International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v10i2s.5945
***Article Received**: 10 October 2022 **Revised**: 15 November 2022 **Accepted**: 24 December 2022*

128-bit Vectorization on Cha-Cha20 Algorithm for Device-to-Device Communication

[1]M. Satish Kumar, [2]Dr. PVRD Prasada Rao
[1]Research Scholar, Department of Computer Science and Engineering
[2]Professor, Department of Computer Science and Engineering
Koneru Lakshmaiah Education Foundation, Guntur, Andhra Pradesh, India
E-Mail: satish.manchala@yahoo.com.
[1]https://orcid.org/0000-0002-4227-2202

Abstract: In 5G networks, device to device(D2D) communication is the advanced technology, and it has the major advantages when compared with traditional systems. The coverage area of the system increases by device-to-device transmission which renders with low latency. No doubt, for the upcoming technologies D2D communications is an added advantage but in various views this kind of transmission is still under risk. D2D communication is transferring the information from one device to another device without the involvement of base station. So, the communication is possible with less delay time. Attackers are possible into the D2D communications. To provide the security in communication, encryption algorithms are used. The main theme of the Encryption algorithms is it should execute with less delay time to provide faster communication and in particular CHA-CHA20 offers the secure communication for encrypting the data packets to securing the data. Vectorization on Cha-Cha20 Algorithm provides the security with less delay time compared to AES encryption Algorithm and Cha-Cha20 algorithm. Compared to other encryption algorithms in cryptography, Cha-Cha20 is suited for resource constrained devices.

Keywords: Device to device communication, AES encryption algorithm, Cha-Cha20 encryption algorithm, Vectorization on Cha-Cha20 algorithm.

I. Introduction:

Device to Device communication is a raising technique in LTE-A systems. Considering the benefits of these proximity appliances, this presents high output with less delay time and discharges mobile network congestion. Apart from this it provides various types of user-friendly services. More research must be done in terms of design issues in D2D. Security in D2D transmission is still in inceptive state.

5G is the union of various techniques which executes under identical policy. Numerous techniques like Massive MIMO, Ultra-Dense network, Small Cell, Cognitive radio etc. are the primary techniques utilized in 5G community [1]. Each one is responsible for specific property to the network. Massive MIMO has the property of amplifying the spectrum capabilities. Small cells also perform the job of controlling numerous clients effectively. To make the system trustworthy and better these capable techniques are essential. D2D transmission comes into this category that can provide this technique with less lag and greater capability. It is smartly defined as the direct
interconnection among the users without any restriction from the device [3],[4]. D2D communication offers numerous benefits but more prone to security towards the end users, where more research has to be concentrated in this area.[5],[6].

1.1. D2D COMMUNICATION

D2D is the most advanced technology [7] that allows transmission on direct basis among the two devices. It can play major role in upcoming wireless networks. These networks permit mobile communication with optimal performance [8],[9]. On the basis of functionality these networks are divided into the various forms as described below.

1.1.1. Classification of Device-to-Device Communication: D2D communication is of two types. In-band and Out-band.

1.1.1. In-band D2D communication: In-band uses mobile frequencies for transmission among the devices which don't need of decryption of data among the users. In-band additionally classified into 2 types, Underlay and Overlay. Underlay avoids the use of committed paths for transmitting and intervention management is easy, but in Overlay it is difficult to manage intervention and to execute the transmission among the devices dedicated paths are offered.

1.1.2. Out-band D2D communication: Out-band do not use mobile frequencies. This transmission need encrypting of data among the users. Out-band is again divided into 2 types Controlled and Autonomous. In Controlled maintaining intervention is easy to setup and in Autonomous maintaining intervention is not in the hands of system. Committed paths are no more needed and QoS is less.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v10i2s.5945
***Article Received**: 10 October 2022 **Revised**: 15 November 2022 **Accepted**: 24 December 2022*

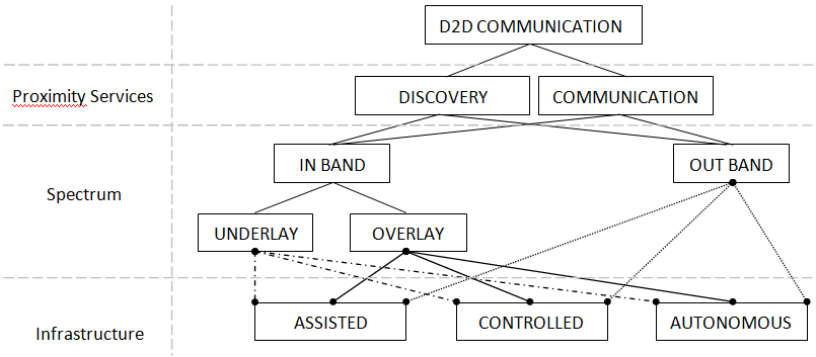

Fig.1. In-band and Out-band communication

Moreover, due to the intrinsically open nature of cellular communications and also the dynamics of cellular communications causes network / communication vulnerability for security attacks such as Fake Message, Privacy Violation and Eavesdropping. Security for the communicated data is required to protect data from hackers and attacks. For example, mobile health applications strongly Require the security and privacy applications. The major security requirements for 5G based D2D communication is follows:

- **Confidentiality and Integrity:** packets are encrypted and modified only by the approved entities.
- **Authentication:** identification of each users involved in the data transmission
- **Privacy:** hides the private information about the system from the attackers.

Encryption algorithms like AES (Advanced Encryption Standard) and Cha-Cha 20 provides security. This paper addresses the security issues of confidentiality and also authentication for delivering the secure D2D based multimedia communications. The energy consumption of D2D is addressed in this paper by means of secure communication through Cha-Cha20 encryption algorithm. Further, minimizes the data traffic for multimedia services offering in the 5G environment.

The rest of the paper is structured by follows Section 2 contains the literature review for the proposed manuscript. Section 3 covers AES Algorithm. Cha-Cha20 algorithm will be explained in section v. Section 4 introduces about Vectorization on Cha-Cha20 algorithm. Results and analysis can be made in section-V. Section-VI completes with future scope.

II. Literature Review

A lightweight and robust security based D2D communication model is presented for data transmission. This concept is suited for healthcare applications. This paper uses elliptic curved discrete logarithm problem which does not uses any pairing. This gives the lowest computational cost for comparing with the previous security schemes. Further this paper addresses the unforgeability and confidentiality for communicating between the devices [11]. One of the famous security techniques for security is AES which is used for encryption technology [12] that protects the data security and integrity for IoT. In general, AES algorithm is fixed with the packet length of 128 bits and also key size considered as 128 bits, 192 bits and 256 bits. Depends upon the number of rounds, key size is considered.

Cha-Cha20 is a stream cipher encryption algorithm that used for high-speed embedded algorithms. For instance, deployment of fast and secure connections between IoT nodes and other systems are introduced. For testing the algorithm to test the hardware efficiency, ARM Cortex M4 processors are used [13]. In [14], fault data injection attack is detected and mitigated using Cha-Cha and Salsa20 encryption algorithms. This algorithm is tested for three kinds of architectures as IA-32, Intel 64 and others. The attackers extract the keys from the algorithm for the number of round operations.

III. AES *Encryption Algorithm:*

AES is Advanced Encryption Standard used for the encryption in cellular mobile communication and D2D communication. Inputs are128-bit plain text and key. To get 128-bit cipher text, steps in AES are:

- **Add round key:** It performs the XOR operation of plain text with 128-bit key.
- **Substitution bytes:** Matrix values will change according to standards-box. Example: If 8-digit of add round output is 11100111, and then value present at 15th row and 7th column i.e., 10101101 will be substituted in place of 11100111.
- **Shift rows:** Circularly right-shifts the matrix by 1.
- **Mix columns:** Multiplies the output of shift rows by pre-defined matrix.
- **Add round key:** Performs the XOR operation for output of mix columns with 128-bit key [2].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v10i2s.5945
***Article Received**: 10 October 2022 **Revised**: 15 November 2022 **Accepted**: 24 December 2022*

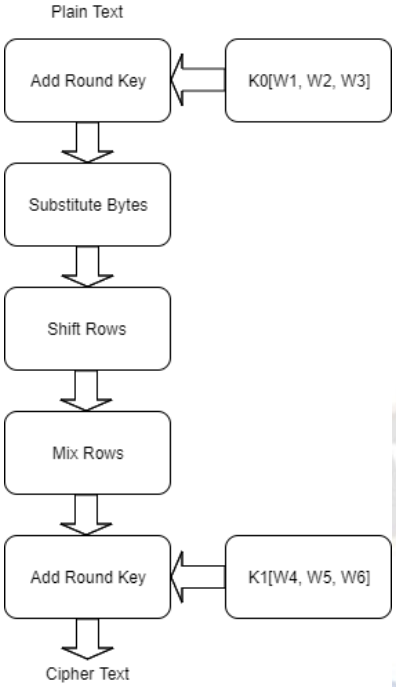

Fig.2. AES Algorithm Steps

In AES algorithm, all these steps are performed 10 times to get cipher text. Cipher text is 128-bit length.

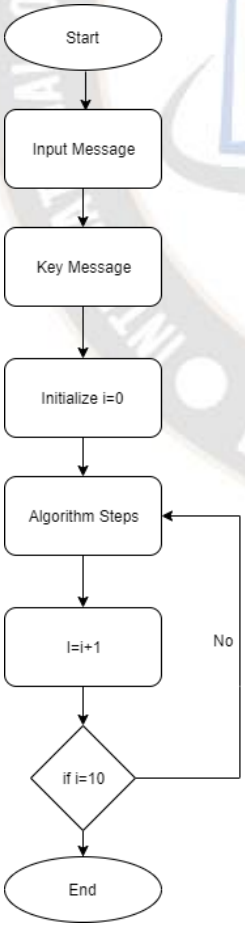

Fig.3. Flow chart of AES algorithm

IV. ChaCha-20 Algorithm:

Cha-Cha20 algorithm is another encryption algorithm. In Cha-Cha20 algorithm inputs are input message of 256 bits, 3 nonce number of 32-bits, 32-bit counter value and 128-bit predefined constants [3]. Algorithm steps are as follows:

- **Matrix generation:** In matrix generation, first row consists of constant values. Second and third row are the inputs. Fourth row contains counter and nonce values.
- **Cha-Cha Algorithm steps:** Algorithm steps consist of XOR operations, add and shift operations. These operations will be performed for 20 times so the name Cha-Cha20.
- **Output:** Output of Cha-Cha20 algorithm is 512 bits.

Fig5 shows the block diagram of Cha-Cha20 Algorithm.

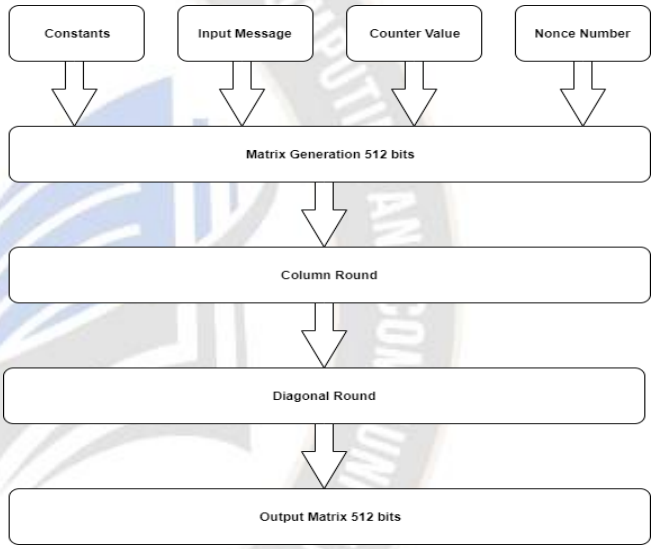

Fig 4. Block diagram of Cha-Cha20 Algorithm

In the Cha-Cha 20 algorithm column round is performed 10times and diagonal round is performed 10 times to output matrix if 512 bits.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v10i2s.5945
Article Received: 10 October 2022 Revised: 15 November 2022 Accepted: 24 December 2022

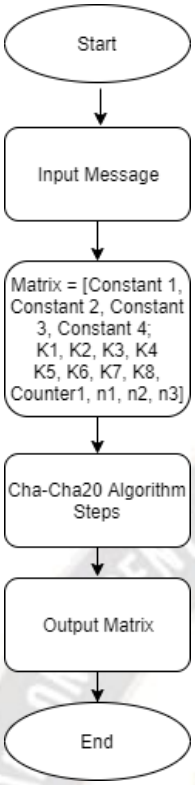

Fig.5.Flowchart for cha-cha 20 algorithm

In the Cha-Cha-20 algorithm steps two operations are going to perform. Those two operations are column round and diagonal round.

Figure 7 shows the flowchart for column round and figure 8 shows the diagonal round in detail.

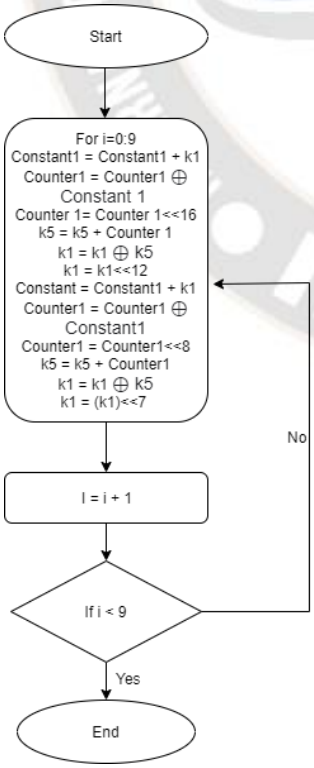

Fig.6.Column round flow chart

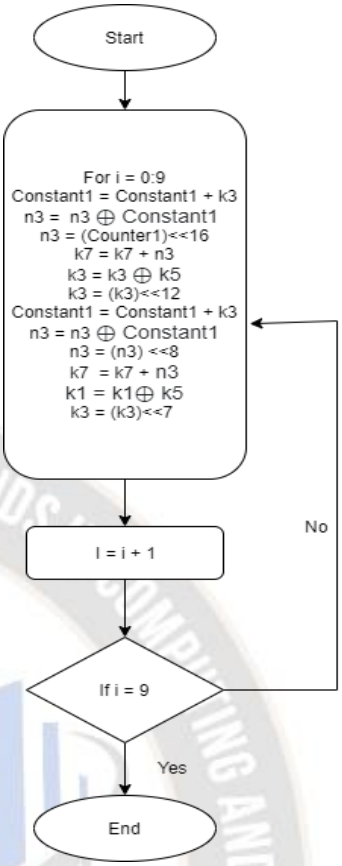

Fig.7.Diagonal round flow chart

V. VECTORIZATION ON CHACHA20ALGORITHM:

Output from the Cha-Cha20 algorithm is 4Å~4 matrix of 512 bits with each element is of 32-bits. 128- Vectorization is performed on each row. Complete row1 forms vector1, row2 forms vector2 and so on.

Vectorization1 = Row1(Col1+Col2+Col3+Col4) = 32bits+32bits+32bits+32bits
Vectorization2 = Row2(Col1+Col2+Col3+Col4) = 32bits+32bits+32bits+32bits
Vectorization3 = Row3(Col1+Col2+Col3+Col4) = 32bits+32bits+32bits+32bits
Vectorization4 = Row4(Col1+Col2+Col3+Col4) = 32bits+32bits+32bits+32bits

In Cha-Cha 20 we are performing diagonal round and column round operations in four sets because in Cha-Cha 20 input matrix is 4×4 matrix. But in vectorization technique we converted 4×4 matrix to 4×1 matrix so diagonal round and column round operations can be performed only one time and code executes with lesser delay time [6].

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v10i2s.5945
Article Received: 10 October 2022 Revised: 15 November 2022 Accepted: 24 December 2022

VI. RESULS AND NALYSIS

In this section, the performance of the AES and Cha-Cha algorithm is compared with each other. Execution time and output for the AES algorithm are as follows. Execution time is the actual time required for the completion of encryption and the decryption process.

```
          LineStyle:  -
          LineWidth: 0.5000
             Marker: 'none'
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [1×32 double]
              YData: [1×32 double]
              ZData: [1×0 double]

  Show all properties


time =

   22.7349
```

Fig.8.Execution time of AES

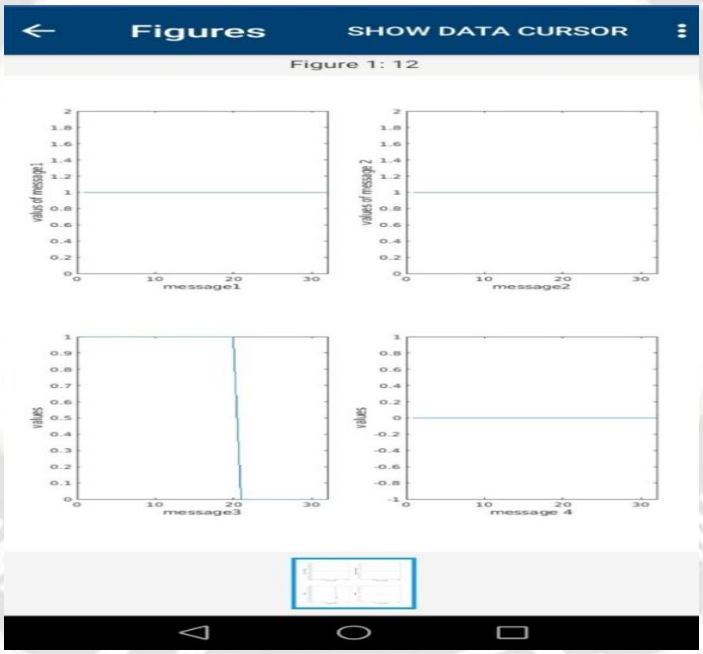

Fig.9.AES Algorithm output

Execution time and output are as follows:

```
         MarkerSize: 6
    MarkerFaceColor: 'none'
              XData: [1×32 double]
              YData: [1×32 double]
              ZData: [1×0 double]

  Show all properties


time1 =

   11.2052
```

Fig.10.Cha-Cha20 Execution time

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v10i2s.5945
Article Received: 10 October 2022 Revised: 15 November 2022 Accepted: 24 December 2022

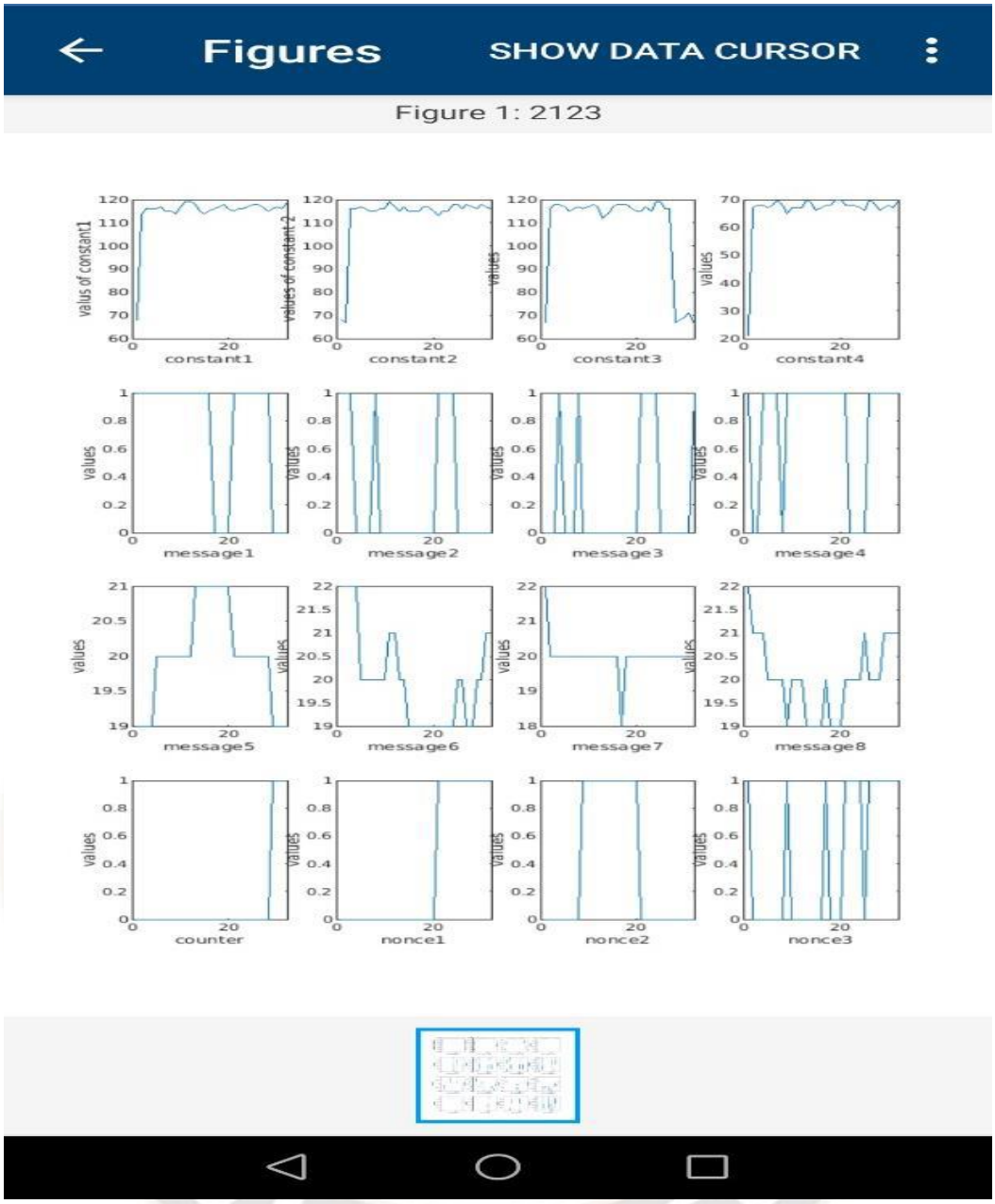

Fig.11.Cha-cha20 output waveform

Execution time and output of vectorization are as follows:

```
            LineStyle: '-'
            LineWidth: 0.5000
               Marker: 'none'
           MarkerSize: 6
      MarkerFaceColor: 'none'
                XData: [1×32 double]
                YData: [1×32 double]
                ZData: [1×0 double]

  Show all properties


time1 =

   10.5376
```

Fig.12.Vectorization on Cha-cha20 Execution time

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v10i2s.5945
Article Received: 10 October 2022 Revised: 15 November 2022 Accepted: 24 December 2022

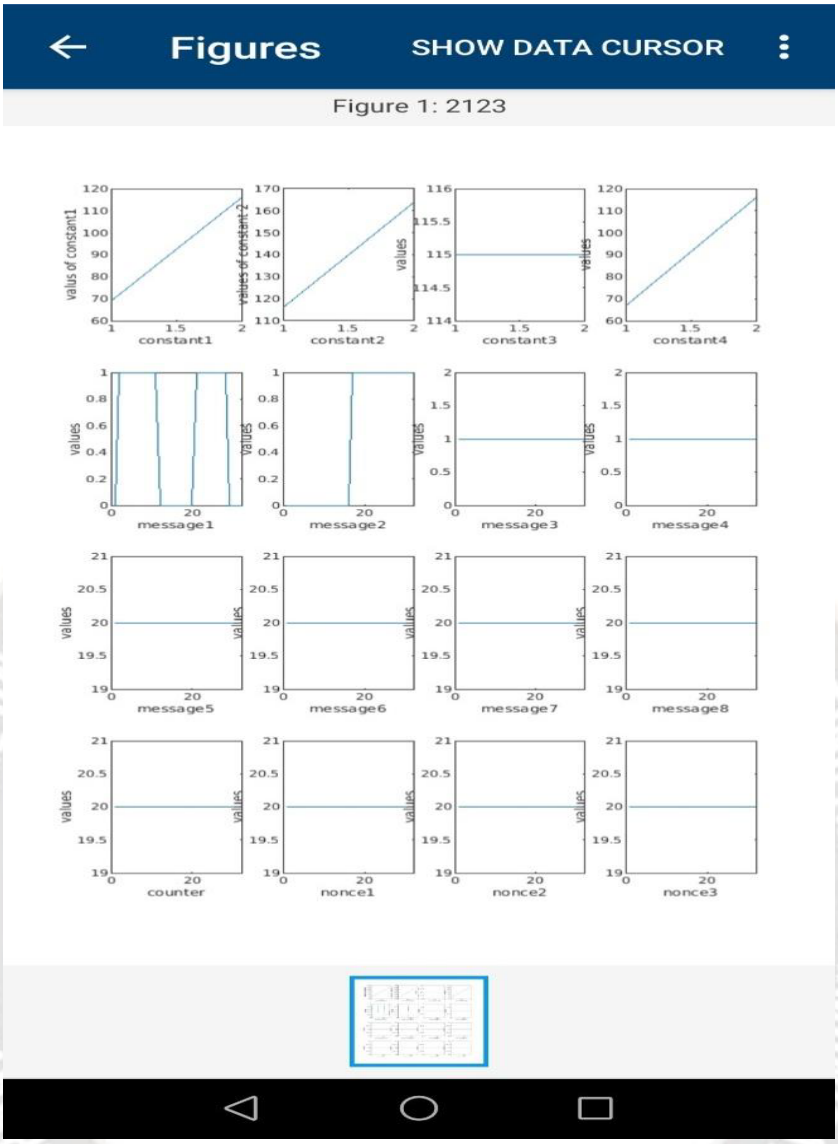

Fig.13.Vectorization on Cha-Cha20 output waveform

Table 1. Comparison Table for AES Algorithm

Parameters	AES Algorithm	Cha-Cha-20 Algorithm	Vectorization on Cha-Cha 20
Input Message	Code on AES Cha-Cha and Vectorization	Code on AES Cha-Cha and vectorization	Code on AES Cha-Cha and vectorization
Output bits	128 bits	512 bits	512 bits
Time taken for execution	22.7seconds	11.2seconds	10.53seconds
Though put	5.6bits/sec	45.71bits/sec	48.62bits/sec
Efficiency	54.3%	54.3%	54.3%
BER	45.6%	45.6%	45.6%

As we see from the comparison table 1 AES algorithm takes more time for execution on mobile devices. Cha-Cha 20 algorithm takes moderate time and vectorization on cha-cha20 algorithm least time for execution. Throughput is least for AES and more vectorization.

VII. CONCLUSION

From the above discussion we can conclude that 128-bit Vectorization on Cha-Cha20 algorithm works faster than Cha-Cha20 and AES algorithms. So, for the device-to-device communication vectorization on Cha-Cha20 algorithm used for encryption. Cha-Cha20 algorithm provides the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 10 Issue: 2s
DOI: https://doi.org/10.17762/ijritcc.v10i2s.5945
***Article Received**: 10 October 2022 **Revised**: 15 November 2022 **Accepted**: 24 December 2022*

encryption, but for more security poly1305 authentication algorithm can be used along with Cha- Cha20 in device-to-device communication. As compared to the AES for Cha-Cha 20, the performance is increases by 11% than Cha-Cha 20. As a future scope we work on the 256-bit vectorization and poly-1305 authentication algorithm for improving the security and integrity of the proposed scheme.

Declaration:

Ethics Approval and Consent to Participate: *No participation of humans takes place in this implementation process*

Human and Animal Rights: *No violation of Human and Animal Rights is involved.*

Funding: *No funding is involved in this work.*

Conflict of Interest: *Conflict of Interest is not applicable in this work.*

Authorship contributions: *There is no authorship contribution*

Acknowledgement: *There is no acknowledgement involved in this work*

References

[1]. M. Agiwal, A. Roy and N. Saxena, "Next Generation 5G Wireless Networks: A Comprehensive Survey", IEEE Communications Surveys & Tutorials, 2016.; 18(3): 1617-1655

[2]. Gandotra P., Jha R.K., and Jain S., "A survey on device-to-device (d2d) communication: Architecture and security issues," Journal of Network and Computer Applications, 2017; 78: 9 – 29,

[3]. D. Ma and G. Tsudik, "Security and Privacy in Emerging Wireless Networks [Invited Paper]," IEEE Wireless Communications, 2010; 17(5): 12–21

[4]. Koskela, T., Hakola, S., Chen, T., Lehtomaki, J., "Clustering concept using device to-device communication in cellular system" In: Proceedings of the IEEE Wireless Communications and Networking Conference, 2016.

[5]. R. Alkurd, R. M. Shubair, and I. Abualhaol, "Survey on Device to-Device Communications: Challenges and Design Issues," in Proceedings of the IEEE 12th International New Circuits and Systems Conference (NEWCAS), 2014; 361–364,.

[6]. F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks," in Proceedings of the 7th International Workshop on Security Protocols, 1999; 172–182,.

[7]. D. Feng, L. Lu, Y. Yuan-Wu, G. Y. Li, S. Li, and G. Feng, "Device-to Device Communications in Cellular Networks," IEEE Communications Magazine, 2014; 52(4): 49–55,.

[8]. Arash Asadi, Qing Wang and Vincenzo Mancuso, "A Survey on Device-to-Device Communication in Cellular Network", IEEE Communications Surveys & Tutorials, 2014; 16(4): 1801 - 1819.

[9]. Militano, L., "Device-to-Device Communications for 5G Internet of Things.", IOT, EAI, 2015

[10]. Asadi, A., Wang, Q., & Mancuso, V. (2014). A Survey on Device-to-Device Communication in Cellular Networks. IEEE Communications Surveys & Tutorials, 16, 1801-1819

[11]. Zhang, A., Wang, L., Ye, X., & Lin, X. (2017). Light-Weight and Robust Security-Aware D2DAssist Data Transmission Protocol for Mobile-Health Systems. IEEE Transactions on Information Forensics and Security, 12(3), 662–675

[12]. Su, N., Zhang, Y., & Li, M. (2019). Research on Data Encryption Standard Based on AES Algorithm in Internet of Things Environment. 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2071- 2075.

[13]. Santis, F.D., Schauer, A., & Sigl, G. (2017). Cha-Cha20-Poly1305 authenticated encryption for high-speed embedded IoT applications. Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, 692 697.

[14]. Fukushima, K., Xu, R., Kiyomoto, S., & Homma, N. (2017). Fault Injection Attack on Salsa20 and Cha-Cha and a Lightweight Countermeasure. 2017 IEEE Trustcom/BigDataSE/ICESS.

[15]. Goll, M., & Gueron, S. (2014). Vectorization on Cha-Cha Stream Cipher. 2014 11th International Conference on Information Technology: New Generations.