An Analysis of the Security of Compressed Sensing Using an Artificial Neural Network

Shadan Ghaffaripour, Fadi Younis, Hoi Ting Poon, and Ali Miri
Department of Computer Science
Ryerson University
Toronto, Ontario, Canada
shadan.ghaffaripour, fadi.younis, hoiting.poon, ali.miri@ryerson.ca

***Abstract*—Compressed sensing (CS) schemes have been used in a wide number of applications in practice. Recently, they have been proposed for use in encryption algorithms because of their properties. In this paper, we present an empirical security analysis of compressed sensing-based encryption. Using a neural network model, we will show that the security of this type of encryption can be compromised. We consider at least three different scenarios in which an attack could occur causing partial information about the plaintext to be revealed without knowledge of the CS secret key.**

***Index Terms*—compressed sensing, signal processing, cryptanalysis, security, neural networks**

I. Introduction

Efficient transmission of multimedia signals over a network typically requires effective compression of these signals. In many instances, protecting the confidentiality of these signals is another system requirement. CS-based encryption schemes aim to address these requirements by combining compression and encryption. All CS-based schemes use the inherent redundancy in the structure of most signals to compress and recover them, using fewer random data samples compared to that required by the Shannon-Nyquist sampling theorem [1], [2]. The compressed-signal representation is obtained by multiplying the original signal by a *sensing matrix* to reduce its dimension. More specifically, compressed sensing recovers a signal $X \in \mathbb{R}_{n\times 1}$ from its linear measurements $Y = AX$, where $A \in \mathbb{R}_{m\times n}$ is a transform matrix with m being much smaller than n [3]. The more incoherent the measurement matrix is, within the domain in which the signal is sparse, the lower the value of m is allowed to be [4]. Most current proposed CS-based encryptions [5], [6], [7], [8], [9], [10], [11], [12], [13] rely on the fact that the sensing matrix must be known to reconstruct the compressed signal at the receiver, i.e. the sensing matrix A is used as the key. In this paper, we will show that it can be possible to obtain the original plaintext signal without knowledge of the key.

Section II presents our approach, the details of our experiments and analysis of our results. Conclusions can be found in Section III.

II. Main

Our proposed approach pertains to the generation of several artificial neural networks, to find the functional relationships capable of mapping ciphertexts to their corresponding plaintexts, without having any knowledge of keys used. We are particularly interested in investigating whether our proposed approach would provide a more effective method than brute-force, when using different keys matrices and/or plaintexts.

For each of the attacks described in subsections below, a dataset of 250,000 entries, containing plaintexts and their corresponding ciphertexts were generated. Regardless of the attack type, the plaintext(s), key matrix(ces) and ciphertext(s) will always have the following properties:

- Plaintext(s): Size = 1×20, Normally distributed, Density = 0.2 (equivalent to Sparsity = 0.8)
- Key Matrix(es): Size = 20×16, Normally distributed
- Ciphertext(s): Size = 1×16

Since CS-based encryption is commonly proposed for use in media applications, the plaintext data values are normalized to be between 0 and 256, to be representative of greyscale image pixel values. The principal goal is to minimize plaintext prediction error, such that the error distribution tends towards that of a zero-mean normal distribution, indicating that the attack has a high probability of success. It should also be mentioned that, where appropriate, the output from the neural network may be scaled, filtered using a threshold or both to ensure that it stays in range of possible plaintext values.

A. Variable Plaintext, Fixed Key

In this section, a known plaintext attack has been carried out on the compressed-sensing encryption scheme. This means that the key matrix remained unchanged throughout the process of data generation. It is a valid argument to make that the attackers can capture a large enough set of data and the corresponding encrypted version, to train an artificial neural network model to investigate the relationship between the two. It may be the case that the encryption key never gets updated in a system or at least, not very often. The attackers can, therefore, take advantage of this by launching an attack during the period that the key has remained unchanged. Figure 1 illustrates the error distribution, pertaining to one of the plaintext elements, resulting from such neural networks. To evaluate the effectiveness of this method, the output of the artificial neural network, i.e. the predicted plaintext, is compared against: (1) A simple random guessing approach, in which the attacker uses randomly generated numbers from the

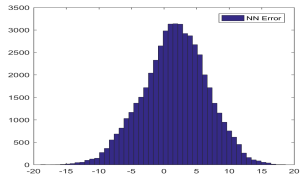
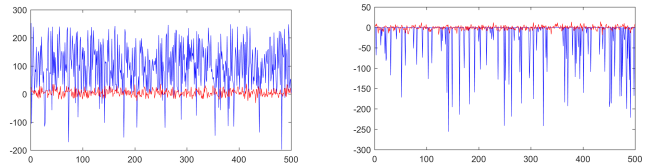


Fig. 1. Error distribution of neural network



(a) Errors related to non-zero elements (b) Errors related to zero elements

Fig. 2. Comparison of neural network error versus random guessing

same distribution and the same range as the plaintext values to guess the corresponding plaintext of an encrypted signal; (2) A more educated random guessing approach, where the attacker not only knows the plaintext distribution and range, but also knows the exact location of non-zero elements in each plaintext. Figure 2 and Figure 3 depict the error resulting from the neural network output versus error arising from the two random guessing approaches. We can see that, in both cases, the proposed approach predicts the plaintext with fewer errors. It is worth mentioning that considering the inherent sparsity of the plaintext, knowledge of the locations of non-zero elements is clearly substantial. Our results show that an attacker with knowledge of locations of non-zero elements is 56% more accurate in its prediction than an attacker without the knowledge.

To provide a better basis for comparison, Figure 4 depicts the three different error distributions overlaid in one graph.

In summary, we concluded the following:

- The neural network narrows down the search space by a factor of 8 when compared to the random guessing approach and a factor of 5 when knowledge of non-zero element locations are known, at the 90% confidence level. That is, an attacker needs to verify up to 8 times less elements than the random guessing approaches to learn the plaintext.
- On average, the neural network prediction is within 5 of the actual plaintext (3.36 for zero elements and 10.7 for non-zero elements). It is 7 times more accurate than the random guessing approaches for zero elements and 10 times more accurate for non-zero elements.

Given these points, the experiment demonstrated that, under the assumption of fixed key, the trained neural network could partially retrieve the original data without any knowledge of the encryption key.

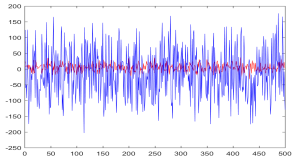


Fig. 3. Neural network prediction error versus random guessing with knowledge of non-zero locations

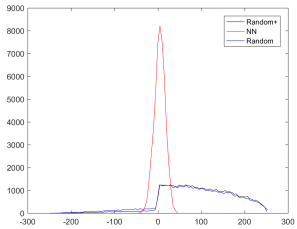


Fig. 4. Neural network's error distribution versus random guessing with knowledge of non-zero value locations

B. Fixed Plaintext, Variable Key

In this experiment, a CS-based encryption scheme is examined in the case where a single plaintext block X is encrypted by a variable set of key matrices. We calculated the dissimilarity or error between the predicted plaintext, $X_{\text{neural net}}$, from the neural network and the expected plaintext, X_{expected}. The result of the latter step is an error matrix denoted as $X_{\text{difference}}$. To properly visualize the performance, we generated a series of histograms (see Figure 5 as an example). Each histogram was created from one of the elements in the $X_{\text{difference}}$ vector. In a similar experiment, we generated the plaintext and the keys from a uniform distribution instead of the previously used normal distribution. As in the first experiment, the error

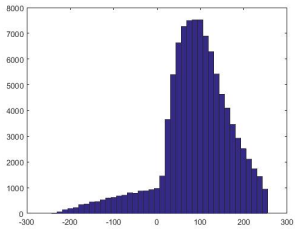
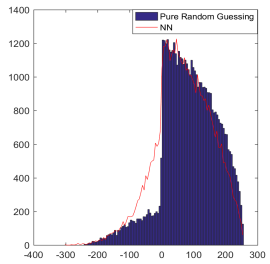
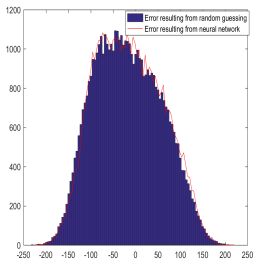


Fig. 5. Distribution of error between actual and predicted plaintext

between the plaintext predicted by the neural network and the expected plaintext were examined. The purpose of designing the above experiments was to analyze the effect that changing the distribution from normal to uniform would have on the neural network's prediction success. Ideally, it is expected that the neural network will give an accurate prediction for the location of the non-zero plaintext elements, or at least narrow down the range of possible values. A non-uniform prediction error distribution plot can be of great benefit to an adversary. A narrow normal curve would aid the attacker in eliminating

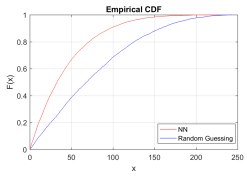


(a) Neural network's error distribution versus pure random guessing

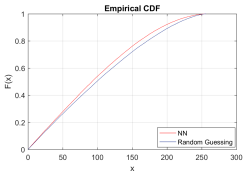


(b) Neural network's error distribution versus random guessing with knowledge of non-zero value locations

Fig. 6. Neural network's error distribution versus random guessing approaches



(a) Negative Errors



(b) Positive Errors

Fig. 7. Cumulative distribution of prediction errors of the neural network versus random guessing

certain candidate values based on their probability, and rather focus on the elements in the middle of the curve. Changing the key distribution and observing the effect on the correlation gives insight into the conditions that benefit or complicate the attacker's attempt at breaking the cryptographic system.

C. Variable Plaintext, Variable Key

Under the general scenario, the neural network performance was tested against a dataset consisting of different plaintexts, each of which was encrypted with a different key. The ciphertext generated from the dataset is then fed into the neural network model, the output of which is used for evaluation purposes.

Results from Figure 6a indicate that the neural network slightly outperforms random guessing. To further analyze the results, the cumulative distribution function was applied to the prediction errors, resulting from the neural network and random guessing approaches. Figure 7 shows these plots generated separately for positive and negative errors. Based on such comparisons, the superiority of the neural network stood out for negative errors. However, the difference between the two models was negligible for positive errors, with the neural network still performing slightly better. These observations could provide more evidence to confirm the hypothesis that the neural network reveals partial information about the plaintext in the general setting. Figure 6b depicts the error distribution of random guessing with knowledge of non-zero value locations versus the error distribution resulting from the neural network's output. The similarity of the two distribution errors might suggest that the neural network could infer the locations of the non-zero values in sparse plaintext.

III. Conclusions

In this paper, we've put the security of standard compressed-sensing cryptography under investigation. Specifically, we've shown that compressed sensing is not entirely secure under certain circumstances, and that it's possible to learn crucial information about the plaintext using the ciphertext, without the compressed CS key present in the process. We achieved the latter using an artificial neural network trained on blocks of plaintext and their corresponding ciphertext. The neural network could be used to derive a noticeable correlation between the plaintext and ciphertext through supervised learning. The basis of the security of the compressed-sensing model is that the ciphertext does not reveal or leak any relevant information about the key nor the plaintext. We conducted three different experiments and showed that using a neural network is better than random guessing. Our results therefore indicate that a neural network approach can aid an attacker to narrow down the search space and predict the plaintext elements.

References

[1] A. Orsdemir, H. O. Altun, G. Sharma, and M. F. Bocko, "On the security and robustness of encryption via compressed sensing," in *MILCOM 2008-2008 IEEE Military Communications Conference*. IEEE, 2008, pp. 1–7.

[2] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, March 2008.

[3] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[4] S. Qaisar, R. M. Bilal, W. Iqbal, M. Naureen, and S. Lee, "Compressive sensing: From theory to applications, a survey," *Journal of Communications and Networks*, vol. 15, no. 5, pp. 443–456, Oct 2013.

[5] Y. Rachlin and D. Baron, "The secrecy of compressed sensing measurements," in *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, Sept 2008, pp. 813–817.

[6] C. Wang, B. Zhang, K. Ren, and J. M. Roveda, "Privacy-assured outsourcing of image reconstruction service in cloud," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 1, pp. 166–177, June 2013.

[7] Q. Wang, W. Zeng, and J. Tian, "A compressive sensing based secure watermark detection and privacy preserving storage framework," *IEEE transactions on image processing*, vol. 23, no. 3, pp. 1317–1328, 2014.

[8] G. Hu, D. Xiao, T. Xiang, S. Bai, and Y. Zhang, "A compressive sensing based privacy preserving outsourcing of image storage and identity authentication service in cloud," *Information Sciences*, vol. 387, pp. 132–145, 2017.

[9] D. Zhang, X. Liao, B. Yang, and Y. Zhang, "A fast and efficient approach to color-image encryption based on compressive sensing and fractional fourier transform," *Multimedia Tools and Applications*, pp. 1–18, 2017.

[10] X. Chai, Z. Gan, Y. Chen, and Y. Zhang, "A visually secure image encryption scheme based on compressive sensing," *Signal Processing*, vol. 134, pp. 35–51, 2017.

[11] J. Dong, Z. Zhao, and H. Li, "An image encryption algorithm based on compressed sensing and dual-tree complex wavelet transform," in *Proceedings of the 5th International Conference on Bioinformatics and Computational Biology*. ACM, 2017, pp. 30–33.

[12] Y. Li, B. Song, R. Cao, Y. Zhang, and H. Qin, "Image encryption based on compressive sensing and scrambled index for secure multimedia transmission," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 12, no. 4s, p. 62, 2016.

[13] N. Zhou, S. Pan, S. Cheng, and Z. Zhou, "Image compression–encryption scheme based on hyper-chaotic system and 2d compressive sensing," *Optics & Laser Technology*, vol. 82, pp. 121–133, 2016.