Scientific Tamil Lexicon: The Revelation of Cryptographic Connection Between Tamil Language and Galois Field

A Preprint

Shan Suthaharan
Department of Computer Science
University of North Carolina at Greensboro
Greensboro, NC 27402
s_suthah@uncg.edu

January 16, 2022

Abstract

This paper presents a computational framework that helps enhance the confidentiality protection of communication in cybersecurity by leveraging the scientific properties of the Tamil language and the advanced encryption standard (AES). It defines a product set of vowels and consonants sounds of the Tamil language and reveals its connection to Hardy-Ramanujan prime factors and Tamil letters as a one-to-one function. It also reveals that the letters of the Tamil alphabet, combined with the digits from 1 to 9, form a Galois field of 2^8 over an irreducible polynomial of degree 8. In addition, it implements these two mathematical properties and builds an encoder for the AES algorithm to transform the Tamil texts to their hexadecimal states, and replace the pre-round transformation module of AES. It empirically shows that the Tamil-based encoder enhances the cryptographic strength of the AES algorithm at every step of its encryption flow. The cryptographic strength is measured by the runs test scores of the bit sequences of the ciphers of AES and compared with that of the English language. This modeling and simulation approach concludes that the Tamil-based encryption enhances the cryptographic strength of AES than English-based encryption.

Keywords Tamil language · Finite field · Galois field · Cryptography · Encryption · Pattern recognition · Confidentiality · Hardy-Ramanujan number · Planets · Shannon's idea · Confusion · Diffusion

1 Introduction

Tamil language has many scientific wonders; however, its scientific properties that are related to cryptographic strengths are yet to be studied in detail. The letters in the Tamil alphabet and their phonetic alignment to form sentences may play a major role in developing a robust confidentiality protection for cybersecurity. Confidentiality protection is one of the essential requirements in cybersecurity. The advanced encryption standard (AES) is an algorithm that is currently standardized and widely used in many cybersecurity applications. The AES algorithm first encodes a message to hexadecimal states and then performs several encryption tasks that generate confusion and diffusion properties–the two major cryptographic properties defined based on Shannon's idea, Dodis et al. [2016]. Hence the message encoding is one of the essential tasks in the encryption processes; however, the encoding of the English language text using the positional information of the letters in the alphabet develops a vulnerability at that step, which then may propagate through the remaining encryption steps of the AES algorithm. English alphabet with only a set of 26 letters generates a significant redundancy; hence, it is difficult to develop a robust encoding technique. Therefore, it is important to study this problem using other languages by leveraging their scientific properties. Tamil language is a perfect candidate for this purpose; hence, its cryptographic properties are studied in this paper.

It is interesting to learn how the vowels and consonants are named in Tamil. The vowels are called Uyir Ezhuthu, meaning the soul letters, and the consonants are called Mei Ezhuthu, meaning the body letters. One way to interpret this is that the vowels give a soul to the context of the Tamil language while the consonants give a body to the context. However, another way of interpreting this may be that the Tamil vowels sounds–when spoken with proper pronunciation–give benefits to the soul and the consonants sounds give benefits to the body, in turn may give many health benefits by enhancing circadian rhythms (it is just a hypothetical statement). It is also interesting to learn how other letters in the Tamil alphabet are formed by combining the vowels and consonants, and they are called UyirMei Ezhuthu meaning soul+body letters.

Tamil language–while providing a phonetic expressive communication–may bring scientific benefits to many applications that include natural language processing, information security, and artificial intelligence. It carries many hidden computational constituents that must be studied and understood in scientific perspectives to be adapted in modern applications. For instance, the power of the divine sound and the vibration of the word ஓம் (ōm) must be studied in depth to understand its healing power. The word ஓம் is regularly used by Tamils (in particularly by Tamils in Sri Lanka) in daily dialogues to express an agreement that provides the same meaning as the word "yes" in English dialogues. The well-known Indologist, Asko Parpola, also mentioned in his book that the word ஓம் contributes to Dravidian languages Parpola [2015]. More information about his descriptions and discussions on the word ஓம் can be found in the page 170 of his book Parpola [2015]. Hence the Tamil language, if properly spoken, may also bring health benefits to the speakers of Tamil language. Today, we can witness the contribution of the ஓம் sound to physical and mental health through yoga in the modern society; however, its scientific significance is still to be explored in detail.

Tamil language occupies the 17^{th} position in the top 200 most spoken languages in the world–according to the current statistics available at `https://www.ethnologue.com/guides/ethnologue200` (accessed online on 01/09/2022). This website also informs that the Tamil language is spoken by 85 million people in the world. The first top 20 languages are: English (1348 M), Mandarin Chinese (1120 M), Hindi (600 M), Spanish (543 M), Standard Arabic (274 M), Bengali (268 M), French (267 M), Russian (258 M), Portuguese (258 M), Urdu (230 M), Indonesian (199 M), Standard German (135 M), Japanese (126 M) Marathi (99 M), Telugu (96 M), Turkish (88 M), Tamil (85 M), Yue Chinese (85 M), Wu Chinese (82 M), and Korean (82 M). Among these top 20 languages, Tamil may be the only language that has the number of letters in the alphabet that is closer to 256 such that it can meaningfully and securely occupy an 8-bit processor in a computing device. The 8-bit processor is a key player in many secure storage and computing applications in computer science that include cryptography and network security applications, and secure reconfigurable hardware applications, Gura et al. [2004], Suthaharan [2008], Rodriguez-Henriquez et al. [2007].

The paper by Bharatharaj et al. [2021] also states that a large number of people speaks Tamil language from several countries that include Fiji, India, Malaysia, Mauritius, Singapore, South Africa and Sri Lanka. In the last three decades a large number of Tamil community has migrated to Australia, Europe, and North America; hence, it is expected that this number could be higher. Therefore an advancement in Tamil language research can benefit the globalization. In recent years, there has been a growing interest in its applications to cryptography (Rajendiran et al. [2013], Geetha et al. [2019]), speech recognition (Lokesh et al. [2019]), character recognition (Ulaganathan et al. [2020], Sigappi and Palanivel [2012]), and semantics analysis (Selvam et al. [2008]). Most recently, the mental health research, related to pandemic, has also been performed in Tamil population where the Tamil language could have an indirect influence in the study (Ashok et al. [2019], Bharatharaj et al. [2021]), in addition to the cultural and geographical contributors.

The focus of this paper is in the application of Tamil language to cryptography. Tamil language has many hidden patterns that might be useful for cryptographic applications. With the reduction in redundancy–caused by the large number of letters in the alphabet–Tamil language can provide stronger confidentiality protection than English language, when applied to encrypt messages and share them over the Internet. In Geetha et al. [2019], for example, it has been stated that the 247 letters in Tamil alphabet makes it difficult to crack the Tamil message hidden in a cipher. They mapped (encoded) the translated Tamil texts to randomly selected 2-bit combination of English letters and used the AES algorithms to encrypt the encoded Tamil texts. However, limited studies have been performed to understand the mathematical and scientific properties of Tamil language and their connections to develop interpretable cybersecurity environment.

The AES algorithm helps protect the confidentiality of a block of texts by applying encryption modules Heron [2009]. As the encryption standard, it is widely used in many applications that require confidentiality protection. The AES algorithm adapts the concept of Galois Field of 2^8 with an irreducible polynomial of degree 8. It takes a block of characters and encrypts it through a sequentially established mathematical transformations while adding a set of keys so that the plaintext of a ciphertext cannot be recovered by an

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	ஃ (akh)	அ (u)p	ஆ (a)rt	இ (i)t	ஈ (ea)t	உ p(u)t	ஊ pr(oo)f	எ (e)nter	ஏ (a)te	ஐ (ai)sle	ஒ (o)pt	ஓ (o)at	ஔ (ou)t
2	க் tal(k)	க	கா	கி	கீ	கு	கூ	கெ	கே	கை	கொ	கோ	கௌ
3	ங் ba(ng)	ங	ஙா	ஙி	ஙீ	ஙு	ஙூ	ஙெ	ஙே	ஙை	ஙொ	ஙோ	ஙௌ
4	ச் cat(ch)	ச	சா	சி	சீ	சு	சூ	செ	சே	சை	சொ	சோ	சௌ
5	ஞ் s(inge)	ஞ	ஞா	ஞி	ஞீ	ஞு	ஞூ	ஞெ	ஞே	ஞை	ஞொ	ஞோ	ஞௌ
6	ட் pu(t)	ட	டா	டி	டீ	டு	டூ	டெ	டே	டை	டொ	டோ	டௌ
7	ண் do(ne)	ண	ணா	ணி	ணீ	ணு	ணூ	ணெ	ணே	ணை	ணொ	ணோ	ணௌ
8	த் ma(th)	த	தா	தி	தீ	து	தூ	தெ	தே	தை	தொ	தோ	தௌ
9	ந் te(nth)	ந	நா	நி	நீ	நு	நூ	நெ	நே	நை	நொ	நோ	நௌ
10	ப் to(p)	ப	பா	பி	பீ	பு	பூ	பெ	பே	பை	பொ	போ	பௌ
11	ம் To(m)	ம	மா	மி	மீ	மு	மூ	மெ	மே	மை	மொ	மோ	மௌ
12	ய் p(ie)	ய	யா	யி	யீ	யு	யூ	யெ	யே	யை	யொ	யோ	யௌ
13	ர் butte(r)	ர	ரா	ரி	ரீ	ரு	ரூ	ரெ	ரே	ரை	ரொ	ரோ	ரௌ
14	ல் p(ill)	ல	லா	லி	லீ	லு	லூ	லெ	லே	லை	லொ	லோ	லௌ
15	வ் lo(ve)	வ	வா	வி	வீ	வு	வூ	வெ	வே	வை	வொ	வோ	வௌ
16	ழ் nood(le)	ழ	ழா	ழி	ழீ	ழு	ழூ	ழெ	ழே	ழை	ழொ	ழோ	ழௌ
17	ள் pu(ll)	ள	ளா	ளி	ளீ	ளு	ளூ	ளெ	ளே	ளை	ளொ	ளோ	ளௌ
18	ற் ki(te)	ற	றா	றி	றீ	று	றூ	றெ	றே	றை	றொ	றோ	றௌ
19	ன் p(in)	ன	னா	னி	னீ	னு	னூ	னெ	னே	னை	னொ	னோ	னௌ

Figure 1: It illustrates the alphabet of the Tamil language (It is defined as HRT-Grid in this paper). There are 247 Tamil letters that make 13×19 sounds by directly using 12 vowels (first row; columns 2 to 13), 18 consonants (first column; rows 2 to 19, and a special letter akh (first cell) and combining them. It also presents the pronunciation for the vowels and consonants, as a guide, to pronounce other letters. However, the readers are encouraged to refer Schiffman [1999] for detailed information on the Tamil language pronunciation.

adversary. In essence, it uses multiple rounds of encryption where each round has a set of transformations that include SubBytes, ShiftRows, MixColumns, and AddRoundKey to achieve its encryption goals. These transformations are operated on a Galois field of 2^8 with an irreducible polynomial, multiplication and addition operators, and multiplicative inverse; hence, deliver cryptographic strengths to the encryption flow. The AES, when applied to Tamil texts, it is expected to provide stronger ciphers than English, since Tamil alphabet has much larger number of letters than English. However, the approach proposed in this paper adds the scientific and mathematical properties of Tamil language to the encryption flow of the AES algorithm.

2 Objectives

The main goal of this paper is to study the cryptographic properties of the Tamil language and report the analytical findings while delivering an approach that utilizes these properties to improve the encryption flow of AES with the Tamil language. The goal includes the establishment of mathematical connections between the Tamil language alphabet, Hardy-Ramanujan number 1729, and the Galois field of 2^8 by revealing the latent features of the Tamil language that are useful for cryptography applications, including confidentiality protection, authentication, and integrity of messages in a Tamil text. In other words, this research is to computationally (i.e., with modeling and simulation) show that the encryption of Tamil texts using the

0	1	2	3	4	5	6	7	8	9	10	11	12
A	B	C	D	E	F	G	H	I	J	K	L	M
13	14	15	16	17	18	19	20	21	22	23	24	25
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Figure 2: In the AES algorithm, the letters in the English alphabet are encoded by using their location profiles in the English Alphabet. This figure illustrates the assigned numbers for the letters in the English alphabet in the ordered list: The letter "A" is assigned with the location index "0", "B" is assigned with the location index "1", ..., "Z" is assigned with the location index "25", Stallings [2006], Forouzan [2008].

AES algorithm can provide much stronger confidentiality protection than the use of the English texts with AES. Hence, our first objective is to define a product set that is formed by the ordered pairs of the sounds of vowels and consonants of the Tamil language and reveal a mathematical connection between this set of sounds and the prime factors of the Hardy-Ramanujan number and the letters of the Tamil alphabet. The second objective is to show that the letters of the Tamil alphabet, when mixed with the digits from 1 to 9, can develop a Galois Field of 2^8 over an irreducible polynomial of degree 8. The third objective is to construct an encoder using these mathematical properties of the Tamil language and study cryptogrpahic strengths induced by the encoder at each step of AES–without the AddRoundKey operations–within each round and between the rounds of AES. The final objective is to compare the cryptographic strengths of the English alphabet and Tamil alphabet under the operations of AES by using the randomness of the bits sequence, determined by the runs test, as the quantitative measure of the cryptographic strength of a cipher.

3 Modern Encryption

The modern encryption technique leverages the intrinsic cryptographic properties of a finite field and the Shannon's idea, Stallings [2006], Dodis et al. [2016]. The 8-bit words and the corresponding integer representations performed in computer applications triggered the use of the finite field called the Galois field GF(2^8) with an irreducible polynomial of degree 8, Heron [2009], Nasser et al. [2016]. They also contributed to the development of the strong AES algorithm that provides cryptographic strengths via the Galois field operations, the multiplicative inverse, the add round keys module, and the confusion and diffusion properties.

3.1 Galois Field

A Galois field is a set of n-bit ($n > 0$) integers together with two operations addition and multiplication that are defined over an irreducible polynomial of degree n that satisfies the following axioms of a finite field, Stallings [2006]: closure under addition, associative law of addition, commutative law of addition, existence of additive identity, existence of additive inverse, closure under multiplication, associative law of multiplication, commutative law of multiplication, distributive law, existence of multiplicative identity, existence of multiplicative inverse, and no zero dividers. The mathematical operations in Galois field is generally represented by the addition and multiplication tables along with additive and multiplicative inverses while satisfying the aforementioned axioms. The readers are encouraged to refer the scientific materials in Stallings [2006], Forouzan [2008], Dodis et al. [2016] to obtain detailed information. It is well understood that each element of the Galois field appears the same number of times in the addition and multiplication tables (except the zero in the multiplication table) and this property makes the Galois field a perfect mathematical tool for the modern cryptographic systems. In other words, this property of Galois field notably contributes to the security features of AES, because the probability of occurrence of each integer for an operation over a Galois field is the same and it is equal to $1/2^n$, where $n = 8$ for AES. For a large value of n, this probability is very small and consequently the cryptographic strength becomes stronger. The AES algorithm is one of the most popular encryption algorithms that are currently used by the industry and government agencies.

Alphabet	A	B	C	D	E	F	G	H	I	J	K	L	M
Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12
Hex	00	01	02	03	04	05	06	07	08	09	A	B	C

Alphabet	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Decimal	13	14	15	16	17	18	19	20	21	22	23	24	25
Hex	D	E	F	10	11	12	13	14	15	16	17	18	19

Figure 3: An encoder for the English Alphabet (Decimal to Hex), Forouzan [2008].

I	A	M	G	O	I	N	G	T	O	S	C	H	O	O	L
08	00	0C	06	0E	08	0D	06	13	0E	12	02	07	0E	0E	0B

Figure 4: An encoded English sentence (in Hexadecimal forms).

3.2 AES Algorithm

In the AES algorithm, the Galois field of 8-bit integers represents a finite field that consists of 256 (i.e. 2^8) 8-bit integers $0, 1, 2, \ldots, 255$ and two arithmetic operations (addition and multiplication) on the 8-bit integers that are represented by a pair of hexadecimal numbers $(\texttt{00}, \texttt{01}, \texttt{02}, \ldots, \texttt{FF})$ and defined over an irreducible polynomial $R(x) : x^8 + x^4 + x^3 + x + 1$ of degree 8, Daemen and Rijmen [2001]. There are 30 possible irreducible polynomials attached to an 8-bit integer, Stallings [2002]. As stated in Stallings [2006], the developers of AES have selected the first irreducible polynomial on the list to construct the addition and multiplication tables, and additive and multiplicative inverses. The AES algorithm, as stated earlier in this paper, assumes each byte (8-bits) of its input bits (plaintext that is the message to be encrypted) is an element in the Galois field of 8-bit integers in which the addition and multiplication operations are defined over the irreducible polynomial $R(x) : x^8 + x^4 + x^3 + x + 1$ of degree 8. It is evident from the design of AES algorithm, Heron [2009] and Stallings [2006], that the AddRoundKey operation is added to each round to make the the inverse transformation highly resistant to the adversarial attacks that attempt to recover plaintext from a given ciphertext. In this paper, our goal is not to develop another encryption technique rather to show the cryptographic strengths of the Tamil language that could support the AES algorithm to become much stronger with less number of rounds and intermediate encryption.

4 Proposed Methods

The proposed method presents three new definitions: Hardy-Ramanujan Tamil Grid (HRT-Grid), Astrological Planet Tamil Grid (APT-Grid) and Galois Field Tamil Grid (GFT-Grid), and develops mathematical connections between them. The HRT-Grid is a grid that is formed by the product set of the sounds of vowels and consonants of Tamil letters and that has the dimension of 13×19, where 13 and 19 are the prime factors of the Hardy-Ramanujan number 1729. Note that the Hardy-Ramanujan number 1729 proved to have significant mathematical contributions; for example, Ono has shown its connection to K3 surface (Ono and Trebat-Leder [2016]). The APT-Grid is a set with the dimension 3×3 that consists of nine digits $1, 2, \ldots, 9$ where each digit may represent an astrological planets, Ramasubramanian [2019]. Similarly, the GFT-Grid represents a grid that has the dimension 16×16, where the elements of the GFT-Grid, that are represented by 8-bit integers, are the letters of the Tamil alphabet. The proposed method also presents two mathematical modeling approaches and one simulation architecture. The first modeling component defines a one-to-one mapping between the HRT-Grid and the Tamil alphabet, as shown in Figure 1. The second modeling component defines a one-to-many mapping between the Tamil alphabet (247 letters) and the 16×16 $(= 256)$ elements of the GFT-Grid such that the elements of this square grid satisfies the properties of a

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	Sun 1	ஃ	அ	ஆ	இ	ஈ	உ	ஊ	எ	ஏ	ஐ	ஒ	ஓ	ஔ	க்	க
1	கா	கி	கீ	கு	கூ	கெ	கே	கை	கொ	கோ	கௌ	ங்	ங	ஙா	ஙி	Mercury 2
2	ஙீ	ஙு	ஙூ	ஙெ	ஙே	ஙை	ஙொ	ஙோ	ஙௌ	ச்	ச	சா	சி	சீ	சு	சூ
3	செ	சே	சை	சொ	சோ	சௌ	ஞ்	ஞ	ஞா	ஞி	ஞீ	ஞு	ஞூ	ஞெ	Venus 3	ஞே
4	ஞை	ஞொ	ஞோ	ஞௌ	ட்	ட	டா	டி	டீ	டு	டூ	டெ	டே	டை	டொ	டோ
5	டௌ	ண்	ண	ணா	ணி	ணீ	ணு	ணூ	ணெ	ணே	ணை	ணொ	ணோ	Moon 4	ணௌ	த்
6	த	தா	தி	தீ	து	தூ	தெ	தே	தை	தொ	தோ	தௌ	ந்	ந	நா	நி
7	நீ	நு	நூ	நெ	நே	நை	நொ	நோ	நௌ	ப்	ப	பா	Mars 5	பி	பீ	பு
8	பூ	பெ	பே	பை	பொ	போ	பௌ	ம்	ம	மா	மி	மீ	மு	மூ	மெ	மே
9	மை	மொ	மோ	மௌ	ய்	ய	யா	யி	யீ	யு	யூ	Jupitar 6	யெ	யே	யை	யொ
A	யோ	யௌ	ர்	ர	ரா	ரி	ரீ	ரு	ரூ	ரெ	ரே	ரை	ரொ	ரோ	ரௌ	ல்
B	ல	லா	லி	லீ	லு	லூ	லெ	லே	லை	லொ	Saturn 7	லோ	லௌ	வ்	வ	வா
C	வி	வீ	வு	வூ	வெ	வே	வை	வொ	வோ	வௌ	ழ்	ழ	ழா	ழி	ழீ	ழு
D	ழூ	ழெ	ழே	ழை	ழொ	ழோ	ழௌ	ள்	ள	Rahu 8	ளா	ளி	ளீ	ளு	ளூ	ளெ
E	ளே	ளை	ளொ	ளோ	ளௌ	ற்	ற	றா	றி	றீ	று	றூ	றெ	றே	றை	றொ
F	றோ	றௌ	ன்	ன	னா	னி	னீ	னு	Ketu 9	னூ	னெ	னே	னை	னொ	னோ	னௌ

Figure 5: It illustrates the proposed GFT-Grid that comprises the HRT-Grid and APT-Grid. Note that the hexadecimal numbers (rows and columns) are assigned purely based on the 8-bits numbering, rather than the actual Unicode numbering assigned for Tamil letters or the actual characterization incorporated in the Unicode. However, the concept can be easily translated to the Unicode system with modifications. The placement of the APT-Grid on the GFT-Grid is parameterized.

Galois field of GF(2^8) that plays a major role in the AES algorithm Stallings [2006]. The APT-Grid is used to parameterize the elements of the GFT-Grid and systematically complete the 9 holes in the GFT-Grid.

The proposed simulation architecture presents a computational framework that helps us model and analyze the HRT-Grid and GFT-Grid properties, and evaluate the cryptographic strength of the Tamil language using the AES algorithm and the runs test scores as the measure of randomness. In other words, the transformation of the contextual properties of the Tamil alphabet to a mathematical measurement domain is modeled by leveraging the concepts of Galois field and the AES algorithm along with the quantified cryptographic strength of the transformations and the encoding Tamil texts. It also uses a Python programming environment with the libraries that include the galois (which allows to perform Galois field's mathematical operations), itertools module and its groupby method, and collections module and its counter tool.

In addition, the method uses the English plaintext "I am going to school" and its Tamil translated versions to study the effects of the proposed HRT-Grid and GFT-Grid in the application of AES and show the cryptographic strength of the Tamil text. Five versions of the Tamil translation of the English plaintext "I am going to school" are included in the proposed analysis. Figures 2, 3 and 4 explain the assignment of

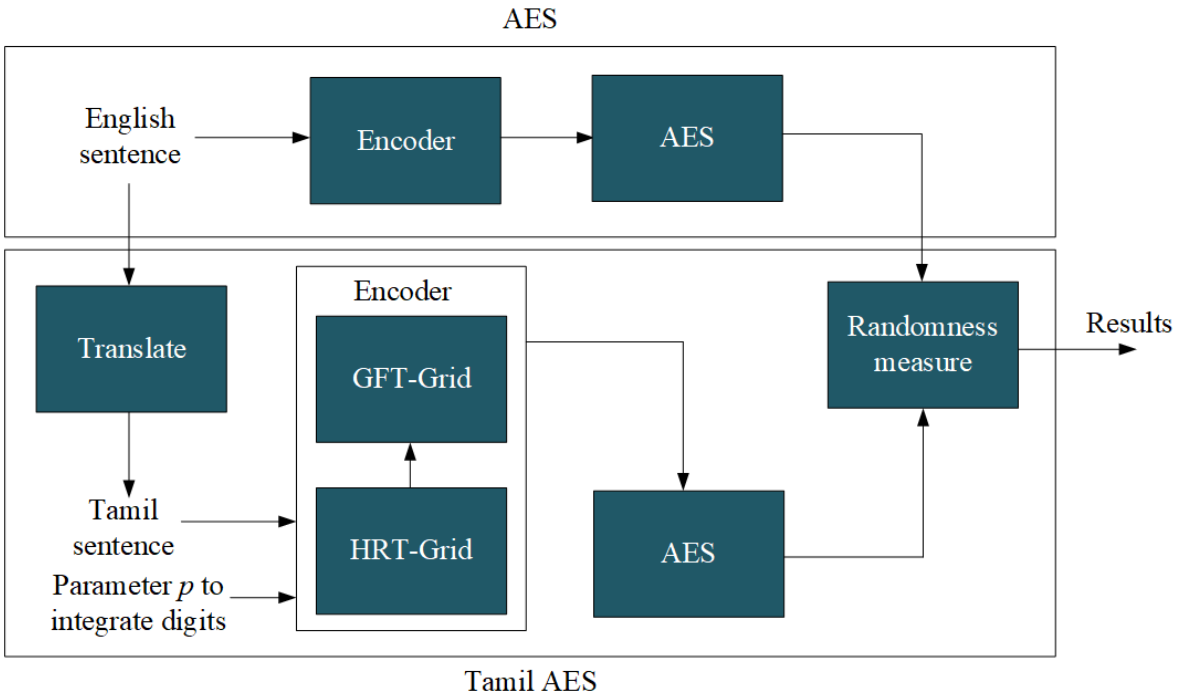

Figure 6: It illustrates the proposed computational framework. The placement of the elements of the APT-Grid on the GFT-Grid in Figure 5 is parameterized by p in this framework such that it can be used for message authentication and integrity in addition to the establishment of the confidentiality protection.

weights to the letters of English alphabet, their representation in hexadecimal format, and the pre-round transformation of a block of texts that carry the message of "I am going to school", respectively. The encoding of its Tamil counterpart is discussed in the following subsections 4.1, 4.2, and 4.3 in detail.

4.1 Tamil Alphabet to HRT-Grid

The first task is to mathematically model the relationship between the Tamil language and the prime factors of the Hardy-Ramanujan number 1729. Tamil language has 247 characters in its alphabet, as shown in Figure 1. It has 12 vowels (first row; columns 2 to 13), 18 consonants (first column; rows 2 to 19), and a special character. The remaining 216 characters are formed by combining the sounds (phonics) of the 12 vowels and 18 consonants. Hence, to mathematically model the Tamil language, the vowels set V and the consonants set C are respectively defined as follows:

$$V = \{\phi, \text{அ}, \text{ஆ}, \ldots, \text{ஔ}\}_{13}$$

$$C = \{\phi, \text{க்}, \text{ங்}, \ldots, \text{ன்}\}_{19}$$

Note that both the vowels and consonants sets include a null element ϕ. Figure 1 represents the product set $V \times C$ that has all the ordered pairs of the elements of the vowel set V and consonants set C of Tamil alphabet. Hence, it has 247 tuples $< v, c >$, where $v \in V$ and $c \in C$. We can now reveal that the first pair $< \phi, \phi >$ of the product set forms the special Tamil character ஃ (akh) which is called the Ayutha Ezhuthu (tool letter) in Tamil–It could have been used as a tool to mold other letters. Hence, a one-to-one sound function $f : V \times C \rightarrow X$ can be defined for the Tamil letters as follows:

$$f(v, c) = x, \tag{1}$$

where $v \in V$, $c \in C$, and $x \in X$. The set X represents the Tamil alphabet that consists of the 247 characters, as shown in Figure 1. The product set $V \times C$ has the dimension of 13×19 that are the two of the three

I	A	M	G	O	I	N	G	T	O	S	C	H	O	O	L
நா	ன்	ப	ள்	ளி	க்	கூ	ட	ம்	செ	ல்	லு	கி	ன்	றே	ன்
நா	ன்	ப	ள்	ளி	க்	கூ	ட	ம்	செ	ல்	கி	ன்	றே	ன்	8
நா	ன்	ப	ள்	ளி	க்	கூ	ட	ம்	செ	ல்	லு	கி	றே	ன்	8
நா	ன்	ப	ள்	ளி	க்	கு	செ	ல்	கி	ன்	றே	ன்	7	8	9
நா	ன்	ப	ள்	ளி	க்	கு	செ	ல்	லு	கி	றே	ன்	7	9	8

Figure 7: It shows the English and Tamil text examples used in the experiments.

prime factors of the Hardy-Ramanujan number 1729; hence, the domain $V \times C$ of the Tamil sound function f is called the Hardy Ramanujan Tamil Grid (HRT-Grid) in this paper. Let's now algebraically expand the HRT-Grid into two square grids as follows:

$$13 \times 19 = (16 - 3) \times (16 + 3) = 16^2 - 3^2, \tag{2}$$

$$13 \times 19 + 3^2 = 16^2. \tag{3}$$

It describes that the number of letters in the Tamil alphabet (or the number of elements in the HRT-Grid) plus the number of digits in the set $Y = \{1, 2, \ldots, 9\}$ is equal to the maximum number of 8-bit integers that we could have. The square grid of 9 digits and the square grid of 256 integers are respectively called the Astrological Planet Tamil Grid (APT-Grid) and Galois Field Tamil Grid (GFT-Grid). The modeling of the GFT-Grid and the revelation of the cryptographic connection between the Tamil language and the Galois field are presented in the subsection below.

4.2 HRT-Grid to Galois Field

Now suppose we have a set of 4-bit integers that are represented by the hexadecimal numbers $H = \{0, 1, \ldots, 9, A, B, C, D, E, F\}_{16}$, then we can define a product set $Q = H \times H = \{< 0, 0 >, < 0, 1 >, \ldots, < F, F >\}_{256}$ that consists of all the 256 ordered pairs of the 16 elements of the set H. Hence, it defines an 8-bit integer by pairing the 4-bit integers. We can then define a one-to-one mapping $g : X \cup Y \rightarrow H \times H$.

$$g(y) = h_1 h_2, \tag{4}$$

where $y \in X \cup Y$, and h_1 and h_2 are hex number that form 8-bit integers $h_1 h_2$. The set Y represents the set of 9 digits $Y = \{1, 2, \ldots, 9\}$. The one-to-one mapping between y and $h_1 h_2$ may be defined by many ways depending on whether $y \in X$ or $y \in Y$. One way to generate this mapping is presented in Figure 5 that uses the following one-to-one mapping for $y \in Y$:

```
g(1)=00; g(2)=1F; g(3)=3E; g(4)=5D; g(5)=7C; g(6)=9B; g(7)=BA; g(8)=D9; g(9)=F8.
```  (5)

In mathematics, a field is an infinite domain of elements (or observations) on which two pairs of operations–addition/subtraction and multiplication/division–are performed with an exception of not allowing division by zero. In practical applications, the data domains are generally finite; hence, a finite field must be defined. It is well-known that a field to be finite, the number of elements in the field must satisfy a^p, where the base a and the exponent p are a prime number and a positive integer, respectively. In this case, the finite field is called the Galois Field and denoted by GF(a^p). In cryptography, the AES technique uses the finite field GF(2^8) since 2 is a prime number and the elements are considered an 8-bit numbers; hence, the number of elements in the Galois field GF(2^8) is 256 (or 16^2). In the proposed model, the one-to-one function g maps every element y of the HRT-Grid and the APT-Grid to an 8-bit integer $h_1 h_2$ that lies in GFT-Grid. Hence, the GFT-Grid has 256 characters (or hexadecimal numbers); thus, it forms a Galois field GF(2^8) with a multiplication operator $\otimes$ and addition/subtraction operator $\oplus$ over the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$. The

நா	ன்	ப	ள்	ளி	க்	கூ	ட	ம்	செ	ல்	லு	கி	ன்	றே	ன்
6E	F2	7A	D7	DB	0E	14	45	87	30	AF	B4	11	F2	ED	F2
நா	ன்	ப	ள்	ளி	க்	கூ	ட	ம்	செ	ல்	கி	ன்	றே	ன்	8
6E	F2	7A	D7	DB	0E	14	45	87	30	AF	11	F2	ED	F2	D9
நா	ன்	ப	ள்	ளி	க்	கூ	ட	ம்	செ	ல்	லு	கி	றே	ன்	8
6E	F2	7A	D7	DB	0E	14	45	87	30	AF	B4	11	ED	F2	D9
நா	ன்	ப	ள்	ளி	க்	கு	செ	ல்	கி	ன்	றே	ன்	7	8	9
6E	F2	7A	D7	DB	0E	13	30	AF	11	F2	ED	F2	BA	D9	F8
நா	ன்	ப	ள்	ளி	க்	கு	செ	ல்	லு	கி	றே	ன்	7	9	8
6E	F2	7A	D7	DB	0E	13	30	AF	B4	11	ED	F2	BA	F8	D9

Figure 8: It illustrates the five versions of the Tamil texts that provide the same message as "I am going to School". It also provides the encoded versions of these five texts using the proposed GFT-Grid that satisfies the Galois Field GF(2^8). Note that the digits from the APT-Grid are selected empirically and added as padding to the encoded Tamil texts; however, these padded sequences can be used to authenticate the message or confirm the integrity of the message in addition to their contribution to confidentiality protection.

multiplication and addition tables of this Galois Field have 256×256 dimension. For example, a Tamil letter in the multiplication table is a multiplication of another two Tamil letters in the table over the irreducible polynomial $x^8 + x^4 + x^3 + x + 1$:

$$\text{நா} \otimes \text{ம்} = \texttt{6E} \otimes 87 = \texttt{0E} = \text{க்}$$

$$\text{நா} \oplus \text{ம்} = \texttt{6E} \oplus 87 = \texttt{E9} = \text{றீ}$$

Hence, we can generate a substitution box (called S-Box as in AES algorithm) for Tamil language encryption as an encoder, to be integrated in the AES's encryption flow. The GFT-Grid in Figure 5 is the S-Box that is used in this paper; however, as stated earlier, there are many possible assignments that will lead to more secure framework if the adapted assignment is not disclosed. These assignments may also be used for authentication and integrity of the messages. As per the concept of Galois field the multiplication of the two Tamil letters நா and ம் is mapped back to the Tamil letter க் in the Galois field over the irreducible polynomial $x^8+x^4+x^3+x+1$. Similar the addition of them also mapped back to the letter றீ in the Galois field. There are 256 pairs of Tamil letters mapped to the same letter; hence, given a letter in the ciphertext it is difficult to perform a brute force attack to recover its corresponding pair in the plaintext.

In the AES encryption, these Galois field operations are performed multiple times at multiple steps and multiple rounds in the encryption flow (forming a nested operation);hence, given letter in a cipehrtext, it is very difficult to traverse back in the encryption flow and recover the plaintext of the ciphertext. For example,

$$(\text{ந} \otimes \text{ன்}) \otimes \text{றி} = (\texttt{6D} \otimes \texttt{F2}) \otimes \texttt{E8} = \texttt{A3} \otimes \texttt{E8} = 44 = \text{ட்}$$

$$(\text{ந} \oplus \text{ன்}) \oplus \text{றி} = (\texttt{6D} \oplus \texttt{F2}) \oplus \texttt{E8} = \texttt{9F} \oplus \texttt{E8} = \texttt{BA} = 7 \text{ (Saturn)}$$

As per the concept (associative law) of Galois field the multiplication of the three Tamil letters ந, ன், and றி is mapped back to the Tamil letter ட் in the Galois field over the irreducible polynomial $x^8+x^4+x^3+x+1$. Similar the addition of them also mapped back to the digit 7 (that represents the astrological planet Saturn)

08	0E	13	07		6E	DB	87	11
00	08	0E	0E		F2	0E	30	F2
0C	0D	12	0E		7A	14	AF	ED
06	06	02	0B		D7	45	B4	F2

Figure 9: It illustrates the transformation of the ciphertexts to states (in hexadecimal).

in the Galois field. This iterative application of the Galois field operators, combined with many-to-one mapping, it is very difficult for the adversaries to perform a brute force attack on the Tamil ciphers than the English ciphers to recover their plaintexts. Hence, the addition of modules to generate the confusion and diffusion properties with the propagation effect makes it more difficult for the adversaries to attack.

4.3 Computational Framework

The computational framework of the proposed method, presented in Figure 6, provides a simulation architecture that helps systematically compile the presented mathematical approaches to model and simulate the cryptographic modules for encrypting English and Tamil texts using the AES algorithm. This computational framework shows two separate streams of simulation modules, the English-based AES simulation (the top stream) and the Tamil-based AES simulation (the bottom stream) to develop encryption flows. It also shows a common module "Randomness Measure" that helps calculate the runs test scores and compare the ciphers of English and Tamil texts. The English-based AES simulation takes an English text and passes through an Encoder module and the AES module. The first module provides a simple encoder that allows the English text to be transformed to a hexadecimal state. It can be treated as the pre-round transformation of the AES algorithm. As stated earlier, the example of English text that is considered in this paper is "I am going to School" and it is presented in Figure 7. This figure also presents five versions of the Tamil language texts that deliver the same message. The state output of the first module is delivered to the AES module which then applies SubBytes, ShiftRows, and MixColumns transformations. The ciphertext outputs of these transformations are then given to the "Randomness Measure" module to calculate their runs test scores.

In contrast, the Tamil-based AES simulation takes a Tamil text and passes through five modules: Translate, HRT-Grid, GFT-Grid, and AES. The Translate module takes an English text and translate it to a Tamil text that delivers the same message as the English text. Alternatively, a Tamil text can be directly fed into the next module (i.e., the HRT-Grid module) without using the Translate module. Figure 8 shows some examples of the Tamil texts that provide the same meaning as "I am going to School". The Tamil text is then delivered to the HRT-Grid module which allows a one-to-one mapping to an index of the HRT-Grid. The output of this module is then delivered to the GFT-Grid module that provides the GFT-Grid and an S-box to encode the Tamil text by utilizing the Tamil alphabet and the APT-Grid. This encoded text is then delivered to the AES module, as performed earlier, which then applies SubBytes, ShiftRows, and MixColumns transformations. The intermediate ciphertext outputs are then given to the "Randomness Measure" module to calculate their runs test scores. These two simulations allow us to compare the cryptographic strengths of various texts (English and Tamil) at every step of the encryption flow of the AES algorithm. The state representations of the English and Tamil encoders are presented in Figure 9. We can clearly see a significant redundancy (non-randomness property) in the encoded ciphertext of the English text than that of the Tamil text.

5 Experimental Results

In our experimental analysis, using the proposed computational framework, we studied the effects of English and Tamil plaintexts at the intermediate and the final steps of the AES algorithm without using the round keys. The flow of analysis that is presented in Figure 6 is adapted to systematically perform the experiments. We know that the use of the round keys provide significant protection against the attacks that attempt to recover the plaintext from a given ciphertext. In our experiments, we are only interested in the effect of plaintexts (English and Tamil) in developing randomness in the intermediate and the final ciphertexts, and the confidentiality protection of the AES modules. The English and Tamil plaintexts in Figure 7 are used.

Table 1: Comparison of cryptographic strength after pre-round transformation

Plaintext	Mean μ	Std σ	Runs Test Score ρ	Random (Y/N)
English Sentence	51.859	4.468	3.102	N
Tamil Sentence 1	64.609	5.600	0.287	Y
Tamil Sentence 2	64.438	5.585	0.078	Y
Tamil Sentence 3	64.609	5.600	0.109	Y
Tamil Sentence 4	63.438	5.496	0.079	Y
Tamil Sentence 5	63.734	5.222	0.048	Y

Table 2: Comparison of cryptographic strength after SubBytes transformation

Plaintext	Mean μ	Std σ	Runs Test Score ρ	Random (Y/N)
English Sentence	62.359	5.400	1.045	Y
Tamil Sentence 1	65.000	5.635	0.000	Y
Tamil Sentence 2	65.000	5.635	0.355	Y
Tamil Sentence 3	64.984	5.633	0.003	Y
Tamil Sentence 4	63.438	5.496	0.079	Y
Tamil Sentence 5	64.984	5.633	0.003	Y

5.1 Runs Test Validation

The cryptographic strength of the proposed Tamil-based encryption approach is validated using the runs test and compared with that of the English-based encryption approach. The runs test measure has been widely used in statistical process analysis to determine the randomness of bit sequences Alwan [2000]. It defines a testing hypothesis H_0: A sequence is random with an alternative hypothesis H_1: the sequence is not random, and develops the runs test measure ρ as follows (Alwan [2000] and NIST [Accessed Online 12/16/2021]):

$$\mu = \frac{2n_1n_2}{n_1+n_2} + 1 \qquad \sigma = \sqrt{\frac{2n_1n_2(2n_1n_2 - n_1 - n_2))}{(n_1+n_2)^2(n_1+n_2-1)}}, \tag{6}$$

$$\rho = \frac{|r-\mu|}{\sigma}, \tag{7}$$

where μ is the expected number of runs in the bit sequence, σ is the standard deviation of the number of runs, n_1 and n_2 are the number of zeros (0) and ones (1) in the bit sequence, and the variable r represents the number of runs found in the sequence. As stated in NIST [Accessed Online 12/16/2021]), we can reject the null hypothesis H_0: A sequence is random with 95% confidence if $\rho > 1.96$, provided n_1 and n_2 are greater than 10. We can use the deviation between ρ and 1.96 as a measure of cryptographic strength to compare the performance of English and Tamil encryption. The non-random bit sequences in a ciphertext can expose the deterministic patterns which could be used by adversaries to recover its plaintext by using a brute force attack, as an example. Two coding (Python) examples that use the runs test to validate the randomness of English and Tamil texts are given in Listings 1 and 2 at the end of this paper. The outputs of these Python modules are presented at the bottom of each listing. We can see significant randomness, as diffusion, in the placement of 0s and 1s in the Tamil encoded text than that of English text. This provides only a visual interpretation; however, the randomness will be measured using the runs test score in the experiments.

5.2 Experiment 1

The English sentence "I am going to School" in Figure 7 is used in this experiment along with the flow of analysis at the top stream of Figure 6. As traditionally performed in the AES algorithm, this sentence is encoded to hexadecimal form using the location profiles of the letters in the English alphabet. Figure 2 shows the location information and Figure 3 shows their hexadecimal representations. The values in Figure 3 are used for encoding the English text for the pre-round transformation of AES. The encoded text, in its bit sequence form, is analyzed to measure the level of randomness using the runs test score ρ. The results

Table 3: Comparison of cryptographic strength after ShiftRows transformation

Plaintext	Mean μ	Std σ	Runs Test Score ρ	Random (Y/N)
English Sentence	62.359	5.400	1.045	Y
Tamil Sentence 1	65.000	5.635	0.355	Y
Tamil Sentence 2	65.000	5.635	0.887	Y
Tamil Sentence 3	64.984	5.633	1.068	Y
Tamil Sentence 4	64.938	5.496	0.366	Y
Tamil Sentence 5	64.984	5.633	0.358	Y

Table 4: Comparison of cryptographic strength after MixColumns transformation

Plaintext	Mean μ	Std σ	Runs Test Score ρ	Random (Y/N)
English Sentence	64.750	5.612	1.559	Y
Tamil Sentence 1	64.000	5.546	1.082	Y
Tamil Sentence 2	64.938	5.629	0.189	Y
Tamil Sentence 3	64.984	5.633	0.003	Y
Tamil Sentence 4	64.000	5.546	0.180	Y
Tamil Sentence 5	64.984	5.633	0.713	Y

are presented in the first row of Table 1. The runs test score of 3.102 (i.e. above 1.96) indicates the resulting encoded text at the pre-round step is not random for the English plaintext "I am going to School".

The encoded text is then used to generate the ciphertexts (without using the round keys) by applying the SubBytes, ShiftRows, and MixColumns transformations in the encryption flow of the AES algorithm. The randomness of the ciphertexts of these transformations are then measured by calculating the runs test scores as previously performed. The results are respectively presented in the first row of the Tables 2 to 4. As we can observe, the SubBytes transformation achieves randomness with the runs test score of 1.045, the ShiftRows transformation achieves randomness with the runs test score of 1.045 also, and the MixColumns transformation achieves randomness with the runs test score of 1.559. Hence, the addition of these transformations provides an increased cryptographic strength to the English-based AES encryption.

5.3 Experiment 2

The translated Tamil sentences of the English sentence "I am going to School" in Figure 7 are used in this second experiment along with the flow of analysis that is presented at the bottom stream of Figure 6. The encoded versions of the Tamil texts using the Galois Field of Tamil language–by applying the GFT-Grid in Figure 5–are shown in Figure 8. All the encoded Tamil texts, in their bit sequence forms, are then analyzed and the level of randomness are measured by using the runs test score ρ. The results are presented in the second row through to sixth row of Table 1. All the five Tamil texts pass the randomness tests for the pre-round transformation of the AES algorithm with the lower scores that range from 0.048 to 0.287, and the average score of 0.1202 which is significantly lower than the runs test score 3.102 of the English text. Therefore, by comparing all the runs test scores in Table 1, we can say that the Tamil texts carry cryptographic properties that are useful for AES encryption; hence, help achieve the stronger cryptographic strength at the pre-round transformation stage of the AES encryption algorithm.

The encoded Tamil texts are then used, as previously, to generate the ciphertexts (once again without using the round keys) by applying the SubBytes, ShiftRows, and MixColumns transformations in the encryption flow of the AES algorithm. The randomness of the ciphertexts of these transformations for each Tamil text are then measured by calculating the runs test scores as performed for the English text. The results are respectively presented in the corresponding rows of Tables 2 to 4. The results of SubBytes in Table 2 shows that the Tamil texts achieve significant randomness with the scores that range from 0 to 0.355 with the average score of 0.088. We can also observe that the Tamil sentence 1 has achieved a perfect randomness based on the runs test score, while the Tamil sentences 3, 4, and 5 achieve very high randomness.

Table 3 shows the results after the application of ShiftRows transformation of the AES encryption flow to the Tamil texts used. The results show that the Tamil texts yield the randomness with the runs test scores that

Table 5: Summary of cryptographic strengths

Plaintext	Pre-round	SubBytes	ShiftRows	MixColumn
English	V. Weak	Weak	Weak	Weak
Tamil 1	V. Strong	V. Strong	V. Strong	Weak
Tamil 2	V. Strong	V. Strong	Strong	V. Strong
Tamil 3	V. Strong	V. Strong	Weak	V. Strong
Tamil 4	V. Strong	V. Strong	V. Strong	V. Strong
Tamil 5	V. Strong	V. Strong	V. Strong	Strong

range from 0.355 to 1.068 with the average score of 0.6068. Similarly, Table 4 presents the results after the application of MixColumns transformation of the AES encryption flow to the Tamil texts used. The results show that the Tamil texts yield the randomness with the runs test scores that range from 0.003 to 1.082 with the average score of 0.4334. Tables 1 to 4 also provide mean and the standard deviations that are used to calculate the runs test scores. The similar standard deviation values indicate that the runs test scores calculated for the English text and the corresponding translated Tamil versions are on similar agreement; hence, the meaning of their scores interpretable in the same way. Hence, by combining the results in these Tables, we can conclude that the Tamil text can provide a very high cryptographic strength to the AES encryption flow through its mathematical properties associated with the Galois Field GF(2^8).

6 Summarized Discussion

The summary of the results are presented in Table 5 based on the following linear scale of the runs test scores: The range $\rho > 1.96$ means the cryptographic strength is very weak (i.e., non-random); the interval $1 < \rho \leq 1.96$ means the cryptographic strength is weak; the interval $0.5 < \rho \leq 1$ means the cryptographic strength is strong; and the interval $0 \leq \rho \leq 0.5$ means the cryptographic strength is very strong. The results in every row of Table 5 clearly indicate that the Tamil language provides a stronger cryptographic strengths than English language. Similarly the results in the second and the third columns clearly indicate that the Tamil language provides very strong cryptographic strengths to the pre-round and SubBytes steps of AES, while creating a possible weaker situations in ShiftRows and MixColumns (fourth and fifth columns of Table 5). However, by combining the strengths in their associated cryptographic modules, Tamil language can overall provide very strong cryptographic strengths to the encryption flow of the AES algorithm.

It also suggests that the encryption of the Tamil text with only the pre-round transformation is sufficient to develop a very strong cryptogrphic strength in the ciphertext. It can also give much stronger cryptographic strength when the round keys are added. When the other rounds with SubBytes, ShiftRows, and MixColumns, are added to generate confusion and diffusion, then the Tamil language texts are expected to give a remarkable cryptographic strengths to the AES algorithm, based on the results presented in Table 5.

7 Conclusion

This paper revealed the mathematical relationship between the Tamil language and the Hardy-Ramanujan number. It also showed that the letters of the Tamil alphabet, combined with the digits 1 to 9, are the members of a Galois field of GF(2^8) with an irreducible polynomial of degree 8. This paper also presented an encoder (a pre-round encryption module) that transforms the Tamil texts to hexadecimal states. This encoder can replace the pre-round transformation of AES when the goal is the encryption of Tamil texts. The experimental analysis showed that this encoder can induce increased randomness in the intermediate ciphers that are the outputs of the SubBytes, ShiftRows, and MixColumns transformations of the AES encryption flow. Therefore, based on this empirical study, the Tamil language could offer cryptographic strengths to the AES-based confidentiality protection and enhance the cybersecurity requirements. It can also support the message authentication and integrity requirements. Also, note that the goal was to reveal the hidden HRT-Grid and GFT-Grid properties of the Tamil language and show its cryptographic strengths; hence, the Unicode values are not used. However, one could linearly shift the values of the grids relatively and establish a significant correspondence. Therefore, our future research in this topic will focus on the modification of the model to meet the Unicode requirements. The future research will also focus on the adaptive placement of the elements of the APT-Grid on the GFT-Grid and its contributions to the confidentiality protection.

References

Yevgeniy Dodis, Martijn Stam, John Steinberger, and Tianren Liu. Indifferentiability of confusion-diffusion networks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 679–704. Springer, 2016.

Asko Parpola. *The roots of Hinduism: the early Aryans and the Indus civilization*. Oxford University Press, USA, 2015.

Harold F Schiffman. *A reference grammar of spoken Tamil*. Cambridge University Press, 1999.

Nils Gura, Arun Patel, Arvinderpal Wander, Hans Eberle, and Sheueling Chang Shantz. Comparing elliptic curve cryptography and rsa on 8-bit cpus. In *International workshop on cryptographic hardware and embedded systems*, pages 119–132. Springer, 2004.

Shan Suthaharan. Calculating a cryptographic primitive: Suitable for wireless sensor networks. In *2008 Australasian Telecommunication Networks and Applications Conference*, pages 51–56. IEEE, 2008.

Francisco Rodriguez-Henriquez, Arturo Diaz Perez, Nazar Abbas Saqib, and Cetin Kaya Koc. A brief introduction to modern cryptography. *Cryptographic Algorithms on Reconfigurable Hardware*, pages 7–33, 2007.

Jaishankar Bharatharaj, Mohsen Alyami, Marcus A Henning, Hussain Alyami, and Christian U Krägeloh. Tamil version of the fear of covid-19 scale. *International Journal of Mental Health and Addiction*, pages 1–12, 2021.

M Rajendiran, B Syed Ibrahim, R Pratheesh, and C Nelson Kennnedy Babu. Multilanguage based sms encryption techniques. In *Proceedings of International Conference on Advances in Computing*, pages 455–460. Springer, 2013.

R Geetha, T Padmavathy, T Thilagam, and A Lallithasree. Tamilian cryptography: An efficient hybrid symmetric key encryption algorithm. *Wireless Personal Communications*, pages 1–16, 2019.

S Lokesh, Priyan Malarvizhi Kumar, M Ramya Devi, P Parthasarathy, and C Gokulnath. An automatic tamil speech recognition system by using bidirectional recurrent neural network with self-organizing map. *Neural Computing and Applications*, 31(5):1521–1531, 2019.

Nagul Ulaganathan, J Rohith, AS Abhinav, V Vijayakumar, L Ramanathan, et al. Isolated handwritten tamil character recognition using convolutional neural networks. In *2020 3rd International Conference on Intelligent Sustainable Systems (ICISS)*, pages 383–390. IEEE, 2020.

AN Sigappi and S Palanivel. Spoken word recognition strategy for tamil language. *International Journal of Computer Science Issues (IJCSI)*, 9(1):227, 2012.

M Selvam, AM Natarajan, and R Thangarajan. Structural parsing of natural language text in tamil using phrase structure hybrid language model. *International Journal of Computer, Information and Systems Science, and Engineering*, 2:4, 2008.

Venkatesh Ashok, KC Premarajan, Ravi Philip Rajkumar, Bijaya Nanda Naik, et al. Mental health status of flood-affected adults in rural tamil nadu: A cross-sectional study. *CHRISMED Journal of Health and Research*, 6(2):97, 2019.

William Stallings. Cryptography and network security principles and practices 4th edition, 2006.

Behrouz A Forouzan. *Cryptography and Network Security*. McGraw-Hill, 2008.

Simon Heron. Advanced encryption standard (aes). *Network Security*, 2009(12):8–12, 2009.

Yehya A Nasser, Mohammad A Bazzoun, and Samih Abdul-Nabi. Aes algorithm implementation for a simple low cost portable 8-bit microcontroller. In *2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC)*, pages 203–207. IEEE, 2016.

Joan Daemen and Vincent Rijmen. Reijndael: The advanced encryption standard. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 26(3):137–139, 2001.

William Stallings. The advanced encryption standard. *Cryptologia*, 26(3):165–188, 2002.

Ken Ono and Sarah Trebat-Leder. The 1729 k 3 surface. *Research in Number Theory*, 2(1):1–6, 2016.

K Ramasubramanian. Yantras or mystic diagrams: A wide area for study in ancient and medieval indian mathematics. In *Gaṇitānanda*, pages 227–260. Springer, 2019.

Layth C Alwan. *Statistical process analysis*. McGraw-Hill/Irwin, 2000.

NIST. Runs Test for Detecting Non-randomness. `https://www.itl.nist.gov/div898/handbook/eda/section3/eda35d.htm`, Accessed Online 12/16/2021.

Listing 1: Runs test that validates the randomness of English.

```
import math
from itertools import groupby
from collections import Counter

stateE = ["08", "00", "0C", "06", "0E", "08", "0D", "06", "13", "0E", "12", "02", "07"
    , "0E", "0E", "0B"]

binStringE = bin(int(stateE[0], 16))[2:].zfill(8)
for ii in range(1, 16):
    tmpE = bin(int(stateE[ii], 16))[2:].zfill(8)
    binStringE = binStringE+tmpE

print(binStringE)
m1 = binStringE.count('0')
m2 = binStringE.count('1')
print(m1,m2)

ER2 = ((2*m1*m2)/(m1+m2))+1
SD2 = math.sqrt((2*m1*m2*(2*m1*m2-m1-m2))/(((m1+m2)**2)*(m1+m2-1)))

cE = Counter(k for k, g in groupby(binStringE))
runsE = cE['0']+cE['1']
Z2 = abs(runsE-ER2)/SD2
print(Z2)
print("H0: The sequence is random")
print("Since Z2 = 3.1016........ > 1.96; with 95% confidence we reject H0")
print("The English state is NOT random")
```

0000100000000000000011000000011000001110000010000000110100000110000100110000111000010010000000100000011100001110000011100000111

Listing 2: Runs test that validates the randomness of Tamil.

```
import math
from itertools import groupby
from collections import Counter

stateT = ["6E", "F2", "7A", "D7", "DB", "0E", "14", "45", "87", "30", "AF", "B4", "11"
    , "F2", "ED", "F2"]

binStringT = bin(int(stateT[0], 16))[2:].zfill(8)
for ii in range(1, 16):
    tmpT = bin(int(stateT[ii], 16))[2:].zfill(8)
    binStringT = binStringT+tmpT

print(binStringT)
n1 = binStringT.count('0')
n2 = binStringT.count('1')
print(n1,n2)

ER1 = ((2*n1*n2)/(n1+n2))+1
SD1 = math.sqrt((2*n1*n2*(2*n1*n2-n1-n2))/(((n1+n2)**2)*(n1+n2-1)))

cT = Counter(k for k, g in groupby(binStringT))
runsT = cT['0']+cT['1']
Z1 = abs(runsT-ER1)/SD1
print(Z1)
print("H0: The sequence is random")
print("Since Z1 = 0.287.... < 1.96; with 95% confidence we accept H0")
print("The Tamil state is random")
```

0110111011110010011110101101011111011011000011100001010001000101100001110011000010101111101101000010001111100101110110111110010