Investigation of an Orthogonal Encryption Technique for Deep Learning

ThienNgo Le
Department of Computer Science
Colorado School of Mines
Golden, CO, USA
thienngole@mymail.mines.edu

***Abstract*— Encrypted Deep Learning Services is a great solution for those who work in the AI field, but it is limited in resources. While avoiding expensive investment and providing flexibility on scaling, encrypted deep learning services also provide the privacy and confidentiality of the data for its users. However, this study will show that it may cost users more than what they can save due to the loss of up to *18%* of accuracy on the trained model. Also, this study examines possible vulnerabilities.**

Keywords—Machine Learning; Encryption; Deep Learning; Deep Learning Services; Encrypted Deep Learning Services; security and privacy; Known Plaintext Attack.

1. INTRODUCTION

Deep learning service (DLS) is getting more popular every day in our computing world. It is a great solution for small companies, researchers, and individuals to avoid an expensive investment on the high-performance computing system for the AI field. Deep learning services provide users fast and stable needed infrastructure. By using DLS, it is fairly easy to scale up the system without worrying about all the aspects of scaling, and users only have to pay for what they use. It is also a great deal for all short-term users.

Orthogonal encryption is one of many methods that are invented to use in DLS to protect the privacy and confidentiality of the data when using DLS, which is called Encrypted Deep Learning Services (EDLS). EDLS encrypts rows of data before sharing for the training process. This means that we can use EDLS to train, predict, and validate our models without losing the confidentiality of the data.

This study designed an experiment to evaluate the accuracy of a deep learning model trained on encrypted data and the strength of orthogonal encryption technique by applying the orthogonal encryption technique on two datasets, MINST and CIFAR10. It turns out that the accuracy of trained models and the strength of this orthogonal encryption technique depend heavily on the characteristic on the dataset it is applied to rather than the crypto algorithm itself.

2. DEEP LEARNING SERVICES AND ORTHOGONAL ENCRYPTION

2.1 Deep learning services

Generally, every machine learning model involves three phrases: configuration, training, and deploying. The training process of any machine learning model requires high computing power, so hardware configuration including high-end CPUs, GPUs, or FPGAs is normally used to accelerate the training process. However, the cost of such a system is an expensive investment that is not affordable to many people. Therefore, machine learning/deep learning as services is a good solution. Deep learning service (DLS) is a set of services that provide machine learning tools as part of cloud computing services offered by many big cloud providers such as Microsoft, Amazon, and IBM. DLS providers will host everything and be responsible for all aspects of scaling, then provide all the necessary assets to the users and the users only have to pay for the resources that they use. By utilizing these services, users can save time, money, and resources that would have been invested into hosting their own system. This makes DLS one of the fastest growing cloud services. According to Industry Research, DLS will grow with the compound annual growth rate of over 38 percent during the period of 2019-2023[1].

One of the benefits of DLS is that it provides users fast and stable needed infrastructure. DLS will free small companies, organizations, and even individuals from the burden of building their own in-house infrastructure from scratch. By using DLS, it is fairly easy to scale up the system without worrying about all the aspects of scaling. Since users only have to pay for what they use, DLS is also a great deal for all short-term users. However, when it comes to confidentiality, it is hard for DLS to guarantee this. Traditionally, to ensure data confidentiality, data owners need to host an in-house or private deep learning service. An in-house deep learning service may not be a viable option for many people and is often an expensive solution. Ensuring confidentiality of the data is a big challenge in DLS because in many cases, it is very important on who can get access to the training data, especially when the data is sensitive. For example, when a group of researchers want to train a model to detect a new disease, they would not want any third party to have access to their training data because it may contain information about patients, and it may be illegal for them to share. Hence, Encrypted Deep Learning Services (EDLS) was introduced as a solution. In EDLS only encrypted rows of data are shared for the training process. This means that we can use EDLS to train, predict, and validate our models without any revealing of the original data. This enables researchers to get access to more sensitive data for researching and still ensures the confidentiality of the data. Figure 1 shows a sample architecture of an EDLS.

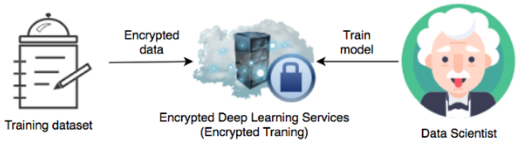


Figure 1: Encrypted Deep Learning Service Model

There have been quite a few encryption techniques introduced in EDLS such as Homomorphic Encryption and Orthogonal Encryption. This study will focus on investigating Orthogonal Encryption in EDLS.

2.2 *Orthogonal Transformation based Encryption*

Orthogonal transformation is a linear transformation that preserves the length of vectors. Let T be a linear transformation; if $||T(v)|| = ||v||$ where v is a vector, then T is called an orthogonal transformation. If $T(v) = A(v)$ then A is called an orthogonal matrix; matrix A columns and rows are orthogonal vectors [8]. Orthogonal transformation can use the transformational properties of the orthogonal matrix to transform an image into a new image that contains no information and usability of the original image. The original image can be retrieved by performing the dot product of the new image with the inverse of the orthogonal matrix. Orthogonal transformation has been applied to many areas such as in design of a feedforward neural network with optimum number of links and input nodes, design of a neural network operating with orthogonalized data for periodic process, and convergence assessment of networks [9].

Some researchers proposed to use orthogonal transformation as an encryption technique. They named this type of encryption as orthogonal encryption. Based on the properties of orthogonal transformation, orthogonal encryption can use an orthogonal matrix to encrypt a clear image (plain image) into a new image (cipher image) that has no information and usability of the plain image. The cipher image can be decrypted by the decryption key which is the inverse of the orthogonal matrix. A good orthogonal encryption method must be able to provide reduction in image size and the ability to be compressed after the encryption process.

This study investigates a specific orthogonal encryption technique introduced in patent US20190087689A1: METHODS AND PROCESSES OF ENCRYPTED DEEP LEARNING SERVICES [2]. There are many ways to retrieve an orthogonal matrix; the introduced schema performs a QR decomposition on an image to obtain the orthogonal matrix. Specifically, the operation decomposes an image as matrix A into a product A=Q*R where Q is an orthogonal matrix and R is an upper triangular matrix. Orthogonal matrix Q is then used as an encryption key to perform matrix multiplication with plain images to calculate cipher images. The inverse of Q will be used as the decryption key to decrypt cipher images by performing matrix multiplication with cipher images to obtain the corresponding plain image. According to [2], the proposed orthogonal encryption technique not only ensures the confidentiality of the data but also uses less memory and has faster prediction speed. The proposed orthogonal encryption technique uses 15 times less memory and significantly faster prediction speed than Crypto net, a Neural-Networks over data encrypted with Homomorphic Encryption. Figure 2 shows an image A decomposed into R (upper triangular matrix) and Q (orthogonal matrix used as encryption key). The decryption key, Q^{-1}, is the inverse of Q. The cipher image is the product of the plain image from the MNIST dataset and encryption key Q. The decrypted image is the product of the cipher image and the decryption key Q^{-1}.

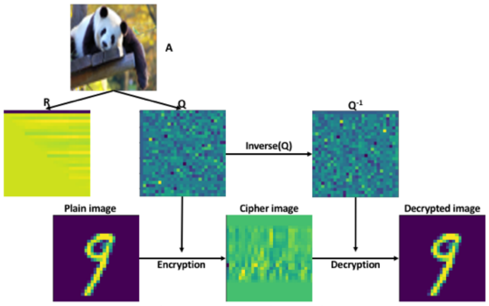


Figure 2: Orthogonal encryption

3 GOAL AND EXPERIMENT DESIGN

This section describes the experiments and the goals of this study. This study will use a convolutional neural network as a deep learning model to train on original versions and encrypted versions of both MNIST and CIFAR10 datasets, then measure the accuracy on models trained on encrypted data compared to clean data and analyze the strength of orthogonal encryption algorithms.

3.1 *Goals*

When it comes to machine learning, the cost of errors on any machine learning model can be really huge. For example, a false positive cancer diagnosis will cost both patient and hospital financially and mentally. Optimizing the accuracy of a machine learning model can mitigate that cost. Therefore, ***model accuracy*** is important. When it comes to cryptography, ***strength of a crypto algorithm*** is the most important characteristic because it ensures the privacy and confidentiality of the data.

The accuracy of a deep learning model depends heavily on the dataset it has been trained on. Training the same deep neural network on both un-encrypted and encrypted versions of two popular datasets in machine learning classification, MNIST and CIFAR10, will result in more reasonable accuracy evaluation.

Since it involves encryption, the encryption key could be a factor that affects the training data and the accuracy of the model. Training models on datasets that are encrypted by different keys will add more reliability to the measurement process. Datasets encrypted by different keys also provide a better detail to analyze the strength of the crypto algorithm. This study has two main goals:

1. *Measure and compare the accuracy of convolutional neural network models on clean data and encrypted data.*
2. *Evaluate the strength of the orthogonal encryption technique by analyzing encrypted images' patterns and possible vulnerabilities.*

Trained models will be evaluated to report accuracies for analysis by answering questions such as, will encrypted datasets reduce the accuracy of deep learning models? If so, what could be the reason? If not, why encrypting data with orthogonal encryption does not change the accuracy of deep learning models? Will encryption keys affect the accuracy of models trained on the datasets they encrypted? If so, why do encryption keys affect model accuracy? The strength of orthogonal encryption method then will be analyzed by describing the observation of apparent patterns of encrypted images by different dimensions such as the same image encrypted by different keys and different images encrypted by the same key. Known plaintext attack vulnerability will also be analyzed to measure the strength of the orthogonal encryption method since it is a critical attack that every crypto algorithm has to protect against.

3.2 Experimental Design

This section will describe in detail the datasets used in the experiment, MNIST and CIFAR10. The structure of convolutional neural networks used in the experiment will also be described in this section. Lastly, specific training processes will be proved in this section.

3.2.1 Datasets and Tasks

MNIST and CIFAR10 are the two datasets used in this study. MNIST is a handwritten digits dataset that contains 70,000 image samples of 10 digits from 0 to 9. All the images are in gray scale and have a size of 28x28 pixel. The MNIST dataset was constructed by the National Institute of Standards and Technology (NIST) in 1999[3]. According to NIST [3], the dataset contains 70,000-digit images from 500 different writers taken from Census Bureau employees and American high school students. This is one of the most popular datasets that has been used for training in many image processing systems and also the dataset that is used in the patent mentioned above [2] to demonstrate the orthogonal encryption technique.

The CIFAR10 dataset is a subset of an 80 million tiny images dataset constructed by Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton from the University of Toronto. The CIFAR10 dataset contains 60,000 color images of size 32x32 pixel divided equally into 10 classes of completely mutually exclusive objects: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck [4].

3.2.2 Convolutional Neural Network (CNN) Model

A Convolutional Neural Network (CNN) is a deep learning algorithm commonly applied to computer vision. CNN models can take in an input image, assign importance to various aspects/objects in the image and then be able to differentiate one from the other [10]. CNNs are the advanced version of fully connected deep learning networks that can reduce overfitting by taking advantage of the hierarchical pattern in data and using smaller and simpler patterns to assemble more complex patterns. CNN reduces images into a form that is easier to process but still retains all the features of the images to result in a good prediction. Therefore, CNN required much lower pre-processing than other classification algorithms [10]. CNNs are widely used for image classification and recognition because of their high accuracy and low computing cost. Figure 6 shows a sample of a CNN structure.

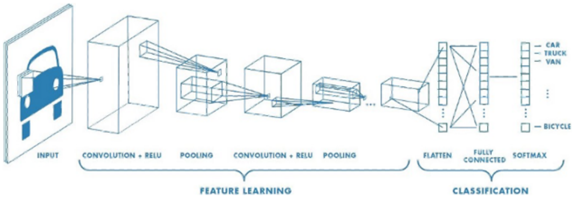


Figure 6: Sample Architecture of Convolutional Neural Network
Source: https://towardsdatascience.com

There are many choices for CNN architecture; this experiment used a typical CNN model with feature extraction by alternating convolution layers with subsampling layers. A subsampling layer is also called a Pooling which is a function that normally immediately follows a convolution layer in a CNN model to down sample the spatial size of output height and width. By doing this, the subsampling layer will reduce the number of parameters to be learned by the network to avoid overfitting during the training process. Therefore, implementing subsampling layers will help minimize the computational complexity of the CNN networks and give higher accuracy.

3.2.3 Experiments

This experiment will first train CNN models with clean images of MNIST and CIFAR10 datasets and then evaluate the models to get accuracies later used as the base to compare with accuracies from models trained by encrypted datasets. In the second task, MNIST and CIFAR10 datasets will be encrypted by 10 different keys using orthogonal encryption, then CNN models will be trained on them, and the accuracies will be analyzed. In the third task, encrypted images will be displayed for observation and analysis to focus on patterns of encrypted images. The last task in this experiment will discuss and demonstrate a possibility of known plaintext attack on orthogonal encryption.

(1) Accuracy on classifying clean examples.

In order to verify CNN models and get the accuracy on clean datasets, both clean MNIST and CIFAR10 datasets are used to train CNN models in this task. The MNIST dataset will be trained with three different CNN models: model C-PX1 is trained with one subsampling layer, model C-PX2 is trained with two subsampling layers, and model C-PX3 is trained with three subsampling layers. These models will be trained on 60,000 train images size 28x28x1 (gray scale images size 28x28 pixels), and each model will be tested with 10,000 test images which have the same size as train images. Max Pooling [5] is used as subsampling layers for CNNs in this experiment. Figure 3 shows an example of Max Pooling implementation on CNN structure.

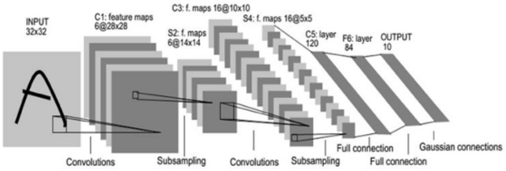


Figure 3: Max Pooling on CNN structure
Source: https://towardsdatascience.com

The CIFAR10 dataset will be also trained on the Convolutional Neural Network that has the same architecture as models trained on the MNIST dataset but will be trained on 60,000 images of size 32x32x3 (color images size 32x32) and tested on 10,000 images having the same size as train images. The accuracy of each model will be then reported.

(2) Accuracy on classifying encrypted examples.

One goal of this study is to evaluate the accuracies of orthogonal encrypted datasets on deep learning models compared to clean datasets. This task will generate ten encrypted datasets from each MNIST and CIFAR10 dataset with ten different encryption keys, and then use them to train on the exact CNN models that trained un-encrypted datasets. The first five keys will be derived from QR decomposition of five randomly selected pictures. The other five keys will be generated by creating 32x32 matrices using the ortho_group class in a python library, SciPy. There are differences in size of images in MNIST and CIFAR10 datasets; images in MNIST are size of 28x28 and images in CIFAR10 are size of 32x32. So, the size of encryption keys for MNIST and CIFAR10 datasets also need to be 28x28 and 32x32 respectively because they have to match with the size of the images in order to perform the encryption calculation. Therefore, the five randomly selected pictures will be resized into 28x28 and 32x32 before performing QR decompositions to get five 28x28 Q matrices for MNIST dataset encryption and five 32x32 Q matrices for CIFAR10 dataset encryption. For the generated encryption matrices, five 32x32 random orthogonal matrices will be generated. For these five matrices, they will be keeping the same size for CIFAR10 dataset encryptions and then be resized into five 28x28 matrices for MNIST dataset encryption. Figure 4 shows 5 selected pictures to use as the key and their corresponding encryption matrices size of 28x28 and 32x32. Figure 5 demonstrates the process of generating 28x28 matrices from 32x32 matrices to use as encryption keys for the MNIST dataset.

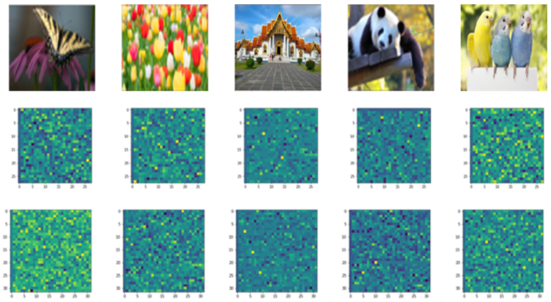
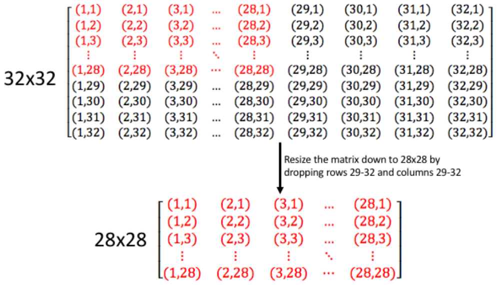


Figure 4: Pictures and their Qs matrices from QR decomposition



$$32\text{x}32 \quad \begin{bmatrix} (1,1) & (2,1) & (3,1) & \cdots & (28,1) & (29,1) & (30,1) & (31,1) & (32,1) \\ (1,2) & (2,2) & (3,2) & \cdots & (28,2) & (29,2) & (30,2) & (31,2) & (32,2) \\ (1,3) & (2,3) & (3,3) & \cdots & (28,3) & (29,3) & (30,3) & (31,3) & (32,3) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ (1,28) & (2,28) & (3,28) & \cdots & (28,28) & (29,28) & (30,28) & (31,28) & (32,28) \\ (1,29) & (2,29) & (3,29) & \cdots & (28,29) & (29,29) & (30,29) & (31,29) & (32,29) \\ (1,30) & (2,30) & (3,30) & \cdots & (28,30) & (29,30) & (30,30) & (31,30) & (32,30) \\ (1,31) & (2,31) & (3,31) & \cdots & (28,31) & (29,31) & (30,31) & (31,31) & (32,31) \\ (1,32) & (2,32) & (3,32) & \cdots & (28,32) & (29,32) & (30,32) & (31,32) & (32,32) \end{bmatrix}$$

Resize the matrix down to 28x28 by dropping rows 29-32 and columns 29-32

$$28\text{x}28 \quad \begin{bmatrix} (1,1) & (2,1) & (3,1) & \cdots & (28,1) \\ (1,2) & (2,2) & (3,2) & \cdots & (28,2) \\ (1,3) & (2,3) & (3,3) & \cdots & (28,3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (1,28) & (2,28) & (3,28) & \cdots & (28,28) \end{bmatrix}$$

Figure 5: Matric 32x32 to 28x28

The reason for processing the key like this is to keep the keys that are used to encrypt MNIST and CIFAR10 datasets as similar as possible even if the keys have to be different in size. This will ensure that all the images are performed by the same transformation during the encryption process.

After finishing encrypting, the datasets then are used to train on CNN deep neural networks that have the same architectures as networks used to train un-encrypted datasets. The training process will end up with 20 trained models: ten are trained with encrypted MNIST datasets and ten are trained with encrypted CIFAR10 datasets. The accuracy of each model will be reported.

(3) Visual analysis of encrypted examples.

Even with encrypted images, the patterns appearing on them can still provide a lot of information. Analyzing samples of encrypted images may show some patterns and characteristics related to the strength of orthogonal encryption. This section will use some sample encrypted images to perform some comparison and analysis.

The first task will analyze encrypted images from the same object that are encrypted by 10 different keys. Specifically, each digit in the MNIST dataset and each object in the CIFAR10 dataset will be encrypted with 10 different encryption keys and then displayed for comparison.

The second task will analyze encrypted images from different objects that are encrypted by the same key. In this task, each encryption key will be used to encrypt 10 different digits in the MNIST dataset and 10 different objects in the CIFAR10 dataset.

The third task in this section will analyze encrypted images from different objects that are encrypted by different keys. Each digit in the MNIST dataset will be encrypted by one of the 10 keys and the same process will be applied to the CIFAR10 dataset. The result then will be displayed for analysis.

(4) Possibility of known plaintext attacks.

A known plaintext attack is a type of cryptanalysis in which the attacker knows at least one pair of clear images (plain images) and encrypted images (cipher images) then uses this to reveal secret information such as an encryption key. Encrypted archive ZIP files are a well-known example that is prone to known plaintext attacks. There are many types of software available on the internet that help determine the key needed to decrypt the archived files [11]. In image encryption (digital encryption), various cryptanalysis attacks have been introduced such as in [12][13][14][15][16]. According to the schema introduced in [11], a single-pixel encrypted system is totally vulnerable to known plaintext attack.

Known plaintext attacks are critical to any crypto algorithms. According to previous works on digital encryption, known plaintext attacks succeed on many digital encryption methods. This task will perform some analysis of the Orthogonal Encryption to find possible vulnerabilities. The orthogonal encryption technique being investigated in this study works based on the following function: C = P * K where C is the cipher image, P is the plain image, and K is the orthogonal matrix used as the encryption key. The decryption key D will be the inverse of K. Theoretically, it is possible to derive the encryption key K if C and P are known. K can be derived by the equation K = C * P-1 where K is the encryption key, C is the cipher image, and P-1 is the inverse of the plain image. We may not be able to directly find P-1 from P if P is a singular matrix because the singular matrix does not have any inverse [6]. According to [7], any singular matrix could be converted to a non-singular matrix. Matrices P are the presentation of images, so the singular characteristic of matrix P depends heavily on the image that it represents. If the dataset contains matrices that represent gray scale images, there will be more singular matrices P than in the dataset that contains matrices representing color images. To test these hypotheses, the first task in this section will use a python script to analyze and report the number of singular and non-singular matrices contained on MNIST and CIFAR10 datasets. The second task will use some found non-singular matrices to derive the key by using the equation $K = C * P^{-1}$. Encryption key K' will then be reversed to find the decryption key D' to decrypt some images and compare resulting images with images decrypted by decryption key D.

4 Experiemntal Results and Analysis

This section will present the results of the experiment designed in section 4.2. Specifically, section 5.1 will present the results of the experiment performed on MNIST dataset; section 5.2 will present the results of the experiment on the CIFAR10 dataset; lastly section 5.3 will give a brief comparison, analysis and summary of all results.

4.1 Results on the MNIST dataset.

(1) Accuracy on classifying clean examples.

As mentioned in section 3.2.3(1), the clear MNIST dataset is trained with three versions of CNN networks (one subsampling layer, two subsampling layers, and three subsampling layers). Table 1 below shows the models' accuracies trained on 50,000 images and tested on 10,000 images from MNIST dataset. Overall, all three versions got 99% on accuracy; Two and three subsampling layer models got approximately 0.3% higher in accuracy compared to one subsampling layer model. This occurs because the two subsampling layer and three subsampling layer models can reduce overfitting during the training process.

Table 1: Accuracy on classifying clean MNIST dataset.

One subsampling layer	Two subsampling layers	Three subsampling layers
0.9900	0.9930	0.9937

(2) Accuracy on classifying encrypted examples.

Table 2 below shows the accuracies of ten CNN models trained on ten encrypted MNIST datasets that are encrypted by ten different encryption keys. Key1 to key5 are retrieved from performing QR decompositions on five randomly selected images; key6 to key10 are retrieved by orthogonal matrix generating using Scipy library in Python. Overall, the two and three subsampling layer models have higher accuracy than the one subsampling layer model. This is the same as the model trained on unencrypted data. The two and three subsampling layer models have approximately 0.2% higher accuracy than the one subsampling layer models. On average, models trained on datasets encrypted by QR decomposition keys have higher accuracy than those trained on datasets encrypted by orthogonal matrix generating keys (approximately 0.2%). The average accuracy of models trained on encrypted datasets is approximately 1% lower than models trained on a clean MNIST dataset. Because orthogonal encryption works based on matrix multiplication, in each 28x28 pixel encrypted image there are 43,120 multiplication and addition operations performed. Each pixel on an encrypted image is determined by 28 multiplications and 27 additions; each image contains 28x28=784 pixel, so each encrypted image will need 43,120 operations ([28 + 27]*784 = 43,120). Since all images in the dataset are encrypted by the same key, the more operations are performed, the more they make cipher images of the dataset closer to each other. This is a reasonable explanation for why the accuracy of models trained on encrypted datasets are a little bit lower than those models trained on clean datasets.

(3) Visual analysis of encrypted examples.

(3)(1) Same object encrypted by different keys

This task uses three sample digits from MNIST dataset. Figure 7 shows a clean image of digit 2. Figure 8 shows the encrypted images of digit 2 using 10 different keys designed for use in this experiment. Figure 9 and Figure 10 show un-encrypted and encrypted images of digit 3. Figure 11 and Figure 12 are the representations of digit 8. ***These samples show that the same object encrypted by different keys produces different cipher images, but there are still patterns of the un-encrypted images passed on those cipher images***. Focusing on those yellow pixels of cipher images in Figure 8, the shape of digit 2 is still presented in cipher images of digit 2 in Figure 10. This characteristic is also presented on cipher images of digit 3 and digit 8. The cipher images 2, 3, 4, 5, and 8 in Figure 10 show 2 shapes of digit 3 represented by yellow pixels; this could be generated during the orthogonal transformation process. The same characteristic and pattern are also presented in the cipher images of digit 8 in Figure 12. More demonstrations of other digits are presented in Appendix C. ***This shows that the orthogonal encryption technique does not completely hide all the information of the original image; There are common patterns from cipher images of the same object even if they are encrypted by different keys presented***.

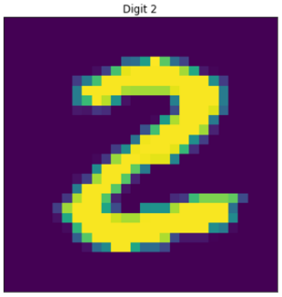


Figure 7: Un-encrypted digit 2

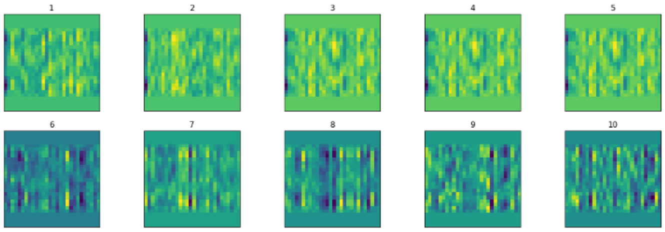


Figure 8: Digit 2 encrypted by 10 different encryption keys

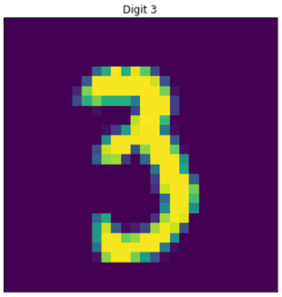


Figure 9: Un-encrypted digit 3

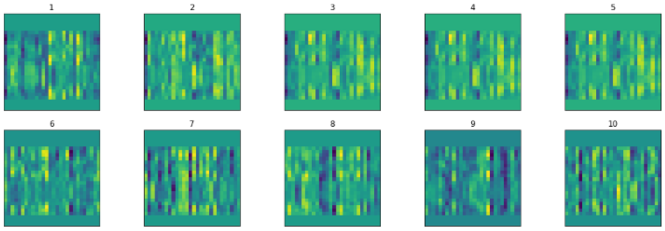


Figure 10: Digit 3 encrypted by 10 different encryption keys

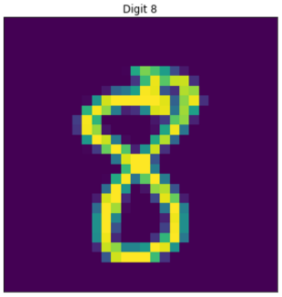


Figure 9: Un-encrypted digit 8

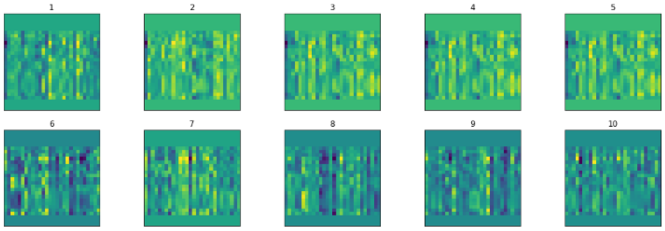


Figure 12: Digit 8 encrypted by 10 different encryption keys

(4) Possibility of known plaintext attacks.

4.2 Results on the CIFAR10 dataset.

This experiment will first train CNN models with clean images of MNIST and CIFAR10 datasets and then evaluate the models to get accuracies later used as the base to compare with accuracies from models trained by encrypted datasets. In the second task, MNIST and CIFAR10 datasets will be encrypted by 10 different keys using orthogonal encryption, then CNN models will be trained on them, and the accuracies will be analyzed. In the third task, encrypted images will be displayed for observation and analysis to focus on patterns of encrypted images. The last task in this experiment will discuss and demonstrate a possibility of known plaintext attack on orthogonal encryption.

(1) Accuracy on classifying clean examples.

(2) Accuracy on classifying encrypted examples.

(3) Visual analysis of encrypted examples.

(4) Possibility of known plaintext attacks.

4.3 Summary of thr Results.

This experiment will first train CNN models with clean images of MNIST and CIFAR10 datasets and then evaluate the models to get accuracies later used as the base to compare with accuracies from models trained by encrypted datasets. In the second task, MNIST and CIFAR10 datasets will be encrypted by 10 different keys using orthogonal encryption, then CNN models will be trained on them, and the accuracies will be analyzed. In the third task, encrypted images will be displayed for observation and analysis to focus on patterns of encrypted images. The last task in this experiment will discuss and demonstrate a possibility of known plaintext attack on orthogonal encryption.

5 Conclusion and Future Work

The biggest challenge and limitation for this experiment was the fact that the router used was connected to the Colorado School of Mines network. The network setup was required to follow the school's network security policies, and this caused a number of different issues. Initially, there were technical difficulties with getting the router up and running, which caused delays in getting started. Beyond that, there are also a number of restrictions as to what can be done on the Mines network. The Mines network is much more protected and secure than an average home environment where many IoT devices can be found. Many tools and attacks that we wanted to use could have been performed on a normal home network but were impossible on the Mines network. For example, the IT department (ITS) restricts certain types of traffic like ICMP when it comes from an internal router. In addition, since our router was connected to the Mines network, we were hesitant and at times unwilling to perform certain kinds of attacks that had the potential to go beyond the lab and into the overall network.

Another big challenge had to do with scheduling. Initially, we planned to alternate each team's access to the lab every week. However, splitting up the lab time between the red and blue teams like this made it difficult for each team to be able to accomplish their goals in a timely manner. Therefore, about halfway through the experiment, it was decided that the lab space and time would be shared and either team could work at the lab whenever they wanted or needed to. This greatly increased productivity, but at the cost of privacy and secrecy. By sharing the lab time, both teams were often working at the same time and therefore each team knew what the other team was working on. Therefore, the red team often knew what vulnerabilities the blue team had not yet secured against and the blue team often knew what attacks the red team was planning. While in some ways this speed up the experiment as the teams were cooperating, it also might have added a certain level of dependence to the experiment. Since the red team was focused on exploiting the vulnerabilities that we knew had not been taken care of, it is possible that we focused less on coming up with new and unusual ideas that were not related to what the blue team was doing.

References

[1] Z. Zhang, M. C. Y. Cho, C. Wang, C. Hsu, C. Chen and S. Shieh, "IoT Security: Ongoing Challenges and Research Opportunities," *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, Matsue, 2014, pp. 230-234.

[2] R. Mahmoud, T. Yousuf, F. Aloul and I. Zualkernan, "Internet of things (IoT) security: Current status, challenges and prospective measures," *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, London, 2015, pp. 336-341.

[3] C. Lee, L. Zappaterra, Kwanghee Choi and Hyeong-Ah Choi, "Securing smart home: Technologies, security challenges, and security requirements," *2014 IEEE Conference on Communications and Network Security*, San Francisco, CA, 2014, pp. 67-72.

[4] A. Dorri, S. S. Kanhere, R. Jurdak and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Kona, HI, 2017, pp. 618-623.

[5] Kang, W.M., Moon, S.Y. & Park, J.H. An enhanced security framework for home appliances in smart home. *Hum. Cent. Comput. Inf. Sci.* 7, 6 (2017) doi:10.1186/s13673-017-0087-4

[6] Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J.A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., & Zhou, Y. (2017). Understanding the Mirai Botnet. *USENIX Security Symposium.*

[7] Mateti, Prabhaker (2005), Hacking Techniques in Wireless Networks: Forged Deauthentication, Department of Computer Science and Engineering, Wright State University.

[8] Bellardo, John; Savage, Stefan (2003-05-16), "802.11 Denial-of-Service Attacks: Real Vulnerabilities and Practical Solutions", Proceedings of the USENIX Security Symposium, Aug 2003