Mathematical Statistician and Engineering Applications
ISSN: 2094-0343
2326-9865

Performance Analysis of Intelligent Key Cryptography (IKC) System

Rojasree V.
Research Scholar,
Department of Computer Science,
Rajah Serfoji Govt. College(A),
(Affiliated to Bharathidasan University),
Thanjavur-613005, Tamilnadu, India.
rojasree.v@gmail.com

Gnana Jayanthi J.
Assistant Professor,
Department of Computer Science,
Rajah Serfoji Govt. College(A),
(Affiliated to Bharathidasan University),
Thanjavur-613005, Tamilnadu, India.
jgnanajayanthi@gmail.com

Article Info
Page Number: 67 - 78
Publication Issue:
Vol 71 No. 4 (2022)

Abstract

Apart from the two primitive techniques in cryptography, a new encryption technique called Intelligent Key Cryptography (IKC) is proposed by (i) incorporating Tamil Font metrics, for the process of both encryption and decryption; (ii) utilizing system date, time and the bio information of the sender and receivers for user's registration and (iii) generating Intelligent Keys. As in symmetric key cryptography, one key is generated for encryption and decryption and the highlight of this newly proposed IKC cryptography is that the key is intelligently generated and not shared among the users either with the sender or the receiver. This IKC is also strong enough to overcome the bottlenecks of the Quantum computing as IKC does not use any mathematical calculations. The performance of this proposed IKC is compared with Symmetric AES as well as with Asymmetric RSA and ELGAMAL with several parameters and it is proven that IKC performs a faster encryption and decryption processes than AES, RSA and ELGAMAL. Further, IKC proves to be more secured in terms of Integrity, Data Authentication, Entity Authentication, Confidentiality and Non-Repudiation.

Article History
Article Received*: 25 March 2022*
Revised*: 30 April 2022*
Accepted*: 15 June 2022*
Publication*: 19 August 2022*

Keywords: Cryptography; AES; RSA; ELGAMAL; Secret Key; Public Key; Asymmetric Key Cryptography; Symmetric Key Cryptography; Intelligent Key Cryptography.

Introduction

In the fast Internet world people are trapped by assigning their day-to-day life by the means of IoT and Internet automation by sharing sensitive data over the Internet thus making life under risk of theft and non-privacy. This makes it necessary to implement a stubborn secured cryptography algorithm where people can use these advanced facilities without fear.

Cryptography is a process in which a message shared through internet is concealed between the pertinent sender and the receiver by authenticating the correctness of the message sent to the receiver. The security of the shared message can be assured by plenty of available cryptography algorithms. The most common of these are the encryption algorithms which are of two types (i) Symmetric (private) and Asymmetric (public) key encryptions. Symmetric key encryption algorithms share the same key for both encryption and decryption whereas the Asymmetric key encryption algorithms use

two different keys one secret key and another common key. Both these keys are involved in encryption and decryption. To name a few symmetric key algorithms include DES, AES, 3DES, IDEA Twofish etc., and that of the asymmetric algorithms include RSA, Digital Signatures, Message Digest algorithms, ECC etc. This research work involves the performance analysis of one such algorithms newly designed as a research.

From the related work done it is observed that all these cryptographic algorithms contain flaws and are exposed to attacks by the intruders. Thus, motivating towards this research work to design a new cryptographic algorithm by including strength and by eradicating the vulnerabilities of these algorithms.

This research work involves the innovation of a new algorithm called Artificially Intelligent Random Key Generator (AIRKG) which could be used to overcome the vulnerabilities of the primitive algorithms. the performance analysis of AIRKG is carried out in this paper

Related Work

The performance of various cryptography algorithms were performed to analyse the functioning of the primitive schemes and also to identify the strengths and weakness of the various cryptographic algorithms. Symmetric key cryptosystem, asymmetric key cryptosystem, hashing were all taken into consideration and a good number of papers reviewed. From the review point of view AES a symmetric key block cipher had marked its footprints in the past and was later proved to be inefficient just because of the common key sharing [Rojasree *et* al., 2021a] [Rojasree *et* al., 2021a],. The vulnerability of the common key made it necessary to design a new era of cryptography where in two keys on private key for encryption and the other public key for decryption was proposed as Diffie-Hellman algorithm [Rojasree *et* al., 2021d].

The sprouting of super fast quantum computers created a threat in the security of almost all the strong considered asymmetric algorithm just because it involved complex calculations which were difficult to be solved by human on the contrary easily broken by quantum-computers. Considering all these strengths and weakness of the cryptographic algorithms, the new IKC is designed and implemented and in this paper its performance is analyzed with three chosen (AES, RSA, ElGamal) previously strong algorithms.

IKC-Architecture

In AIRKG cryptography the intermediary code is generated based on the message given by the sender. The message is a finite idea that the sender desires to communicate to the receiver, this idea is finite but cannot be formulated mathematically. Based on the message formed by the sender the intermediary code varies and in the later stages of encryption, the *Thedhimani* Chakra is used which is exact instant the time when the sender is sending the message. This time occurs only once will never occur thereafter. Even if the sender sends the same message again the cipher text differs as the time is changed. These two-indefinite finite in message communication is the strength of IKC which helps it to be strong against Quantum Cryptographic attacks. All the digital world activities are recorded and takes place with the system time included in it. This is a boon to IKC cryptographic system and makes it an added strength in securing the message.

Performance Evaluations

The proposed research work is evaluated in a Windows 10 Operating System, with i5 9th generation processor and 8GB RAM. With the precise knowledge that the performance varies with newest fast working processors and higher value of RAM.

Added to this a comparative analysis with the execution time for encryption and decryption is done using C++ code snippet by calculating the execution time. The following table, (Table-I) shows the code that is used to calculate the execution time.

Implementation Setup

The proposed work is implemented and simulated on Microsoft Visual Studio 2019 working on Windows 10 OS. The C++ code is developed in the Visual Studio Project and the performance analysed by the tools available in the application.

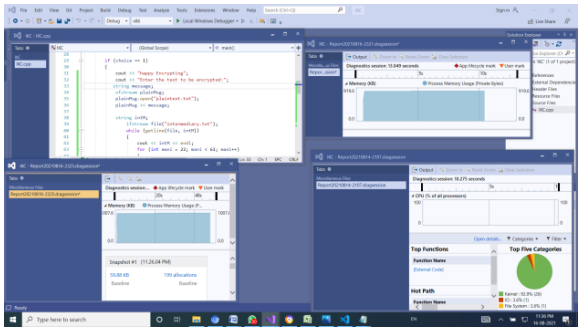

Fig.1- Development Environment

Implementation Scenario

The proposed IKC algorithm is implemented in C++. From the literature review the toughest considered algorithm of the past namely AES, RSA and ElGamal are taken into consideration for comparison with the proposed IKC architecture. To carry-out the comparison process, C++ coding for AES, RSA and ElGamal are downloaded from github [3] and executed in the same environment where IKC is implemented. The configuration of the implemented computer is as given in section 4.1. Even though this comparison is like comparing the behaviours of giraffe, zebra, cow and elephant for their characteristic behaviours just because they are herbivorous animals. Robust and popular algorithms of the past are chosen to be compared with the proposed IKC.

TABLE-I : Code to calculate the execution time

```
#include <bits/stdc++.h>
using namespace std;
void fun()
{
   for (int i=0; i<10; i++)
   {
   }
}
int main()
{
      time_t start, end;
      time(&start);
ios_base::sync_with_stdio(false);
fun();
time(&end);
double time_taken = double(end - start);
cout << "Time taken by program is : " << fixed
        << time_taken << setprecision(5);
```

```
        cout << " sec " << endl;
    return 0;
}
```

Performance Metrics

The performance metrics used to compare the proposed Intelligent Key Cryptography with existing cryptography algorithms such as AES, RSA and ELGAMAL are (1) Key Size, (2) Key Uniqueness (3) Rounds, (4) Block Size, (5) Cipher Type, (6) Encryption Time, (7) Decryption Time and (8) Security Level.

The metrics such as key size, key uniqueness, rounds, block size, cipher type and security level are compared and analysed from the theoretical point of view of AES, RSA, ElGamal and IKC algorithms. The encryption time and decryption time are compared by executing the C++ code of AES, RSA, ElGamal and IKC in the same environment as mentioned in section 4.1.

Key Size : The key size is defined as the size of the secret used to encrypt and / or decrypt the data. A thorough knowledge of the cryptographic literature always shows a vague explanation about fixing the key size required for a cryptographic algorithm. The key size is always shattered and often very difficult to be clearly defined. The cryptographic literature specifies a key size for all the algorithms that is a conservative estimate for the key sizes for the different schemes used for the next few years.

Moreover, the vulnerability of almost all the keys lies in the selection of the keys and the key space availability in all the primitive schemes. This is a fact in the cryptographic world. It is taken into consideration that the choice of the keys is the major part and maintaining the secrecy of the keys is yet another burden in all primitive algorithms.

It has become a mindset that larger keys are safe in cryptography. This is because keys of smaller size can be easily identified thus making the security algorithm weak. The larger key size is afforded at the cost of extra system resources this is an overhead in almost all primitive cryptographic algorithm.

In IKC there are two keys used, one is “Tamil Unicode’ characters and the other key is the ‘thedhiMani’ (Time). These two keys are used in the encryption-decryption of the plaintext. The encryption scheme is designed in such a way that the keys need not be exchanged between the users residing in the secret box of the encryption-decryption algorithm, in contrast when compared to the primitive algorithm. The Tamil Unicode is kept in the IKC structure, thus activated whenever an encryption-decryption algorithm is triggered. Similarly, the thedhiMani key is activated whenever the cipher is converted to intermediate code and vice versa is called. Just as both these are residing in the system as a system resource unlike the primitive encryption scheme the key exchange is not prevailing. The Tamil Unicode is the same universally so need not be exchanged between the users. The thedhiMani value is required for decryption after received by the receiver, this is included in the file where the ciphertext is stored as a system log. These logs are accessible only by the operating system of whatever device is being used. Accessing the system-oriented files is possible only with proper authentication of the concerned user. Thus, the keys chosen in IKC are intelligent and dynamic.

For all primitive algorithms there is a specific scheme wherein the key size is fixed based of the usage and security level of the cryptosystem. For the purpose of ease of use IKC can also be regularized based on the usage as shown in table (TABLE-I) above. Also, the key is compared with AES, RSA, and ELGAMAL and summarized in table, TABLE-II as follows.

For all primitive algorithms, there is a specific scheme wherein the key size is fixed based of the usage and security level of the cryptosystem. For the purpose of ease of use IKC can also be

regularized based on the usage as shown in table (TABLE-I) above. Also, the key is compared with AES, RSA, and ELGAMAL and summarized in table, TABLE-II as follows.

TABLE-II : Key Size Comparison

S.No.	Algorithm	Key Size
1	AES	128,192,256bits
2	RSA	>1024bits
3	ELGAMAL	1024bits
4	IKC	224 bits

The primitive algorithms considered here for comparison, the AES, RSA and the ELGAMAL all uses keys of size that are with some limitations AES uses key size of 128/ 192/ 256 bits each becoming a separate mode of AES. The RSA and ElGamal requires keys of more than 1024bits for a normal encryption process to be secure.

Key Uniqueness: In spite of the large key sizes used, there are proofs of timing attack and meet-in-the-middle attack in AES, RSA, and ELGAMAL algorithms. The NIST suggests a key space in which the legitimate keys are available thus giving a confined number of keys. This makes the keys to be repeated in after a long term of usage.

The cryptographic key is considered to be strong enough if it is not guessable and is unique so as no attacks could happen. The uniqueness of the keys decides the strength of the cryptographic algorithm.

In the proposed IKC crypto system the major key is thedhiMani chakra the Time factor, which is always a unique key every second in this universe. The time keeps on increasing and changes every second thus increasing the security assurance at the same time not increasing the storage overhead. The time factor is a universal entity that is a finite infinity and will never return to zero.

TABLE-III: Key Uniqueness Comparison

S.No.	Algorithm	Key Uniqueness
1	AES	Not Assured
2	RSA	Not Assured
3	ELGAMAL	Not Assured
4	IKC	Assured

Unlike the primitives AES, RSA and ELGAMAL, the key space in IKC is not confined to a specific limit. This adds on a stamp to the uniqueness of IKC.

Rounds: The ancestor symmetric key algorithms that are considered strong has to undergo multiple rounds in order to provide strong security.

Asymmetric algorithms RSA and ElGamal are not known to undergo multiple rounds. RSA and ElGamal take plaintext as input and provides ciphers as output.

In symmetric key algorithms normally, multiple rounds are implemented in order to avoid the identification of the plain text from the cipher by analysing the alphabet frequency. More rounds shuffle the characters and alphabet multiple time and hence the frequency cannot be guessed.
In asymmetric cryptographic algorithm rounds are not implemented because the mathematical equations produce chains of keys which will hide the character frequency.

The proposed IKC on the other hand reads the plaintext character by character, or in other words, the plaintext is read into the algorithm as 1byte, then written as intermediary code which is then converted to cipher text sequence. Similarly, in the decryption stage the cipher text is converted into intermediary code and then converted to plaintext.

TABLE IV: Rounds in IKC

S.No.	Usage	Chakras included	No. of Rounds (with respect to thedhiMani)

1.	Basic	lipi and thedhiMani	1
2.	Second Level	lipi, thedhiMani and mani	2
3.	Third Level	lipi, thedhiMani, mani and nimidam	3
4.	Fourth Level	lipi, thedhiMani, mani, nimidam and vinadi	4
5.	Fifth Level	lipi, thedhiMani, mani, nimidam, vinadi and madham	5
6.	Then After	Based on the length of the message these chakras can go for more rounds	n

In IKC the main chakra the thedhiMani is of 14 elements which is used in encryption and decryption this undergoes a rotation which can be taken as rounds with respect to thedhiMani chakra. The number of rotations of the thedhimani chakra is dependent on the length of the message. As the length of the message increased the rounds of the thedhiMani chakra also increases so as to convert the entire plaintext to cipher and vice versa. The table, (TABLE-IV) depicts how the rounds of IKC can be counted.

IKC architecture is specifically designed with rounds that is entirely different from that in the symmetric block cipher. In IKC the number of rounds undergone in the algorithm is dynamic which is dependent on the length of the plain text.

TABLE-V: Rounds Comparison

S.No.	Algorithm	Block Size
1	AES	Un-Employed
2	RSA	Un-Employed
3	ELGAMAL	Un-Employed
4	IKC	Employed with specific criterion that dynamically depends on the length of plain text

The table (TABLE-V) summarises the rounds in IKC as well as the primitive algorithms, considered for comparative study. In AES, RSA and ElGamal rounds has no significance, but in *IKC the rounds add on to the speciality of dynamic strength of the crypto system.*

Block Size: Generally, the block size in cryptography is the input acceptance capacity of the cryptography algorithm.

In symmetric cryptography the blocks of data are used in counter based invoking method, where the blocks of plain text are queued, to be encrypted one after another as the counter (CTR) goes to ensure the privacy of the message. In Asymmetric algorithms the blocks are not enacted as it slows down the process.

Proposed IKC uses stream of characters, and every character entered from the keyboard is taken into the IKC as 16bit character so as to be encrypted or decrypted. Thus, in IKC, a stream of 16bit / 2byte block read.

In asymmetric key algorithms the plain text takes as a whole and encrypted at a stretch, so the input is one full plain text and the output is one full cipher text.

The table (TABLE-VI) summarises the 128bits block used in AES, and that blocks are not apt in asymmetric algorithms, whereas in IKC the plain text is given as a text file into the algorithm and in the algorithm the plain text is read in blocks of 16bits at a time.

TABLE-VI: Block Size Comparison

S.No.	Algorithm	Block Size
1	AES	128bits
2	RSA	Not Fixed
3	ELGAMAL	Not Fixed
4	IKC	Reads 16bits or 2bytes

Cipher Type: The cipher type of a cryptographic algorithms is defined based on how the keys are distributed among the users as symmetric (common key) and asymmetric (one common and one private key). Another way to classify the cipher is by the method in which the plain text is scanned into the crypto system as stream (stream of bits) or as block cipher (blocks of bits) are scanned into the system.

In IKC, the Plain text is sent to the encryption algorithms as a stream of characters, and these characters are shuffled to a different code using the Tamil Unicode values, every character entered by the user is encrypted by a 2 Byte key at a time and appended into a file of encrypted text. The same key is used for both at the sender and the receiver end.

Hence this IKC can be classified to be a symmetric stream cipher with a difference that key used once is never repeated for ever, thus adding on as a one-time-pad too.

Unlike primitive symmetric cipher, in IKC, there is no physical exchange of the keys. IKC key is generated in the system where the sender is present and from where the sender creates the message. If in IoT, the key is processed in the edge devices where security is implemented. In emailing, the key resides in the server where the user's login in to exchange their data. In cloud, IKC is placed into the common interface where in all the devices that uses IKC are connected.

TABLE-VII: Cipher Type Comparison

S.No.	Algorithm	Cipher Type
1	AES	Symmetric Block cipher
2	RSA	Asymmetric Cipher
3	ELGAMAL	Asymmetric Cipher
4	IKC	Symmetric, One-Time Padded, Dynamic Key Cipher

The table (TABLE-VII) summarises the classification of ciphers as AES as symmetric block cipher, RSA and ElGamal as asymmetric ciphers and IKC as Symmetric, One-Time Padded, Dynamic key cipher.

Hence the IKC key is activated from the interface and distributed to the corresponding participants thus making IKC dynamic and Intelligent.

Encryption Time: The time taken by an encryption algorithm to convert a plaintext into a cipher text is the encryption time.

The encryption time for the proposed IKC cryptography algorithm, with the existing algorithms particularly the symmetric AES, and the asymmetric RSA and ElGamal, are observed from the experimental study and are presented in the following table Table(VIII) for the given plain text of size 32kb, 126kb, 200kb, 246kb, 280kb respectively.

TABLE-VIII : Comparison of Encryption Time

Plain Text Size (in kb)	Encryption Time (in Seconds)			
	AES	RSA	ELGAMAL	IKC
32	0.15	0.13	0.45	0.11
126	0.46	0.52	1.03	0.35
200	0.72	0.74	1.41	0.42
246	0.95	1.11	1.75	0.83
280	1.12	1.39	1.83	1.06

From the table Table (VII), a graph is plotted as given in figure Fig.2 below. It is observed from the table and figure that the encryption time for the proposed IKC work is lesser compared to AES, RSA and ElGamal algorithms.

This is because IKC is designed and developed with the reduction of size of the data structures used to store the plain text and keys; and by reducing the repeated calls to encryption functions by implementing header files and constructors, thus reducing the storage overhead too. 1.3% of encryption time is reduced while compared with AES; 1.4% of encryption time is reduced while compared with RSA algorithm and 0.9% of encryption time is reduced while compared with ElGamal.

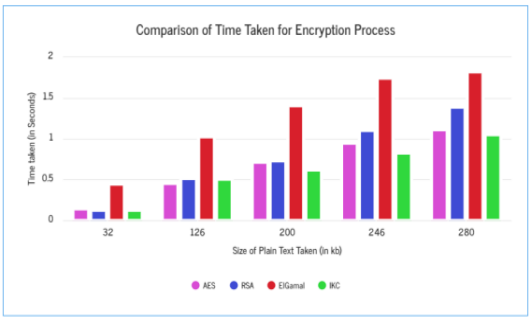

Fig 2.- Comparison of Encryption Time

The throughput is calculated by using the following formula *thorughput =(total text file size)/ (total evaluation time of algorithm).* Table VIII shows the throughput value calculated for AES, RSA, ElGamal and IKC. The bargraph comparison of the encryption time throughput is shown in figure fig 3.

TABLE-IX : Throughput of Encryption Time

Time & Throughput	AES	RSA	ElGamal	IKC
Total Average Tikme (Sec)	0.68	0.778	1.294	0.554
Throughput (kb/sec)	260	227.2494	136.6306	319.1336

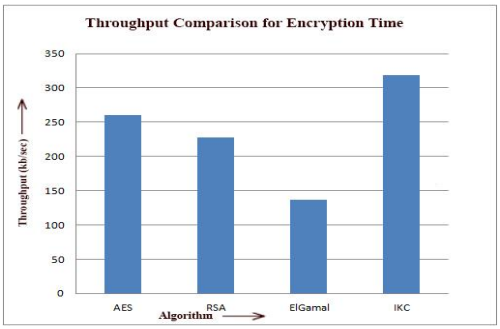

Fig 3 - Throughput of Encryption Time

Decryption Time: The time taken by a decryption algorithm to convert a cipher text into a plain text is the decryption time.

As done for encryption, the decryption time for the proposed IKC cryptography algorithm, is compared with AES, RSA, and the ElGamal. The results observed from the experiment carried out is recorded in the table IX below and the corresponding comparison chart is shown in the figure Fig.3. below. For plain texts of sizes 32kb, 126kb, 200kb, 246kb, 280kb the AES, RSA, ElGamal and IKC are checked for decryption time and are found that IKC is 1.3% lesser time than AES 1.3% lesser time than RSA and 0.8% lesser decryption time than ElGamal.

TABLE-X : Comparison of Decryption Time

Bits	AES	RSA	Elgamal	IKC
32	0.15	0.15	0.43	0.16
126	0.44	0.43	0.85	0.46
200	0.63	0.63	1.13	0.51
246	0.83	0.83	1.3	0.79
280	1.1	1.1	1.64	1.05

Fig 4 - Comparison of encryption time

The throughput is calculated by using the decryption time observerd for AES, RSA, ElGmal and the proposed IKC. Table X shows the throughput value calculated for AES, RSA, ElGamal and IKC. The bargraph comparison of the throughput of decryption time is shown in fig 5.

TABLE-XI : Comparison of Decryption Time

Time & Throughput	AES	RSA	ElGamal	IKC
Total Average Tikme (Sec)	0.63	0.628	1.07	0.504
Throughput (kb/sec)	280.6349	281.5287	165.2336	350.7937

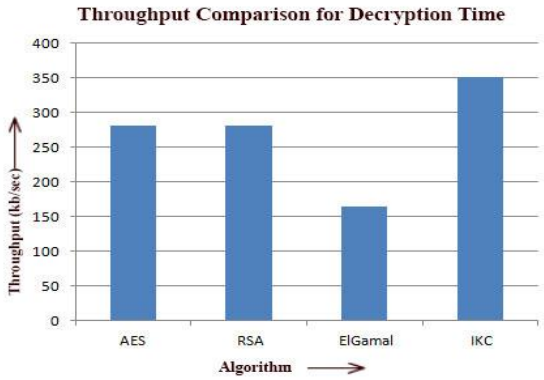

Fig 5 - Throughput of Decryption Time

Security Level: The security effectiveness of IKC based on the objectives of cryptography are (i) Integrity – in IKC this is achieved by registering all the users into the IKC environment with their unique biometric identity, unique identification of the machine codes of the devices used by the users for communication. These identities are also encrypted using the same IKC methodology and stored in the core location of the communication system whether it is IoT, cloud, client-server, or a kick-start pen drive, magnetic card, etc. The encryption and decryption algorithm verify this registration to ensure the integrity of the users. (ii) Data-Authentication & Entity Authentication – the sender and receiver information are transmitted after verification along with the file is the secure shell of IKC. (iii) Confidentiality – is preserved by locking the entire system of communication under the surveillance of IKC. At the instant the user logs into IKC system all the tasks are handled by IKC to assure confidentiality. (iv) non-repudiation – is assured by the device identification and the time stamp used in IKC.

Moreover, IKC proves to withstand the threats of post-quantum cryptography as there are no fixed mathematical calculations used and to be more specific the encryption scheme is itself based on the universal dynamic entity the TIME. There are calculations to calculate the day time month and year from the latitudes on earth, but not with the birth of the message in the brain of the human. Every message communicated is unique which is combined with the uniqueness of the TIME to get the crypt text in IKC. The quantum computers require a finite procedure to solve a problem. but both the message and the encryption criteria are not a finite process it is random and dynamic for a machine to solve.

Key Distribution: In every cryptographic algorithm the key plays the major role. In IKC the key is not just one fixed value to be applied in the algorithm. Instead it is a process in which the key is dynamically generated during the process of encryption or decryption at the time of message is sent.

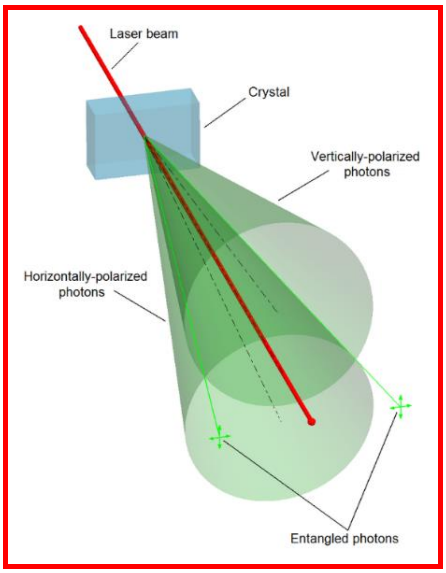

Fig 6 - Quanum Entanglemt

The first main key generation is with the Lipi chakra, which consists of the Tamil Unicode character set. Based on the time at which the message to be sent to the recepient is generated an intermediary code with the help of lipi chakra and thedhiMani chakra is created using the Kuzhuvukuri (The set of dynamic rules based on Tamil alphabets) is created. This is the first level of key generation. The intermediary code is then encrypted using the thedhiMani the time constant and then sent to the nodal point. During the transfer of these messages to the nodal point. The intermediary code is generated at the time of message is created on the sender side and encrypted. The time and the cipher text are sent to the nodal point by means of quantum entanglement method so that the intruder could not access the message in the transmission. Quantum Entanglement is a method developed by ISRO specifically for crypto communication for key exchange. Quantum entanglement is a physical phenomenon where the key to be communicated are sent as entangled photos by polarizing the photons. As this concept is a physical phenomenon it is not discussed in detail in this research work.

Attacks : The encryption decryption is done based on the universally dynamic changing time constant. The time is like a one time pad the time occured once can never occur for ever in the past. The time constat is the actual infinite value of the Universe. Taking into consideration the rare behaviour of the TIME of the nature this algorithm is developed. The travel of is time is infinite and is unique for every millisecond. Hence the strength of Vernam One Time pad is included but with the advantage of dynamically changing time, eliminating the storage overhead and key generation overhead. Thus attacks like brute-force attacks can be eliminated.

All the user's identity is registered in the nodal device and only the registered users and the devices could use IKC hence by physically safeguarding the nodal device the Man-in-the-Middle-Attack, Side-Channel-Attack is controlled.

Quantum attacks based on the mathematical calculations are not possible as there is not mathematical equations or formulas used for encrypting and decrypting.

The Propability of finding the plaintext if the TIME is known is takled by quantum escape.

Quantum Escape: There is always a probability for the question "What if the TIME of encryption is captured by the intruder?" An analysis of this could be done using the probability.

The strokes from the QWERTY keyboard are 35 shifted keys and 35 undhifted keys, 10 numbers totaling to 80 keys.Out of these 80 keys 10 keys are special characters to form combination letters making a total of C(80,10)= 1,646,492,110,120.

The Lipi chakra there are ((18+4) x 12)+2 = 252 characters with a probability of P(252) and in the thedhiMani chakra there are 14 numerals each with 0-9 with

P(10)=1/10. The elements of remaining chakras-probabilities 1/24, 1/60, 1/31, and 1/12. All these cannot be brought into a numerical formula but still follows a rule based on the TIME variation and the TAMIL Grammar. These forms a combination of letters making a total of C(252, 14) = 33,074,458,506,490,886,793,000. The combination of these two huge numbers gives the possibility of occurrence of characters in the cipher text of IKC at a specific time. This is a huge value and is exhausting to find without knowing what the actual message is send from the sender.

This dynamicity in the consistency is the strength of IKC

Conclusion

The proposed IKC architecture in this research work is a novel architecture designed for providing high level of security than the two different crypto algorithms particularly RSA and ELGAMAL algorithms. This IKC architecture is specifically designed to be secure against a cryptanalytic attack by a quantum computer. The proposed and implemented algorithm is tested by establishing as test bed with text files of various sizes and its performance is compared with the two primitive algorithms namely RSA and ELGAMAL. The metrics like the key size, CPU usage, the memory usage, the execution time of the encryption and decryption, the key space and key uniqueness are compared, analysed and inferred. Further, the comparative study proved that there is no overhead in CPU usage, Memory storage thus adding a badge to the proposed IKC algorithm very economical. Adding on to this the efficiency of IKC in terms of key space, key size, key uniqueness and key strength proved IKC to be a secure algorithm that could withstand the vulnerability of primitive crypt algorithms including the post-quantum threats of the crypt system. This proposed IKC architecture in particular, guarantees and proves to satisfy the basic requirements (i) Integrity (ii) Data-Authentication, (iii) Entity-Authentication, (iv) Confidentiality and (v) Non-Repudiation that forms the basis of a valid cryptography algorithm.

Acknowledgement

The authors sincerely express their special thanks and sincere gratitude to Tamil Nadu State Council for Higher Education (TNSCHE) and Department of Science and Technology (DST), India, for sponsoring this research works.

References

[1]. [Rojasree *et* al., 2021a], Rojasree, V, Gnana Jayanthi, J, “Research Intuitions of Stream Cipher”, Accepted for the publication in the Journal of Chengdu University of Technology, ISSN-NO-1671-9727, September 2021.

[2]. [Rojasree *et* al., 2021b], Rojasree, V., Gnana Jayanthi, J., “Research Intuitions of Block Cipher Techniques”, In the International Journal of Management, Technology and Engineering (IJMTE), ISSN: 2249-7455, Volume XI, Issue I, PP:14-20, @ UGC Approved Journal, January-2021.

[3]. https://github.com/SergeyBel/AES

[4]. https://github.com/64nd4lf/simulating-rsa

[5]. https://github.com/kevinhaeni/Crypto.Cipher.ElGamal.Cpp