EasyChair Preprint

№ 1956

Enhancing the Security of Twitter Data Using SRB20 Algorithm

Bagath Basha Chan and S Rajaprakash

November 14, 2019

Enhancing The Security of Twitter Data Using SRB20 Algorithm

C Bagath Basha[1] and S Rajaprakash[2]
[1] Research Scholar, Department Of CSE
[1] Vinayaka Mission's Research Foundation, Salem, Tamil Nadu, India.
chan.bagath@gmail.com
[2] Associate Professor, Department Of CSE,
[2] Aarupadai Veedu Institute of Technology, Vinayaka Mission's Research Foundation, Chennai, Tamil Nadu, India.
srajaprakash_04@yahoo.com

Abstract

Today's world witness the buildup of information through social media like Twitter and Facebook that are tremendously growing day by day. Exploitation these social media users tweet on several topics from anywhere within the world through the web. Tweet is employed to investigate positive and negative tweets to create polarity scores and additionally in predicting the long run. These polarity scores can be extracted from Twitter although polarity score knowledge are placed below security. Hacking the info simply and dynamic the score results cause immeasurable problems like affecting the economic status worldwide, company product, and therefore the name of corporations. To tackle these problems Daniel Bernstein proposed a family of Salsa which has quicker cryptography because of quarter round with better information security. In the present work, a novel algorithm is proposed by modifying the Salsa20/4 to enhance further the security of the polarity scores, an important requirement of the present world. The proposed algorithm has three stages. The first stage consists of column operations, second stage is secret key and the third stage is constructing the diagonal keys in N-1 steps. The proposed algorithm is tested with different file sizes to calculate the encryption time. The results show that the proposed encryption algorithm takes relatively large execution time but provides high data security when compared to that of the existing algorithms of Salsa20/4, SRB18, and SRB19.

Keywords: Salsa, encryption, decryption, security, polarity.

1 Introduction

During the last decade the globe has witnessed the increase of big social media knowledge hold on in Twitter and Facebook. They need been utilized by multiusers to tweet on any topics in social media

through the net. These tweets kind the premise to research the polarity scores. The analyzed polarity scores want security as a result of Twitter application is one in every of the illustrious sources for public data within the world. Therefore output of the analyses plays an important role in creating the choices on the choice of any merchandise knowing their quality by the general public sector. In default twitter doesn't have such a decent security of that knowledge.

In this association to investigate the information and secure them. Daniel Bernstein introduced Salsa, a family of stream ciphers, specializing in cryptography work cheap for a good scope of utilizations. Salsa20 handles the keys of 128 bits or 256 bits, however Daniel Bernstein suggested secret key of 256 bits. Salsa20 with twenty spherical projected by Daniel Bernstein (Zeng-yu Shao and Lin Ding, 2012) is quicker than Advanced Encryption Standard (AES) and provides higher security (Mishal Almazrooie et al., 2015). The Salsa has reduced spherical versions from Salsa20/20 to Salsa20/12 that's eight rounds diminished from twenty rounds, and Salsa20/20 to Salsa20/8 that's twelve spherical attenuated from twenty round (Daniel J. Bernstein, 2008). Salsa20/12 and Salsa20/8 aimed toward to scale back rounds and to extend the cryptography speed. The Salsa once more reduced the spherical versions from Salsa20/20 to Salsa20/5, that's fifteen rounds attenuated from twenty rounds, and subsequently Salsa20/20 is diminished to Salsa20/6, that's fourteen rounds attenuated from twenty rounds, in 2006 (Paul Crowley, 2006), (Simon Fischer et al., 2006). Basically, Salsa20/7 was conferred in 2007. Further the Salsa family, diminished from Salsa20/20 to Salsa20/4, called quarter round of Salsa20/4 is aiming at to decrease the cryptography time. Salsa20/4 operates solely three spherical and every round has quarter rounds. Salsa20/4 cryptography speed is quicker in comparison to alternative Salsa20 variants. This Salsa20 family concentrates solely on the speed of cryptography however not on any up the information security. To beat the drawbacks of Salsa family stream ciphers in securing the information, we tend to propose during this work a completely unique rule named as: Somasundaram Rajapraksah and Bagathbasha 20 (SRB20).

2 Related Work

Improved the related cipher attack on Salsa20 family. This family has best attack model have Salsa20/12 and Salsa20/8 (Lin Ding, 2019). Analyzed the Salsa and ChaCha reduced rounds, then they proposed novel algorithm is Probabilistic Neutral Bits and this algorithm is faster than the existing attack (Sabyasachi Dey et al., 2017).

Improved the attack on 128 key bit of Salsa7 and ChaCha6 (Kakumani K. C. Deepthi and Kunwar Singh, 2018). Increased the buffer size reduces the iterations of the loop and accordingly reduces the overall encryption time on ChaCha20 (Raj R. Parmar et al., 2017).

Improved the attack of ChaCha7 and Salsa8 with proper choice of IVs (Subhamoy Maitra, 2016). Significantly improved the time complexities of Salsa in 7 rounds and ChaCha in 6 rounds (Arka Rai Choudhuri and Subhamoy Maitra, 2016).

They designed the family of Salsa20 in 2005 which is faster than AES and this family gives more important for encryption speed (Daniel J. Bernstein, 2008). The proposed the 3 Vs of big data: **Volume** stands for the size of data; **Velocity** is produced the data from various sources; and **Variety** is a data generated from different formats: structure data, semi structure data, and unstructured data (Doug Laney, 2001). They talk about the polynomial and transformation operations to implement the security of analyzed data (Joan Daemen and Vincent Rijmen, 1999).

New proposed method is the emotion method. This method used to identify the current news and to detect the related opinion in Twitter. This method is mainly used to remove the false positive and improve the accuracy (Cuneyt Gurcan Akcora et al., 2010).

To improve the short text message from author's profile using proposed approaches. This approach is used to remove the noisy data, and the result shows the better accuracy when compared to Bag of Words (BOW) (Bharath Sriram et al., 2010).

They discussed about movie reviews and to compare the performance of machine learning methods: Navie Bayes (NB), Support Vector Machine (SVM), and Maximum entropy classification. They showed the performance of machine learning methods, and the performance of SVM tend is best when compared to other methods (Bo Pang et al., 2002).

They discussed about machine learning methods and a new novel algorithm is sentiment analysis method to calculate the accuracy and compare with them for survey purposes (Bo Pang et al., 2002).

The talk data using Salsa20 stream ciphers cryptography algorithms. This algorithm is used to analyze the processing time of both encryption and decryption time is silently fast, first packet takes a few milliseconds and next packet takes one millisecond. The results show the performance of Salsa20 is best in the data security (Diyana Afdhila et al., 2016).

Improved the correlation attacks between ChaCha8 and Salsa20/9. They are tried to analyze the possibility of reducing the complexity of existing attack, but correlation attack is not useful for analyzing the ChaCha8 (Prateek Yadav et al., 2016).

Analyzed the rotations of performance level for Salsa20/8 image encryption and to compare the performance with cryptosystem. The Salsa20/8 security is not good but encryption speed is fast (Alireza Jolfaei et al., 2012).

Analyzed the quarter round of Salsa, ChaCha, and proposed Modifeied ChaCha(MCC) algorithms. These algorithms are used to produce more dispersion metrics with higher rotations but the MCC algorithms also produce more dispersion metrics with lesser rotations. The MCC has more dispersion than Salsa and ChaCha (Rajeev Sobti and Geetha Ganesan, 2016).

Hongjun has introduced the related cipher attacks in 2002. They applied to the Salsa20 stream cipher. The reducer round version of Salsa20 is flexible round and independent rounds is key schedule rounds. The results show the related cipher attack may be applied to stream ciphers (Zeng-yu Shao and Lin Ding, 2012).

They discussed about the Salsa20/4 and proposed chaotic Salsa algorithms are used to compare the speed and dispersion level. The proposed algorithm is faster than the Salsa20/4, but the dispersion level is same (Mishal Almazrooie et al., 2015).

ChaCha8 stream cipher proposed by Daniel J. Bernstein, the design of ChaCha8 is changed from Salsa20/8, to improve the dispersion per round and show the differences between Salsa20 and ChaCha (Daniel J. Bernstein, 2008) .

The various methods are available for the analysis of data in Twitter and also on security of the data. In this related works, as discussed about both data analysis (Polarity scores) and security of data (Salsa20). The Salsa20 family stream cipher encryption time is reduced only not increase the data security. The proposed algorithm SRB20 is modifying from the SRB18.

3 Methodology And Proposed Algorithm

This work deals with the info of a selected topic or space collected from Twitter. The info are accustomed classify the tweets using Rstudio on Twitter. This tweet is employed to research positive and negative tweets to form a polarity scores. The results of the polarity scores can be extracted from Twitter. This knowledge is born-again into a matrix and also the planned algorithmic program SRB20 was applied to the present matrix of order N by N. SRB20 algorithmic program has 3 stages of the secret writing method. The primary stage is exchange the column operation within the matrix. The second stage is that the secret key which matrix elements are multiplied by key. Third stage is diagonal elements in N-1 rounds that are swapping the all diagonal elements within the initial row.

The advantage of this planned algorithmic program SRB20 is takes longer for decode the info compared to existing algorithmic program SRB19, SRB18 and Salsa20.

3.1 Twitter

Twitter is one in all the planet public social media, by the user wont to tweet on any topic in social media at anytime with anyone, and anyplace within the world through net. This tweet is employed to research the polarity scores in Twitter information. Polarity scores have positive and negative tweets represented by the symbol '+', '-' severally. These tweets are collected from a selected space and topic from Twitter, then, it ought to be like sentences, which tweets can build polarity score.

The positive and negative sample words are below

- Positive words - **'good, lucky, bless, like, interest, happy'**, etc...
- Negative words - **'bad, don't, not, sad, won't, bore'**, etc…

Vocabulary	**Analyzing First Tweet**	**Analyzing Second Tweet**	**Analyzing Third Tweet**
I	1	1	1
Am	1	0	0
Not	-1	0	-2
Interested	+1	0	+1
To	1	0	1
Join	1	0	0
This	1	1	2
College	1	1	2
Because	1	0	0
No	-1	0	0
Sports	1	0	0
Like	0	+1	+1
Very	0	1	0
Much	0	1	0
So	0	0	1
Visit	0	0	1
		. .	
Class label	Negative (-)	Positive (+)	Neutral

Table 1: Investigating the tweets for extremity score.

Example - tweets,

1. First tweet: "I am not interested to join this college because no sports".
2. Second tweet: "I like this college very much".
3. Third tweet: "I not like this college so not interested to visit this college".

Table 1 gives two simple example tweets of creating the reviews, first *tweet "I am not interested to join this college because no sports"* (Class: Negative), this words has two negative words and one positive word, so it is negative tweets. Second tweet *"I like this college very much"* (Class: positive), this words has one positive words and no negative word, so it is positive tweets. Third tweet *"I not like this college so not interested to visit this college"* (Class: Neutral), this words has two negative

words and positive word, so it is neutral tweets. This polarity scores result data could be extracted in the form of matrix as shown in Table 2 [24].

Dataset	#Positive	#Negative	#Neutral
RAJINI	519	4	477
AKSHAY	360	2	638
DHONI	538	97	365
KOHLI	493	25	482

Table 2: Dataset

It is evident from Table 2 that the data prone to have security issues. The hackers can easily access the polarity scores and can make changes in their values. If so, they affect the organization name, economic status, company brand, and country name. To overcome these issues, in this work a novel secured algorithm called SRB20, is used to secure the polarity score result and data.

3.2 SRB20 Algorithm

3.2.1 SRB20 Encryption Algorithm

SRB20 encryption algorithm consists of five steps.

Step 1: Extracting the data from Twitter.

Step 2: Analyzed twitter data are stored in the matrix **K**.

Step 3: Operation are used to exchange the all the columns of the matrix.

$$C = K_{ij} \leftrightarrow \left(K_{i\,(j+(N-M))}\right) \tag{1}$$

where C is a matrix.

K represents the column operation of the matrix and i various from 1, to N. N is order of matrix, and M is odd numbers.

Step 4: Multiplying the matrix elements of **C** by a Secret key S and representing a real number except 0. We obtain

$$\text{FEM}\mathbf{B} = \text{S}\mathbf{C} \tag{2}$$

where FEM**B** is first encrypted matrix **B.**

Step 5: Operation is used to move all the diagonal elements to the first row. Each column operations has a clock wise operation.

$$\text{EM}\mathbf{A} = \text{FEM}\mathbf{B},$$

where EM**A** is encrypted matrix **A**.

$$\begin{aligned} a_{ij} &= a_{ij}, \\ a_{(k+1)j} &= a_{(k+j)j}, \\ &if\ i > N\ Then \\ i &= k - N \end{aligned} \tag{3}$$

In this operations $a_{(k+1)j}$ and $a_{(k+j)\,j}$ are matrix element, k various from 0, to N; i various from 1, to N; j various from 1, to N; and N is the order of matrix.

3.2.2 SRB20 Decryption Algorithm

SRB20 decryption algorithm consists of two steps.

Step 1: All the first row elements to form the diagonal elements of the respective columns. Each column operation has an anti-clock wise operation.

FDM**A** = EM**A**

Where FDM**A** i**s** first decrypted matrix **A**

$$\begin{aligned} &a_{ij} = a_{ij}, \\ &a_{(k+1)j} = a_{(k+n)j}, \\ &if\ i > N\ Then \\ &i = k - N, \\ &where\ n = \left(N - \left(N-1\right)\right), N, \left(N-1\right), \left(N-2\right) \& N = N \end{aligned} \tag{4}$$

In this operations $a_{(k+1)j}$ and $a_{(k+j)\,j}$ are matrix element, k various from 0, to N; i various from 1, to N; j various from 1, to N; and N is the order of matrix

Step 2: Dividing the matrix elements of **A** by the secret key S.

$$\text{SDM}\mathbf{B} = \mathbf{A} / \text{S} \tag{5}$$

where SDM**B** is second decrypted matrix **B.**

Step 3: Operation are used to bring the original column elements and produce the original matrix.

$$C = K_{ij} \leftrightarrow \left(K_{i\,(j+(N-M))}\right) \tag{6}$$

where C is a matrix.

K represents the column operation of the matrix and i various from 1, to N. N is order of matrix, and M is odd numbers.

3.3 Working Of SRB20 Algorithm

- The proposed SRB20 encryption algorithm is developed from modifying the SRB19 algorithm and it is developed from SRB18 algorithm.

$$\mathbf{K} = \begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} \end{pmatrix}$$

Where **K** is the analyzed twitter data matrix

Encryption

- Exchange the all the columns of the matrix **K** (Equation (1)).

$$\mathbf{C} = \begin{pmatrix} c_{16} & c_{15} & c_{14} & c_{13} & c_{12} & c_{11} \\ c_{26} & c_{25} & c_{24} & c_{23} & c_{22} & c_{21} \\ c_{36} & c_{35} & c_{34} & c_{33} & c_{32} & c_{31} \\ c_{46} & c_{45} & c_{44} & c_{43} & c_{42} & c_{41} \\ c_{56} & c_{55} & c_{54} & c_{53} & c_{52} & c_{51} \\ c_{66} & c_{65} & c_{64} & c_{63} & c_{62} & c_{61} \end{pmatrix}$$

- Multiplying **C** by secret key S (Equation (2)).

$$\text{FEM}\mathbf{B} = \text{W}\ \mathbf{C}$$

$$\text{First Encrypted Matrix of } \mathbf{B} = \begin{pmatrix} b_{16} & b_{15} & b_{14} & b_{13} & b_{12} & b_{11} \\ b_{26} & b_{25} & b_{24} & b_{23} & b_{22} & b_{21} \\ b_{36} & b_{35} & b_{34} & b_{33} & b_{32} & b_{31} \\ b_{46} & b_{45} & b_{44} & b_{43} & b_{42} & b_{41} \\ b_{56} & b_{55} & b_{54} & b_{53} & b_{52} & b_{51} \\ b_{66} & b_{65} & b_{64} & b_{63} & b_{62} & b_{61} \end{pmatrix}$$

The secret key S can have any value starting from $\pm 2 .... \pm n$ where n is integral number except 0.

All diagonal element are shifted to form the first row to construct the encrypt data matrix (Equation (3)). In this process the first column is not changed. The second column matrix elements are interchanged by the column operation to bring the diagonal element (2,2) to the first row (1,2) as described below: $(2,2) \Rightarrow (1,2)$, $(3,2) \Rightarrow (2,2)$, $(4,2) \Rightarrow (3,2)$, $(5,2) \Rightarrow (4,2)$, $(6,2) \Rightarrow (5,2)$, and $(1,2) \Rightarrow (6,2)$. The third column matrix elements are interchanged by the column operation to bring the diagonal element (3,3) to the first row (1,3) as described below: $(3,3) \Rightarrow (1,3)$, $(4,3) \Rightarrow (2,3)$, $(5,3) \Rightarrow (3,3)$, $(6,3) \Rightarrow (4,3)$, $(1,3) \Rightarrow (5,3)$, and $(2,3) \Rightarrow (6,3)$. The fourth column matrix elements are interchanged by the column operation to bring the diagonal element (4,4) to the first row (1,4) as described below: $(4,4) \Rightarrow (1,4)$, $(5,4) \Rightarrow (2,4)$, $(6,4) \Rightarrow (3,4)$, $(1,4) \Rightarrow (4,4)$, $(2,4) \Rightarrow (5,4)$, and $(3,4) \Rightarrow (6,4)$. The fifth column matrix elements are interchanged by the column operation to bring the diagonal element (5,5) to the first row (1,5) as described below: $(5,5) \Rightarrow (1,5)$, $(6,5) \Rightarrow (2,5)$, $(1,5) \Rightarrow (3,5)$, $(2,5) \Rightarrow (4,5)$, $(3,5) \Rightarrow (5,5)$, and $(4,5) \Rightarrow (6,5)$. The sixth column matrix elements are interchanged by the column operation to bring the diagonal element (6,6) to the first row (1,6) as described below: $(6,6) \Rightarrow (1,6)$, $(1,6) \Rightarrow (2,6)$, $(2,6) \Rightarrow (3,6)$, $(3,6) \Rightarrow (4,6)$, $(4,6) \Rightarrow (5,6)$, and $(5,6) \Rightarrow (6,6)$.

Encrypted Matrix of **A** (EM**A**) on the FEM**B** = $\begin{pmatrix} a_{16} & a_{25} & a_{34} & a_{43} & a_{52} & a_{61} \\ a_{26} & a_{35} & a_{44} & a_{53} & a_{62} & a_{11} \\ a_{36} & a_{45} & a_{54} & a_{63} & a_{12} & a_{21} \\ a_{46} & a_{55} & a_{64} & a_{13} & a_{22} & a_{31} \\ a_{56} & a_{65} & a_{14} & a_{23} & a_{32} & a_{41} \\ a_{66} & a_{15} & a_{24} & a_{33} & a_{42} & a_{51} \end{pmatrix}$

Decryption

Shifting the elements of the first row to form of diagonal elements of the respective columns to decrypt the data (Equation (4)). In this process the first column is not changed. The second column matrix elements are interchanged by the column operation to bring the first row (6,2) to the diagonal element (1,2) as described below: $(6,2) \Rightarrow (1,2)$, $(1,2) \Rightarrow (2,2)$, $(2,2) \Rightarrow (3,2)$, $(3,2) \Rightarrow (4,2)$, $(4,2) \Rightarrow (5,2)$, and $(5,2) \Rightarrow (6,2)$. The third column matrix elements are interchanged by the column operation to bring the first row (5,3) to the diagonal element (1,3) as described below: $(5,3) \Rightarrow (1,3)$, $(6,3) \Rightarrow (2,3)$, $(1,3) \Rightarrow (3,3)$, $(2,3) \Rightarrow (4,3)$, $(3,3) \Rightarrow (5,3)$, and $(4,3) \Rightarrow (6,3)$. The fourth column matrix elements are interchanged by the column operation to bring the first row (4,4) to the diagonal element (1,4) as described below: $(4,4) \Rightarrow (1,4)$, $(5,4) \Rightarrow (2,4)$, $(6,4) \Rightarrow (3,4)$, $(1,4) \Rightarrow (4,4)$, $(2,4) \Rightarrow (5,4)$, and $(3,4) \Rightarrow (6,4)$. The fifth column matrix elements are interchanged by the column operation to bring the first row (3,5) to the diagonal element (1,5) as described below: $(3,5) \Rightarrow (1,5)$, $(4,5) \Rightarrow (2,5)$, $(5,5) \Rightarrow (3,5)$, $(6,5) \Rightarrow (4,5)$, $(1,5) \Rightarrow (5,5)$, and $(2,5) \Rightarrow (6,5)$. The sixth column matrix elements are interchanged by the column operation to bring the first row (1,6) to the diagonal element (6,6) as described below: $(1,6) \Rightarrow (6,6)$, $(2,6) \Rightarrow (1,6)$, $(3,6) \Rightarrow (2,6)$, $(4,6) \Rightarrow (3,6)$, $(5,6) \Rightarrow (4,6)$, and $(6,6) \Rightarrow (5,6)$.

First Decrypted Matrix of **A** (FDM**A**) = $\begin{pmatrix} a_{16} & a_{15} & a_{14} & a_{13} & a_{12} & a_{11} \\ a_{26} & a_{25} & a_{24} & a_{23} & a_{22} & a_{21} \\ a_{36} & a_{35} & a_{34} & a_{33} & a_{32} & a_{31} \\ a_{46} & a_{45} & a_{44} & a_{43} & a_{42} & a_{41} \\ a_{56} & a_{55} & a_{54} & a_{53} & a_{52} & a_{51} \\ a_{66} & a_{65} & a_{64} & a_{63} & a_{62} & a_{61} \end{pmatrix}$

- Dividing **A** by secret key S (Equation (5))

SDM**B**= **A** / S

SDM**B** = $\begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & b_{16} \\ b_{21} & b_{22} & b_{23} & b_{24} & b_{25} & b_{26} \\ b_{31} & b_{32} & b_{33} & b_{34} & b_{35} & b_{36} \\ b_{41} & b_{42} & b_{43} & b_{44} & b_{45} & b_{46} \\ b_{51} & b_{52} & b_{53} & b_{54} & b_{55} & b_{56} \\ b_{61} & b_{62} & b_{63} & b_{64} & b_{65} & b_{66} \end{pmatrix}$

- The secret key S can have any value starting from $\pm 2 .... \pm n$ where n is integral number except 0

- To bring all the columns in the original order in the matrix (Equation (6)).

$$\mathbf{C} = \begin{pmatrix} k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} \\ k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} \\ k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} \\ k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} \\ k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} \\ k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} \end{pmatrix}$$

Illustration 1: let consider the following matrix with order N=4.

$$\mathrm{K} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$$

Encryption

- Now exchange the columns of the matrix **C**.

$$\mathbf{C} = \begin{pmatrix} 4 & 3 & 2 & 1 \\ 8 & 7 & 6 & 5 \\ 12 & 11 & 10 & 9 \\ 16 & 15 & 14 & 13 \end{pmatrix}$$

- Now multiplying **C by** S= 6**,** we give encrypt matrix

$$\mathrm{FEM}\mathbf{B} = \mathrm{W}\mathbf{C}$$

$$\mathrm{FEM}\mathbf{B} = \begin{pmatrix} 24 & 18 & 12 & 6 \\ 48 & 42 & 36 & 30 \\ 72 & 66 & 60 & 54 \\ 96 & 90 & 84 & 78 \end{pmatrix}$$

- All diagonal element are moved to form the first row to construct the encrypt data matrix using equation (3).

E_1 Operation

$a_{(0+1)1} \Rightarrow a_{(0+1)1}$ $\quad$ $a_{(1+1)1} \Rightarrow a_{(1+1)1}$

$a_{(1)1} \Rightarrow a_{(1)1}$ $\quad$ $a_{(2)1} \Rightarrow a_{(2)1}$

$a_{11} \Rightarrow a_{11}$ $\quad$ $a_{21} \Rightarrow a_{21}$

$a_{(2+1)1} \Rightarrow a_{(2+1)1}$ $\quad$ $a_{(3+1)1} \Rightarrow a_{(3+1)1}$

$a_{(3)1} \Rightarrow a_{(3)1}$ $\quad$ $a_{(4)1} \Rightarrow a_{(4)1}$

$a_{31} \Rightarrow a_{31}$ $\quad$ $a_{41} \Rightarrow a_{41}$

$$\text{EM}\mathbf{A} = \begin{pmatrix} 24 & 18 & 12 & 6 \\ 48 & 42 & 36 & 30 \\ 72 & 66 & 60 & 54 \\ 96 & 90 & 84 & 78 \end{pmatrix}$$

E_2 Operation

$a_{(0+1)2} \Rightarrow a_{(0+2)2}$ $\quad$ $a_{(1+1)2} \Rightarrow a_{(1+2)2}$

$a_{(1)2} \Rightarrow a_{(2)2}$ $\quad$ $a_{(2)2} \Rightarrow a_{(3)2}$

$a_{12} \Rightarrow a_{22}$ $\quad$ $a_{22} \Rightarrow a_{32}$

$a_{(2+1)2} \Rightarrow a_{(2+2)2}$ $\quad$ $a_{(3+1)2} \Rightarrow a_{(3+2)2}$

$a_{(3)2} \Rightarrow a_{(4)2}$ $\quad$ $a_{(4)2} \Rightarrow a_{(5)2}$

$a_{32} \Rightarrow a_{42}$ $\quad$ $a_{42} \Rightarrow a_{52}$

$a_{42} \Rightarrow a_{12}$

$$\text{EM}\mathbf{A} = \begin{pmatrix} 24 & 42 & 12 & 6 \\ 48 & 66 & 36 & 30 \\ 72 & 90 & 60 & 54 \\ 96 & 18 & 84 & 78 \end{pmatrix}$$

E_3 Operation

$a_{(0+1)3} \Rightarrow a_{(0+3)3}$ $\quad$ $a_{(1+1)3} \Rightarrow a_{(1+3)3}$

$a_{(1)3} \Rightarrow a_{(3)3}$ $\quad$ $a_{(2)3} \Rightarrow a_{(4)3}$

$a_{13} \Rightarrow a_{33}$ $\quad$ $a_{23} \Rightarrow a_{43}$

$a_{(2+1)3} \Rightarrow a_{(2+3)3}$ $\quad$ $a_{(3+1)3} \Rightarrow a_{(3+3)3}$

$a_{(3)3} \Rightarrow a_{(5)3}$ $\quad$ $a_{(4)3} \Rightarrow a_{(6)3}$

$a_{33} \Rightarrow a_{53}$ $\quad$ $a_{43} \Rightarrow a_{63}$

$a_{33} \Rightarrow a_{13}$ $\quad$ $a_{43} \Rightarrow a_{23}$

$$\text{EM}\mathbf{A} = \begin{pmatrix} 24 & 42 & 60 & 6 \\ 48 & 66 & 84 & 30 \\ 72 & 90 & 12 & 54 \\ 96 & 18 & 36 & 78 \end{pmatrix}$$

$a_{(0+1)4} \Rightarrow a_{(0+4)4}$ $\quad$ $a_{(1+1)4} \Rightarrow a_{(1+4)4}$

$a_{(1)4} \Rightarrow a_{(4)4}$ $\quad$ $a_{(2)4} \Rightarrow a_{(5)4}$

$a_{14} \Rightarrow a_{44}$ $\quad$ $a_{24} \Rightarrow a_{54}$

$a_{24} \Rightarrow a_{14}$

E_4 Operation

$a_{(2+1)4} \Rightarrow a_{(2+4)4}$ $\quad$ $a_{(3+1)4} \Rightarrow a_{(3+4)4}$

$a_{(3)4} \Rightarrow a_{(6)4}$ $\quad$ $a_{(4)4} \Rightarrow a_{(7)4}$

$a_{34} \Rightarrow a_{64}$ $\quad$ $a_{44} \Rightarrow a_{74}$

$a_{34} \Rightarrow a_{24}$ $\quad$ $a_{44} \Rightarrow a_{34}$

$$\text{EM}\mathbf{A} = \begin{pmatrix} 24 & 42 & 60 & 78 \\ 48 & 66 & 84 & 6 \\ 72 & 90 & 12 & 30 \\ 96 & 18 & 36 & 54 \end{pmatrix}$$

- Finally, the analyzed twitter data has been encrypted successfully.

Decryption

$$\text{EM}\mathbf{A} = \begin{pmatrix} 24 & 42 & 60 & 78 \\ 48 & 66 & 84 & 6 \\ 72 & 90 & 12 & 30 \\ 96 & 18 & 36 & 54 \end{pmatrix}$$

- Moving the elements of the first row to form of diagonal elements of the respective columns to decrypt the data using equation (4).

D_1 Operation

$a_{(0+1)1} \Rightarrow a_{(0+1)1}$ $\quad$ $a_{(1+1)1} \Rightarrow a_{(1+1)1}$

$a_{(1)1} \Rightarrow a_{(1)1}$ $\quad$ $a_{(2)1} \Rightarrow a_{(2)1}$

$a_{11} \Rightarrow a_{11}$ $\quad$ $a_{21} \Rightarrow a_{21}$

$a_{(2+1)1} \Rightarrow a_{(2+1)1}$ $\quad$ $a_{(3+1)1} \Rightarrow a_{(3+1)1}$

$a_{(3)1} \Rightarrow a_{(3)1}$ $\quad$ $a_{(4)1} \Rightarrow a_{(4)1}$

$a_{31} \Rightarrow a_{31}$ $\quad$ $a_{41} \Rightarrow a_{41}$

$$\text{FDM}\mathbf{A} = \begin{pmatrix} 24 & 42 & 60 & 78 \\ 48 & 66 & 84 & 6 \\ 72 & 90 & 12 & 30 \\ 96 & 18 & 36 & 54 \end{pmatrix}$$

D_2 Operation

$a_{(0+1)2} \Rightarrow a_{(0+4)2}$ $\quad$ $a_{(1+1)2} \Rightarrow a_{(1+4)2}$

$a_{(1)2} \Rightarrow a_{42}$ $\quad$ $a_{(2)2} \Rightarrow a_{(5)2}$

$a_{12} \Rightarrow a_{42}$ $\quad$ $a_{22} \Rightarrow a_{52}$

$a_{22} \Rightarrow a_{12}$

$a_{(2+1)2} \Rightarrow a_{(2+4)2}$ $\quad$ $a_{(3+1)2} \Rightarrow a_{(3+4)2}$

$a_{(3)2} \Rightarrow a_{(6)2}$ $\quad$ $a_{(4)2} \Rightarrow a_{(7)2}$

$a_{32} \Rightarrow a_{62}$ $\quad$ $a_{42} \Rightarrow a_{72}$

$a_{32} \Rightarrow a_{22}$ $\quad$ $a_{42} \Rightarrow a_{32}$

$$\text{FDM}\mathbf{A} = \begin{pmatrix} 24 & 18 & 60 & 78 \\ 48 & 42 & 84 & 6 \\ 72 & 66 & 12 & 30 \\ 96 & 90 & 36 & 54 \end{pmatrix}$$

$a_{(0+1)3} \Rightarrow a_{(0+3)3}$ $a_{(1+1)3} \Rightarrow a_{(1+3)3}$

$a_{(1)3} \Rightarrow a_{(3)3}$ $a_{(2)3} \Rightarrow a_{(4)3}$

$a_{13} \Rightarrow a_{33}$ $a_{23} \Rightarrow a_{43}$

D_3 Operation

$a_{(2+1)3} \Rightarrow a_{(2+3)3}$ $a_{(3+1)3} \Rightarrow a_{(3+3)3}$

$a_{(3)3} \Rightarrow a_{(5)3}$ $a_{(4)3} \Rightarrow a_{(6)3}$

$a_{33} \Rightarrow a_{53}$ $a_{43} \Rightarrow a_{63}$

$a_{33} \Rightarrow a_{13}$ $a_{43} \Rightarrow a_{23}$

$$\text{FDM}\mathbf{A} = \begin{pmatrix} 24 & 18 & 12 & 78 \\ 48 & 42 & 36 & 6 \\ 72 & 66 & 60 & 30 \\ 96 & 90 & 84 & 54 \end{pmatrix}$$

$a_{(0+1)4} \Rightarrow a_{(0+2)4}$ $a_{(1+1)4} \Rightarrow a_{(1+2)4}$

$a_{(1)4} \Rightarrow a_{(2)4}$ $a_{(2)4} \Rightarrow a_{(3)4}$

$a_{14} \Rightarrow a_{24}$ $a_{24} \Rightarrow a_{34}$

$a_{24} \Rightarrow a_{34}$

D_4 Operation

$a_{(2+1)4} \Rightarrow a_{(2+2)4}$ $a_{(3+1)4} \Rightarrow a_{(3+2)4}$

$a_{(3)4} \Rightarrow a_{(4)4}$ $a_{(4)4} \Rightarrow a_{(5)4}$

$a_{34} \Rightarrow a_{44}$ $a_{44} \Rightarrow a_{54}$

$a_{44} \Rightarrow a_{14}$

$$\text{FDM}\mathbf{A} = \begin{pmatrix} 24 & 18 & 12 & 6 \\ 48 & 42 & 36 & 30 \\ 72 & 66 & 60 & 54 \\ 96 & 90 & 84 & 78 \end{pmatrix}$$

- Dividing the matrix elements of **A** by the secret key S.

$$\text{SDM}\mathbf{B} = \mathbf{A} / \text{S}$$

$$\text{SDM}\mathbf{B} = \begin{pmatrix} 4 & 3 & 2 & 1 \\ 8 & 7 & 6 & 5 \\ 12 & 11 & 10 & 9 \\ 16 & 15 & 14 & 13 \end{pmatrix}$$

- To bring all the columns in the original order in the matrix and produce the original matrix.

$$\mathbf{C} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \end{pmatrix}$$

- Finally, the encrypted matrix data can be decrypted successfully.

4 Result & Discussion

The SRB20, SRB19, Salsa, and SRB18 algorithm have the same order of the matrix and memory size. These algorithms are based on various matrix dimensions and memory size.

$$\text{File Size of matrix} = 4 + (3\,(N-1)*2) + 2(N-1)^2 \quad \textbf{(7)}$$

where N is order of matrix.

Using this above equation calculation the file size of the matrix. If number of characters is greater than the order of the matrix then adds one byte with each additional character. In 1x1 matrix order element memory size is 4 bytes. In 2x2 matrix order has (1,1) element memory size is 4 bytes, (1,2) element memory size is 3 bytes, (2,1) element memory size is 3 bytes, and (2,2) element memory size is 2 bytes. In 3x3 matrix order has (1,1) element memory size is 4 bytes, (1,2) element memory size is 3 bytes, (1,3) element memory size is 3 bytes, (2,1) element memory size is 3 bytes, (2,2) element memory size is 2 bytes, (2,3) element memory size is 2 bytes, (3,1) element memory size is 3 bytes, (3,2) element memory size is 2 bytes, and (3,3) element memory size is 2 bytes. Thus the total file size of a 3x3 order is 24 bytes. Similarly, to all order matrix and file size are show in Table 3.

S. No.	Order of Matrix	File Size
1	3x3	24 bytes
2	5X5	76 bytes
3	6x6	111 bytes
4	10x10	312 bytes
5	15x15	822 bytes
6	20x20	1531 byes
7	40x40	6580 bytes

Table 3: File Size

Configuration: Intel Core i5 8th Generation CPU with 20 GB RAM running on Windows 10.

Table 4 is the SRB20, SRB19, SRB18 and Salsa algorithm has compared for encryption speed is micro seconds (μs). These algorithm has 24 bytes of data taking encryption time is 329.3 μs for Salsa, 397 μs for SRB18, 1184.9 μs for SRB19 and 1751 μs for SRB20. Similarly, 76 bytes of data taking encryption time are 349.2 μs for Salsa, 457.3 μs for SRB18, 1384.1 μs for SRB19 and 1693 μs for SRB20. 111 bytes of data taking encryption time are 343 μs for Salsa, 403.3 μs for SRB18, 1215.1 μs for SRB19 and 1712.3 μs for SRB20. 312 bytes of data taking encryption time are 345.3 μs for Salsa, 439.7 μs for SRB18, 1329.4 μs for SRB19 and 6599.1 μs for SRB20. 822 bytes of data taking encryption time are 512.5 μs for Salsa, 440.3 μs for SRB18, 1439.2 μs for SRB19 and 2895 μs for SRB20. 1531 bytes of data taking encryption time are 401.6 μs for Salsa, 467.6 μs for SRB18, 1534.8

µs for SRB19 and 2082.7 µs for SRB20. 6580 bytes of data taking encryption time are 274.2 µs for Salsa, 489.2 µs for SRB18, 1881.8 µs for SRB19 and 2484.9 µs for SRB20.

S. No.	File Size	Encryption Speed(µs)			
		Salsa	SRB18	SRB19	SRB20
1	24 bytes	329.3	397	1184.9	1751.0
2	76 bytes	349.2	457.3	1384.1	1693.0
3	111 bytes	343	403.3	1215.1	1712.3
4	312 bytes	345.3	439.7	1329.4	6599.1
5	822 bytes	512.5	440.3	1439.2	2895.0
6	1531 byes	401.6	467.6	1534.8	2082.7
7	6580 bytes	274.2	489.2	1881.8	2484.9

Table 4: Time Complexity for Encryption

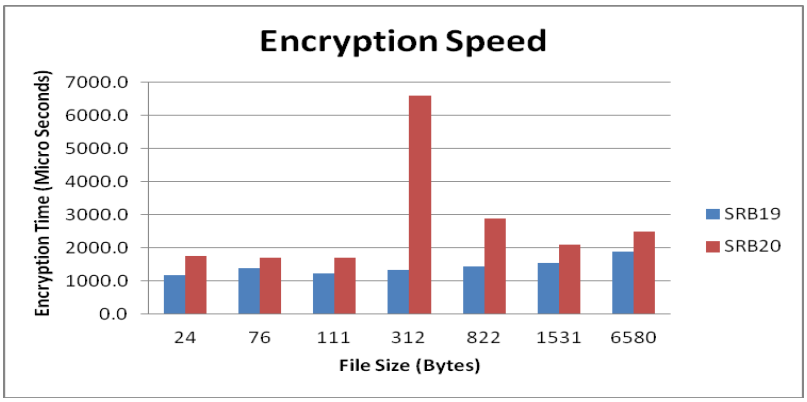

Figure 1: Encryption speed of SRB19 & SRB20

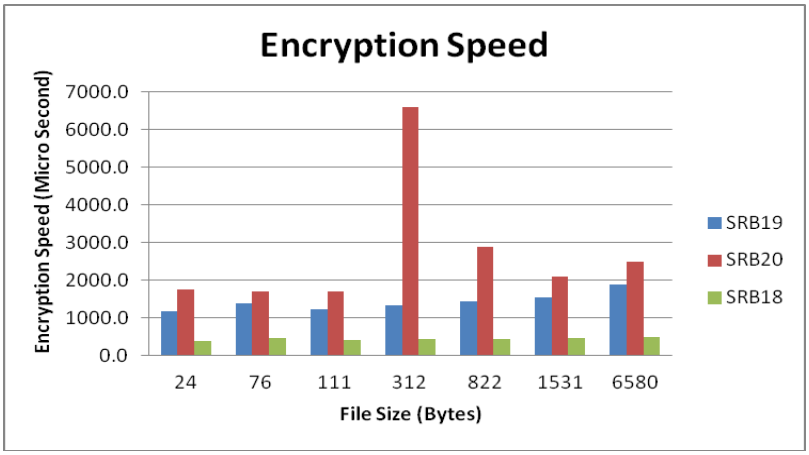

Figure 2: Encryption Speed of SRB Family

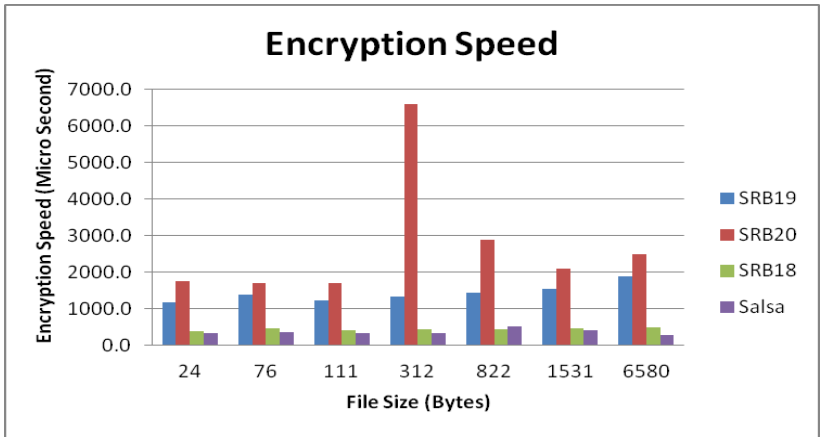

Figure 3: Comparison of Encryption Speed

From Figure 1, 2 and 3. It is observed that the order of the matrix is 3,5,6,10,15,20, and 40 for SRB20, SRB19, SRB18 and Salsa algorithm has compared for encryption speed is micro seconds (µs). These algorithms have an order of matrix 3 and the time taken for encryption are 329.3 µs for Salsa, 397 µs for SRB18, 1184.9 µs for SRB19 and 1751 µs for SRB20. Similarly, order of matrix 5 and the time taken for encryption are 349.2 µs for Salsa, 457.3 µs for SRB18, 1384.1 µs for SRB19 and 1693 µs for SRB20. Order of matrix 6 and the time taken for encryption are 343 µs for Salsa, 403.3 µs for SRB18, 1215.1 µs for SRB19 and 1712.3 µs for SRB20. . Order of matrix 10 and the time taken for encryption are 345.3 µs for Salsa, 439.7 µs for SRB18, 1329.4 µs for SRB19 and 6599.1 µs for SRB20. . Order of matrix 15 and the time taken for encryption are 512.5 µs for Salsa, 440.3 µs for SRB18, 1439.2 µs for SRB19 and 2895 µs for SRB20. Order of matrix 20 and the time taken for encryption are 401.6 µs for Salsa, 467.6 µs for SRB18, 1534.8 µs for SRB19 and 2082.7 µs for SRB20. . Order of matrix 40 and the time taken for encryption are 274.2 µs for Salsa, 489.2 µs for SRB18, 1881.8 µs for SRB19 and 2484.9 µs for SRB20. Finally, SRB20 algorithm takes more time while compared to a SRB family and also Salsa algorithm, but SRB20 algorithm has more security when compared to SRB family and Salsa algorithm.

5 Conclusions

In this work, to classified the tweets using Rstudio on Twitter. These tweets are used to analyze positive and negative tweets to make polarity scored. The polarity scores result data could be extracted from Twitter. Extracted data are prone to have security issues. Hacking the data easily and changed the score results lead to lots of issues like affecting the economic status, company brand, and reputation of firms. Salsa20/4 is faster encryption because of quarter round which also better security. The novel algorithm is proposed by modified the Salsa20/4 to enhance the further the security of the accumulated data. SRB18 has two stages. The first stage of SRB18 is the secret key and the second stage of SRB18 is the diagonal elements in N-1 steps. SRB18 results show that the encryption algorithm takes high execution time because of secret key and high data security when compared to Salsa20 variant. SRB19 has two stages. The first stage of SRB19 is the column operations and the second stage of SRB19 is the diagonal elements in N-1 steps. SRB19 results show that the encryption algorithm takes high execution time compared to both SRB18 and Salsa20 variant. The proposed algorithm SRB20 has three stages. The first stage of SRB20 is column operation , second stage of SRB20 is the secret key and the third stage of SRB20 is the diagonal elements in N-1 steps. SRB20

results show that the encryption algorithm takes high execution time because of high data security when compared to SRB18, SRB19, and Salsa20 variant. The advantage of SRB20 algorithm is decryption algorithm which is huge time taken when compared to other algorithms. In future, the secret key will be as fraction value and still need to add more operations of the data security.

References

Lin Ding (2019). Improved Related-Cipher Attack on Salsa20 Stream Cipher. *IEEE Access* (vol. 7, pp. 30197 – 30202).

Sabyasachi Dey and Santanu Sarkar (2017). Improved analysis for reduced round Salsa and Chacha. *Discrete Applied Mathematics*. Elsevier.

Kakumani K. C. Deepthi and Kunwar Singh (2018). Cryptanalysis of Salsa and ChaCha: *Revisited. ICST Institute for Computer Sciences, Social Informatics and Telecommunications Engineering* (pp. 324–338).

Raj R. Parmar, Sudipta Roy, Debnath Bhattacharyya, Samir Kumar Bandyopadhyay, and Tai-Hoon Kim (2017). Large-Scale Encryption in the Hadoop Environment: Challenges and Solutions. *IEEE Access* (vol. 5, pp.7156-7163).

Subhamoy Maitra (2016). Chosen IV cryptanalysis on reduced round ChaCha and Salsa. *Discrete Applied Mathematics* (pp. 88-97). Elsevier.

Arka Rai Choudhuri and Subhamoy Maitra (2016). Significantly Improved Multi-bit Di erentials for Reduced Round Salsa and ChaCha. *IACR Transactions on Symmetric Cryptology* (vol. 2016, no. 2, pp. 261-287).

Daniel J. Bernstein (2008). The Salsa20 family of stream ciphers. *In New Stream Cipher Designs; Berlin, Germany* (pp. 84-97). Springer.

Doug Laney (2001). 3D data management: Controlling data volume, velocity, and variety. Application Delivery Strategies. META Group.

Joan Daemen and Vincent Rijmen (1999). The Rijndael block cipher. *NIST Computer Security Resource Center*. Document version 2 (pp. 1-45).

Cuneyt Gurcan Akcora, Murat Ali Bayir, Murat Demirbas, and Hakan Ferhatosmanoglu (2010). Identifying breakpoints in public opinion. *1st Workshop on Social Media Analytics* (pp. 62–66).

Bharath Sriram, David Fuhry, Engin Demir, Hakan Ferhatosmanoglu, and Murat Demirbas (2010). Short text classification in Twitter to improve information filtering. *In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development* (pp. 841–842).

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan (2002). Thumbs up?: Sentiment classification using machine learning techniques. *In Proceedings of the Conference on Empirical Methods Natural Language Processing 2002* (vol. no.10. pp. 79–86).

Anastasia Giachanou and Fabio Crestani. Like It or Not: A Survey of Twitter Sentiment Analysis Methods. *ACM Computing Surveys* (vol. 49, no. 2. pp.28:1-28:41).

Diyana Afdhila, Surya Michrandi Nasution, and Fairuz Azmi (2016). Implementation of Stream Cipher Salsa20 Algorithm to Secure Voice on Push to Talk Application. *IEEE Asia Pacific Conference on Wireless and Mobile* (pp. 137–141).

Prateek Yadav, Indivar Gupta, and S.K.Murthy (2016). Study and Analysis of eSTREAM Cipher Salsa And ChaCha. *In 2nd IEEE International Conference on Engineering and Technology*.

Alireza Jolfaei, Abdolrasoul Mirghadri, and Ahmadreza Vizandan (2012). Impact of Rotations in the Salsa20/8 Image Encryption Scheme. *International Journal of Computer Theory and Engineering* (vol. 4, no. 6. pp. 938-943).

Rajeev Sobti and Geetha Ganesan (2016). Analysis of Quarter Rounds of Salsa and ChaCha Core and Proposal of an Alternative Design to Maximize Diffusion. *Indian Journal of Science and Technology* (vol. 9(3)).

Zeng-yu Shao and Lin Ding (2012). Related-Cipher Attack on Salsa20. *Fourth International Conference on Computational and Information Sciences* (pp. 1182-1185). IEEE.

Mishal Almazrooie, Azman Samsudin, and Manmeet Mahinderjit Singh (2015). Improving the Diffusion of the Stream Cipher Salsa20 by Employing a Chaotic Logistic Map. *Journal of Information Processing Systems* (vol.11, no.2. pp.310-324).

Daniel J. Bernstein (2008). ChaCha, a variant of Salsa20. *Workshop Record of SASC* 2008 (pp. 1-6).

Paul Crowley (2006). Truncated Differential Cryptanalysis of Five Rounds of Salsa20. *SASC 2006 - Stream Ciphers Revisited, Workshop Record* 2006(pp.198-202). Retrieved from http://www.ecrypt.eu.org/stvl/sasc2006/

Simon Fischer, Willi Meier, Come Berbain, Jean-Francois Biasse, and M.J.B. Robshaw (2006). Non-Randomness in eSTREAM Candidates Salsa20 and TSC-4. *Indocrypt. LNCS* 4329 (pp.2-16). Springer.

Bagath Basha C and Somasundaram K (2019). A Comparative Study of Twitter Sentiment Analysis Using Machine Learning Algorithms in Big Data. *International Journal of Recent Technology and Engineering* (vol. 8. pp. 591-599).