To Compare Encryption Algorithms on Application and Database Layer on the Basis of Computation Time

Jaskaran Kaur, Richa Sharma
Computer Science Department, LPU, Punjab, India,
royaldhillon89@gmail.com

***Abstract* - In data communication, information security plays an important role. Encryption algorithms play a vital role in information security system. These algorithms consume a significant amount of computing resources such as CPU time, memory and battery power and computation time. This paper performs comparative analysis of two algorithm; RSA and transposition cipher on two levels- application level and database level considering a parameter such as computation time. Encryption at the database level ,and application level has proved to be the ideal method to protect sensitive data.**

***Keywords*- Transposition Cipher, cryptography, database level encryption, application level encryption.**

1. INTRODUCTION

In today's scenario, communication without security is not reliable. The main goal of this research is to provide the fair computation time comparison of various Encryption Algorithms at different text data size to evaluate the average speed of Encryption and Decryption process at application level and database level. Application Level Encryption moves the encryption/decryption process to the applications that generate the data. Encryption is thus performed within the application that introduces the data into the system; the data is sent encrypted, thus naturally stored and retrieved encrypted, to be finally decrypted within the application. Database-level encryption allows securing the data as it is inserted to, or retrieved from the database [1]. Encryption Algorithms provides the security to the information which is exchange over internet. The encryption algorithms are usually summarized into two popular types: Symmetric key encryption and Asymmetric key encryption. Symmetric-key algorithms are a class of algorithms for cryptography that use the same cryptographic keys for both encryption of plaintext and decryption of cipher text. Asymmetric cryptography is also known as public key cryptography and relies on the use of two unique keys—the public key and the private key [2]. The two main characteristics that identify and differentiate one encryption algorithm from another are its ability to secure the protected data against attacks and its speed and efficiency in doing so.

A. Cryptography

In cryptography, cipher text is the result of encryption performed on plaintext using an algorithm, called a cipher. Cipher text is also known as encrypted or encoded information because it contains a form of the original plaintext that is unreadable by a human or computer without the proper cipher to decrypt it. Decryption, the inverse of encryption, is the process of turning cipher text into readable plaintext. Cipher text is not to be confused with code text because the latter is a result of a Code, not a cipher [11]. Cryptography is the science of writing in secret code and is an ancient art; the first documented use of cryptography in writing dates back to circa 1900 B.C. when an Egyptian scribe used non-standard hieroglyphs in an inscription [12]. Some experts argue that cryptography appeared spontaneously sometime after writing was invented, with applications ranging from diplomatic missives to war-time battle plans. It is no surprise, then, that new forms of cryptography came soon after the widespread development of computer communications [6]. In data and telecommunications, cryptography is necessary when communicating over any untrusted medium, which includes just about any network, particularly the Internet.

B. RSA Algorithm

RSA is a commonly adopted public key cryptography algorithm [3]. The first, and still most commonly used asymmetric algorithm. RSA is named for the three mathematicians who developed it, Rivest, Shamir, and Adleman. RSA has been widely used for establishing secure communication channels and for authentication the identity of service provider over insecure communication medium. RSA today is used in hundreds of software products and can be used for key exchange, digital signatures, or encryption of small blocks of data. RSA uses a variable size encryption block and a variable size key. The key pair is derived from a very large number, n, that is the product of two prime numbers chosen according to special rules.

RSA involves a public key and a private key**.** The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. The keys for the RSA algorithm are generated the following way [9]:

1. Choose two distinct prime numbers p and q.
   For security purposes, the integer's p and q should be chosen at random, and should be of similar bit-length. Prime integers can be efficiently found using a primality test.
2. Compute $n = pq$.
   n is used as the modulus for both the public and private keys
3. Compute $\varphi(n) = (p\square-\square 1)(q\square-\square 1)$, where φ is Euler's totient function.
4. Choose an integer e such that $1 < e < \varphi(n)$ and greatest common divisor of $(e,\varphi(n)) = 1$, i.e. e and $\varphi(n)$ are co-prime.
   e is released as the public key exponent.

5. Determine $d = e^{-1} \bmod \varphi(n)$; i.e. d is the multiplicative inverse of e mod $\varphi(n)$.
   This is more clearly stated as solve for d given (d*e)mod $\varphi(n)$ = 1.This is often computed using the extended Euclidean algorithm.*d* is kept as the private key exponent.

C.Transposition Algorithm

All the techniques examined so far involve the substitution of a ciphertext symbol for a plaintext symbol. A very different kind of mapping is achieved by performing some sort of permutation on the plain text. This technique is referred to as a transposition cipher. The simplest cipher is the rail fence technique, in which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows [10].

II. RELATED WORK

In this section, we have studied a number of articles that make a comparison in terms of computation time between the common encryption algorithms. It was shown in [2] that the Blowfish has better performance and 3DES has least efficient than other selected algorithms. The selected algorithms are AES, DES, 3DES and Blowfish. In [3], AES, DES and RSA algorithms are used for performance evaluation and it was concluded that DES algorithm consumes least encryption time and AES algorithm has least memory usage. As shown in [4], new comparative study between DES, 3DES and AES were presented in to nine factors, Which are key length, cipher type, block size, developed, cryptanalysis resistance, security, possibility key, possible ACSII printable character keys, time required to check all possible key at 50 billion second, these eligible's proved the AES is better than DES and 3DES. To determine the performance of memetic algorithm in comparison with genetic algorithm for the cryptanalysis of SDES in [5] the performance comparison on the average number of key elements and computation time comparison for recovering the keys from the search space.

III. EXPERIMENTAL SET UP DESIGN

For our experiment a Laptop with 1GB RAM Dual-core Processor (Intel) and Windows 7 Home Premium (32- Bit) is used in which the performance data are collected. In this experiment software encrypts the text file size that ranges from character length 5 to 30. The computation time of encryption as well as decryption schemes is calculated one by one. For this RSA and Transposition Algorithms are applied on application layer and Database layer. At application level Php v5.3.5 is used and at database level MySQL v5.1 is used.

IV. METHODOLOGY

To compare computation time I will apply RSA algorithm and Transposition Algorithm on both the programming level and database level to find out which one performs better on which level.

1. Application Layer: used PHP as the application layer to implement the code for RSA and Transposition.
2. Database Layer: used the My SQL as database for database.

V. IMPLEMENTATION STEPS

1. Installed Microsoft Expression Web v4 to manage Php v5.3.5
2. Installed Premium Soft Navicat Premium Enterprise Edition v9.1.8 to manage MySQL v5.1
3. Coded RSA and Transposition in Standard Php
4. Coded RSA and Transposition in Standard MySQL.
5. Comparison of computation time using different character length.

VI. RESEARCH DESIGN

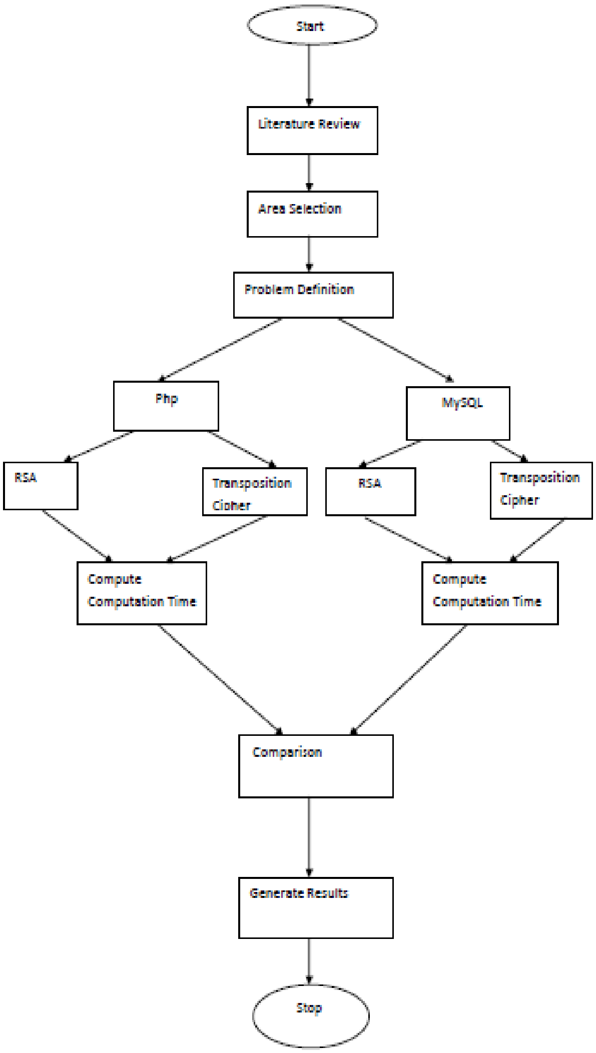
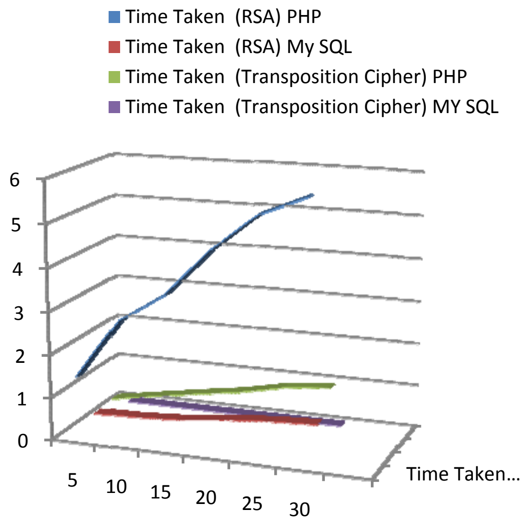


Fig 1: Research Design

VII. EXPERIMENTAL RESULTS

Table 1: Shows the encryption time taken by RSA and Transposition Cipher at PHP and MySQL.

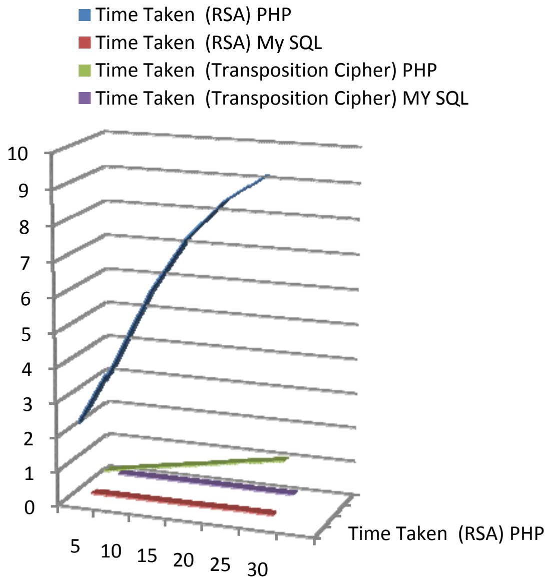
No. of Characters / Symbols	Time Taken (RSA)		Time Taken (Transposition Cipher)	
	PHP	My SQL	PHP	MY SQL
05	1.475	0.297	0.3580	0.002416
10	2.879	0.312	0.5379	0.002862
15	3.542	0.374	0.7139	0.003196
20	4.617	0.500	0.8779	0.003583
25	5.458	0.593	1.105	0.003979
30	5.901	0.642	1.212	0.004369



Graph 1: Shows the encryption time taken by RSA and Transposition Cipher at PHP and MySQL.

Table 2: Shows the decryption time taken by RSA and Transposition Cipher at PHP and MySQL.

No. of Characters / Symbols	Time Taken (RSA)		Time Taken (Transposition Cipher)	
	PHP	My SQL	PHP	MY SQL
05	2.39	0.015	0.3920	0.002691
10	4.242	0.015	0.5870	0.003041
15	6.321	0.015	0.7669	0.003481
20	7.892	0.016	0.9409	0.003827
25	8.963	0.031	1.135	0.004206
30	9.68	0.032	1.310	0.004632



Graph 2: Shows the decryption time taken by RSA and Transposition Cipher at PHP and MySQL.

Conclusion

Importance of cryptography has dramatically increased as information Security has become an important issue in data communication. Encryption algorithm play an important role in communication security where encryption time, Memory usages output byte and battery power are the major issue of concern. This thesis presents the performance evaluation of selected encryption algorithms at Application Level and Database level. The selected algorithms are RSA and Transposition. Based on the text used and the presented simulation results show the numerous points.

Firstly, it was concluded that Transposition cipher consumes less time for encryption and decryption than RSA algorithm at both application level and database level. Secondly, transposition is less secure than RSA algorithm.

Future Scope

Transposition Cipher is less secure and it can be easily breakable. So we can compare RSA with other encryption algorithms. In this work, comparison between RSA with Transposition Cipher has been done by taking into account the encryption and decryption time at application level and database level. But in future, more issues such as memory usages, output byte and key length can be considered and also comparison can be done on application level, database level and both application and database level.

References

[1]. Luc Bouganim and Yanli GUO(2009),"Database Encryption", INRIA Rocquencourt Le Chesnay, France.
[2].Gurjeevan Singh, Ashwani Kumar Singla, K.S. Sandha(IJCST-2011)," Through Put Analysis of Various Encryption Algorithms," Patiala, Punjab, INDIA.
[3].Shashi Mehrotra Seth, Rajan Mishra(IJCST-2011)," Comparative Analysis Of Encryption Algorithms For Data Communication,"(June)Bhadhurgarh, Harayana, India.
[4]. Hamdan.O.Alanazi, B.B.Zaidan, A.A.Zaidan, Hamid A.Jalab, M.Shabbir and Y. Al-Nabhani(JOC-2010)," New Comparative Study Between DES, 3DES and AES within Nine Factors", March, Journal of computing.
[5]. Poonam Garg(IJNSA-2009)," A Comparison between Memetic algorithm and Genetic algorithm for the cryptanalysis of Simplified Data Encryption Standard algorithm," Institute of Management Technology, India.
[6]. http://en.wikipedia.org/wiki/MySQL
[7]. http://www.php.net
[8].http://en.wikipedia.org/wiki/PHP
[9].http://en.wikipedia.org/wiki/RSA_algorithm
[10].http://www.thestudymaterial.com/presentation-seminar/electronics-presentation/53-cryptography-presentation.html?start=3
[11]. http://en.wikipedia.org/wiki/Ciphertext
[12]. http://www.securitytube.net/video/112