Received January 31, 2019, accepted March 14, 2019, date of publication March 20, 2019, date of current version April 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2906326

A Lightweight Cellular Automata Based Encryption Technique for IoT Applications

SATYABRATA ROY[1], UMASHANKAR RAWAT[1], AND JYOTIRMOY KARJEE[2]
[1]Department of Computer Science and Engineering, Manipal University Jaipur, Jaipur 303007, India
[2]TCS Research and Innovation, Bangalore 560066, India

Corresponding author: Umashankar Rawat (umashankar.rawat@jaipur.manipal.edu)

ABSTRACT The Internet of Things (IoT) devices are resource-constrained devices with limitations such as low computation power, low communication capabilities, low bandwidths, high latency, and short-lived power. Therefore, securing communication among these devices is a key challenge for various sensitive applications. However, the conventional encryption and decryption algorithms, known as ciphers, cannot be implemented because of their inherent complexities of implementation and power requirements. One of the promising options available is to implement light-weight ciphers for these resource-constrained devices. Moreover, the choice of lightweight encryption tool has a great dependency on the type of IoT devices being used in an application. In this paper, a lightweight cellular automata (CA)-based cipher, named as Lightweight CA Cipher (LCC), has been proposed for IoT applications. In the proposed method, encryption is done at the perception layer, where the sensor nodes are deployed and decryption is done at the network layer where gateway devices are installed. The experimental results show that the proposed method is efficient than some of the existing ciphers like DES, 3DES when randomness, execution time, and implementation simplicity are considered as prime requirements. This cipher passes the randomness tests as prescribed by the National Institute of Standards and Technology (NIST), and it also passes all the DIEHARD tests and it establishes the security feature of LCC. Though it is specially designed for resource-constrained environments, it can be scaled up for a large number of sensor nodes.

INDEX TERMS Internet of Things, cellular automata, ciphers, encryption, gateway devices, security.

I. INTRODUCTION

Internet of Things (IoT) is a heterogeneous collection of "things" embedded along with varieties of sensors, actuators, software, and electronics components, which are connected via Internet to collect and exchange data with each other. The IoT devices are furnished with sensors and some processing power which enable them for being deployed in various environments such as in farming/agricultural lands (to monitor crops), forests, and industrial warehouses. Figure 1 depicts a variety of common IoT applications, including smart grids, smart parking, and smart health care.

The report of International Data Corporation published in 2013 states that the expeditious growth in the number of IoT devices deployed is predicted to touch the figure of 41 billion within 2020 with a market of 8.9 trillion dollars [1]. Absence of any human interference is the main

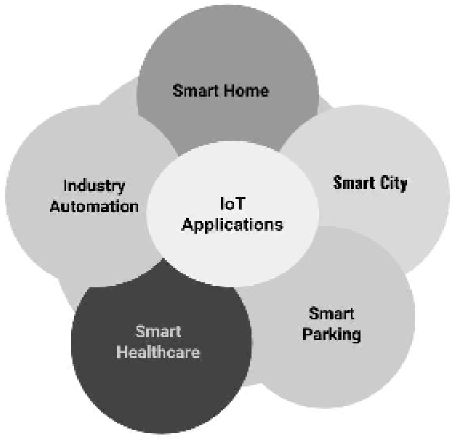

FIGURE 1. IoT applications.

difference between IoT and traditional Internet. Individual person's behavioral information can be created, analyzed and based on these; proper actions need to be considered for the

The associate editor coordinating the review of this manuscript and approving it for publication was Ennan Zhai.

application of IoT devices [2]. Various useful services supplied by different IoT applications are of great benefit to human-life. However they comes with a very huge cost while considering individual's security and privacy protection. Since IoT manufacturers did not succeed in implementing a robust and secure system at the device level, the security experts have already warned about the potential risk of using huge numbers of vulnerable devices connected to the Internet [3].

The first IoT botnet was discovered by a researcher at Proofpoint, a security enterprise firm in December 2013. According to their survey, more than 25% of botnets were comprised of devices except computers. These include smart baby monitoring systems, smart TVs, and smart household appliances. Later, a Manchester, New Hampshire-based domain name services (DNS) provider, Dyn, experienced disruption of services as a consequence of a well-coordinated attack [4]. Many users in October, 2016, reported the in-accessibility of many popular websites including Twitter, SoundCloud, Netflix, Reddit, Etsy, and Spotify. and The New York Times, induced by a distributed denial of service (DDoS) attack using a consumer IoT device network.

The security and privacy issues remain a huge concern for IoT devices and it added a whole new degree of disruption to the online privacy for the consumers. These are mainly because of the capabilities of these devices. These devices not only gather personal information like phone numbers, and names. but they can also keep track of every user-activities, like when a particular user is in his room, when will they go out for dinner and what all they had in dinner, etc. Hence, following these never ending array of breaches of major private data, the consumers are very much worried about keeping too many personal information/data-bases in a private or public cloud with valid reasons [5].

There are various published survey articles on IoT security and privacy issues with their challenges. A detailed analysis about the existing solutions related to four standard IoT communication protocols i.e. physical, Medium Access Control (MAC), Network and Application are provided by Granjal *et al.* [6]. They have also discussed the possible cross-layer communication mechanisms. The research challenges such as recent security trade-offs and privacy policies are presented by Sicari *et al.* [7] in a work where they have identified and categorized the possible security issues in eight broad categories as 1) access control; 2) authentication; 3) privacy; 4) confidentiality; 5) trust; 6) mobile security; 7) secure middleware and 8) policy enforcement. They have also raised some open research issues and suggested some way outs for proceeding into those directions. Cui *et al.* introduced a proxy aided attribute based encryption mechanism at edge level [8]. It will provide security in the edge which is closer to the devices and would also contribute greatly in decreasing latency of data processing. Dao *et al.* [9] have proposed a social networking based authentication protocol for low-powered IoT devices. This authentication protocol provides multi-security levels specific to each IoT applications. Zhou *et al.* developed an efficient access control mechanism of very fine granularity for bulk operations of interactive IoT devices [10]. Roman *et al.* [11] analyzed the centralized and distributed IoT architectures where they have proposed an attacker-model which is applied to both distributed and centralized IoT infrastructures.

In symmetric key cryptosystem, only one secret key is used for both encryption and decryption. Sensor data have been encrypted using L2D-CASK with the help of FPGA in [12]. Here, the proposed method uses a key-length of 512 bits as hybrid CA rule vector [13] and it uses simple XOR operations with either 3 inputs or 2 inputs at a time. Since, sensor nodes have low memory space, the proposed method addresses this constraint without degrading the performance of the encryption in terms of randomness and computational complexities. It has been possible because of the inherent capabilities of CA to generate chaotic sequences [14]. Proposed methods shows equivalent encrypting efficiency as that of DES and 3-DES in terms of randomness. Proposed method is resistant to different cryptanalysis attacks like brute-force, linear cryptanalysis attacks, differential cryptanalysis attacks and it satisfies the confusion and diffusion properties. The proposed method randomly selects both the key configurations to be applied for encryption and the number of iterations to be executed for encryption and decryption. This has made the algorithm more robust in nature.

The rest of the paper is organized as follows. A brief background and related works done is presented in section II. Section III, describes how the proposed method using CA can be applied in the three-layer architecture. Section IV consists of the result evaluation and analysis of the proposed method. This shows the efficacy and feasibility of the work. Section V concludes the proposed work done and gives some future research directions.

II. BACKGROUND AND RELATED WORK

Although, IoT devices have been implemented in various engineering applications, but these are not being used widely for security requirements use-cases due to the IoT device constraints. Trappe *et al.* [15] discussed various important reasons for not being able to use security features in case of IoT devices in contrast of using them in traditional internet. They have discussed many constraints and the immediate effects of using current cryptographic tools that are being used in traditional internet. Among all the other limitations, battery capacity and computation power of the devices are the two major constraints.

- **Extended Battery Life**: The IoT devices may be deployed in environments where there is no availability of charging such as in agricultural field or in forests. What they will have in such case is a very limited energy for executing the pre-designated functionality. Hence, heavy security routines can drain the devices' energy to zero. Three possible approaches, as suggested by [16], can be incorporated to tackle these issues. First, to use minimum security requirements at device level, which is not a very enthusiastic one, especially in case of dealing with sensitive data. The second being the use of enlarged battery capacity, which is also not always possible because the limited size of the devices would not allow the usage of a large battery size. These devices should be lightweight in nature. The third approach is energy harvesting by the deployed device. The devices can be made to harvest energy from natural resources like sunlight, heat, wind, water-current and vibration, etc. But, this approach would end up requiring upgraded and sophisticated hardware that would boost the price significantly.

- **Lightweight Computation**: Conventional cryptography is not applicable on IoT systems [15] as the devices have limited memory space. The low memory space is incapable of handling the larger requirement of computing and storage capacity by modern cryptographic algorithms. To support security features in the constrained environment, reuse of some existing functionalities has been proposed in the literature. For example, an authentication system at physical layer can be established with the help of signal processing at the receiver end to check whether a transmission has come from an intended transmitter at the intended locations or not. Alternatively, the analog information can be encoded efficiently by using explicit characteristics of the transmitters. These analog features cannot be controlled or forecasted and can serve as a key. Since, this type of authentication uses radio signals, it has very little overhead or no overhead of energy constraints.

Shafagh et al. [17] suggested an algorithm for query processing in encrypted format for IoT. The scheme allows secure storing of encrypted IoT information on the cloud and it also supports efficient processing of database queries on encrypted data. More specifically, they have used other lightweight cryptographic tools replacing additive homomorphic and order-preserving encryption with the help of Elliptic Curve Elgamal as well as order preserving mutable encoding algorithms. Here, they have made some changes to cope up with the computation constraints of IoT devices. The proposed scheme uses an end-to-end (E2E) system to substitute web based application for communication. This E2E system stores every encrypted information received from the personal devices on a cloud database. Further, the encoding/decoding part is executed at the client-end. However the key generator will be placed at the personal device. This eliminates the requirement of using a trusted proxy having access of all secret keys. The system architecture involves three distinct parties: 1) the users; 2) the cloud and 3) the IoT devices. A gateway or a smart device like any wearable can directly upload the application generated data on cloud for data storage.

An efficient approach for reducing latency for IoT during the process of different complex queries on encrypted data is proposed by Kotamsetty and Govindarasu [18]. This approach uses breaking down of the large result-set generated by queries into small-sized data-sets. This will allow the computational works to be carried out on a smaller data sets at the time of remaining encrypted data fetching operations. They proposed a new algorithm to be applied on the initial data sets and it would adoptively fit according to the size for minimizing the difference between communication and computation latency in every iteration, gradually. This is applied to decide appropriately the data size that would be requested in every iteration. The algorithm is proposed to have two variants: the first one comes with an initial size which is a fraction of the size of a large query. The second variant, on the other hand, starts with a fixed initial size. The experimental results illustrated that this approach outshines the existing solutions with respect to larger query processing latency for IoT.

A lightweight encryption technique for smart homes was proposed by Salami *et al.* [19]. This scheme is based on stateful identity-based encryption (IBE), where there is no need of digital certificates because of using public keys that are made of simple identity strings. This method is popularly known as Phong, Matsuka and Ogata's scheme of stateful IBE. It is a hybrid scheme where the Diffie-Hellman encryption scheme is combined with stateful IBE. The encryption is done in two phases: key encryption and data encryption to improve its efficiency. The primary focus is on second one, i.e. data encryption. This is because the key encryption phase produces larger ciphertext size as compared to the size produced by data encryption. This division, obviously ends up in two sub-algorithms: KEYEncrypt and DATAEncrypt. KEYEncrypt is used to encrypt a given session key and the DATAEncrypt is used to encrypt data. The ciphertext generated from these algorithms is communicated separately in such a way that the data ciphertext generated from second sub-algorithm needs not to be attached with key ciphertext each time it is transmitted. Results of the study depicts that this technique can resist known plaintext attacks. It also conveyed that it outperforms the existing regular IBE techniques with respect to the speed of encryption process. Further, it also reduces the communication overhead by one-third.

In recent years, IoT applications are implemented using different types of architectures like two layer architecture, three layer architecture, SoA-based architecture, etc. In each layer there are security concerns related to integrity, confidentiality, availability, authentication and identification, privacy and trust. Y. Yang *et al.* has reported security issues at each layer of existing IoT architecture [16]. They have also shown possible types of attacks and some probable preventive measures in the literature. Among all the available architecture models, Three-layer architecture model and Service-oriented Architecture (SoA) model are being implemented widely. In the three-layer architecture model, as shown in figure 2, IoT framework is distributed in three primary layers: 1) perception layer 2) network layer and 3) application layer that are described below [20].

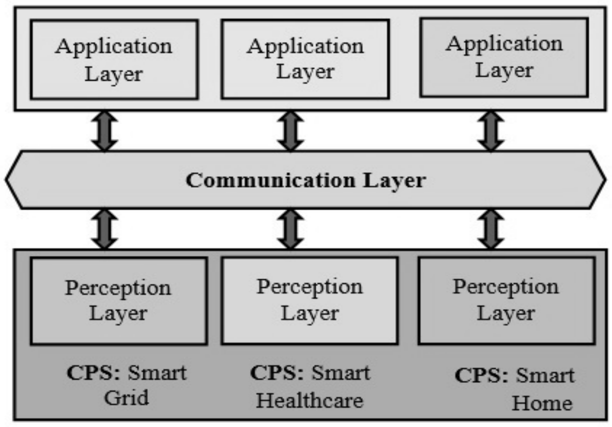

FIGURE 2. IoT and CPS integration.

1) PERCEPTION LAYER

This layer is also termed as physical layer where it interacts with physical devices like sensors and actuators, etc.. The purpose of this layer is to connect the things (i.e. devices) with IoT network for providing a wide range of services by transmitting the data to and from upper layers.

2) NETWORK LAYER

This layer is also called the transmission layer which is considered as a middle-layer in the IoT architecture. It is responsible for receiving processed information from perception layer and then finding possible routes for transmission of these data and information to different IoT hubs, devices and applications through integrated inter and intra networks. Implementation of this layer is very critical because of the heterogeneous nature of different enabling communication technologies like Wifi and Bluetooth, etc. which uses various different network protocols. It also becomes very tedious job to integrate several heterogeneous network devices like hubs, switches, cloud computing platforms and gateways, etc. into a single platform.

3) APPLICATION LAYER

This layer is recognized as business layer, which is considered as the top-most layer in three-layered IoT architecture model. It receives data from the network layer and uses these for providing different service requests like storage services and analyzing services as required in smart grid, smart city and smart home, etc.

This three layer architecture is rudimentary for IoT systems and hence implemented widely in most of the cases, but due to the diverse nature of operations and service requests required at different layers, a more sophisticated architecture, called SoA-based architecture has been introduced. For example, the transmission layer needs different data aggregations, computing techniques, whereas the application layer requires different data mining and analytic techniques for service provided to complex IoT applications. So, to make this architecture more robust, generic and flexible in nature, an extra layer named service layer is proposed to be implemented in between application layer and network layer [20], [21].

The recent studies show that the security approaches adapted for IoT applications suffer from the complexity of setting up the platform for encryption and decryption at different layers. Key management is one of the major areas of concern in this context. Besides, low computational power and low availability of memory spaces in sensor devices have made the encryption and decryption techniques more difficult to implement at device level. Here, a novel CA-based encryption technique has been proposed where it takes care of the resource-constraints at the device level and would be easily implemented due to the inherent implementation-simplicity of CA. This method would encrypt the raw sensed data from sensor devices to the intermediate fog nodes where these data can be decrypt. The proposed method is a symmetric key cipher and is equivalent with the conventional DES and 3-DES techniques in terms of randomness. The randomly chosen CA rule configuration and iterations as secret key has contributed in achieving this randomness. However, the key management is not addressed in this technique.

III. PROPOSED SYSTEM MODEL

In this section, detailed functionalities of the proposed work have been discussed. At first, the significance and key features of CA in lightweight encryption techniques and how the proposed approach exploits it to develop LCC is discussed. Then, Group Cellular Automata (GCA) based algorithm called Lightweight CA Cipher (LCC), which is deployed at IoT devices for data encryption, has been proposed. The decryption process is performed at the fog nodes in the network layer as depicted in figure 7.

A. SIGNIFICANCE OF CA IN LIGHTWEIGHT ENCRYPTION

Cellular automata (CA), introduced by von Neumann [22] and Ulam [23], are known to be capable of modeling complex systems, though the basic model of it is a simple grid-like structure. It was proposed to study various biological processes like self-reproduction, spread of diseases, droplet behavior, etc. Any system consisting many similar discrete components can be modeled as cellular automata that performs local interactions and produce deterministic results. Generally, complex systems can be divided into identical sub-components each of which obeys some simple rule. The massive number of components act together to make up the entire bigger system yielding a complex behavior.

CA can be used in lightweight cryptography because of its inherent simplicity in implementation in hardware or sensor devices using VLSI/FPGA circuits. Any type of controller board can be used to implement it. The initial seed is the only requirement for its dynamic next state evolution. With appropriate selection of rule or rule vector, CA is also capable of generating chaotic sequences and pseudo random numbers (PRNs).

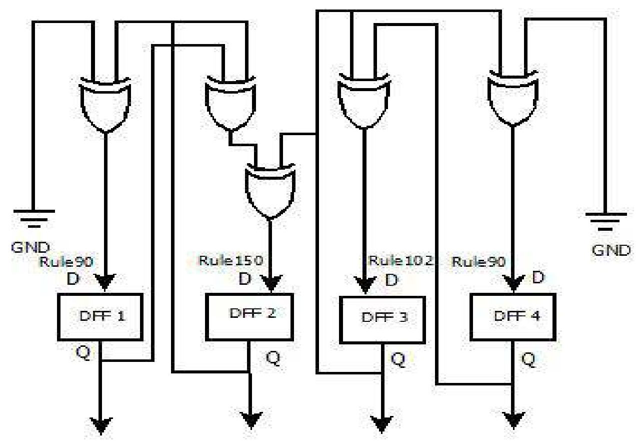

FIGURE 3. A hybrid null-boundary CA.

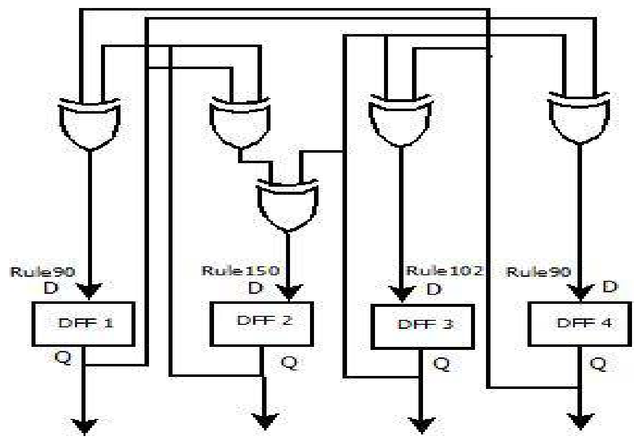

FIGURE 4. A hybrid periodic-boundary CA.

The most important characteristic of using CA in lightweight cryptography is that prediction of the next state by looking at the initial state is next to impossible, because very similar initial state configurations also lead to a totally divergent and completely different final states. Reversion of state values to any given timestamp from current state is computationally hard problem and is very costly.

Recently some good cryptographic algorithms using CA characteristics have been introduced [24]–[26] and they have shown comparatively better behaviors at times with the likes of DES, 2-DES etc. In this paper, a very efficient rule selection algorithm is used to generate a random sequence. This approach is a symmetric key encryption technique that uses same key (rule vectors) for encryption and/or decryption. There are many well-known algorithms available for encryption of IoT data-sets at application layer or network layer.

B. KEY FEATURES OF CA

All the cells of cellular automata evolve by a local and simple rule that determines the value of each of them in the next step depending on the values of neighbors involved in the previous step. In summary, the values of each cell at step t + 1 is a function of values of the neighbors of the cell under consideration at step t. Mathematically it is represented as:

$$S_i^{t+1} = f_i(S_{i-1}^t, S_i^t, S_{i+1}^t) \tag{1}$$

A 1-D CA having n number of cells linked together as a straight line is termed as non-circle CA, where the left and right neighbors of the leftmost and rightmost cells, respectively are considered to have state-value as '0' and hence it is known as null-boundary CA. When the extreme two cells of a 1-D CA is considered to be adjacent to each other, it is called a periodic-boundary CA. When all the f_i's are similar $\forall i = 1, 2, \ldots, n$, the CA becomes homogeneous or uniform CA, otherwise it is called non-homogeneous or hybrid CA. Block diagrams of 1-D hybrid null-boundary and periodic-boundary CA are shown in figure 3 and figure 4, respectively. Suppose, G is a field and all $S_i \in G$, then an affine CA is defined as:

$$f_i(S_{i-1}^t, S_i^t, S_{i+1}^t) = c_{i-1}S_{i-1}^t + c_iS_i^t + c_{i+1}S_{i+1}^t + k_i \tag{2}$$

where, $c_{i-1}, c_i, c_{i+1}, k_i \in G$, is the multiplication and + is the addition operation under G.

In case of a 1-D CA having 2-states and 3-neighbors inclusive of the cell under consideration, 2^3 distinct neighborhood arrangements can be configured and there can be 2^{2^3} unique mappings from all these 2^3 neighborhood configurations. Each mapping is called as a CA rule. For example, a CA featured by a rule, called Rule 60 represents an evolution from neighborhood arrangement to the next state as described below:

111	110	101	100	011	010	001	000	
0	0	1	1	1	1	0	0	$(60)_{10}$

The corresponding combination-logical expression for Rule 60 is:

$$Rule60 : S_i^{t+1} = (S_i^t \oplus S_{i-1}^t) \tag{3}$$

The expression signifies that the state value of i^{th} cell at time stamp t + 1 is dependent on the state values of i^{th} cell and the left neighbor i.e. $(i-1)^{th}$ cell at time stamp t. Similarly, the combinational logic expression for some other rules which are used in this paper, are given below:

$$Rule102 : S_i^{t+1} = (S_{i+1}^t \oplus S_i^t) \tag{4}$$
$$Rule150 : S_i^{t+1} = (S_{i-1}^t \oplus S_i^t \oplus S_{i+1}^t) \tag{5}$$
$$Rule51 : S_i^{t+1} = \overline{S_i^t} \tag{6}$$
$$Rule153 : S_i^{t+1} = \overline{(S_{i+1}^t \oplus S_i^t)} \tag{7}$$
$$Rule195 : S_i^{t+1} = \overline{(S_i^t \oplus S_{i-1}^t)} \tag{8}$$
$$Rule75 : S_i^{t+1} = \overline{S_{i-1}^t}(\overline{S_i^t} + S_{i+1}^t) + S_{i+1}^t + S_i^t\overline{S_{i+1}^t}S_{i-1}^t \tag{9}$$
$$Rule90 : S_i^{t+1} = S_{i-1}^t \oplus S_{i+1}^t \tag{10}$$

So, a CA rule can be expressed as:

$$CA \quad Rule-No. = \sum_{k=0}^{n-1} S_k i^k \tag{11}$$

where, S_i is the state value (0 or 1), i is the current cell and k is a positive integer. A CA rule implemented by using EXNOR and/or EXOR logic expression is known as additive CA. Further, if an additive CA is implemented using EXNOR logic, it is called a complemented CA, otherwise it is called a non-complemented CA. Here, (3), (4), (5) and (10) represent non-complemented CA rules; whereas, (6), (7) and (8) represent complemented CA rules. (9) is not an expression of any additive CA for obvious reasons. A CA rule vector is a combination of different rules applied on different cells of a hybrid CA. Each CA has a characteristic matrix T and a characteristic polynomial [27]. A CA is termed as a group cellular automata (GCA) if and only if $T^m = \mathrm{I}$, where, I is the identity matrix and m is a positive integer. The complement operation on a group CA also generates a group CA. State transition of a CA by using characteristic matrix can be represented as:

$$S_i^{t+1} = [T][S_i^t] \tag{12}$$

C. PROGRAMMABLE CELLULAR AUTOMATA (PCA)

If the logical expressions of the CA rules are observed, it can be found out that there are some rules whose representation differs only in one position, for some others, the positional difference is two and so on. So, with careful application of control signals and switches, different CA rules can be applied on a cell at different time stamps. So, an m-cell CA structure can be utilized efficiently for 2^m rule configurations. The configurations can be implemented in hardware by using some control lines, switches and a ROM having control program loaded in it. By using EEPROM, one can allow massive flexibilities to this PCA structure by periodically changing the control logic. A configurable 1-D PCA cell of non-complemented and complemented CA having 2-states, 3- neighborhood, are shown in Figure 5 and Figure 6. Cellular automata states can also be represented as permutation groups having some predefined order of that group [27].

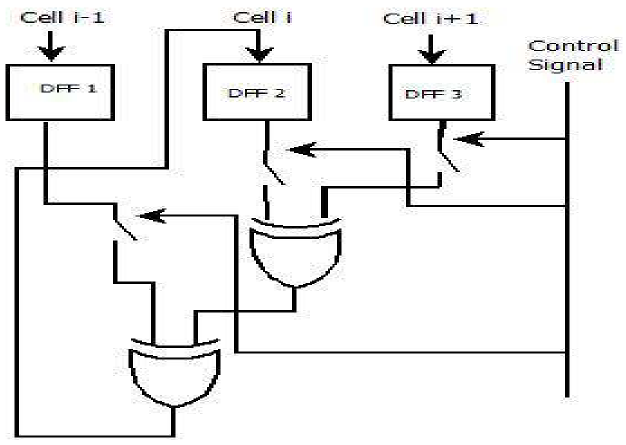

FIGURE 5. A non-complemented PCA cell.

Encryption at IoT devices is considered as perception layer and the decryption of data at the fog node is considered as the

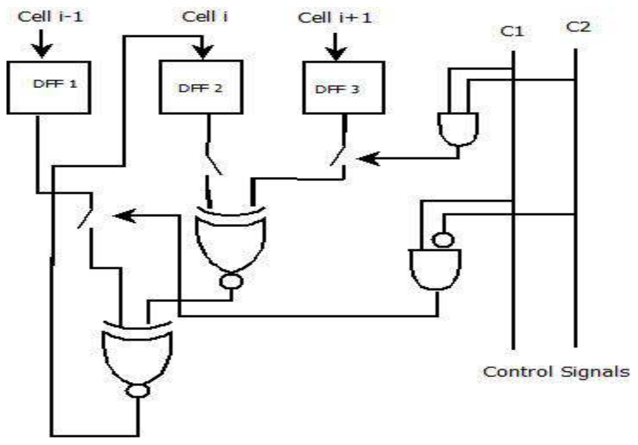

FIGURE 6. A complemented PCA cell.

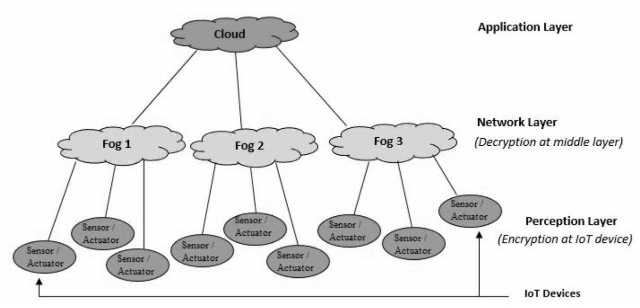

FIGURE 7. Proposed architecture model.

network/middle layer. In an IoT application scenario, assume W number of IoT devices are installed randomly. In case of any event, M number of devices out of W wake up and the remaining devices remain in sleeping mode in the network. Also assume, M number of devices form a single cluster and a Cluster Head is selected at random in the network [28]. Cluster Head is responsible for collecting data from each node in the cluster and then it transmits the collected data to the fog node deployed in the IoT framework.

Observed data, d_k is the sum of sensed data s_k and some noise, n_k by each device, k. Mathematically, it is represented as:

$$d_k = s_k + n_k \tag{13}$$

Here, the observed data d_k is extracted by device k under Additive White Gaussian Noise (AWGN) channel [29]. After extraction of d_k, the device k transmits it to the Cluster Head at every timestamp t. The Cluster Head, deployed at perception layer, stores each dk in a matrix D. D is a matrix that stores observed data d_k as a block of sensed data, under a given time interval t given by:

$$D = \begin{bmatrix} d_1^1 & d_1^2 & \dots\dots & d_1^N \\ d_2^1 & d_2^2 & \dots\dots & d_2^N \\ \dots\dots & \dots\dots & \dots\dots & \dots\dots \\ d_M^1 & d_M^2 & \dots\dots & d_M^N \end{bmatrix} \tag{14}$$

Here, a block of data, which is extracted by every device k at t, is denoted as N. A noise is added to each value d_i^j, once the matrix D is obtained. After that, it is encrypted through lightweight CA based cipher.

The proposed cipher is a symmetric key cipher, where GCA rule vectors are used as the secret key. These secret keys will be selected at random.

There are 256 rules for a one dimension, 3-neighborhood CA. Each rule has its own logical expression for next state production. For example, logical expressions for some rules are mentioned through equations (4) to (10). The GCA-rule characteristic matrix, T has the following property in the fundamental state transition function:

$$T^{2n} = I \tag{15}$$

where, 2n is the cycle length of the GCA rule. The proposed technique has three major algorithms: Rule-vector Generator, Encryption algorithm and Decryption algorithm.

As mentioned earlier, cellular automata rules can be generated depending on the radius and number of possible values that a cell can have as state-value. In case of 1-D cellular automata with radius, r = 1 (three possible neighbors, i.e. left, right and itself), 256 possible CA rules can be generated, but all the combinations do not form GCA rules. So, to generate GCA rule combinations, Algorithm 1 (Rule-vector generator) has been used.

Algorithm 1 Rule-Vector Generator

Input: RV8 (All the 256 rules generated from a 1-D, 3-neighborhood CA), 512 bits Initial Values (IV).

Output: RVList512, a list of GCA rule vectors having cycle length equals to 8.

```
1: Begin
2: Generate all possible permutations with repetition, for
   8-length CA rules, RV8.
3: Randomly choose any one RV8. Expand this RV8 by
   applying concatenation operation to make it a 512-length
   sequence, RV512.
4: Initialize the PCA cell with an arbitrary bit-stream
   of 512 length, IV.
5: Apply RV512 to the PCA cells for 8 iterations.
6: If the output after step 4 is equal to the IV, store the
   RV512 in a list, RVList512.
7: Repeat steps 2 to 5 until all 256 combinations are
   verified.
8: End
```

This algorithm randomly takes eight CA rules, RV8 out of available 256 CA rules and then generates a GCA rule vector, RVList512, having 512 CA rules that can be used in encryption. In this process it takes an initial vector of 512 random bits to verify the GCA property. This RVList512 has cycle length equals to 8. So, for the eavesdropper, it would cost an exponential order time (almost infeasible) to guess the 512 rules, nullifying the possibilities of brute-force attack.

In step 2 of Algorithm 1, all possible RV8s are generated and any one RV8 is chosen randomly. Step 3 expands this RV8 into RV512, a combination of 512 CA rules to verify whether this combination forms a GCA having cycle length of 8. In step 4 through step 6 this RV512 is tested to generate the same initial vector after 8 iterations. Thus cycle length of 8 is guaranteed. Step 7 generates and stores all such RV512s into a list named as RVList 512.

Algorithm 2 Encryption Algorithm (Deployed at Perception Layer)

Input: The 512 bits data from D.

Output: 512 bits Ciphertext, D_CT.

```
1: Begin
2: Start with 512 random bits padded with 0's in its extreme
   left and right obtained from D.
3: Choose a random number within the range (1-8) as
   iteration.
4: Pass the array from step 2 as input to the PCA.
5: Take next 3 bits from input array.
       1) Randomly choose a rule vector, RV512 generated
          by Rule-vector Generator.
       2) Apply the rule vector to get the next bit at next
          timestamp.
       3) Store the output bits in an array, D_CT.
       4) Repeat steps 5.1 to 5.4 till all the bit-sets of length
          3 are considered.
6: Repeat steps 4 to 5.4 till the end of the iteration chosen
   at step 3.
7: End
```

Algorithm 2 is the actual encryption algorithm which would be deployed at IoT devices at the perception layer. It takes 512-bits block of data at a time and generates 512-bits of ciphertext as output. The number of iterations is also chosen randomly by the sender. Step 2 sets up the null-boundary CA by adding two zeros to the extreme left and extreme right side respectively. The number of iteration is chosen randomly at step 2. At step 4, the 512-bits block of step 2 is passed as input, where at step 5, a randomly chosen RV512 from RVList512 is applied in the 512-bits block. The random RV512 is applied as per 1-D, 3-neighborhood CA characteristic, i.e. taking three bits of data at a time. This encryption process is applied to the entire data set. Step 6 confirms that the process is repeated for the very number of iterations chosen at step 3. The output is stored in a matrix named as D_CT.

Algorithm 3 is the decryption algorithm and is applied by the receiver situated at the network layer/middle layer. The decryption algorithm is exactly the opposite of encryption algorithm. Here it is assumed that the receiver gets the number of iterations to be run for decryption from the sender.

IV. SIMULATION AND RESULT ANALYSIS

In this section, first security aspect is analyzed and then the simulation and results are analyzed in details.

Algorithm 3 Decryption Algorithm (Deployed at Network/ Middle Layer)

Input: 512 bits Ciphertext, D_CT.
Output: The 512 bits original data D_DT.

```
1: Begin
2: Start with 512 random bits padded with 0's in its extreme
   left and right obtained from D_CT.
3: Get the iteration chosen at step 2 of Encryption Algo-
   rithm.
4: Pass the array from step 2 as input to the PCA.
5: Take next 3 bits from input array.
     1) Get the same rule vector, RV512 generated by Rule-
        vector Generator at step 5.1. of Encryption Algo-
        rithm.
     2) Apply the rule vector to get the next bit at next
        timestamp.
     3) Store the output bits in an array, D_DT.
     4) Repeat steps 5.1 to 5.4 till all the bit-sets of length
        3 are considered.
6: Repeat steps 4 to 5.4 till the end of the iteration chosen
   at step 3.
7: End
```

A. SECURITY ANALYSIS OF LCC

LCC is a block cipher, for which, diffusion and confusion are two essential criteria. LCC satisfies these criteria as follows:

1) DIFFUSION

The main aim of diffusion is to distribute the repeated bits or redundant bits of plaintext throughout the ciphertext. In other words, every bit of plaintext should affect as many bits of ciphertext as possible. LCC achieves the diffusion by smart arrangement of neighborhood bits of CA along with XOR/XNOR operations. LCC uses null boundary CA, and expansion of RV8 to make RV512 is done to get the GCA property. The encryption consists of many iterations (as chosen). It can be observed that right from the first iteration, each intermediate output is depending on the previous bit sequence. So the final ciphertext will depend entirely on the plaintext, i.e. the LCC achieves complete diffusion.

2) CONFUSION

The relationship between the secret key and the ciphertext must be non-linear and as complex as possible. LCC attains high degree of confusion because of the following reasons:

- In each iteration, the intermediate outputs depend on the selected RV512 and the 512 bit plaintext. The number of iterations to be performed is also chosen randomly. Both the RV512 and number of iteration is served as secret key in the symmetric key LCC.
- The iterations of LCC is based on both complemented and non-complemented GCA rules. These factors make LCC highly non-linear as well as these factors contributes greatly to prevent LCC from linear cryptanalysis attack.

Differential cryptanalysis attack is considered to be an important attack against block ciphers. It is a conventional attack introduced by Biham and Shamir to apply against DES. This attack utilizes difference transmission from plaintext to ciphertext. This propagation of difference is used to set probabilities to the probable keys and finally to identify the most probable key. If the maximum differential probability happens to be very small, then differential cryptanalysis attack against any block cipher becomes ineffective. Here, 8 bit input and 8 bit output block is considered to be an S-box. It can be observed that difference in one input bit of an S-box reflects 64 different bytes after one round. So, after randomly chosen n number of iterations, the differential probability will generate an ultra-low probability value. It makes LCC resistant to differential cryptanalysis attacks.

B. RESULT ANALYSIS OF LCC

Here a deployment scenario has been considered for experimental purpose, where 25 IoT devices are deployed in an outdoor environment (such as dense forests, agriculture fields and hills, etc.). However, the number of IoT devices can be any number. In the deployment environment, fog nodes and a Cloud are also deployed. The IoT devices which are within the communication range of a fog node, transmit the observed data to the fog node and further forward data to the Cloud.

For the data processing, each IoT device generates 64 observations at each timestamp. To capture the data processing, input to the Encryption algorithm is a matrix, D having dimension, $M \times N = 25 \times 64$, where 25 IoT devices are used and each device generates 64 observations at each time interval. Thus, the whole sensed data is divided as a block of 25×64 matrix, D, transmitted from fog node to the Cloud. The proposed algorithm is easy to implement when compared to DES and 3-DES. Besides, the NIST tests and DIEHARD tests are performed on LCC cipher, where it generates 512 bits of output for 512 bits of input and 25 such nodes have been considered for experimental purpose. This number can be scaled up/down to any number. However, for NIST tests 6,40,000 output bits were considered at a time. The details of each NIST test like test purpose, function call, test statistic and reference distribution, test description, decision rules, interpretation of results can be found at [30]. These statistical tests are based on either standard normal distribution (bell-shaped curve) or chi-square(χ^2) distribution (left skewed curve) as reference distribution. Comparison of the value of the test statistics obtained from the encryption algorithm is carried out using standard normal distribution, whereas chi-square(χ^2) distribution is used to compare the fitness of the observed frequencies of sample with the corresponding expected frequencies. The mathematical expression used for standard normal distribution is $Z = (x - \mu)/\sigma$, where x is the test statistic value of the sample, μ and σ^2 are the expected value and variance respectively. The mathematical expression used for chi-square(χ^2) distribution is $\chi^2 = \sum((o_i - e_i)^2/e_i)$, where, o_i and e_i are the observed frequency and expected frequency respectively.The test results are shown in Table 1.

TABLE 1. Results of NIST tests.

Test Name	P-Value	Result
Frequency	0.192763	Pass
Block Frequency	0.867142	Pass
Cumulative Sums	0.987324	Pass
Runs	0.198243	Pass
Longest Run of Ones	0.197232	Pass
Rank	0.015963	Pass
FFT	0.048716	Pass
Non-Overlapping Template	0.982624	Pass
Universal	0.983744	Pass
Approximate Entropy	0.962606	Pass
Serial	0.120393	Pass
Linear Complexity	0.110952	Pass

Standard normal distribution is referred for a) The Frequency (Monobit) Test, b) The Cumulative Sum (Cusums) Test and c) The FFT test. Half normal distribution curve is referred for "Universal Statistical" Test. The chi-square(χ^2) distribution is referred for a) The Block Frequency Test, b) The Runs Test, c) The Longest Runs of Ones in a Block Test, d) The Rank Test, d) The non-overlapping Template Matching Test, e) The Approximate Entropy Test, f) The Serial Test and g) The Linear complexity Test. If the obtained P-value is <0.01, then the sequence is non-random, otherwise it is random. For testing purpose, 6,40,000 output bits are considered. Test results of the LCC, mentioned in Table 1 clearly shows that each test result has passed the threshold value for passing. Hence, the encryption algorithm proposed is random.

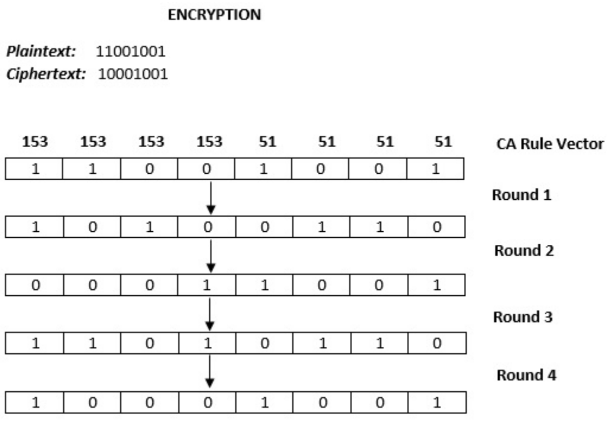

FIGURE 8. Encryption using PCA rules.

There are 2^8 possible CA rules to be applied in the 512 bits PCA cells. Hence, the intruder has to check 2^{4096} combinations which is computationally infeasible for brute-force attacks. The values of bits at next timestamps of any 8-bit binary number by a selected GCA rule is given in figures 8 and 9. Here CA rule numbers 153 and 51 are used as example and null boundary CA is considered.

The initial values are loaded in a vector of 8 bits, for example. Then extreme left and right cells are considered to have value '0'. When CA rules, as mentioned in the figures 8 and 9,

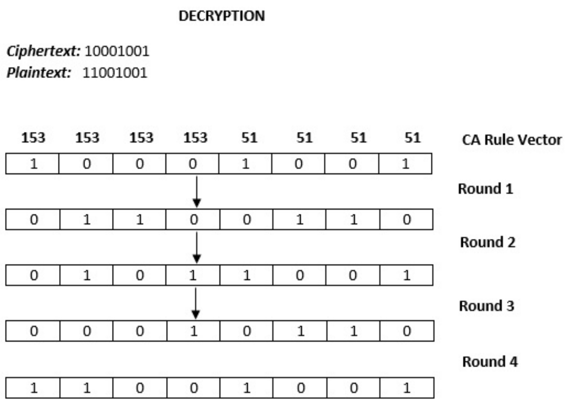

FIGURE 9. Decryption using PCA rules.

TABLE 2. Part of original matrix, D, after adding noise.

19	19	19	18	18	17	18	19
19	19	19	18	18	17	17	19
19	19	19	18	18	17	17	19
19	19	19	18	18	17	17	19
19	19	19	18	18	17	17	19
19	19	19	18	18	17	17	19
19	19	19	18	18	17	17	19
19	19	19	18	18	17	17	19
19	19	19	18	18	17	17	19
19	19	19	18	18	17	17	19
19	19	19	18	18	17	17	19
19	19	19	18	18	17	17	19
19	19	19	18	18	17	18	19
19	19	19	18	18	17	18	19
19	19	19	18	18	17	18	19
19	19	19	18	18	18	18	19
19	19	19	18	18	17	18	19
19	19	19	18	18	17	18	19
19	19	19	18	18	17	18	19
19	19	19	18	123	17	18	19
19	19	19	18	18	17	19	19
19	19	19	18	18	17	19	19
19	19	19	18	18	17	19	19
19	19	19	18	18	17	19	19
19	19	19	18	18	17	19	19

are applied in each cell, then their next state values are changed as depicted. Thus, after four rounds, cipher text is generated. Similarly, in Figure 9, the decryption is performed by following the reverse operations. The same operations can be followed for 64 bits. Table 2 and 3 show portions of data obtained from D and D_CT.

The proposed algorithm has also been gone through the die-hard tests [31]. The test result is shown in Table 4. It can be observed from Table 4 that the values are within the range [0,1) which signifies that the ciphertext sequence contains truly independent bits of random numbers. Based on

TABLE 3. Part of encrypted matrix, *D_CT*.

255	255	255	186	186	85	186	255
255	255	255	186	186	85	85	255
255	255	255	186	186	85	85	255
255	255	255	186	186	85	85	255
255	255	255	186	186	85	85	255
255	255	255	186	186	85	85	255
255	255	255	186	186	85	85	255
255	255	255	186	186	85	85	255
255	255	255	186	186	85	85	255
255	255	255	186	186	85	85	255
255	255	255	186	186	85	85	255
255	255	255	186	186	85	85	255
255	255	255	186	186	85	186	255
255	255	255	186	186	85	186	255
255	255	255	186	186	85	186	255
255	255	255	186	186	186	186	255
255	255	255	186	186	85	186	255
255	255	255	186	186	85	186	255
255	255	255	186	186	85	186	255
255	255	255	186	151	85	186	255
255	255	255	186	186	85	255	255
255	255	255	186	186	85	255	255
255	255	255	186	186	85	255	255
255	255	255	186	186	85	255	255
255	255	255	186	186	85	255	255

TABLE 4. Result of Die-Hard tests.

Sl. No.	Test Name	P-value	Result
1	Birthday Spacings	0.76126822	Pass
2	Overlapping 5-Permuatation	0.85405887	Pass
3	Binary Rank 32*32	0.36803064	Pass
4	Binary Rank 6*8	0.90049058	Pass
5	Bitstream	0.12152550	Pass
6	OPSO	0.94002074	Pass
7	OQSO	0.95170724	Pass
8	DNA	0.46795723	Pass
9	Count-the-1's	0.78557626	Pass
10	Count-the-1's 2	0.90936671	Pass
11	Parking Lot	0.61714448	Pass
12	Minimum Distance	0.85584412	Pass
13	3D Spheres	0.16620345	Pass
14	Squeeze	0.93467455	Pass
15	Overlapping Sums	0.14856887	Pass
16	Runs	0.83756616	Pass
17	Craps	0.88105480	Pass

the p-values obtained, it is also observed that the test result is "pass" for all the DIE-HARD tests. The performance of LCC has been compared with the existing algorithms such as DES and 3-DES and PCA-based existing cipher [25]. The comparative analysis in term of randomness, size and runtime is shown in Figure 10 and Figure 11. It can be noted that the Z-value obtained considering standard normal distribution curve as reference is lesser in case of LCC. It signifies that the proposed method is capable of generating true random sequences. It can also be observed from Figure 10 that the runtime is also lesser in case of LCC which signifies that this can be implemented for the resource-constrained sensor devices.

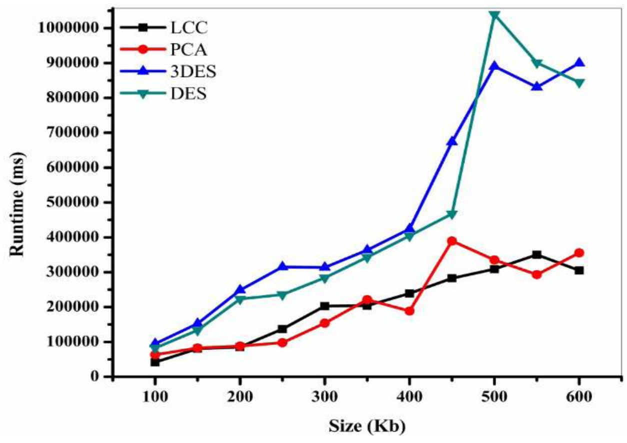

FIGURE 10. Runtime with increase in size.

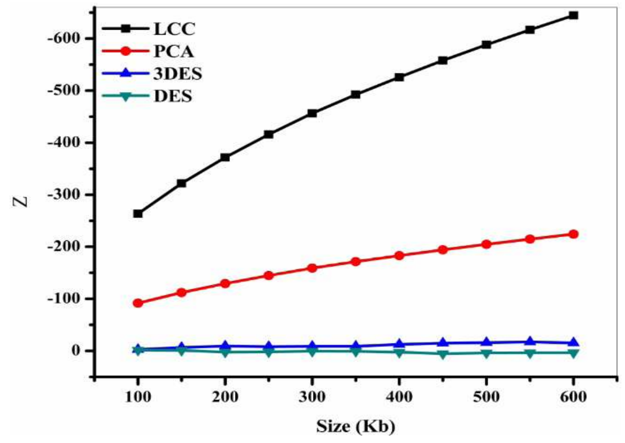

FIGURE 11. Standard normal distribution (Z) with increase in size.

V. CONCLUSION

This paper presents an efficient and lightweight method of encryption technique that can be used in IoT applications. This method is a symmetric key encryption technique. It has efficiently handled the limitations of IoT nodes deployed at perception layer. Due to its inherent simplicity yet ability of creating chaotic sequence has served the purpose. LCC can be used in any type of sensor nodes. However, symmetric key encryption technique suffers from key management complexities. So, this method is no exception regarding key management. By using some efficient key management strategies, LCC can be successfully implemented to prevent data theft from the communication channel between perception layer and network layer. LCC can also be embedded during fabrication of sensors and thus it would make those sensors more secure. Besides, the algorithms are capable of implemented in parallel programming paradigm, which would reduce the runtime more.

ACKNOWLEDGMENT

The initial idea of this work was carried out by all the three authors at Manipal University Jaipur, India. Further the work (in-terms of implementations for simulation and experimental results) was conducted at Manipal University Jaipur, India by the first author as a part of his Ph.D. work guided by the second author. The third author constantly contributed in this work and the work was finalized for submission while he was associated with TCS Research & Innovations.

REFERENCES

[1] IOT Analytics. (2014). *Internet of Things: Definition, History, Disambiguation*. [Online]. Available: https://iot-analytics.com/Internet-of-things-definition/

[2] A. P. I. Saif and S. Peasley. (2015). *Safeguarding the Internet of Things: Being Secure, Vigilant, and Resilient in the Connected Age*. [Online]. Available: https://dupress.deloitte.com/dup-us-en/deloitte-review/issue-17/Internet-of-things-data-security-andprivacy.html

[3] M. Rouse. (2013). *IoT Security (Internet of Things Security)*. [Online]. Available: http://internetofthingsagenda.techtarget.com/definition/IoT-security-Internet-of-Things-security

[4] B. Lam and C. Larose. (2016). *How Did the Internet of Things Allow the Latest Attack on the Internet?*. [Online]. Available: https://www.privacyandsecuritymatters.com/2016/10/howdid-the-Internet-of-things-allow-the-latest-attack-on-the-Internet/

[5] T. Cloud. (2013). *IoT Past and Present: The History of IoT, and Where It's Headed Today*. [Online]. Available: http://talkincloud.com/cloud-computing/iot-past-and-present-historyiot-and-where-its-headed-oday?page=2

[6] J. Granjal, E. Monteiro, and J. S. Silva, "A secure interconnection model for IPv6 enabled wireless sensor networks," in *Proc. IFIP Wireless Days*, Oct. 2010, pp. 1–6.

[7] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in Internet of Things: The road ahead," *Comput. Netw.*, vol. 76, pp. 146–164, Jan. 2015.

[8] H. Cui, X. Yi, and S. Nepal, "Achieving scalable access control over encrypted data for edge computing networks," *IEEE Access*, vol. 6, pp. 30049–30059, 2018.

[9] N.-N. Dao, Y. Kim, S. Jeong, M. Park, and S. Cho, "Achievable multi-security levels for lightweight IoT-enabled devices in infrastructureless peer-aware communications," *IEEE Access*, vol. 5, pp. 26743–26753, 2017.

[10] Q. Zhou, M. Elbadry, F. Ye, and Y. Yang, "Flexible, fine grained access control for Internet of Things: Poster abstract," in *Proc. 2nd Int. Conf. Internet-of-Things Design Implement. (IoTDI)*, New York, NY, USA, 2017, pp. 333–334. [Online]. Available: http://doi.acm.org/10.1145/3054977.3057308

[11] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed Internet of Things," *Comput. Netw.*, vol. 57, no. 10, pp. 2266–2279, 2013.

[12] K. J. Kumar, K. C. K. Reddy, and S. Salivahanan, "Novel and efficient cellular automata based symmetric key encryption algorithm for wireless sensor networks," *Int. J. Comput. Appl.*, vol. 13, no. 4, pp. 30–37, 2011.

[13] P. Anghelescu, E. Sofron, C.-I. Rincu, and V.-G. Iana, "Programmable cellular automata based encryption algorithm," in *Proc. Int. Semiconductor Conf.*, vol. 2, Oct. 2008, pp. 351–354.

[14] M. Henricksen, "A critique of some chaotic-map and cellular automata-based stream ciphers," in *Advances in Computer Science—ASIAN 2009. Information Security and Privacy*, A. Datta, Ed. Berlin, Germany: Springer, 2009, pp. 69–78.

[15] W. Trappe, R. Howard, and R. S. Moore, "Low-energy security: Limits and opportunities in the Internet of Things," *IEEE Security Privacy*, vol. 13, no. 1, pp. 14–21, Jan. 2015.

[16] Y. Yang, L. Wu, G. Yin, L. Li, and H. Zhao, "A survey on security and privacy issues in Internet-of-Things," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1250–1258, Oct. 2017.

[17] H. Shafagh, A. Hithnawi, A. Droescher, S. Duquennoy, and W. Hu, "Poster: Towards encrypted query processing for the Internet of Things," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, New York, NY, USA, 2015, pp. 251–253. [Online]. Available: http://doi.acm.org/10.1145/2789168.2795172

[18] R. Kotamsetty and M. Govindarasu, "Adaptive latency-aware query processing on encrypted data for the Internet of Things," in *Proc. 25th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Aug. 2016, pp. 1–7.

[19] S. Al Salami, J. Baek, K. Salah, and E. Damiani, "Lightweight encryption for smart home," in *Proc. 11th Int. Conf. Availability, Rel. Secur. (ARES)*, Aug./Sep. 2016, pp. 382–388.

[20] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.

[21] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.

[22] J. von Neumann, *Theory Of Self Reproducing Automata*, A. W. Burks, Ed. Champaign, IL, USA: University of Illinois Press, 1966.

[23] S. Ulam, "Random processes and transformations," in *Proc. Int. Congr. Math.*, vol. 2, 1950, pp. 264–275.

[24] S. Roy, S. Nandi, J. Dansana, and P. K. Pattnaik, "Application of cellular automata in symmetric key cryptography," in *Proc. Int. Conf. Commun. Signal Process.*, Apr. 2014, pp. 572–576.

[25] S. Roy, N. Bhatia, and U. S. Rawat, "A novel cryptosystem using cellular automata," in *Proc. Int. Conf. Commun. Signal Process. (ICCSP)*, Apr. 2017, pp. 1781–1785.

[26] S. Roy, J. Karjee, U. S. Rawat, N. D. Pratik, and N. Dey, "Symmetric key encryption technique: A cellular automata based approach in wireless sensor networks," *Procedia Comput. Sci.*, vol. 78, pp. 408–414, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050916000843

[27] S. Nandi, B. K. Kar, and P. P. Chaudhuri, "Theory and applications of cellular automata in cryptography," *IEEE Trans. Comput.*, vol. 43, no. 12, pp. 1346–1357, Dec. 1994.

[28] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proc. 33rd Annu. Hawaii Int. Conf. System Sci.*, vol. 2, Jan. 2000, p. 10.

[29] J. Karjee and H. S. Jamadagni. (2011). "Data accuracy model for distributed clustering algorithm based on spatial data correlation in wireless sensor networks." [Online]. Available: https://arxiv.org/abs/1108.2644

[30] *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. Accessed: Nov. 15, 2018. [Online]. Available: https://www.nist.gov/publications/statistical-test-suite-random-and-pseudorandom-number-generators-cryptographic

[31] G. Marsaglia. (1996). *DIEHARD*. [Online]. Available: http://stat.fsu.edu/~geo/diehard.html

SATYABRATA ROY received the B.Tech. and M.Tech. degrees in CSE from the Maulana Abul Kalam Azad University of Technology, West Bengal (Formerly known as West Bengal University of Technology), and KIIT University, Bhubaneswar, respectively. He is currently pursuing the Ph.D. degree in the area of lightweight security with Manipal University Jaipur. He is having eight years of teaching experience. He is currently an Assistant Professor with the Department of Computer Science and Engineering, Manipal University Jaipur. His major research interests include cryptography and cellular automata. He has published many research papers in referred journals and international conferences.

UMASHANKAR RAWAT received the Ph.D. degree in computer science and engineering from the Jaypee University of Engineering and Technology, Guna, in the area of linux file system security, in 2013, and the M.E. degree in computer engineering from Shri Govindram Seksaria Institute of Technology and Science, Indore, in 2003. He is having 18 years of teaching experience. He is currently a Professor with the Department of CSE, Manipal University Jaipur. His current research interests include information systems security and distributed systems.

JYOTIRMOY KARJEE received the B.E. degree in electronics, and the M.E. degree in IT from Swami Ramanand Teerth Marathwada University and the Maulana Abul Kalam Azad University of Technology, West Bengal (Formerly known as West Bengal University of Technology), and the Ph.D. degree in engineering from the Indian Institute of Science, Bengaluru, India. He did his postdoctoral research with the Technical University of Munich, Germany. He was a Scientist with the TCS Research and Innovation, Bengaluru. Prior to TCS, he was with Manipal University Jaipur, as an Associate Professor (2015–2016), and held a position as an Assistant Professor (2015). He was also associated with the Sikkim Manipal Institute of Technology as a Lecturer (2006–2008). Moreover, he also worked in industry as a Data Scientist with Enlightiks, in 2014. He worked as a Software Trainee Engineer with Prakriti Inbound Pvt. Ltd., in 2003 and 2005, respectively. He has published several papers in reputed journals/conferences and filed many patents. He is interested in statistical signal processing, wireless sensor networks, wireless communications, network security, machine learning, and the Internet of Things. He is a member of the ACM. He is a recipient of the Heritage Erasmus Mundus Scholarship (Fellowship) to undertake his postdoctoral research.