Ultra BRIGHT: A Tiny And Fast Ultra Lightweight Block Cipher For Iot

Deepti Sehrawat, Nasib Singh Gill

Abstract: IoT provides a virtual view of real-life things in a smart environment where security and privacy are of prime concern. Lightweight cryptography is aimed to provide security solutions in IoT based applications. Various algorithms have been presented in this area which are either software-based implementation or hardware-based implementation of lightweight ciphers. In this paper, a software-oriented 32-bit ultra-lightweight block cipher, named UBRIGHT, is proposed for a resource-constrained smart environment. The performance of the proposed cipher, UBRIGHT is assessed in this paper and it fulfills SAC, key sensitivity test, and randomness test. The results of the performance evaluation of the proposed cipher's show that it requires low memory and has good speed.

Index Terms: 32-bit cipher, IoT security, lightweight block cipher, ultra-lightweight block cipher, Ultra BRIGHT, UBRIGHT.

1. INTRODUCTION

Internet of Things (IoT) is also referred to as "Smart Objects Network" where numerous IoT enabled devices are connected. In this smart environment, the resources are accessed centrally via IPv6 [1]. Here, sensitive data is shared among new technologies of IoT [2]. To utilize the benefits of these smart technologies in constrained end nodes, security must be kept at priority. A network might suffer at great extent even if a single node is compromised. Besides, embedded devices in a resource-constrained environment have some limitations like they have limited processing power, low memory, limited computational ability, etc. IoT security covers numerous responsibilities such as establishing access control policies, protection of keys using security mechanisms, etc. [3]. Ciphers are of two types, symmetric and asymmetric key. For authentication, confidentiality and integrity checks mainly symmetric key ciphers are used as these have better performance [4]. Ciphers are used to provide secure communication in the network. Furthermore, traditional crypto algorithms aim to provide high-level security but because of limited resources, IoT based embedded devices could not apply existing cryptographic functions. In this direction, to provide security to resource-constrained IoT environment lightweight cryptography came into existence [5]. This new field of cryptography is more efficient to provide end-to-end communications and adoptability to constrained devices [6]. In this direction, numerous researchers have proposed different lightweight ciphers which are either software-based or hardware-based. In comparison to the hardware-based ciphers, software-oriented proves to be more flexible at comparably lower manufacturing and maintenance costs. Besides, hardware-oriented block ciphers are vulnerable to side-channel attacks, whereas, software-oriented ciphers are free from this type of attack. So, there is a need for a good software-oriented security solution that can provide good security against such mathematical attacks. This paper proposes an ultra-lightweight block cipher, named UBRIGHT. Fast diffusion is achieved in the proposed design by utilizing some performance enhancement techniques like the use of key whitening, round permutation, and different amounts of rotations. Besides, it's simple and efficient structure utilizes small code size, low cost, and gives high speed. The performance of UBRIGHT is analyzed on a 32-bit CPU.

- *Ms. Deepti Sehrawat is currently pursuing Ph. D in Computer Science from M. D. University, Rohtak, India. E-mail: dips.scorpio@gmail.com*
- *Dr. Nasib Singh Gill is at present senior most Professor of Department of Computer Science & Applications, M. D. University, Rohtak, India. E-mail: nasibsgill@gmail.com*

2 RELATED WORK

Block ciphers have fixed length blocks and different operations are applied on these blocks a number of times. Besides, three main operations in cryptography are encryption, decryption, and key schedule. For a cipher to be safe enough there must exist adequate confusion and diffusion. After deep analysis, different types of cryptographic algorithms, especially block ciphers are culminated in [7]. The proposed design of ultra-lightweight block cipher, UBRIGHT is based on ARX structure. To introduce non-linearity in ARX (Addition-Rotation-XOR) ciphers, "modular addition" is used whereas word-wise rotation and XOR operations are used for improved diffusion. It is the internal structure of the cipher where different ARX ciphers vary. In hardware-based ciphers modular addition is considered as a costly operation but in software-based implementations, it is quite cheaper as it requires no additional registers and provides good diffusion. Furthermore, as reported by Felics [8], [9] these types of block ciphers have best performance for microcontrollers. Some of the block ciphers for constraint environment i.e. lightweight block ciphers followed the ARX based structures, these are LEA [10], Compact LEA [11], SIMON [12], SPECK [12], BRIGHT [13], [14, [15], Chaskey [16], and Road RunneR [17].

3 UBRIGHT/ ULTRA BRIGHT – PROPOSED DESIGN

This section presents a security design, named UBRIGHT for IoT based smart applications. UBRIGHT stands for Ultra BRIGHT, which follows the design of BRIGHT [14] cipher. The proposed design has following main features:

1. To provide resistance against weak key attacks, key whitening is used.
2. To further improve the security, rotations of the same amount are not used.
3. In addition to addition mod 2b, round permutations are also applied so that good diffusion can be achieved.
4. Integer type length is defined explicitly for
5. For optimal usage of registers, integer type length is defined explicitly.

3.1 Notation

Following notations are used in the paper:

V_i	i^{th} Word/ i^{th} Branch
+	Addition modulo 2^n
^	n-bit exclusive OR
x<<<m	Left circular shifts by m-bits
x>>>n	Right circular shifts by n-bits
Mk	Master-key
k_i	i^{th} key retrieved from Mk
&	Bitwise AND
C	Constant

3.2 Design

UBRIGHT cipher is proposed for IoT based applications. The design of the proposed ULBC "Ultra-lightweight block cipher" is described into two parts (data processing and key-scheduling). UBRIGHT is a 2-branch GFN “Generalized Feistel Network” and hence reduces the restrictions on the design of internal structure and functions. For k € GF $(2)^n$, 2n round function is a function dependent on key which is a map given by (1):

$$\boldsymbol{R_k}: \boldsymbol{GF}\ (\boldsymbol{2})^{\boldsymbol{n}} \times \boldsymbol{GF}\ (\boldsymbol{2})^{\boldsymbol{n}} \rightarrow \boldsymbol{GF}\ (\boldsymbol{2})^{\boldsymbol{n}} \quad (1)$$

The above map of the cipher is described by the operations it applies. The encryption and decryption functions have four layers. These four layers are pre key-whitening, ARX round operations, round permutation, and post key-whitening. The structure of the proposed cipher, UBRIGHT is depicted in Fig. 1. First, the input plain text is partitioned into two equal-sized words which are also called branches. Then these two words are passed to the first layer which applies pre key-whitening. The output of the previous layer is fed as an input to the next layer for further processing. Like, after applying initial key whitening in the first layer, the two words are now fed to the second layer as a round input where ARX based operations are performed on these two words. The operations of the 2nd and 3rd layers are executed ‘R’ times, where ‘R’ is the number of rounds. The internal structure of the proposed UBRIGHT cipher consists of functions which are so arranged that it prevents the cipher from most of the attacks and also provides fast confusion and diffusion.

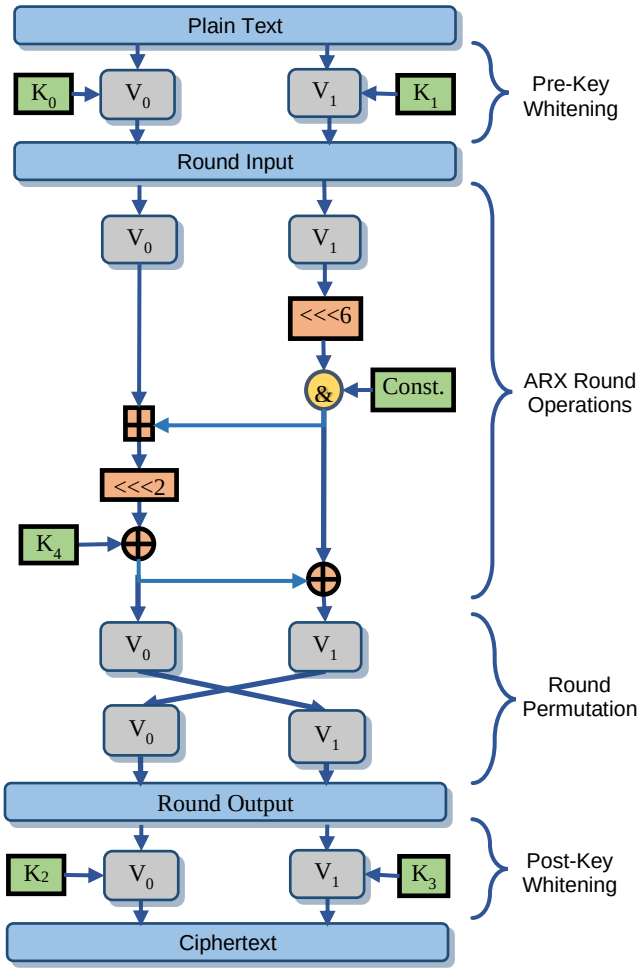

Figure 1. *Layers in the proposed cipher*

3.2.1 Key Whitening

Various researchers have successfully applied key whitening to provide resistance against various attacks. The same was utilized in [18], the key was first XORed before its use as a pre key-whitening affect. This provides the cipher resistance against MITM and brute force attacks but does not guard the cipher from any linear and differential cryptanalysis. So considering this, the proposed cipher UBRIGHT uses key whitening and both pre key whitening and post key whitening are applied to the cipher i.e. before and after round operations. Besides, it is because of the key whitening that makes to extend the attack almost impossible even by one round as it requires to search for all n-keys.

3.2.2 ARX Operations

ARX based ciphers provide more efficient software implementation than any other type of operations in a parallel way. In the proposed UBRIGHT cipher, ARX operations are arranged in an efficient way that makes the cipher to perform fast diffusion. Addition modulo 2^n is used in combination with XOR. Addition modulo 2^n are non-linear operations and when these operations are used along with other operations it propagates differences indefinitely at the same speed as it would be done by bitwise XOR. So, addition modulo 2^n adds non-linearity to the proposed design. If rotation is considered, the proposed cipher does not use the same amount of rotations. The use of addition modulo 2^n and different amounts of shifts add diffusion in the cipher, thereby making a strong diffusion

layer. Besides, to further increase the rate of diffusion constants are also added to the proposed design.

3.2.3 Round Permutation

GFNs have usually slow diffusion property and some attacks can be applied easily to these ciphers like impossible differential cryptanalysis. Contrary, ARX based ciphers have reasonable diffusion speed. So, to improve the diffusion rate of the proposed cipher, a round permutation is applied as the last step in each round.

3.3 Encryption Algorithm

Encryption is composed of round functions that can be represented by (2), read from right to left.

$$\boldsymbol{R_{kr-1}}^{\circ}\ \boldsymbol{R_{kr-2}}^{\circ}\ \ldots\ \boldsymbol{R_{k1}}^{\circ}\ \boldsymbol{R_{k0}} \qquad (2)$$

ROUNDS: 22
INPUT: 32-bits block and m-bits master-key.
OUTPUT: 32-bits block

1. Compute 32-bits input for the second layer after performing pre-whitening on two equal-sized words by partitioning 32-bits plain text 'P' into 2 equal parts and XORing with sub-key i.e.
   $P = P_0 \hat{} k_0 \mid P_1 \hat{} k_1$.
2. Apply ARX operations on P_i for i^{th} round i.e.
   $P_{i1}=(P_{i1} <<< 6)$,
   $P_{i0}=(P_{i0} + (Pi_1 \& C) <<< 2$,
   $P_{i1}=P_{i1} \hat{} (P_{i0} \hat{} k_4)$
3. Perform Round Permutation i.e. $P_{i0}|P_{i1}$, $P_{i1}|P_{i0}$.
4. Repeat operations of step 2 and step 3 for the rest of the rounds.
5. Apply post-whitening as a last step i.e.
   $P = P_0 \hat{} k_2 \mid P_1 \hat{} k_3$.

In the proposed design, there is a low decryption overhead, i.e. applying decryption is quite easy. Here, the same round functions are used but after reversing the order of operations, round keys and constants.

3.4 Key Scheduling

Key scheduling of UBRIGHT is inspired from the Chaskey [16] and RoadRunneR [17] lightweight ciphers. These ciphers do not use any key schedule, as keys are just XORed into the state. The same is followed in the proposed UBRIGHT cipher and on-the-fly key scheduling is used. The 80-bit master key is divided into 5 words of 16-bit, given by (3):

$$Mk = k1 \mid k2 \mid k3 \mid k4 \mid k5 \qquad (3)$$

4 PERFORMANCE EVALUATION

IoT devices interact in a resource-constrained environment with other associated devices and a higher-end backend server. Hence, a good security algorithm should support good performance on different platforms especially, a 32-bit CPU. Implementation results of the proposed cipher are observed on a processor Intel (R) Core (TM) i5-2430M CPU @ 2.20 GHz. The results of the proposed UBRIGHT cipher are obtained on a 32-bit processor in C-language. Fig. 2 and fig. 3 shows the results for encryption and decryption for two different plaintexts.

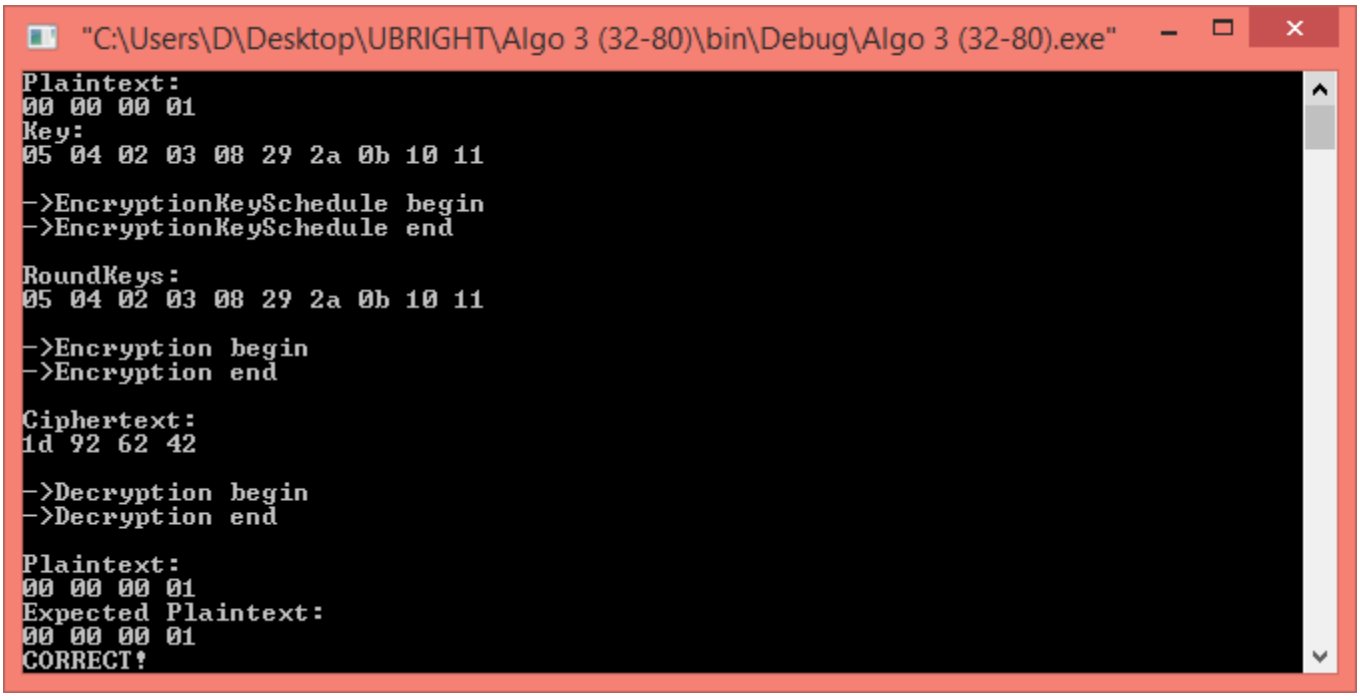

Figure 2. *Encryption and decryption of proposed UBRIGHT cipher for plaintext = 00 00 00 00*

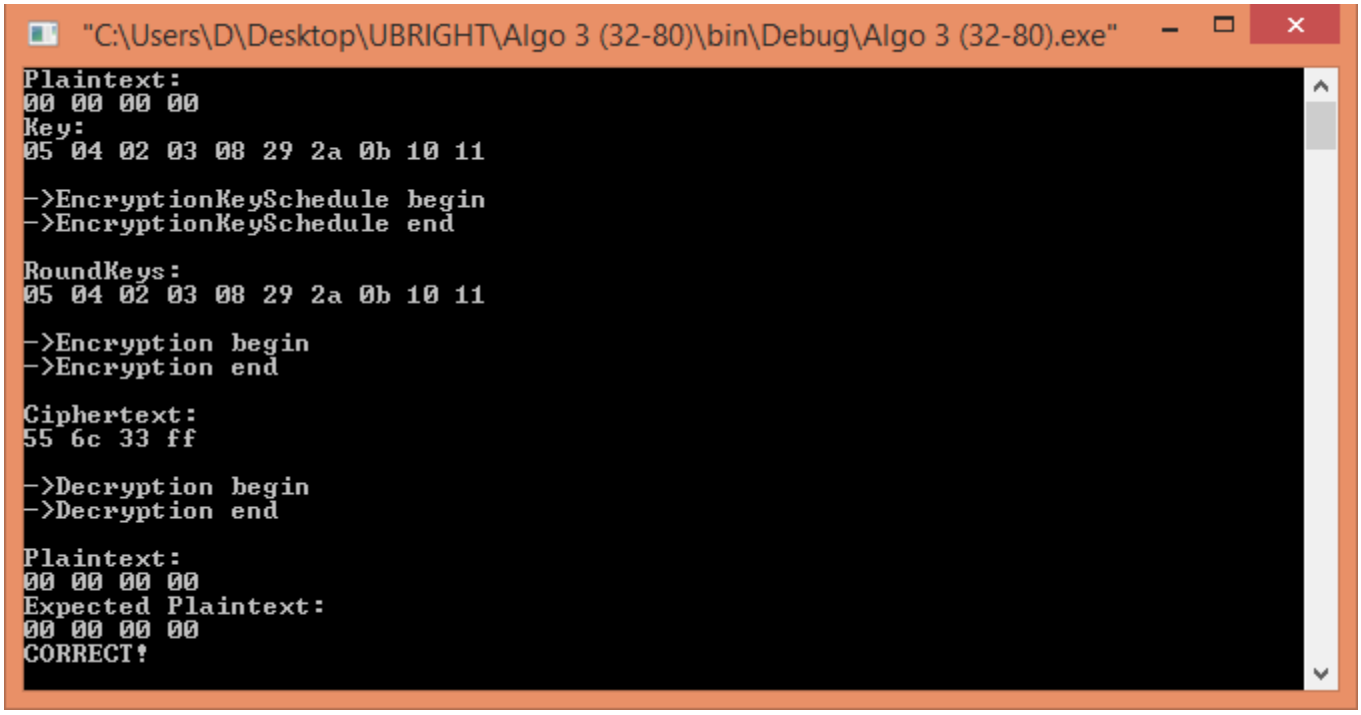

Figure 3. *Encryption and decryption of proposed UBRIGHT cipher for plaintext = 00 00 00 01*

To test the fitness of any cipher two parameters are considered as most important. These are confusion and diffusion. Confusion is to make a ciphertext-plaintext relation as complicated as possible. Whereas, diffusion means that every plaintext bit and key bit must influence every plaintext bit which is important to hide the statistical structure of the plaintext. Memory consumption, speed, and throughput are the main primitives to access the suitability of any cipher and finding the best one from several algorithms [19]. Among all these primitives, memory is the most prevailing. Also, optimized software implementations result in fast speed thus consuming low power [20]. To evaluate the performance and security of the UBRIGHT, following parameters are taken:

4.1 Strict Avalanche Criteria (SAC):

If any single bit change in plaintext or key (input) bring about 50% changes in the ciphertext, then it is considered as it satisfies Strict Avalanche Criteria. Besides, if it is proved that a cipher fulfills SAC, then that cipher has a higher probability to prevent most of the attacks. Otherwise, it is considered that poor randomization occurs. Table 1 summarizes the results of diffusion for proposed cipher when only one bit of the plaintext has changed. The last row of Table 1 gives the number of bits changed whenever a single bit is altered in the plaintext. The average diffusion rate, percentage of diffusion and diffusion range of UBRIGHT is also given in table 1. These values prove that in UBRIGHT significant amount of diffusion is achieved. Besides, the proposed cipher achieves a different amount of diffusion which shows the strength of cipher's properties. It is proved that the proposed cipher fulfill SAC criteria as it achieves around 50% diffusion.

Table 1 *summarizes SAC and randomness test results for UBRIGHT, when one bit is changed in the plaintext (Key (Hexadecimal) = 05 04 02 03 08 29 2a 0b 10 11).*

Plaintext	Ciphertext (Binary)					
					Zero's	One's
00 00 00 00	01010101	01101100	00110011	11111111	12	20
00 00 00 01	00011101	10010010	01100010	01000010	20	12
00 00 00 02	10010110	01010000	10010100	01110100	19	13
00 00 00 03	10101111	10001010	00000011	01011100	17	15
00 00 00 04	10011110	11111001	10010001	00101110	14	18
00 00 00 05	11110111	11000000	01110001	10111111	12	20
Number of bits changed	(0, 1)=18 (0, 2)=17 (0, 4)=16 (1, 3)=13 (2, 3)=16 (4,5)=14 Average diffusion=15.67 (48.96%) Diffusion Range= 13-18				15.66	16.33

Plaintext is in hexadecimal and ciphertext is in binary form.

4.2 Key Sensitivity

To test the key sensitivity of a cipher it is checked whether data is retrieved from the key or not when the key has a minute difference from that of the original key. To check the key sensitivity, SAC is used. The results of key sensitivity test for UBRIGHT are summarized in table 2 which gives the diffusion, average diffusion, diffusion percentage and diffusion range for the test data whenever there is a single bit change in the key. The results show that the amount of diffusion is not the same for all test cases, there is a variation in diffusion amount and good diffusion range is achieved. The minimum amount of diffusion for the test data is 12 and the maximum amount of diffusion is 20. It shows the strength of cipher's properties. Good diffusion range shows that it is not easy for an attacker to guess the number of bits altered whenever a single bit is altered. The results of the key sensitivity test for UBRIGHT cipher shows that nearly 50% of the bits are changed whenever a single bit is

***Table 2**. Summarizes key sensitivity and randomness test results for UBRIGHT cipher, whenever one single-bit is changed in the key (Plaintext (Hexadecimal) = 00 00 00 00).*

S. No	Key	Ciphertext				Number of	
						Zero's	One's
1	00 04 02 03 08 29 2a 0b 10 11	1010 0110	1110 1100	0010 0101	1101 0011	15	17
2	01 04 02 03 08 29 2a 0b 10 11	1010 0100	1101 0010	1011 0110	0101 1011	15	17
3	02 04 02 03 08 29 2a 0b 10 11	0101 1101	1111 1001	0010 0111	1111 1101	10	22
4	03 04 02 03 08 29 2a 0b 10 11	0110 1100	1101 0101	1000 1001	1001 1000	17	15
5	04 04 02 03 08 29 2a 0b 10 11	0110 1001	0101 0110	1011 1100	1010 0101	15	17
6	05 04 02 03 08 29 2a 0b 10 11	0101 0101	0110 1100	0011 0011	1111 1111	12	20
Number of bits changed		(1, 2)=12 (1, 5)=20 (3, 4)=15			(1, 3)=15 (2, 4)=16 (5, 6)=17	14	18
Average diffusion = 15.83 (49.47%) Diffusion Range = 12-20							

Key is in hexadecimal and ciphertext is in binary form.

4.3 Randomness Test

It is the diffusion characteristic of any cipher which affects the randomness test. Randomness test is based on the round function of any cipher and its ability to produce random output. No input/output relation is found in a cipher if it passes the randomness test. There are two steps in this test; first, the sample sequence is extracted from the algorithm and then, statistical randomness test is performed on the obtained sample. For any Ψ matrices, the following criteria should meet:

1. There should be an equal number of 1's and 0's.
2. 1's and 0's should be randomly distributed in the sample.
3. For any $i \neq j$, Ψ_i & Ψ_j should be dissimilar.

The result of the randomness test for the proposed UBRIGHT cipher is presented in table 2 and table 4. It gives an average number of 0's and 1's which are almost equal. Besides, 0's and 1's are randomly distributed and for each $i \neq j$, Ψ_i & Ψ_j are dissimilar. So the proposed cipher passes randomness test also.

4.4 Execution Time

Execution time of a cryptographic algorithm is the time it takes in the encoding and decoding of a particular data. In IoT based applications, because of limited resource availability, lower execution time is preferred. So, algorithm having lower execution time is considered better than others. Results of the execution time of UBRIGHT are summarized in table 3. This is given in terms of speed on a 32-bit CPU.

***Table 3:** Memory and execution time of proposed cipher implemented on a 32-bit platform.*

Memory (Bytes)			Cost (Cycles/byte)	Speed (Mbytes/sec)
Encryption	Decryption	Key-Schedule		
479	477	203	40	52.51

The speed can be further increased but it results in increased memory. So, this is a memory-speed tradeoff and can be managed according to the application requirements.

4.5 Memory Utilization

IoT based smart devices are constrained in terms of memory, these devices have a limited amount of available memory. So, ciphers having low memory utilization are preferred over others. Proposed cipher, UBRIGHT consumes a smaller amount of memory and has a high speed which is considered in favor of its deployment in IoT. Table 3 summarizes the memory utilization for encryption, decryption, and key-schedule of UBRIGHT. Further code savings are possible but only at the cost of lower throughput. Code savings are possible by using the concept of loop unrolling.

5 CONCLUSION

IoT is an innovative technology that allows objects/ devices to act as smart things in a smart environment. To relish the welfares of this smart environment, security is considered to be a foremost aspect in the constrained end nodes. For a constrained IoT environment, "Lightweight cryptography" came into existence. In this paper, a newly proposed ultra-LBC, UBRIGHT is presented. The analysis of newly proposed cipher on a 32-bit processor shows that it fulfill SAC, passes key sensitivity and randomness tests. Remarkable results are obtained for execution time and memory utilization of UBRIGHT cipher. Besides, it has a low cost, fast execution speed, and low memory utilization.

REFERENCES

1. D. Dinu, Y. L. Corre, D. Khovratovich, L. Perrin, J. Großschädl, and A. Biryukov, "Triathlon of lightweight block ciphers for the Internet of things", Journal of Cryptographic Engineering, Springer Berlin Heidelberg, pp. 1-20, 2018. https://doi.org/10.1007/s13389-018-0193-x
2. M. B. Shemaili, C. Y. Yeun, K. Mubarak, and M. J. Zemerly, "A New Lightweight Hybrid Cryptographic Algorithm for The Internet of Things", in: Proc. of 7th International Conference for Internet Technology and Secured Transactions (ICITST-2012), pp. 87-92, 2012.
3. D. Sehrawat, and N. S. Gill, "Security Requirements of IoT Applications in Smart Environment", in: Proc. of 2nd International Conference on Trends in Electronics and Informatics (ICOEI 2018), pp. 324-329, 2018. Available: https://doi.org/10.1109/ICOEI.2018.8553681
4. G. Hatzivasilis, K. Fysarakis, L. Papaefstathiou, and C. Manifavas, "A review of lightweight block ciphers", Journal of Cryptographic Engineering, pp.1-44, 2018. Available: https://doi.org/10.1007/s13389-017-0160-y
5. M. Katagi, and S. Moriai, "Lightweight Cryptography for the Internet of Things", Sony Corporation, 2016.
6. D. Sehrawat, and N. S. Gill, "Deployment of IoT based Smart Environment: Key Issues and Challenges", International Journal of Engineering & Technology. vol. 7,

no. 2, pp. 544-550, 2018. Available: http://dx.doi.org/10.14419/ijet.v7i2.9504
7. D. Sehrawat, N. S. Gill., and M. Devi, "Comparative Analysis of Lightweight Block Ciphers in IoT-Enabled Smart Environment", in: Proc. 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), IEEE, pp. 915-920, 2019. Available: https://doi.org/10.1109/SPIN.2019.8711697
8. A. Biryukov, and L. Perrin, "State of the art in lightweight symmetric cryptography", IACR Cryptology ePrint Archive, 2017.
9. A. Biryukov et al., "FELICS—Fair Evaluation of Lightweight Cryptographic Systems", NIST Workshop on Lightweight Cryptography, 2015. https://www.cryptolux.org/index.php/FELICS
10. D. Hong, J. K. Lee, D. C. Kim, D. Kwon, K. H. Ryu, and D. G. Lee, "LEA: A 128-bit block cipher for fast encryption on common processors". In: Proc. of International Workshop on Information Security Applications, Springer, Cham, pp. 3-27, 2013. Available: https://doi.org/10.1007/978-3-319-05149-9_1
11. H. Seo, Z. Liu, J. Choi, T. Park and H. Kim, "Compact implementations of LEA block cipher for low-end microprocessors", In: Proc. of International Workshop on Information Security Applications, Springer, Cham, pp. 28-40, 2015. Available: https://doi.org/10.1007/978-3-319-31875-2_3
12. R. Beaulieu, D. Shors, J. Smith, S. T. Clark, B. Weeks and L. Wingers, "The SIMON and SPECK Families of Lightweight Block Ciphers", Cryptol ogy ePrint Archive, Report 2013/404. Available: https://doi.org/10.1007/978-3-319-31875-2_3
13. D. Sehrawat, and N. S. Gill, "BRIGHT - Proposed Family of Lightweight Block Ciphers for IoT-Enabled Smart Environment", International Journal of Innovative Technology and Exploring Engineering (IJITEE). vol. 8, no. 9, pp. 584-592, 2019.
14. D. Sehrawat, and N. S. Gill, "BRIGHT: A Small and Fast Lightweight Block Cipher for 32-bit Processor", International Journal of Engineering and Advanced Technology. vol. 8, no. 5, pp. 1549-556, 2019. Available: https://www.ijeat.org/wp-content/uploads/papers/v8i5/E7302068519.pdf
15. D. Sehrawat, and N. S. Gill, "Performance Evaluation of Newly Proposed Lightweight Cipher, BRIGHT", International Journal of Intelligent Engineering & Systems. vol. 12, no. 4, pp. 71-80, 2019. Available: http://www.inass.org/2019/2019083108.pdf
16. N. Mouha, B. Mennink, A. V. Herrewege, D. Watanabe, B. Preneel, and I. Verbauwhede, "Chaskey: an efficient MAC algorithm for 32-bit microcontrollers", in: Proc. of International Workshop on Selected Areas in Cryptography, Springer, pp. 306-323, 2014. Available: https://doi.org/10.1007/978-3-319-13051-4_19
17. A. Baysal, and S. Şahin, "Roadrunner: A small and fast bitslice block cipher for low cost 8-bit processors", in: Proc. of International Workshop on Lightweight Cryptography for Security and Privacy, Springer, Cham, pp. 58-76, 2015. Available: https://doi.org/10.1007/978-3-319-29078-2_4
18. D. Sehrawat, and N. S. Gill, "Analysis of Security Attacks on Lightweight Block Ciphers and their Countermeasures", Journal of Engineering and Applied Sciences, vol. 13, no. 20, 2018. Available: 10.3923/jeasci.2018.8439.8447
19. D. Sehrawat, and N. S. Gill, "A Review on Performance Evaluation Criteria and Tools for Lightweight Block Ciphers", International Journal of Advanced Trends in Computer Science and Engineering, vol. 8, no. 3, pp. 630-639, 2019. Available: 10.30534/ijatcse/2019/47832019
20. D. Sehrawat, and N. S. Gill, "Design Considerations of Lightweight Block Ciphers for Low-Cost Embedded Devices", International Journal of Recent Technology and Engineering (IJRTE),vol. 8, no. 2, 2019.