Deep Learning
Term Project

(Enhancement of the reliability and security of the PUF authentication key in the wireless communication environment)

20191064

Jihoon LEE

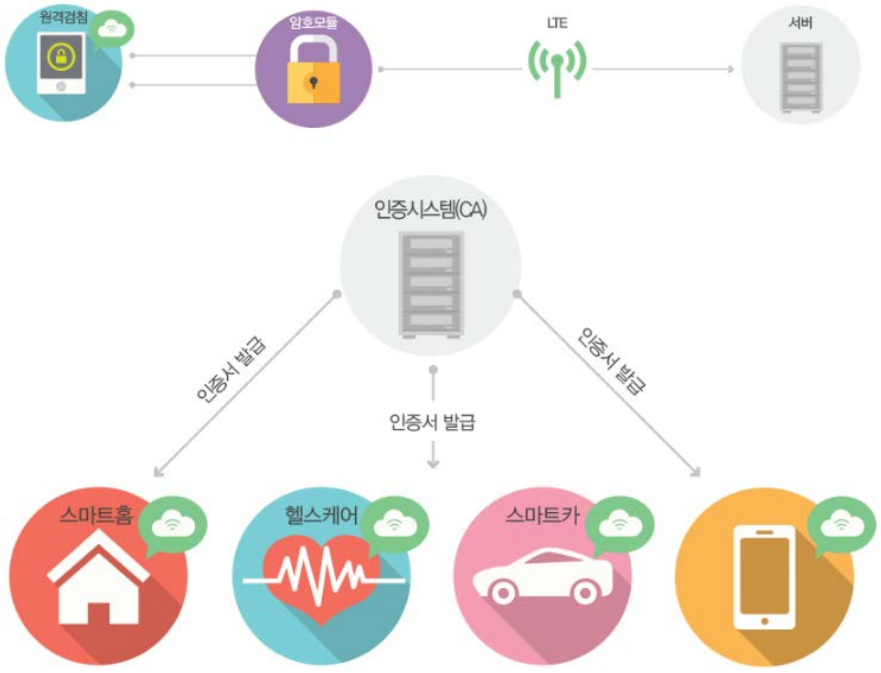
- INTRODUCTION

- **Authentication keys**

  - Roles

  : Required for encryption and decrypted cryptogram

  : data confidentiality, data integrity, authentication and non-repudiation features



https://www.dreamsecurity.com/solution/sol1_4_.php

- INTRODUCTION

- **Error Correction Code (ECC)**

  - Definition

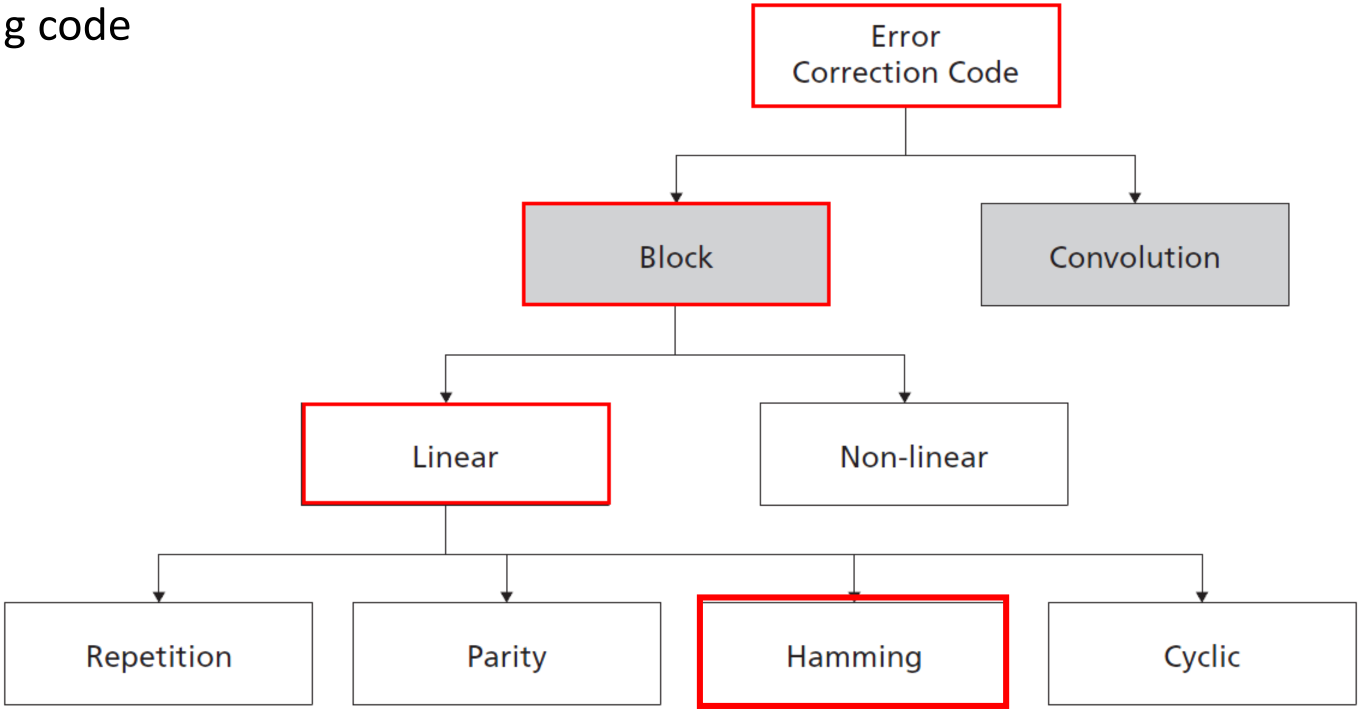
  : Code that detects and corrects data when problems occur

  - Objects

  : Improved reliability for data transmission in wireless communication

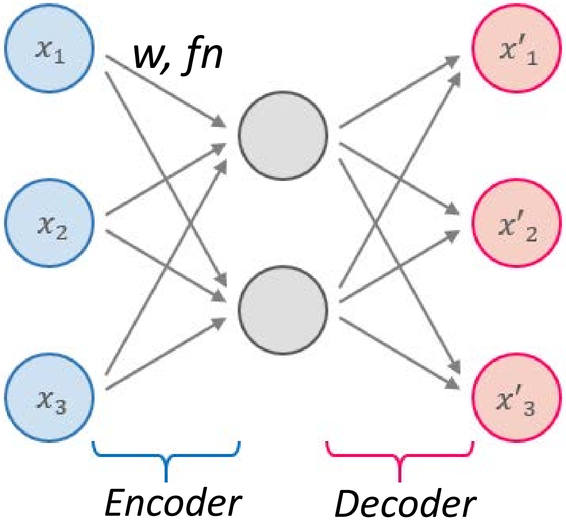
  - Examples

  : Binary Hamming code

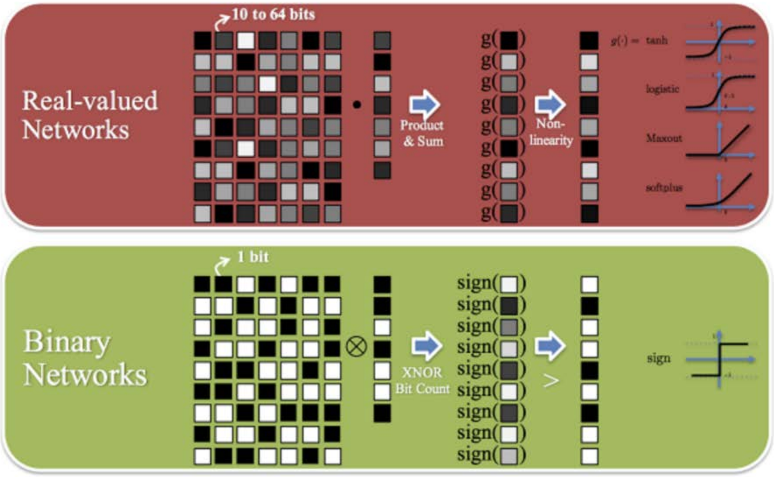


- INTRODUCTION

- **Autoencoder**
  - Unsupervised learning
  - Determine the parameter value so that the output comes close to the input
  - Extract meaningful features(ex. ReLU CNN)
- **BNN(Binarized Neural Network)**
  - Neural net with binary(1, -1) weights and activation function
  - Bit reduction to accelerate deep learning



- ✓ *w: Weights*
- ✓ *fn: Activation function*



https://github.com/jaygshah/Binary-Neural-Networks

- DATASETS

- **SRAM PUF(Physical Unclonable Function)**

  - Definition

  : Generate a security key using differences in the microstructure of semiconductors produced in the same manufacturing process

  - Roles

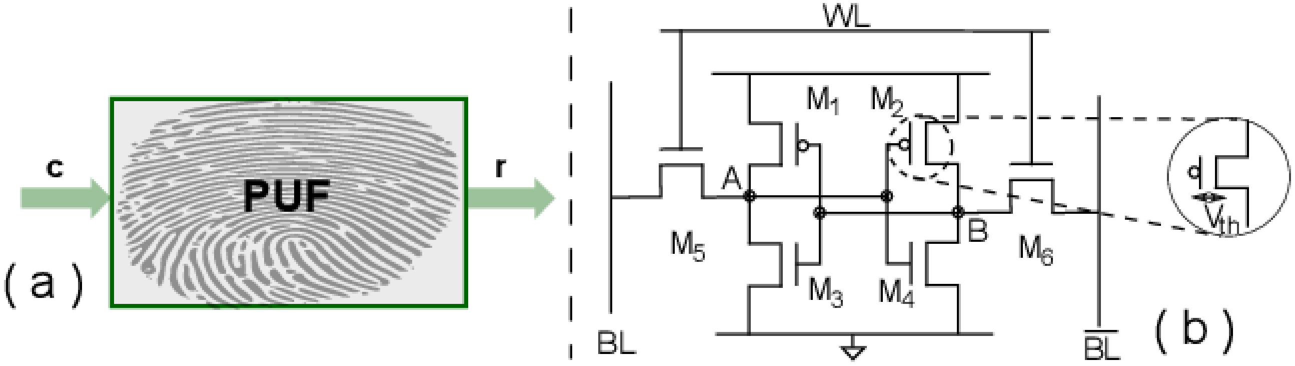
  : Optimized solution for IoT device security due to high security with a small chip

  - Properties

  ① Physical cloning is impossible by randomness

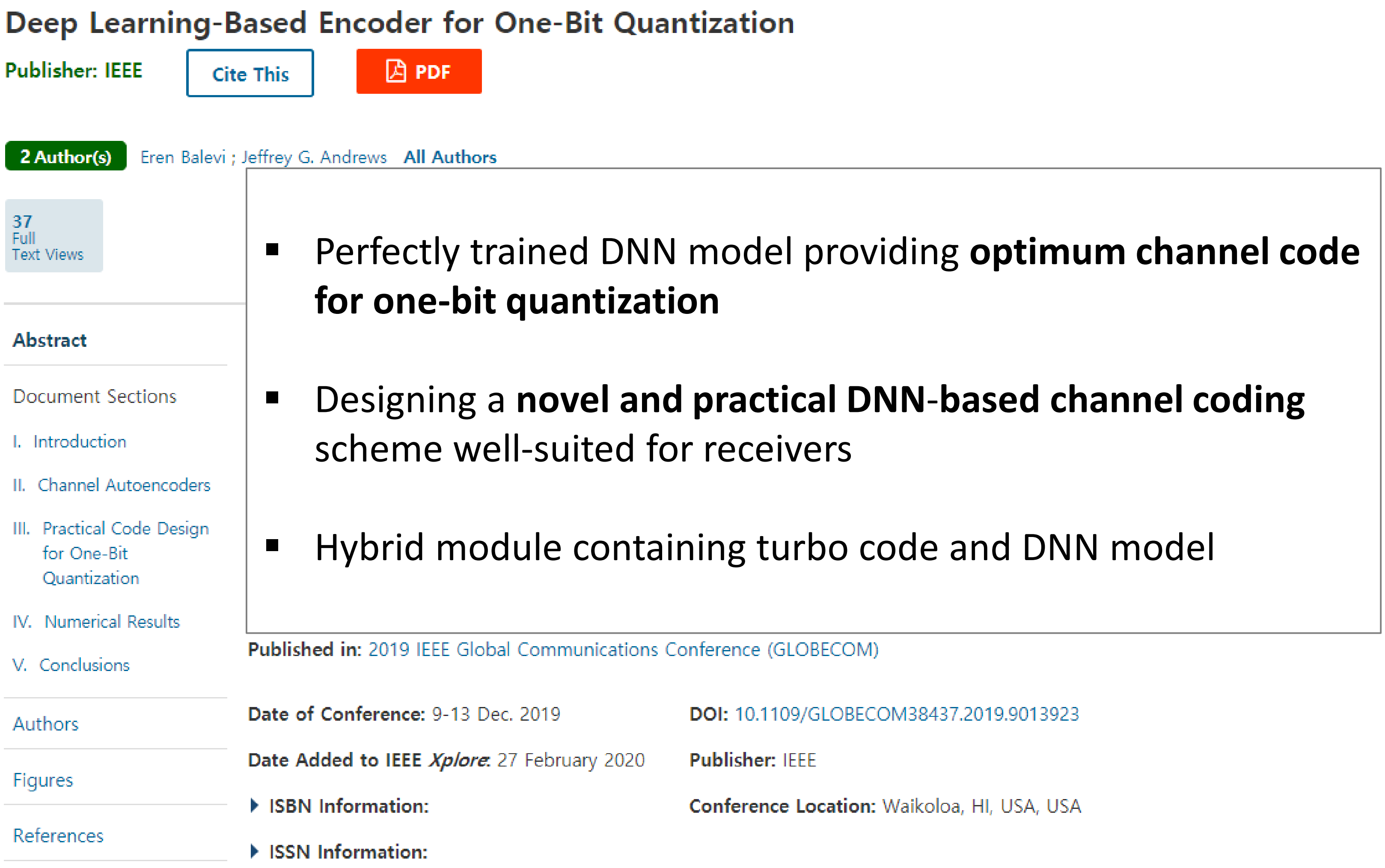
  ② Secure key management is possible

  ③ Data composed of small number of bits (Ex. 8bits)



Gao, Y., Su, Y., Yang, W., Chen, S., Nepal, S., & Ranasinghe, D.C. (2019). Building Secure SRAM PUF Key Generators on Resource Constrained Devices. 2019 IEEE ICPCCW, 912-917.

RELATED STUDIES



Deep Learning-Based Encoder for One-Bit Quantization

Publisher: IEEE

2 Author(s) Eren Balevi ; Jeffrey G. Andrews All Authors

37 Full Text Views

Abstract

Document Sections

I. Introduction

II. Channel Autoencoders

III. Practical Code Design for One-Bit Quantization

IV. Numerical Results

V. Conclusions

Authors

Figures

References

- Perfectly trained DNN model providing **optimum channel code for one-bit quantization**
- Designing a **novel and practical DNN-based channel coding** scheme well-suited for receivers
- Hybrid module containing turbo code and DNN model

Published in: 2019 IEEE Global Communications Conference (GLOBECOM)

Date of Conference: 9-13 Dec. 2019

Date Added to IEEE *Xplore*: 27 February 2020

▸ **ISBN Information:**

▸ **ISSN Information:**

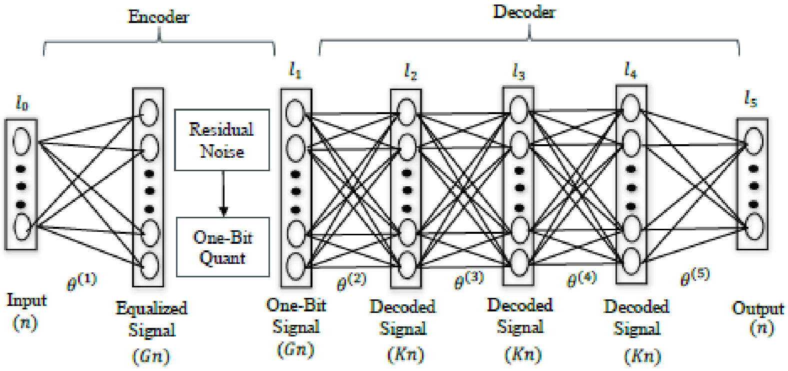
DOI: 10.1109/GLOBECOM38437.2019.9013923

Publisher: IEEE

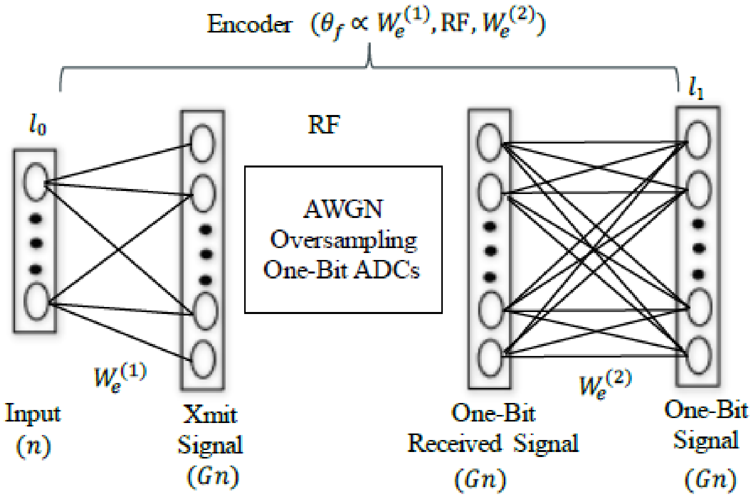
Conference Location: Waikoloa, HI, USA, USA

RELATED STUDIES

- **Schema**



(a)



- PROBLEM

- **SRAM PUF(Physical Unclonable Function)**

  - authenticated even with a small number of bits (n bits)

  → However, this also implies the risk that an attacker

  → randomly authenticate and pass through an attempt to hack

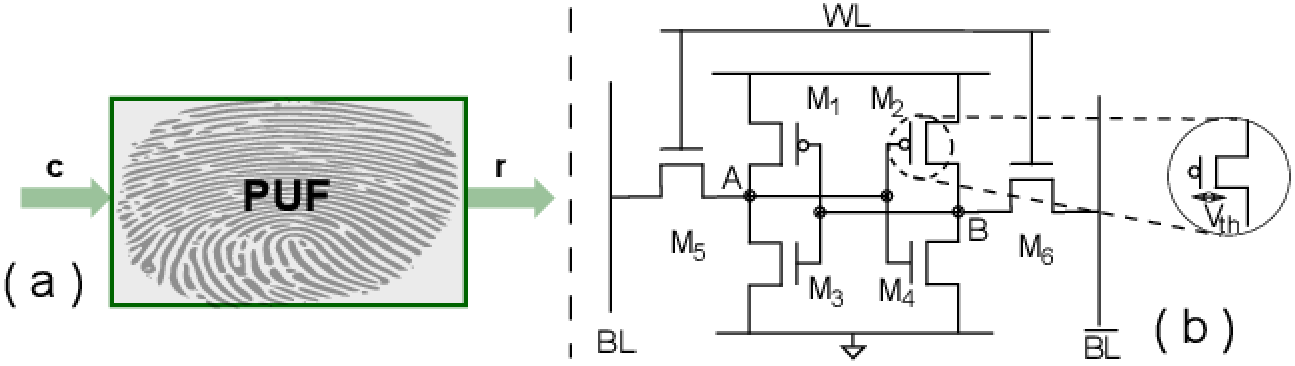
- **Application to Model**

  - Performed communication by increasing the size (k, k>n) of the hidden layer

  - Reduced risk of hacking trial

  → 16-bits or 32-bits per neuron rather than an integer

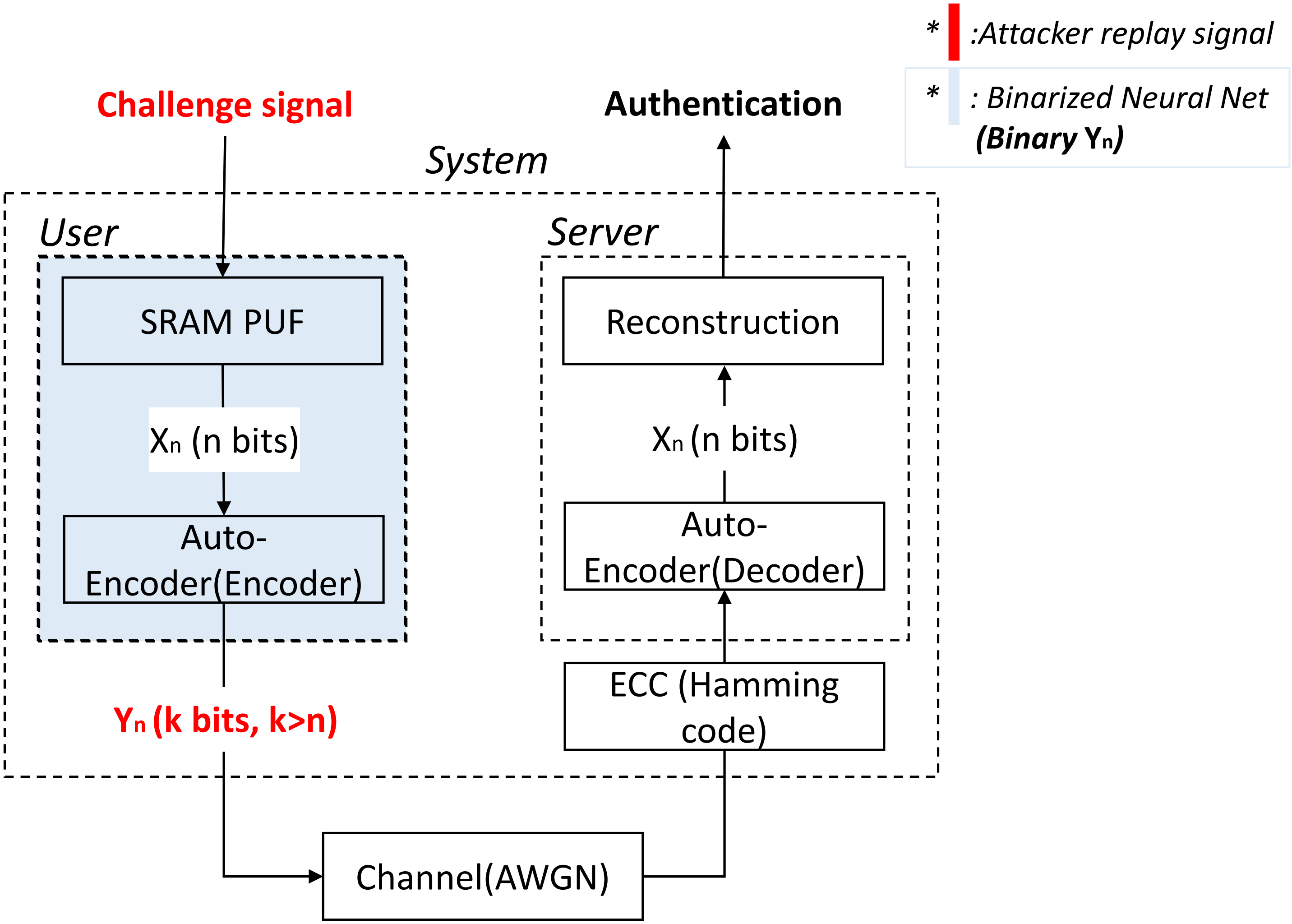
  → Disadvantage in terms of size



Gao, Y., Su, Y., Yang, W., Chen, S., Nepal, S., & Ranasinghe, D.C. (2019). Building Secure SRAM PUF Key Generators on Resource Constrained Devices. 2019 IEEE ICPCCW, 912-917.

PROPOSED METHOD

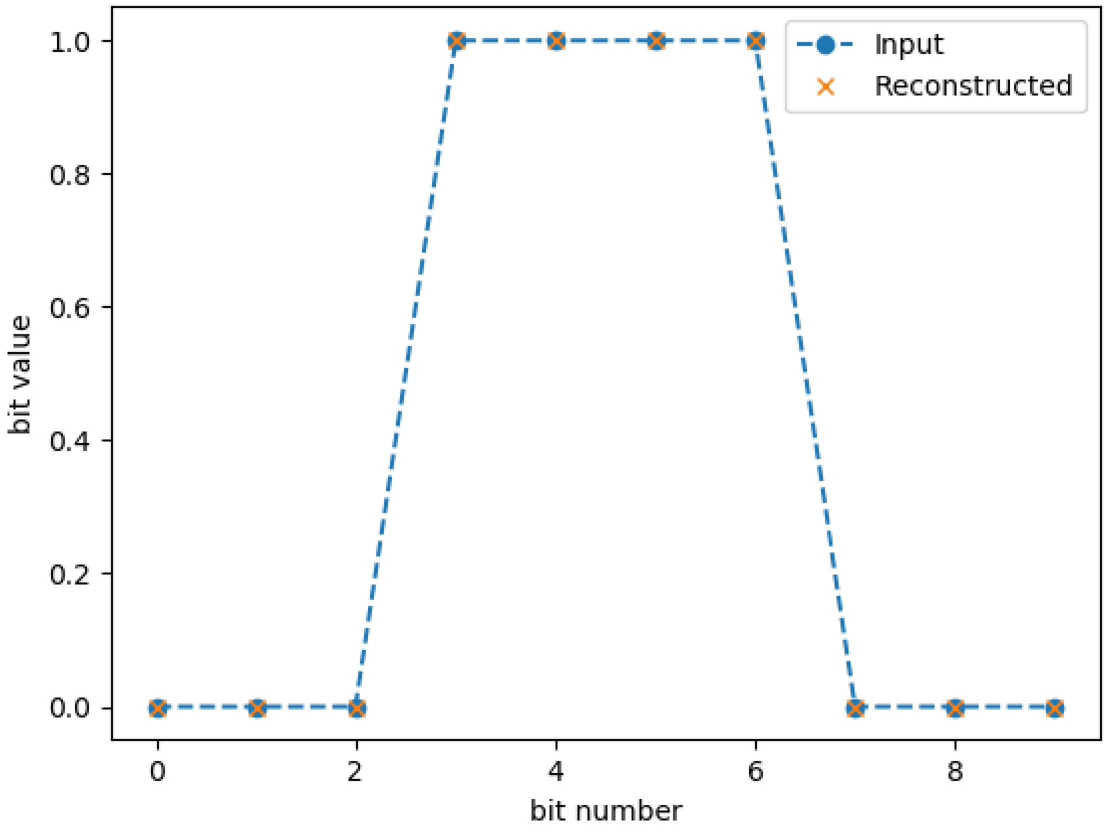
- **Schema**



Reliability, Security ↑ for Keys

- RESULTS

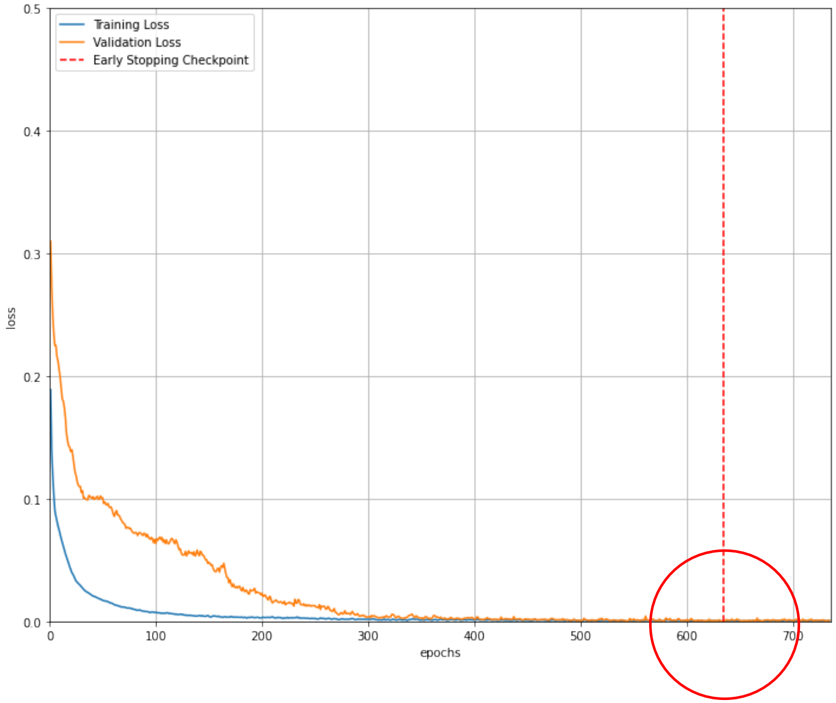
- **Paper Follow up**
  - Check about perfectly reconstructed after decoding



RESULTS

- **Proposed Method**

  - Reconstruction w. 0.02% error after decoding
  - Input: Size 10, Bit string of 0~1023(2^10) number
  - Latent vector: Size 20
  - Adam optimizer, MSE loss, Learning rate: 0.001



Loss = 100 * (# of False bits) / (# of total bits)

= 100 * 2 / (1024*10)

= 0.02[%]

- CONCLUSION

- **Failure of application to various Autoencoder model**
  - ex. Denoise-Autoencoder, adjustment # of hidden layers, Parameter tuning
- **Information loss**
  - when converted float data to binary data in a hidden layer
- **Autoencoder with different model structure**
  - increased size of hidden layer
- **Computational efficiency on BNN**

Thank you