Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 26 (2023), No. 1, pp. 153–173
DOI : 10.1080/09720529.2021.1932889

Dependency of lightweight block ciphers over S-boxes : A deep learning based analysis

Girish Mishra *
Scientific Analysis Group
Defence R & D Organization
New Delhi 110054
India

and

Defence Institute of Advanced Technology (DU)
Pune 411025
Maharashtra
India

S. V. S. S. N. V. G. Krishna Murthy
Defence Institute of Advanced Technology (DU)
Pune 411025
Maharashtra
India

S. K. Pal
Scientific Analysis Group
Defence R & D Organization
New Delhi 110054
India

Abstract

Lightweight ciphers have been proposed from time to time to tackle various usage issues in IoT (Internet of Things) scenario. PRESENT and DoT are two such ciphers. PRESENT has been a benchmark cipher to establish and validate the security features of any new lightweight block cipher. This paper presents deep learning based analysis of a widely known lightweight cipher PRESENT and relatively less known lightweight cipher

*E-mail: gmishratech28@gmail.com

DoT. Our observations show that both the ciphers are extremely sensitive to any intentional or unintentional error occurring in their S-boxes. In our experiments, these two ciphers maintain the desired immunity to such errors when either of the randomly generated ten non-linear S-boxes are used in their corresponding algorithms. However, the ciphers fail to withstand our deep learning based attack when either no S-box is used or a reverse S-box is used in their algorithms. This work validates the fact that S-box should always be carefully chosen while designing a block cipher. The observations from our work suggest the designers to be observant towards vulnerabilities arising out of any intentional fault attacks from the adversary.

Subject Classification: *94A60, 68P25.*

Keywords: *Deep learning, Lightweight cipher, PRESENT, DoT.*

1. Introduction

The age of IoT is witnessing expanding demand of compact electronic devices with some cryptographic application inside. This has prompted crypto researchers to propose lightweight ciphers providing good efficiency without compromising with the security of private data. The need for any new block cipher has been non-existent after establishment of Advance Encryption Standard (AES) as a block cipher which has rigorously been studied and analyzed by the crypto community since its standardization by NIST in 1997. But, unsuitability of AES in resource constrained devices such as sensor networks and RFID tags, has encouraged researchers to explore designing lightweight block ciphers. PRESENT is one of these ciphers which has ultra-lightweight design appropriate to resource constrained hardware environments. It was introduced by Bogdanov et al. in 2007 [1]. DoT is also a lightweight encryption design introduced by

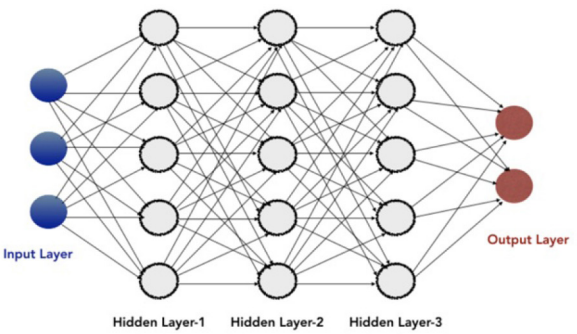

Figure 1

Fundamental deep learning architecture

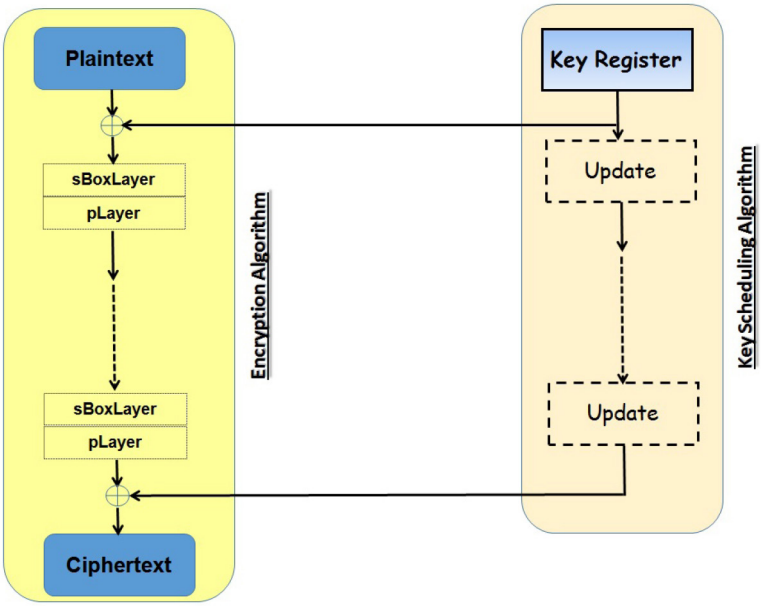

Figure 2

Basic layout of PRESENT cipher

Patil et al [2] which is based on SPN (substitution permutation network) structure.

Deep learning under the umbrella of machine learning methods has emerged as a robust extension of artificial neural networks (ANN). In other words, deep learning is a class of machine learning algorithms [3] that uses multiple hidden layers to evolve higher level features from the raw input data progressively with the objective of solving the regression or classification problems. Deep learning (DL), also invariably known as deep neural networks, comprises of several types of architectures such as Convolution Neural Network (CNN), Recurrent Neural Network (RNN), Deep Belief Network (DBN), etc. The usage of these architectures has been explored in diversified areas such as machine translation, natural language processing, computer vision, medical image processing, etc. [4] [5]. Bhattacharya et al. [6], recently in their work, summarized the state-of-the-art scientific studies linked with the applications of deep learning in COVID-19 medical image processing and discussed the issues and challenges associated with it. Dubey and Jain [7] illustrated the significance of different deep learning models in the computer vision domain and

explored them for the task of face recognition in images and videos. They also compared several contemporary algorithms for the same task. The work by Popel et al. [4] applied deep learning methods to enhance the standard of human translation and in certain circumstances, they suggest that deep learning may replace humans in machine translation applications where keeping the meaning intact is the primary goal. Gupta et al. [8] used deep learning approaches for the recognition of human emotion in speech data. The authors performed extensive comparative analysis depending upon the mathematical and statistical observations. Apart from these applications, the prospects of DL networks in cryptographic applications have recently gained good attention of researchers [9] [10]. Deep learning is superior to traditional machine learning algorithms due to its ability to compose the features from raw data of the initial layer. In subsequent layers these features are blended together in forming the higher level features for its further usage in final decision making. This superiority of DL networks is very noteworthy as feature extraction, and further scanning of it for feature selection in conventional machine learning methods involves a hefty process.

The cryptanalysts [9] [10] have recently made an intensive use of deep learning. These authors, contrary to conventional wisdom, have claimed to produce a very powerful deep learning based cryptographic distinguishers. Before this, the crypto researchers have traversed the usage of machine learning to cryptography domain in a very confined manner. Rivest, in his survey paper [11] in 1991, highlighted the contributed ideas and techniques between cryptography and machine learning. He visualized machine learning and cryptanalysis as sister fields as the both domains share the common goal of learning an unknown function by observing input-output behavior.

Abadi and Andersen [12] in 2016, presented a milestone work by claiming that in a multiagent system, the neural networks can learn to use secret keys in order to protect the information from other adversarial neural networks. Ehsan et al. [13] in 2017, addressed the issue of privacy for client's own raw data and developed the techniques in 2017 for providing solutions to run deep neural networks over encrypted data. In 2018, Stjepan Picek et al. [14] analyzed convolutional neural networks for side-channel analysis. Wang [15] carried out side-channel analysis of block cipher AES based on deep learning in 2019. Gohr [9] in 2019, proposed improved attacks on round-reduced lightweight block cipher Speck32/64 . Recently in 2020, Baski et al [16] simulated the all-in-one differentials for non-Markov ciphers through deep learning.

Our Contribution

Our contribution in this paper is to show how the absence of s-box or using any S-box without much analysis may adversely affect the block ciphers. This observation comes out from our experiments on two lightweight ciphers namely PRESENT and DoT. Both the lightweight block ciphers do not withstand either the absence of S-box or the use of a reverse S-box. Their algorithms become vulnerable in either of the scenarios. By reverse S-box, we mean that for each element begining from the start is mapped to the element starting from the end towards the beginning. For example, for a 4 x 4 reverse S-box, '0' is mapped to 'F', '1' is mapped to 'E',...,'F' is mapped to '0'. All elements here are expressed in hex representation. In our experiment, the first objective is to see if the individual plaintext bits could be rightly predicted from the available corresponding ciphertext data in known-plaintext attack mode. To show this, a large number of plaintext-ciphertext pairs are generated for a fixed key and then the problem is to predict a particular plaintext bit from a new ciphertext data generated using the same key. Likewise, the prediction of all individual plaintext bits is done from this ciphertext data. For the first objective, no tempering is done with the algorithms of PRESENT and DoT ciphers and their own S-box has been used. Then, the second objective is doing the same experiment when the S-box from both the ciphers are completely removed (i.e. S-box does not exist in the algorithm). Apart from this, the similar experiments have been done for randomly selected ten different S-boxes and the reverse S-box. In the first experiment, we could not predict the plaintext bits with better than chance probability. In later experiment, both algorithms suffered badly with the absence of their corresponding S-boxes and plaintext bits could be predicted with significantly much higher probability than the chance probability of 0.5. The chance probability in case of tossing a coin means the occurrence of head or tail is equally probable that is with probability ½. The experimentation for fulfilling the two objectives has been carried out using deep learning approach which is explained in next sections.

The paper is organised as follows: Preliminaries are given in Section 2, which briefly describes the both lightweight ciphers and deep learning methods. Section 3 explains our strategy for exploring biases in prediction of individual plaintext bits from available ciphertext data in known-plaintext attack model. Section 4 discusses the results and analysis for the same. Finally, we conclude the paper in Section 5.

2. Preliminaries

In this section, basic deep learning details necessary for comprehending the proposed method are discussed. Brief design descriptions of lightweight block ciphers are also provided here in the section.

2.1 *Description of Deep Learning Methods*

Deep learning networks are inspired from the functioning of the human brain. So, the deep learning arrangement basically consists of hierarchical learning structures. These hierarchical structures are generally mix of linear component (weighted sum of input neurons) and non-linear component (activation function over weighted sum and bias). The presence of non-linearity is essential so as to deal with real life problems.

The advent of deep learning methods have seen the wide-spread applications due to their advantage of extracting features from the raw data themselves which have not been feasible in traditional machine learning methods. The deep learning networks use multiple intermediate hidden layers between input and output for extracting higher level features from the abstract raw input data. In image processing problems, initial intermediate layers identify the edges present in the image data, whereas the subsequent intermediate layers determine more meaningful features such as digits, letters, faces or motions. A common deep learning architecture with one input layer, followed by several hidden layers and then by the output layer is shown in Figure 1.

Some recent developments during last few years, such as the availability of advance GPU (Graphical Processing Unit) and TPU (Tensor Processing Unit) machines by Google [17] for faster computations and use of the activation function ReLU (Rectified Linear Unit) as an alternative to the classical sigmoid activation function, has facilitated stacking many hidden layers in the network structure [18]. These two activation functions are defined as:

$f(x) = max(0, x)$ **(ReLU activation function)**

$g(x) = 1/1 + e^{-x}$ **(Sigmoid activation function)**

The input layer holds the unprocessed raw input. The intermediate layers sum up the weighted inputs from the previous layer to apply ReLU activation function subsequently. Then, the output layer uses Softmax

function for completing the classification/prediction task. Softmax function is as given below:

$$F(X_i) = \frac{e^{X_i}}{\sum_{i=0}^{k} e^{X_i}} \text{ where } i = 0, 1, 2, ..., k$$

k denotes the number of possible outputs. The multiple intermediate layers between input and output layers enable the networks in learning the abstract mapping from input data to the corresponding output target data. In broad terms, neural networks with multiple hidden layers are known as Deep Learning. Some well known deep learning techniques are Convolution Neural Networks, Recurrent Neural Networks and Long Short Term Memory networks [19].

2.2 *Brief Description of Lightweight Cipher PRESENT*

PRESENT is an ultra-lightweight cipher designed by Bogdanov et al [1] in 2007 specifically for extremely constrained environments such as RFID tags and sensor networks. PRESENT is an SP-network design and consists of 31 rounds. The designers have provisioned the support for 80 and 128-bit key lengths while the block length is of 64 bits. The cipher scheme is depicted in Figure 2 [20].

As mentioned earlier, PRESENT consists of 31 rounds and each of which uses addRoundKey (an XOR operation with a round key K_i for $1 \le i \le 31$), pLayer(linear bitwise permutation), and sboxLayer (non-linear substitution layer). The post-whitening at the end is done using the round key K_{32}. The sboxLayer uses a 4 x 4 S-box applied 16 times in parallel in each round. S-box used in sboxLayer and permutation used in pLayer are given in Table 1 and Table 2 respectively.

Table 1

Table of S-box in PRESENT cipher [1]

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Table 2

Table of Permutation in PRESENT cipher [1]

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P(i)	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
P(i)	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
P(i)	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
P(i)	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

Algorithm 1 ***(Encryption Algorithm of PRESENT Cipher)***

1. **Input:** $P = P_1^{64}$ and Round Keys RK_i obtained from Master Key K
2. **Output:** $C = P_{32}^{64} \oplus RK_{32}$
3. for i = 1 to 31
4. $U^{64} = P_i^{64} \oplus RK_i$
5. $U^{64} \rightarrow (N_1 \| N_2 \| \ldots \| N_{16})$
6. $U^{64} = (S_B(N_1) \| S_B(N_2) \| \ldots \| S_B(N_{16}))$
7. $P_{i+1}^{64} = P(U^{64})$
8. end for

Algorithmic view of crypto algorithm of PRESENT is shown in Algorithm 1. The detailed description for key scheduling algorithm is provided in [1].

2.3 *Brief Description of Lightweight Cipher DoT*

DoT is a relatively less known and new lightweight cipher than PRESENT. This lightweight block cipher design was published by Patil et al. in 2019 [2]. It has Substitution Permutation Network (SPN) structure based basic framework, which is designed for 64-bit block size and 128-bit key size. DoT is based on SPN structure that processes its complete plaintext input into 64/80 bit blocks. Each block goes through its round function in each of 31 rounds. Round function comprises of a 4-bit S-box which is applied sixteen times in parallel, however its diffusion layer is

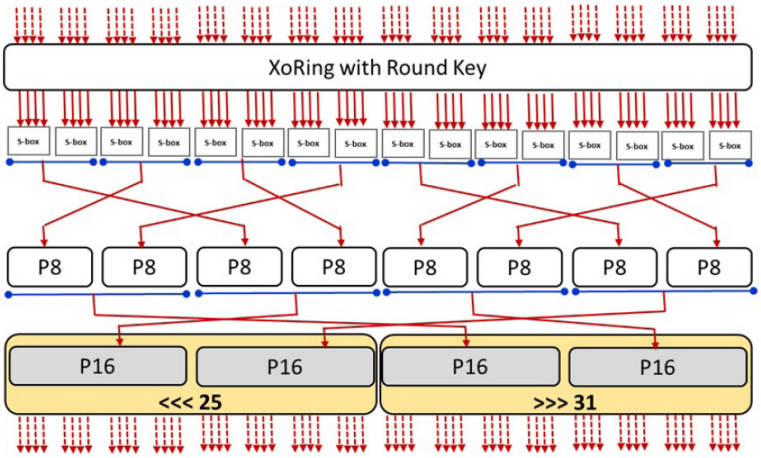

Figure 3

Basic layout of DoT cipher

composed of several components like byte/word shift, bit/nibble/byte/word permutation and circular shift operations.

The broad layout of round function of DoT is shown in Figure 3. S-box(S_B) and byte permutation ($P8$) used in DoT cipher are shown in Table 3 and Table 4 respectively. First, the input key is XoRed with Round Key. Then, each of 16 nibbles of the resulting data passes through the S-box in parallel. After this, byte permutation ($P8$) is applied to construct next state of data. Now, the permutation is applied on four 16-bit words to create 64-bit data. This 64-bit data is then split into two words each of 32-bit size. First 32-bit word is left rotated within itself by 25 bit position whereas second 32-bit word is rotated right within itself by 31 bit position to finally provide the output data of one round.

The algortihmic structure of crypto algorithm as explained in [2] [21] is discussed in Algoritm 2.. These papers ([2] [21]) may be referred for the detailed description of key scheduling algorithm.

Algorithm 2 ***(Encryption Algorithm of DoT Cipher)***

1. **Input:** $P = P_1^{64}$ and Round Keys RK_i obtained from Master Key K
2. **Output:** $C = P_{32}^{64} \oplus RK_{32}$
3. for i = 1 to 31 do
4. $U^{64} = P8(S_B(P_i^{64} \oplus RK_i))$

5. $U^{64} \rightarrow (B_1 \| B_2 \| B_3 \| B_4 \| B_5 \| B_6 \| B_7 \| B_8)$
6. $U_L^{32} = (B_1 \| B_2 \| B_3 \| B_4)$
7. $U_R^{32} = (B_5 \| B_6 \| B_7 \| B_8)$
8. $P_{i+1}^{64} = (U_L^{32} <<< 25) \| (U_R^{32} >>> 31)$
9. end for

Table 3

Table of S-box in DoT cipher [2]

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	3	F	E	1	0	A	5	8	C	4	B	2	9	7	6	D

Table 4

Table of Permutation (P8) in DoT cipher [2]

i	0	1	2	3	4	5	6	7
P8(i)	2	4	6	0	7	1	3	5

2.4 *Pre-requisite for Crypto Components Used in S-P Network*

Substitution-permutation network (SPN) is a type of iterated block cipher, which takes a block of the plaintext and a key as inputs, and applies several rounds of substitution stage followed by a permutation stage in order to produce one block of ciphertext output [22]. The non-linear substitution stage is firstly applied on mixing of the key bits and the plaintext bits to bring into the fundamental principle of Shannon's confusion [23] into the design of any conventional cryptographic systems. Confusion aims to conceal the presence of any algebraic structure in the system. Thereafter, the linear permutation stage dissipates the redundancies to establish Shannon's diffusion in architecture of the network [23].

Simply looking at the encryption process in S-P network, it is clear that an encryption function maps a combination of the plaintext and the key to a ciphertext. In other words, any ciphertext is function of a plaintext and a key. Now, designing any block cipher should ideally focus on removal of any footprint or hint of both plaintext and key. This requirement is established by confusion and diffusion as mentioned by Shannon in his paper [23]. In S-P network based block ciphers, a cryptographically strong S-box is used as the non-linear component. In principle, an S-box could be

linear as well as non-linear. However, it should be non-linear with high degree from cryptographic point of view for introducing strong confusion. The purpose of S-box with the basic philosophy of bringing confusion is quantified by several nonlineariy criteria [24]. A yardstick for measuring nonlinearity of an $(n,m)-$function $F:F_2^n \mapsto F_2^m$ is given by [25] [24]:

$$\text{nonlinearity } (F) = 2^{n-1} - \frac{1}{2} \max_{v \in F_2^{m^*};\, w \in F_2^n} \left| \sum_{x \in F_2^n} (-1)^{v.F(x)+w.x} \right|;$$

where $u.v$ denotes the usual inner product of u and v.

Saarinen [26] performed cryptographic analysis of all 4×4 S-boxes. In layman's words, for a 4 x 4 S-box, if x_1, x_2, x_3, x_4 are input bits and y_1, y_2, y_3, y_4 are output bits, then an S-box, having all its component Boolean functions of the form $y_i = a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4$ (where $i = 1,2,3,4$ and a_i are 0 or 1), is a linear S-box. An S-box having non-linear terms (such as $x_1x_2x_3$ or x_2x_4) in expression of its component Boolean functions would be a non-linear s-box. However, Saarinen [26] carried out an extensive analysis to exactly define these desciptions.

Generally, being the only non-linear component in S-P network, it is mandatory to do an intensive analysis in choosing a cryptographically strong S-box. There are several design criteria [27] for generating cryptographically sound s-boxes such as Bijection, Strict avalanche criterion, Bit independence criteria, Non-linearity and Balancedness. Although all the criteria can not be met simultaneously to their maximum effects, but achieving all of them to an optimum level could be done by testing each component Boolean functions against each of the other criteria. Bijection requires a one-to-one mapping from input vectors to output vectors for any $n \times n$ S-box. Strict avalanche criterion means if one input bit is changed, approximately 50 % output bits must be changed. Bit independence criterion requires that output bits act independently from each other. Non-linearity assures that the S-box is not a linear mapping from the input to the output. Balancedness means that each Boolean vector responsible in generating the S-box has the same number of 0s and 1s in its truth table.

In S-P network, a permutation box (P-box) is used after S-box to feed the bits further into the S-boxes of next round. A good permutation function should ensure the distribution of output bits from an S-box in one round to as many S-box inputs as possible in the next round. For this, it should not have any fixed points in its expression.

2.5 *Deep Learning for Analysis of Crypto algorithms*

Having pseudo random characteristics and being without any patterns in some data set, may frustrate any machine learning algorithm in decision making. This is the problem that a well designed crypto algorithm may pose to deep learning methods. The main objective in designing any crypto algorithm is ensuring randomness in key stream and ciphertext so that adversaries may not be smart enough in exploiting any visible arrangement within the data. So, the designers analyse their crypto algorithms thoroughly by applying various mathematical and statistical tests [28] in order to kill chances of availability of any usable patterns. But sometimes even after checking the designs comprehensively, the human made crypto algorithm may leave some clues to adversaries. So, deep learning methods may search for these clues and sometimes be fortunate enough in bringing forward the same for its further use to mount cryptanalytic attacks on the algorithm.

3. Method Adopted for Analysis of Lightweight Block Ciphers

In this section, we explain the approach used in our work which primarily uses deep learning network. A simple layout of this approach is shown in Figure 4. We adopt this approach for analysis of both the lightweight ciphers with target of predicting the individual bits of the 64-bit plaintext (i.e. i^{th} bit of the 64-bit plaintext, where $i = 0,1,\ldots,63$) from its corresponding ciphertext. This analysis is done in known-plaintext attack model. A pointwise description of the method is as follows:

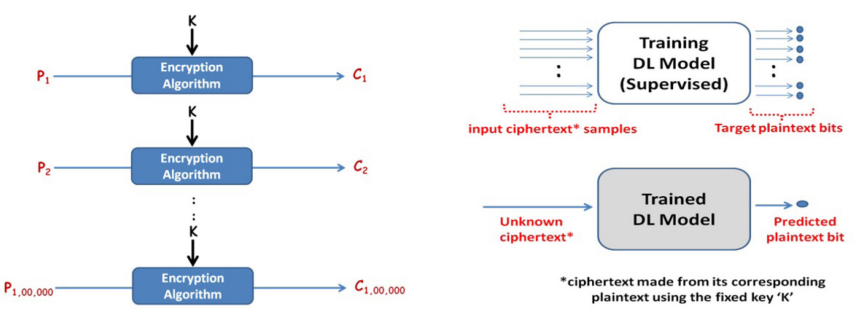

Figure 4

Simple layout of the approach

Pointwise description of the adopted method

Let us consider a block cipher with Key *K*. Let *P*, *C* and RK_i denote the Plaintext, the Ciphertext and the i^{th} Round Key respectively.

1. Let $P = p_0 p_1 \ldots p_n$ be an n-bit plaintext; where each p_i denotes an individual bit of the plaintext.
2. Taking *K* as master key, run key scheduling algorithm to generate the round keys RK_i where $i = 0, 1, \ldots, r$ and *r* denotes the number of rounds used in the block cipher.
3. Using round keys of Step-2, run the encryption algorithm over plaintext P to generate the ciphertext $C = c_1 c_2 \ldots c_n$.
4. Iterate Step-1, Step-2 and Step-3 *N* number of times for *N* different values of *P* to generate $"P - C"$ pairs i.e. $"Plaintext - Ciphertext"$ pairs.
5. Create a dataset consisting of these *N* pairs.
6. Divide dataset into two parts namely Training dataset and Test dataset having 80% and remaining 20% of above mentioned *N* pairs respectively.
7. Pass the dataset to deep learning network by considering Ciphertext *C* as input and p_0 bit of the corresponding Plaintext *P* as the target.
8. Train the network using Training dataset to create a deep learning model.
9. Validate the model on Test dataset and collect the result which shows the prediction probability of model for p_0 bit of *P*.
10. Repeat the process mentioned in steps from 7 to 9 to experiment with $p_1, p_2, \ldots, p_n$ bits of *P*.
11. i^{th} bit of plaintext *P* is biased If Prediction Probability (for i^{th} bit of *P*) is significantly higher than 1/2 (0.5), else it is random.

The primary objective of the work is to observe how much the absence of S-box or use of any other random S-box without requisite analysis affects two ciphers. The methodology adopted, to fulfil this objective, is to observe whether any of the individual bits of n-bit plaintext could be predicted from a given ciphertext with better than chance probability in known-plaintext attack set-up. In both ciphers, we experimented with crypto algorithm under following four categories:

1. Original S-box is used.

Randomly generated S-box1

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	5	8	9	C	0	4	F	2	D	3	1	B	E	7	A	6

Randomly generated S-box2

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	7	0	1	9	6	B	F	C	A	5	3	8	E	4	2	D

Randomly generated S-box3

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	4	B	E	0	5	8	D	7	3	9	F	2	A	6	C	1

Randomly generated S-box4

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	3	A	E	8	2	1	0	9	C	7	D	F	6	4	B	5

Randomly generated S-box5

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	3	A	8	5	B	E	7	0	D	9	4	C	2	F	1	6

Randomly generated S-box6

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	D	1	4	5	0	8	7	A	B	9	6	2	3	F	C	E

Randomly generated S-box7

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	C	A	9	5	3	F	7	1	6	B	D	8	0	2	4	E

Randomly generated S-box8

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	9	D	3	E	F	8	5	7	0	A	4	B	C	1	2	6

Randomly generated S-box9

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	A	E	1	9	D	C	0	2	B	5	8	4	7	3	6	F

Randomly generated S-box10

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	4	0	3	9	6	5	D	2	8	7	B	E	F	A	1	C

Identity S-box

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Reverse S-box

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

Original S-box of PRESENT Cipher

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Original S-box of DoT Cipher

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(x)	3	F	E	1	0	A	5	8	C	4	B	2	9	7	6	D

Figure 5

All S-boxes used in our experiment

2. Ten different randomly generated S-boxes $(S-box1, S-box2, \ldots, S-box10)$ are used one by one.
3. No S-box is used (i.e. *Identity S-box* is used where each element of S-box gets mapped to itself).
4. *Reverse S-box* is used, where Reverse S-box is the S-box in which each element starting from the begining is mapped to the element from the end in reverse order.

It is easily understood that Identity S-box is equivalent to using no S-box or absence of the S-box in the crypto algorithm. All S-boxes mentioned in four categories are shown in Figure 5.

The deep learning arrangement employed in carrying out our experiments contains two dense layers with each layer having 10 neurons and *ReLU* activation function, followed by an output layer accompanied with 256 neurons and Softmax activation function. The model created with this arrangement is compiled with *adam* optimizer and *categorical_crossentropy* as loss function [3]. Training is done in only three epochs and after these many epochs, the results are collected for training and validation data. Similar prediction models are also created for predicting each of the 64 bits of the plaintext. The deep learning framework engaged in our work is shown in Figure 6.

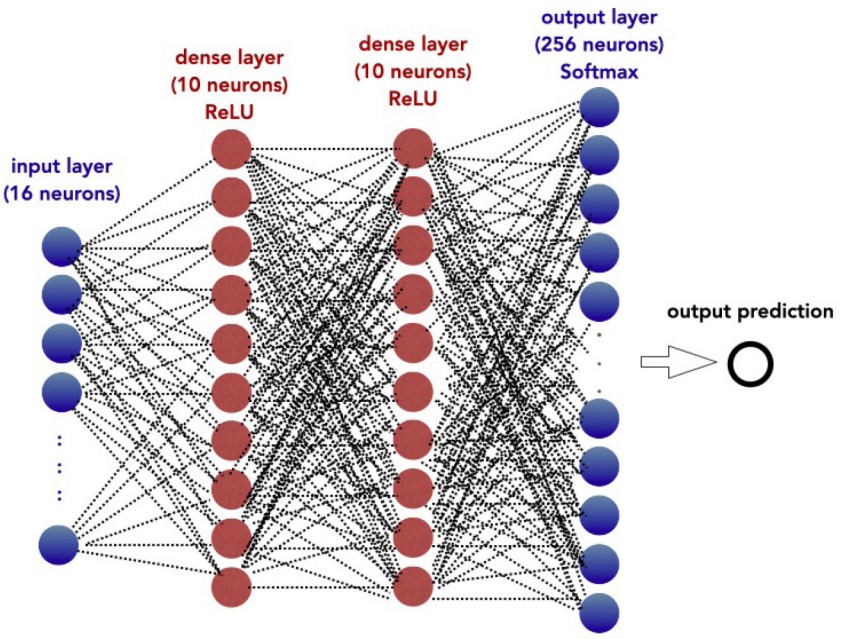

Figure 6

Deep Learning arrangement used in the experiment

4. Results and Analysis

On lines of the description as mentioned in Section 3, we carried out the experiment by collecting 1,00,000 "*Plaintext – Ciphertext*" pairs. Each ciphertext of the pair is generated from its corresponding plaintext using a fixed key. The results in all four categories of the experiments are shown in Figure 7 and Figure 8.

From the results, as shown in these figures, it is evidently visible that both PRESENT cipher and DoT cipher possess the existential biases in prediction of the individual plaintext bits from an unknown ciphertext when either no S-box or reverse S-box is used in their respective crypto algorithms. This is again reiterated that the experiments are carried out in known-plaintext attack set-up. This clearly means that a robust confusion characteristic, which is ensured by presence of a strong S-box in the algorithm, is very essential along with good diffusion property in any block cipher. However, results also show that, our deep learning approach has not been able to find anything meaningful in both the ciphers when we used their own original S-boxes or any of the randomly generated ten

Table 5
Properties of S-boxes used in our experiments

S. No.	S-box Used	Degree	Non-linearity	Balancedness (Yes/No)	Correlation Immunity
1.	S-box1	2	4	Yes	0
2.	S-box2	2	4	Yes	0
3.	S-box3	3	4	Yes	0
4.	S-box4	2	2	Yes	0
5.	S-box5	2	2	Yes	0
6.	S-box6	2	2	Yes	0
7.	S-box7	2	2	Yes	0
8.	S-box8	3	2	Yes	0
9.	S-box9	2	2	Yes	0
10.	S-box10	2	2	Yes	0
11.	S-box (PRESENT)	2	2	Yes	0
12.	S-box (DoT)	3	2	Yes	0
13.	Identity S-box	1	0	Yes	0
14.	Reverse S-box	1	0	Yes	0

S-boxes in the experiment. This finding evidently indicates that the design of an S-P network based block cipher algorithm should use a strong S-box to withstand any machine learning based attack. A rigorous testing of such S-box is firmly recommended. The observations from our experiments, clearly hint of a possible cryptanalytic attack on block ciphers in absence of a well analysed S-box in the algorithm.

Highlighting the reasons behind and the meaning of the results certainly brings some crucial facts and observations which may be helpful in cryptographic designs. Both the ciphers could not withstand our deep learning based attack in two cases when either Identity S-box or Reverse S-box were used in the respective crypto algorithms. These S-boxes are linear in nature as explained in Section 2 and shown in Table 5. This implies that there remains no non-linear component in crypto algorithm and therefore predicting individual bits of the plaintext from the corresponding unknown ciphertext becomes a linear problem. Although it is well known and fundamentally correct that no designer would almost certainly use a linear S-box but the fact to be cautious is that there could be some other category of weak S-boxes that may get

employed unintentionally without much analysis in a block cipher by some designers. Also, if the S-box is residing somewhere in the hardware component of cryptosystem, an expert and active attacker may target this part of the system in introducing the fault to convert a strong S-box into a weak S-box with very few manipulations. So, our work indirectly suggests that the designers must be careful enough towards safeguarding a practical cryptosystem from any such faults.

Our deep learning based approach could not exploit any weaknesses in case of using original S-boxes or randomly generated S-boxes in corresponding algorithms. The logic behind this fact is that these S-boxes are non-linear in nature. These component Boolean functions in these S-boxes are of degree 2 or 3 as shown in Table 5 which ensures that any particular plaintext bit is a non-linear function in ciphertext variables with having very high degree terms in its algebraic expression. Due to this, approximating the non-linear function towards a less-complex reduced function becomes a complicated task for any deep learning method. Therefore, the rate of predicting any individual bit of the plaintext from the corresponding ciphertext has a uniform distribution.

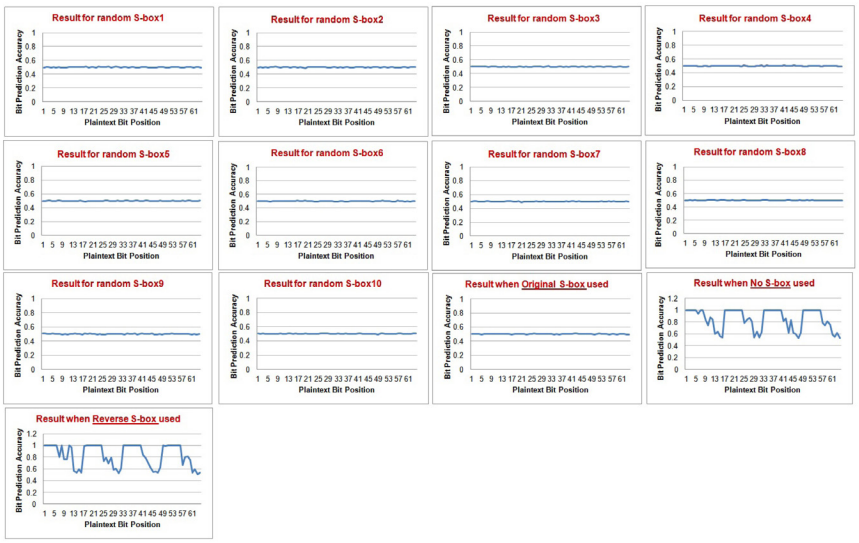

Figure 7

Prediction results for bits in 64-bit plaintext from unknown ciphertext in PRESENT cipher

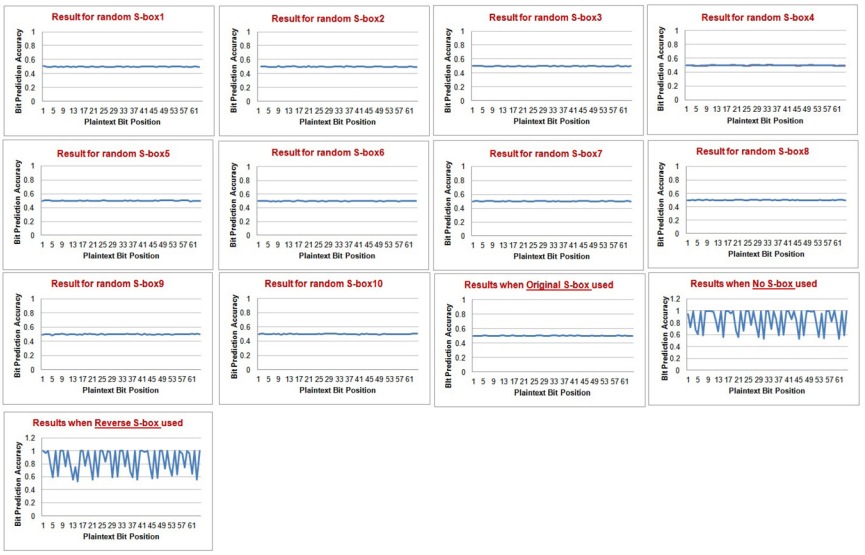

Figure 8

Prediction results for bits in 64-bit plaintext from unknown ciphertext in DoT cipher

5. Conclusion

In this paper, we have shown deep learning approach for analysing block ciphers. In particular, we have used this approach to do cryptanalysis of two lightweight block ciphers namely PRESENT and DoT. Through our experiments, we have aptly conveyed the importance of good confusion property in designing any block cipher. It has been shown that, a well-analysed S-box should be used in designing a block cipher so that it can resist any machine learning based cryptanalytic attack such as the attack mentioned in this paper. Further, it may safely be concluded that various machine learning algorithms can be explored to analyse different primitives used in the ciphers. This may come up with an additional mechanism apart from the existing ones to assess the strength of crypto algorithms.

References

[1] Andrey Bogdanov, Lars R Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and

Charlotte Vikkelsoe. Present: An ultra-lightweight block cipher. In *International workshop on cryptographic hardware and embedded systems*, pages 450–466. Springer, 2007.

[2] Jagdish Patil, Gaurav Bansod, and Kumar Shashi Kant. Dot: A new ultra-lightweight sp network encryption design for resource-constrained environment. In *Proceedings of the 2nd International Conference on Data Engineering and Communication Technology*, pages 249–257. Springer, 2019.

[3] Li Deng, Dong Yu, et al. Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4):197–387, 2014.

[4] Martin Popel, Marketa Tomkova, Jakub Tomek, Lukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtsky'. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature communications*, 11(1):1–15, 2020.

[5] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiong Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2722–2730, 2015.

[6] Sweta Bhattacharya, Praveen Kumar Reddy Maddikunta, Quoc-Viet Pham, Thippa Reddy Gadekallu, Chiranji Lal Chowdhary, Mamoun Alazab, Md Jalil Piran, et al. Deep learning and medical image processing for coronavirus (covid-19) pandemic: A survey. *Sustainable cities and society*, 65:102589, 2021.

[7] Arun Kumar Dubey and Vanita Jain. A review of face recognition methods using deep learning network. *Journal of Information and Optimization Sciences*, 40(2):547–558, 2019.

[8] Vedika Gupta, Stuti Juyal, Gurvinder Pal Singh, Chirag Killa, and Nishant Gupta. Emotion recognition of audio/speech data using deep learning approaches. *Journal of Information and Optimization Sciences*, 41(6):1309–1317, 2020.

[9] Aron Gohr. Improving attacks on round-reduced speck32/64 using deep learning. In Annual International Cryptology Conference, pages 150–179. Springer, 2019.

[10] Anubhab Baksi, Jakub Breier, Xiaoyang Dong, and Chen Yi. Machine learning assisted differential distinguishers for lightweight ciphers. *IACR Cryptol. ePrint Arch.*, 2020:571, 2020.

[11] Ronald L Rivest. Cryptography and machine learning. In *International Conference on the Theory and Application of Cryptology*, pages 427–439. Springer, 1991.

[12] Martín Abadi and David G Andersen. Learning to protect communications with adversarial neural cryptography. arXiv preprint arXiv:1610.06918, 2016.

[13] Ehsan Hesamifard, Hassan Takabi, and Mehdi Ghasemi. Cryptodl: Deep neural networks over encrypted data. arXiv preprint arXiv:1711.05189, 2017.

[14] Stjepan Picek, Ioannis Petros Samiotis, Jaehun Kim, Annelie Heuser, Shivam Bhasin, and Axel Legay. On the performance of convolutional neural networks for side-channel analysis. In *International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 157–176. Springer, 2018.

[15] Huanyu Wang. Side-channel analysis of aes based on deep learning, 2019.

[16] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The simon and speck families of lightweight block ciphers. IACR Cryptol. ePrint Arch., 2013:404, 2013.

[17] Anne Bonner. Getting started with google colab. https://towardsdatascience.com, 2019. URL https://towardsdatascience.com/getting-started-with-google-colab-f2fff97f594c.

[18] Kevin Jarrett, Koray Kavukcuoglu, Yann LeCun, et al. What is the best multi-stage architecture for object recognition? In 2009 IEEE 12th international conference on computer vision, pages 2146–2153. IEEE, 2009.

[19] Jason Brownlee. Long Short-term Memory Networks with Python: Develop Sequence Prediction Models with Deep Learning. Machine Learning Mastery, 2017.

[20] Girish Mishra, SVSSNVG Krishna Murthy, and SK Pal. Neural network based analysis of lightweight block cipher present. In *Harmony Search and Nature Inspired Optimization Algorithms*, pages 969–978. Springer, 2019.

[21] Manoj Kumar. Full-round differential attack on dot block cipher. IACR Cryptol. ePrint Arch., 2019:1285, 2019.

[22] Liam Keliher, Henk Meijer, and Stafford Tavares. Modeling linear characteristics of substitution-permutation networks. In International

Workshop on Selected Areas in Cryptography, pages 78–91. Springer, 1999.

[23] Claude E Shannon. Communication theory of secrecy systems. *The Bell system technical journal*, 28(4):656–715, 1949.

[24] Claude Carlet and Cunsheng Ding. Nonlinearities of s-boxes. *Finite fields and their applications,* 13(1):121–135, 2007.

[25] Kaisa Nyberg. On the construction of highly nonlinear permutations. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 92–98. Springer, 1992.

[26] Markku-Juhani O Saarinen. Cryptographic analysis of all 44-bit s-boxes. In *International Workshop on Selected Areas in Cryptography,* pages 118–133. Springer, 2011.

[27] Jennifer Miuling Cheung. The design of s-boxes. San Diego State Univ, 2010.

[28] Juan Soto. Statistical testing of random number generators. In *Proceedings of the 22nd national information systems security conference*, volume 10, page 12. NIST Gaithersburg, MD, 1999.

Received October, 2020

Revised February, 2021