Bright - Proposed Family of Lightweight Block Ciphers for IoT-Enabled Smart Environment

Deepti Sehrawat, Nasib Singh Gill

Abstract: ***Lightweight security algorithms are tailored for resource-constrained environment. To improve the efficiency of an algorithm, usually, a tradeoff is involved in lightweight cryptography in terms of its memory requirements and speed. This paper proposes a software-oriented new family of lightweight block ciphers, BRIGHT. Proposed family of ciphers support a range of block and key sizes for constraint environment. BRIGHT family has 6 variants and all variants fulfill Strict Avalanche Criteria and key sensitivity test. It is believed that BRIGHT family of ciphers provides better security and performance in IoT-enabled smart environment. Our aim, while designing BRIGHT is to enhance the cipher for IoT applications. For this, we have used the concept of key whitening that helps to resist against attacks like MITM and brute-force. Round permutation in BRIGHT results in stronger and faster diffusion and provides resistance against linear, differential, impossible differential, related-key rectangle, biclique, MITM, and statistical saturation attack which is likely to be applied to GFN based ciphers. BRIGHT using round constant thwarts attacks like rotational cryptanalysis, self-similarity, invariant attack, related-key attacks, and weak key attacks.***

Index Terms: ARX, BRIGHT, Cryptographic solutions, Feistel block ciphers, GFN, Lightweight block cipher.

I. INTRODUCTION

A huge number of emerging application areas are utilizing constrained devices in a heterogeneous sensor network. One such area is the Internet of Things (IoT) which enables various resource constraints devices to communicate in the network. Various wireless technologies are playing a vital role in our lives, tagging technologies like NFC, RFID, and 2D barcode are used to identify physical objects over the internet [1]. IoT is a very wide term unfolding the fact that, in the near future, most of the devices will be connected to one another instead of connecting persons together.

Now-a-days the

IoT demands high security and data privacy in sensor network where a large number of sensor nodes are communicating their private data [2]. Rarely a few of them make use of powerful processors and utilizes the same crypto algorithms as usual desktop PCs. However, most of them use micro-controllers which have low computational ability, low power, and low resource utilization. So cryptographic solutions need to be readdressed and redesigned. In this direction to address security issues, a relatively new field of lightweight cryptography came into existence. Lightweight security algorithms are tailored for a resource-constrained environment and include RFID (Radio Frequency Identification Devices) tags, sensor nodes, contactless smart cards, smart health devices and many more [3]. Tremendous work has already been done in the area of lightweight cryptography. Software implementations of lightweight cryptography consider low resource processors, which is significant because encryption in embedded systems can be carried out in H/W but decryption is done in S/W. Software oriented cipher designs provide more flexibility at lower costs on manufacturing and maintenance as compared to hardware implementations [4]. Even providing strong resistance against mathematical attacks could not protect hardware oriented block ciphers from side channel attacks thereby losing its keys. So a good software design is required to provide enough security guard against attacks.

In this paper, a new family of Lightweight Block Ciphers (LBC), named BRIGHT is proposed. BRIGHT family of ciphers is software oriented and the structure of the proposed design is so designed that it leads to fast diffusion. The performance of the BRIGHT family ciphers is evaluated by performing Strict Avalanche Criteria (SAC) and Key Sensitivity Test. All versions of the proposed cipher fulfill SAC and key sensitive test. Besides, the operations in the BRIGHT family of ciphers is so arranged that it shows good resistance against various attacks.

The rest of the paper is structured as follows: Section 2 presents related work. A new family of lightweight block ciphers is proposed in section 3 after deep analysis of various existing LBCs and attacks on existing ciphers which are culminated in the publication in [5, 6]. In section 4 performance of proposed BRIGHT cipher is evaluated. Section 5 presents security analysis of the proposed design and finally, section 6 concludes the paper.

II. RELATED WORK

Lightweight block ciphers have comparably smaller internal states in the form of key size and block size. Feistel round functions are simple and require more number of rounds. ARX (modular Addition-Rotation-bitwise XOR) designs have even more efficient software implementations.

Some lightweight block ciphers with ARX structure are SIMON [2], SPECK [2], TEA [7], LEA [8], HIGHT [9], and Chaskey [10]. TEA uses a simple round function and key schedule. It has lower block length and a larger number of rounds [7]. LEA has three variants with block size of 128-bits and key sizes 128, 192 and 256-bits long. It is based on 4-branch Generalized Feistel

Revised Manuscript Received on July 05, 2019.

Deepti Sehrawat, Department of Computer Science and Applications, Maharshi Dayanand University, Rohtak, Haryana, India

Nasib Singh Gill, Department of Computer Science and Applications, Maharshi Dayanand University, Rohtak, Haryana, India.

Network (GFN) and is optimized for 32-bit processor than a 64-bit processor [8]. HIGHT is optimized for 8-bit operations. SPECK and SIMON family of lightweight ciphers have 10 instances of each. SIMON cipher with a block size of 128-bit is comparable with LEA, LEA's performance exceeds that of SIMON in both 32-bit and 64-bit processors. Whereas, the performance of SPECK exceeds LEA'S performance in 64-bit processor because of SPECK's 64-bit addition in a 64-bit processor. CHASKEY ciphers is a permutation-based Message Authentication Code (MAC) LBC. It does not follow any key schedule it simply consists of shift and XOR operations with the state for two sub-keys [10].

III. BRIGHT CIPHER

A. Design Considerations

The proposed design is a family of LBCs, named BRIGHT. BRIGHT is a 4-branch Generalized Feistel Network GFN) that has comparably light round functions. It has low decryption overhead and provable security margin. Smaller block sizes result in low memory footprints when implemented in software and key sizes less than 80 bits provides low-security margin against brute-force search attacks. So, considering this in mind, the minimum block size chosen for the BRIGHT family of block ciphers is 64-bits and minimum key size chosen is 80-bits. Here the term lightweight does not only mean that a security algorithm is suitable for some constraint platform but it should be platform independent. Proposed family of BRIGHT cipher has an application independent design choice that provides good performance on multiple platforms. Since devices and the applications vary greatly, proposed cipher provides a wide range of options in the form of block sizes and key sizes. Prevailing block sizes are 64-bit and 128-bit and key sizes are related to the desired security level, for instance, for low-cost devices 64-bits key would be sufficient while on the other hand, more sensitive applications may require 256 bits of key. So, we have designed BRIGHT with two block sizes 64-bits and 128-bit supporting several key sizes viz. 80, 96, 128, 192 and 256 bits with three key sizes in total to go along with each of the block sizes. To provide enough security margin against existing attacks, we first attempt to find out the minimum number of rounds '*R*' required to perform complete diffusion for each variant of BRIGHT family. Starting from the very first round, proposed cipher fulfills SAC i.e. in every round of BRIGHT family ciphers around 50% of bits are changed. All variants of BRIGHT family ciphers take at most 8 rounds for complete diffusion. The actual number of rounds for each variant is found by applying the formula *4R* for BRIGHT 64/80 variant and the subsequent variants are incremented by 1 round to its previous variant like BRIGHT 64/96 has *4R+1* rounds. This provides resistance to the ciphers against attacks.

The block size, key size, and the number of rounds for all variants of the BRIGHT family of lightweight ciphers are given in Table 1.

Table 1. Parameters for all versions of BRIGHT

Block Size	**Key size** ***mn***	**Word Size** ***n***	**Key Words** ***m***	**Rounds**	**Rotation Amount** ***a***	***b***
4n						
64	80	16	5	32	2	6
	96		6	33	2	6
	128		8	34	2	6
128	128	32	4	35	5	8
	192		6	36	5	8
	256		8	37	5	8

Talking about key scheduling, the BRIGHT family of ciphers provides a good level of security against related key attacks.

B. Notation

Throughout the paper the following notations are used:

$\mathbf{V_i}$	Word/ Branch
$\boxplus$	Addition modulo 2^n
$\oplus$	*n*-bit exclusive OR
x<<<m	Left circular shifts by m-bits
x>>>m	Right circular shifts by m-bits
$\mathbf{Rk_i}$	Round key for i^{th} round
&	Bitwise AND
C	Constant

C. Proposed Algorithm Design

Most of the algorithms designed so far were aimed and do well on a particular platform and were not intended for high performance across a wide range of devices, except a few like SIMON and SPECK. In this direction, to fill the need for secure, flexible and platform independent LBC a family of block ciphers, named BRIGHT is proposed in this paper. A total of six instances of BRIGHT have been proposed for better performance on microcontrollers. The notion for different variants of BRIGHT specifies its block size and key length, like BRIGHT 64/80 refers to the BRIGHT lightweight block cipher having a block size of 64 bits and a key size of 80 bits. The encryption function takes a plaintext, $P=P_0P_1P_2P_3$, along with 4 whitening keys (wk_i) where, $0<=i<4$ and *n* round keys for *n*-rounds denoted as RK_i, (Round key for i^{th} round) extracted from the master-key, *Mk.*

D. General Structure

BRIGHT cipher employs a 4-branch GFN by applying various operations on it. For $k \in GF(4)^n$, the round function of BRIGHT 4*n* is a key-dependent function and this function is given by a map by using (1).

$$R_k: GF(4)^n \times GF(4)^n \rightarrow GF(4)^n \quad (1)$$

This map is defined by different operations applied on it like key whitening, ARX (modular addition, XOR and circular shift) operations, and round permutation. The effect and operations on single round function R_{ki} are shown in Fig. 1.

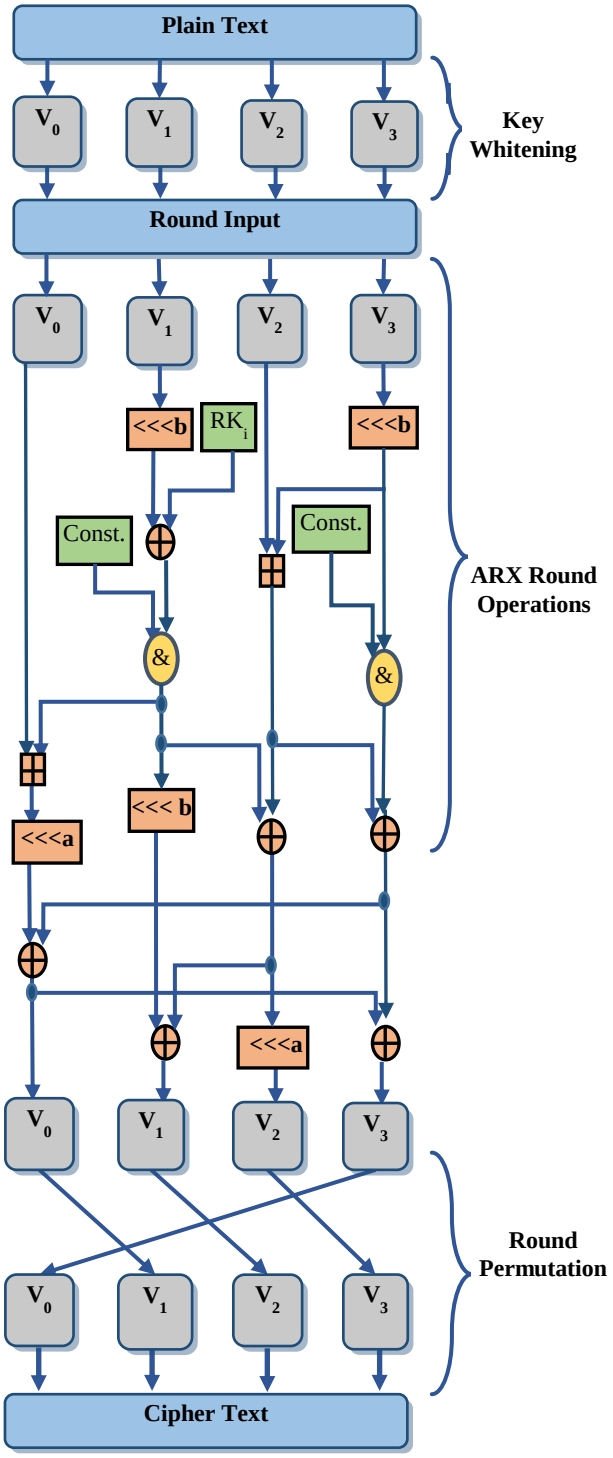

Figure.1 Structure of BRIGHT cipher

Decryption does not add any complexity to the cipher, in it the order of operations are just reversed. There are 3 layers in the BRIGHT family of ciphers which are explained below:

Layer 1: Key Whitening

Key whitening is a concept in which actual key is XORed prior to its use which thwarts the weak key attacks, as in SIT [11]. Key whitening does not thwart most analytical attacks, for example, linear and differential cryptanalysis but it provides resistance against MITM attack and also increases the complexity of a brute force attack. The BRIGHT family of ciphers implements key whitening by adding a sub-key prior to the first round. It is supposed that an attacker cannot predict input after whitening. In most of the attacks, due to key whitening, it is not possible to extend the attack by even one round as this requires searching for all of the *n*-keys (256-bit keys in 256-bit version). So key whitening plays an important role in providing resistance against some attacks. In the proposed BRIGHT family of ciphers, initial round key whitening is applied i.e. the key is XORed prior to its use in the round function.

Layer 2: ARX Operations

ARX stands for Addition, Rotation, and XOR operations. ARX designs have very efficient software implementations and efficiently implements the encryption, decryption and key schedule procedures in a parallel way. The BRIGHT family uses ARX structures in which the operations are so arranged that it leads to fast diffusion and provides strong resistance against attacks. Following operations are performed on each round of n-bit words of BRIGHT:

1) Addition modulo 2^n is used to introduce sufficient non-linearity and as a result, improves the diffusion and helps to thwart structural attacks. This is done at the same speed as by bitwise XOR. By using a non-linear operation, addition modulo 2^n, the differences are propagated indefinitely.
2) Left and right circular shifts are used to add diffusion in the cipher [3]. By using a different amount of rotation, BRIGHT offers different trade-offs between security (linear and differential probabilities) and efficiency.
3) Bitwise XOR operation is used in combination with addition modulo to improve the diffusion.

Layer 3: Round Permutation

Usually, GFN has slow diffusion property and round permutation provides faster diffusion. Fast diffusion in a cipher thwarts those attacks that are made possible because of slow diffusion property of block ciphers. Same is the case with PICCOLO [12] cipher and as a result, round permutation provides resistance against impossible differential cryptanalysis in PICCOLO. Proposed cipher, BRIGHT implements round permutation as a last step in each round which results in fast diffusion and as a result complete diffusion can be achieved in a lower number of rounds in all versions of the BRIGHT cipher.

Encryption can be defined by the composition as in (2):

$$R_{kr-1} \circ R_{kr-2} \circ \dots R_{k1} \circ R_{k0} \qquad (2)$$

This composition of round functions can be read from right to left. Same round function is used in decryption and the order of key whitening, round constants, round keys, and ARX operations are reversed. Addition modulo 2^n is replaced by subtraction modulo 2^n. BRIGHT operations are fast and are supported in 8/16/32/64-bit platforms in an efficient and parallel way which lead to fast encryption and small code size. We have removed the last round key whitening which was used by most of the existing ciphers to enhance the performance and it also lowers the code size.

E. **Key Schedule**

Key scheduling, being a crucial component of a block

cipher can break down entire design if otherwise not designed cleverly. Weak key scheduling gives classes of weak keys i.e. keys that break the cipher easily like, zero correlation cryptanalysis, MITM attacks and their variants [13]. There are different ways of using a key schedule in a block cipher, the first approach uses linear key scheduling which performs bit-permutations and other linear operations as in case of DES [14]. Some others also used masking with some fixed round constants along with the first approach like SIMON [2], and Piccolo [12]. The last and important approach uses a non-linear component along with the combination of the first two approaches as is done in SPECK [2] cipher. An iterative block cipher makes use of mixing of round keys which can be used with the plaintext to introduce non-linearity.

Key scheduling of BRIGHT cipher is motivated from the SPECK [2] key scheduling which is compact in memory and is secure. In the key scheduling part, a user-defined master key is stored in the register key, which is represented by using (3):

$$Key = k^n k^{n-1} k^{n-2} \dots k^2 k^1 k^0 \quad (3)$$

Instead of using the entire key, a sub-key is derived from the master key which is fed to the rounds, each round uses different keys. More rounds introduce more confusion and diffusion and hence as a result provides more security. To provide resistance against attacks, attacks must be slower than exhaustive key search i.e. brute force attack. More the key size, longer it takes for exhaustive key search attack resulting in a more secure cipher. Furthermore, larger key sizes require more rounds to dissipate the effect of every key bit on the ciphertext in a similar way. So, with the increase in the key size, there must be an increase in the number of rounds so that measurable differences could not be found and this, in turn, prevent cipher from attacks.

Key schedule of the BRIGHT family of ciphers does the minimal mixing so as to thwart related-key attacks. A sequence of *n* key-words is retrieved from the master key for *n*-rounds (one for each round) which are used in n-rounds to make encryption/decryption process strong enough.

Key scheduling part uses the round function to generate round keys k_i. Let '*K'* be a key for BRIGHT *4n* block cipher. We can write K = $(l_{m-2}, \dots l_0, k_0)$ where $l_i, k_0 \in GF(4)^n$, for a value of *m* in {2,3,4}. Sequences k_i and l_i are defined by (4) and (5) respectively.

$$L_{i+m-1} = (k_i + l_i \ggg 2)\text{^}i \quad (4)$$

$$K_{i+1} = k_i \lll 5\text{^}l_{i+m-1} \quad (5)$$

The value k_i is the i^{th} round key, for $0 \leq i < T$.

IV. Performance Evaluation

Confusion and diffusion are two main parameters which test the suitability of a cipher. Confusion means that there must exist a complicated relation of ciphertext with plaintext and key. This is achieved by mixing operations in a complicated way. Diffusion means that every ciphertext bit must be influenced by every plaintext bit and the key bit. Spreading every plaintext bit influence over many ciphertext bits hides the statistical structure of the plaintext.

Key sensitivity is another term which can be used for the performance evaluation of ciphers. An algorithm is said to be key sensitive if retrieving original data is not possible when a single bit is changed in the original key. For this, Avalanche test is used to estimate the extent of changes in the resulting ciphertext. As per Strict Avalanche Criteria (SAC), a test is considered to be perfect if a single bit change in input (key/ plaintext) results in a 50% change in the bits. Cipher fulfilling SAC has a higher probability to thwart all possible attacks. Contrary, if SAC is not satisfied, it is considered that poor randomization occurs and cipher is not considered good.

The results of the Avalanche test for BRIGHT 64/80 is given in Table 2 by finding the number of bits changed in ciphertext when one bit is changed in the plaintext. The values of plaintext and ciphertext are given in hexadecimal. To find the effect of SAC, first, these hexadecimal values are converted to binary values than the amount of change in bits is calculated. The chosen master key for which SAC is performed is also given. The result shows that out of 64, 34 bits are changed (around 53%). This result is obtained from the results of Fig. 2(a) and Fig. 2(b). The results of the Avalanche test shows that BRIGHT 64/80 fulfills SAC.

Table 3 gives the result for key sensitive test for BRIGHT 128/128, by finding the number of bits changed in the ciphertext when there is a single bit change in plaintext. Here also the values of chosen master key, plaintext and obtained ciphertext are given in hexadecimal. These values are first converted to binary equivalents to obtain the number of bits got changed in ciphertext when only one bit is changed in the plaintext. Out of 128 bits, 64 bits are changed (50%). This result is obtained from the results of Fig. 3(a) and Fig. 3(b).

Similar type of results are obtained for all variants of the proposed family of ciphers. Table 4 summarizes the result of SAC for all versions of BRIGHT family ciphers and Table 5 gives the result of key sensitive test for all versions of BRIGHT. To obtain the results of table 4 and table 5, a total of 9 diffusion values are taken to find the average diffusion and diffusion percentage. It is verified that all versions of BRIGHT family fulfill SAC criteria and key sensitive test. Besides, performance of BRIGHT is evaluated on different parameters and compared with various benchmarked lightweight block ciphers on different platforms (64-bit and 32-bit) which are culminated in the publications [15, 16].

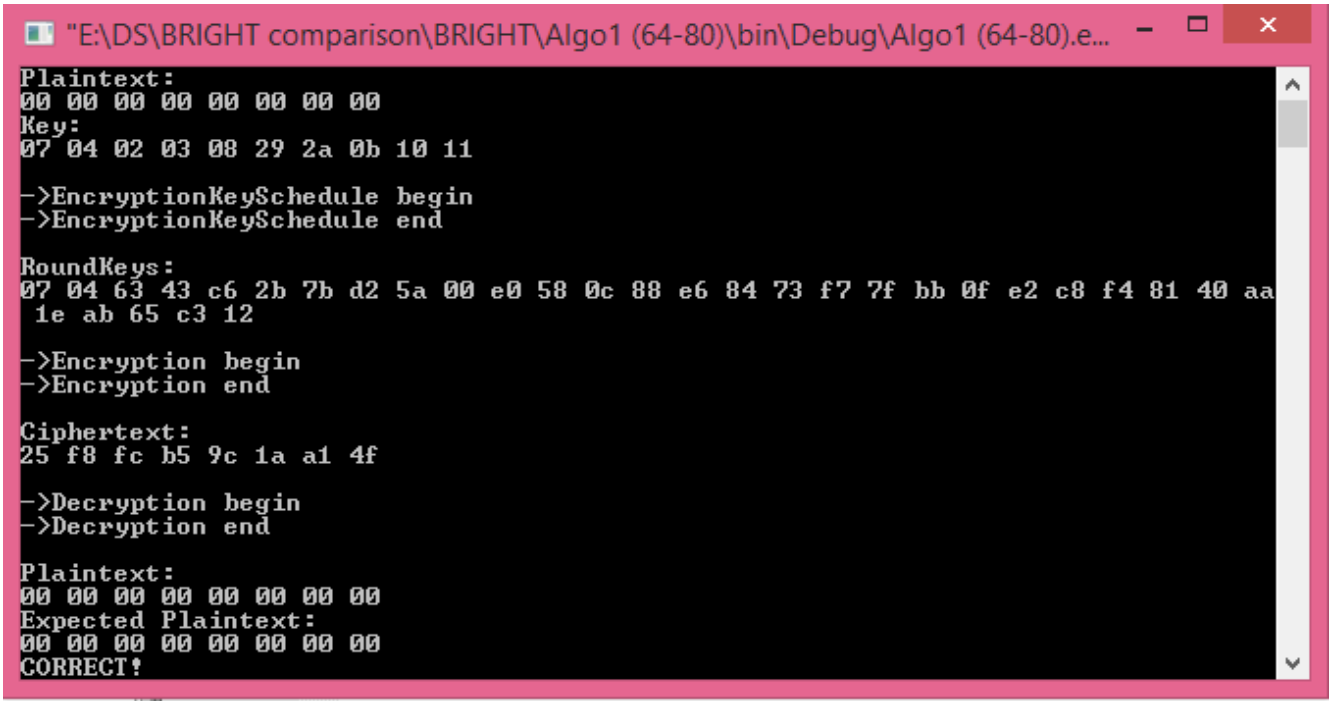

Figure 2 (a): Represent encryption and decryption of BRIGHT 64/80 (Plaintext = 00 00 00 00 00 00 00 00).

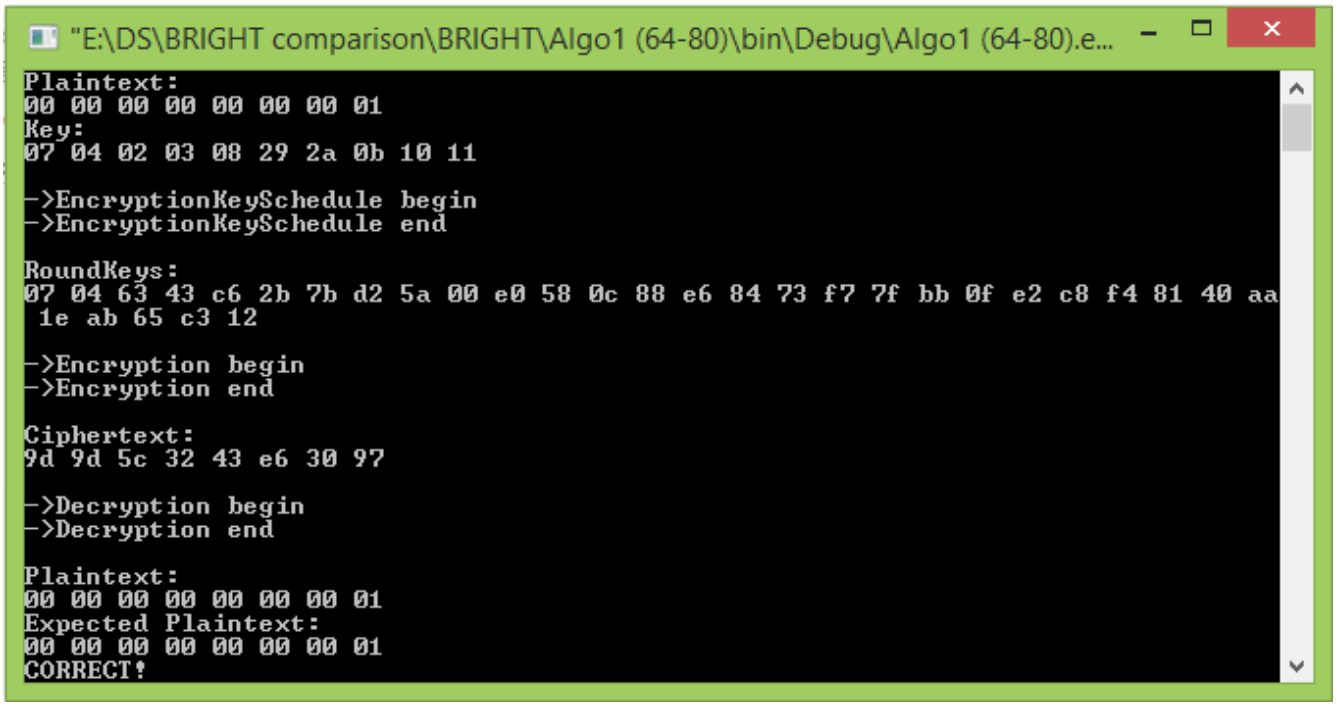

Figure 2 (b): Represent encryption and decryption of BRIGHT 64/80 (Plaintext = 00 00 00 00 00 00 00 01).

Table 2: Diffusion for BRIGHT 64/80, when there is a single-bit change in plaintext.

BRIGHT 64/80, Key (Hexadecimal)= 07 04 02 03 08 29 2a 0b 10 11		
Plain Text (Hexadecimal)	**Cipher Text (Hexadecimal)**	**Number of bits changed**
00 00 00 00 00 00 00 00	25 f8 fc b5 9c 1a a1 4f	34
00 00 00 00 00 00 00 01	9d 9d 5c 32 43 e6 30 97	

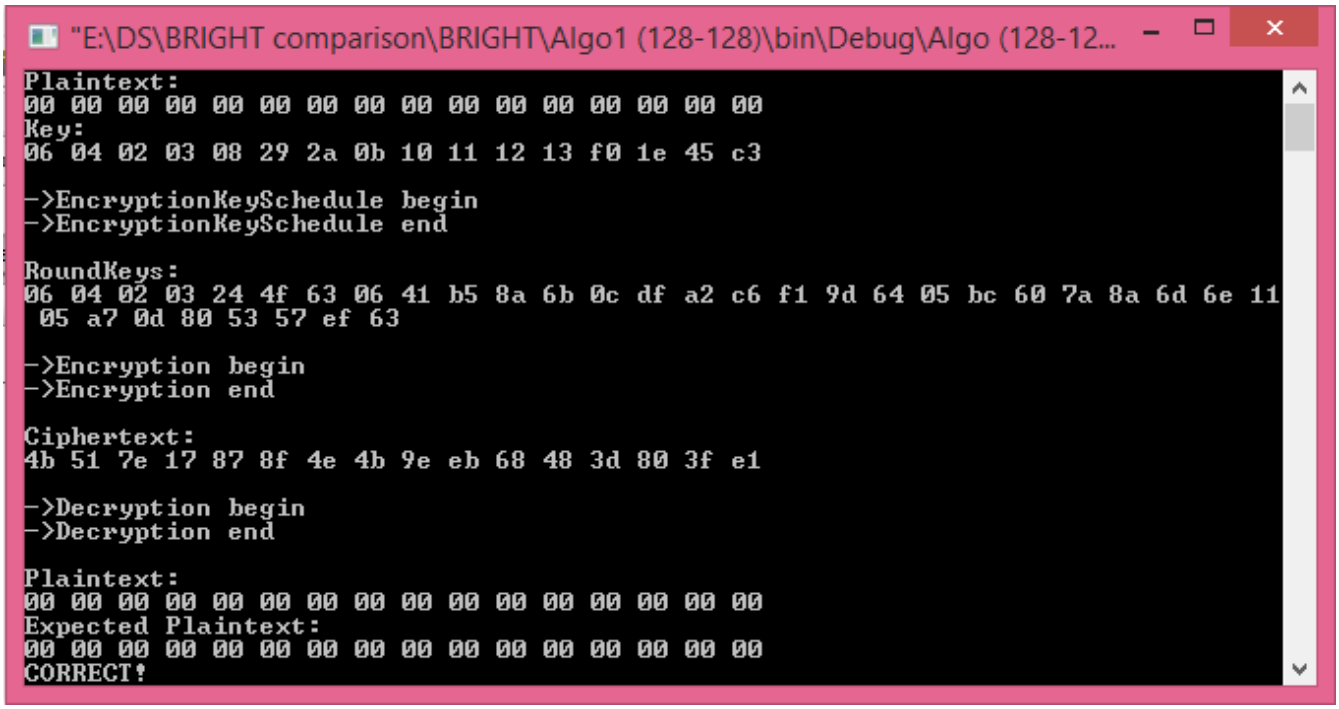

Figure 3 (b): Represent encryption and decryption of BRIGHT 128/128 (Key = 06 04 02 03 08 29 2a 0b 10 11 12 13 f0 1e 45 c3).

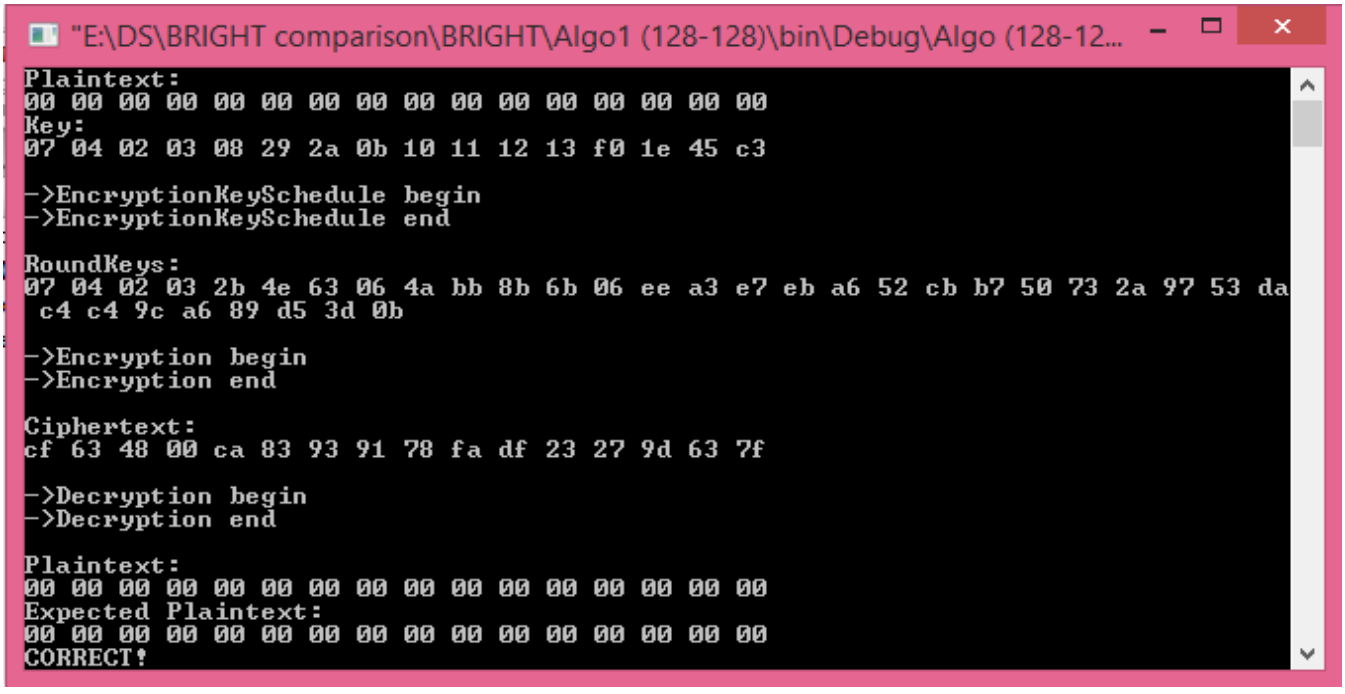

Figure 3 (b): Represent encryption and decryption of BRIGHT 128/128, (Key = 07 04 02 03 08 29 2a 0b 10 11 12 13 f0 1e 45 c3).

Table 3: Diffusion for BRIGHT 128/128, when there is a single-bit change in key.

BRIGHT 128/128, Plaintext (Hexadecimal)= 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		
Key (Hexadecimal)	**Cipher Text (Hexadecimal)**	**Number of bits changed**
06 04 02 03 08 29 2a 0b 10 11 12 13 f0 1e 45 c3	4b 51 7e 17 87 8f 4e 4b 9e eb 68 48 3d 80 3f e1	64
07 04 02 03 08 29 2a 0b 10 11 12 13 f0 1e 45 c3	cf 63 48 00 ca 83 93 91 78 fa df 23 27 9d 63 7f	

Table 4: Summarizes the average diffusion when there is a single-bit change in the plaintext.

Block Length	Average diffusion	Diffusion Percentage
BRIGHT 64/80	32.67 bits	51.04 %
BRIGHT 64/96	31.89 bits	49.82 %
BRIGHT 64/128	33.66 bits	52.60 %
BRIGHT 128/128	62.78 bits	49.05 %
BRIGHT 128/192	65.33 bits	51.04 %
BRIGHT 128/256	63.56 bits	49.65 %

Tale 5: Summarizes the average diffusion when there is a single-bit change in key.

BRIGHT Variants	Average diffusion	Diffusion Percentage
BRIGHT 64/80	31.33 bits	48.96
BRIGHT 64/96	31.89 bits	49.82
BRIGHT 64/128	33 bits	51.56
BRIGHT 128/128	64.56 bits	50.43
BRIGHT 128/192	65.11 bits	50.87
BRIGHT 128/256	63.22 bits	49.39

V. SECURITY ANALYSIS

A cipher is considered secure if a cipher proves to be secure against all known cryptanalytic attacks until it is realized otherwise. A number of cryptanalytic attacks can be applied to a cipher. In this paper, our main goal for the security of BRIGHT is to prevent possible attacks and to provide sufficient security margin against unknown attacks. Design of proposed BRIGHT family of ciphers is such that it prevents the effect of most of the attacks. BRIGHT claims to provide security against the following attacks:

A. Related Key Attack

This attack performs cipher operations with partially known keys or complete unknown keys. This attack is made possible when there is either slow diffusion in the cipher or there exists some kind of symmetry in the key expansion block. The internal structure of the proposed cipher is so designed that it performs fast diffusion. Every single round of BRIGHT family ciphers fulfill SAC i.e. in each round at least 50% of the bits got changed. Besides, the key expansion of BRIGHT is aimed for fast and non-linear diffusion. Furthermore, random and uniform key extraction thwarts related key attacks.

B. Differential Cryptanalysis

It is also known as a chosen-plaintext attack which is used mainly in block ciphers to analyze how input differences lead to output differences. So far, no rigorous security proof for ARX ciphers exists, the only way of evaluation of a cipher against this attack is searching for an optimal differential. In this attack, either a distinguisher is build or a key is recovered if there exists certain input/output difference in a non-random way. ARX based block ciphers use addition modulo 2n which is a non-linear operation and it propagates differences indefinitely. BRIGHT cipher utilizing the same ARX structure provides enough resistance against differential attack. Furthermore, round permutation used in BRIGHT cipher results in stronger and faster diffusion and this results in a stronger internal structure of the proposed family of BRIGHT ciphers.

C. Brute-force Attack

Key whitening is a concept in which a key is added prior to its application to the rounds and it is assumed that it is not possible for the attacker to predict the input after whitening and output before whitening. BRIGHT cipher by utilizing key whitening increases complexity against brute force attack. Mostly key whitening is applied by adding a sub-key both, prior to the first round and after the last round. BRIGHT cipher applies key whitening internally by adding key-whitening only prior to the first round. This not only provides resistance against attacks but also results in low memory utilization. Also, using larger size keys provides enough protection to the cipher, especially key sizes greater than 80-bits thwarts this attack. All instances of BRIGHT cipher utilizes key sizes greater than 80-bits. This adds on to the protection of BRIGHT ciphers against brute-force attack.

D. Impossible Differential Cryptanalysis

Block ciphers having slow diffusion rate are more susceptible to this attack. PICCOLO [12] uses round permutation which results in faster diffusion and is free from impossible differential cryptanalysis. BRIGHT also uses round permutations at the end of each round and is supposed that it thwarts the impossible differential attack.

E. Weak Key Attacks

This attack is made possible when the actual key value drives the non-linear operations. Protection to this attack is possible by XORing actual key prior to its use, the same was utilized by SIT [11] cipher which has proven security against weak keys. However, the proposed algorithm by performing pre-key whitening thwarts the effect of this attack as the key is first XORed with the states and then fed to the f-function. Furthermore, use of round constants also provides protection to the cipher against weak key attacks. Our proposed cipher BRIGHT uses round constants, this provides double protection to the cipher against this attack.

F. Algebraic Attack

An over-defined system of algebraic equations is derived to apply algebraic equations. As round function of the newly proposed family of BRIGHT ciphers is of degree 4 as a Boolean vector function, it is difficult to convert an equation system in BRIGHT into an over-defined system. So, BRIGHT provides resistance to this type of attack.

G. Interpolation and Higher Order Differential Attack

These attacks target low degree algebraic block ciphers. Since the degree of a round function of BRIGHT is 4, its full round degree as vector Boolean function is even higher. Furthermore, saturation attack is even more effective than interpolation and higher order differential attack. Hence the results of saturation attack on BRIGHT cipher is more than that of higher order differential attack.

H. Boomerang Attack

Boomerang attack is also known as a differential-differential attack and is intended mainly for the ciphers using asymmetric round functions. A quartet structure is created in this attack at an intermediate value (halfway through the cipher). A block cipher E is given by cascade of E0 and E1, given by (6):

$$E = E0 \circ E1 \tag{6}$$

To weaken the effect of this attack, BRIGHT utilizes good differentials throughout the entire cipher. Maximum 8 rounds are required by the BRIGHT cipher to have complete diffusion.

I. Slide Attack

If an iterative process in round function exhibits self-similarity to some degree then Slide attacks are applicable. It is independent of the number of rounds and round function properties. Slide attacks can exploit the weaknesses of the key scheduling part and even general structural properties of a cipher. It all depends upon the design of a cipher that to what extent this attack exploits the cipher. Iterative block ciphers in which there is a repeating sub-key for all rounds or having a periodic key schedule are more vulnerable to such attacks. Auto-key ciphers which have a data-dependent choice of round sub-keys are easily affected by slide attacks. To prevent from slide attacks, cipher designers have to avoid self-similarity in the rounds of the cipher. This could be accomplished by adding some round counters or random constants in each round [6]. Not using periodic key scheduling is another countermeasure [6]. Zhang et al. (2015) added different round constants in the key schedule and this provides resistance against slide key, as a result, the proposed cipher RECTANGLE prevents from slide attacks [17]. A similar study in [18] presents a LBC, TWINE, providing resistance against slide attacks by applying different constants in the key schedule in each round. PRESENT cipher uses a round dependent counter which provides resistance to the cipher against slide attacks [19]. Key scheduling of BRIGHT cipher is inspired from SPECK [2] key schedule and there is no self-similarity in the rounds of the cipher. Also, we have used round constants in each round which helps to provide resistance against slide attacks.

VI. CONCLUSION

Design and implementation of a lightweight cipher go simultaneously and this has revealed some significant limits and inherent conditions. For example, to achieve a certain security level some lower bounds are placed on the size of a block and key for different implementation environments. Open problems remain in lightweight cryptographic mechanisms that require keen attention. In this paper, a security framework for IoT enabled applications is proposed. Proposed cipher, BRIGHT is a family of LBCs with 6 instances, supporting block sizes of 64-bit and 128-bit. It enables users to match their security needs with application requirements by supporting a range of cryptographic solutions. A good tradeoff will ensure stable progress on the road to realize and secure the IoT as envisioned. This paper set a base for further research work that helps the researchers in the area of IoT security. Our future work consists of detail performance evaluation and cryptanalysis of the newly proposed family of lightweight ciphers, BRIGHT on different software platforms for possible attacks. Furthermore, we invite researchers for the cryptanalysis of the BRIGHT cipher.

REFERENCES

1. D. Sehrawat, and N. S. Gill. Deployment of IoT based Smart Environment: Key Issues and Challenges. *International Journal of Engineering & Technology.* 7(2), pp. 544-550, 2018. [Online]. Available: http://dx.doi.org/10.14419/ijet.v7i2.9504
2. R. Beaulieu, S. T. Clark, D. Shors, B. Weeks, J. Smith, and L. Wingers. The SIMON and SPECK lightweight block ciphers. In: *Proc. of Design Automation Conference (DAC), 52nd ACM/EDAC/IEEE,* IEEE, 2015, pp. 1-6. [Online]. Available: 10.1145/2744769.2747946
3. A. Biryukov, and L. P. Perrin. (2017, June). State of the art in lightweight symmetric cryptography. *IACR Cryptology ePrint Archive.* [Online]. Available: http://eprint.iacr.org/2017/511
4. D. Sehrawat, and N. S. Gill. Security Requirements of IoT Applications in Smart Environment. In: *Proc. 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI),* IEEE, 2018, pp. 324-329. [Online]. Available:

https://doi.org/10.1109/ICOEI.2018.8553681
5. 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, 2019, pp. 915-920. [Online]. Available: https://doi.org/10.1109/SPIN.2019.8711697
6. D. Sehrawat and N. S. Gill. (2018). Analysis of Security Attacks on Lightweight Block Ciphers and their Countermeasures. *Journal of Engineering and Applied Sciences*. 13(20), pp: 8439-8447. [Online]. Available: 10.3923/jeasci.2018.8439.8447
7. D. J. Wheeler, and R. M. Needham. TEA, a tiny encryption algorithm. In: *Proc. of International Workshop on Fast Software Encryption, Springer*, Berlin, Heidelberg, 1994, pp. 363-366.
8. 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, 2013, pp. 3-27. [Online]. Available: https://doi.org/10.1007/978-3-319-05149-9_1
9. D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. S. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, and H. Kim. HIGHT: A new block cipher suitable for low-resource device. In: *Proc. of International Workshop on Cryptographic Hardware and Embedded Systems, Springer*, Berlin, Heidelberg, 2006, pp. 46-59. [Online]. Available: https://doi.org/10.1007/11894063_4
10. 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*, 2014, pp. 306-323. [Online]. Available: https://doi.org/10.1007/978-3-319-13051-4_19
11. M. Usman, I. Ahmed, M. I. Aslam, S. Khan, and U. A. Shah. (2018, March). SIT: a lightweight encryption algorithm for secure internet of things. *International Journal of Advanced Computer Science and Applications*, 8(1). [Online]. Available: 10.14569/IJACSA.2017.080151
12. K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai. "Piccolo: an ultra-lightweight blockcipher." In: Proc. of *International Workshop on Cryptographic Hardware and Embedded Systems*, Springer, Berlin, Heidelberg, 2011, pp. 342-357. [Online]. Available: https://doi.org/10.1007/978-3-642-23951-9_23
13. R. Avanzi. A Salad of Block Ciphers. *IACR Cryptology ePrint Archive*. 2016:1171, 2016.
14. E. Biham. A fast new DES implementation in software. In: *Proc. of International Workshop on Fast Software Encryption, Springer*, Berlin, Heidelberg, 1997, pp. 260-272. [Online]. Available: https://doi.org/10.1007/BFb0052352
15. D. Sehrawat, and N. S. Gill. Performance Evaluation of Newly Proposed Lightweight Cipher, BRIGHT. *International Journal of Intelligent Engineering & Systems.* 12(4), pp. 71-80, 2019. [Online]. Available: http://www.inass.org/2019/2019083108.pdf
16. 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*. 8(5), pp. 1549-556, 2019. [Online]. Available: https://www.ijeat.org/wp-content/uploads/papers/v8i5/E7302068519.pdf
17. W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede. (2015, December). RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. *Science China Information Sciences* 58(12), pp. 1-15. [Online]. Available: https://doi.org/10.1007/s11432-015-5459-7
18. T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi. Twine: A lightweight, versatile block cipher. In: *Proc. of Proceedings of 19th International Workshop on Selected Areas in Cryptography, Toronto*, 2012, pp. 339–354.
19. O. Özen, K. Varıcı, C. Tezcan, and Ç. Kocair. Lightweight block ciphers revisited: Cryptanalysis of reduced round PRESENT and HIGHT. In: *Proc. of Australasian Conference on Information Security and Privacy, Springer*, Berlin, Heidelberg, 2009, pp. 90-107. [Online]. Available: https://doi.org/10.1007/978-3-642-02620-1_7

AUTHORS PROFILE

Ms. Deepti Sehrawat has passed Master of Science and Applications from Department of Computer Science & Applications, M. D. University, Rohtak, India in 2008 and Master of Philosophy from C. D. L. University, Sirsa in 2009. She is currently pursuing Ph. D under the supervision of renowned academician and researcher – Professor Nasib Singh Gill of M. D. University. She has published more than 25 research papers in reputed National and International Journals and Conference Proceedings including IEEE. Her main research work focuses on IoT, Lightweight Cryptography Algorithms, Network Security and Privacy, Big Data Analytics and Data Mining. She has 8 years of teaching experience.

Dr. Nasib Singh Gill is at present senior most Professor of Department of Computer Science & Applications, M. D. University, Rohtak, India and is working in the Department since 1990. He has earned his Doctorate in Computer Science in the year 1996 and carried out his Post-Doctoral research at Brunel University, West London during 2001-2002. He is a recipient of Commonwealth Fellowship Award of British Government for the Year 2001. Besides, he also has earned his MBA degree. He has published more than 245 research papers in reputed National & International Journals, Conference Proceedings, Bulletins, Edited Books, and Newspapers. He has authored seven books. He is a Senior Member of IACSIT as well as a fellow of several professional bodies including IETE and CSI. He has been serving as Editorial Board Member, Guest Editor, Reviewer of International/National Journals and a Member of Technical Committee of several International/National Conferences. He has guided so far 8 Ph.D. scholars as well as guiding about 7 more scholars presently in the areas – IoT, Information and Network Security, Computer Networks, Measurement of Component-based Systems, Complexity of Software Systems, Decision Trees, Component-based Testing, Data mining & Data warehousing, and NLP.