1)\ 1^)$.

Complexity for random variables e and f:

- The random variable e remains

$O(\log 2(w-1) * - * - * (x\ 1)\ (y\ 1)\ (z- +1)$

Applying gcd ($e\ w\ (-1) * - * - * (x\ 1)\ (y\ 1)\ (z-1)$)), e and (N) are coprime to one another so

Gcd ($e\ w$, $(-1) * - * - * (x\ 1) * (y\ 1) * (z - =1))\ 1$,

while the complexity is $O((\log 2\ \log 2(\ w-1)*(\log 2x-1)*(\log 2y-1)*(\log 2z-1)_)^4 +1)$.

- Equally, Complexity of computing the random value f is

$O((\log 2\ \log 2(\ w-1)*(\log 2x-1)*(\log 2y-1)*(\log 2z-1)_)^4 +1)$.

Linking the above complexity, it illustrates that Expand RSA (ERSA) it demonstrates more complex than traditional RSA algorithm.

Conclusion: Looking at large prime number apply on ERSA keys with the hybrid of DES, it shows that the security in encrypting database is stronger than using traditional of two primes system. The larger the primes key, the more it expands the period needs to hack the framework which brands the encryption algorithm. The cipher text and plain text message using ERSA is straightforward and outweighed the traditional RSA. The achievement of this design is estimated with a period of time taken for brute force attack.

REFERENCES

AbdElminaam, D. S. (2018). Improving the security of cloud computing by building new hybrid cryptography algorithms. *International Journal of Electronics and Information Engineering*, *8*(1), 40-48.

Al Mamun, S., Mahmood, M. A., & Amin, M. A. (2021). *Ensuring Security of Encrypted Information by Hybrid AES and RSA Algorithm with Third-Party Confirmation.* Paper presented at the 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS).

Bhandari, A., Gupta, A., & Das, D. (2016). *Secure algorithm for cloud computing and its applications.* Paper presented at the Cloud System and Big Data Engineering (Confluence), 2016 6th International Conference.

Budiman, M. A., & Rachmawati, D. (2018). Using random search and brute force algorithm in factoring the RSA modulus. *Journal of Computing and Applied Informatics, 1*(2), 48-56.

Chaudhary, R., Jindal, A., Aujla, G. S., Kumar, N., Das, A. K., & Saxena, N. (2018). LSCSH: Lattice-Based Secure Cryptosystem for Smart Healthcare in Smart Cities Environment. *IEEE Communications Magazine*, 25.

Denis, R., & Madhubala, P. (2021). Hybrid data encryption model integrating multi-objective adaptive genetic algorithm for secure medical data communication over cloud-based healthcare systems. *Multimedia Tools and Applications, 80*(14), 21165-21202.

Hamdan.O.Alanazi, B. B. Z., A.A.Zaidan, Hamid A.Jalab, M.Shabbir and Y. Al-Nabhani. (2010). New Comparative Study Between DES, 3DES and AES within Nine Factors. *Journal of Computing and Applied Informatics, 2*(3), 152-157.

Hammawa MB, Owolabi O, Abdulganiyu A, & Amit, M. (2019). Building information security using new expanded RSA cryptosystem. *International Journal of Research in Advanced Engineering and Technology, 5*(4), 65-68.

Lalitha, R., & Srinivasu, P. N. (2017). An efficient Data Encryption through Image via Prime order symmetric key and Bit shuffle Technique *Computer Communication, Networking and Internet Security* (pp. 261-270): Springer.

Meneses, F., Fuertes, W., Sancho, J., Salvador, S., Flores, D., Aules, H., . . . andNuela, D. (2016). RSA Encryption Algorithm Optimization to Improve Performance and Security Level of Network Messages. *International Journal of Computer Science and Network Security (IJCSNS), 16*(8), 55-62.

Murad, S. H., & Rahouma, K. H. (2022). Hybrid Cryptography for Cloud Security: Methodologies and Designs *Digital Transformation Technology* (pp. 129-140): Springer.