**Yishay Asher & Steve Gutfreund**

**Project Guide**

# Preparation

1. clone the project from <https://github.com/SimSteve/Project.git>
2. make sure you keep the hierarchy of the folders as is (otherwise the imports won’t work)
3. the working directory should be: C:\<PATH TO THE PROJECT>\Project\
4. python version 3.6
5. make sure to have the following packages:
   * tensorflow
   * numpy
   * matplotlib
   * pycrypto (OR pycryptodome for WINDOWS, but rename the crypto folder to Crypto)

# naming convention

All trained models are being saved according to the following convention:

DATASET\_ARCHITECTURE\_ENCRYPTION\_NORM\_PADSIZE

Example:

fashion\_modelB\_CTR\_0NORM\_0PADDED

Note: The NORM parameter excepts only two possible values; 0NORM and 0.5NORM. A model you wish to attack with the CW attack, should be trained with 0.5NORM.

(NORM does not affect the accuracy, it just makes a linear shift on all the pixels.)

# train a new model

python .\src\trainer.py

[-h] <-d dataset> <-m architecture> [-e encryption] [-p padsize] [-n normalization]

-h show this help text

-d specifying the dataset; mnist or fashion <must>

-m specifying the model architecture; modelA or modelB <must>

-e specifying the encryption method; UNENCRYPTED, PERMUTATED, ECB, CBC or CTR. default is UNENCRYPTED [optional]

-p specifying the number of rows to pad, default is 0 [optional]

-n specifying the normalization (img / 255.0 - n), default is 0 [optional]

Example:

python .\src\trainer.py –d fashion –m modelB –e CTR

# predicting an image

python .\src\predictor.py [-h] <-f filename> [-i index]

-h show this help text

-f specifying the filename of the model <must>

-i specifying the index, if non specified than randomly chosen [optional]

Example:

python .\src\predictor.py fashion\_modelB\_ECB\_0NORM\_PADDED –i 613

# evaluate model

python .\src\evaluation.py [-h] <-f filename> [-n amount]

-h show this help text

-f specifying the filename of the model <must>

-n specifying the amount of images, default is 10000 [optional]

Example:

python .\src\evaluation.py –f mnist\_modelA\_PERMUTATED\_0.5NORM\_32PADDED –n 1000

# plot a collage of encrypted images

python .\src\collage\_of\_encrypted\_images.py

[-h] <-d dataset> <-e encryption> [-p padsize] [-c classes] [-i images]

-h show this help text

-d specifying the dataset; mnist or fashion <must>

-e specifying the encryption method; PERMUTATED, ECB, CBC or CTR <must>

-p specifying the number of rows to pad, default is 0 [optional]

-c specifying the number of classes, default is 10 [optional]

-i specifying the number images for each class, default is 10 [optional]

Example:

python .\src\collage\_of\_encrypted\_images.py –d mnist –e PERMUTATED

# visualize an attack

python .\src\visualize\_attack.py [-h] <-f filename> [-i index] [-c CW\_mode]

-h show this help text

-f specifying the filename of the model <must>

-i specifying the index, if non specified than randomly chosen [optional]

-c specifying carlini mode; 2,0 or i. default is 2 [optional]

Example:

python .\src\visualize\_attack.py –f fashion\_modelA\_CTR\_0.5NORM\_0PADDED

# attacking a dataset

python .\src\dataset\_attack.py [-h] <-f filename> [-i amount] [-c CW\_mode]

-h show this help text

-f specifying the filename of the model <must>

-i specifying the amount, default is 1000 [optional]

-c specifying carlini mode; 2,0 or i. default is 2 [optional]

Example:

python .\src\dataset\_attack.py mnist\_modelB\_PERMUTATED\_0NORM\_0PADDED