Defence Science Journal, Vol. 71, No. 5, September 2021, pp. 647-655, DOI : 10.14429/dsj.71.16843

Distinguishing Lightweight Block Ciphers in Encrypted Images

Girish Mishra[#,@,*], S.K. Pal[#], S.V.S.S.N.V.G. Krishna Murthy[@], Kanishk Vats[$], and Rakshak Raina[!]

[#]*DRDO-Scientific Analysis Group, Delhi - 110 054, India*
[@]*DRDO-Defence Institute of Advanced Technology, Pune - 411 025, India*
[$]*Delhi Technological University, Delhi - 110 042, India*
[!]*Bennett University, Greater Noida - 201310, India*
[*]*E-mail: gmishratech28@gmail.com*

ABSTRACT

Modern day lightweight block ciphers provide powerful encryption methods for securing IoT communication data. Tiny digital devices exchange private data which the individual users might not be willing to get disclosed. On the other hand, the adversaries try their level best to capture this private data. The first step towards this is to identify the encryption scheme. This work is an effort to construct a distinguisher to identify the cipher used in encrypting the traffic data. We try to establish a deep learning based method to identify the encryption scheme used from a set of three lightweight block ciphers viz. LBlock, PRESENT and SPECK. We make use of images from MNIST and fashion MNIST data sets for establishing the cryptographic distinguisher. Our results show that the overall classification accuracy depends firstly on the type of key used in encryption and secondly on how frequently the pixel values change in original input image.

Keywords: Deep learning; Cryptography; Cryptanalysis; Lightweight block ciphers; MNIST; Fashion-MNIST

1. INTRODUCTION

In an era of IoT, lightweight block ciphers provide a powerful way of encrypting the user data to ensure much-needed privacy. Billions of gadgets might need their own encryption schemes and the adversary on the other hand will need to identify the used scheme. This anticipatory fact brings the need to construct a distinguisher from the adversary's perspective. The distinguisher for distinguishing the ciphers will also help the designer's point of view as it may help to assess the cryptographic strength of the ciphers.

The researchers have generally tried to develop two types of distinguishers; one which distinguishes between a cipher and random data. The other one predicts the class of cipher data. The development of the first type of distinguisher is based on the fact that an adversary should not be able to ascertain whether Oracle is sending the data through an encryption scheme or a random source. The second type of distinguisher tries to identify the encryption scheme used during the communication.

Rivest[1] pioneered in exploring the possibilities of the connection between cryptography and machine learning. He emphasised over the fact that how one area can contribute ideas and techniques to the other. He further perceived machine learning and cryptanalysis as sister fields as both share similar concerns and notions. After generating theoretical interest with this landmark paper and the subsequent availability of the plentiful advanced computing resources and the better-established theories, the researchers explored ML applications in cryptography in more depth.

With the rapid growth in the availability of affordable internet services to the vast population and simultaneously emerging multimedia technologies, the image and video data are being transmitted over the network in a substantial amount. Therefore, the protection of multimedia data is a vital requirement. The researchers have been coming with different encryption approaches for protecting the confidentiality of this data. For example, chaos-based image encryption[2], chaotic maps[3], cosine-transform-based chaotic system[4], the combination of an elliptic curve with Hill Cipher[5] and AES[6] are some of these approaches. On the other side, there have been continuous efforts to mount cryptanalytic attacks on encrypted images[7,8].

Linus LAGERHJELM[9], in his master thesis, used Convolutional Neural Networks (CNN) to perform classification tasks over encrypted MNIST image dataset[10]. He considered it a traditional image recognition task and showed the encrypted images to the network to predict the class label. In a 10-class (encrypted MNIST image dataset) problem, he achieved the success rate of 10% and 42% for images encrypted in CBC and ECB modes, respectively. The better results for ECB mode can be attributed to not having the desired randomness characteristic, as it is well established that ECB is the weakest mode of encryption. De mello[11], *et al.* used machine learning techniques to identify encryption algorithms in a ciphertext-only setup. The plaintext corpora for

Received : 22 February 2021, Revised : 05 April 2021
Accepted : 12 April 2021, Online published : 02 September 2021

experiments were taken from seven different languages. These plaintexts were encrypted by seven encryption algorithms in ECB and CBC modes. They used six machine learning algorithms for classification. The identification success in the case of ECB mode was as per expectation, i.e., significantly high. In comparison, the success was not up to the mark when experiments were performed for CBC mode. Wang[12], *et al.* performed an encrypted image classification task to design a framework using a multilayer extreme learning machine that they claim to classify the encrypted images before carrying out the actual decryption. They carried out their experiments for letter databases and handwritten digits. They could demonstrate that their proposed framework was efficient and accurate for classifying the encrypted images.

The construction of distinguisher by directly working over encrypted data looked infeasible, and for the first time, DL-based differential distinguisher was developed by Gohr[13]. This was a significant achievement as the results produced by this experiment were better than the classical approaches. He adopted an all-in-one approach for differential cryptanalysis and learned how a single input difference affects all possible output differences. This work has been seen as the first remarkable breakthrough of any ML technique for cryptanalysis, which opened a new research direction in cryptanalysis. This direction was subsequently adhered to by some more researchers[14,15,16,17] as well.

This paper is an effort to develop a distinguisher that directly works over encrypted images. As shown in Fig. 1, three ciphers, namely LBlock, PRESENT, and SPECK, have been used for encryption. The image database considered for our experiments is MNIST handwritten digit database[10] and Fashion-MNIST database[18]. MNIST dataset, containing grayscale handwritten digit images, has been a benchmark dataset for researchers to validate their ML algorithms. Later on, the Fashion-MNIST dataset was introduced by researchers from a company called Zalando. This dataset's idea was a suitable and compatible replacement of the MNIST dataset for a standard benchmarking of any state-of-the-art machine learning algorithm. However, for our work, we consider both the datasets to carry out the experiments in two scenarios; first when a fixed key is used for encryption of all images and the second when different keys are used for different images. Our results clearly show that the developed approach for distinguisher works very well in the MNIST digit database case. In contrast, the success rate decreases when the same experiment is done for the Fashion-MNIST database. We conclude with the observation that this difference in result is because of frequently changing pixel values (i.e., shades) in the later database.

2. LIGHTWEIGHT BLOCK CIPHERS (LBLOCK, PRESENT AND SPECK)

Three lightweight block ciphers, each based on a different design principle, have been considered for the experiments. LBlock is a Feistel cipher, PRESENT is an SPN based cipher scheme and the design of SPECK uses ARX architecture. The descriptions of these ciphers are given in following subsections.

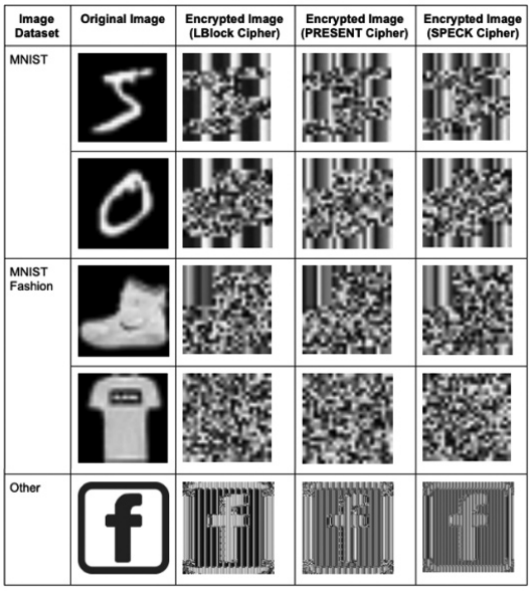

Figure 1. Some encrypted image samples.

2.1 Description of LBlock Cipher

LBlock cipher, introduced by Wu[19], *et al.,* uses an 80-bit key to encrypt 64-bit plaintext block and generates the corresponding 64-bit ciphertext block. This 32-round cipher employs a variant of Feistel structure. As described[19], the encryption algorithm is shown in Algorithm 1. The key expansion algorithm is omitted here, and the paper[19] may be referred to for more details.

Algorithm 1 (*Encryption Algorithm of LBlock Cipher*)

1. Input : $P = X_1 \| X_0$ *and Round Keys* K_i;
   $i = 1,2,\ldots\ldots,32$ *obtained from Master Key*
2. Output : $C = X_{33} \| X_{32}$
3. for $i = 2$ *to* 33 *do*
4. $\quad T = X_{i-1} \oplus K_{i-1}$
5. $\quad T \rightarrow T_7 \| T_6 \| T_5 \| T_4 \| T_3 \| T_2 \| T_1 \| T_0$
6. $\quad R = S_7(T_7) \| S_6(T_6) \| S_5(T_5) \| S_4(T_4) \| S_3(T_3) \| S_2(T_2) \| S_1(T_1) \| S_0(T_0)$
7. $\quad R \rightarrow R_7 \| R_6 \| R_5 \| R_4 \| R_3 \| R_2 \| R_1 \| R_0$
8. $\quad U = R_6 \| R_4 \| R_7 \| R_5 \| R_2 \| R_0 \| R_3 \| R_1$
9. $\quad X_i = U \oplus (X_i <<< 8)$
10. end for

The notations used in the algorithm are :

P : *plaintext of size* 64 *bits*

C : *ciphertext of size* 64 *bits*

K : *master key of size* 80 *bits*

K_i : i^{th} *round key*

S_i : $i^{th} S-box$ (*Shown in Table* 1)

$<<< 8$: *left cyclic shift with* 8 *bits*

$\|$: *Concatenating two binary strings*

$\oplus$: *exclusive – OR*

Table 1. S-boxes used in LBlock cipher

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S_0(x)$	E	9	F	0	D	4	A	B	1	2	8	3	7	6	C	5
$S_1(x)$	4	B	E	9	F	D	0	A	7	C	5	6	2	8	1	3
$S_2(x)$	1	E	7	C	F	D	0	6	B	5	9	3	2	4	8	A
$S_3(x)$	7	6	8	B	0	F	3	E	9	A	C	D	5	2	4	1
$S_4(x)$	E	5	F	0	7	2	C	D	1	8	4	9	B	A	6	3
$S_5(x)$	2	D	B	C	F	E	0	9	7	A	6	3	1	8	4	5
$S_6(x)$	B	9	4	E	0	F	A	D	6	C	5	7	3	8	1	2
$S_7(x)$	D	A	F	0	E	4	9	B	2	1	8	3	7	5	C	6
$S_8(x)$	8	7	E	5	F	D	0	6	B	C	9	A	2	4	1	3
$S_9(x)$	B	5	F	0	7	2	9	D	4	8	1	C	E	A	3	6

The designers' thorough security analysis showed LBlock achieving necessary security edge against known cryptanalytic attacks and the cipher being efficient in different hardware environments and various software platforms. To date, no researcher has projected any known full-round cryptanalytic attack on LBlock.

2.2 Description of PRESENT

PRESENT is one of the most analysed lightweight cipher, which was introduced by Bogdanov[20] , *et al.*. PRESENT cipher, a Substitution-Permutation Networks (SPN) based architecture, comprises of 31 rounds. It encrypts 64-bit plaintext using an 80-bit or 128-bit key to generate the equivalent 64-bit ciphertext. The encryption algorithm of PRESENT cipher is shown in Algorithm 2. The description of key scheduling algorithm is left out here, and the same can be found in[20] with complete details.

Algorithm 2 (Encryption Algorithm of PRESENT Cipher)

1. Input : $P = b_{63}b_{62}...b_0$ *and Round Key* $K_i = k^i{}_{63}k^i{}_{62}...k^i{}_0$;
   $i = 1,2,\ldots\ldots,32$ *obtained from* $80-bit$ *Master Key* K
2. Output : $C = Z_{31}$
3. for $i = 1$ *to* 31 *do*
4. for $j = 0$ *to* 63 *do*
5. $b_j = b_j \oplus k^i{}_j$
6. end for
7. for $k = 0$ *to* 15 *do*
8. $w_k = b_{4k+3} \| b_{4k+2} \| b_{4k+1} \| b_{4k}$
9. $Y_k = S(w_k)$
10. $Y_k \rightarrow y_{4k+3} \| y_{4k+2} \| y_{4k+1} \| y_{4k}$
11. end for
12. $Y_i = y_{63}y_{62}...y_0$
13. $Z_i = P(y_{63})P(y_{62})...P(y_0)$
14. $Z_i \rightarrow b_{63}b_{62}...b_0$
15. end for

S-box (S) and Permutation (P) used in the algorithm are shown in Table 2 and Table 3 respectively.

The designers' idea behind designing a lightweight block cipher PRESENT instead of a stream cipher was that they thought about block ciphers being versatile compared to stream ciphers. Moreover, a stream cipher can easily be obtained from a block cipher by running the latter in counter mode. Moreover, they acknowledged a better understanding of block cipher by the crypto research community. The authors targeted the hardware environment for encryption algorithm implementation, and therefore, PRESENT is not specifically software-friendly.

2.3 Description of SPECK

SPECK is a block cipher proposed by the National Security Agency (NSA). It has ten variants, each variant is characterised by individual block size $2n$ and key size mn. For example, SPECK64/128 refers to the variant of SPECK block cipher with 64-bit block size and 128-bit key size. We have used this version of SPECK cipher for our experiments. The SPECK64/128 performs a mapping from a plaintext of two 32-bit words (x_0, y_0) into ciphertext (x_{27}, y_{27}) by using a sequence of 27 rounds. The Round function is defined as follows:

$$R^k(x,y) = (((x >>> 8) \boxplus y) \oplus k, (y <<< 3) \oplus ((x >>> 8) \boxplus y) \oplus k)$$

where k is the round key. We have left describing key scheduling algorithm and the same and the more details about the encryption algorithm are described in[21].

The designer of SPECK claimed the cipher to be one of the fastest contemporary ciphers and being fully secure against chosen-plaintext attack and chosen-ciphertext attack. Many countries have been concerned about the standardisation efforts of SPECK as they were apprehensive about susceptible weaknesses in the design. Although in response, the designer agency (National Security Agency, USA) stated about a large number of security analysis research activities from all around the world to support its claim about the algorithm being secure and the absence of any knowledgeable flaw in the cipher design.

Table 2. S-box (S) used in PRESENT cipher

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S_0(x)$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Table 3. Permutation (*P*) used in PRESENT cipher

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

3. THE DESCRIPTION OF DATABASE

For performing the experiments, two different databases have been used. The description of these are given in following subsections:

3.1 MNIST Database

Modified National Institute of Standards and Technology database (i.e. The MNIST database[10]) is broadly a large database of handwritten digits frequently used in training of many diversified image processing applications. MNIST database is widely used for developing various machine learning models[22,23]. The MNIST database was generated by modifying the samples from NIST's original datasets[24]. The creators of MNIST believed that the original database was not the right choice for machine learning experiments as the NIST's training dataset was taken from working professionals, whereas the testing dataset was collected from high school students. The MNIST database accommodates 70,000 images consisting of 60,000 training and 10,000 testing images. Each image in the database is of 28 × 28 pixel size. Few samples of images are shown in Fig. 2.

Figure 2. Few image samples from MNIST dataset.

3.2 Fashion-MNIST Database

Xiao[18], *et al.* presented the Fashion-MNIST dataset containing 70,000 images of various fashion products from 10 different categories, each category having 7,000 grayscale images. The size of each image is 28 × 28. The complete dataset is segregated into a training set with 60,000 images and a test set having 10,000 images. The thought behind Fashion-MNIST dataset was to have an advanced replacement of the original MNIST dataset for benchmarking purposes of machine learning algorithms. Xiao[18], *et al.* mention that when MNIST dataset was proposed, the researchers could not have envisaged the widespread rise in performances of deep learning techniques. Both the datasets have the same data format, identical image size, and exactly the same structure of training and testing splits[18]. Few samples of images from Fashion-MNIST dataset are shown in Fig. 3.

4. DEEP LEARNING APPROACH FOR EXPERIMENTS

Deep Learning, the latest discipline in machine learning techniques consisting of various learning methods, are primarily based on Artificial Neural Network (ANN) framework. Deep Learning (DL), which are also termed as Deep Neural Networks (DNN), consists of a wide variety of architectures such as Deep Belief Network (DBN), Convolution Neural Network (CNN), and Recurrent Neural Network (RNN), which are used in diversified domains to solve various machine intelligence problems such as medical image processing, natural language processing, computer vision, machine translation, etc.[25,26]. These DL architectures have emerged with the outstanding success in the respective domains, which mostly outclassed the human expertise. Deep learning inherently has an additional advantage in comparison to traditional machine learning algorithms due to its ability of representation learning. Representation learning denotes a class of techniques that in an automatic way discovers the representations directly from the raw data. In short, deep learning is a category of machine learning algorithms[27], which employs intermediate multiple hidden layers of neurons to ensure feature learning straight from the raw input data.

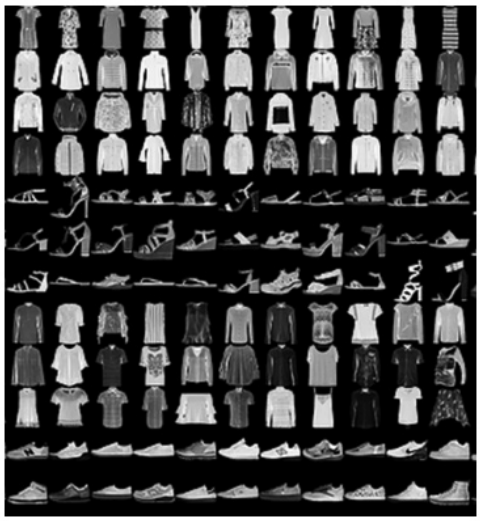

Figure 3. Few image samples from fashion-MNIST dataset.

In our experiments, we have used two types of DL networks; (i) Convolution Neural Network (CNN)[28,29], and (ii) DenseNet[30].

4.1 Convolution Neural Networks

The evolving advancements in Computer Vision has got attributed to one specific algorithm from deep learning, which is Convolutional Neural Network (CNN). The working of CNN is primarily based over the fact that the input will be a collection of images. As shown in Fig. 4, CNNs are three layered architectures; (i) Convolutional layers, (ii) Pooling layers, and (iii) Fully-connected layers.

In Convolution layer, the convolution is performed over input data using some filters during the forward computation phase. The output from this convolution is termed as feature map. The filters extract and forward the features detected from the input data in the form of the feature map (as shown in Fig. 5). From the example shown in the same figure, the output after applying the filter (values in the feature map) can be shown as:

$$Y_{i,j} = \sum_{a=1}^{3}\sum_{b=1}^{3} w_{a,b} x_{i+a,j+b}$$

$$for\ i,j = 1,2,3,4,5$$

$$where, X = (x_{i,j}) \in \mathbb{R}^{7\times 7}$$

is the input image. The image is convoluted with the filter $W = (w_{a,b}) \in \mathbb{R}^{3\times 3}$ without using any extra padding in itself. The $w_{a,b}$ components are the weights of the filter. $Y = (y_{i,j}) \in \mathbb{R}^{5\times 5}$ is the feature map after convolution. During the backward computation process, the training model learns the filter weights in order to minimise the final loss.

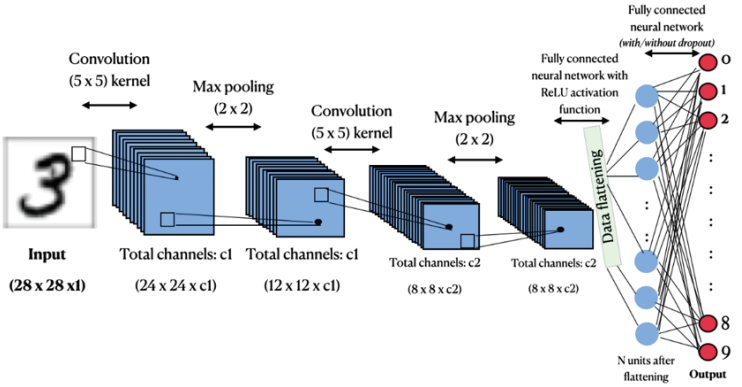

Figure 4. Sample figure of convolution neural network.

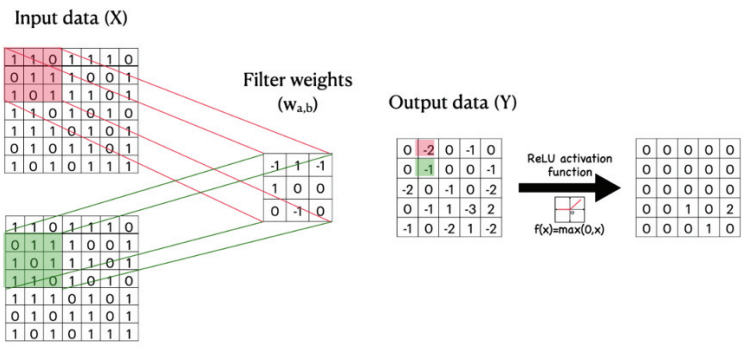

Figure 5. Example of a convolution layer.

In Max pooling layer, the feature map is split into smaller regions and the maximum values from all regions are concatenated to form the output of the layer. The max pooling layer helps in reducing the computation complexity.

After convolution and max pooling layers, the fully-connected layers perform in the identical manner any standard ANN (Artificial Neural Network) does and then by using activation functions (such as Softmax, ReLU) finally they produce the class scores for doing the classification.

Learning through the filters helps in extracting high level features of the data. Thus, the filters along with max pooling layers in CNN works as a method of dimensionality reduction. Thus, the internal feature extraction capability of CNN in finding and forwarding the most valuable information perform extremely well for image classification task.

The overall architecture of CNN consists of two main parts: (i) feature extractor and (ii) classifier. The feature extractor part is generally composed of stacked operations of convolution, activation, and pooling layers. Thereafter, the classifier comprises of some fully-connected layers of neurons. During the forward propagation process, each layer forwards its output as the input for the next layer. Thus the lower level layers propagate the features which subsequently result in deriving the higher-level features. Finally, the classification probabilities are calculated in the last output layer. In a classification task, the output layer is generally activated by softmax function.

The convolution layer reduces the number of parameters and the computational time. The output of the convolution layer goes through a non-linear activation function (generally ReLU). After this, the pooling layer performs the downsampling operation to keep only the useful features and discard the redundancies. For example, if a 2 × 2 max-pooling layer is used with the stride size 2, then the output size is half of the input size. The backward propagation process during training uses the optimizer, such as Adam with the objective of minimising the total loss.

4.2 Densely Connected Convolutional Networks

Densely Connected Convolutional Networks or DenseNet was visualised by Huang[30], *et al.* in 2018 on the basis of observation that CNNs can be significantly deeper, efficient and more accurate in training if the connections between layers are shorter and closer to the input and the output. This observation also brings the possibility of addressing the problem of vanishing gradient. In vanishing gradient problem, the gradients get washed out once it reaches to any end of the network. This occurs as the final gradient is calculated using chain rule of differentiation which involves the multiplication of intermediate small differentiation values. In DenseNet, each layer is connected with all other layers in a feed-forward fashion. So, there is clear difference that the traditional CNNs with L layers have L connections, one connection each between one layer and the very next layer, whereas DenseNet involves $\frac{L(L+1)}{2}$ direct connections. In DenseNet, each layer uses

the feature maps from all its preceding layers as inputs. Also, each layer's feature maps behave as inputs for all subsequent layers. The connection between all pairs of layers are naturally advantageous as it vanishes the vanishing-gradient problem, and builds up the strong feature propagation and reuses. A sample figure of DenseNet is shown in Fig. 6.

The longer path (i.e., large number of hidden layers) between the input and the output layer in a DL network brings in vanishing gradient problem. This problem means that the information disappears enroute to its destination, which causes the declined performance or the less accuracy of the model. DenseNet was specifically developed to resolve this problem. In DenseNet architecture, an output from the previous layer moves as an input for the second layer through a composite function. The operation of composite function is composed of: Convolution layer, pooling layer, batch normalisation, and non-linear activation layer. Therefore, if there are L number of layers after the input layer, then the network involves $\frac{L(L+1)}{2}$ direct connections.

The DenseNet has several variants such as DenseNet-121, DenseNet-160, and DenseNet-201. The numerical values indicate the exact number of layers within the network. For example, the architecture of DenseNet-121 consists of one initial layer with convolution and pooling, followed by a dense block 1. It is then followed by a transition layer 1. Subsequently, the architecture uses dense block 2, transition layer 2, dense block 3, transition layer 3 and dense block 4. Here, each of dense block 1, dense block 2, dense block 3 and dense block 4 involves two convolutions with 1 × 1 and 3 × 3 sized kernels repeated 6, 12, 24 and 16 times respectively. Each transition layer involves one 1×1 convolution layer and one 3 × 3 average pooling layer with stride 2. Finally, the structure completes itself with the classification layer. The number 121 associated with DenseNet signifies 1 initial layer, 6, 12, 24, and 16 repeated instances of 2 convolutions inside dense blocks, 3 transition layers and 1 classification layer. So, overall the resultant is $1+\left[(6+12+24+16)*2\right]+3+1=121$.

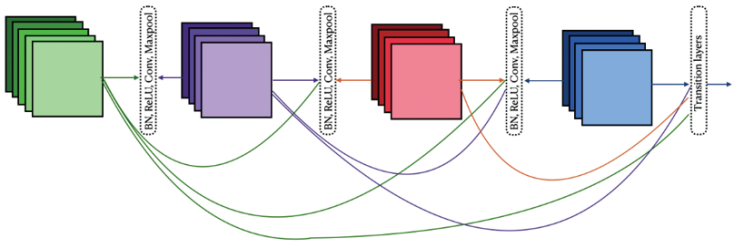

Figure 6. Sample figure of DenseNet.

5. RESULTS AND ANALYSIS

As mentioned earlier, we train and validate our CNN and DenseNet models over two different datasets, namely MNIST and Fashion-MNIST. We further segregate our experiments into two subcategories: (i) the block cipher used for encrypting the images using a fixed key, and (ii) the block cipher uses a different random key for encrypting each image. The encryption has been done in ECB (Electronic Code Book) mode. In first subcategory i.e., for fixed key, we consider 1000 encrypted images for experiments. In second category i.e., for random keys, we carried out our experiments only for MNIST digit image dataset. The results have not been convincing in the second subcategory. Here the classification accuracy is close to 1/3, which indicates the models' inability to capture any meaningful patterns from the encrypted images. Due to this reason, we did not perform the experiments for the Fashion-MNIST dataset. For building the CNN model, 3 convolution layers followed by flattened data and 2 hidden layers are used. Besides this, following hyperparameters have been used in training the CNN model: Batch size= 32, Epochs= 40, and Validation split= 0.2. The DenseNet model used for experiments is a 5-layer block with a growth rate of k=4. Apart from this, the other parameters are Adam optimiser, categorical cross-entropy and accuracy as metrics for evaluating the model.

The detailed results are shown in Table 4. These results are also shown graphically (row-wise sequence) in Fig. 7. The results from our experiments are better in case of MNIST dataset than Fashion-MNIST dataset. We analysed the results and observed that the reason behind this bias is due to MNIST dataset having more regularity (pixel values changing less frequently) than the other dataset. Using the fixed encryption key, maps all identical blocks in the images made with the same sequence of pixel values to a fixed block value after encryption. However, in the case of using different encryption keys for different images, these identical blocks get mapped to different block values after encryption, and our models do not capture the distinguishing features and therefore result in poor accuracy (as expected for any dataset having pseudorandom characteristics in itself). In future, we plan to use few other DL models over some additional datasets. In addition, the results are sequentially shown row-wise through graphs in Fig. 7.

Table 4. Results showing classification accuracy for our experiments

Key use scenario	Dataset	Model	Samples per class	Training accuracy (fraction)	Validation Accuracy (fraction)
Fixed Key	MNIST Digit	CNN	1000	0.9975	0.9878
	MNIST Digit	DenseNet	1000	0.9995	0.9567
	MNIST Fashion	CNN	1000	0.9113	0.6450
	MNIST Fashion	DenseNet	1000	0.7617	0.6544
Random Key	MNIST Digit	CNN	1000	0.3491	0.3111
	MNIST Digit	DenseNet	1000	0.3733	0.3677

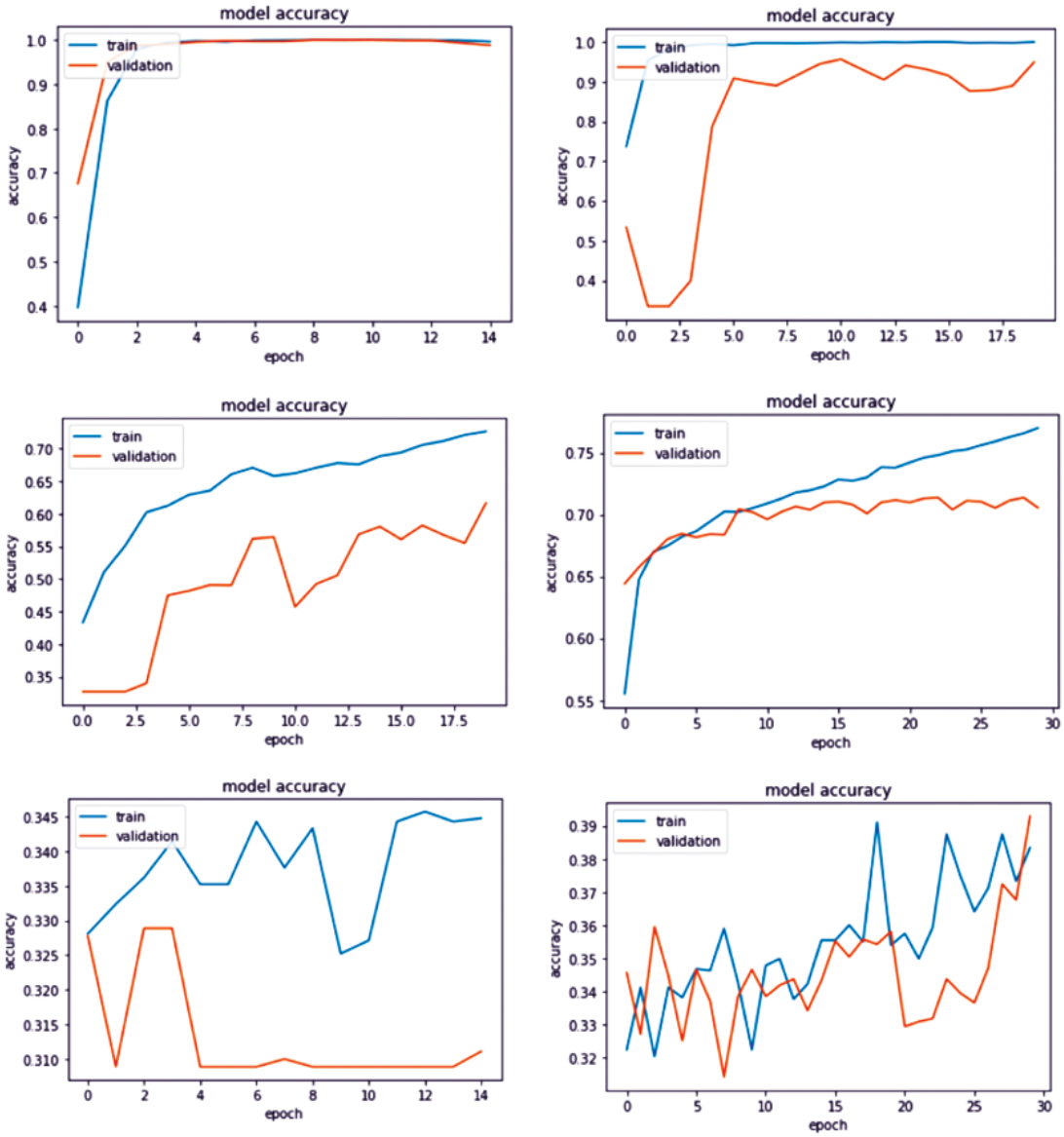

Figure 7. The classification results of DL model.

6. CONCLUSION

In this paper, we proposed a deep learning based approach for constructing a distinguisher for lightweight block ciphers. The objective of this cipher is to classify the lightweight cipher. In other words, the focus of this distinguisher is to identify the cipher used in an encrypted image. Three lightweight ciphers based on three different design principles have been chosen for our experiments because of their importance in IoT infrastructure. We performed the classification work over two popular datasets viz. MNIST and Fashion-MNIST and achieved the better classification accuracy for MNIST data in comparison to Fashion-MNIST. Finally, we concluded that the frequent changes in pixel values adversely affects the overall classification success. In future, the similar classification work will be carried out for other modes of encryption such as cipher block chaining (CBC), Output feedback (OFB) and CTR (Counter). Other DL methods will also be utilised for analysing the cipher data generated from images as well as from text.

REFERENCES

1. Rivest, R.L. Cryptography and machine learning. *In* Proceedings of the International Conference on the Theory and Application of Cryptology, Springer, Berlin, Heidelberg, 1991, pp. 427-439. doi: 10.1007/3-540-57332-1_36
2. Al-Maadeed, S.; Al-Ali, A. & Abdalla, T. A New Chaos-Based Image-Encryption and Compression Algorithm. *J. Elect. Comput. Eng.,* 2012, 1–11. doi:10.1155/2012/179693
3. Lu, Q.; Zhu, C. & Deng, X. An Efficient Image Encryption Scheme Based on the LSS Chaotic Map and Single S-Box. *IEEE Access*, 2020, 8, 25664–25678. doi:10.1109/access.2020.2970806
4. Hua, Z.; Zhou, Y. & Huang, H. Cosine-transform-based chaotic system for image encryption. *Information Sciences*, 2019, **480**, 403–419. doi:10.1016/j.ins.2018.12.048
5. Dawahdeh, Z. E.; Yaakob, S. N. & Razif bin Othman,

R. A new image encryption technique combining Elliptic Curve Cryptosystem with Hill Cipher. *J. King Saud University – Comput. Info. Sci.*, 2018, **30**(3), 349–355.
doi:10.1016/j.jksuci.2017.06.004

6. Zhang, Y.; Xueqian Li, & Wengang Hou. A fast image encryption scheme based on AES. *In* 2017 2nd International Conference on Image, Vision and Computing (ICIVC), 2017.
doi:10.1109/icivc.2017.7984631
7. Li, C., Lin, D., Feng, B., Lu, J., & Hao, F. (). Cryptanalysis of a Chaotic Image Encryption Algorithm Based on Information Entropy. *IEEE Access,* 2018, **6**, 75834–75842.
doi:10.1109/access.2018.2883690
8. Wang, H., Xiao, D., Chen, X., & Huang, H. (). Cryptanalysis and enhancements of image encryption using combination of the 1D chaotic map. *Signal Processing*, 2018, 144, 444–452.
doi:10.1016/j.sigpro.2017.11.005
9. Lagerhjelm, L. Extracting information from encrypted data using deep neural networks, 2008, https://www.diva-portal.org/smash/get/diva2:1284274/FULLTEXT01.pdf Accessed on 14 Jan 2021]
10. LeCun, Y. The MNIST database of handwritten digits. http://yann. lecun. com/exdb/mnist/. 1998. [Accessed on 14 Jan 2021]
11. de Mello, F. L. & Xexéo, J. A. Identifying Encryption Algorithms in ECB and CBC Modes Using Computational Intelligence. *J. UCS*, 2018, **24**(1), 25-42.
12. Wang, W.; Vong, C.M.; Yang, Y. & Wong, P.K. Encrypted image classification based on multilayer extreme learning machine. *Multidimensional Systems and Signal Processing*, 2017, **28**(3), pp.851-865.
doi:10.1007/s11045-016-0408-1
13. Gohr, A. Improving attacks on round-reduced speck32/64 using deep learning. *In* Proceedings of Annual International Cryptology Conference, Springer, 2019, pp. 150–179.
doi: 10.1007/978-3-030-26951-7_6
14. Baksi, A.; Breier, J.; Dong, X. & Yi, C. Machine learning assisted differential distinguishers for lightweight ciphers. *IACR Cryptol. ePrint Arch.*, 2020, p.571, https://eprint.iacr.org/2020/571.pdf [Accessed on 14 Oct 2020].
15. Jagielski, M.; Carlini, N.; Berthelot, D.; Kurakin, A. & Papernot, N. High accuracy and high fidelity extraction of neural networks. *In* 29th {USENIX} Security Symposium ({USENIX} Security 20), 2020, pp. 1345-1362.
16. Xiao, Y.; Hao, Q. & Yao, D. D. Neural cryptanalysis: metrics, methodology, and applications in cps ciphers. *In* Proceedings of the Conference on Dependable and Secure Computing (DSC), IEEE, 2019, pp. 1-8.
doi:10.1109/dsc47296.2019.8937659
17. Bellini, E. & Rossi, M. Performance comparison between deep learning-based and conventional cryptographic distinguishers. https://eprint.iacr.org/2020/953.pdf [Accessed on 14 Jan 2021]
18. Xiao, H.; Rasul, K. & Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.
arXiv preprint arXiv:1708.07747, 2017.
19. Wu, W. & Zhang, L. LBlock: a lightweight block cipher. *In* International conference on applied cryptography and network security. Springer, Berlin, Heidelberg, 2011, pp. 327-344.
doi:10.1007/978-3-642-21554-4_19
20. Bogdanov, A.; Knudsen, L.R.; Leander, G., Paar, C.; Poschmann, A.; Robshaw, M.J.; Seurin, Y. & Vikkelsoe, C. PRESENT: An ultra-lightweight block cipher. *In* International workshop on cryptographic hardware and embedded systems. Springer, Berlin, Heidelberg, 2007, pp. 450-466.
doi:10.1007/978-3-540-74735-2_31
21. Beaulieu, R.; Shors, D.; Smith, J.; Treatman-Clark, S.; Weeks, B. & Wingers, L. The SIMON and SPECK lightweight block ciphers. *In* Proceedings of the 52nd Annual Design Automation Conference*,* 2015, pp. 1-6.
doi:10.1145/2744769.2747946
22. Schott, L.; Rauber, J.; Bethge, M. & Brendel, W. Towards the first adversarially robust neural network model on MNIST, 2018.
arXiv preprint arXiv:1805.09190.
23. Cohen, G.; Afshar, S.; Tapson, J. & Van Schaik, A. EMNIST: Extending MNIST to handwritten letters. *In* International Joint Conference on Neural Networks (IJCNN), 2017, pp. 2921-2926.
doi:10.1109/ijcnn.2017.7966217
24. Grother, P.J. NIST special database 19. Handprinted forms and characters database, National Institute of Standards and Technology, 1995, p.10.
25. Chen, C.; Seff, A.; Kornhauser, A. & Xiao, J. Deepdriving: Learning affordance for direct perception in autonomous driving. *In* Proceedings of the International Conference on Computer Vision, IEEE, 2015, pp. 2722-2730.
doi: 10.1109/iccv.2015.312
26. Bahdanau, D.; Cho, K. & Bengio, Y. Neural machine translation by jointly learning to align and translate, 2014. arXiv preprint arXiv:1409.0473. https://arxiv.org/abs/1409.0473 [Accessed on 01 Aug 2020].
27. Deng, L. & Yu, D. Deep learning: methods and applications. Foundations and trends in signal processing, Now Publishers Inc. Hanover, MA, USA, 2014, **7**(3–4), pp. 197-387.
doi: 10.1561/9781601988157
28. Albawi, S.; Mohammed, T.A. & Al-Zawi, S. Understanding of a convolutional neural network. *In* International Conference on Engineering and Technology (ICET), 2017, pp. 1-6. Ieee.
doi:10.1109/icengtechnol.2017.8308186
29. O'Shea, K. & Nash, R. An introduction to convolutional neural networks, 2015. arXiv preprint arXiv:1511.08458.
30. Huang, G.; Liu, Z.; Van Der Maaten, L. & Weinberger, K.Q. Densely connected convolutional networks. *In* Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700-4708.
doi:10.1109/cvpr.2017.243

CONTRIBUTORS

Mr Girish Mishra has been working as a Scientist in DRDO-Scientific Analysis Group, since 2003. He has worked and contributed in various areas related to Cryptography, Information Security and Machine Learning. He has published more than 15 research papers in various international journals and conferences. He received DRDO Young Scientist Award in 2010 and Lab Scientist Award in 2007. His current areas of interest include Machine Learning, Blockchain Technology, and Cryptography.
His contribution in the current study is development, execution, and analysis of the concept and the algorithm.

Dr Saibal Kumar Pal is working as a Senior Research Scientist and Divisional Head at DRDO-Scientific Analysis Group, Delhi. He completed PhD from University of Delhi in the area of Information Security. He has co-authored 3 books on Electronic Governance, AI & Data Science & has more than 250 research publications in peer-reviewed journals & international conference proceedings. His areas of interest are Cryptography, Cyber Security, Computational Intelligence and Information Systems. He is a recipient of Lab Scientist Award in 2010 and DRDO Scientist of the Year Award in 2012.
In the current study, he has been involved in development of the concept and technique.

Dr S.V.S.S.N.V.G. Krishna Murthy obtained PhD from Indian Institute of Technology Kanpur, Kanpur. Subsequently he was offered a Post Doctoral position under Erasmus Mundus – WILL Power (Window India Learning Link Power) Fellowship '2010-11', at Ecole Centrale Paris, France. He is currently Associate Professor & Head, Department of Applied Mathematics, Defence Institute of Advanced Technology, Deemed University, Pune. His research interests include Mathematical Modelling, Finite Element Analysis in fluid flow through Porous Media, Numerical Method's for Partial differential equations, Numerical parallel Algorithms, Computational Fluid Dynamics. He has authored more than 30 publications in reputed journals other than conferences.
In the current study, he has been involved in development of the concept and technique.

Mr Kanishk Vats is a third-year Bachelor of Technology student of Software Engineering at Delhi Technological University. He has worked on restoring degraded images using super resolution convolutional neural network. His areas of interest are Software Engineering, and Machine Learning.
In the current study, he is involved in execution and verification of test results.

Mr Rakshak Raina is a third-year Bachelor of Technology student of Computer Science at Bennett University. He has worked on different projects in areas of Machine Learning and IoT Security. His areas of interest are Data Analytics and Machine Learning.
In the current study, he is involved in execution and verification of test results.