LED: Learnable Encryption with Deniability

Zhe-Wei Lin[1], Tzu-Hung Liu[1], and Po-Wen Chi[1][0000−0001−5663−078X]

Department of Computer Science and Information Engineering,
National Taiwan Normal University
60947043s@gapps.ntnu.edu.tw, 40747031s@gapps.ntnu.edu.tw,
neokent@gapps.ntnu.edu.tw

Abstract. User privacy is an important issue in the cloud machine learning service. In this paper, we raise a new threat about the online machine learning service, which comes from outside superior authority. The authority may ask the user and the cloud to disclose secrets and the authority can monitor the user behavior. We propose a protection approach called learnable encryption with deniability (LED), which can convince the outsider of the fake data and can protect the user privacy.

Keywords: privacy-preserving machine learning · learnable encryption · deniable encryption

1 Introduction

Machine learning has gradually become mainstream in recent years. According to its diversified development, model training needs to consume more and more resources. Therefore cloud services become an option for people to train their models. Also, the change in usage and service types let the users take predict services through it. With cloud services, everyone can train, update the model and predict at any time and anywhere.

Nevertheless, with the increase in machine learning based on cloud services, user privacy caught much attention. For example, when a user wants to use the training service, it needs to upload the training data to the cloud service provider (CSP) and undoubtedly, the training data is leaked to the CSP. Moreover, when making a prediction query, the queried data is also known to the CSP and the user loses its privacy. To solve this privacy issue, there are some research fields which are proposed. The first one is the integration of the machine learning service and homomorphic encryption [1, 2, 3]. This kind of approach simulates a prediction process as a circuit and run this circuit directly over encrypted queried data. So the query answer is also in the encrypted form. However, this approach cannot protect the training data. The other solution is called learnable encryption [4, 5]. This approach trains the encrypted data directly and builds models over encrypted data. So the encrypted query can be put into the model and the prediction result is derived. This approach is specific to images, where the encryption is based on block scrambling. By scrambling image blocks, the image is kept secret to human but the characteristics can still be found through machine learning.

However, except the above issue which focuses on the CSP, in this era, there is another kind of privacy issue raised by the superior authority. The superior authority generally has power to monitor user behaviors, including learning and prediction [6]. Even the data is encrypted, the authority can force the user and the CSP to provide secret keys lawfully and common encryption schemes do not work. To preserve user privacy in this scenario, we propose Learnable Encryption with Deniability (LED). With learnable encryption, our scheme can protect data sets and predictive queries from the CSP. As for deniability, LED makes a user to submit fake keys to the outside coercer and to mislead it to a fake behavior, keeping the real behavior secret.

Our contributions about this work is as follows.

- **Learnable encryption with deniability**: LED can make a learnable encryption model have deniability, and it can protect the privacy from superior coercion.
- **Prediction accuracy enhancement**: We predict query with distributional multi-models that make accuracy enhanced.
- **Performance evaluation**: The results are experimentally verified and can also meet expectations.

The rest of this paper is organized as below, we will first introduce related work in section 2, followed by the scheme and technique of LED in section 3, and provide experimental results and explanations in section 4. Finally, in section 5 we will present our conclusions and future works.

2 Related Works

2.1 Privacy-Preserving Machine Learning Schemes

Machine learning is a very powerful tool. In recent years, many cloud services have been launched to allow users to use their machines for training and prediction. Users will provide their training data to cloud service providers, and users' data privacy is An important issue, when users' data privacy is not secure, users will not use cloud training services, and people will lose the benefits of cloud services. This problem is also called *privacy preserving data-mining* by Agrawal and Srikant [7]. To solve this problem means balancing cloud service provider and user data privacy, there are lots of research works and we will introduce some of them.

One kind of solutions is based on fully homomorphic encryption. Homomorphic encryption makes a user can operate data with any circuits in the encryption form. With this feature, it is possible to run a neural network-like network on the encrypted data. Bos et al. [8], Dowlin et al. [9] used this concept to develop a privacy-preserving image prediction service called *CryptoNet*. First, they used lots of images in the plaintext form as the training set and derived a model. Then, they implemented the prediction process with the homomorphic encryption technique. Since the homomorphic encryption is not computationally

efficient enough, Dowlin et al. found some properties of the prediction circuit, simplified the prediction process and therefore they could only use some homomorphic operations to improve the service performance. The drawback of CryptoNet is that it cannot protect the privacy of the training dataset because training is operated in the plaintext form. Chabanne et al [10]. proposed another enhancement approach. They used Taylor series as the approximating function and could provide better performance. However, the training data is still open to the service provider.

Another solution is based on the Secure Multi-Party Computation (SMC). The data owner and the cloud service provider work together in the training and prediction phases. Due to the characteristics of SMC, the cloud service provider has no knowledge of the user input data during the calculation. Yao's garbled circuits [11] and the Goldreich-Micali-Wigderson secret-sharing protocol [12] are both SMC technologies. Many studies are based on these two technologies. Rouhani et al. developed a method called DeepSecure based on garbled circuits [13]. Mohassel et al. used Mohassel et al. applied a similar idea with additive homomorphic encryption to speed up operations [14] Liu et al. used a lattice-based additive homomorphic encryption to generate multiplication triples for multiparty computation [15]. Generally speaking, SMC's solution requires frequent interactions between users and cloud service providers, especially for a complex operation like training and prediction. Therefore, it is not a practical solution.

2.2 Learnable Image Encryption

Unlike homomorphic encryption schemes that can process encryption data, Tanaka[4] and Kiya[16, 17] proposed another concept called **learnable image encryption**. This approach is to train and to predict directly over encrypted data instead of considering their plaintext. Generally, a data which is encrypted means that it is randomized and therefore, it is hard to find the patterns. However, Tanaka and Kiya found that if a image is encrypted in a static scrambling transformation, the figure cannot be recognized by the outsider and the figure's characteristics is still maintained.

This study main introduction Tanaka's proposed Learnable Image Encryption [4]. Encryption image encryption processing flow is shown in Figure 1. First, the 8bit-RGB image is divided into blocks of $M * M$ pixels, and the second step is to divide the image in each block. The pixel is divided into the first 4 bits and the last 4 bits to obtain 6-channel image blocks. The intensity of the positions of the randomly selected pixels in the third step is reversed. Finally, the pixels in the blocks are shuffled and each block is combined to obtain our encrypted image.

2.3 Deniable Encryption

The concept of deniable encryption was originally proposed in [18]. The main feature about deniable encryption is that one ciphertext can be opened to a fake

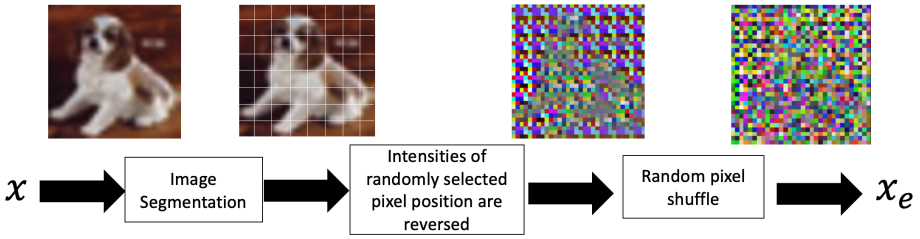


Fig. 1. learnable image encryption process[4]

data instead of the original message. Generally speaking, when being coerced, the sender and the receiver will claim the fake with providing convincing proofs. In this paper, we apply this idea on the learnable encryption scheme.

In this work, we make use of two deniable techniques. The first one is proposed by by Paolo Gasti et al[19]. Paolo Gasti et al. creates a redundant space in each ciphertext. The redundant space is claimed to be a random string, where actually is filled with the encryption result of a fake message. When the user is asked to open the encrypted data, the user can open the message in the redundant space so that the outsider will get the fake data. The other technique is called multi-distributional deniable technique, which was first proposed by Waters [20]. Watsers uses two sets of algorithms, one is normal and is claimed to be used while the other one is actually used. The algorithm outputs of corresponding algorithm outputs are computationally indistinguishable. Therefore, the outsider cannot challenge the user's claim and the user can do different things in these two sets of algorithms.

3 Learnable Encryption with Deniability

3.1 Scenario

Here we describe the user scenario, which is shown in figure 2. The user wants to predict the label of a given image, which is a dog in figure 2. The user first prepares another fake image and form a new image by combining these two images. The user also use a guiding image to indicate the location of the real wanted image. In figure 2, we use a triangle to show the real image is on the first part of the image. Then, the user encrypts this composite image, including the guiding image, which is encrypted by a pre-shared key between the user and the CSP, the real image and the fake image, which are encrypted learnably, and sends this encrypted query to the CSP. After receiving the query, the CSP decrypts the guiding image, and answer the prediction result to the user.

When being coerced, we assume that the outside coercer gets the encrypted query and asks the user and the CSP to open this query. In this case, the user and the CSP will convince the coercer that the guiding image is a rectangle, which implies the real image is on the second part. After the sencond image is

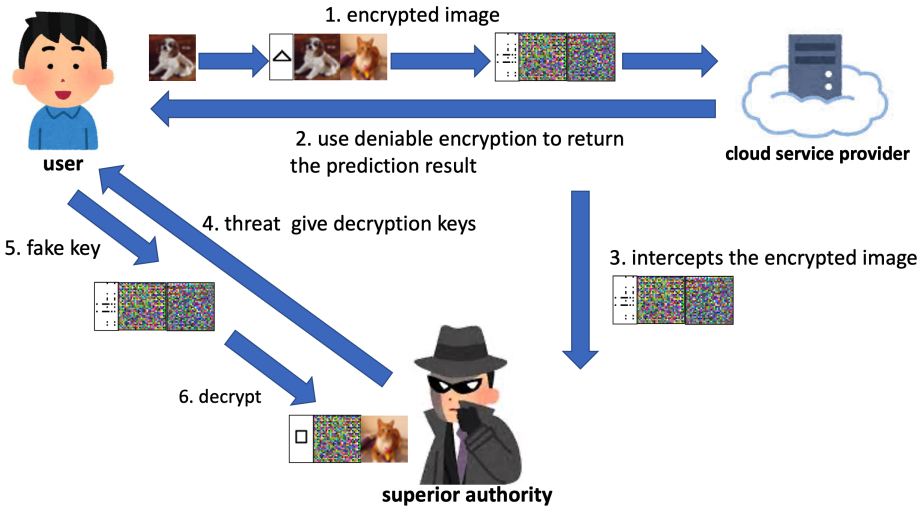


Fig. 2. user scenario

opened, the coercer will see a cat and the dog, which is the real image, is kept secret.

In this scenario, we assume that the CSP is honest-but-curious. This means that the CSP will correctly operate all step defined in our scheme. Because all machine learning operations are over encrypted data, the CSP can know nothing about the training data and prediction queries.

3.2 Leanable Encryption with Deniability

In this section, we show how to provide deniability over learnable encryption. We use a deniable technique called multi-distributional deniability. We build two sets of prediction processes, one is normal, which is claimed to be used, and the other is deniable, which is actually used. We want to make sure that the prediction queries and the opened keys in these two processes are computationally indistinguishable. Therefore, the outside coercer cannot challenge the user claim.

First, we will see how the normal prediction process works, where the process is shown in figure 3. First, the user learnably encrypts the training data with a secret key k and uploads it to the CSP for building a model. Note that the CSP does not know k and has no idea about the training dataset. When a user wants to use an online image prediction service for a target image, the user generates a random image with the same size as the target image. The user flips a coin b and form a new image $I = I_0||I_1$, where I_b is the learnably encrypted target image by k and I_{1-b} is the random image. The user then generates a guiding image to represent b and scrambles the guiding image with a key sk. sk is a pre-shared key between the user and the CSP. The prediction query will be the scrambled guiding image and a composite image. When receiving the prediction query, the

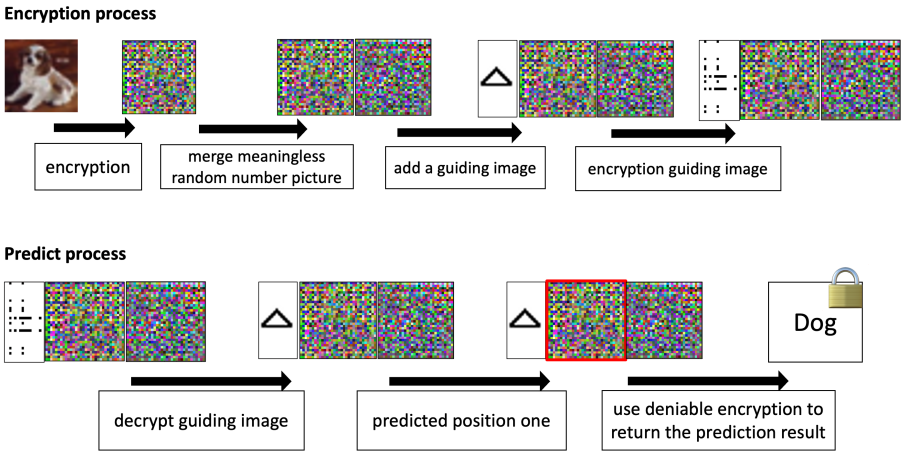


Fig. 3. Normal prediction process

CSP uses sk to recover the guiding image and gets b, finding the correct location of the encrypted target image. The CSP predicts the encrypted target image through the pre-trained model. Next, the CSP answers the query by the answer which is deniable ciphertext from the prediction label and a random label. The user can correctly decrypt the image label by decrypting the answer.

Now, we will see how the deniable prediction process works, where the process is shown in figure 4. First, the user learnably encrypts the training data with two secret keys k_0, k_1 and uploads them to the CSP for building models. Note that the CSP does not know k_0, k_1 and has no idea about the training datasets. When a user wants to use an online image prediction service for a target image, instead of using a random image, the user prepares a fake image with the same size as the target image. The user flips a coin b and form a new image $I = I_0||I_1$, where I_b is the learnably encrypted target image by k_b and I_{1-b} is the learnably encrypted fake image by k_{1-b}. The user then generates a guiding image to represent b and scrambles the guiding image with a key sk. sk is a pre-shared key between the user and the CSP. The prediction query will be the scrambled guiding image and a composite image. When receiving the prediction query, the CSP uses sk to recover the guiding image and gets b, finding the correct location of the encrypted target image. The CSP predicts the encrypted target image and fake image through the corresponding models. Next, the CSP answers the query by the answer which is deniable ciphertext from the prediction labels, where one is the target label and the other is the fake label. The user can correctly decrypt the image label by decrypting the answer.

When being coerced, the user should provide sk, k to answer the outside coercer. If the prediction process is deniable and the user wants to hide the queried image, the user can instead provide $sk\prime$ and k_{1-b} and mislead the coercer

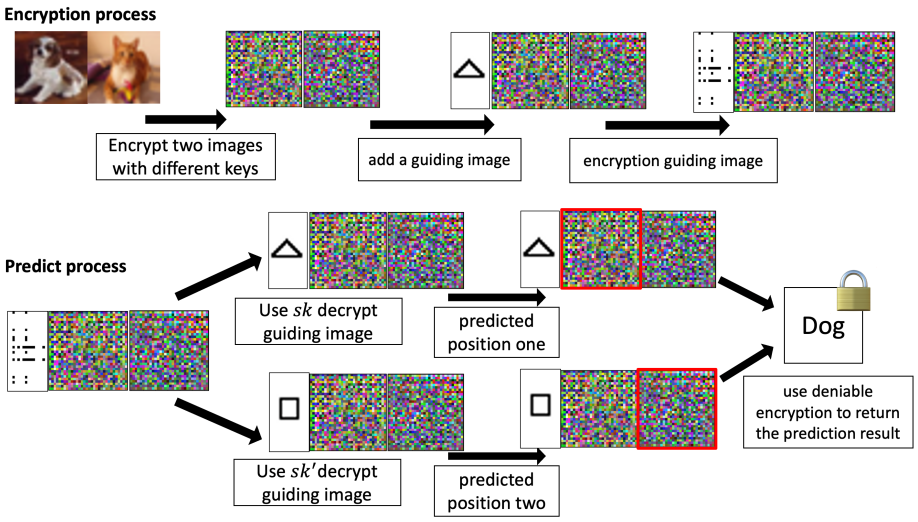


Fig. 4. Deniable prediction process

that the user queries the fake image. The fake scrambling key $sk\prime$ is a fake key to convince the coercer that the target image is in the part $1-b$. The detail about $sk\prime$ generation is described in the next subsection. Since sk and k are both can be treated as a scrambling mapping, sk,$sk\prime$ and k_0,k_1 are definitely indistinguishable. Therefore, the coercer has no reason to reject the user claim.

3.3 Deniable Key Generation

In this section, we describe how to generate a deniable key sk'. As mentioned in the previous subsection, we scramble a guide image to a garbled image by sk, which is a scrambling mapping key. Now, we want to find a fake key sk', which can recover the garbled guiding image to a fake guiding image. For example, sk is a key that can scramble a triangle to a guiding image and sk' can be used to recover the garbled guiding image to a square as the fake indication shown as Figure 4. The steps are as follows, First, we can find the color with the highest proportion of pixels in the garbled image. Then we reshape those pixels into a square. After randomly arranging the rest of the pixels, it will become the fake guide image. The mapping relation will be the deniable key, sk'.

3.4 Prediction Accuracy Enhancement

According to Learnable Image Encryption's work [4], the prediction accuracy of learnable encryption is around Learnable Image Encryption. However, in our experiment, which will be described in section 4.2, the prediction accuracy of

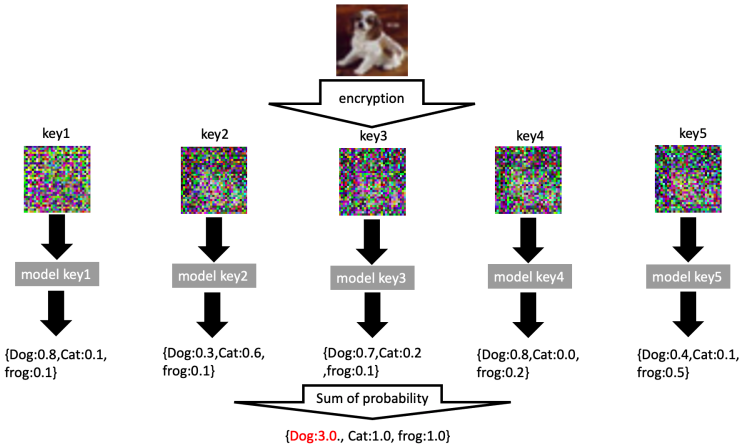


Fig. 5. Multi-models prediction (probabilities in this figure are examples)

our learnable encryption implementation[1] is only around 0.75, which is not acceptable for the practical use. Therefore, we propose an enhancement approach to improve the prediction accuracy.

As shown in Figure 5, Our idea is based on the dependable computing. For each training image, we make the image encrypted n times. So we can get n training sets and derive n models, where each model has its own keys. Then n models predict n results and then sum the predicted probabilities of each model, and the highest probability obtained is our predicted result

4 Evaluation

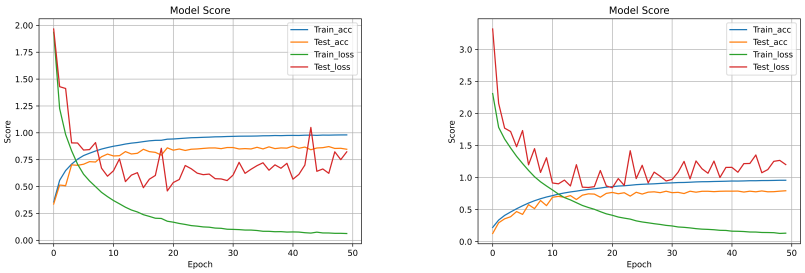
In this section, we make some experiments to evaluate the influences caused by learnable encryption and multi-models prediction. We mainly used the cifer10 data set. Its pre-processing part only performs normalization to inputs and one hot encoding to categories. The encryption method is block-wise scrambling proposed by learnable encryption [5]. All experiments are training with 50 epochs, and the batch size is set to 64.

4.1 Experiment Schemes

In the following experiments, we will set different parameters to contrast:

1. The model will be set with two different structures to check the consistency. One is a general pyramidal residual network (ResNet) same as learnable encryption. The other will be a simple CNN model with 8 convolution layers which is smaller than the ResNet(half as many params as ResNet).

[1] Our implementation is based on Learnable Image Encryption work without the deniability feature.



(a) Experiment 1 (b) Experiment 2

Fig. 6. Training history

Table 1. VALIDATION ACCURACIES OF CIFAR DATASET

Accuracies	ResNet	CNN
Plain images	0.8469	0.8393
Encrypted images	0.7788	0.7469
Multi-models prediction	0.8507	0.8289

2. The data set have two types. One is original cifer10 but only does the preprocessing, the other one is encrypted by block-wise scrambling. When training with encrypted data, we will put a block-wise adaptation network before the residual and CNN networks. It can measure the loss of feature and accuracy caused by learnable encryption.
3. We will use the method proposed in this paper. Training 20 models with data sets encrypted by different keys. Then using the prediction to vote the results. And we will check the difference between voting by the predicted label of each model or summing up all the models' predictive probability for each label before outputting the predicted label. And the effect from the model of the number will also perform.

4.2 LED Prediction Accuracy

The training process is shown in Figure 6. In Figure 6(a) is the ResNet train with plain images. We can find the training loss went down belong a perfect curve and the training accuracy was close to 1, but the testing accuracy didn't go up after about the 10th epoch. It shows that may have an overfitting problem. And in Figure 6(b), although the data set used here is encrypted, the training process seems to be the same as Figure 6(a). The only difference is the accuracy and loss of testing are worse than before. And there are the same circumstances when it comes to CNN. Moreover, we found that a smaller batch- size can make better accuracy, but we still set it to 64 for efficiency.

The results listed in the table are the comparison between the accuracy of two model training by plain images and encrypted images. And the result of multi-

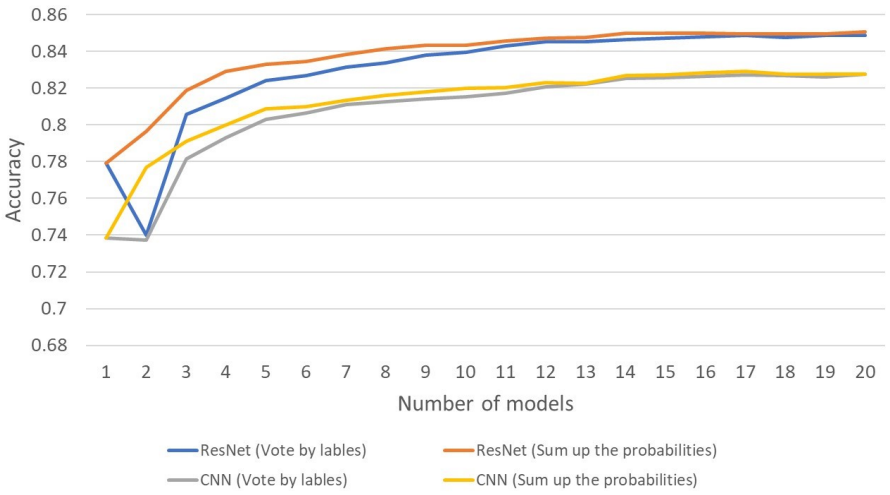


Fig. 7. Model Number vs. Prediction Accuracy.

models prediction used 20 models to perform. Having an accuracy of 0.847 and 0.840 when training with plain images, the accuracy using learnable encryption reduces by almost 0.1. But with the method we proposed, multi-model prediction, can reduce the feature loss caused by learnable encryption, and make the accuracy return to the almost same level as training with the plain image. And it works on both structures of models.

4.3 Accuracy vs. Number of Models

We also check the impact of the model number. Figure 7 is the evaluation result. We can see that with the model number growing, the prediction accuracy gets improved. And if there are 10 or more models, there are almost the same between voting by labels and summing up the probabilities. Otherwise, if using fewer models, summing up the probabilities will be better than voting by labels. Although the more models made the higher accuracy, it costs more resources. At least 3 or 4 models can make accuracy leap forward, and it will almost reach the limit when using 6 or 7 models. For efficiency, if not emphasized in extremely highest accuracy, 3 to 6 models can be enough to recover the loss. However, the trade-off is the prediction size also grows linearly. Because user privacy is very important, we think that the cost is affordable.

5 Conclusion

In this paper, we proposed new privacy-preserving machine learning service called learnable encryption with deniability. This scheme can not only protect

user data from the CSP, but also keep privacy from being coerced. We also propose a multi-models technique to keep privacy and prediction accuracy at the same time.

Currently, this approach can only protect a single prediction query. That is, if the coercer collects lots of past queries, the proof will not be consistent. Therefore, the user needs to change the whole setting at each query. Our next step is to tackle this problem.

References

[1] Joon-Woo Lee et al. "Privacy-Preserving Machine Learning with Fully Homomorphic Encryption for Deep Neural Network". In: *CoRR* abs/2106.07229 (2021). arXiv: 2106.07229. URL: https://arxiv.org/abs/2106.07229.

[2] Robert Podschwadt, Daniel Takabi, and Peizhao Hu. "SoK: Privacy-preserving Deep Learning with Homomorphic Encryption". In: *CoRR* abs/2112.12855 (2021). arXiv: 2112.12855. URL: https://arxiv.org/abs/2112.12855.

[3] Syed Imtiaz Ahamed and Vadlamani Ravi. *Privacy-Preserving Chaotic Extreme Learning Machine with Fully Homomorphic Encryption*. 2022. DOI: 10.48550/ARXIV.2208.02587. URL: https://arxiv.org/abs/2208.02587.

[4] Masayuki Tanaka. "Learnable Image Encryption". In: *CoRR* abs/1804.00490 (2018). arXiv: 1804.00490. URL: http://arxiv.org/abs/1804.00490.

[5] Koki Madono et al. "Block-wise Scrambled Image Recognition Using Adaptation Network". In: *CoRR* abs/2001.07761 (2020). arXiv: 2001.07761. URL: https://arxiv.org/abs/2001.07761.

[6] Wikipedia contributors. *Internet censorship in China — Wikipedia, The Free Encyclopedia*. [Online; accessed 15-September-2022]. 2022. URL: https://en.wikipedia.org/w/index.php?title=Internet_censorship_in_China&oldid=1110094504.

[7] Rakesh Agrawal and Ramakrishnan Srikant. "Privacy-preserving Data Mining". In: *SIGMOD Rec.* 29.2 (May 2000), pp. 439–450. ISSN: 0163-5808. DOI: 10.1145/335191.335438. URL: http://doi.acm.org/10.1145/335191.335438.

[8] Joppe W. Bos et al. "Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme". In: *Cryptography and Coding*. Ed. by Martijn Stam. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 45–64. ISBN: 978-3-642-45239-0.

[9] Ran Gilad-Bachrach et al. "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy". In: *International Conference on Machine Learning*. 2016, pp. 201–210.

[10] Hervé Chabanne et al. *Privacy-Preserving Classification on Deep Neural Network*. Cryptology ePrint Archive, Report 2017/035. https://eprint.iacr.org/2017/035. 2017.

[11] A. C. Yao. "How to generate and exchange secrets". In: *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*. Oct. 1986, pp. 162–167. DOI: 10.1109/SFCS.1986.25.

[12] O. Goldreich, S. Micali, and A. Wigderson. "How to Play ANY Mental Game". In: *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*. STOC '87. New York, New York, USA: ACM, 1987, pp. 218–229. ISBN: 0-89791-221-7. DOI: 10.1145/28395.28420. URL: http://doi.acm.org/10.1145/28395.28420.

[13] Bita Darvish Rouhani, M. Sadegh Riazi, and Farinaz Koushanfar. "Deepsecure: Scalable Provably-secure Deep Learning". In: *Proceedings of the 55th Annual Design Automation Conference*. DAC '18. San Francisco, California: ACM, 2018, 2:1–2:6. ISBN: 978-1-4503-5700-5. DOI: 10.1145/3195970.3196023. URL: http://doi.acm.org/10.1145/3195970.3196023.

[14] P. Mohassel and Y. Zhang. "SecureML: A System for Scalable Privacy-Preserving Machine Learning". In: *2017 IEEE Symposium on Security and Privacy (SP)*. May 2017, pp. 19–38. DOI: 10.1109/SP.2017.12.

[15] Jian Liu et al. "Oblivious Neural Network Predictions via MiniONN Transformations". In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. CCS '17. Dallas, Texas, USA: ACM, 2017, pp. 619–631. ISBN: 978-1-4503-4946-8. DOI: 10.1145/3133956.3134056. URL: http://doi.acm.org/10.1145/3133956.3134056.

[16] Hitoshi Kiya. "Compressible and Learnable Encryption for Untrusted Cloud Environments". In: *CoRR* abs/1811.10254 (2018). arXiv: 1811.10254. URL: http://arxiv.org/abs/1811.10254.

[17] G. Chen et al. "Encrypted Image Feature Extraction by Privacy-Preserving MFS". In: *2018 7th International Conference on Digital Home (ICDH)*. Nov. 2018, pp. 42–45. DOI: 10.1109/ICDH.2018.00016.

[18] Ran Canetti et al. "Deniable Encryption". In: *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*. Ed. by Burton S. Kaliski Jr. Vol. 1294. Lecture Notes in Computer Science. Springer, 1997, pp. 90–104. DOI: 10.1007/BFb0052229. URL: https://doi.org/10.1007/BFb0052229.

[19] Paolo Gasti, Giuseppe Ateniese, and Marina Blanton. "Deniable cloud storage: sharing files via public-key deniability". In: *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society*. 2010, pp. 31–42.

[20] Adam O'Neill, Chris Peikert, and Brent Waters. "Bi-Deniable Public-Key Encryption". In: *Advances in Cryptology – CRYPTO 2011*. Ed. by Phillip Rogaway. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 525–542. ISBN: 978-3-642-22792-9.