A Review of Adversarial Neural Cryptography

Alisamar Husain
Dept. of Electrical Engineering
Jamia Millia Islamia

May 22, 2021

Abstract

Artificial neural networks are well known for their ability to selectively explore the solution space of a given problem. One of the recent applications of this feature is in the field of neural cryptography, which provides an opportunity to use ANNs to encrypt data such that it cannot be decrypted by an attacker.

In this paper we examine the efficacy, feasibility and general practicality of the use of *adversarial neural cryptography*, as coined by Abadi et al. in [1], and neural cryptography in general. We test systems recommended in the literature and examine their use in data transmission systems for the purpose of encrypting data from the perspective of securing a communication channel.

1 Introduction

The field of cryptography is broadly concerned with algorithms and protocols that ensure the secrecy and integrity of information. Cryptographic mechanisms are typically described as programs or Turing machines. By this definition, an appropriate neural network can possibly be considered a cryptographic function.

1.1 Terminology

Certain terms are frequently used while talking about cryptographic mechanisms and it is beneficial to have an understanding of what these refer to. Some of these will be used in this paper to commonly identify certain parts of the system and some are abbreviations made for convenience.

A **party** is a machine, or actor in general, which is using a communication channel to communicate with another machine. There are two major types of parties which we are concerned with, **participants and attackers.**

A **participant** is a party which actively takes part in the communication and sends messages on the channel. The goal of encryption is to ensure that the communication between any two parties can only be intercepted and understood by them.

An **attacker** is a party which attempts to intercept and understand the communication between two participants.

Attackers Attackers are also described in those terms, with bounds on their complexity (e.g., limited to polynomial time) and on their chances of success (e.g., limited to a negligible probability). A mechanism is deemed secure if it achieves its goal against all attackers. For instance, an encryption algorithm is said to be secure if no attacker can extract information about plaintexts from ciphertexts. Modern cryptography provides rigorous versions of such definitions, like those given by Goldwasser & Micali. [2]

1.2 Symmetric Encryption

2 Related Work

3 Methodology

We use a simple setup in order to build and test the networks. The models are first implemented using the Python programming language and after obtaining a suitably trained and validated model, we can move on to testing.

3.1 Tools Used

The models are implemented using PyTorch, a popular framework for building neural networks in Python. The models are built per the specification given in the literature and trained with the help of several common Python libraries. We use Tensorboard to monitor the training process and log the training metadata.

4 Results

4.1 ANC

4.2 Cryptonet

5 Conclusions

References

[1] M. Abadi and D. G. Andersen, "Learning to protect communications with adversarial neural cryptography," *arXiv preprint arXiv:1610.06918*, 2016.

[2] S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, vol. 28, no. 2, pp. 270–299, 1984. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0022000084900709