Wireless Personal Communications (2020) 112:21–36
https://doi.org/10.1007/s11277-019-07013-6

Tamilian Cryptography: An Efficient Hybrid Symmetric Key Encryption Algorithm

R. Geetha[1] · T. Padmavathy[1] · T. Thilagam[1] · A. Lallithasree[1]

Published online: 17 December 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract
Cryptography is one of the most accepted and successful data security methods used at present by most of the organizations. Encrypting the message using natural languages reduces the encryption time and improves the performance. Tamil is a most ancient Dravidian language spoken in the southern part of India. Here we use Tamil language for encrypting the text named as Tamilian Cryptography. Since Tamil language has 247 characters it is very difficult to crack the message. The message to be encrypted is translated to Tamil language, this translated text is then mapped to a randomly generated 2-bit combination of English alphabets. The result of this process is called intermediate cipher (inter cipher). This is an hybrid algorithm since the inter cipher is encrypted using Advanced Encryption Standard algorithm to enhance the confidentiality of the text. Since the Tamilian Cryptography makes use of three phases namely Translation, Mapping and Encryption, it makes the data much more secure than the existing algorithms like blowfish and Data Encryption Standard (DES) and also the person who tries to decrypt must have the knowledge of Tamil language as well the mapping details to see the original data and This algorithm exhibits stronger avalanche effect of 95% which is greater than blowfish and DES. The evaluation of the proposed algorithm shows that it executes faster and has comparatively lesser encryption time, less memory overhead.

Keywords Security · Tamil language · Cryptography · AES · DES · Blowfish

1 Introduction

In the modern era every task has been computerised and lots of data transfer happens. It is highly necessary to protect these data from unauthorized people. Any piece of data in the wrong hands is threat and also people won't use these modern technologies if they feel insecure, So the cryptography domain is a valuable domain and it is the base for many modern applications [1–3].

Cryptography is a means of protecting the information through the usage of codes. Later this protected information can be read only by those for whom the information is intended

✉ R. Geetha
geetha@saec.ac.in

[1] Department of Computer Science and Engineering, S.A. Engineering College, Chennai, India

to read and process. Encryption techniques shields the confidential data such as credit card numbers through encoding and converting the information into unreadable encrypted text. Later on this encoded data can be decrypted or made readable only with the help of a secret key. In computer science, cryptographic techniques involve concepts from mathematics and a series of calculations based on rules called algorithms that make the information more secure and difficult to decrypt. Many deterministic mathematical concepts helps in generating the cryptographic keys, few involve in generating and verifying the digital signatures to protect the privacy of data [4–6].

The two major cryptographic techniques based on the number of keys used are symmetric key and the asymmetric key cryptography. Symmetric key cryptography involves the use of a secret key together with the encryption and decryption algorithms that help protect the message content. The strength of symmetric key cryptography depends on the number of key bits. It is relatively faster than asymmetric key cryptography. A key distribution problem arises because the key must be transferred from the sender to the receiver through a secure channel. Asymmetric key cryptography is also known as public key cryptography because it involves usage of a public key along with secret key. It solves the problem of key distribution as both parties uses different keys for encryption/decryption. It is not feasible to use for decrypting bulk messages as it is very slow compared to symmetric key cryptography.

The most widely used symmetric key encryption algorithm earlier is the Data Encryption Standard (DES) which makes use of 56 bit key and encrypts a data size of 64 bits. Later on since DES was broken by cryptanalyst symmetric block cipher algorithms like Blowfish and Advanced Encryption Standard (AES) were developed. AES is considered to be efficient and reliable since it uses a 128-bit, 192-bit or 256-bit key. Each algorithm has its own drawback with respect to protecting the confidential data.

To make encryption more effective natural language text can be used [7, 8]. Here we use the Indian Tamil language since it has 247 characters and it is very difficult to crack the message. It is used along with AES to make the encryption process more robust. This robust process involves three phases. The first phase is TRANSLATION, where the plain text in English language to Tamil language. This translation is done using Google translator API. This API translates the message in English language to Tamil Language. Tamil language has 247 characters and the person who wants to cryptanalyse the data must be well versed in Tamil. The second phase is MAPPING where each and every Tamil alphabets are mapped to two 2-bit combination of English alphabets and the result of this phase is called intermediate cipher. The final phase is ENCRYPTION, which involves in encrypting the intermediate cipher using a existing encryption algorithm (AES). The data or message goes through all these phases so the data becomes more secure. In this algorithm, since the message is translated to Tamil language which has more number of alphabets (247 alphabets) it makes the algorithm much more efficient than the existing algorithms. Tamilian encryption provides a secure and faster encryption technique that can used by banking sectors and Government organisation to transfer data in a much more secure way.

2 Related Work

Recent encoding algorithms play an important role in guaranteeing the protection of data technology and communications systems. It offers confidentiality, authentication, integrity and non-repudiation. Safiah et al. [9] proposed NASE (Novel Algorithm in Symmetric Encryption) as new formula in rhombohedra encoding that supports Feistel.

Encryption structure with fresh more options which will improve the protection of encrypted knowledge. NASE involves generating a random Block variety size, iterations and completely different keys for every block. The planned formula works for quite one language (for example, English, Arabic, etc.) and is feasible to use double or triple encoding with completely different keys to further improve the security. NASE is extremely quick and simply applied to completely different applications.

Security within the e-commerce dealings may be terribly essential downside till up to now, all designed attacks are coded in English so it is smart to send the message programmed and encoded in a language apart from English.

Gupta et al. [10] introduced a replacement cryptography technique that uses multiple set of language characters. They have worked alone at English and Hindi languages.

In this paper Vidhya and Paul [11] proposed an encryption method using an Indian local language, Malayalam. The proposed method consists of custom Unicode based technique with embedding based on indexing, i.e. the original message is encoded to Malayalam text with custom UNICODE values generated for the Malayalam text. The proposed method is more precise in the encoding process and in the decoding process. The method achieved a precision rate of .95 and decoding rate of .81.

Khairullah [12] presented a simple and novel approach for steganography through transliteration. A phonetic keyboard layout is very popular for writing languages having non-roman alphabets. Bengali, a language spoken by 230 million people, is a fair example and in this work the Bengali digital text is used for data hiding. For several characters in the Bengali alphabet, there are multiple options to represent a character in it's equivalent roman form using a phonetic keyboard layout. The main idea of the proposed method is to exploit this special feature of Bengali phonetic keyboard layouts to hide secret information in form of bits. One of these options can be used to represent the bit '0' and the other option can represent the bit '1' in a document without any risk of understanding by any intermediate user. The results show that the capacity of the method is 1.2%, which is adequate for a text steganography system with very low risk of machine detection. This method can be easily adapted and applied for any other language having non-roman alphabet.

Taha et al. [13] proposed an algorithm for information hiding using Arabic text. The new algorithm improves the length of the secret message that can be embedded in an Arabic text document without affecting its quality as much as possible. The proposed algorithm utilizes different characteristics and properties of Arabic language. It utilizes both the Arabic extension character (Kashida) and small space characters. Each existing Kashida can hide one bit and each existing space can hide three bits. The proposed algorithm was tested for different length stego-text messages. It provides superiority in achieving high capacity hiding ratio in comparison with the most related Kashida-based techniques and spaces-based techniques.

In this paper Hamzah et al. [14] proposed a framework that uses Arabic calligraphy to hide information. The phases of the framework are preparation phase, embedding phase and extraction phase. The embedding phase uses string matching to generate stego text and accompanying letter shapes according to a secret message. The framework also includes

corpus creation and a modification of the Aho–Corasick string-matching algorithm. The Arabic font Naskh was used as a case study. A set of Arabic poetry and proverbs were used as a dataset. The framework was evaluated on capacity and security. Because the visual difference between the cover and the stego-cover must be unnoticeable to the human in any stego-system, the security in this framework is satisfying due there is no cover used. The cover represents the secret message itself and it provides high capacity to hide data also.

Roy and Venkateswaran [15] present a text based steganography technique based on the Vedic Numeric Code. Frequency of the letters in English alphabet in conjunction with Vedic Numeric Code is used for the steganography technique. No separate importance is given for vowels and consonants.

Vijaya Bharati and Jyothi Prasad [16] deals with a practical scheme for encoding an Indian language, telugu. The proposed technique uses the Telugu Text and their attributes to hide the secret message. It is based on the fact that the ordering of the attributes in the Text has no impact on the appearance of the document. This ordering can be used to hide the secret messages efficiently. The proposed technique essentially has three components, key file generation, hiding process and extracting process. The key component of the technique is the generation of Cover Message. The Cover Message is essentially a collection of key combinations stored in the form of rows and columns. These combinations are generating by encryption of the saved Text documents. The attributes combinations used in the Text are used to generate a Cover Message.

Changder et al. [17] presents a new linguistic approach through Indian Languages by considering the flexible grammar structure of Indian Languages. To add more security to the system, instead of hiding the original message it is converted to an irrelevant binary stream by comparing the message bits with the pixel values of an Image. Thereafter, the bits of this binary stream are encoded to some part-of-speech and by creating meaningful sentences starting with a suitable word belonging to the mapped part-of-speech, the proposed method hides the message inside a cover file containing some innocuous sentences. Similarly in receiving side, the algorithm finds the corresponding part-of-speech of the starting word of each sentence and place the bit stream of the mapped part-of-speech to recover the converted message. After comparing these bits with the Image pixels, the algorithm extracted the original message from the cover file. The proposed method exhibits satisfactory result on some Indian Languages like Bengali.

Even though many natural languages are used in the literature for the protection of information from being read by malicious third parties they still lag in protecting the text completely. Our proposed model is considered to be more robust than the other existing algorithms since it uses a language that supports more letters and uses AES in addition to improve the security.

3 Tamilian Cryptography

In the proposed system we propose a new hybrid symmetric key encryption algorithm which is more secure and shows strong avalanche effect (small change in the plaintext bit produces considerable change in the cipher text bits) by having a hard brute force. This Encryption algorithm makes use of Tamil language for encryption. To make the algorithms more secure we make use of the advantage of the Tamil language that it has 247 characters by nature. Thus the cryptanalysts should have at least minimum knowledge of the Tamil language and should try all 247 characters while analysing the cipher text to yield the plain

text leading to have a larger time to identify the plain text during which it expires eventually. These algorithms only generate an intermediate cipher. This is then given to AES algorithms to make it more secure. Even if cryptanalysts find the intermediate cipher he then need to handle AES algorithms to find out the plain text.

Tamil is a Dravidian language spoken predominantly by the Tamils of India and Sri Lanka, and by the Tamil diaspora, Douglas and Chindians. Tamil is an official language of two countries: Sri Lanka and Singapore and the official language of the Indian state of Tamil Nadu.

There are 247 alphabets in Tamil language. This includes 13 vowels and 18 consonants. The remaining alphabets are consonantal vowels which are the combinations of these vowels and consonants as shown in Figs. 1 and 2.

The plain text undergoes three phases in the proposed system the translation phase, the mapping phase and the encryption phase. The plain text in English language is first translated to Tamil language using an language translation API. Then the translated text is produced. This translated Tamil text enters the next phase. In the next phase each letter of the Tamil text is mapped with two English alphabets which is generated randomly, the result of this phase is called intermediate cipher. Now the intermediate cipher must be encrypted this is done using AES Encryption algorithm.

To decrypt the encrypted text also consists of the same three phases but in reverse order. First the encrypted text (AES) is decrypted which gives the intermediate cipher. The intermediate cipher is mapped with 2-bit key which gives the translated text (text in Tamil). Now the out from the previous phase is translated to English to get the original message.

AES uses longer keys, such as 128, 192 and 256 bits for encryption. Therefore, it makes the AES algorithm more robust against piracy. It is the most common security protocol used for a wide variety of applications, such as wireless communications, financial transactions, e-commerce, encrypted data storage, etc. It is one of the most widespread commercial and open source solutions used throughout the world. No one can hack your personal information. For 128 bits, about 2128 interruption attempts are required. This makes hacking very difficult, as it is a very secure protocol. Thus AES adds additional advantage to the proposed system. The advantages of the proposed system involves difficult brute force, reduced hacker count and regional language.

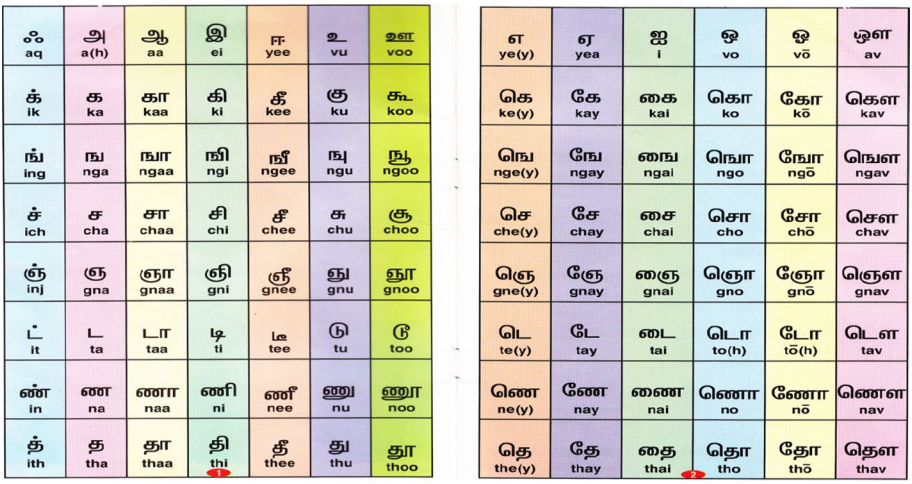

Fig. 1 Tamil language letters

Consonants

Tamil	Translit.	Category	Sound
க்	k	vallinam	[k] or [g]
ங்	ṅ / ng	mellinam	[ŋ]
ச்	c / ch / s / sh	vallinam	[c] or [s]
ஞ்	ñ / nj	mellinam	[ɲ]
ட்	ṭ / T / t	vallinam	[ʈ] or [ɖ]
ண்	ṇ / N	mellinam	[ɳ]
த்	t / th	vallinam	[t̪] or [ð]
ந்	n	mellinam	[n]
ப்	p	vallinam	[p] or [b]
ம்	m	mellinam	[m]
ய்	y	idaiyinam	[j]
ர்	r	idaiyinam	[ɾ]
ல்	l	idaiyinam	[l]
வ்	v	idaiyinam	[ʋ]
ழ்	ḻ / zh / z	idaiyinam	[ɻ]
ள்	ḷ / L	idaiyinam	[ɭ]
ற்	ṟ / R	vallinam	[ɾ]
ன்	ṉ / n	mellinam	[n]

Vowels

Tamil	Translit.	Sound
அ	a	[ə]
ஆ	ā / A / aa	[aː]
இ	i	[ɪ]
ஈ	ī / I / ii	[iː]
உ	u	[ʊ]
ஊ	ū / U / uu	[uː]
எ	e	[ɛ]
ஏ	ē / E / ee	[eː]
ஐ	ai	[ai̯]
ஒ	o	[ɔ]
ஓ	ō / O / oo	[oː]
ஔ	au	[əu̯]

Compound forms

Tamil	Translit.	IPA
க	ka	[kə]
கா	kā	[kaː]
கி	ki	[kɪ]
கீ	kī	[kiː]
கு	ku	[kʊ]
கூ	kū	[kuː]
கெ	ke	[kɛ]
கே	kē	[keː]
கை	kai	[kai̯]
கொ	ko	[kɔ]
கோ	kō	[koː]
கௌ	kau	[kəu̯]

Grantha letters

Tamil	Translit.	Sound	Traditional substitute
ஜ்	j	[ʤ]	ச் (c)
ஷ்	ṣ / ś / sh	[ʂ] or [ɕ]	ச் (c)
ஸ்	s	[s]	ச் (c)
ஹ்	h	[h]	க் (k)
க்ஷ்	kṣ / kś / ksh	[kʂ] or [kɕ]	ட்ச் (tc)

Fig. 2 Tamil consonants, vowels and compound forms

3.1 Translation

The plain text is translated from English language to Tamil language, this translation is done with the help of Google translator API. Numbers cannot be translated with this API, hence the numbers are first converted to words and then it is translated to Tamil.

A language translator translates a text written in one language to other language. It is a very helpful tool allows people to understand text written in unknown language. Google translator is very famous and the best translator available, it is a lot easier to use and also free of cost.

3.2 Mapping

Here in mapping each and every Tamil alphabets are mapped to two 2-bit combination of English alphabets and the result of this phase is called intermediate cipher. Every occurrence of a letter does not have the same mapping there are two 2-bit combination for every Tamil alphabet each occurrence may have one of the two 2-bit combination. The selection of this 2-bit combination of alphabets is made random to make the algorithm effective. It is literally impossible or at least very difficult for the hackers to get the data or text than the

traditional algorithms and even though the hacker could crack this mapping he would only get the result of the translation phase i.e. text in Tamil language.

3.3 Logic of Mapping

Generating two bit combination of alphabets results in generating 676 two bit characters as shown below. Here the table shows all the two bit combinations of English alphabets. These combinations are used to map Tamil characters which are translated from the plain text.

Here in mapping each and every Tamil alphabets are mapped to two 2-bit combination of English alphabets and the result of this phase is called intermediate cipher as shown in Fig. 3. Every occurrence of a letter does not have the same mapping there are two 2-bit combination for every Tamil alphabet each occurrence may have one of the two 2-bit combination.

Mapping is to all the 247 Tamil characters as shown in Fig. 4.

3.4 Encryption and Decryption

This is the final module here we encrypt the intermediate cipher, which is the result of mapping phase using a existing encryption algorithm (AES). Advanced Encryption Standard (AES) is considered more reliable because it uses a 128-bit, 192-bit or 256-bit key. Combining the new algorithm with the existing AES algorithm a highly secure encryption is created. AES algorithm is used since it is more Secure, consumes less memory and more flexible.

	BA	CA	DA	EA	…...	….....	….....	YA	ZA
AB	BB	CB						YB	ZB
AC	BC	CC						YC	ZC
AD	BD	CD	DD					YD	ZD
AE	...			EE				YE	ZE
AF	...				FF			YF	ZF
...	...					GG		YG	...
...	...							….	...
….	...							YY	...
AZ	BZ	CZ	DZ	EZ	…….	……..	…….	……..	ZZ

Fig. 3 Mapping table logic

க	AS,GB
ச	PL,AS
ன	RS,TA
…..	…..
…..	…..
அ	YU,SB
ஆ	KY,MA
இ	RX,NV
ஈ	VQ,JG
…..	…..

Fig. 4 Bit mapping

To hack the data the hacker must decrypt the AES encryption, even though one could be successful in that then 2-bit mapped intermediate cipher must be compromised which is random and changes frequently. If even it was compromised the hacker would only get the translated version (TAMIL TEXT) and must know tamil to read the actual data.

3.5 AES Encryption and Decryption

AES algorithm is an iterated block decipher algorithm with a fixed block size of 128 and a variable key length. The AES algorithm operates on 128 bits of data and generates 128 bits of output. The length of the key used to decrypt this input data can be 128, 192 or 256 bits.

AES encryption makes use of four transformations namely substitute bytes, Shift rows, mix columns and add round key as shown in Fig. 5. The number of rounds chosen are 10 where each round makes use of all the four aforementioned transformations except the last round. The last round uses only three transformation and omits the mix column transformation.

AES decryption too makes use of four transformations used by the encryption algorithm, but in the reverse order. As like encryption the last round uses only three transformations and omits the mix column transformation.

Here user A is the sender and user B in the receiver. The plaintext go through three stages as shown in Fig. 6.

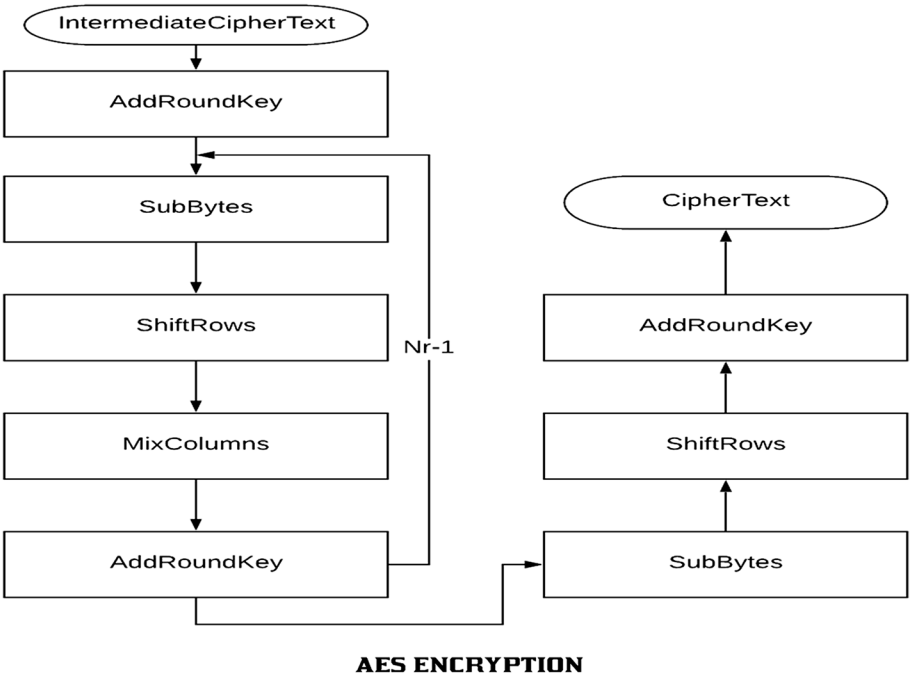

Fig. 5 AES encryption and decryption transformations

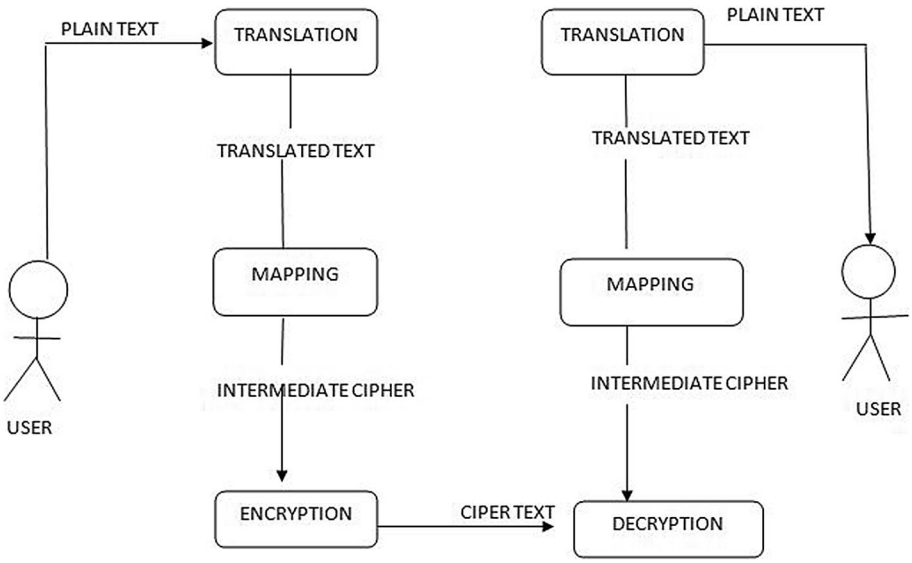

Fig. 6 Phases of Tamilian Cryptography

The plain is first translated to Tamil. The translated text is mapped with 2-bit key. The mapped intermediate cipher is encrypted using AES. The process is reversed for decryption.

The overall architecture of our proposed model (encryption and decryption) is shown in Fig. 7.

The Pseudocode of our proposed model (encryption and decryption) is depicted as follows

---

Pseudocode - Tamilian Encryption

Input: The plain text to be encrypted

Output: The cipher text

1. Define function **tamilian_encryption(plain text)**
2. **While** text in plain text
   Translate each text to Tamil and move to next text
3. **End while**
4. **While text in translated text**
   Map text to 2-bit combination and move to next text
5. **End while**
6. **While text in mapped text**
   AES_Encryption(text)
7. **End while**
8. Return Cipher text
9. Define function **AES_encryption(plain text)**
10. Perform the steps in AES algorithm with the plain text.
    Init_permutation(plain text)
    Sub_bytes(plain text)
    shift_rows(plain text)
    mix_columns(plain text)
    add_round_key(plain text)
11. Return Cipher text
12. Define main
13. Get the input plain text from the user
14. Call the function tamiilian_encryption(plain text)

Pseudocode - Tamilian Decryption

Input: The cipher text to be decrypted

Output: The plain text

1. Define function **tamilian_decryption(cipher text)**
2. **While text in cipher text**
   AES_decryption(text)
3. **End while**
4. **While text in mapped text**
   Perform the reverse mapping
5. **End while**
6. **While** text in translated text
   Translate each text to plain text and move to next text
7. **End while**
8. Return Plain text
9. Define function **AES_decryption(cipher text)**
10. Perform the steps in AES algorithm with the cipher text.
    add_round_key(cipher text)
    inverseshift_rows(cipher text)
    inverseSub_bytes(cipher text)
    inversemix_columns(cipher text)
    Inverse Init_permutation(plain text)
11. Return Plain text
12. Define main
13. Get the input Cipher text
14. Call the function tamiilian_decryption(cipher text)

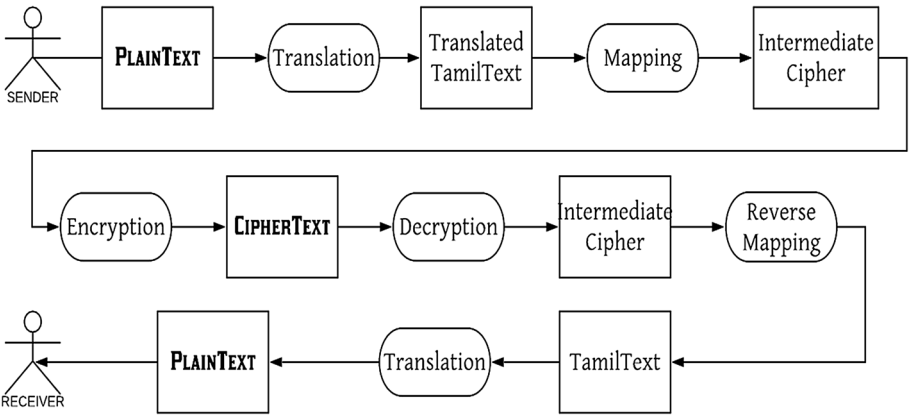

Fig. 7 Tamilian Cryptography (encryption and decryption)

4 Performance Evaluation

The performance of our proposed model has been evaluated using python 3.7.4. Here is the performance evaluation comparison of Tamilian Cryptography with DES and Blowfish algorithms in terms of encryption time, memory overhead and avalanche effect.

4.1 Encryption Execution Time

Execution time of any cryptographic algorithm involves encryption and decryption time which involves changing the plain text to cipher text and vice versa. Since Tamilian cryptography phases like translation and mapping involves translating the plain text from English to tamil language and mapping of tamil characters to English letters which is a simple process it takes less time to encrypt a file of size 25 kb when compared to DES and Blowfish as shown in Fig. 8. The encryption time using Tamilian cryptography, Blowfish and DES for varying plain text size based on the number of characters is shown in Fig. 9.

4.2 Memory Usage

Memory allocation is a function assigned to computer programs and services with physical or virtual memory space. Memory allocation is the process of allocating part of the system memory or the complete part of the program to execute program and processes. The memory allocation is achieved through the process called memory management.

In storage systems, throughput refers to either the amount of data that can be received and written to the storage medium or read from media and returned to the requesting system, typically measured in mega bytes per second (MBPS).

Different encryption algorithms make use of different number of variable as per their requirement for execution thus varies the memory allocation for the same. Here is the detailed memory occupation of different encryption algorithms. Figure 10 shows that the memory requirement of our proposed model is comparatively lesser than Blowfish and DES since the translation and mapping phases consumes less space.

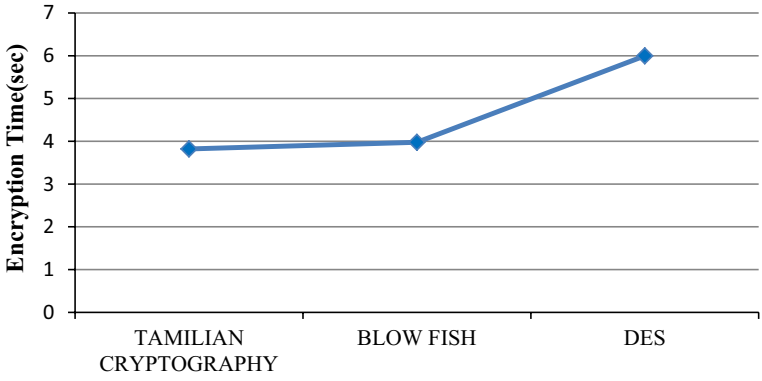

Fig. 8 Encryption execution time of various algorithms

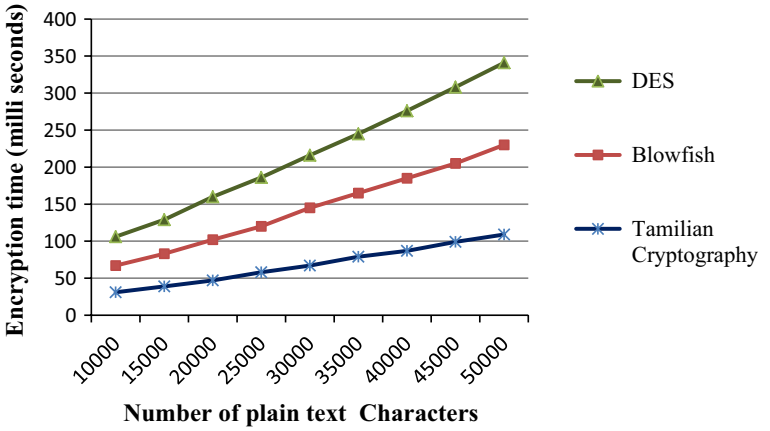

Fig. 9 Encryption execution time for varying characters

4.3 Avalanche Effect

Avalanche Effect is one of the desirable properties of any block cipher cryptographic algorithm. It insists the algorithm to change as many number of bits of cipher text as possible if even a single bit of plaintext is changed. A strong avalanche effect is desirable for a good cryptographic algorithm. Avalanche effect is calculated and expressed in percentage as shown in Eq. 1

$$\text{Avalanche effect}\,(\%) = \frac{\textit{Total number of altered bits in the ciphertext}}{\textit{Total number of bits in the ciphertext}} * 100 \qquad (1)$$

The security of the proposed algorithm is measures with the avalanche effect. Figure 11 shows that our proposed model shows a better avalanche effect for 16 characters plaintext while changing the initial, intermediate and final bit positions since it makes use of AES algorithm when compared to Blowfish and DES.

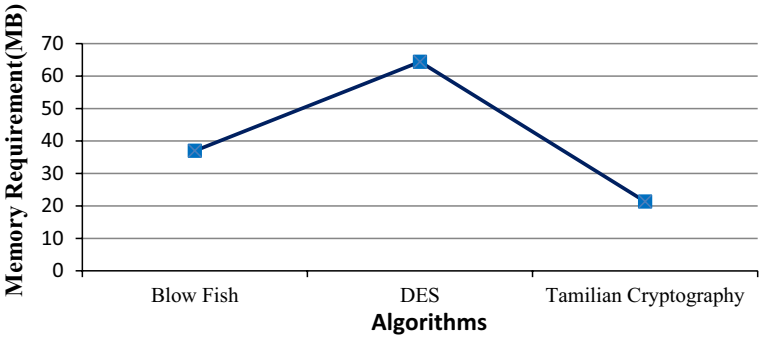

Fig. 10 Memory requirement of various algorithms

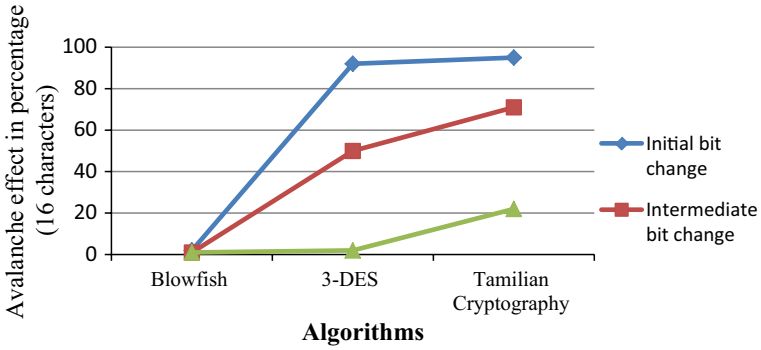

Fig. 11 Avalanche effect of various algorithms

5 Conclusion

The development of technology has made people to feel insecure about their data. Millions of data is generated by people every second and it must be protected from intruders. A new hybrid and efficient symmetric key cryptographic algorithm named Tamilian Cryptography has been proposed in this paper. By encrypting the data with Tamilian Cryptography the transfer of data across the network can be made secure. The evaluation of the proposed algorithm shows that it is superior in terms of memory usage, time for encryption and produces strong avalanche effect when compared to DES and Blowfish algorithms.

References

1. Sapkal, K., & Shrawankar, U. (2016). Transliteration of secured SMS to Indian regional language. *Procedia Computer Science, 78,* 748–755.
2. Jing, X., Hao, Y., Fei, H., & Li, Z. (2012). Text encryption algorithm based on natural language processing. ACM Digital library, pp. 670–672.
3. Hughes, J. P., Hibbard, E. A., Cole, J. L., & Anderson, C. (2011). 1619.2-2010—IEEE standard for wide-block encryption for shared storage media. IEEE. https://doi.org/10.1109/ieeestd.2011.5729263.
4. Baskar, C., Balasubramani, C., & Manivannan, D. (2016). Establishment of light weight cryptography for resource constraint environment using FPGA. *Procedia Computer Science, 78,* 165–171.
5. Chatterjee, R., Bonneau, J., Juels, A., & Ristenpart, T. (2015). Cracking–resistant password vaults using natural language encoders. In *IEEE symposium on security and privacy*. https://doi.org/10.1109/sp.2015.36.
6. Kurniawan, D. H., & Munir, R. (2016). Double chaining algorithm, a secure symmetric-key encryption algorithm. In *International conference on advanced informatics: concepts, theory and application (ICAICTA)*. https://doi.org/10.1109/icaicta.2016.7803097.
7. Ogden, W. C., & Davis, M. W. (2002). Improving cross-language text retrieval with human interactions. In *Proceedings of the 33rd annual Hawaii international conference on system sciences*. https://doi.org/10.1109/hicss.2000.926726.
8. Nagarhalli, T. P., Bakal, J. W., & Jain, N. (2016). A survey of Hindi text steganography. *International Journal of Scientific & Engineering Research, 7,* 55–61.
9. Baker, S. I. B., & Al-Hamami, A. H. (2017). Novel algorithm in symmetric encryption (NASE). In *IEEE international conference on new trends in computing sciences (ICTCS)*. https://doi.org/10.1109/ictcs.2017.54.
10. Gupta, A., Semwal, S., & Johari, R. (2016). METHS: Mapping from English language to Hindi language for secure commercial transactions. In *International conference on computing, communication and automation (ICCCA)*. https://doi.org/10.1109/ccaa.2016.7813700.

11. Vidhya, P. M., & Paul, V. (2015). A method for text steganography using Malayalam text. *Procedia Computer Science, 46,* 524–531.
12. Khairullah, M. (2019). A novel steganography method using transliteration of Bengali text. *Journal of King Saud University Computer and Information Sciences, 31*(3), 348–366. https://doi.org/10.1016/j.jksuci.2018.01.008.
13. Taha, A., Hammad, A. S., & Selim, M. M. (2018). A high capacity algorithm for information hiding in Arabic text. *Journal of King Saud University Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2018.07.007.
14. Hamzah, A. A., Khattab, S., & Bayomi, H. (2019). A linguistic steganography framework using Arabic calligraphy. *Journal of King Saud University Computer and Information Sciences*. https://doi.org/10.1016/j.jksuci.2019.04.015.
15. Roy, S., & Venkateswaran, P. (2013). A text based steganography technique with Indian root. *Procedia Technology, 10,* 167–171.
16. Vijaya Bharati, P., & Jyothi Prasad K. S. S. (2016). Cryptic transmission of Telugu text. In *International conference on information communication and embedded systems (ICICES)*. https://doi.org/10.1109/icices.2016.7518877.
17. Changder, S., Ghosh, D., & Debnath, N. C. (2010). Linguistic approach for text steganography through Indian text. In *International conference on computer technology and development*. https://doi.org/10.1109/icctd.2010.5645862.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Dr. R. Geetha received B.E. degree in Computer Science and Engineering from Madras University in 1999, M.E. in Computer Science and Engineering from Anna University in 2006 and Ph.D., in 2017 from School of Computing and Information Technology, Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology (Vel Tech Dr. RR & Dr. SR Technical University), India. She has over 20 years of teaching experience and working currently in S.A. Engineering College as Professor. Her research interests include Wireless Networks, Security Schemes in Wireless networks. She is a Life member of Indian Society for Technical Education (ISTE) and member of Computer Society of India (CSI). She is the author/coauthor of several research papers in international conferences and journals.

T. Padmavathy received her B.E. Degree in Computer Science And Engineering from Anna University in 2010 and M.E. in Computer Science And Engineering from Anna University in 2015. Her research interests include computer and network security, wireless network. She is the author/co-author of several paper in international conference and journals.

T. Thilagam received B.E. degree in Computer Science and Engineering from Anna university in 2010, M.E. in Computer Science and Engineering from St.Peter's University in 2012. She is an Assistant Professor in the Department of Computer Science and Engineering, S.A. Engineering College, Chennai-77, India. She is going Ph.D. in Veltech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai, India. Her current research interests include cloud computing, network security and privacy, digital image processing and Distributed Data mining.

A. Lallithasree received her B.E. Degree in Computer Science And Engineering from Anna University in 2012 and M.E. in Computer Science And Engineering from St Peters University in 2014. Her research interests include computer and information security, networking in secure database. She has published several papers in international conference and journals.