UNIVERSITY OF CALIFORNIA
RIVERSIDE

Towards Security at the Internet Edge: From Communication to Classification

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Fangfang Yang

December 2022

Dissertation Committee:

Dr. Shaolei Ren, Chairperson
Dr. Daniel Wong
Dr. Wei Ren

The Dissertation of Fangfang Yang is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I would like to take this opportunity to express my deepest appreciate to my advisor, committee, lab mates, friends and family. First and foremost, I am extremely grateful to my advisor, Prof. Shaolei Ren, who generously provides expertise, guidance and support during my Ph.D. study. His immense knowledge and plentiful experience have encouraged me in both my academic research and daily life. I could not have undertaken this journey without him. I would also like to thank my dissertation committee, Prof. Daniel Wong and Prof. Wei Ren, for their tremendous support and valuable suggestions in my final defense exam. In addition, many thanks to my lab mates. Thanks for your insights and help to my research work. I had the pleasure of working and collaborating with you. I also want to thank my friends, who are always there supporting and trusting me. Last but not least, I want to express my gratitude to my beloved family. Thank you for all your support, encouragement and love. My PhD life becomes easier and happier because of you.

Funding Acknowledgement. I appreciate the funding support from National Science Foundation under grant number ECCS-1610471 and CNS-1551661.

Publication Acknowledgement. The text of this dissertation, in part or in full, is a reprint of the material as it appears in list of publications below. The co-author Dr. Shaolei Ren listed in that publication directed and supervised the research which forms the basis for this dissertation.

- Fangfang Yang, Mohammad Atiqul Islam, and Shaolei Ren. ”PowerKey: Generating Secret Keys from Power Line Electromagnetic Interferences.” In International Conference on Network and System Security, pp. 354-370. Springer, Cham, 2020.

Added to the dissertation as Chapter 2. Fangfang Yang is the major contributor of the project. Mohammad Atiqul Islam assisted in evaluation and experiments.

- Fangfang Yang, Mohammad Atiqul Islam, Fan Wu, and Shaolei Ren. "CompKey: Exploiting Computer's Electromagnetic Radiation for Secret Key Generation." In 2021 IEEE Conference on Communications and Network Security (CNS), pp. 281-289. IEEE, 2021.

  Added to the dissertation as Chapter 3. Fangfang Yang is the major contributor of the project. Mohammad Atiqul Islam and Fan Wu assisted in evaluation and experiments.

- Fangfang Yang and Shaolei Ren. "On the vulnerability of hyperdimensional computing-based classifiers to adversarial attacks. "In International Conference on Network and System Security, pp. 371-387. Springer, Cham, 2020.

  Added to the dissertation as Chapter 4. Fangfang Yang is the major contributor of the project.

To my husband.

ABSTRACT OF THE DISSERTATION

Towards Security at the Internet Edge: From Communication to Classification

by

Fangfang Yang

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, December 2022
Dr. Shaolei Ren, Chairperson

The increasing adoption of Internet-of-Things (IoT) devices and an explosion in sensor data are fueling frequent data communication between edge devices and intelligence moving towards the Internet edge. As a side effect, the number of potential threats and possible attacks against security and privacy among edge devices has grown drastically. In this dissertation, we focus on strengthening security at the Internet edge, from securing communication between edge devices to achieving trustworthiness of on-device classification. First, we propose PowerKey to secure communication between multiple plugged edge devices in an electrical domain. Concretely, PowerKey generates secret communicating keys for communications between devices plugged into nearby power outlets by exploiting electromagnetic interferences (EMI) spikes with randomly varying but consistent frequencies. Second, to achieve secure communications for unplugged edge devices, we propose another secret key generation method, called CompKey, which allows wireless edge devices in the proximity of a third-party computer to securely associate with each other by exploiting electromagnetic radiation (EMR) emitted from the computer. Next, for trustworthy on-device classification, we study the adversarial attacks on brain-inspired hyperdimensional computing (HDC) classifiers. Finally, we consider an ultra-efficient version of HDC classifiers — low-dimensional computing (LDC) classifiers — and propose an interval bound propagation (IBP) technique to achieve certified robustness against adversarial attacks subject to L_∞ norm-bounded perturbation.

Contents

List of Figures **xii**

List of Tables **xv**

1 Introduction **1**

1.1 Internet Edge 2

1.2 Securing Communication Between Edge Devices 3

1.2.1 Potential Attacks 3

1.2.2 Proximity-Based Authentication 4

1.3 Reliability of On-Device Classification 7

1.3.1 On-Device Classification 7

1.3.2 Robustness Against Adversarial Attacks 9

1.4 Thesis Contributions 10

2 PowerKey: Generating Secret Keys From Power Line Electromagnetic Interferences **13**

2.1 Introduction 13

2.2 Preliminaries on Power Line EMI 16

2.2.1 Overview of EMR/EMI. 16

2.2.2 Existing Research on Exploiting EMR/EMI. 17

2.3 Problem and Threat Model 17

2.3.1 Problem Statement 17

2.3.2 Threat Model 19

2.4 An In-Depth Look at High-Frequency EMI Spikes 19

2.4.1 Sources for High-Frequency EMI Spikes. 19

2.4.2 Characteristics of EMI Spikes. 20

2.5 The Design of PowerKey 22

2.5.1 Offline Pre-Processing 23

2.5.2 Quantize Frequencies of EMI Spikes 24

2.5.3 Extract Secret Keys 25

2.6 Evaluation Methodology 27

2.6.1 Experiment Setup. 27

2.6.2 Evaluation Metrics. 28
2.7 Evaluation Results 29
2.7.1 Analysis of EMI Spike Frequencies. 29
2.7.2 Performance of PowerKey. 30
2.7.3 Security Analysis of PowerKey. 31
2.8 Related Works 32
2.9 Conclusion 34
2.10 Appendix 34
2.10.1 Experiment Setup. 34
2.10.2 Analysis of EMI Spike Frequencies in the Office Suite. 35
2.10.3 Results for Key Generation in the Lab 35

3 CompKey: Exploiting Computer's Electromagnetic Radiation for Secret Key Generation 37
3.1 Introduction 37
3.2 Problem and Threat Model 40
3.2.1 Problem Definition 40
3.2.2 Threat Model 41
3.2.3 Remarks 42
3.3 Characteristics of Computer EMR 44
3.3.1 Memory Bus EMR 44
3.3.2 Memory Bus EMR as a Secret Source 46
3.4 The Design of CompKey 50
3.4.1 Extraction of EMR Frequency Signal 50
3.4.2 Difference-Based Encoding Method 51
3.4.3 Reconciliation 53
3.4.4 Privacy Amplification 54
3.5 Experimental Methodology 55
3.5.1 Experiment Setup 55
3.5.2 Performance Metrics 56
3.6 Evaluation Results 58
3.6.1 Performance of CompKey 58
3.6.2 Randomness of Secret Key 59
3.6.3 Sensitivity of CompKey 60
3.6.4 Security Analysis 63
3.7 Related Works 64
3.8 Conclusion 65

4 On the Vulnerability of Hyperdimensional Computing-Based Classifiers to Adversarial Attacks 66
4.1 Introduction 66
4.2 Preliminaries on HDC Classifiers 68
4.2.1 Random Indexing 69
4.2.2 Multiply-Add-Permute Operation 69
4.3 A HDC Classifier on MNIST Dataset 70

4.3.1 Mapping . . . 70
4.3.2 HDC Classifier . . . 72
4.4 Threat Model . . . 73
4.5 A Modified Genetic Algorithm . . . 74
4.5.1 Genetic Algorithm . . . 74
4.5.2 Modification for Perturbation Reduction . . . 77
4.5.3 Effect of Perturbation Reduction . . . 78
4.6 Evaluation Results . . . 79
4.6.1 HDC Classifier Training . . . 79
4.6.2 Attack on HDC Classifier With RMR . . . 81
4.6.3 Attack on HDC Classifier With FMR . . . 83
4.7 Related Works . . . 85
4.8 Conclusion . . . 86
4.9 Appendix: Additional Results . . . 86

5 Achieving Certified Robustness for Brain-Inspired Low-Dimensional Computing Classifiers 92
5.1 Introduction . . . 92
5.2 Preliminaries . . . 94
5.2.1 LDC Classifier . . . 94
5.2.2 Adversarial Attacks . . . 96
5.2.3 Robustness Verification . . . 97
5.3 Interval Bound Propagation . . . 98
5.3.1 Interval Bound Propagation . . . 98
5.3.2 Elision of Last Layer . . . 101
5.4 Results . . . 101
5.4.1 Experiment Setup . . . 102
5.4.2 Nominal Accuracy and Verified Accuracy . . . 104
5.4.3 Robustness Against PGD . . . 106
5.4.4 Robustness Against Memory Errors . . . 107
5.5 Conclusion . . . 108

6 Conclusions 109

Bibliography 112

List of Figures

1.1 Configuration of Internet edge. 2
1.2 Proximity-based secret key generation. 5
1.3 HDC classification. 8

2.1 Overview of a trust domain (i.e., authenticated electrical domain in [64]). . 18
2.2 Frequency analysis of voltage signal. (a) Without the additional computer; (b) With the additional computer. 20
2.3 PSD of voltage signals. (a) Outlet 1 in the lab. (b) Outlet 2 in the lab. (c) Outside the lab (i.e., outside trusted rooms). 21
2.4 The design overview of PowerKey. 22
2.5 (a) Correlation coefficients of EMI spike frequencies in the office. (b) Entropy with different quantizations. 30
2.6 Performance of PowerKey in the office suite. 30
2.7 (a) Mutual information: “AB” (Alice-Bob), “A-Att” (Alice-Attacker), and “B-Att” (Bob-Attacker). (b) Bit error rate. “AB” means Alice/Bob; “Volt” means the attacker uses the highest EMI spike for each window from its collected signals; “Stat” means estimating the EMI spike frequencies based on their probability distributions. 33
2.8 (a) Layout of the office. (b) Layout of the lab. 34
2.9 Histogram of 8 different EMI spike frequencies in the office suite. 35
2.10 Distribution of 8 different EMI spike frequencies in the office suite. “S-n” means the n-the EMI spike. $\sigma = 1, 1, 1, 1, 1, 4, 1, 1$Hz for the 8 EMI spike windows, respectively. 35
2.11 (a) Correlation coefficients of EMI spike frequencies in the lab. (b) Key matching rate for four devices in the lab. 36
2.12 Performance of PowerKey in the lab. 36

3.1 Two parties (Alice and Bob), who are located in the vicinity of a computer (within r_1 from the computer), can perceive the same radiation emitted from the computer. An attacker can only reside beyond some distance, r_2, from the source computer. 40

3.2 Frequency spectrum of memory bus EMR for four different kinds of computers. The blue line represents the EMR that is obtained when the computer is off. And the red line is the captured signal when the source computer is turned on. 44
3.3 (a) Histogram of EMR frequency of Acer Aspire V3. (b) Histogram of frequency difference between two devices collecting EMR signals 0.5m away from Acer Aspire V3. 47
3.4 Acer Aspire V3-372T: SNR vs. distance under different RAM access frequencies. 48
3.5 (a) The normalized EMR frequency variation pattern during 100s when Alice and Bob are placed close to a source computer and synchronously collect the radiation data. (b) The normalized EMR frequency variation pattern of Alice and an attacker. The attacker launches a copy attack and EMR using its receiving device. 48
3.6 Experiment setup. We take two different kinds of computers — Dell XPS and Dell OptiPlex — as source computers labeled as A and B, respectively. The leftmost computer is another Dell XPS with the same component and configuration as source computer A and acts as an interfering computer. Similarly, the upper right one is the same as source computer B and acts as an interfering computer. 55
3.7 (a) Spectrogram of Alice's and Bob's EMR spike in frequency window $[f_1, f_2]$ over 10 seconds. f_2-f_1 is 100Hz. (b) Comparison of frequency variation pattern between Alice and Bob. 58
3.8 BER and KMR of two experiments with Dell XPS and Dell OptiPlex as source computers, respectively. 59
3.9 Different encoding thresholds result in different entropies and BERs. 60
3.10 KMR with different distances when RAM access frequency of the source computer varies from 100% to 20%. 62
3.11 Performance of attackers. 63

4.1 (a) The encoder in our HDC classifier encodes a digital image (called sample image) to a sample hypervector. (b) The overview of the HDC classifier. An associative memory storing class hypervectors is generated using the training dataset. Then, a test sample can be classified based on its similarity to class hypervectors. 71
4.2 Comparison of different adversarial attacks that mislead the HDC classifier to classify "6" as "2". (a) Original benign image. (b) Adversarial image by basic genetic algorithm (GA). (c) Adversarial image by genetic algorithm with critical gene crossover (GA-CGC). (d) Adversarial image by our proposed genetic algorithm with critical gene crossover and perturbation adjustment (GA-CGC-PA). 79
4.3 (a) Per-image accuracies of benign/adversarial images shown in Fig. 4.7. (b) Query counts needed to generate adversarial images shown in Fig. 4.7. . . . 81
4.4 ASR of digits 0-9 for HDC classifier with FMR. 84

4.5 Box plot of perturbation noise added by GA-CGC-PA for the HDC classifier with FMR. Each box plot shows the values for the maximum/minimum/median/75th percentile/25th percentile, excluding outliers. 84
4.6 Box plot of query count needed by GA-CGC-PA for the HDC classifier with FMR. Each box plot shows the values for the maximum/minimum/median/75th percentile/25th percentile, excluding outliers. 86
4.7 Attacks on the HDC classifier with RMR. The first row shows the original images. The second row shows the perturbation noise added by the attacker. The third row shows the adversarial images, and the corresponding misclassified labels are given at the top of each image. 91
4.8 Attacks on the HDC classifier with FMR (hard). The first row shows the original images. The second row shows the perturbation noise added by the attacker. The third row shows the adversarial images, and the corresponding misclassified labels are given at the top of each image. 91
4.9 Attacks on the HDC classifier with FMR (vulnerable). The first row shows the original images. The second row shows the perturbation noise added by the attacker. The third row shows the adversarial images, and the corresponding misclassified labels are given at the top of each image. 91

5.1 Illustration of LDC model with IBP method. The L_∞ norm-bounded perturbation with radius ϵ (in blue) is propagated through layers of LDC model. The interval bound (in gray), represented as $|\bar{z}_k, \underline{z}_k|$, is propagated simultaneously through layers, which always encompasses the blue region. 96
5.2 Nominal accuracy of LDC models with different training perturbation radii, ϵ_1, ϵ_2, ϵ_3, ϵ_4, and ϵ_5, representing perturbation radius of 0, 0.02, 0.05, 0.08, and 0.1 respectively. The blue line are the results without elision technique and the red line represents the one with eliding the last layer. (a) MNIST dataset. (b) Fashion MNIST dataset. 103
5.3 Verified accuracy against different test perturbation radii from 0 to 0.12. Five different models, ϵ_1, ϵ_2, ϵ_3, ϵ_4, and ϵ_5, associate with training perturbation radius of 0, 0.02, 0.05, 0.08, and 0.1 respectively. (a) MNIST dataset. (b) Fashion MNIST dataset. 104
5.4 Attack success rate (ASR) of PGD attacking method to five LDC models, ϵ_1, ϵ_2, ϵ_3, ϵ_4, and ϵ_5, trained with IBP with training perturbation radius of 0, 0.02, 0.05, 0.08, and 0.1 respectively. (a) MNIST dataset. (b) Fashion MNIST dataset. 106
5.5 Classification accuracy of five trained models, ϵ_1, ϵ_2, ϵ_3, ϵ_4, and ϵ_5, with faulty memory cells, when the probability of failure for each memory cell is 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2} and 10^{-1}. ϵ_1, ϵ_2, ϵ_3, ϵ_4, and ϵ_5 correspond to training perturbation radius of 0, 0.02, 0.05, 0.08, and 0.1 respectively. (a) MNIST dataset. (b) Fashion MNIST dataset. 107

List of Tables

2.1 Frequency Quantization Schemes. 28

3.1 Memory bus clock frequency information of different computers. 45
3.2 Randomness test 60

4.1 Perturbation for Images Shown in Fig. 4.7 82
4.2 Perturbation for Images Shown in Fig. 4.8 and Fig. 4.9. The values for Fig. 4.9 are shown in parentheses. 87

5.1 Configuration of training procedure for each dataset with different training perturbation radii. 102

Chapter 1

Introduction

The emergence of Internet of Things (IoT) has provided the opportunity to connect different isolated devices into a communicating thing, which enables plenty of smart services ranging from building automation to health monitoring. The technological advances in electronics, computer science and networks have led to an exponential increase in the number of Internet-connected sensing and computing edge devices. Meanwhile, the data volume collected and communicated by edge devices from their environment is climbing, which is projected to reach a massive total of 79.4 zettabytes by 2025 [71, 92, 121]. Given the huge amount of data flow among edge devices, data privacy and security are becoming one of the major concerns with regard to IoT adoption. In this thesis, we focus on the security at the Internet edge, the region which is closest to the source of the data.

In this chapter, we first introduce Internet edge and security concerns from communication to classification. In what follows, we depict securing communication between edge devices and reliability of on-device classification. Last, we present the thesis's contributions.

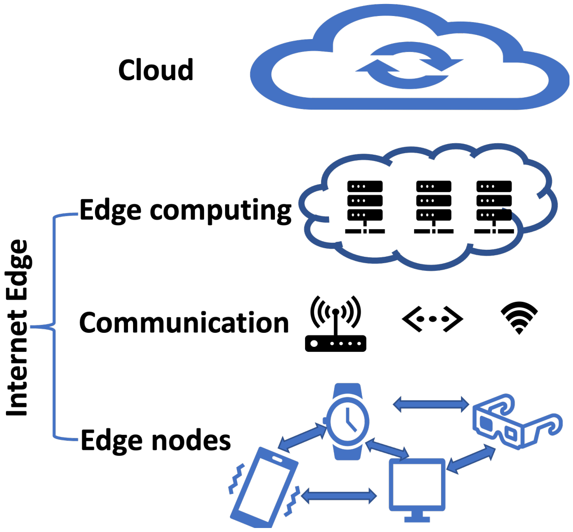


Figure 1.1: Configuration of Internet edge.

1.1 Internet Edge

With IoT evolving rapidly, billions or even trillions of devices will be connected to Internet, like Apple watches, Oculus Rift helmets, Google Nest and Fitbit sports trackers, which are located at the edge of the Internet. According to the reference model of IoT, Internet edge is the lowest level in IoT system, where data are collected and generated and the most essential and limited tasks are carried out. As shown in Fig. 1.1, there are three levels at the edge of the Internet, edge nodes level, communication level and edge computing level. Edge nodes level consists of heterogeneous and powerful devices. Via cable or wireless media like WiFi or bluetooth, communication level enables transmission of data, information or commands between edge devices [79, 92, 103]. Due to latency, bandwidth decrease and privacy concern in cloud computing, most computing, services and intelligence are brought as locally as possible and incorporated on end devices to ease cloud traffic. If the loads are beyond the capacity of edge devices, they can offload the data and tasks to the edge servers for processing, which is called edge computing [7,35,86,97].

Since a plethora of data is generated, communicated and processed at the edge side of Internet, it is extremely important to study the security and immunity of Internet edge. In the following sections, we will present two security aspects at the edge side of Internet, securing communication between edge devices and the reliability of on-device classification.

1.2 Securing Communication Between Edge Devices

In this section, the security of the communication level in Internet edge is presented. We first talk about the potential attacks against device-to-device communication. We then introduce the countermeasures, proximity-based authentication.

1.2.1 Potential Attacks

One of the most common threats to communication among edge devices is unauthorized conversation. In a smart community, each edge node is supposed to transmit data with authorized nodes. Without authentication, an attacker could easily hack the whole system. For example, in a smart home scenario, an attacker could control the heating system by sending tailored data to the thermostat which needs the smoke detector's data in order to shut down the heating system in an emergency situation [85,92].

Another most popular attack happened at the communication level is eavesdropping attack, also called sniffing attack, where attacker intentionally listening to radio channel between two authenticating parties. Valuable and sensitive data, like usernames and passwords, can be easily extracted when the communication is unencrypted. Besides, the attacker can utilize this captured information to design other tailored attacks [92, 94, 101].

Encryption can be used to defeat eavesdropping threat. Traditionally, to guarantee security and secrecy, the transmitted data is encrypted through cryptographic techniques, such as the classic public key encryption Diffie-Hellman (DH). DH is a public-key cryptography which employs a key exchange protocol and enables two parties to obtain shared secret keys to encrypt and decrypt their conversation [2]. However, since DH does not verify the identity of participating devices, an attacker could easily impersonate a legitimate device and establish a shared key with one or both of the valid devices [25, 119, 140, 151]. As a consequence, this makes DH protocol vulnerable to spoofing and man-in-the-middle attacks, where an attacker can easily establish a shared key with one of or both of parties.

1.2.2 Proximity-Based Authentication

To overcome the drawbacks of traditional Diffie-Hellman encryption method, researchers propose proximity-based authentication to secure communications between edge devices. Based on the fact that an attacker is usually located in a farther distance than that between two communicating parties and that devices coming into close vicinity can sense similar ambient environment, the basic idea of proximity-based authentication is to use ambient physical signal to prove proximity or generate secret keys for two legitimate devices. As shown in Fig. 1.2, two legitimate devices, Alice and Bob, could adopt the dynamic characteristics of natural ambient signals for establishing a secure and authentic communication channel between them. Concretely, Alice and Bob can extract a shared secret key from their common randomly-varying ambient environment and use the shared

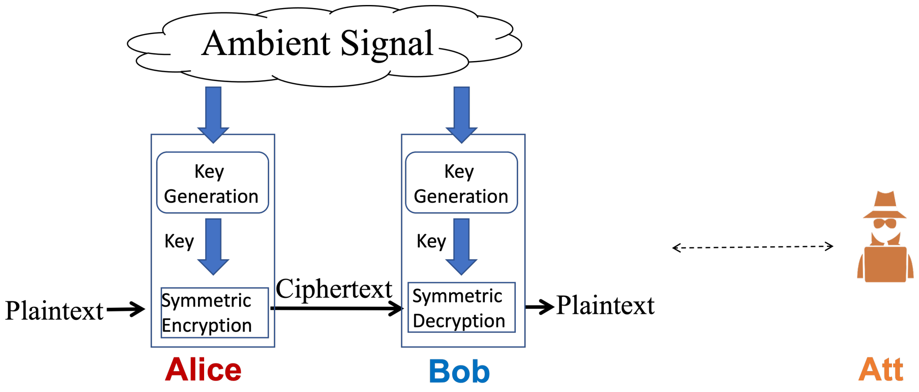


Figure 1.2: Proximity-based secret key generation.

key to encrypt transmission of message. Thus, the physical proximity can serve as a proof of device authentication, because adversaries are not allowed to approach the vicinity.

Such ambient sources studied in literature include radio-related signals, WiFi signal, ambient audio signal, ambient luminosity and biometrics. Radio is mostly used for this purpose. Amigo scheme is the first one to explore radio signal to authenticate. In Amigo, the author utilizes a binary classifier, trained a priori, to make the decision as to whether two communication parties are colocated by comparing the received signal strength (RSS) of radio packets overheard by two parties. Ensemble is another protocol taking advantage of wireless radio for authentication. In Ensemble paper, the author makes use of an ensemble of trusted devices as witness and record the radio packets transmitted between two pairing devices. And instead of relying on pairing devices to decide, it depends on all the witness and decide authentication according to the votes by all witness devices. ProxiMate paper proposed to extract a shared secret key bits from common radio environment, which is directly used as encryption key [54, 82, 127]. According to the rule that devices reside within half of a wavelength of radio signal will perceive similar signal fluctuations, radio-based authentication is limited to the applications with small authentication distance. Especially for ProxiMate, which intends to extract an accurate key bit, the distance between two devices should be as close as possible, which hurts the practical authentication implementation. Also, the nature of radio signal that it can go through the wall makes it insecure in authentication process. So an attacker who are in different rooms separated by a wall but are still near the paring device can be easily classified as legitimate party.

A lot of other papers leverage acoustic signal to realize authentication. Similar to radio-based authentication, Kim's paper is based on the observation that devices within proximity can perceive similar ambient sound. This paper first rely on Diffie-Hellman prototype to exchanging public key between two parties and make use of similar environmental sound as authentication signature. Acoustic signal has also been utilized to measure the distance between two endpoints and decide collocation accordingly. PCASA and PIANO are two schemes to use audio signal to estimate the separation distance between two devices. The main idea is to multiply the time difference of message sent and received by sound speed to approximate the distance. But considering the processing delay it is hard to accurately estimate the time difference. It is also easily attacked by replaying attack [32, 41, 57, 113].

The above methods have a high limitation on the authentication distance. In this thesis, we present PowerKey which utilize the high frequency signal in the power grid to generate secret keys for plugged edge devices in an electrical domain. PowerKey can extend the authentication distance to a room or a floor, which will be detailed in chapter 2. Besides, in chapter 3, we exploit electromagnetic radiation emitted from the computer, CompKey, to secure unplugged wireless devices in the proximity of a computer. CompKey exhibit that devices can securely communicate with each other when they are located within $50cm$ away from the source computer.

1.3 Reliability of On-Device Classification

In this section, we discuss on-device classifier and its robustness against adversarial attacks.

1.3.1 On-Device Classification

Considering that machine learning algorithms require less computational power in inference phase than in the training phase, traditionally, the training process of machine learning algorithm is executed in the cloud and the inference phase could happen on the large computing device. However, for modern small edge devices which require fast and accurate predicting capabilities, ML classifiers must be tailored to fit and execute efficiently on those devices with limited computational power and storage capacity [69, 93].

To enable the deployment of machine learning models on resource-constrained edge devices and keep minimal loss in accuracy, ML developers leverage many model compression techniques to reduce the number of trainable parameters and minimize the number of computations thus reducing memory, energy and execution latency. straightforwardly, two way helps reduce the size of the model, lower precision of trainable weights and fewer weights. By default, numbers, weights and activations, are stored as float32 type variables. The idea of quantization is to reduce the number of bits adopted, e.g. from 32 bits to 8 bits, to significantly reduce the size of the model and computation cost. Besides, by drop-

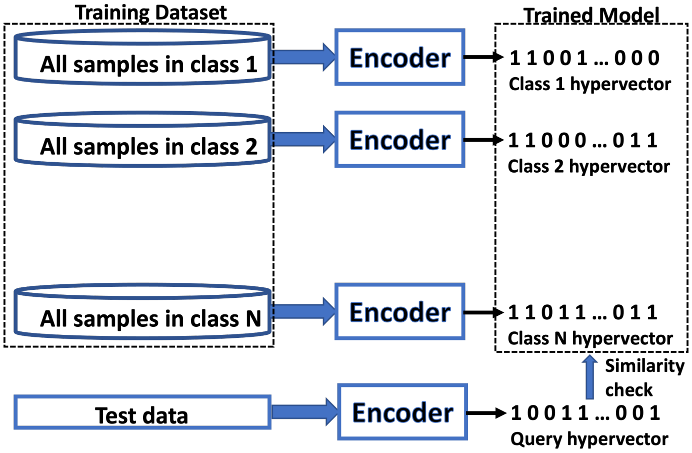


Figure 1.3: HDC classification.

ping trainable weights, pruning technique has become a powerful technique to achieve light weight machine learning models. According to the literature, there are many redundant parameters in a network that do not contribute to the performance of the model. Removing these parameters will speed up the inference process without affecting the accuracy and generalization of the model [20, 86]. By learning a small student model from a large teach model, knowledge distillation has proven to be a good practice to deploy machine learning model to small devices. In knowledge distillation, the lighter student model is trained with logits of the pretrained large teach model to realize knowledge transfer [20, 33].

Even if the above mentioned techniques are in use, it is impractical to implement these computational intensive ML algorithms on real-time tiny devices. Inspired by brain's computational abilities, hyperdimensional computing (HDC) classification, a novel lightweight framework, has open new avenue for resource-constrained applications [40]. HDC paradigm represents information with a high dimensional binarized vectors (hypervectors) and computes with hypervectors rather than conventional numerical values. The binarized hypervector could be performed with basic logical operation, which makes HDC inherently suitable for in-memory computing thus efficient on-device inference. As shown in Fig. 1.3, there are encoding, training and similarity check stage in HDC classification. The encoding stage maps different representation of inputs into hypervectors. During training, simple addition, multiplication and permutation operations are performed to generate a hypervector for each class. The inference of HDC classifier is simply by looking for similarity between encoded query hypervector to each trained class hypervector [40, 45].

1.3.2 Robustness Against Adversarial Attacks

Robustness of a machine learning model refers to the susceptibility of an ML model to intended perturbations in the input data, which are carefully crafted to attack the model. This attacking algorithm is called adversarial attacks. There are two types of adversarial settings, white-box attack and black-box attack. White-box attack indicates the strongest adversaries who have full knowledge about the target model including its parameters and architecture. On the other hand, black-box adversaries have no idea about the details of the model and hack the model only based on the model output information. Sometimes, the case where the attacker has access to the output logits is called gray-box scenario. In either setting, the aim of the adversarial is to create input data points which are visually indistinguishable from 'normal' examples but drastically change the prediction of the model. Based on the divating direction, adversarial attacks can be also be categorized into targeted attack and untargeted attack. The goal of targeted attack is to mislead the model to classify the tailored example to a target class. An untargeted attacker makes the model misclassify the perturbed image as any class other than the original true class [11, 95].

A massive of work has been developed on defensive methods to address adversarial attacks. However, one of the major drawbacks of them is that they are targeted to specific attacks and fail to generalize. Provable robustness is thus proposed to achieve generalization [23, 63, 68]. Intuitively, a machine learning model will be regarded as robust if its output is insensitive to any small changes added to the original input. The change and perturbation added to the input is usually measured using L_p norm. A certified robustness of a model within L_p norm-bounded ball refers to the situation when the model is guaranteed to give the correct answer under any attacker, no matter the strength of the attacker and no matter how the attacker manipulates the input within L_p norm range.

Unlike traditional machine learning algorithms with lots of researchers working on improving their robustness, HDC-based machine learning models are rarely studied with respect to reliability against adversarial attacks [100, 147]. In this thesis, we will focus on adversarial attacks and certified robustness of HDC-based classification paradigm in chapter 4 and chapter 5.

1.4 Thesis Contributions

In this thesis, we exploit the security at the edge side of Internet. Specifically, we present the techniques to secure communication between edge devices and display the reliability of on-device classification.

We first exhibit proximity-based secret key generation method PowerKey in chapter 2, which leverages ambient power line electromagnetic interferences (EMI) to generate secret keys for multiple plugged edge devices in an electrical domain. PowerKey includes an offline pre-processing stage using K-mean clustering to identify common EMI spikes as well as runtime extraction of EMI spike frequency for key generation. During evaluation, we conducted real experiments in two different locations — one research lab and one suite with multiple offices. Our results demonstrated that PowerKey can successfully generate secret keys in a robust and reasonably fast manner (i.e., with 100% key matching rate at a bit generation rate of up to 52.7 bits/sec).

To overcome the limitation of PowerKey that communicating devices have to be plugged into an electrical domain, we propose another proximity-based secret key generation method CompKey in chapter 3. CompKey allows wireless devices in the proximity of a computer to securely associate with each another by exploiting electromagnetic radiation (EMR) emitted from the computer. We observe that the memory bus inside a computer can emit EMR and that only devices in the vicinity of the computer can reliably extract frequency information from the signal. We design a difference-based encoding method to encode EMR's frequency information, which fluctuates randomly with time. We show via experiment evaluation that participating devices can reliably achieve around 10 bits/s bit generation rate and 100% key matching rate when they are located within 50cm away from the source computer. Moreover, the experiment results with the presence of attackers demonstrate that our method is robust against eavesdroppers and strong copy attackers who can imitate the key generation process.

In addition, in chapter 4, we dive into the adversarial attacks of on-device HDC classification. Specifically, using handwritten digit classification as an example, we construct a HDC classifier and formulate a grey-box attack problem, where an attacker's goal is to mislead the target HDC classifier to produce erroneous prediction labels while keeping the amount of added perturbation noise as little as possible. We propose a modified genetic algorithm to generate adversarial samples within a reasonably small number of queries, and further apply critical gene crossover and perturbation adjustment to limit the amount of perturbation noise. Our results show that slightly-perturbed adversarial images generated by GA-CGC-PA can successfully mislead the HDC classifier to wrong prediction labels with a large probability (i.e., 78% when the HDC classifier uses a fixed majority rule for decision).

Finally, in chapter 5, we investigate the certified robustness of an efficient HDC-based machine learning paradigm, low-dimensional computing (LDC) classification model. Concretely, we adopt interval bound propagation (IBP) technique to train a LDC classification model that is provably robust against L_∞ norm-bounded adversarial attacks. The L_∞ norm-bounded bounding box around the original input is propagated through layers of LDC model using interval arithmetic. After propagation, the worst case prediction logits can be computed based on the upper bound and the lower bound of the output bounding box. By minimizing the loss between the worst case prediction and the true label, the predicted label could be kept invariant over all possible adversarial perturbations within L_∞ norm-bounded ball. The experiment results corroborate that our trained models with IBP exhibit immunity and robustness against strong project gradient descent (PGD) attacking scheme and memory errors.

Chapter 2

PowerKey: Generating Secret Keys From Power Line Electromagnetic Interferences

2.1 Introduction

The fast growing adoption of inter-connected Internet-of-Things (IoT) devices, such as smart thermostats, WiFi access points and smart power sockets, has been dramatically changing the way we interact with our daily work and living environments. The number of edge devices, located at the edge of the Internet, is exponentially increasing. Meanwhile, demand for security as well as usability is also soaring. In particular, a crucial concern is how to quickly establish a shared secret key among various co-located IoT devices without users' manual efforts.

Today, authentication for many IoT devices are often delegated to mobile-based apps rather than performed on their own in an autonomous manner. This usually needs to be done for each IoT device through a separate mobile app, since IoT devices may not be using a unified interface provided by third-party vendors. Moreover, the current way to establish secure connections is often *one-time* (during the initial setup) and the secret keys typically remains unchanged for a long time, which poses hidden security threats.

In recent years, exploiting ambient contexts to generate dynamically shared or symmetric secret keys has been emerging as a promising solution to device authentication [53, 64, 82, 127, 140, 145, 148]. The key idea is that two or more physically co-located devices can sense similar *ambient* signals, which can serve as a proof of device authenticity. For example, the prior literature has extensively exploited radio frequency signals such as WiFi [53,83,127,140], acoustic signals [87,88,141], body electric/movement signals (for wearable devices) [78, 131, 145, 148], among many others. However, a major limitation of these techniques is that they are mainly suitable for devices that are very close to each other. For example, to leverage ambient WiFi signals (e.g., amplitude and phase) for key generation, two devices must be placed within half a wavelength (i.e., a few centimeters), since otherwise the WiFi signal's attributes can be dramatically different between the devices [127,140]. While key generation based on wireless channel reciprocity (i.e., two communicating devices will experience similar channel conditions) can apply for a longer distance [77,129,151], channel reciprocity is limited to two participating devices. Moreover, it contains little entropy in the generated keys if the two devices are relatively stationary (which is the case for indoor plugged-in IoT devices) [64,129].

More recently, [64] has considered securing IoT devices within an authenticated electrical domain (e.g., a residential house, or a company's office suite) and proposed to exploit the amplitudes of voltage harmonics in the power network for symmetric key generation. Nonetheless, as amplitudes of voltage harmonics are subject to wiring topologies and hence consistent only among nearby outlets, the key matching rate can decrease significantly (to below 90%) when the devices are a few meters away from each other. Thus, this cannot continuously secure IoT devices with a high successful rate.

Contributions. We address the limitation of unreliable key generation under the same setting considered in [64], and present PowerKey, which exploits the consistency of electromagnetic interference (EMI) spike frequencies among outlets within an authenticated electrical domain to secure plugged-in IoT devices. Concretely, multiple devices, even in different rooms connected within a shared electrical domain, can see similar EMIs generated by switching mode power supplies (SMPS). These power supplies are used by many electronic devices such as computers, printers and TVs, and create prominent frequency spikes in the $40 \sim 150$kHz range because of high-frequency switching operation [104, 114, 115]. Importantly, the frequencies of the EMI spikes vary randomly and, if detectable at participating outlets, will be the same at these outlets. Thus, they can be used as a reliable common source of randomness for symmetric key generation.

A key challenge is that most EMI spikes are limited to a small area due to very weak strengths and only a few spikes are detectable as common signals at participating outlets for legitimate devices. Thus, we propose K-means clustering as offline pre-processing to locate the frequency windows over which these common EMI spikes exist at participating outlets. At runtime, legitimate devices can extract secret key information from the selected EMI spikes.

To evaluate PowerKey, we conduct experiments in two locations — an office suite with multiple rooms and a research lab. We show that with PowerKey, devices can successfully generate symmetric secret keys in a robust and reasonably fast manner (i.e., 100% successful at a bit generation rate of up to 52.7 bits/sec). Moreover, even considering a strong attacker that knows all the details of PowerKey but collects voltage signals from an outside outlet, we show that the chance of an attacker obtaining the secret key is practically zero.

2.2 Preliminaries on Power Line EMI

2.2.1 Overview of EMR/EMI.

Electromagnetic radiation (EMR) is generated when electromagnetic fields drive the movement of atomic particles, such as an electron. Another associated concept is electromagnetic interference (EMI), which occurs whenever electromagnetic fields are disturbed by an external source through induction, electrostatic coupling, or conduction [134]. EMI can be broadly classified as radiated EMI and conducted EMI: radiated EMI (typically $>$ 300MHz) propagates in radio frequencies over the air, whereas conducted EMI ($<$ 300MHz) traverses through power lines [36].

2.2.2 Existing Research on Exploiting EMR/EMI.

EMR signals are good indicators of the system power consumption for power attacks [13]. Electronic devices plugged into power outlets also generate noises (i.e., conducted EMI) propagating through power lines [36, 96]. The prior literature has tapped into power line EMI for simple gesture recognition by sensing its EMI-induced electrical potential [24]. Also, conducted EMI strengths can be extracted to infer a television's content [28] and stealthy data exfiltration from computers [117]. Other studies include exploiting power line EMI for detecting appliance on/off activities in a smart home [36, 38], for estimating data center-level power usage information to launch load injection attacks [52], among others. In addition, the consistent deviation in power grid's nominal 50/60Hz frequency has also been leveraged for wide-area (e.g., city-scale) clock synchronization [72, 128]. By contrast, we exploit switching-induced EMI spikes in $40 \sim 150$kHz for a new and important purpose — key generation to secure IoT device communications. [1]

2.3 Problem and Threat Model

2.3.1 Problem Statement

Considering the same setting as in [64], multiple IoT devices are plugged into a power network (e.g., smart thermostats and wireless access points) and need to agree on symmetric secret keys for authenticated communications.

[1] Givena power network and a time window, the frequencies of switching-induced EMI spikes are unique (i.e., spatial-temporal uniqueness) and hence can be exploited for purposes other than key generation. For example, proof of location: when a computer is stolen and used elsewhere, the frequency statistics/patterns of EMI spikes will differ, which can prompt additional security measures such as passwords.

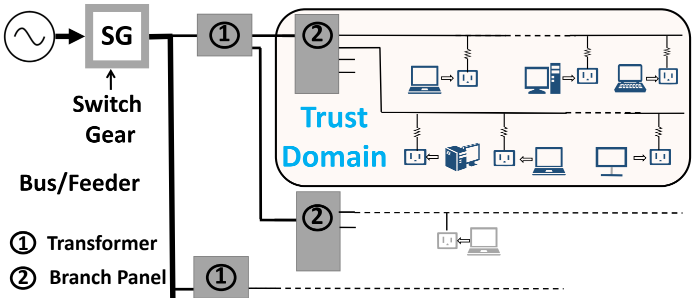


Figure 2.1: Overview of a trust domain (i.e., authenticated electrical domain in [64]).

Trust Domain. In [64], the concept of authenticated electrical domain is introduced, which is also referred to as a trust domain and can be a small single-tenant commercial building or a tenant in a large commercial building with restricted physical accesses. Fig. 2.1 illustrates a building's power network with a standard design [124]. Each panel box delivers electricity to multiple nearby rooms/outlets through parallel branch circuits protected by individual circuit breakers. In reality, each panel box often serves a small commercial building, a residential house, or a tenant (i.e., company) in an office complex, which is an authenticated electrical domain [64].

Legitimate Devices. A legitimate device can be any plugged-in device, such as smart light bulb and WiFi access point, that is physically located within a trust domain. Thus, the same as in [64], being physically in a trust domain also equals to authenticity. Legitimate devices are synchronized with a granularity of 100ms, which is not restrictive since device-to-device (wireless) communications require even better synchronization [31]. All legitimate devices can sample the voltage signals from the outlets they are plugged in [64].

2.3.2 Threat Model

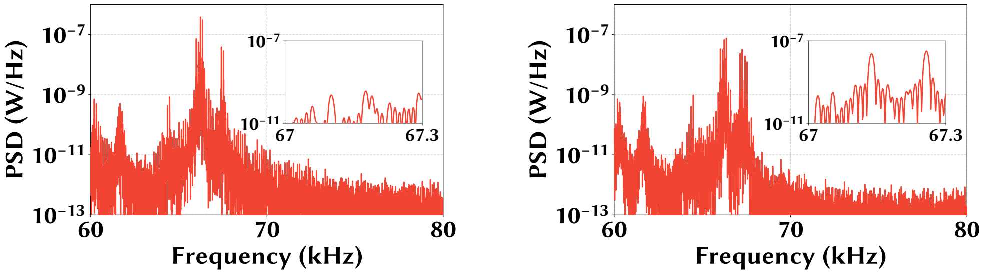
Following the threat model in [64, 87, 88, 140], attackers cannot forcibly enter the trust domain to acquire the voltage signals or obtain secret keys. The attacker is able to decode all message exchanged between any parties during key generation process. Thus, it knows all the details of PowerKey. The attacker can plug a voltage sensor into a power outlet to directly detect EMI spike frequencies. But, it can only do so outside the trust domain.

2.4 An In-Depth Look at High-Frequency EMI Spikes

All power outlets over a large area beyond a single trust domain share the same fundamental frequency as well as harmonics (i.e., multiples of 50/60Hz) [128]. Thus, the low frequency information does not meet confidentiality requirement for key generation, motivating us to explore high-frequency EMI spikes.

2.4.1 Sources for High-Frequency EMI Spikes.

Many electronic appliances (e.g., computers, televisions, compact fluorescent lights) employ switching-mode power supplies (SMPS), a crucial part of which is the high-frequency switching circuit. Moreover, a power factor correction (PFC) circuit is mandated by international regulations to improve power quality for devices with a rating of more than 75W, which applies to all desktop computers (including certain laptops) and many other appliances [115]. The core of a PFC circuit also relies on the high-frequency switching operation (typically between $40 \sim 150$kHz) [115]. Consequently, the rapid switching operation in PFC and SMPS produces high-frequency conducted EMI, which has been extensively reported by prior studies [36, 38].



(a) (b)

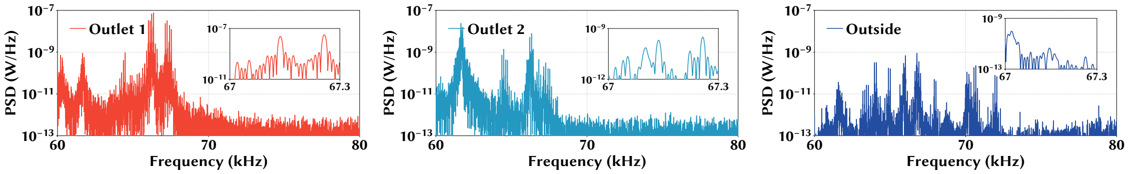
Figure 2.2: Frequency analysis of voltage signal. (a) Without the additional computer; (b) With the additional computer.

To demonstrate EMI spikes, we show in Fig. 2.2(a) the power spectral density (PSD) of voltage signals collected from a power outlet in our lab. Then, we turn on an additional desktop computer and show the new PSD in Fig. 2.2(b), which clearly demonstrates the creation of two new EMI spikes (as well as a few weaker spikes) centered around 67.2kHz.

2.4.2 Characteristics of EMI Spikes.

While the amplitudes of EMI spikes can vary significantly depending on the measurement point [64], their frequencies exhibit the following characteristics: they vary rapidly over time, and some of them can remain consistent among multiple power outlets within a trust domain. We perform fast Fourier transform (FFT) on voltage signals to examine the frequency characteristics (detailed experiment setup in Section 3.5).

Varying Randomly. The switching frequency of each SMPS unit can vary randomly within a certain range, depending on the instantaneous load and random drift-



(a) (b) (c)

Figure 2.3: PSD of voltage signals. (a) Outlet 1 in the lab. (b) Outlet 2 in the lab. (c) Outside the lab (i.e., outside trusted rooms).

ing [115]. Fig. 2.9 in the appendix presents the probability distributions of eight EMI frequencies. Note that, due to frequency orthogonality, power line communication does not interfere with switching-induced EMI spikes [5].

Some EMI Spikes are Consistent for Nearby Power Outlets. While most EMI spikes have weak strengths, we see in Figs. 2.3(a) and 2.3(b) that two different outlets in our lab still have consistent EMI spikes around 67.2kHz. The consistent EMI spikes depend on the locations of the outlets: when the set of outlets changes, the set of common EMI spikes also change.

Undetectable From Outside the Trust Domain. Most EMI spikes are localized to nearby outlets due to, e.g., fading over long wires. Moreover, because of physical isolation in different panel boxes, EMI spikes generated within a trusted domain typically vanish and become undetectable from outside the trusted domain. To see this, we collect voltage signals simultaneously both from outlets in our lab and from an outlet in a different electrical domain next to our lab. From Fig. 2.3(c), we see that the outside outlet has dramatically different frequency patterns than the outlets in our lab. Actually, even for two outlets both in our lab, their voltage signals' frequency patterns shown in Figs. 2.3(a) and 2.3(b) are different, despite the similarity over certain frequency bands.

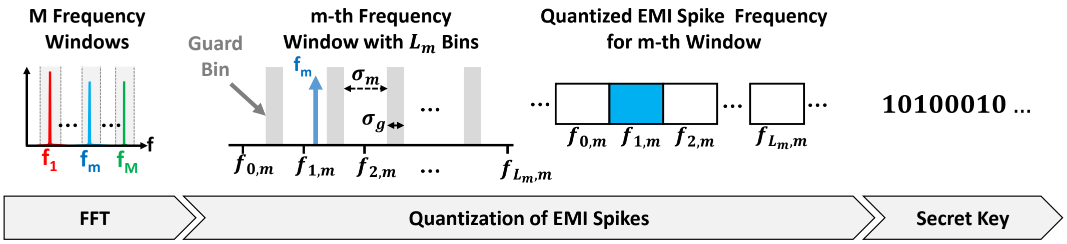


Figure 2.4: The design overview of PowerKey.

Even though a strong attacker outside the trust domain might detect some leaked EMI spikes from within the trust domain, it is very unlikely that the attacker can detect *all* the common EMI spikes used by legitimate devices for key generation because of the spatial uniqueness of conducted EMI signals [64].

2.5 The Design of PowerKey

PowerKey is built inside the power supply unit of plugged-in IoT devices. It consists of a high-pass filter (to filter out the dominant 50/60Hz component), an analog-to-digital circuit (ADC), a data communication interface, plus a micro-controller unit. PowerKey is mainly responsible for sending digitized voltage signals to the IoT device, which runs our algorithms. The total hardware cost at scale is below US$5 [64]. Note that sampling voltage signals with 300kHz or higher (to recover signals of up to 150kHz) is not restrictive, as a simple SMPS is already controlled to sample and quantize the voltage signals at a high frequency. We refer to [64] for the detailed implementation. The key difference between PowerKey and VoltKey in [64] is that PowerKey runs FFT, whereas VoltKey leverages the amplitudes of voltages harmonics. Next, we describe PowerKey in detail.

2.5.1 Offline Pre-Processing

Among numerous (mostly weak) spikes, PowerKey first identifies a set of EMI spikes, whose frequencies vary independently from each other (for more entropy) and are detectable among the participating devices.

• **Step 1.** Each device collects voltage signals for T seconds synchronously as training data and then divides the signal into $N = \frac{T}{\Delta t}$ non-overlapping segments with equal duration Δt.

• **Step 2.** The devices perform FFT analysis on each segment of their own collected voltage signals and pick up EMI spikes over the $40 \sim 150$kHz band. For the i-th segment, the devices exchange the frequencies of their own EMI spikes (i.e., local maxima of frequencies) and find the common ones, denoted by the set $\{f_1^i, f_2^i \cdots f_{M_i}^i\}$. Repeat this operation for all the N segments. Here, if the frequencies of an EMI spike at two devices have a difference no more than a threshold η, the two devices are said to have a common EMI spike.

• **Step 3.** Based on N sets of common EMI spikes, we run K-means clustering [56] to find frequency clusters. Then, we perform correlation analysis to remove strongly-correlated EMI spikes and find EMI spikes with little correlation. For each of the remaining M common EMI spike, we identify its frequency windows $[f_{m,L}, f_{m,R}]$, where $f_{m,L}$ and $f_{m,R}$ represent the lower and upper bounds of the m-th EMI spike frequency. Later, the devices use the detected frequency windows to find EMI spike frequencies at runtime.

The pseudo code is described in Algorithm 1. The K-means algorithm and correlation analysis can be run by a leading device, which then sends back the results to other devices. Re-execution of Algorithm 1 is needed only when the power network environment

Algorithm 1 Identify Freq. Windows for Common EMI Spikes

```
Collect voltage signals from devices' outlets for $T$ seconds and divide their own signals into $N = \frac{T}{\Delta t}$ segments each with a duration of $\Delta t$ seconds.
For the $i$-th segment ($i = 1, 2, \cdots, N$), compare the voltage signals of all devices and find the set of common EMI spike frequencies $\{f_1^i, f_2^i \cdots f_{M_i}^i\}$.
Based on the common EMI spike frequencies, run $K$-means clustering [56] to find $K = \max\{M_1, M_2, \cdots M_N\}$ clusters, each corresponding to one EMI spike.
Calculate the correlation coefficient matrix of the EMI spike frequencies. Only one EMI spike is kept if multiple spikes have strongly correlated frequencies.
Return $M$ frequency windows $[f_{m,L}, f_{m,R}]$ for $m = 1, 2, \cdots M$.
```

significantly changes (e.g., some common EMI spikes disappear). Note that the actual EMI frequency, not the range identified offline, is needed to extract keys at runtime.

2.5.2 Quantize Frequencies of EMI Spikes

At runtime, within a certain frequency window, the common EMI spike can result in slightly different frequencies at different devices due to measurement errors. Thus, we quantize EMI spike frequencies into discrete bins. We introduce a hyperparameter σ as the quantization threshold. In this paper, if the frequency difference is no more than σ Hz for 80% of the time, then σ is chosen as the default quantization step size. To further mitigate the frequency discrepancies, we insert a guard frequency band of size σ_g between two valid quantized frequency bins. Fig. 2.4 provides an illustration of the frequency quantization. For example, a device detects a EMI spike frequency of f within a frequency window $[f_L, f_R]$

and the chosen quantization step size is σ. Then, the frequency is quantized into a bin with index of $\lfloor \frac{f - f_L}{\sigma + \sigma_g} \rfloor$.

2.5.3 Extract Secret Keys

For key generation, participating devices convert indexes of valid EMI frequencies into binary bits using, e.g., Grey codes. Then, the devices shall exchange the information to remove invalid EMI spikes whose frequencies fall into guard bins. Finally, they perform reconciliation and privacy amplification.

Converting Frequency Index Into Binary Bits.

If the EMI spike frequency at any participating device falls into an invalid frequency guard band, then it becomes less certain to decide its corresponding frequency bin. Thus, the corresponding EMI spike window is discarded to avoid secret key discrepancies. The devices first find their own invalid windows (if any) and exchange this information with other participating devices. For the remaining valid EMI spike windows, the indexes of their frequency bins will be converted into binary bits.

Reconciliation.

For better presentation, we focus on two legitimate devices, i.e., Alice and Bob, while it can also be extended to more than two devices [64, 87]. Based on the valid EMI spike frequency windows and indexes, Alice and Bob each end up with a n-bit sequence, denoted by $\widetilde{K}_a$ and $\widetilde{K}_b$, respectively. While it is rare to have different $\widetilde{K}_a$ and $\widetilde{K}_b$, it can still occur in practice.

To improve the key matching rate between Alice and Bob, we apply a crucial step — reconciliation process [82, 148], which uses error correction coding to fix the bit differences/errors at the expense of slowing down bit generation rate. Specifically, the key idea is that both Alice's n-bit sequence $\widetilde{K}_a$ and Bob's n-bit sequence $\widetilde{K}_b$ can actually be viewed as error-corrupted versions of a shared symmetric key, and errors can be fixable using error correction coding. Consider an (n, k, r) error correction code scheme $\mathcal{C}$, which maps any k-bit sequence into a n-bit codewords $(n > k)$ through a one-to-one encoding function and can correct up to r error bits. Meanwhile, there exists a many-to-one decoding function that maps any n-bit string into one of the 2^k valid codewords. Let $g_e(\cdot)$ and $g_d(\cdot)$ be the encoding and decoding functions of $\mathcal{C}$, respectively. First, Alice can first decode its n-bit string $\widetilde{K}_a$ and then produces the codeword $g_e(g_d(\widetilde{K}_a))$ that is the closest to $\widetilde{K}_a$. Then, Alice computes the bit-wise error string $\Delta\widetilde{K} = \widetilde{K}_a - g_e(g_d(\widetilde{K}_a))$ and sends it to Bob, which can be in cleartext without encryption. Then, if the bit error rate is roughly estimated and the number of error bits is no more than r, Bob can obtain Alice's n-bit sequence $\widetilde{K}_a$ with a high probability based on $\Delta\widetilde{K} + g_e(g_d(\widetilde{K}_b - \Delta\widetilde{K}))$.

To sum up, if $\widetilde{K}_a$ and $\widetilde{K}_b$ generated from Alice's and Bob's respective quantized EMI spike frequencies differ in no more than r bits, the reconciliation process using the coding scheme $\mathcal{C}$ can ensure that both Alice and Bob eventually possess the same n-bit string.

Privacy Amplification.

During the reconciliation process, Alice's bit-wise error string $\Delta\widetilde{K} = \widetilde{K}_a - g_e(g_d(\widetilde{K}_a))$, which contains partial information of its n-bit string $\widetilde{K}_a$, is communicated to Bob and meanwhile also possibly leaked to attackers. To address the leakage of partial information about the keys, privacy amplification can be applied: instead of using all the n-bit strings to generate their keys, Alice and Bob can shrink their n-bit strings by $(n - k)$ bits to properly create k-bit strings, thus preventing attackers from acquiring partial information about the k-bit strings [82, 148].

2.6 Evaluation Methodology

2.6.1 Experiment Setup.

We conduct experiments in two different trust domains — an office suite with multiple individual rooms and a research lab, as shown in the appendix. The office suite is shared by multiple faculty members while the lab has more than 20 workstations. We use the office suite as our default location with multiple faculty offices accessible through a corridor.

Voltage Signal Collection and Processing. For proof of concept, we use a Rigol 1074Z oscilloscope as a proxy ADC to collect voltage signals from the power outlets that are then transferred to a laptop for processing, while one can also follow the design in [64] and insert an additional FFT module.

Error Correction Coding. We use the following commonly-used error correction coding (ECC) schemes with varying degrees of error tolerance [22]. (i) *Hamming Code*, a linear perfect error correction scheme that encodes every 4 bits of data with 3 parity bits and can withstand 1-bit error in the data. (ii) *Golay Code*, another linear code which encodes 12 bits data into 23 bits and can correct up to 3 error bits. (iii) *Reed-Solomon Code (RS)*,

Table 2.1: Frequency Quantization Schemes.

Quantization Scheme	**Q1**	**Q2**	**Q3**	**Q4**	**Q5**
Valid Frequency Bin Size (Hz)	σ	σ	σ	σ+1	σ+1
Guard Bin Size (Hz)	0	σ-1	σ	σ-1	σ

a non-linear cyclic code that can detect and correct multiple errors: an $RS(n,k)$ encoding can correct up to $\lfloor \frac{n-k}{2} \rfloor$ bit errors. In our evaluation, we use three variations of the RS code — $RS(7,3)$, $RS(15,5)$, and $RS(15,3)$.

Frequency Quantization and Guard Bin Size. We set σ as the step size if the frequency difference between any two outlets is no greater than σ for 80% of the time. As shown in Table 2.1, we test five different quantization schemes with varying step sizes and guard bands, denoted as $Q1, Q2 \cdots, Q5$.

Experiment Durations. We first collect 500 seconds of voltage data simultaneously from the chosen power outlets to identify the common EMI spike windows offline (Section 2.5.1), and determine the quantization scheme. We use $\Delta t = 100$ms as the length of each voltage signal segment. For online evaluation, we use the same segment length and run the experiments for 60 minutes.

2.6.2 Evaluation Metrics.

We consider the following standard metrics to evaluate our algorithm in terms of speed, accuracy and randomness.

- **Bit Generation Rate.** It is the number of secret bits generated per unit time. Consider a segment size of Δt seconds and M common EMI spikes with frequency windows $[f_{m,L}, f_{m,R}]$, quantization step size σ_m and frequency guard band size $\sigma_{g,m}$, for

$m = 1, 2, \cdots M$. The bit generation rate (BGR) in bits per second with ECC $\mathcal{C}(n, k, r)$ is given by BGR $= \frac{k}{n} \cdot \frac{1}{\Delta t} \sum_{m=1}^{M} \log_2 \lfloor \frac{f_{m,R} - f_{m,L}}{\sigma_m + \sigma_{g,m}} \rfloor$.

- **Bit Error Rate.** It indicates the probability of differences between secret keys extracted by two or more devices. A low bit error rate (BER) is desirable.

- **Key Matching Rate.** This indicates, on average, the percentage of keys generated by PowerKey can be used as a valid shared secret key. We use the standard AES 128-bit key as the length requirement.

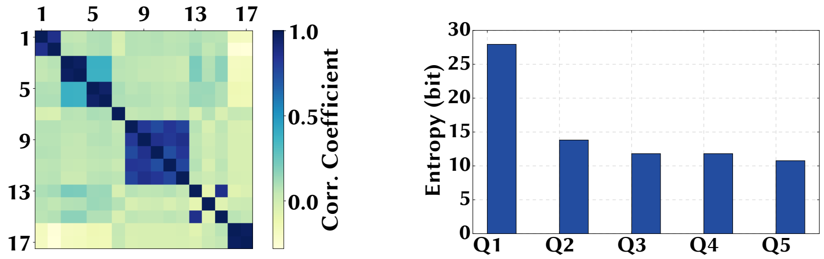
In addition, we also consider Entropy and Mutual Information. Entropy measures the amount of information contained in the random variable we generate from the EMI spike frequencies. Mutual information quantifies the amount of dependency between two random variables and we use this to measure the information possibly obtained by an attacker.

2.7 Evaluation Results

In this section, we present our evaluation results in the office suite, while the results in the lab are deferred to the appendix. Our results demonstrate that with the design of PowerKey, multiple devices can successfully generate symmetric secret keys in a robust and fast manner (i.e., with a 100% key matching rate at a bit generation rate of 52.7 bits/sec).

2.7.1 Analysis of EMI Spike Frequencies.

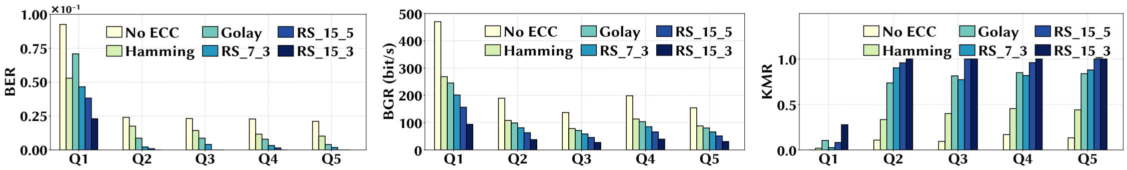
By pre-processing the voltage signals in the office suite, we identify a total of 17 common EMI spikes out of hundreds of spikes. As shown in Fig. 2.5(a), only 8 of the 17 spikes are uncorrelated, while the remaining spikes are redundant and need to be removed.



(a) Office

(b) Entropy

Figure 2.5: (a) Correlation coefficients of EMI spike frequencies in the office. (b) Entropy with different quantizations.



(a) Bit error rate

(b) Bit generation rate

(c) Key matching rate

Figure 2.6: Performance of PowerKey in the office suite.

We also show the histograms of the 8 independent EMI spike frequencies and the frequency differences at the two outlets in Fig. 2.9 and Fig. 2.10 in the appendix, respectively. It can be seen that each of the 8 EMI spike frequencies varies within a narrow window. We also run randomness test on frequencies of the 8 EMI spikes in Matlab using runstest(·). The results are all positive, verifying the randomness of EMI spike frequencies with a 95% significance level [84].

2.7.2 Performance of PowerKey.

We now examine the performance of PowerKey.

Entropy of EMI Spike Frequencies. Fig. 3.8(a) shows the impact of our quantization configurations on the overall entropy of the 8 EMI spike frequencies. Naturally, when the EMI spike frequency is mapped to fewer bins, the amount of entropy also decreases but still is better than some of the existing literature whose ambient signals can only have $1 \sim 2$bits [82, 148].

Bit Error Rate. We now look at the bit error rate under different quantization and ECC schemes and show the results in Fig. 2.6(a). We see that either quantization or ECC alone cannot achieve a low bit error rate. By combining quantization with an appropriate ECC scheme (e.g., $RS(15, 5)$ or $RS(15, 3)$), PowerKey essentially achieves a zero bit error rate in practice.

Bit Generation Rate. We show the bit generation rate in Fig. 2.6(b). As in the prior literature [82, 148], the bit generation rate only considers how many secret key bits Alice and Bob can generate, without accounting for possible errors. Clearly, both quantization and ECC reduce the bit generation rate, but they are needed to achieve a high key matching rate as we show next.

Key Matching Rate. Next, we show the key matching rate (KMR) between Alice and Bob in Fig. 2.6(c) for the standard AES 128-bit key. We see that ECC plays a vital role to correct mismatched bits between Alice and Bob. Specifically, the RS codes perform the best, achieving nearly 100% key matching rate when combined with quantization. By contrast, when using amplitudes of voltage harmonics for key generation for devices 18m (approx. 60ft) away, the key matching rate reduces to below 90% [64].

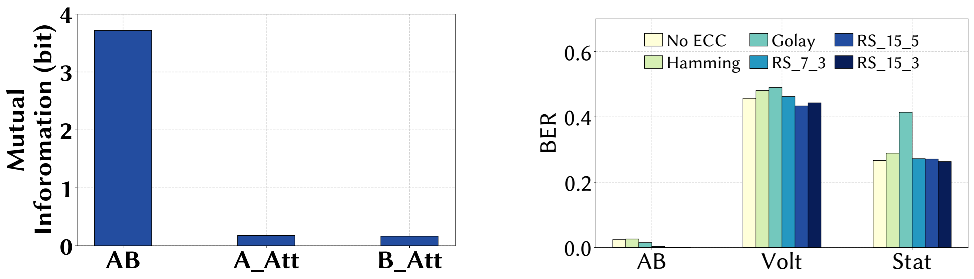
2.7.3 Security Analysis of PowerKey.

We consider an attacker that can collect voltage signals from outside the trust domain, be synchronized with Alice/Bob, and knows all the details of PowerKey (including the common EMI spike windows located offline). In our experiment, we choose an outlet next to the entrance to our office suite. We assume that the attacker uses its most prominent EMI spikes, or estimates the EMI spike frequencies based on their probability distribution, within each valid EMI frequency window. Thus, the attacker is assumed to follow the same procedure as a legitimate device, except for that it extracts EMI spike frequencies from outside the trust domain.

We first calculate the mutual information between two parties in Fig. 2.7(a). We see that the mutual information between the attacker and Alice/Bob is much lower compared to that between Alice and Bob, thus showing that the attacker's signal contains little information about Alice's/Bob's. Next, we show the bit error rate in Fig. 2.7(b) for quantization scheme Q4 (Table 2.1) and see that, under various strategies, the attacker's error rate is significantly higher than that of Alice/Bob, resulting in almost random bits. Further, it achieves a practically zero key matching rate, and hence we omit the result. The reason that the attacker is not able to acquire the secret key is that the common EMI spikes located offline are spatially unique to the power outlets to which legitimate devices are connected.

2.8 Related Works

For key generation, the prior research has exploited radio frequency signals such as WiFi [53, 83, 127, 140], acoustic signals [77, 87, 88, 141], body electric/movement signals (for wearable devices) [78, 131, 145, 148], among many others. Nonetheless, the existing approaches can suffer from a limited distance [53, 83, 127, 140], low key matching rate [148], and/or low bit generation rate [82, 83, 127].While key generation based on wireless channel



(a) (b)

Figure 2.7: (a) Mutual information: "AB" (Alice-Bob), "A-Att" (Alice-Attacker), and "B-Att" (Bob-Attacker). (b) Bit error rate. "AB" means Alice/Bob; "Volt" means the attacker uses the highest EMI spike for each window from its collected signals; "Stat" means estimating the EMI spike frequencies based on their probability distributions.

reciprocite can apply for a longer distance [77, 129, 151], channel reciprocity often needs time-division multiplexing and is limited to two participating devices each time. Moreover, it contains little entropy in the generated keys if the two devices are relatively stationary [129]. Other studies [87, 88] look at secret key generation within a single room by utilizing ambient acoustic/luminous characteristics, but they require long-term statistics of the ambient signals and hence take several minutes or even longer to produce a valid key.

The recent study [64] considers key generation for plugged-in IoT devices under the same setting as ours, but it leverages amplitudes of voltage harmonics that are consistent only among nearby outlets. Thus, when the inter-device distance increases (e.g., 10m), the key matching rate can significantly decrease.

Finally, our work is also relevant to studies that exploit conducted EMI for side channel inference/attacks [28, 105, 117]. Nonetheless, PowerKey is novel in that it exploits EMI spike frequencies for an orthogonal and important goal — secret key generation.

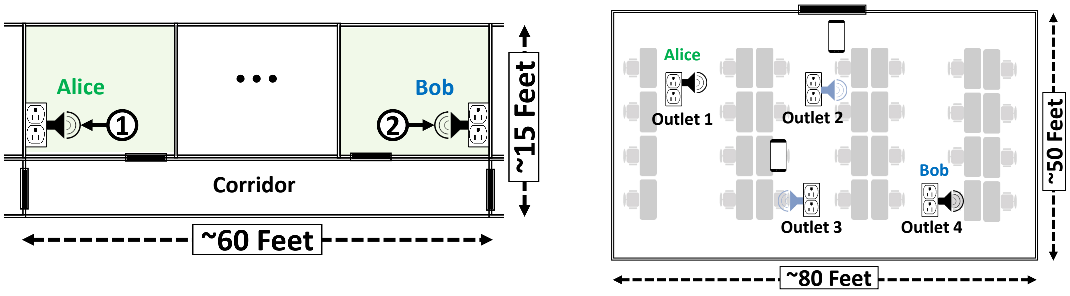
2.9 Conclusion

In this paper, we proposed a novel key generation approach, called PowerKey, based on EMI spikes in an authenticated electrical domain. PowerKey includes an offline pre-processing stage to identify common EMI spikes as well as runtime extraction of EMI spike frequency for key generation. For evaluation, we conducted real experiments in two different locations — one research lab and one suite with multiple offices. Our results demonstrated that PowerKey can successfully generate secret keys in a robust and reasonably fast manner (i.e., with 100% key matching rate at a bit generation rate of up to 52.7 bits/sec).

2.10 Appendix

2.10.1 Experiment Setup.

We conduct experiments in two different trust domains — an office suite with multiple individual rooms (Fig. 2.8(a)) and a research lab (Fig. 2.8(b)).

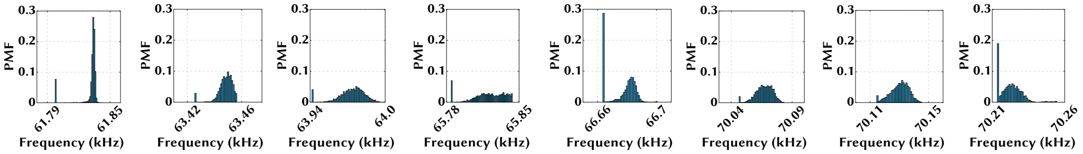


(a) Office (b) Lab

Figure 2.8: (a) Layout of the office. (b) Layout of the lab.

2.10.2 Analysis of EMI Spike Frequencies in the Office Suite.

We show the histograms of the 8 independent EMI spike frequencies and the frequency differences at two outlets in Fig. 2.9 and Fig. 2.10, respectively. We see that the two outlets share certain time-varying EMI spike frequencies with only minor differences.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 2.9: Histogram of 8 different EMI spike frequencies in the office suite.

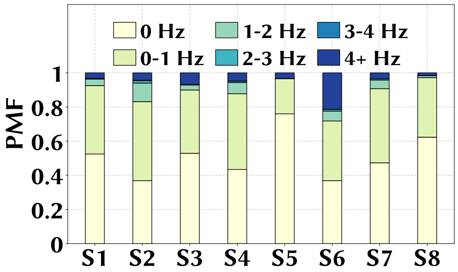
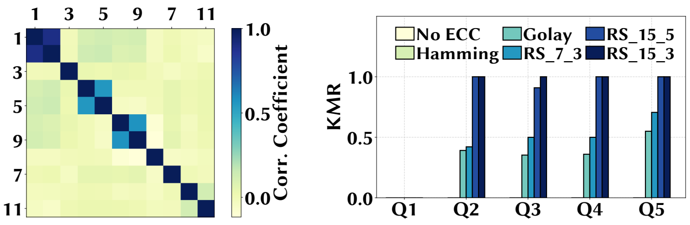


Figure 2.10: Distribution of 8 different EMI spike frequencies in the office suite. "S-n" means the n-the EMI spike. $\sigma = 1, 1, 1, 1, 1, 4, 1, 1$Hz for the 8 EMI spike windows, respectively.

2.10.3 Results for Key Generation in the Lab

We now run experiments in a lab with 20+ desktops shown in Fig. 2.8(b).

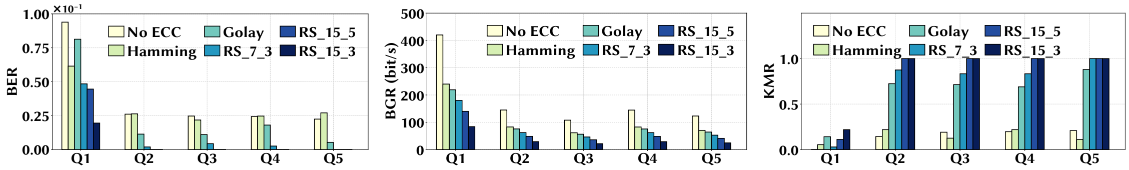
Analysis of EMI Spike Frequencies. After offline pre-processing, PowerKey identifies a total of 11 EMI spikes for the lab. Then, as shown in correlation analysis in Fig. 2.11(a), 8 of the 11 spikes are uncorrelated, while the remaining ones are redundant and need to be removed.



(a) Lab (b)

Figure 2.11: (a) Correlation coefficients of EMI spike frequencies in the lab. (b) Key matching rate for four devices in the lab.

Key Generation Performance. We show the key generation performance for the lab. The main results are deferred to Fig. 2.12. We can see that in terms of all the evaluation metrics, the performance of PowerKey is consistent with that in the office setting. Likewise, the attacker can barely obtain secret keys successfully, with a high bit error rate and practically zero key matching rate.



(a) Bit error rate (b) Bit generation rate (c) Key matching rate

Figure 2.12: Performance of PowerKey in the lab.

Multiple Devices. Next, we consider four devices associated with four outlets in Fig. 2.8(b). Our results in Fig. 2.11(b) show that with an appropriate quantization and ECC scheme, PowerKey can still generate secret keys with a negligible bit error rate and almost 100% key matching rate, demonstrating its reliable key generation.

Chapter 3

CompKey: Exploiting Computer's Electromagnetic Radiation for Secret Key Generation

3.1 Introduction

Device-to-device (D2D) technology is a crucial part of the next-generation wireless communications, allowing mobile devices to communicate with each other directly when they come into proximity [4, 118]. Nonetheless, it also faces privacy and security concern because of the broadcast nature of wireless transmission [53, 82, 127, 140, 148].

Traditionally, to guarantee security and secrecy, the transmitted data is encrypted through cryptographic techniques, such as the classic public key encryption Diffie-Hellman (DH). However, since DH does not verify the identity of participating devices, an attacker with a directional antenna could easily impersonate a legitimate device and establish a shared key with one or both of the valid devices [25,120,140,151].

To address the limitations of DH, recent researchers exploit the dynamic characteristics of natural ambient signals in order to construct a secure and authentic communication channel between two or more collocated devices. Concretely, devices in proximity of each other can derive a shared secret key from their common time-varying ambient environment. In addition, the physical proximity can serve as proof of device authentication. The ambient sources appearing in literature include radio related signal, such as RSSI and CSI [73,82,83,116,127,140], ambient audio signal, ambient luminosity [57,87,113,141] and biometrics [78,131,148].

However, the existing methods have significant limitations. First, these methods may require legitimate devices to be placed very close to each other, e.g., less than 4cm in WiFi-based approaches [82,127,140]. The reason is that they use signals' amplitude attribute for key generation, which changes dramatically beyond a half-wave length distance. Similarly, to extract identical biological characteristics for secret key generation, participating devices have to be touched by a single person [78,131,148]. Second, some of these methods rely on specialized sensing apparatus. For example, leveraging biological signals requires specialized receiving sensors, like electromyography sensor or accelerometer [78,148]. Third, some of these methods may take a long time to generate the secret key. For example, techniques based on Received Signal Strength Indicator (RSSI) have largely limited key generation rates, since only one RSSI value can be extracted from one packet [53,82,83,116]. Exploiting the ambient sound or ambient luminosity based on its (slow-varying) statistics is also subject to a low key generation rate [87,88]. Last but not least, some of these methods require two devices directly exchange signals between each other (e.g., channel reciprocity), which makes it not suitable for multiple-device key generation [129,151].

In this paper, we discover a computer's electromagnetic radiation (EMR) signal as a localized (hence secure only for devices nearby the computer), and randomly varying ambient signal for secret key generation. Moreover, we exploit the computer EMR's frequency information, instead of amplitude (like WiFi-based approaches [151]), to overcome the half-wavelength distance requirement and preserve integrity among legitimate devices that are within a range of the source computer (empirically 50cm in Section 4.6).

Concretely, we design CompKey, a scheme leveraging EMR over the memory bus clock frequency of a computer to generate shared secret key between two or more legitimate devices which are close to the computer. In order to extract the frequency information of the radiation, we first perform Fast Fourier Transform (FFT) to get the frequency spectrum and locate the frequency band based on the frequency of the most dominant spike. We then filter the signal over this frequency band to get rid of other interfering spikes. Based on the filtered signal, we extract the frequency of the highest spike at every time step and get the time-varying frequency information. This paper adopts difference-based encoding method instead of quantization approach, which is prone to long run zeros and ones. We convert the frequency difference of two adjacent time steps to binary bits and use reconciliation process to get rid of the bit discrepancy caused by fading effect and imperfection of measuring equipment. To evaluate CompKey, we conduct experiments in our lab office, showing that CompKey can achieve 10 bits/s bit generation rate and 100% key matching rate.

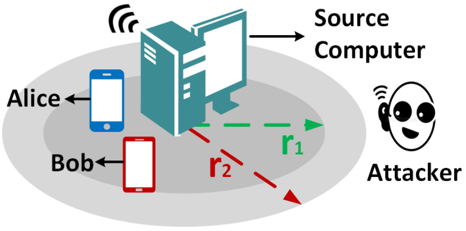


Figure 3.1: Two parties (Alice and Bob), who are located in the vicinity of a computer (within r_1 from the computer), can perceive the same radiation emitted from the computer. An attacker can only reside beyond some distance, r_2, from the source computer.

3.2 Problem and Threat Model

3.2.1 Problem Definition

Fig. 3.1 provides an illustration of CompKey. Alice and Bob are located within r_1 from a computer (hereafter called source computer) and would like to securely communicate with each other. An attacker can only reside r_2 away from the source computer to launch attacks. In practice, r_1 and r_2 can be both conservatively chosen (i.e., small r_1 and large r_2) to ensure that legitimate devices can have a high success rate of key generation while keeping attackers away at a safer distance.

Source Computer. A computer that emits EMR signals for nearby legitimate devices for authentication is called a source computer. The source computer can be a work desktop or regular personal laptop, but cannot be the small size tablet computer that does not emit significant EMR.

Legitimate Devices. Legitimate devices are devices located near the source computer and would like to securely communicate with each other. They could be mobile phones, laptop, and wearable devices. Note that the source computer itself can also be a legitimate device.

In Section 4.6, we will show that legitimate devices need to be located within 0.5m from the source computer. Legitimate devices can receive the source computer's EMR signals, whose frequency ranges (e.g., around 800MHz and 1600MHz) are close to those of current cellular/WiFi signals. In other words, the existing antenna on mobile devices for receiving cellular and WiFi signals is also capable of capturing computer's EMR signals, provided that its receiving frequency is tuned to proper frequency ranges.

Potential Applications. Nowadays, it is very common that devices interact with each other nearby through wireless channels. For example, mobile devices are often used for verification purposes to access laptop or other office resources, while personal health data is frequently exchanged between wearable devices and smart phones. By using CompKey, we can build a local circle of trust for interaction between nearby devices around a source computer.

3.2.2 Threat Model

We assume that legitimate devices are located in the proximity of a source computer and malicious adversaries can only launch attacks at a distance of at least r_2 away from the source computer. The attacker cannot put any tapping device around the source computer as otherwise it would be discovered when approaching the computer. We consider the following two kinds of attacks.

Eavesdropping Attack. The attacker can overhear the EMR signal emitted from the source computer at a distance and also eavesdrop the legitimate devices' communication.

Copy Attack. The attacker can not only capture all the information transmitted over public wireless channel but also obtain component details of the source computer.

Moreover, the attacker is able to imitate the computer's working status during the key generation process. We refer to this strong adversary as copy attacker. Thus, the copy attacker is able to find another computer with exactly identical memory bus clock and play the same programs in that computer to imitate the memory bus status. The attacker then records the radiation wave and generates its own key following CompKey steps.

3.2.3 Remarks

- **Software-Defined Radio (SDR) Capability.** While the size of existing antennas on mobile devices is suitable for capturing computer's EMR signals, the receiving frequency needs to be tuned to proper frequency ranges. Fortunately, such SDR capability is being integrated by major vendors like Intel into baseband solutions (to accommodate multi-standard communications with a low cost) [51]. Thus, SDR does not present an insurmountable barrier and CompKey will be more universally applicable in future devices.

- **Synchronization.** Like in other proximity-based authentication schemes [87, 127, 151], legitimate devices need to be synchronized for secret key generation by using CompKey. The synchronization requirement in CompKey is easy. Specifically, as shown in our experiments in Section 4.6, CompKey extracts EMR frequency every 0.08s, for which synchronization requirement is much less stringent than for normal communication that needs millisecond-level synchronization.

- **Authentication Distance.** Through empirical evaluation, we consider $r_1 =$ 0.5m away from the source computer as the authentication distance, within which devices can reliably extract secret keys. On the other hand, attackers are kept at a distance of $r_2 =$ 2m away from the source computer. Although 1.5m is already enough to stop attackers from getting keys, we use 2m as a *safer* distance limit. This is common in the literature of proximity-based authentication [82, 127] where a distance greater than the authentication distance is assumed to keep attackers away.

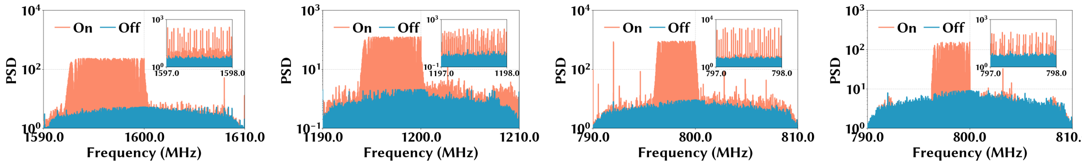
- **Other Attacks.** We discuss a few other attacks.

Jamming Attacks. In our threat model, the attacker's goal is to obtain secret keys, instead of completely blocking Bob/Alice communications. If very strong noise is injected, then CompKey may not work, but the attacker cannot obtain keys either. Thus, like in the existing proximity-based authentication [82, 88, 127, 131, 141, 148], we do not consider attackers who inject or jam radio signals. In fact, legitimate devices can also easily discover the existence of such an attacker: in the presence of such an attacker that injects a high EMR signal to mimic the source computer's signal, the resulting EMR signals received by legitimate devices will not decrease significantly when the legitimate devices moves some distance away from the source computer.

Untrustworthy Source Computer. Like in the existing literature [78,88], we assume the EMR signal produced by the source computer is not directly compromised by attackers. Even if an attacker can compromise a source computer and acquire its random EMR signals, it also needs to know exactly when Bob and Alice tap into EMR signals to extract keys, which can be non-trivial for attackers.

- **Limitations.**

CompKey is designed to provide additional protection (e.g., as part of multi-factor authentication). And such additional protection is successful only when attackers are at an adequate distance from the source computer. If our threat model is violated (e.g., an



(a) HP ProBook 450 (b) Dell XPS 8920 (c) Acer Aspire V3-372T (d) Dell OptiPlex 9020

Figure 3.2: Frequency spectrum of memory bus EMR for four different kinds of computers. The blue line represents the EMR that is obtained when the computer is off. And the red line is the captured signal when the source computer is turned on.

attacker is hidden inside a compromised source computer), then CompKey may not work as designed, which is also the limitation in other proximity-based authentication [82, 88, 127, 131, 141, 148]. In such a case, however, it can still be non-trivial to acquire the secret key because the attacker also needs to know exactly when Bob and Alice tap into random EMR signals for key generation.

3.3 Characteristics of Computer EMR

3.3.1 Memory Bus EMR

Random access memory (RAM) refers to computer memory that temporarily stores and retrieves data at a high speed, which will be processed by the CPU. Data transferring process between CPU and RAM is controlled by memory bus clock. During this process, state transition of digital circuit will induce transitioning voltage signal, which causes changes in electric fields. Data transferring through memory buses will introduce an alternating current, which leads to variation in magnetic fields. The combination of changing electric field and changing magnetic field is termed electromagnetic field (EMF). Digital data paths connecting CPU and RAM have long unterminated wires and can serve as an-

Table 3.1: Memory bus clock frequency information of different computers.

Computer	**CPU**	**RAM**	**MC (MHz)**
HP ProBook 450	Core i5-6200U	DDR4 SDRAM	1600
Dell XPS 8920	Core i7-7700	DDR4 SDRAM	1200
Acer Aspire V3-372T	Core i5-6200U	DDR3L SDRAM	800
Dell OptiPlex 9020	Core i5-6200U	DDR3 SDRAM	800

Note: MC represents memory clock frequency.

tennas, through which EMF can be radiated to the outside world as EMR. Since the EMF changing rate is controlled by memory bus clock, the EMR frequency should correspond to the memory clock frequency [39].

In order to validate this, we collect the radiation signal from four different computers, the detailed information of which is provided in Table 3.1. In the table, we provide the type of CPU and RAM. Especially, we provide the memory clock frequency of each computer. For each computer, we collect the radiation in situations when the computer is turned on and off. We use a Universal Software Radio Peripheral (USRP) to capture the emitted radiation, tuning it to 20MHz frequency range centered at the the computer's referred memory bus clock frequency. After that, we perform Fast Fourier Transform (FFT) to get its frequency spectrum with 1Hz frequency resolution, which is shown in Fig. 3.2. We can see that for each computer compared to the case when the computer is turned off, there are prominent spikes appearing and clustering around the its memory bus clock frequency when it is turned on. This further verifies that the memory bus clock frequency information can be disclosed through the radiation.

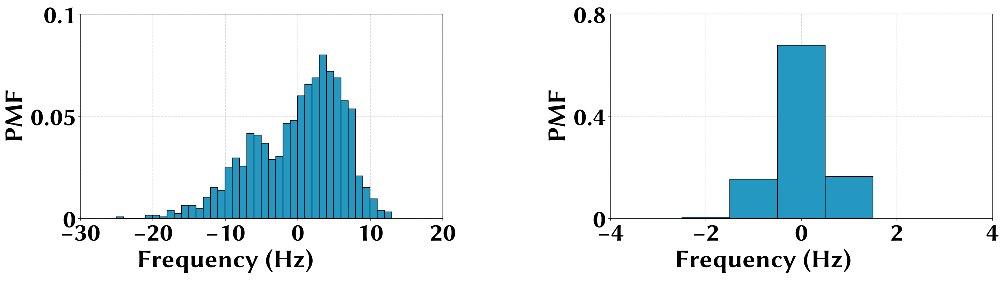
3.3.2 Memory Bus EMR as a Secret Source

The EMR signal needs to meet the following requirements.

Temporal Variation

Memory bus clock is controlled by a clock generator, an electronic oscillator, which aims to synchronize the data transferred between CPU and RAM. Due to minor variations in temperature, silicon characteristics and local electrical conditions, these crystal-based oscillators are subject to diverge and run at slightly different rate from the reference frequency, which is termed clock drift. This physical phenomenon is proved genuinely random and acts as the non-deterministic random source in many hardware random-number generators [135]. Since the frequency of emitted EMR derives from memory bus clock, the frequency information of the EMR signal is accordant with clock drift, which makes it a good candidate to be a random source in authentication key generation.

We run empirical experiments to validate the randomness of the desired EMR frequency. Using USRP, we collect the radiation for 100 seconds. Over every 0.1s, we perform FFT and extract the frequency information. We get 1000 frequency samples in total. We will explain how to track the EMR frequency in Section 3.4. For brevity, we only show results for the Acer Aspire V3 computer. Fig. 3.3(a) gives the probability mass function (PMF) of the 1000 frequency samples. Note that the frequency value shown in the figure is shifted to have a zero mean. We can see that the frequency distributes randomly over 50Hz frequency range. We extract the frequency information of EMR signals, instead of their amplitudes, because amplitudes of EMR signals change dramatically with the distance



(a)

(b)

Figure 3.3: (a) Histogram of EMR frequency of Acer Aspire V3. (b) Histogram of frequency difference between two devices collecting EMR signals 0.5m away from Acer Aspire V3.

and remain approximately unchanged within less than a half wavelength. In other words, if we use EMR's amplitudes as the randomness source for key generation, our authentication distance would be less than 10cm given the EMR's wavelength.

Integrity and Authenticity

In the beginning of this section, we already know that the EMF caused by alternating current going through the memory bus can be radiated to the outside. As long as a device is close to the computer, it will sense the radiation with sufficient energy and extract the EMR spikes through FFT analysis. While EMR signal amplitudes varies significantly over distance, its frequency is less affected by distance, thus meeting the integrity requirement. Fig. 3.3(b) gives the distribution of frequency difference between two participating users, who are placed near a source computer and synchronously detect the radiation emitted from it. We can see that two users get exactly the same normalized frequency for around 70% of the time. The absolute frequency difference between two users is less than 2Hz for almost 100% of the time, which further corroborates the integrity requirement.

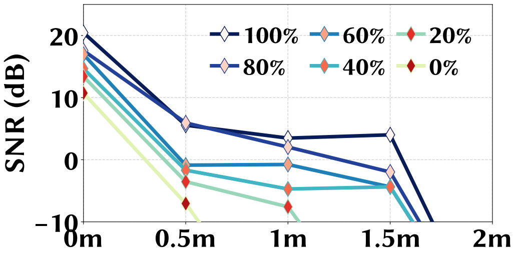
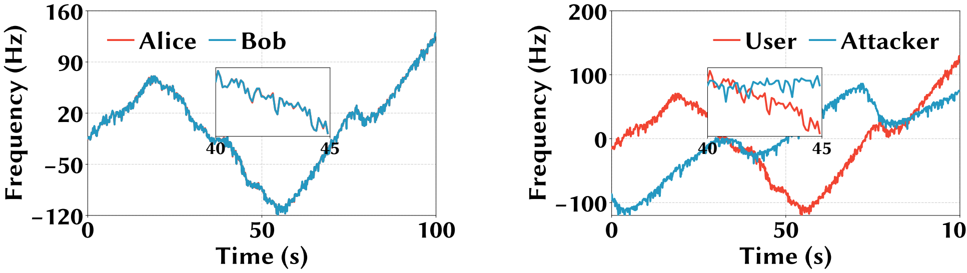


Figure 3.4: Acer Aspire V3-372T: SNR vs. distance under different RAM access frequencies.



(a)

(b)

Figure 3.5: (a) The normalized EMR frequency variation pattern during 100s when Alice and Bob are placed close to a source computer and synchronously collect the radiation data. (b) The normalized EMR frequency variation pattern of Alice and an attacker. The attacker launches a copy attack and EMR using its receiving device.

Like the existing proximity-based authentication [127,151], authenticity is ensured by not allowing malicious attackers approaching the source computer when executing key generation process. Because of the fading effect over distance, the attacker far away from the source computer will not detect the radiation clearly and barely get the secret key. Thus, the party who could generate the correct secret key by hearing the radiation of the source computer is the one that is in the vicinity of the computer and hence is a legitimate participating device by our threat model.

Confidentiality

Confidentiality is another criterion for the signal to be a secret key source. We will elaborate confidentiality of memory bus EMR from the perspective of eavesdropping attacker and copy attacker.

Eavesdropping Attacker As a computer's EMR is naturally a low-energy signal for human safety and compliance requirement [76], the emitted EMR can not propagate a long distance. In addition, the electromagnetic waves are subject to inverse-square law propagation loss. Thus, as long as the eavesdropper is kept some distance away from the source computer, it can not capture the radiation with sufficient energy and is not able to get accurate frequency information.

To show the fading effect over distance, we show the experimental results for Acer Aspire V3-372T, while the results for other computers are similar. We control and change RAM access frequency from 0% to 100% (i.e., percentage of time the RAM is transferring data with CPU). For each RAM access frequency, we collect EMR signals from different distances, from 0m to 2m, and calculate signal to noise ratio (SNR) of the radiation. The value of SNR in dB can be calculated as $SNR = 10 \times \log_{10}(P_s/P_n)$, where P_s is power of EMR signal (excluding noise power) and P_n is power of noise. Fig. 3.4 demonstrates the SNR results, showing that EMR becomes very weak beyond 1.5 meters. In our later evaluation section, we can see that for accurate authentication, devices need to be placed within 0.5 meters away from the source computer, while an attacker located 2m away from the source computer can barely receive the EMR signal for secret key extraction (as further validated in Section 4.6).

Copy Attacker In order to verify the robustness of CompKey against copy attackers, we set up a copy adversarial scenario. Taking computer Acer Aspire V3 as an example, we capture the EMR signals through two USRP antennas, which act as Alice and Bob and are placed near the source computer. We collect EMR signals for 100 seconds and extract one EMR frequency every 0.1s. The frequency variation pattern of Alice and Bob is shown in Fig. 3.5(a). To simulate the copy attacker, we use the same computer, play the same Python program and capture the emission for another 100 seconds serving as copy attacks. Fig. 3.5(b) shows the normalized frequency variation pattern of Alice and Attacker. We can observe that Alice and Bob extract nearly identical EMR frequency, with a correlation coefficient of 0.99. On the other hand, even if the sophisticated copy attacker obtains the source computer and imitates the source computer's RAM activity, it cannot obtain the same frequency variation pattern. From Fig. 3.5(b) we can see that two variation patterns are dramatically different from each other, with only 0.08 correlation coefficient.

To sum up, the above observations confirm that the frequency of a computer's EMR signal is **localized** and **random**, and provide a strong support for exploiting a computer's EMR to generate shared secret keys for nearby devices.

3.4 The Design of CompKey

3.4.1 Extraction of EMR Frequency Signal

In the previous section, we know that legitimate devices close to a source computer can extract the EMR frequency of the source computer. From Fig. 3.2, we can see that the receiving devices will get a spike cluster in frequency spectrum of the captured EMR signal.

The frequency of any one of these spikes will vary in accordance with the memory clock frequency. To validate this, we calculate the correlation coefficient of frequency variation between each two spikes and show that all these spikes have the same variation trend, which means that the frequency of any of these spikes can represent the clock frequency.

We notice that in the spike cluster each two adjacent spikes are separated by around 30kHz, whereas the varying range of the memory clock frequency is comparably small (less than 1kHz). Thus, once we locate one dominant spike in the cluster, the frequency of which is denoted as λ, we can use a bandpass filter, with frequency band $[\lambda - \Delta\lambda, \lambda + \Delta\lambda]$, to preserve this particular spike and get rid of other spikes which all vary in the same manner. For example, we can easily get the frequency variation pattern and track frequency of the highest spike of the filtered signal.

3.4.2 Difference-Based Encoding Method

A straightforward approach is utilizing a quantization approach, which divides the selected frequency value into several levels and encodes the level into binary bits. Even if this method can preserve most of the signal information, it is subject to a large number of consecutive ones or zeros [106], because the EMR frequency only changes marginally over time. Considering this, we choose to convert the frequency difference between two adjacent time steps into binary secret bits. This decision is based on the following two observations. First, the frequency variation is highly random (albeit small) and two participating parties, observing the same varying source, obtain similar patterns, which can be seen from Fig. 3.5(a). Second, the attacker, even if using the same computer and playing the same

Algorithm 2 Difference-based Secret Key Generation

```
**Input:** Memory bus EMR signal $S$, Sampling rate $f_s$, FFT time window size $\Delta t$, Encoding threshold $\sigma$
**Output:** Secret bit list $B$
Divide EMR signal $S$ into $M$ segments each with $\Delta t$ size.
Perform FFT to the first segment of EMR signal.
Get the frequency of the highest spike of the first segment, denoted as $\lambda$.
Initialize a frequency list $f$ with M elements.
**for** $i = 1, 2, \cdots, M$ **do**
    Filter the $i$-th segment with passband $[\lambda - \Delta\lambda, \lambda + \Delta\lambda]$.
    Perform FFT to the filtered signal.
    Get the frequency of the highest spike.
    Store the frequency into the $i$-th element of $f$.
**end for**
**for** $j = 1, 2, \cdots, M$ **do**
    Get difference of $(j + 1)$-th and $j$-th frequency in $f$.
    Compare the difference with $\sigma$.
    Encode the difference according to comparison.
    Append the encoded bits to $B$.
**end for**
**return** $B$
```

program, gets significantly different variation patterns from legitimate ones, as shown in Fig. 3.5(b).

The complete encoding process is elaborated in Algorithm 1. The first step is to get EMR frequency of every time step. In this stage, we first have participating devices collect EMR signals synchronously for a period of time over the frequency band centered at the clock frequency. Then, we divide the EMR signal into non-overlapping segments, each with Δt time window size. After that, we first perform FFT analysis over the first segment, get the frequency of the highest spike, denoted as λ, and then filter the signal using a bandpass filter with a passband, $[\lambda - \Delta\lambda, \lambda + \Delta\lambda]$. In our experiment, $\Delta\lambda$ equals to 50Hz. Finally, we perform FFT on each segment of the filtered signal and get the frequency of the highest spike. After the first stage, we will obtain a frequency list which includes the EMR frequencies of each time step.

The second stage of the algorithm is to encode the frequency difference between two adjacent time steps. Here, we introduce an encoding threshold parameter σ. If the frequency value of current time step is larger than the frequency of previous time step by σ, we regard it as rising and encode it as $'11'$. If the current frequency value is σ less than the previous one, it is treated as dropping and encoded as $'00'$. In the remaining cases, the absolute difference between the present frequency and the previous frequency is less than and equal to σ, and hence it is regarded as unchanged and encoded as $'01'$. Traversing the entire duration, each device will get its own secret bits.

3.4.3 Reconciliation

Reconciliation is a widely-employed method to mitigate and even eliminate minor mismatching bits between two bit sequences via Error Correction Coding (ECC) [82, 148]. We adopt 5 ECC schemes in our paper.

The $C(n,k,r)$ ECC scheme can encode k bits data into a valid n bits codeword by adding $(n-k)$ parity bits, which is a one-to-one encoding function. For a clear representation of reconciliation process, we use $f()$ and $g()$ to represent encoding and decoding functions, respectively. Use k_a and k_b, two n-bit strings, to represent the bit strings obtained by Alice and Bob. First, Alice calculates the corresponding valid codeword closest to her bit string, $f(g(k_a))$. Then, Alice computes an offset, $\delta = k_a \oplus f(g(k_a))$, between her bit string and the codeword. Alice transmits δ to Bob by public medium, which means adversaries can also detect this offset. After receiving the offset, Bob can deduce a bit string by the following equation, which equals k_a with a high probability: $k_a' = \delta \oplus f(g(k_b \oplus \sigma))$. If the mismatching bit between Alice and Bob is no larger than r, k_a' will be equal to k_a. With the reconciliation process, Alice and Bob can ultimately possess an identical secret key with a high likelihood.

3.4.4 Privacy Amplification

Theoretically, there are $(n-k)$ bits of the shared key leaked to the attacker through the offset δ sent over the public medium. In order to get rid of the $(n-k)$ bits leakage, Alice and Bob can use the decoded version of k_a as the final authentication key, $g(k_a)$, which is a k-bit string, instead of directly using k_a. This way, however, sacrifices the bit generation rate, reducing it by a factor of $\frac{n-k}{n}$.

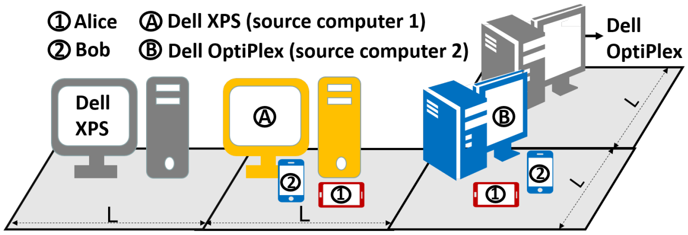

①Alice
②Bob
Ⓐ Dell XPS (source computer 1)
Ⓑ Dell OptiPlex (source computer 2)
Dell OptiPlex
Dell XPS
Ⓐ
Ⓑ
②
①
②
①
L
L
L
L

Figure 3.6: Experiment setup. We take two different kinds of computers — Dell XPS and Dell OptiPlex — as source computers labeled as A and B, respectively. The leftmost computer is another Dell XPS with the same component and configuration as source computer A and acts as an interfering computer. Similarly, the upper right one is the same as source computer B and acts as an interfering computer.

3.5 Experimental Methodology

3.5.1 Experiment Setup

Experiment Location. All the experiments are conducted in our lab office, which is an open space with more than 30 workstations. Each workstation is equipped with an off-the-shelf desktop computer and each two workstations are separated about 1.5 meters away from each other, as illustrated in Fig. 3.6. We focus on Dell XPS and Dell OptiPlex as the source computers, respectively. In each experiment, there exists an interfering computer 1.5m away from the source computer to expose CompKey to an undesired environment. Other computers are not shown in Fig. 3.6 because they are more than 2m away from our source computers and hence have negligible interference.

Experiment Prototype. The experiment prototype of CompKey includes a computer as the radiation source and USRP X310 to collect the radiation signals. The USRP X310 is embedded with UBX 160 daughterboards with a LP0965 Log Periodic PCB antenna, acting as participating devices. The collected signal will be transferred to our HP ProBook 450 computer and processed by CompKey, which is implemented in Python 2.7.

Signal Collecting and Processing. For a specific source computer, the receiving frequency band of USRP will be tuned to 2MHz centered at the source computer's reference memory clock frequency. Each participating device synchronously collects EMR signals and slices the collected signals into N non-overlap segments, each with 0.08s time window size. FFT will be performed over each segment to get the EMR frequency information.

Encoding Threshold. CompKey uses a difference-based encoding algorithm to convert the frequency variation into binary bits based on an encoding threshold σ. We set σ to be 2Hz. Specifically, if the frequency difference between two adjacent times is larger than 2Hz, it will be encoded to '11'. If it is less than 2Hz, it will be encoded to '00'. Otherwise, it is converted to '01'.

Error Correction Coding We consider widely-used ECC schemes, including two linear correcting codes — Hamming Code and Golay Code — and one non-linear correcting code — Reed-Solomon Code (RS). Hamming code can encode every 4 binary bits to 7-bit codeword and correct 1 bit error. Golay code scheme, converting 12-bit string to 23-bit codeword, can fix up to 3 error bits. $RS(n,k)$ can correct up to $\lfloor \frac{n-k}{2} \rfloor$ mismatching bits. In our evaluation, we use three kinds of RS schemes — **RS(7,3)**, **RS(15,5)**, and **RS(15,3)**.

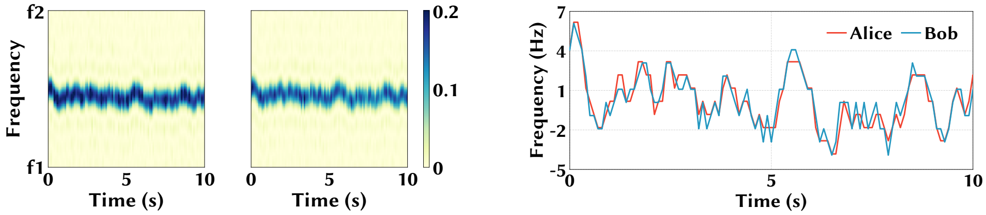
3.5.2 Performance Metrics

Entropy is a measurement of randomness of a random variable. Entropy can reflect randomness of keys from the perspective of uncertainty. It is a good indicator for a signal to be a random key generation source. Given a random variable X with n possible values, $X = [x_0, x_1, ..., x_n]$, its entropy can be obtained by $H(X) = -\sum_{i=0}^{V} Pr[x_i] \log_2 Pr[x_i]$, where $Pr[x_i]$ is the probability of the i-th possibility. By encoding adjacent frequency difference, there are three different frequency variations — up, down and still.

Bit Error Rate (BER) is used to reflect the mismatching level between bits in the same position of two strings. It can be calculated easily by dividing the number of mismatching bits by the total number of bits in the bit string. There are three factors affecting BER in CompKey— FFT time window size, encoding threshold and device distance from the source computer. We will show the impact of these factors using empirical experiments in the next section.

Key Matching Rate (KMR) is also a key metric for secret key generation. It can be calculated by dividing the number of matching keys by the total number of keys. In our experiment, we consider a pair of keys each composed of 60 bits as matching keys if there is no bit discrepancy in any bit position between the two keys.

Bit Generation Rate (BGR) is the number of valid bits generated per second. The higher BGR, the quicker the authentication process finishes and the better the user's experience is. In CompKey, there are three factors determining the BGR. The first one is the FFT time window size Δt, which determines how long it takes to extract the frequency information of each step. The second one is the number of varying possibilities of the EMR signal. More possibilities means more bits generated at a time. Since we decide to encode frequency difference, there are three variations, which at most two bits to represent. The third one is the choice of ECC. In order to get rid of the information leakage, we need to shrink the size of the bit string by a factor k/n. Based on all these, an equation is given to compute BGR of CompKey: $BGR = \frac{n\Delta t}{k} \log_2 V$, where $V = 3$ in our case.



(a)

(b)

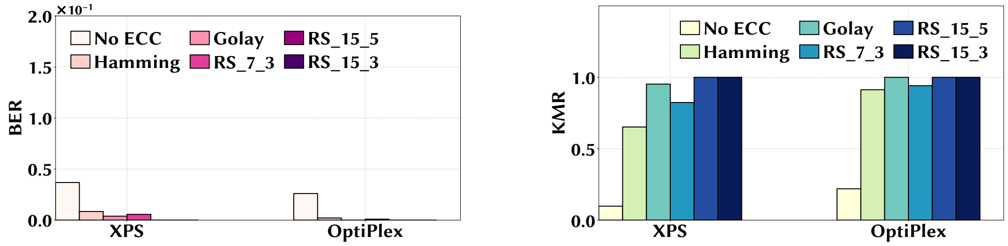
Figure 3.7: (a) Spectrogram of Alice's and Bob's EMR spike in frequency window $[f_1, f_2]$ over 10 seconds. f_2-f_1 is 100Hz. (b) Comparison of frequency variation pattern between Alice and Bob.

3.6 Evaluation Results

3.6.1 Performance of CompKey

We set up two experimental scenarios by using source computers labelled as **A** and **B** in Fig. 3.6, respectively. We will first introduce the first experiment where we use the desktop Dell XPS (computer A in Fig. 3.6) as EMR source. This desktop is with another same Dell XPS 1.5m away on its left hand side and with a Dell OptiPlex 9020 1.5m away on the right hand side, as shown in Fig. 3.6. The interfering Dell XPS is normally used by its owner, who is surfing the Internet. We place Alice 0.5m in front of the source XPS and Bob 0.5m away on the left side of the source computer. Thus, Bob is closer to the interfering XPS and will suffer from more interference. Alice and Bob synchronously collect the EMR signals. In our encoding step, we use 2Hz encoding threshold and take 0.08s as time window size.

By using CompKey, both Alice and Bob extract the frequency of the most prominent spikes from the received EMR signals generated by the source computer. The interfering computer generates weaker EMR than the source computer, and hence its EMR spikes



(a) Bit Error Rate

(b) Key Matching Rate

Figure 3.8: BER and KMR of two experiments with Dell XPS and Dell OptiPlex as source computers, respectively.

will not be picked up by CompKey. We present the spectrogram of Alice's and Bob's EMR signals during 10 seconds, which is shown in Fig. 3.7. To compare their frequency change pattern, we put the normalized frequency together (Fig. 3.7(b)). The frequency changing patterns of Alice and Bob, obtained by CompKey, are highly correlated despite the presence of interfering computers nearby.

As shown in Fig. 3.6, we set up a second testing scenario and take desktop Dell OptiPlex in the middle as the source computer. Alice and Bob are put within 0.5m away from source computer and Bob is closer to the interfering computer.

Fig. 3.8 gives the results of our two experiments, showing that under both testing scenarios CompKey achieves a 100% KMR and demonstrating the practical feasibility of CompKey in the presence of interfering EMR signals.

3.6.2 Randomness of Secret Key

We execute the statistical test suite provided by National Institute of Standards and Technology (NIST) to evaluate the randomness of our generated secret keys [8]. Specifically, if the P-value is more than 1%, the sequence is considered having a high quality of

Table 3.2: Randomness test

Test	**P-value**
Frequency	0.936212
Freq. within Block	0.997614
Binary Matrix Rank	0.176145
Non-overlapping Matching	0.780064
Overlapping Matching	0.633007
Linear Complexity	0.029633
Cumulative Sums (Forward)	0.993956
Cumulative Sums (Reverse)	0.993426

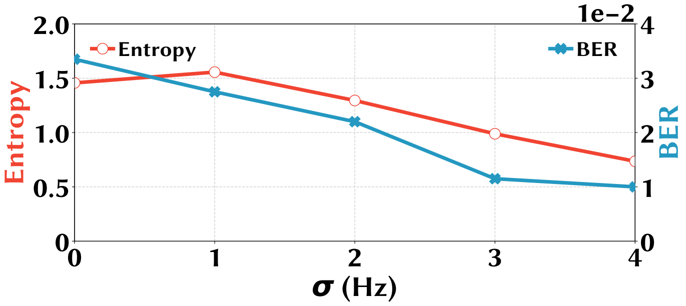


Figure 3.9: Different encoding thresholds result in different entropies and BERs.

randomness and passing this randomness test. Our generated key can pass the statistical tests. The results obtained in our experiment are shown in Table 3.2.

3.6.3 Sensitivity of CompKey

Impact of Parameters

In this section, we will demonstrate the impact of parameters on the performance of CompKey. In our difference based encoding algorithm, there is an important parameter encoding threshold σ and time window size Δt. We see how the encoding threshold σ and time window size Δt affect the performance of our algorithm.

- **Encoding Threshold σ.**

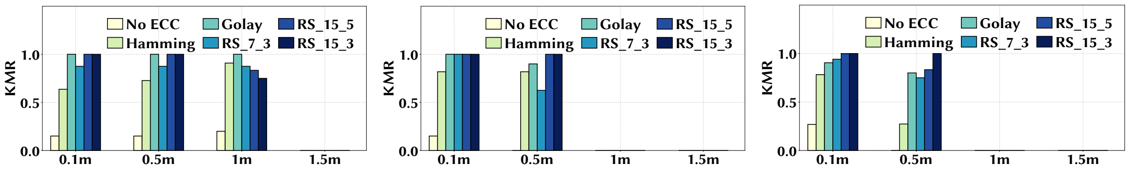
When σ is too small, say zero, CompKey will be very sensitive to local noise, which will easily cause mismatching bits. However, if σ is too large, CompKey is more resistant to environment noise but will convert most of the situations to unchanged/still, which reduces the original entropy. We show in Fig. 3.9 the entropy and BER with respect to different encoding thresholds. Entropy rises a little bit and then decreases. That little increasing entropy at 1Hz may be because when the encoding threshold is 1Hz, some cases with minor changes are converted to unchanged/still, which makes the distribution of three variations more even and hence increases the entropy value. In order to get a small BER and a comparable large entropy, we decide the σ to be 2Hz as a default value.

- **Time Window Size.**

With a smaller Δt, less energy will be collected for each EMR signal segment, which will result in more erroneously estimated spikes and lead to a high BER. Nonetheless, a larger Δt reduces the BGR. We calculate the BER, KMR and BGR with respect to different FFT time window sizes. Our results show that 0.08s is a good FFT window size to maintain a high BGR and a low BER. They are omitted due to space limitation.

RAM Access Frequency of Source Computer

The more frequently RAM is accessed, the more EMR energy. Typically, a computer's RAM is accessed for 20–60% of the time, depending on how many active and background programs are running. To control the RAM access frequency for experiment, we create an array and load it into the RAM. By controlling how frequently we create arrays,



(a) 100%

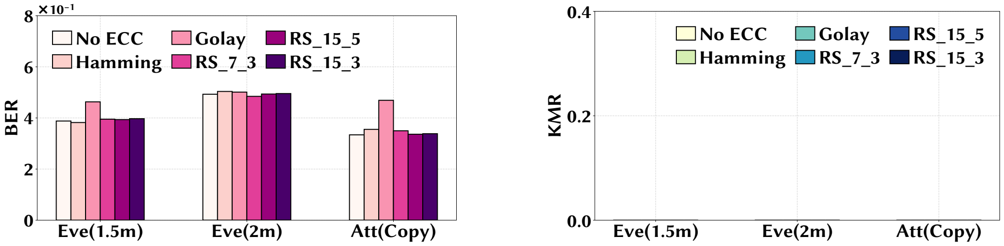
(b) 60%

(c) 20%

Figure 3.10: KMR with different distances when RAM access frequency of the source computer varies from 100% to 20%.

we can manually control the RAM access frequency. For each access frequency of the source computer, we place participating devices at different distances from the source computer and present the KMR.

Fig. 3.10 shows the KMR with different RAM access frequencies. More RAM activities in source computer will make more current flowing through memory bus. Therefore, for 100% RAM access frequency, CompKey can reach 100% KMR when devices are 1m away from the source computer. When the RAM access frequency is 20% or 60%, the two users can extract the same secret key with 100% probability when they are both within 0.5m away from the source computer. We also test the most extreme situation when the RAM is forced to be completely idle with no activities, and no secret keys are successfully generated. In practice, however, computers are rarely completely idle as they run multiple background programs, yielding some RAM activities and hence EMR signals. Thus, CompKey can successfully generate secret keys as long as the legitimate devices are put 0.5m within the source computer, whereas the existing WiFi-based approaches require a distance of a few centimeters between legitimate devices [151].



(a) Bit error rate

(b) Key matching rate

Figure 3.11: Performance of attackers.

Direction

We also evaluate the impact of the direction/angle between legitimate devices and the source computer on CompKey. Our results show that it has little impact on the BER and KMR, and hence are omitted for space limitation.

3.6.4 Security Analysis

For eavesdropping, we consider two Eves, who are capturing the EMR 1.5m and 2m away from the source computer, respectively. We use Dell XPS as the source computer and set its RAM access frequency to be 60% (a fairly strong one to favor attackers). Let the legitimate users and attackers collect the EMR signals at the same time. When Eve is 1.5m away, he can still get some frequency information about the EMR. However, the Eve who is 2m away can get nothing, thus making r_2 =2m a safe threshold distance for attackers in our threat model (shown in Fig. 3.1). The resulting power spectrum density (PSD) of legitimate users and two Eves are omitted due to space limitation. For copy attacks, we use one Dell XPS computer to play a video and two legitimate devices collect EMR from this computer 0.5m away. Meanwhile, we use another identical computer to play the same video and collect the EMR emitted from it at the same time. After collecting the EMR signals, we follow CompKey to generate the secret key and get the performance with different ECCs.

Fig. 3.11 gives the performance of two kinds of attackers. We can see that even if attackers know all the detailed information, they still cannot get the accurate secret key and the KMR is practically zero (even when Eve is 1.5m away from the computer). This demonstrates the security of CompKey against both eavesdropping and copy attackers.

3.7 Related Works

There are numerous studies on proximity-based authentication by extracting shared secret keys from ambient signals. The commonly-used ambient signals are radio-based signal, acoustic signal, and biometrics. Here, we review some of the most related ones and their limitations.

In radio-based authentication studies, received signal strength (RSS) [54,116,127] and channel state information (CSI) [73,82,83,139,140] are two widely-used random signal attributes. However, since only one RSS value can be extracted from one WiFi packet, RSS-based key generation methods are subject to low BGR. Other studies adopt CSI of radio channel for authentication [73,82,140]. However, both RSS-based and CSI-based methods only work for a limited authentication distance because two devices must be placed close to each other to sense the same signal amplitude attribute.

Audio-based authentication approaches make use of the characteristics of acoustic channel [57,113,141]. Nonetheless, it takes time to get the statistics of acoustical attributes, which makes the method subject to low BGR. While key generation based on acoustic channel response (ACR) can help improve BGR, it applies to only two parties to exchange a probe sound. Biometrics is another widely used authentication approach, which utilizes signals from human body for secret key generation [65,130,148]. Since this kind of method needs to capture special signals like body potential signal, it requires the participating devices be equipped with special sensors.

3.8 Conclusion

In this paper, we propose CompKey to secure wireless D2D communications. We observe that the memory bus inside a computer can emit EMR and that only devices in the vicinity of the computer can reliably extract frequency information from the signal. CompKey employs a novel difference-based scheme to encode the frequency variation of computer EMR to a bit string and adopts reconciliation method to alleviate the discrepancy between two bit strings. Through evaluation, we show that devices within 0.5m away from the computer can get identical keys with 10 bits/s BGR and 100% KMR.

Chapter 4

On the Vulnerability of Hyperdimensional Computing-Based Classifiers to Adversarial Attacks

4.1 Introduction

Brain-inspired hyperdimensional computing (HDC) has emerged as an ultra-lightweight classification framework and architecture [30,55,58]. Specifically, HDC exploits the key principle that human brain "computes" based on certain patterns formed by a large number of neurons, without being directly associated with numbers [55]. Instead of computing with numbers like in today's deep neural networks (DNNs), a HDC classifier mimics the way brain cognition works by representing information using a hypervector with binary elements in a very high-dimensional space (e.g., with a dimensionality of $D = 10^4$ or more) [58].

HDC is inherently "in-memory" due to their binarized hypervectors and can be performed using basic logical operations like XOR without the need of sophisticated computation [55]. As a result, HDC classifiers offer several key advantages over conventional DNN-based classifiers, including extremely high energy efficiency, low latency, and strong robustness against hardware-induced component failures [55,58]. For example, recent studies have shown that the energy consumption and inference latency of HDC classifiers are lower by orders of magnitude than their DNN counterparts, yet achieving a reasonable inference accuracy [9,45,48].

HDC classifiers have been increasingly recognized as an alternative to or even replacement of DNNs for classification on edge devices with stringent resource constraints [43, 55, 58]. The quickly expanding list of applications building on HDC classifiers have already included language classification [46], image classification [16, 30], emotion recognition based on physiological Signals [17], distributed fault isolation in power plants [60], gesture recognition for wearable devices [9], and seizure onset detection and identification of ictogenic brain regions [12]. Nonetheless, the security aspect of HDC classifiers remains under-explored. This can raise serious concerns with the safety of HDC classifiers and limit their wider adoption, especially in mission-critical applications such as robot navigation and health monitoring [12,89].

Contribution. In this paper, we make a first-of-its-kind effort to investigate the potential vulnerability of emerging HDC classifiers. More concretely, we consider a threat model in which an attacker can launch grey-box attacks by repeatedly sending perturbed images to the HDC classifier and receiving the Hamming distance output as well as the prediction label from the classifier. We propose a modified genetic algorithm, called Genetic Algorithm with Critical Gene Crossover and Perturbation Adjustment (GA-CGC-PA). GA-CGC-PA only modifies critical genes (i.e., selected important pixels) and iteratively searches for the best candidate adversarial image. GA-CGC-PA also applies perturbation adjustment to further reduce the amount of perturbation noise added to the original benign image. Our evaluation results on handwritten digit classification demonstrate that, for most benign images, the attacker can add a reasonably small amount of perturbation noise and create adversarial images within a limited number of iterations, successfully misleading the target HDC classifier to a wrong prediction label.

4.2 Preliminaries on HDC Classifiers

In HDC, each hypervector is a pseudorandom D-dimensional vector taken by default from $\{-1, 1\}^D$ [30]. Given two hypervectors, Hamming distance (i.e., the number of distinct binary elements) is commonly used as a distance metric to measure their similarity. For the convenience of presentation, Hamming distance is often normalized with respect to the dimensionality D. Thus, two orthogonal hypervectors have a (normalized) Hamming distance of 0.5.

4.2.1 Random Indexing

A HDC classifier projects data onto a hyperdimensional space via random indexing. The almost-certain orthogonality due to the large dimensionality of D demonstrates that any two randomly chosen hypervectors are orthogonal or quasi-orthogonal with an extremely high likelihood [30, 55, 58]. In a hyperdimensional space, there are enormous hypervectors that are orthogonal to each other. Such uncorrelated hypervectors can be used to represent various types of information or features of an object, such as 26 letters in the alphabet set. The hypervectors representing the basic features are called basis hypervectors, which remain unchanged in an application once randomly chosen.

4.2.2 Multiply-Add-Permute Operation

The most widely-used operation in HDC classification framework is Multiply-Add-Permute (MAP).

Binding (Multiplication). Given hypervectors HV_1 and HV_2, binding operation performs element-wise multiplication, denoted as $HV_1 \otimes HV_2$. The operation is used to represent the association of related hypervectors. The resulting hypervector of binding is orthogonal to both of its constituents [55].

Superposition (Addition). Superposition of $HV_1, \cdots, HV_M$ is an element-wise addition of hypervectors denoted as $HV_1 \oplus \cdots \oplus HV_M$. Superposition aims to generate a sum hypervector HV', which can represent a set of operand hypervectors and aggregate information conveyed by them. According to Hebbian Learning, after superposition, any of the constituents is more similar to HV' than a randomly generated hypervector [29, 102].

If the component value of the resultant after addition is positive (i.e., there are more 1s than -1s in superposition), it is converted to 1 and otherwise -1. In the even that the component value of the resultant is zero, it is randomly encoded to 1 or -1 with equal probabilities, which we also refer to as the random majority rule (RMR) [61]. Alternatively, we can always assign 1 or -1 to the component value in such cases (i.e., fix majority rule, or FMR).

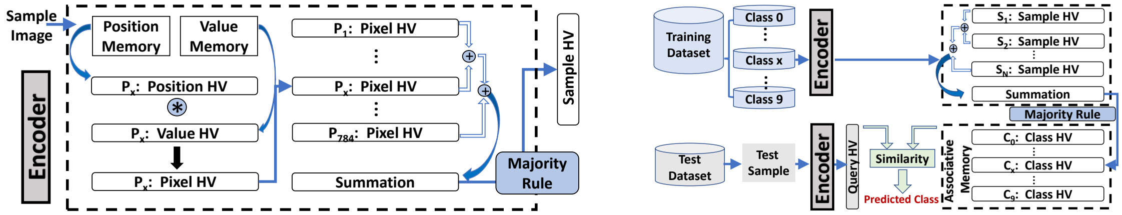
Permutation. The permutation operation generates a dissimilar hypervector by shuffling coordinates of the original hypervector in a pseudo-random manner. A hypervector HV permuted n times is denoted as $\rho^n(HV)$. Permutation is used to store and differentiate the sequence of elements. For example, the letter sequence *abc* can be distinguished from *bac* by permutation.

4.3 A HDC Classifier on MNIST Dataset

As a proof of concept, we construct a HDC classifier on the MNIST dataset [144] for digit recognition, while noting that designing HDC classifiers for more complex tasks is still an active research direction [30].

4.3.1 Mapping

Considering that there are $28 \times 28 = 784$ pixels in an image in MNIST, we employ orthogonal distributed mapping to encode the position information of each pixel. Concretely, we assign a random hypervector to each position (called position hypervector), which automatically ensures that the 784 position hypervectors are distinct and quasi-orthogonal



(a) Encoder

(b) Overview of HDC classifier

Figure 4.1: (a) The encoder in our HDC classifier encodes a digital image (called sample image) to a sample hypervector. (b) The overview of the HDC classifier. An associative memory storing class hypervectors is generated using the training dataset. Then, a test sample can be classified based on its similarity to class hypervectors.

to each other due to the hyperdimensionality. We store these position hypervectors in a look-up table, which is referred to as position memory. Next, we map pixel values to hypervectors, which are called value hypervectors. Clearly, different pixel values are correlated. To preserve similarity of pixel values, we adopt the distance preserving mapping technique and create linearly similar value hypervectors to represent 256 pixel levels, since each pixel value in the MNIST dataset is stored as a 8-bit integer. Typically, the value hypervectors associated with the minimum and maximum pixel values are orthogonal. To do so, we initially pick a random hypervector to represent the minimal pixel value of 0. Then, starting from the initial value hypervector associated with the minimum pixel value, we generate a new value hypervector for the next pixel value by randomly flipping $\frac{D}{2\times255}$ elements of the preceding value hypervector each time. By doing so, we get 256 value hypervectors, including two orthogonal value hypervectors that represent the maximum and minimum pixel values. The 256 value hypervectors are stored in a value memory.

4.3.2 HDC Classifier

Like in conventional classification models [80], a HDC classifier also consists of a training stage and a testing/inference stage, as illustrated in Fig. 4.1(b).

Training

Fig. 4.1(a) illustrates our HDC encoder. Specifically, for each pixel, a pixel hypervector is computed by multiplying the corresponding position hypervector and value hypervector. Next, we add up all the 784 pixel hypervectors and binarize the resulting hypervector using the majority rule, thus generating a sample hypervector that represents the sample image in a hyperdimensional space. To generate a class hypervector, we encode all the sample images in this class into the corresponding sample hypervectors, which are then combined using the superposition/addition operation. Similarly, the majority rule is adopted to guarantee the class hypervector to be binary. Each class hypervector represents the "center" of all sample hypervectors in that class.

Testing/Inference

For testing or inference, using the same encoder as that in the training stage, each new image is first encoded into a query hypervector. Next, we compare the similarity of the query hypervector to each class hypervector in the associative memory in terms of the (normalized) Hamming distance. The HDC classifier will return the label of the class hypervector, which has the minimum Hamming distance to the query hypervector.

4.4 Threat Model

We focus on a grey-box scenario where the attacker can only (repeatedly) send images to the target HDC classifier and obtain the corresponding prediction labels. In addition, for each image, the attacker is also able to receive the Hamming distances between the image's query hypervector and each class hypervector, which thus forms our grey-box model. Our assumption of the attacker's knowing the Hamming distances is the counterpart of knowing softmax probabilities for attacks on standard DNN classifiers.

In the MNIST dataset with $K = 10$ classes, we denote the pixel representation of an input image in a vector form as $X \in \mathbb{R}^{784}$. Then, given the target HDC classifier, we use $\mathbf{f}(X) = [f_1(X), \cdots, f_K(X)] \in [0, 1]^K$ to represent the normalized Hamming distances between the input X's hypervector and the K class hypervectors. The prediction class label t_X is decided as the one with the minimum Hamming distance.

Given a benign image X with its true class label t_0, the attacker would like to create an adversarially perturbed image $\tilde{X} \in \mathbb{R}^{784}$ such that the predicted label $t_{\tilde{X}} = \arg\min_k \{\mathbf{f}(\tilde{X})\}$ for $\tilde{X}$ differs from the true label t_0. Formally, we can define the objective function as

$$g(\tilde{X}, t_0) = \max\{\min_{k \neq t_0} [\mathbf{f}(\tilde{X})] - f_{t_0}(\tilde{X}), -\epsilon\}, \tag{4.1}$$

where $\min_{k \neq t_0} [\mathbf{f}(\tilde{X})]$ is the minimum Hamming distance of the perturbed image to any of the class hypervectors with wrong labels, $f_{t_0}(\tilde{X})$ is the Hamming distance of the perturbed image to the true class hypervector, and a small constant $\epsilon > 0$ indicates that the attacker does not need to add further perturbation if its attack is already successful (i.e., $\min_{k \neq t_0}[\mathbf{f}(\tilde{X})] - f_{t_0}(\tilde{X})$ is already less than $-\epsilon$). Thus, by minimizing $g(\tilde{X}, t_0)$, the attacker can effectively increase the Hamming distance of the perturbed image to the true class hypervector, misleading the HDC classifier to a wrong prediction label.

Meanwhile, the attacker also needs to keep its perturbation to the original image X as minimum as possible. Concretely, the attacker obtains $\tilde{X}$ by minimizing the following regularized objective function:

$$\min_{\tilde{X}} \left\{ g(\tilde{X}, t_0) + c \cdot \|\tilde{X} - X\| \right\}, \tag{4.2}$$

where $\|\tilde{X} - X\|$ is a certain norm that quantifies the difference between $\tilde{X}$ and X, and $c \geq 0$ adjusts the weight for regularization. We can also add multiple norms for regularization. For example, L_2 norm controls the squared difference between two images' pixel values, while L_∞ controls the maximum difference between two images' pixel values.

4.5 A Modified Genetic Algorithm

We first describe a basic genetic algorithm and then propose modifications so as to reduce the amount of perturbation introduced to the original benign input.

4.5.1 Genetic Algorithm

The optimization problem in Eqn. (4.2) involves non-convex integer programming, and $\mathbf{f}(\cdot)$ is non-differentiable and unknown to the attacker. Here, to solve Eqn. (4.2), we propose a modified genetic algorithm, called Genetic Algorithm with Critical Gene Crossover and Perturbation Adjustment (GA-CGC-PA). Concretely, GA-CGC-PA described in Algorithm 3 takes an original input image as an ancestor, from which the first generation of population is generated by natural mutation. A basic genetic algorithm includes four main steps — population initialization, member selection, crossover, and mutation — as described in detail below.

Population Initialization

The first generation is initialized by applying uniformly distributed random noise in the allowed range $(-\sigma_{max}, \sigma_{max})$ to each gene of the ancestor X. For the MNIST dataset, each gene corresponds to one pixel. In total, there are $28 \times 28 = 784$ genes in each individual member, and the algorithm creates N members in each generation.

Member Selection

The quality of each population member is evaluated by computing a fitness score according to the fitness function (additive inverse of Eqn. 4.2). Population members with higher fitness scores are more likely to be selected to reproduce the next generation, whereas members with lower fitness scores are replaced with a higher probability. Towards this end, we compute the softmax of the fitness scores in one generation to obtain the selection probability distribution of the population. We then randomly choose pairs of parents to breed offsprings according to the softmax probability distribution. In order to save the member with the highest fitness score (called elite member) in one generation, an elitism technique [10] is employed, where the genes of the elite member are exactly cloned by a member in the next generation.

Crossover

Our algorithm makes use of uniform crossover to mate two parents. Each gene of an offspring is produced by combining genes of both parents, $Parent_1$ and $Parent_2$, according to the probability distribution $(p, 1-p)$. We get p through dividing the fitness of the first parent P_1 by the sum fitness of both parents. Thus, the child's genes are given as follows:

$$child = p \times Parent_1 + (1-p) \times Parent_2. \tag{4.3}$$

Nonetheless, since it is required that the perturbation made to the original image be kept as minimum as possible, we reduce the number of perturbed genes (pixels) by using a modified version of uniform crossover, which we call critical gene crossover as described in Section 4.5.2.

Mutation

In order to promote diversity within a generation and improve the search power of the genetic algorithm, the child generated by crossover has to be mutated and clipped before becoming a member of the next generation. Like population initialization, random noise is sampled uniformly from a range $(-\sigma_{max}, \sigma_{max})$ and added to the chromosome of the child with a mutation probability ρ. Considering that a feasible solution has to possess a reasonable gene (e.g. pixel value for MNIST dataset), a mutated child is clipped to ensure that its genes are all within an allowable range.

4.5.2 Modification for Perturbation Reduction

While the basic genetic algorithm can generate an adversarial image to fool the HDC classifier, the amount of perturbation can be really significant (see Fig. 4.2(b) for an example), making the adversarial input more easily identified by human perception. Here, we propose to use *critical gene crossover* and *perturbation adjustment* to significantly reduce the amount of perturbation.

Critical Gene Crossover

The standard uniform crossover modifies each pixel of the original image, which unnecessarily introduces redundant perturbation. To reduce perturbation, we propose critical gene crossover to selectively cross the parents' most important genes. To do so, we first make a child by duplicating the parent with the higher fitness score and then select critical genes using the max pooling operation. Next, we renew the critical genes by uniformly crossing those of the two parents. The detailed steps are described in Algorithm 4. We define critical genes as the ones that mostly differentiate images of different classes. For the example of the MNIST dataset, pixels that are close to and form the digit are more important than others that have lower pixel values and mostly form the background, and hence can be chosen as critical genes.

Perturbation Adjustment

Considering the fact that the genetic algorithm generates random mutation in each generation and thus can introduce unnecessary modification to the original image, we propose to further reduce the perturbation by using perturbation adjustment while still keeping the adversarial attack successful. Our perturbation adjustment technique is described in Algorithm 5. It starts by finding an index list $\mathcal{L}$ of modified pixels in the adversarial image compared to the original image. For each pixel in the list $\mathcal{L}$, its value is restored to the original value v_{ori}. Then, we gradually change the value towards the adversarial value v_{adv} and stop this process until the adversarial image can successfully mislead the HDC classifier to a wrong prediction.

4.5.3 Effect of Perturbation Reduction

We present an example of adversarial attacks on the digit "6" using three different algorithms in Fig. 4.2: standard genetic algorithm without modification (GA), modified genetic algorithm with only critical gene crossover (GA-CGC), and modified genetic algorithm with both critical gene crossover and perturbation adjustment (GA-CGC-PA). The HDC classifier is trained on the MNIST dataset as described in Section 4.6.1. In all the three attacks, the HDC classifier misclassifies the digit "6" as "2". Fig. 4.2(a) shows the original benign image for digit "6" which can be correctly classified by the HDC classifier, while Fig. 4.2(b) shows the adversarial image using GA. We can clearly see that many pixels in the original image are modified and added with perturbation noise, making the adversarial image easily identifiable. Fig. 4.2(c) shows the adversarial image generated by GA-CGC after using critical gene crossover. Compared with the result in Fig. 4.2(b), many background pixels in Fig. 4.2(c) are left unchanged and only pixels surrounding the digit are altered. By using GA-CGC-PA with further perturbation adjustment, the adversarial image is shown in Fig. 4.2(d), which looks very similar to the original benign image but is still misclassified

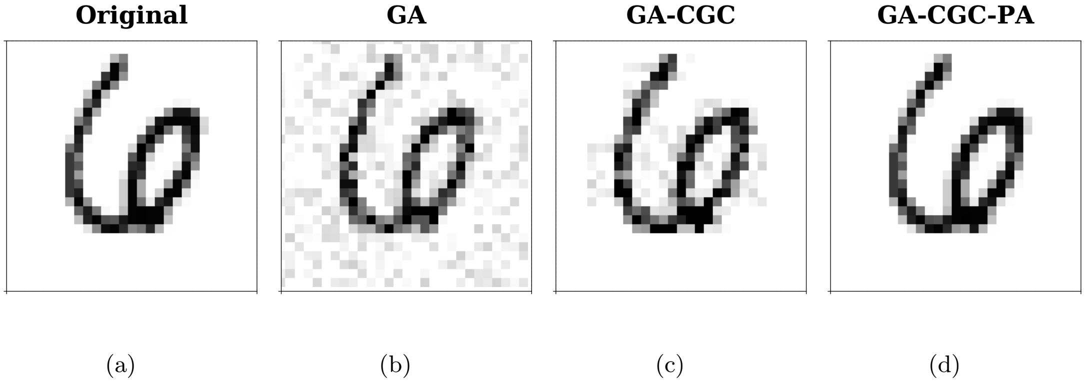


Figure 4.2: Comparison of different adversarial attacks that mislead the HDC classifier to classify "6" as "2". (a) Original benign image. (b) Adversarial image by basic genetic algorithm (GA). (c) Adversarial image by genetic algorithm with critical gene crossover (GA-CGC). (d) Adversarial image by our proposed genetic algorithm with critical gene crossover and perturbation adjustment (GA-CGC-PA).

by the HDC classifier as "2". The number of pixels modified is largely reduced from 438 (by GA)to 9 (by GA-CGC-PA). This shows the clear advantage of GA-CGC-PA over the basic genetic algorithm and only using critical gene crossover, in terms of reducing the amount of perturbation in adversarial images.

4.6 Evaluation Results

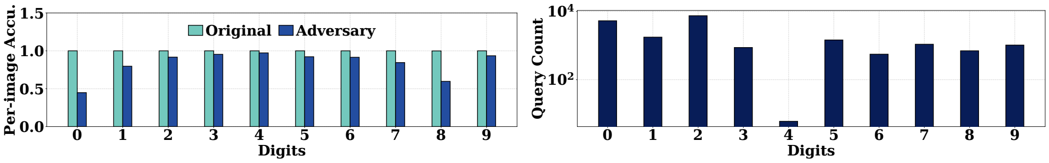
This section validates the effectiveness of our proposed GA-CGC-PA for adversarial attacks on a target HDC classifier using handwritten digit recognition for proof of concept.

4.6.1 HDC Classifier Training

We train a HDC classifier based on the MNIST training dataset [144] as an example. The dimensionality for each hypervector is $D = 10^4$. Then, as described in Section 4.3, we encode each training sample into a sample hypervector and obtain 10 class hypervectors based on the training dataset. Next, we project each test image into a query hypervector and compare it against class hypervectors. Recalling that in the hypervector encoding process, we use the majority rule for vector binarization. By using the random majority rule (RMR) that randomly assigns 1 or -1 in the rare event that the sum is zero after superposition operation, the HDC classifier may assign different labels in different inferences for the same input. To eliminate this uncertainty, we can also apply the fixed majority rule (FMR) that always assigns 1 or -1.

For the HDC classifier with RMR, we execute 1,000 rounds of classification for each test image to calculate the average accuracy, which we also refer to as per-image accuracy. Our HDC classifier can assign correct labels with 100% per-image accuracy for around 70% of the test images, while it behaves less confidently and somtimes yields misclassified results for the remaining images. Consequently, the test images that have 100% per-image accuracy are harder to attack (called hard cases) than those with a lower per-image accuracy (called vulnerable case). In other words, vulnerable images can be considered already "adversarial" to our HDC classifier to some extent. The overall accuracy of our HDC classifier is lower than that of DNNs [43], and can be improved by enlarging the MNIST dataset, which is beyond the scope of our work. We will show later, GA-CGC-PA can successfully mislead the HDC classifier with a high probability regardless of hard or vulnerable cases.

While the MNIST dataset is admittedly simple, we view it as an important proof of concept and starting point to study the vulnerability of emerging HDC classifiers. Importantly, our attack strategy based on genetic algorithms in Section 4.5 is general and applies to any HDC classifiers without being restricted to the MNIST example.



(a)

(b)

Figure 4.3: (a) Per-image accuracies of benign/adversarial images shown in Fig. 4.7. (b) Query counts needed to generate adversarial images shown in Fig. 4.7.

4.6.2 Attack on HDC Classifier With RMR

We first evaluate GA-CGC-PA with the random majority rule (RMR) for the HDC classifier. We use a population size $N = 6$, mutation probability $\rho = 0.05$, max pooling size $2{\times}2$, and critical threshold $\beta = 0$.

We focus on attacking the hard cases (i.e., those images with 100% per-image accuracy), while noting that the already-vulnerable images (i.e., those with less than 100% per-image accuracy) are even easier to attack. Fig. 4.7 in the appendix visually illustrates the benign input images, adversarial perturbation noise, and the corresponding adversarial images. The adversarial images can significantly decrease the HDC classifier's performance, while they are still clearly recognizable by human eyes. Next, we show the corresponding per-image accuracies of both original images and adversarial ones in Fig. 4.3(a). It can be clearly seen that, with GA-CGC-PA, all the images become vulnerable with a per-accuracy lower than 100%. In particular, the sample images for digits "0" and "8" in Fig. 4.7 have the lowest accuracy after attacks and hence are easier to attack than others.

Table 4.1: Perturbation for Images Shown in Fig. 4.7

Digit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
# Modified Pixels	289	160	260	196	185	241	177	192	256	253
L_2**-distance**	4.626	5.073	3.02	3.703	1.37	3.028	2.58	4.017	2.029	1.537
L_∞**-distance**	0.867	0.968	0.643	0.956	0.276	0.653	0.737	0.92	0.401	0.271

Amount of Perturbation

Next, we quantify the adversarial perturbation noise generated and added to the benign images. To have a successful attack, the adversarial images need to not only deceive the HDC classifier but also have as small perturbation as possible compared to benign ones. To this end, the amount of perturbation is an important metric to evaluate the attack algorithm. As in the prior studies on adversarial machine learning [110], we use L_0 norm, L_2 norm, and L_∞ norms to measure the amount of perturbation. Note that while L_0 is not a mathematical norm, it is commonly used to quantify the total number of modified pixels in our context. By definition, L_2 norm indicates the overall perturbation noise added to a benign image, while L_∞ norm measures the maximum per-pixel perturbation noise.

Table 4.1 shows the three norm distances for the perturbation noise added to the benign images shown in Fig. 4.7. It is worth noting that L_2 and L_∞ norms are calculated over the images with normalized pixel values in the range of $[0, 1]$. For all the adversarial images shown in Fig. 4.7, there are fewer than 300 modified pixels. While the L_∞ is large, the L_2 norm is reasonably small for most digits, indicating the overall perturbation added by GA-CGC-PA is not large, which can also be observed from Fig. 4.7.

Query Count

We plot in Fig. 4.3(b) the number of queries used to generate the adversarial images. The result shows that the average query count is up to the order of thousands. In particular, the query count for digit “2” is more than 7k, whereas the digit “4” needs the least number of queries to attack. While the existing adversarial attacks in the literature focus on DNN-based classifiers and different datasets, we note that they typically need an order of 10k or more queries to successfully attack an image [3,74].

4.6.3 Attack on HDC Classifier With FMR

We now turn to the fixed majority rule (FMR) such that the prediction label for a given image is fixed without uncertainties. The hyperparameters for GA-CGC-PA are the same as in Section 4.6.2.

Attack Success Rate

With FMR, the per-image accuracy is either 0 or 1. Thus, we randomly pick 200 correctly classified images for each digit from “0” to “9” from the MNIST dataset. For each image, we apply GA-CGC-PA to generate the corresponding adversarial image subject to a maximum query count of 10^5 (i.e., $I_{\max} = 10^5$ in Algorithm 3). If an adversarial image is successfully generated to fool the HDC classifier within the query limit, it is regarded as a successful attack, and a failed attack otherwise.

We compute the ASR over 200 images for each digit and present the results in Fig. 4.4. It can be seen that GA-CGC-PA is successful for all the digits in most cases, with

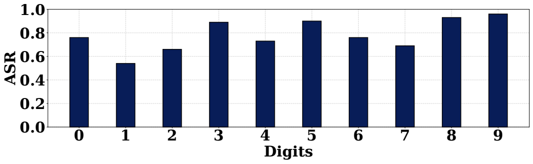
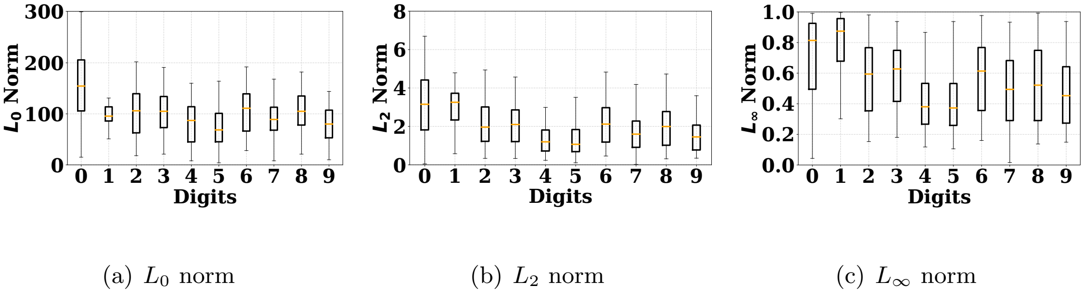


Figure 4.4: ASR of digits 0-9 for HDC classifier with FMR.



(a) L_0 norm

(b) L_2 norm

(c) L_∞ norm

Figure 4.5: Box plot of perturbation noise added by GA-CGC-PA for the HDC classifier with FMR. Each box plot shows the values for the maximum/minimum/median/75th percentile/25th percentile, excluding outliers.

digits “3”, “5”, “8” and “9” having the highest ASR. Considering the 10 digits altogether, we obtain an average ASR of 0.78.

Amount of Perturbation

We provide the bar plot of perturbation amount in terms of L_0, L_2 and L_∞ norms in Fig. 4.5. As one can see from the figure, the median number of modified pixels for most adversarial images is around 100. The L_2 norm for the majority of perturbation noise is between 2 and 4, whereas the L_∞ norm lies mostly between 0.3 and 0.8 for most images.

Additional results, i.e., query count and adversarial examples, are deferred to the appendix.

4.7 Related Works

Adversarial attacks on DNNs can be categorized into white-box attacks, black-box attacks, and grey-box attacks [110]. In a white-box attack, an attacker is assumed to know complete details about the target DNNs [1, 15]. By contrast, in a black-box attack, only benign inputs and the corresponding prediction label (plus softmax probabilities in a grey-box setting) are available to the attacker [74, 98]. More recent studies on black-box or grey-box attacks have proposed to use gradient estimations to generate adversarial samples [18, 125]. Nonetheless, these approaches are generally limited to differentiable objective functions, which is not the case in HDC classifiers that use MAP operation in a hyperdimensional space without differentiable objective functions. Boundary attack is a gradient-free black-box attack, which uses an already-available adversarial sample as a reference [11, 95]. Nonetheless, an adversarial sample is needed at the first place. Genetic algorithm is another effective approach to attacks on DNNs [3, 74, 142]. We leverage a genetic algorithm, but also modify it to reduce perturbation (see Fig. 4.2). Most importantly, we propose a new Hamming distance-based objective function that is tailored to the emerging HDC classifiers.

The existing studies on HDC classifiers have been predominantly focused on improving the energy efficiency, inference latency, privacy preservation, or architecture design [44, 47, 49, 50, 59, 111]. Nonetheless, adversarial attacks on HDC classifiers have been neglected, raising serious concerns with their safety as they are being adopted in increasing more applications including mission-critical scenarios. Our study bridges the gap and demonstrates that, like their DNN counterparts, HDC classifiers can be vulnerable to adversarial inputs and hence need to be better safeguarded.

4.8 Conclusion

In this paper, we study adversarial attacks on HDC classifiers which are emerging for edge inference. We propose a modified genetic algorithm (GA-CGC-PA) to generate adversarial images within a reasonably small number of queries. Our results show that slightly-perturbed adversarial images generated by GA-CGC-PA can successfully mislead the HDC classifier to wrong prediction labels with a large probability. Future research includes more sophisticated attacks on HDC classifiers and, most importantly, effective defense mechanisms.

4.9 Appendix: Additional Results

Query Count for Attacks on HDC Classifier With FMR

We calculate the query counts for the successfully attacked images and show the results in a box plot in Fig. 4.6. We can notice that the median query count of all digits is less than 5,000, which is a reasonably good query efficiency for black-/grey-box attacks [3].

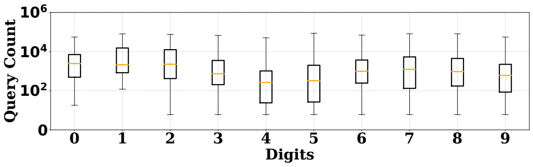


Figure 4.6: Box plot of query count needed by GA-CGC-PA for the HDC classifier with FMR. Each box plot shows the values for the maximum/minimum/median/75th percentile/25th percentile, excluding outliers.

Table 4.2: Perturbation for Images Shown in Fig. 4.8 and Fig. 4.9. The values for Fig. 4.9 are shown in parentheses.

Digit	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
L_0	301(22)	123(82)	95(21)	104(107)	72(42)	117(34)	113(9)	74(86)	114(68)	97(45)
L_2	6.73(0.36)	4.48(0.95)	2.20(0.45)	0.82(0.65)	1.54(0.44)	0.69(0.66)	0.76(0.18)	1.90(1.12)	0.64(0.73)	0.72(0.49)
L_∞	0.92(0.21)	0.96(0.24)	0.69(0.21)	0.21(0.15)	0.53(0.13)	0.19(0.21)	0.23(0.16)	0.56(0.37)	0.18(0.25)	0.20(0.19)

Adversarial Examples

Finally, we visually show some adversarial examples for the HDC classifier with FMR. In the hard case, benign images would have a 100% per-image accuracy had the HDC classifier use RMR. In the vulnerable case, benign images are correctly classified by the HDC classifier with FMR, but would have less than 100% per-image accuracy had the classifier use RMR. That is, the vulnerable images are those borderline images that are already hard to be correctly classified by the HDC classifier.

The benign images, perturbation noise, and adversarial images for hard and vulnerable cases are shown in Fig. 4.8 and Fig. 4.9, respectively. Also, we give the amount of perturbation noises for the two cases in Table 4.2.

It is more difficult to launch successful attacks in the hard case than in the vulnerable case. Thus, as expected, the perturbation noise added by GA-CGC-PA in the hard case is generally less than in the vulnerable case. In particular, in the vulnerable case, the adversarial image is almost identical to the corresponding benign image by human perception. This can also be reflected from the perturbation noise figures and Table 4.2.

Algorithm 3 Modified Genetic Algorithm (GA-CGC-PA)

```
Input: Original input $X$, true label $t_0$, population size $N$, maximum iteration $I_{\max}$
Output: adversarial sample $\tilde{X}$
Create the initial generation $P^0$ from $X$.
$G_{curr} \leftarrow P^0$
for $ite = 1$ to $I_{\max}$ do
    Compute fitness score of each member in $G_{curr}$
    Find elite $Eli = \arg\max_{x \in G_{curr}} fitness(x)$
    Save $Eli$ as a member of next generation $G_{next}$
    if $\arg\min_k(\mathbf{f}(Eli)) \neq t_0$ then
        $\tilde{X} \leftarrow Eli$
        return $\tilde{X}$
        break
    endif
    Compute selection probability $P_{sel}$ of $G_{curr}$
    for $num$=2 to N do
        Choose parents in $G_{curr}$ according to $P_{sel}$
        Apply Critical Gene Crossover (Algorithm 2)
        Apply clipping and add clipped child to $G_{next}$
    endfor
    $G_{curr} \leftarrow G_{next}$
endfor
Apply Perturbation Adjustment Algorithm
```

Algorithm 4 Critical Gene Crossover

```
Input: $Parent_1$ and $Parent_2$, crossover probability $(p, 1-p)$ of $Parent_1$ and $Parent_2$ with $p > 1-p$, maximum $L_\infty$ mutation distance $\sigma_{max}$, mutation probability $\rho$, critical threshold $\beta$
Output: $child$
$child \leftarrow Parent_1$.
Apply $2 \times 2$ max pooling: $child' = maxpooling(child)$
Up-sample $child'$ to the original dimension $28 \times 28 = 784$
Normalize values of $child'$: $child' = \frac{child' - min(child')}{max(child')}$
Find indexes of critical genes such that $child'[idx] > \beta$
Update critical genes of $child$
    $child[idx] = p \times Parent_1[idx] + (1-p) \times Parent_2[idx]$
Mutate $child$
    $child[idx] = child[idx] + B(1, \rho) \times \mu(-\sigma_{max}, \sigma_{max})$
return $child$
```

Algorithm 5 Perturbation Adjustment

```
Input: Original image $X$, true label $t_0$, adversarial image $\tilde{X}$
Find an index list $\mathcal{L}$ for pixels that differ in $X$ and $\tilde{X}$
for $p$ in $\mathcal{L}$ do
    $v_{ori} \leftarrow X[p]$
    $v_{adv} \leftarrow \tilde{X}[p]$
    for $v = v_{ori}$ to $v_{adv}$ do
        $\tilde{X}[p] = v$
        if $\arg\min_k(\mathbf{f}(\tilde{X})) \neq t_0$ then
            break
        endif
    endfor
endfor
```

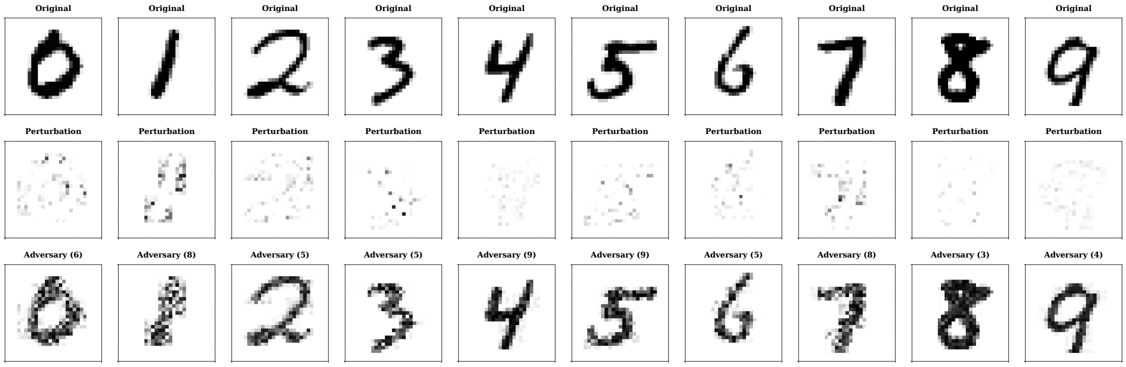


Figure 4.7: Attacks on the HDC classifier with RMR. The first row shows the original images. The second row shows the perturbation noise added by the attacker. The third row shows the adversarial images, and the corresponding misclassified labels are given at the top of each image.

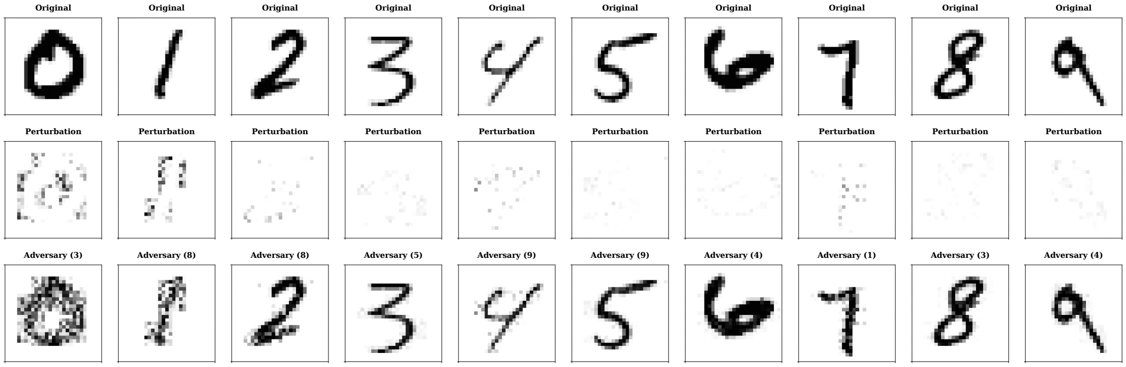


Figure 4.8: Attacks on the HDC classifier with FMR (hard). The first row shows the original images. The second row shows the perturbation noise added by the attacker. The third row shows the adversarial images, and the corresponding misclassified labels are given at the top of each image.

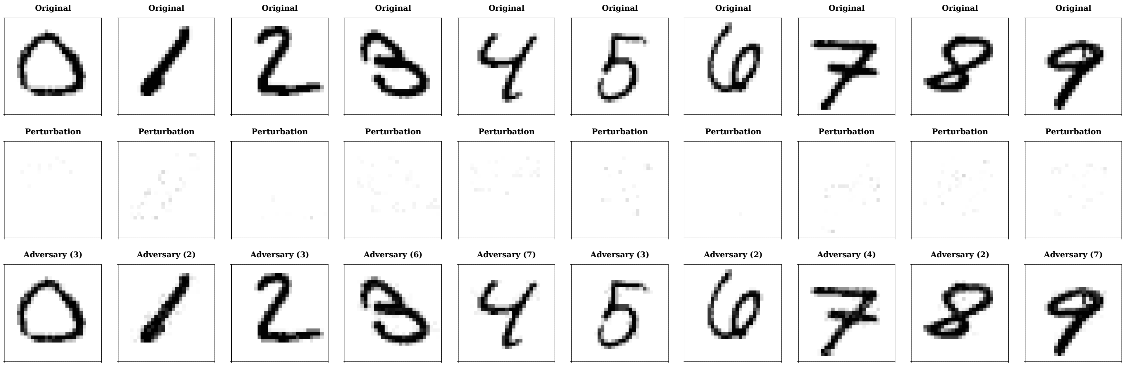


Figure 4.9: Attacks on the HDC classifier with FMR (vulnerable). The first row shows the original images. The second row shows the perturbation noise added by the attacker. The third row shows the adversarial images, and the corresponding misclassified labels are given at the top of each image.

Chapter 5

Achieving Certified Robustness for Brain-Inspired Low-Dimensional Computing Classifiers

5.1 Introduction

Brain-inspired hyperdimensional computing (HDC) classifiers have been emerging as light-weight machine learning alternatives to deep learning models [30, 55, 61, 90, 109]. Especially, a low-dimensional computing (LDC) classification framework has been recently proposed, which compared to traditional HDC-based classification models improves the inference accuracy and meanwhile dramatically reduces the model size, inference latency and energy consumption by orders of magnitude. In addition, the LDC model has represented excellent performance in applications like in computer vision and voice recognition [26].

Nonetheless, recent studies have shown that HDC-based classifiers are vulnerable to carefully crafted adversarial attacks in both white-box and black-box setting [19, 81, 123, 146]. In such attacks, the adversarial perturbations introduced to the original input are visually indistinguishable but could make the output label deviate from the ground truth. There has been significant interest in literature in constructing defence to protect classification models against adversarial attacks, like obfuscating gradients, defensive distillation and retraining technique [6, 37, 81, 99, 123, 126]. Unfortunately, many of these defenses are targeted to specific adversarial attacks. For example, obfuscating gradient technique takes advantage of gradient masking method and provides apparent robustness against white-box iterative optimization attacks [6, 18, 107, 143]. So, these defense techniques were broken soon by another new attacking scheme, which drives the advent of certified defense technique [14, 63].

Certified defenses provide guarantee of robustness against norm-bounded attacks. Methods proposed in work [21,108,137] alter the network configurations such as the network structure and activation function, which make them struggle to generalize across different types of networks. Provable robustness technique via random smoothing requires taking the mean of the output vectors, which is susceptible to the outliers and lead to ambiguous outputs [23, 66, 67, 112, 122]. Paper [63] leverages differential privacy and provides a scheme which requires extra model structure like the separate auto-encoder. Interval bound propagation (IBP) technique bypasses the challenges of these methods. It is comparable to two forward passes through the network, without changing the original network and inducing extra structure [34, 42, 91, 132, 133].

In this paper, we make the first attempt to study the provable robustness of LDC model with IBP for image classification problem. To obtain a certifiably robust LDC model against L_∞ perturbation, the minimum difference between logits of the true label and any other class, called minimum margin, has to be larger than zero for any input perturbation within L_∞ norm-bound ball. To this end, IBP, which is first proposed in [34], is adopted to calculate the lower bound of the minimum margin. An appropriate loss function is defined to guarantee a non-negative value of the lower bound and thus a correct labelling over L_∞ norm-bounded perturbed inputs. For evaluation, we train LDC models across a wide range of L_∞ perturbation radii, referred to as training perturbation radius, based on both MNIST and fashion MNIST dataset. We also employ the elision technique to make the lower bound of the minimum margin tighter and compare the performance of the trained models in terms of nominal accuracy and verified accuracy. Besides, we implement a powerful white box attacking method, project gradient descent (PGD), to each of the trained models and demonstrate a drastic reduction in attack success rate from 100% to below 0.1% with IBP robust training. The trained models also exhibit high performance with memory errors existing.

5.2 Preliminaries

5.2.1 LDC Classifier

In a nutshell, LDC classifier maps the encoding and inference process of HDC classifier into an equivalent neural network that includes a non-binary neural network for value representation followed by a binary neural network layer for sample encoding and another binary layer for inference. After training, it can extract optimized low-dimensional binary vectors to represent features and values for efficient inference.

We focus on the certified robustness of a LDC model for classification tasks. The LDC model can be formulated as a function f_θ: $x \rightarrow \mathbb{R}^C$, where the input data is in a normalized N-dimensional subspace $x \subseteq [0,1]^N$. The model provides confidence scores $f_\theta(x) \subseteq [0,1]^C$ for all C classes. $F_\theta(x) = argmax_{i\in[C]} f_\theta(x)_i$ is the predicted class label of model f_θ given input x. θ is the set of trainable parameters of the model, which is trained to minimize the cross-entropy loss.

Specifically, LDC classifier f_θ is a 3-layer neural network, including value layer, feature layer and class layer respectively, as shown in Fig. 5.1. It can be mathematically represented as follows:

$$\begin{cases} z_0 = x_0 \\ z_1 = Concat(W_0 z_0^i + b_0) \\ z_1 = Bin(Tanh(z_1)) \\ z_2 = Bin(W_1^b z_1) \\ z_3 = W_2^b z_2 \end{cases} \tag{5.1}$$

where $x_0 \subseteq [0,1]^N$ is the input. z_0^i is the ith dimension of z_0 for $i = 1$ to N. $Bin(z) = sign(z)$ and $Concat(z)$ is concatenating operation which joins the weighted sum of each item in the input vector into a single output vector. The trainable parameters $\theta = \{W_0, b_0, W_1^b, W_2^b\}$. The shape of W_0, W_1^b, W_2^b is $(D_v, D_p), (D_f, N \times D_v), (D_c, D_f)$ re-

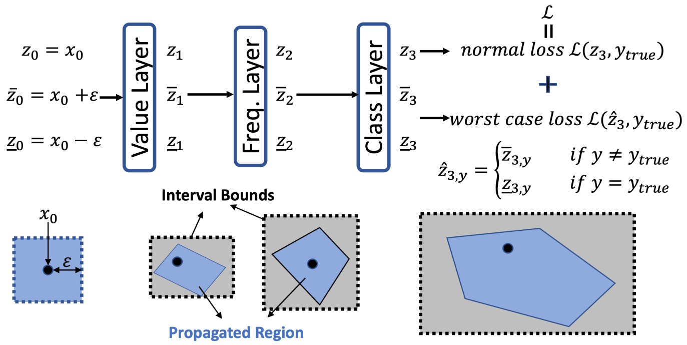

$\mathcal{L}$
$\parallel$
$z_0 = x_0$
$\bar{z}_0 = x_0 + \varepsilon$
$\underline{z}_0 = x_0 - \varepsilon$
Value Layer
z_1
$\bar{z}_1$
$\underline{z}_1$
Freq. Layer
z_2
$\bar{z}_2$
$\underline{z}_2$
Class Layer
z_3
$\bar{z}_3$
$\underline{z}_3$
normal loss $\mathcal{L}(z_3, y_{true})$
+
worst case loss $\mathcal{L}(\hat{z}_3, y_{true})$
$\hat{z}_{3,y} = \begin{cases} \bar{z}_{3,y} & if\ y \neq y_{true} \\ \underline{z}_{3,y} & if\ y = y_{true} \end{cases}$
Interval Bounds
x_0
ε
Propagated Region

Figure 5.1: Illustration of LDC model with IBP method. The L_∞ norm-bounded perturbation with radius ϵ (in blue) is propagated through layers of LDC model. The interval bound (in gray), represented as $|\bar{z}_k, \underline{z}_k|$, is propagated simultaneously through layers, which always encompasses the blue region.

spectively. D_p represents the dimension of each input feature value and D_c is the dimension of final output vector. Take MNIST classification as an example, $D_p = 1$ since each pixel value could be represented as a single scalar. $D_c = 10$ because there are 10 classes in total. D_v and D_f are the hyperparameters of the model representing dimension of value vector and dimension of feature vector in HDC context. Thus, in LDC model there are only affine transformations, $Wz + b$, and monotonic activation functions, $Concat(z)$, $Bin(z)$ and $Tanh(z)$.

5.2.2 Adversarial Attacks

Adversarial attacks can be categorized into two settings, targeted attack and untargeted attack. The goal of targeted attack is to mislead the model to classify the adversarial example to an intended target class, y_{tg}, instead of the true class, y_{true}. On the other hand, untargeted attacker would like to make the model misclassify the perturbed image as any class, y', other than the original true class, y_{true}. The following is the definition of untargeted attack. For given input (x_0, y_{true}), the attacker would like to generate a perturbed input $A_{p,\epsilon}(x_0) = \{x : \|x - x_0\|_p < \epsilon\}$ such that $F_\theta(x) \neq y_{true}$. We use $A_{p,\epsilon}(x_0)$ to denote the perturbed input which is sampled from the region centered at x_0 with ϵ radius, where ϵ represents the perturbation magnitude measured by L_p norm ($p \in \mathbb{N}_+ \cup \{+\infty\}$). Common L_p are L_1, L_2 and L_∞.

5.2.3 Robustness Verification

To certify the robustness of a classifier against norm-bound perturbation, $A_{p,\epsilon}(x_0)$, we need to verify that for any possible perturbed input $x \in A_{p,\epsilon}(x_0)$ the predicted class is always the true label y_{true}. To achieving this purpose, we define a minimum margin, $M(y_{true}, y')$, as the minimum prediction logit difference between the true class label y_{true} and any other class y', when the input x is within the L_p norm-bounded ball by ϵ. $\forall x \in A_{p,\epsilon}(x_0)$ and $y' \neq y_{true}$, we have

$$\begin{aligned} M(y_{true}, y') &= min_x(f_\theta(x)_{y_{true}} - f_\theta(x)_{y'}) \\ &= min_x(e_{y_{true}} - e_{y'})f_\theta(x) \end{aligned} \tag{5.2}$$

where e_i is the i^{th} standard basis vector. For any $y' \neq y_{true}$, if we can verify that $M(y_{true}, y') > 0$, which means the true label will always has the highest confidence score, f_θ is certifiably robust at x_0 within radius ϵ with respect to L_p norm. In our paper, we focus on the L_∞ norm.

5.3 Interval Bound Propagation

In this section, we will discuss IBP in detail and introduce the elision of the last layer technique.

5.3.1 Interval Bound Propagation

It is not trivial to find exact minimum margin $M(y_{true}, y')$ (hereafter $M_{y'}$) and prove $M_{y'} > 0$. Instead, we could look for a loose lower bound of $M_{y'}$ and control the value inside this bound. To this end, we consider the framework of IBP [27,34] to train a provably robust LDC classifier to L_∞ adversarial perturbation of size ϵ. IBP is an algorithm that can be used to find a lower bound of the minimum margin $M_{y'}$ by bounding the activation z_k of each layer. Specifically, it propagates the axis aligned bounding box from layer to layer using interval arithmetic. For L_∞ norm-bounded perturbation by ϵ, lower bounds and upper bounds of each layer can be represented by the following equations.

$$\begin{aligned} \bar{z}_{0,i}(\epsilon) &= x_{0,i} + \epsilon \\ \underline{z}_{0,i}(\epsilon) &= x_{0,i} - \epsilon \\ &... \\ \bar{z}_{k,i}(\epsilon) &= max_{\underline{z}_{k-1}(\epsilon) \le z_{k-1} \le \bar{z}_{k-1}(\epsilon)} h_{k,i}(z_{k-1}) \\ \underline{z}_{k,i}(\epsilon) &= min_{\underline{z}_{k-1}(\epsilon) \le z_{k-1} \le \bar{z}_{k-1}(\epsilon)} h_{k,i}(z_{k-1}) \\ &... \end{aligned} \tag{5.3}$$

where $z_{k,i}$ is the ith coordinate of z_k and $z_k = h_k(z_{k-1})$. In LDC network setting, there are three layers, thus $k = 0$ to 3. $h_k(z)$ is the transformation function of kth layer, which is either affine transformation or element-wise monotonic activation function,

$Concat(\cdot)$, $Bin(\cdot)$ and $Tanh(\cdot)$. For affine layer, $h_k(z_{k-1}) = Wz_{k-1} + b$, obtaining the upper bound and lower bound, i.e. solving the above optimization problem, can be done efficiently with two matrix multiplication as follows.

$$\begin{aligned}
\mu_{k-1} &= \tfrac{\bar{z}_{k-1} + \underline{z}_{k-1}}{2} \\
r_{k-1} &= \tfrac{\bar{z}_{k-1} - \underline{z}_{k-1}}{2} \\
\mu_k &= W\mu_{k-1} + b \\
r_k &= |W| r_{k-1} \\
\bar{z}_k &= \mu_k + r_k \\
\underline{z}_k &= \mu_k - r_k
\end{aligned} \tag{5.4}$$

where $|\cdot|$ is element-wise absolute value operator. When $h_k(z_{k-1})$ is element-wise monotonic activation function, such as $Concat(\cdot)$, $Bin(\cdot)$ and $Tanh(\cdot)$, we have:

$$\begin{aligned}
\bar{z}_k &= h_k(\bar{z}_{k-1}) \\
\underline{z}_k &= h_k(\underline{z}_{k-1})
\end{aligned} \tag{5.5}$$

In the case of LDC classifier, referring to section 5.2, after propagating we could obtain the upper and lower bounds of the output logits, $\bar{z}_3$ and $\underline{z}_3$. With the bounds of z_3 and the IBP method for affine layer, a loose lower bound of minimum margin $M_{y'}$ can be computed as

$$\begin{aligned}\underline{M_{y'}} &= min_{\underline{z}_3 \leq z_3 \leq \bar{z}_3}(e_{y_{true}} - e_{y'})z_3 \\ &= e_{y_{true}}\underline{z}_3 - e_{y'}\bar{z}_3 \\ &= \underline{z}_{3,y_{true}} - \bar{z}_{3,y'} \\ &\leq min_{\underline{z}_0 \leq x \leq \bar{z}_0}(e_{y_{true}} - e_{y'})f_\theta(x) = M_{y'}\end{aligned} \tag{5.6}$$

For any class label y' other than the true label y_{true}, to make the lower bound of minimum margin, $\underline{M_{y'}}$, larger than 0, we can construct worst case prediction $\hat{z}_k$, where the logit of the true class is equal to its lower bound and the other logits are equal to their upper bound. Note that, if $\epsilon = 0$, $\hat{z}_k = z_k$.

$$\hat{z}_{k,y}(\epsilon) = \begin{cases} \bar{z}_{k,y}(\epsilon) & y \neq y_{true} \\ \underline{z}_{k,y}(\epsilon) & y = y_{true} \end{cases} \tag{5.7}$$

We then minimize a worst-case cross entropy loss $\mathcal{L}(\hat{z}_k, y_{true})$ during the training procedure. However, a direct application of worst-case cross entropy loss alone does not work since the propagated bounds are too loose. In reality, As shown in Fig. 5.1, during training stage, we feed the network with both original training input, z_0, its upper bound, $\bar{z}_0$, and lower bound, $\underline{z}_0$, then minimize a combination of normal cross-entropy loss and worst case cross-entropy loss.

$$\mathcal{L} = k\mathcal{L}(z_k, y_{true}) + (1-k)\mathcal{L}(\hat{z}_k, y_{true}) \tag{5.8}$$

Where k is a trade-off parameter, which controls the relative weight of robust training versus fitting to the original input images.

5.3.2 Elision of Last Layer

Considering the fact that the last layer in LDC network is a linear layer, $z_3 = W_2^b z_2$, to make the calculated lower bound of minimum margin, $\underline{M_{y'}}$, tighter, we could elide the bound propagation of the last linear layer.

$$
\begin{aligned}
\underline{M_{y'}} &= min_{\underline{z}_3 \leq z_3 \leq \bar{z}_3}(e_{y_{true}} - e_{y'})z_3 \\
&\leq min_{\underline{z}_2 \leq z_2 \leq \bar{z}_2}(e_{y_{true}} - e_{y'})W_2^b z_2 \\
&= min_{\underline{z}_2 \leq z_2 \leq \bar{z}_2}\hat{W} z_2 = \underline{M_{y'}^e} \\
&\leq min_{\underline{z}_0 \leq x \leq \bar{z}_0}(e_{y_{true}} - e_{y'})f_\theta(x) = M_{y'}
\end{aligned}
\tag{5.9}
$$

So, minimizing $\hat{W} z_2$ over $\underline{z}_2 \leq z_2 \leq \bar{z}_2$, with $\hat{W} = (e_{y_{true}} - e_{y'})W_2^b$, gives a tighter lower bound, $\underline{M_{y'}^e}$, of minimum margin $M_{y'}$. By doing so, we could bypass the additional relaxation induced by the last linear layer.

5.4 Results

We will present our evaluation results based on MNIST and fashion MNIST dataset in this section. We first discuss the experiment setup. In what follows, the nominal accuracy and verified accuracy are shown for each trained model with different training epsilon. Besides, the robustness results of our trained models against PGD attack and memory cell errors are also displayed in this section.

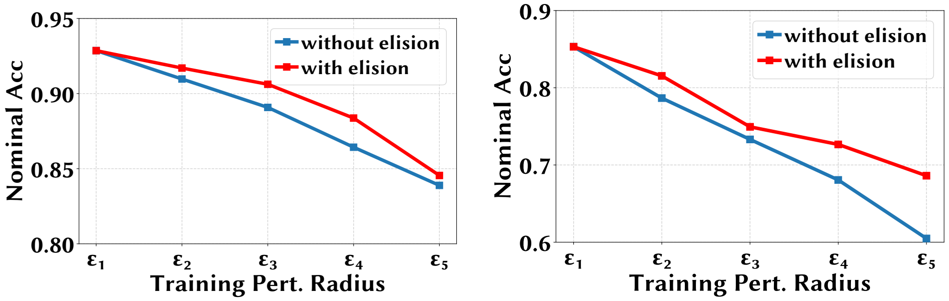
Table 5.1: Configuration of training procedure for each dataset with different training perturbation radii.

Dataset	**Pert. Radius**	**Without Elision**		**With Elision**	
		LR	**WD**	**LR**	**WD**
MNIST	ϵ_0	0.0001	0.0001	0.0001	0.0001
	$\epsilon_{0.02}$	0.0001	0	0.0001	0.0001
	$\epsilon_{0.05}$	0.0001	0	0.0001	0.0001
	$\epsilon_{0.08}$	0.0001	0.0001	0.0001	0.001
	$\epsilon_{0.1}$	0.0001	0.001	0.0001	0.01
Fashion MNIST	ϵ_0	0.0001	0.0001	0.0001	0.0001
	$\epsilon_{0.02}$	0.0001	1e-5	0.001	0.01
	$\epsilon_{0.05}$	0.001	0.01	0.001	0.001
	$\epsilon_{0.08}$	0.0001	0.0001	0.0001	0.01
	$\epsilon_{0.1}$	0.001	1e-5	0.001	0.01

Note: LR represents learning rate. WD represents weight decay.

5.4.1 Experiment Setup

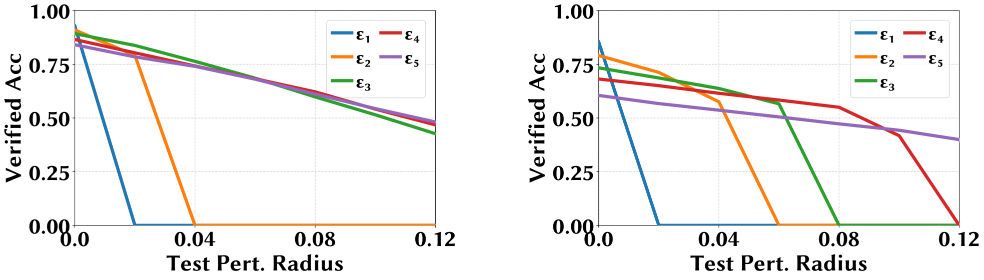
Hyperparameters Setting Hyperparameters of LDC model comply with that in [26]. D_v/D_f is set to 4/64 to get a good trade-off between good accuracy and relatively small model size. The criterion of the training process employs $CrossEntropyLoss(\cdot)$ method. $Adam(\cdot)$ method is adopted as the optimizer following SOTA training strategy [75]. Even if the $Adam(\cdot)$ method intrinsically adapts the learning rate to each parameter, tuning the initial learning rate and decay scheme for $Adam(\cdot)$ yield significant performance improvement [136]. Thus, we implement grid-search mechanism to find the best initial learning rate and weight decay. Besides, we also adopt exponential learning rate decay with decay rate of 0.95 to the provided initial learning rate for a better convergence. Table 5.1 shows the best choice of initial learning rate and weight decay for different dataset with different training perturbation radii. The 5 different perturbation radii ϵ_1, ϵ_2, ϵ_3, ϵ_4, ϵ_5 represent the L_∞ perturbation of 0, 0.02, 0.05, 0.08, and 0.1 associated to the normalized input $x \subseteq [0, 1]^N$.



(a) (b)

Figure 5.2: Nominal accuracy of LDC models with different training perturbation radii, ϵ_1, ϵ_2, ϵ_3, ϵ_4, and ϵ_5, representing perturbation radius of 0, 0.02, 0.05, 0.08, and 0.1 respectively. The blue line are the results without elision technique and the red line represents the one with eliding the last layer. (a) MNIST dataset. (b) Fashion MNIST dataset.

Training Scheduling According to section 5.3, the final loss function to minimize is a combination of normal cross entropy loss and worst case cross entropy loss. The relative importance of the worst case loss is determined by the hyperparameter k. According to literature, it achieves better results by slowly reducing k starting from 1 until 0.5. The same strategy is used for training perturbation radius, staring with 0 and slowly being raised up to the target value. In reality, the total iteration in our experiment is set to 120000 with batch size of 64. During the first 2000 iterations, the model is trained to reduce nominal loss alone, which can be regarded as a warm up period. Starting from the $2000th$ iteration, the model entered a linearly ramp up phase by gradually decreasing parameter k and increasing the perturbation radius. After 10000 iterations, parameter k and the training perturbation radius settle to 0.5 and the target radius respectively.



(a)

(b)

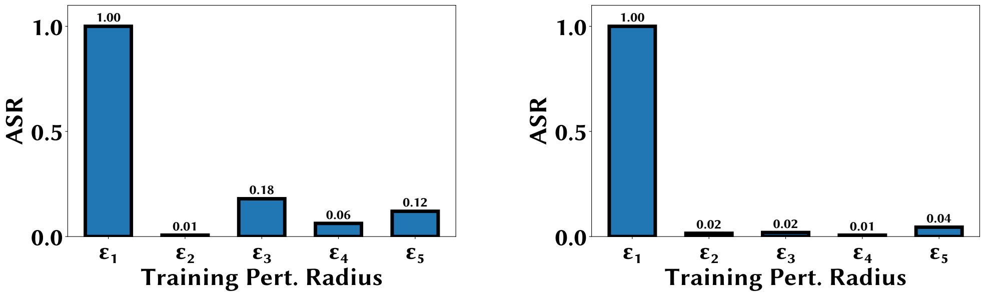
Figure 5.3: Verified accuracy against different test perturbation radii from 0 to 0.12. Five different models, ϵ_1, ϵ_2, ϵ_3, ϵ_4, and ϵ_5, associate with training perturbation radius of 0, 0.02, 0.05, 0.08, and 0.1 respectively. (a) MNIST dataset. (b) Fashion MNIST dataset.

5.4.2 Nominal Accuracy and Verified Accuracy

Using a range of perturbation radii $\epsilon \subseteq \{0, 0.02, 0.05, 0.08, 0.1\}$, we train LDC architecture on both MNIST and fashion MNIST dataset. Note that when training perturbation radius $\epsilon = 0$, the normal training with standard cross-entropy loss is performed. After training, we obtain 5 robust models for each dataset. During testing, we test each of the trained models against adversarial perturbation from 0 to 0.12. We add the test adversarial perturbation to each test image and compute the worst case prediction, based on which we obtain the inference accuracy over the test set, which is called verified accuracy. Note that when test perturbation radius is 0, the nominal test accuracy is obtained, which is called nominal accuracy.

To test the effectiveness of elision technique, we compare the nominal accuracy of five trained models with and without eliding the last layer. Fig. 5.2(a) presents the results based on MNIST dataset. From the figure we can see the nominal test accuracy of standard LDC model with zero training perturbation radius is around 93%. The red line shows the results when eliding the last layer during IBP procedure. The blue line displays that without elision technique. In both lines, the nominal accuracy is decreasing with the increasing of training perturbation radius. This corroborate that the addition of verification loss deteriorate the ability of the model fitting to the dataset. However, the red line is sliding slower then the blue one. This is because that the elision of the last layer makes the calculated bound tighter and the penalty to the nominal accuracy becomes less severe compared to that of standard IBP method without elision scheme. Similarly, we give the experiment results of nominal accuracy on the basis of fashion MNIST dataset, referring to Fig. 5.2(b).

On the other hand, we demonstrate the verified accuracy of the trained models with standard IBP method without elision of the last layer. We choose a spectrum of test perturbation radii from 0 to 0.12 spaced by 0.02. Fig. 5.3(a) gives the results of MNIST dataset. As we can see from the figure, the standard model without robust training presents a zero verified accuracy when the test adversarial perturbation is above 0.02. The model trained with 0.02 training perturbation radius exhibits immunity to test adversarial perturbation of 0.02 and becomes vulnerable again when the test perturbation increases to 0.04. Models with training perturbation radius of 0.05, 0.08, and 0.1 show similar robustness. However, the verified accuracy of the model trained with smaller training perturbation radius degrades more quickly as the test perturbation radius increases. The effectiveness of increasing training perturbation radius becomes more obvious in the results of fashion MNIST dataset. As shown in Fig. 5.3(b), the model trained with higher training perturbation radius show higher verified accuracy especially for large test perturbation radius.



(a)

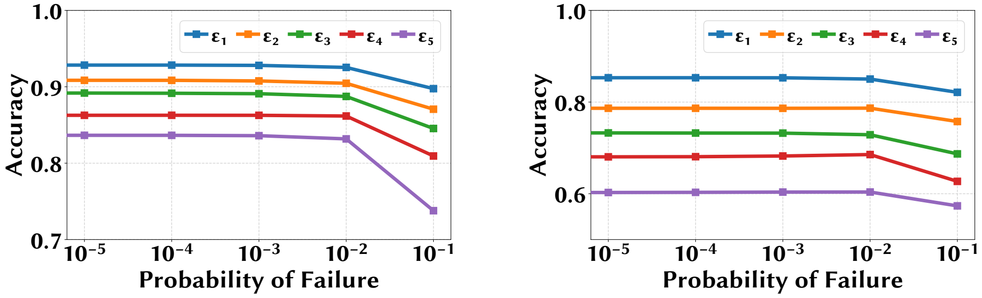
(b)

Figure 5.4: Attack success rate (ASR) of PGD attacking method to five LDC models, ϵ_1, ϵ_2, ϵ_3, ϵ_4, and ϵ_5, trained with IBP with training perturbation radius of 0, 0.02, 0.05, 0.08, and 0.1 respectively. (a) MNIST dataset. (b) Fashion MNIST dataset.

5.4.3 Robustness Against PGD

We also assess the trained models' tolerance against the powerful attack method, PGD. PGD is a white-box attack algorithm which means the attacker has full access to the model, including models' weights and gradients. It is actually an iterative version of FGSM [138]. In our experiment, we calculate the attack success rate (ASR) of PGD method under 200 iterations. Specifically, we use FGSM to introduce adversarial perturbation to each test image, which can be correctly classified by model. We calculate the percentage of images that can be crafted within 200 iterations to mislead the classifier, which is denoted as ASR. We use ASR of PGD to indicate how robust the model is.

Fig. 5.4(a) and fig. 5.4(b) are the results of PGD to the robust LDC models MNIST and fashion MNIST dataset. From the figure, we can see that for both MNIST and fashion MNIST dataset, PGD can achieve 100% ASR attacking the models without robust training. However, to the models trained with IBP, the ASR decrease significantly. PGD fails to attack the models trained with IBP across the full training perturbation spectrum.



(a) (b)

Figure 5.5: Classification accuracy of five trained models, ϵ_1, ϵ_2, ϵ_3, ϵ_4, and ϵ_5, with faulty memory cells, when the probability of failure for each memory cell is 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2} and 10^{-1}. ϵ_1, ϵ_2, ϵ_3, ϵ_4, and ϵ_5 correspond to training perturbation radius of 0, 0.02, 0.05, 0.08, and 0.1 respectively. (a) MNIST dataset. (b) Fashion MNIST dataset.

5.4.4 Robustness Against Memory Errors

In this paper, we also evaluate the performance of LDC model with IBP robust training to erroneous memory cells. To demonstrate the robustness of the models, we conduct RTL fault simulations where we inject memory bit flips during every clock cycle of execution. In the simulation, we set the probability of failure for each memory cell to be 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, respectively.

Fig. 5.5(a) and Fig. 5.5(b) present the test accuracy with memory errors. X axis displays the probability of failure for each memory cell in every clock cycle. From the figures we can see that for both MNIST and fashion MNIST dataset, the accuracy diminishes when the training perturbation radius increases. And for each model, the performance maintains a high accuracy when the probability is lower than 10^{-2} even if the accuracy starts to drop afterwards.

5.5 Conclusion

In this paper, we implement the state-of-the-art verifiably robust training method, IBP, to LDC classifier and provide some baseline experiment results for MNIST and fashion MNIST dataset. Our results prove the effectiveness of IBP algorithm in LDC model and present a robust model against real-life powerful adversarial attacking technique, PGD and against memory errors as well.

Chapter 6

Conclusions

In this dissertation, we investigate and explore the security in Internet edge system from two perspectives, communication between edge devices and reliability of on-device classification.

We first present secret key generation schemes to secure data transmission between edge devices. To overcome the limitation in previous proximity-based authentication literature, that the authentication distance among edge devices is too small, we propose PowerKey in chapter 2, which utilizes EMI within an electrical domain to authenticate plugged edge devices. The experiment results show a 100% KMR at a BGR of up to 52.7 bits/sec. To ease the constrain that devices have to be plugged to the outlets, we design another secret key generation scheme, CompKey, to make use of EMR generated from a source computer. Communicating parties in the vicinity of a source computer could extract secret keys by sensing the same EMR signal. Through evaluation, we show that devices within 0.5m away from the computer can get identical keys with 10 bits/s BGR and 100% KMR.

We then dive into the vulnerability of HDC-based on-device classification models to adversarial attacks. In chapter 4, we propose a grey-box adversarial attack algorithm, GA-CGC-PA, targeted at HDC classifier on MINTS handwritten digits. Our results show that generated adversarial images can successfully mislead the HDC classifier to produce wrong prediction labels while keeping the amount of added perturbation noiss as little as possible. In addition, we study the countermeasures to all kinds of adversarial attacks, certified robustness of HDC based classifier. We focus on an efficient version of HDC based classification model, LDC classifier, and adopt IBP method to train a probable robust model over all possible adversarial perturbations within L_∞ norm-bounded ball. We evaluate the algorithm on both MNIST and fashion MNIST datasets. The experiment results show that our trained models are immune and robust against strong project gradient descent (PGD) attacking scheme and memory errors.

Finally, we hope this thesis can provide some insights for future topics.

• To secure communication between edge devices, both PowerKey and CompKey have assumptions that limit their application. Devices using PowerKey have to be plugged in power outlets and devices with CompKey scheme have to be in the vicinity of a source computer. Relaxing these assumptions could be a promising future study for secret key generation.

• To study the vulnerability of HDC classifiers, we built a HDC classification framework on the basis of MNIST dataset and the experiment results of our algorithm are based on MNIST dataset alone. Experiment with more sophisticated tasks still require more investigation.

- In this thesis, the trained LDC models still have chance to be attacked within norm bounded perturbation and the nominal accuracy is decreasing with the training perturbation radius. A more robust training scheme with less nominal accuracy decay is an interesting topic for future exploration.

Bibliography

[1] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430, 2018.

[2] Mohammad Ahmad Alia and A Yahya. Public–key steganography based on matching method. *European Journal of Scientific Research*, 40(2):223–231, 2010.

[3] Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, Huan Zhang, Cho-Jui Hsieh, and Mani B Srivastava. Genattack: Practical black-box attacks with gradient-free optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1111–1119, 2019.

[4] Asadi, Arash, Qing Wang, and Vincenzo Mancuso. A survey on device-to-device communication in cellular networks. In *IEEE Communications Surveys & Tutorials 16*, 2014.

[5] IEEE Standards Associatio. Ieee draft standard for broadband over power line networks: Medium access control and physical layer specifications amendment: Enhancement for internet of things applications. In *https://standards.ieee.org/project/1901a.html*, 2018.

[6] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, 2018.

[7] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[8] L Bassham. A statistical test suite for the validation of random number generators and pseudo random number generators for cryptographic applications. In *NIST SP*, pages 800–22rev1a, 2010.

[9] Simone Benatti, Fabio Montagna, Victor Kartsch, Abbas Rahimi, Davide Rossi, and Luca Benini. Online learning and classification of emg-based gestures on a parallel ultra-low power platform using hyperdimensional computing. *IEEE transactions on biomedical circuits and systems*, 13(3):516–528, 2019.

[10] Dinabandhu Bhandari, CA Murthy, and Sankar K Pal. Genetic algorithm with elitist model and its convergence. *International journal of pattern recognition and artificial intelligence*, 10(06):731–747, 1996.

[11] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[12] Alessio Burrello, Kaspar Schindler, Luca Benini, and Abbas Rahimi. Hyperdimensional computing with local binary patterns: One-shot learning of seizure onset and identification of ictogenic brain regions using short-time ieeg recordings. *IEEE Transactions on Biomedical Engineering*, 67(2):601–613, 2019.

[13] Robert Callan, Alenka Zajic, and Milos Prvulovic. A practical methodology for measuring the side-channel signal available to the attacker for instruction-level events. In *2014 47th Annual IEEE/ACM International Symposium on Microarchitecture*, pages 242–254. IEEE, 2014.

[14] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2016.

[15] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy*, pages 39–57. Ieee, 2017.

[16] Cheng-Yang Chang, Yu-Chuan Chuang, and An-Yeu Andy Wu. Task-projected hyperdimensional computing for multi-task learning. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 241–251. Springer, 2020.

[17] En-Jui Chang, Abbas Rahimi, Luca Benini, and An-Yeu Andy Wu. Hyperdimensional computing-based multimodality emotion recognition with physiological signals. In *2019 IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 137–141. IEEE, 2019.

[18] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.

[19] Wencheng Chen and Hongyu Li. Adversarial attacks on voice recognition based on hyper dimensional computing. *Journal of Signal Processing Systems*, 93(7):709–718, 2021.

[20] Tejalal Choudhary, Vipul Mishra, Anurag Goswami, and Jagannathan Sarangapani. A comprehensive survey on model compression and acceleration. *Artificial Intelligence Review*, 53(7):5113–5155, 2020.

[21] Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *International Conference on Machine Learning*, pages 854–863. PMLR, 2017.

[22] George C Clark Jr and J Bibb Cain. *Error-correction coding for digital communications.* Springer Science & Business Media, 2013.

[23] Jeremy M Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing, 2019.

[24] Gabe Cohn, Daniel Morris, Shwetak N Patel, and Desney S Tan. Your noise is my command: sensing gestures using the body as an antenna. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 791–800, 2011.

[25] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Trans. Inf. Theor.*, 22(6):644–654, September 2006.

[26] Shijin Duan, Xiaolin Xu, and Shaolei Ren. A brain-inspired low-dimensional computing classifier for inference on tiny devices, 03 2022.

[27] Ruediger Ehlers. Formal verification of piece-wise linear feed-forward neural networks, 2017.

[28] Miro Enev, Sidhant Gupta, Tadayoshi Kohno, and Shwetak N Patel. Televisions, video privacy, and powerline electromagnetic interference. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 537–550, 2011.

[29] E Paxon Frady, Denis Kleyko, and Friedrich T Sommer. A theory of sequence indexing and working memory in recurrent neural networks. *Neural Computation*, 30(6):1449–1513, 2018.

[30] Lulu Ge and Keshab K Parhi. Classification using hyperdimensional computing: A review. *IEEE Circuits and Systems Magazine*, 20(2):30–47, 2020.

[31] A. Goldsmith. Wireless communications. In *Cambridge University Press*, 2005.

[32] Neil Zhenqiang Gong, Altay Ozen, Yu Wu, Xiaoyu Cao, Richard Shin, Dawn Song, Hongxia Jin, and Xuan Bao. Piano: Proximity-based user authentication on voice-powered internet-of-things devices. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 2212–2219. IEEE, 2017.

[33] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

[34] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy A. Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *CoRR*, abs/1810.12715, 2018.

[35] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660, 2013.

[36] Manoj Gulati, Shobha Sundar Ram, and Amarjeet Singh. An in depth study into using emi signatures for appliance identification. In *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-efficient Buildings*, pages 70–79, 2014.

[37] Onat Gungor, Tajana Rosing, and Baris Aksanli. Res-hd: Resilient intelligent fault diagnosis against adversarial attacks using hyper-dimensional computing, 2022.

[38] Sidhant Gupta, Matthew S Reynolds, and Shwetak N Patel. Electrisense: single-point sensing using emi for electrical event detection and classification in the home. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 139–148, 2010.

[39] Mordechai Guri, Assaf Kachlon, Ofer Hasson, Gabi Kedma, Yisroel Mirsky, and Yuval Elovici. Gsmem: Data exfiltration from air-gapped computers over GSM frequencies. In *USENIX Security*, 2015.

[40] Eman Hassan, Yasmin Halawani, Baker Mohammad, and Hani Saleh. Hyper-dimensional computing challenges and opportunities for ai applications. *IEEE Access*, 10:97651–97664, 2021.

[41] Pengfei Hu, Parth H Pathak, Yilin Shen, Hongxia Jin, and Prasant Mohapatra. Pcasa: Proximity based continuous and secure authentication of personal devices. In *2017 14th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9. IEEE, 2017.

[42] Po-Sen Huang, Robert Stanforth, Johannes Welbl, Chris Dyer, Dani Yogatama, Sven Gowal, Krishnamurthy Dvijotham, and Pushmeet Kohli. Achieving verified robustness to symbol substitutions via interval bound propagation. *arXiv preprint arXiv:1909.01492*, 2019.

[43] Y. Bengio I. Goodfellow and A. Courville. Deep learning. In *MIT Press*. http://www.deeplearningbook.org, 2016.

[44] Mohsen Imani, Samuel Bosch, Mojan Javaheripi, Bita Rouhani, Xinyu Wu, Farinaz Koushanfar, and Tajana Rosing. Semihd: Semi-supervised learning using hyperdimensional computing. In *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–8. IEEE, 2019.

[45] Mohsen Imani, Chenyu Huang, Deqian Kong, and Tajana Rosing. Hierarchical hyperdimensional computing for energy efficient classification. In *2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2018.

[46] Mohsen Imani, John Hwang, Tajana Rosing, Abbas Rahimi, and Jan M Rabaey. Low-power sparse hyperdimensional encoder for language recognition. *IEEE Design & Test*, 34(6):94–101, 2017.

[47] Mohsen Imani, John Messerly, Fan Wu, Wang Pi, and Tajana Rosing. A binary learning framework for hyperdimensional computing. In *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 126–131. IEEE, 2019.

[48] Mohsen Imani, Justin Morris, John Messerly, Helen Shu, Yaobang Deng, and Tajana Rosing. Bric: Locality-based encoding for energy-efficient brain-inspired hyperdimensional computing. In *Proceedings of the 56th Annual Design Automation Conference 2019*, pages 1–6, 2019.

[49] Mohsen Imani, Abbas Rahimi, Deqian Kong, Tajana Rosing, and Jan M Rabaey. Exploring hyperdimensional associative memory. In *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 445–456. IEEE, 2017.

[50] Mohsen Imani, Sahand Salamat, Saransh Gupta, Jiani Huang, and Tajana Rosing. Fach: Fpga-based acceleration of hyperdimensional computing by reducing computational complexity. In *Proceedings of the 24th Asia and South Pacific Design Automation Conference*, pages 493–498, 2019.

[51] Intel. Intel mobile communications software-defined radio, https://www.intel.com/content/www/us/en/products/docs/wireless-products/mobile-communications/software-defined-radio.html.

[52] Mohammad A Islam and Shaolei Ren. Ohm's law in data centers: A voltage side channel for timing power attacks. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 146–162, 2018.

[53] Suman Jana, Sriram Nandha Premnath, Mike Clark, Sneha K Kasera, Neal Patwari, and Srikanth V Krishnamurthy. On the effectiveness of secret key extraction from wireless signal strength in real environments. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 321–332, 2009.

[54] Andre Kalamandeen, Adin Scannell, Eyal de Lara, Anmol Sheth, and Anthony LaMarca. Ensemble: cooperative proximity-based authentication. In *MobiSys*, 2010.

[55] Pentti Kanerva. Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors. *Cognitive computation*, 1(2):139–159, 2009.

[56] Tapas Kanungo, David M Mount, Nathan S Netanyahu, Christine D Piatko, Ruth Silverman, and Angela Y Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, 24(7):881–892, 2002.

[57] Nikolaos Karapanos, Claudio Marforio, Claudio Soriente, and Srdjan Capkun. Sound-proof: Usable two-factor authentication based on ambient sound. In *USENIX Security*, 2015.

[58] Geethan Karunaratne, Manuel Le Gallo, Giovanni Cherubini, Luca Benini, Abbas Rahimi, and Abu Sebastian. In-memory hyperdimensional computing. *Nature Electronics*, 3(6):327–337, 2020.

[59] Behnam Khaleghi, Mohsen Imani, and Tajana Rosing. Prive-hd: Privacy-preserved hyperdimensional computing. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2020.

[60] Denis Kleyko, Evgeny Osipov, Nikolaos Papakonstantinou, and Valeriy Vyatkin. Hyperdimensional computing in industrial systems: the use-case of distributed fault isolation in a power plant. *IEEE Access*, 6:30766–30777, 2018.

[61] Denis Kleyko, Abbas Rahimi, Dmitri A Rachkovskij, Evgeny Osipov, and Jan M Rabaey. Classification and recall with binary hyperdimensional computing: Tradeoffs in choice of density and mapping characteristics. *IEEE transactions on neural networks and learning systems*, 29(12):5880–5898, 2018.

[62] Aounon Kumar and Tom Goldstein. Center smoothing: Certified robustness for networks with structured outputs, 2021.

[63] Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy, 2018.

[64] Kyuin Lee, Neil Klingensmith, Suman Banerjee, and Younghyun Kim. Voltkey: Continuous secret key generation based on power line noise for zero-involvement pairing and authentication. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–26, 2019.

[65] Jonathan Lester, Blake Hannaford, and Gaetano Borriello. Are you with me? – using accelerometers to determine if two devices are carried by the same person. In *International Conference on Pervasive Computing*, 2004.

[66] Alexander Levine and Soheil Feizi. Robustness certificates for sparse adversarial attacks by randomized ablation, 2019.

[67] Alexander Levine, Aounon Kumar, Thomas Goldstein, and Soheil Feizi. Tight second-order certificates for randomized smoothing, 2020.

[68] Bai Li, Changyou Chen, Wenlin Wang, and Lawrence Carin. Certified adversarial robustness with additive noise. *Advances in neural information processing systems*, 32, 2019.

[69] En Li, Zhi Zhou, and Xu Chen. Edge intelligence: On-demand deep learning model co-inference with device-edge synergy. In *Proceedings of the 2018 Workshop on Mobile Edge Communications*, pages 31–36, 2018.

[70] Linyi Li, Xiangyu Qi, Tao Xie, and Bo Li. Sok: Certified robustness for deep neural networks. *CoRR*, abs/2009.04131, 2020.

[71] Shancang Li, Li Da Xu, and Shanshan Zhao. 5g internet of things: A survey. *Journal of Industrial Information Integration*, 10:1–9, 2018.

[72] Yang Li, Rui Tan, and David KY Yau. Natural timestamps in powerline electromagnetic radiation. *ACM Transactions on Sensor Networks (TOSN)*, 14(2):1–30, 2018.

[73] H. Liu, Yang Wang, Jie Yang, and Yingying. Chen. Fast and practical secret key extraction by exploiting channel response. In *Proceedings IEEE INFOCOM*, pages 3048–3056. IEEE, 2013.

[74] Xiaolei Liu, Teng Hu, Kangyi Ding, Yang Bai, Weina Niu, and Jiazhong Lu. A black-box attack on neural networks based on swarm evolutionary algorithm. In *Australasian Conference on Information Security and Privacy*, pages 268–284. Springer, 2020.

[75] Zechun Liu, Zhiqiang Shen, Shichao Li, Koen Helwegen, Dong Huang, and Kwang-Ting Cheng. How do adam and training strategies help bnns optimization. In *ICML*, pages 6936–6946, 2021.

[76] Low Electromagnetic Emissions Office Equipment. Electromagnetic emissions of desktop computers, https://www.lowemfoffice.com/desktop$_computers.htm$.

[77] Youjing Lu, Fan Wu, Shaojie Tang, Linghe Kong, and Guihai Chen. Free: a fast and robust key extraction mechanism via inaudible acoustic signal. In *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 311–320, 2019.

[78] Zhiqing Luo, Wei Wang, Jiang Xiao, Qianyi Huang, Tao Jiang, and Qian Zhang. Authenticating on-body backscatter by exploiting propagation signatures. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3):1–22, 2018.

[79] Zhihan Lv. Security of internet of things edge devices. *Software: Practice and Experience*, 51(12):2446–2456, 2021.

[80] A. Rostamizadeh M. Mohri and A. Talwalkar. Foundations of machine learning. In *MIT press*, 2018.

[81] Dongning Ma, Jianmin Guo, Yu Jiang, and Xun Jiao. Hdtest: Differential fuzz testing of brain-inspired hyperdimensional computing, 2021.

[82] Suhas Mathur, Robert Miller, Alexander Varshavsky, Wade Trappe, and Narayan Mandayam. Proximate: proximity-based secure pairing using ambient wireless signals. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 211–224, 2011.

[83] Suhas Mathur, Wade Trappe, Narayan Mandayam, Chunxuan Ye, and Alex Reznik. Radio-telepathy: extracting a secret key from an unauthenticated wireless channel. In *Proceedings of the 14th ACM international conference on Mobile computing and networking*, pages 128–139, 2008.

[84] MathWorks. Run test for randomness, https://www.mathworks.com/help/stats/runstest.html.

[85] Yair Meidan, Michael Bohadana, Asaf Shabtai, Martin Ochoa, Nils Ole Tippenhauer, Juan Davis Guarnizo, and Yuval Elovici. Detection of unauthorized iot devices using machine learning techniques. *arXiv preprint arXiv:1709.04647*, 2017.

[86] Massimo Merenda, Carlo Porcaro, and Demetrio Iero. Edge machine learning for ai-enabled iot devices: A review. *Sensors*, 20(9):2533, 2020.

[87] Markus Miettinen, Nadarajah Asokan, Thien Duc Nguyen, Ahmad-Reza Sadeghi, and Majid Sobhani. Context-based zero-interaction pairing and key evolution for advanced personal devices. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 880–891, 2014.

[88] Markus Miettinen, Thien Duc Nguyen, Ahmad-Reza Sadeghi, and N Asokan. Revisiting context-based authentication in iot. In *Proceedings of the 55th Annual Design Automation Conference*, pages 1–6, 2018.

[89] Anton Mitrokhin, P Sutor, Cornelia Fermüller, and Yiannis Aloimonos. Learning sensorimotor control with neuromorphic sensors: Toward hyperdimensional active perception. *Science Robotics*, 4(30):eaaw6736, 2019.

[90] Ali Moin, Andy Zhou, Abbas Rahimi, Simone Benatti, Alisha Menon, Senam Tamakloe, Jonathan Ting, Natasha Yamamoto, Yasser Khan, Fred Burghardt, et al. An emg gesture recognition system with flexible high-density sensors and brain-inspired high-dimensional classifier. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2018.

[91] Pawel Morawiecki, Przemyslaw Spurek, Marek Smieja, and Jacek Tabor. Fast and stable interval bounds propagation for training verifiably robust models. *CoRR*, abs/1906.00628, 2019.

[92] Arsalan Mosenia and Niraj K Jha. A comprehensive study of security of internet-of-things. *IEEE Transactions on emerging topics in computing*, 5(4):586–602, 2016.

[93] MG Sarwar Murshed, Christopher Murphy, Daqing Hou, Nazar Khan, Ganesh Ananthanarayanan, and Faraz Hussain. Machine learning at the network edge: A survey. *ACM Computing Surveys (CSUR)*, 54(8):1–37, 2021.

[94] Othmane Nait Hamoud, Tayeb Kenaza, and Yacine Challal. Security in device-to-device communications: a survey. *IET Networks*, 7(1):14–22, 2018.

[95] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *CVPR Workshops*, volume 2, page 2, 2017.

[96] Electronic Code of U.S. Federal Regulations. Unintentional radiators, section 15.107 — conducted limits. 2018.

[97] Jianli Pan and James McElhannon. Future edge cloud and edge computing for internet of things applications. *IEEE Internet of Things Journal*, 5(1):439–449, 2017.

[98] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

[99] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks, 2015.

[100] Anay Pattanaik, Zhenyi Tang, Shuijing Liu, Gautham Bommannan, and Girish Chowdhary. Robust deep reinforcement learning with adversarial attacks. *arXiv preprint arXiv:1712.03632*, 2017.

[101] TOMM Peake. Eavesdropping in communication. *Animal communication networks*, pages 13–37, 2005.

[102] Tony A Plate. Holographic reduced representations. *IEEE Transactions on Neural networks*, 6(3):623–641, 1995.

[103] Jorge Portilla, Gabriel Mujica, Jin-Shyan Lee, and Teresa Riesgo. The extreme edge at the bottom of the internet of things: A review. *IEEE Sensors Journal*, 19(9):3179–3190, 2019.

[104] Abraham Pressman. *Switching power supply design*. McGraw-Hill Education, 2009.

[105] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. Whole-home gesture recognition using wireless signals. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 27–38, 2013.

[106] Kannan Srinivasan Qiao, Yue and Anish Arora. Shape matters, not the size: A new approach to extract secrets from channel. In *HotWireless*, pages 37–42, 2014.

[107] Han Qiu, Yi Zeng, Qinkai Zheng, Tianwei Zhang, Meikang Qiu, and Gerard Memmi. Mitigating advanced adversarial attacks with more advanced gradient obfuscation techniques. *arXiv preprint arXiv:2005.13712*, 2020.

[108] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples, 2018.

[109] Abbas Rahimi, Pentti Kanerva, and Jan M. Rabaey. A robust and energy-efficient classifier using brain-inspired hyperdimensional computing. In *Proceedings of the 2016 International Symposium on Low Power Electronics and Design*, ISLPED '16, page 64–69, New York, NY, USA, 2016. Association for Computing Machinery.

[110] Kui Ren, Tianhang Zheng, Zhan Qin, and Xue Liu. Adversarial attacks and defenses in deep learning. *Engineering*, 6(3):346–360, 2020.

[111] Sahand Salamat, Mohsen Imani, Behnam Khaleghi, and Tajana Rosing. F5-hd: Fast flexible fpga-based framework for refreshing hyperdimensional computing. In *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 53–62, 2019.

[112] Hadi Salman, Greg Yang, Jerry Li, Pengchuan Zhang, Huan Zhang, Ilya Razenshteyn, and Sebastien Bubeck. Provably robust deep learning via adversarially trained smoothed classifiers, 2019.

[113] Dominik Schürmann and Stephan Sigg. Secure communication based on ambient audio. In *IEEE Transactions on mobile computing 12.2*, pages 358–370, 2011.

[114] On Semiconductor. Switch-mode power supply reference manual, https://www.onsemi.com/pub/Collateral/SMPSRM-D.PDF.

[115] On Semiconductor. Power factor correction (pfc) handbook, http://www.onsemi.com/pub/Collateral/HBD853-D.PDF.

[116] Hossein Shafagh and Anwar Hithnawi. Come closer: proximity-based authentication for the internet of things. In *MobiCom*, 2014.

[117] Zhihui Shao, Mohammad A Islam, and Shaolei Ren. Your noise, my signal: Exploiting switching noise for stealthy data exfiltration from desktop computers. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(1):1–39, 2020.

[118] Lingyang Song, Dusit Niyato, Zhu Han, and Ekram Hossain. *Wireless Device-to-Device Communications and Networks.* Cambridge University Press, New York, NY, USA, 2015.

[119] William Stallings. Cryptography and network security principles and practices 4th edition, 2006.

[120] William Stallings. Cryptography and network security: principles and practice. In *Upper Saddle River: Pearson*, 2017.

[121] Statista. Internet of things (iot) connected devices installed base worldwide from 2015 to 2025, https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/.

[122] Jiaye Teng, Guang-He Lee, and Yang Yuan. l_1 adversarial robustness certificates: a randomized smoothing approach. 2019.

[123] Rahul Thapa, Dongning Ma, and Xun Jiao. Hdxplore: Automated blackbox testing of brain-inspired hyperdimensional computing. In *2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pages 90–95. IEEE, 2021.

[124] Arch Toolbox. Electrical power systems in buildings, https://www.archtoolbox.com/materials-systems/electrical/electrical-power-systems.html.

[125] Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 742–749, 2019.

[126] Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks, 2018.

[127] Alex Varshavsky, Adin Scannell, Anthony LaMarca, and Eyal de Lara. Amigo: Proximity-based authentication of mobile devices. In *International Conference on Ubiquitous Computing*, pages 253–270. Springer, 2007.

[128] Sreejaya Viswanathan, Rui Tan, and David KY Yau. Exploiting electrical grid for accurate and secure clock synchronization. *ACM Transactions on Sensor Networks (TOSN)*, 14(2):1–32, 2018.

[129] Jiang Wan, Anthony Bahadir Lopez, and Mohammad Abdullah Al Faruque. Exploiting wireless channel randomness to generate keys for automotive cyber-physical system security. In *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*, pages 1–10. IEEE, 2016.

[130] Wei Wang, Lin Yang, and Qian Zhang. Touch-and-guard: secure pairing through hand resonance. In *UbiComp*, 2016.

[131] Wei Wang, Lin Yang, Qian Zhang, and Tao Jiang. Securing on-body iot devices by exploiting creeping wave propagation. *IEEE Journal on Selected Areas in Communications*, 36(4):696–703, 2018.

[132] Yihan Wang, Zhouxing Shi, Quanquan Gu, and Cho-Jui Hsieh. On the convergence of certified robust training with interval bound propagation. In *International Conference on Learning Representations*, 2022.

[133] Colin Wei and J Zico Kolter. Certified robustness for deep equilibrium models via interval bound propagation. In *International Conference on Learning Representations*, 2021.

[134] Wikipedia. Electromagnetic interference., https://en.wikipedia.org/wiki/ Electromagnetic_interference.

[135] Wikipedia. Hardware random number generator, https://en.wikipedia.org/wiki/Hardware_random_number_generator.

[136] Ashia C. Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. The marginal value of adaptive gradient methods in machine learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4151–4161, Red Hook, NY, USA, 2017. Curran Associates Inc.

[137] Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope, 2017.

[138] Fanyou Wu, Rado Gazo, Eva Haviarova, and Bedrich Benes. Efficient project gradient descent for ensemble adversarial attack. *arXiv preprint arXiv:1906.03333*, 2019.

[139] Wei Xi, Xiang-Yang Li, Chen Qian, Jinsong Han, Shaojie Tang, Jizhong Zhao, and Kun Zhao. Keep: Fast secret key extraction protocol for d2d communication. In *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*, pages 350–359. IEEE, 2014.

[140] Wei Xi, Chen Qian, Jinsong Han, Kun Zhao, Sheng Zhong, Xiang-Yang Li, and Jizhong Zhao. Instant and robust authentication and key agreement among mobile devices. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 616–627, 2016.

[141] Pengjin Xie, Jingchao Feng, Zhichao Cao, and Jiliang Wang. Genewave: Fast authentication and key agreement on commodity mobile devices. *IEEE/ACM Transactions on Networking*, 26(4):1688–1700, 2018.

[142] W Xu, Y Qi, and D Evans. Automatically evading classifiers: A case study on pdf malware classifiers. ndss, 2016.

[143] Xing Xu, Jiefu Chen, Jinhui Xiao, Zheng Wang, Yang Yang, and Heng Tao Shen. Learning optimization-based adversarial perturbations for attacking sequential recognition models. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2802–2822, 2020.

[144] C. Cortes Y. LeCun and C. J. C. Burges. The mnist database of handwritten digits. http://yann.lecun.com/exdb/mnist/.

[145] Zhenyu Yan, Qun Song, Rui Tan, Yang Li, and Adams Wai Kin Kong. Towards touch-to-access device authentication using induced body electric potentials. In *The 25th Annual International Conference on Mobile Computing and Networking*. ACM, aug 2019.

[146] Fangfang Yang and Shaolei Ren. Adversarial attacks on brain-inspired hyperdimensional computing-based classifiers. *arXiv preprint arXiv:2006.05594*, 2020.

[147] Fangfang Yang and Shaolei Ren. On the vulnerability of hyperdimensional computing-based classifiers to adversarial attacks. In *International Conference on Network and System Security*, pages 371–387. Springer, 2020.

[148] Lin Yang, Wei Wang, and Qian Zhang. Secret from muscle: Enabling secure pairing with electromyography. In *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*, pages 28–41, 2016.

[149] Jiawei Yuan, Lu Shi, Shucheng Yu, and Ming Li. Authenticated secret key extraction using channel characteristics for body area networks. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 1028–1030, 2012.

[150] Huan Zhang, Hongge Chen, Chaowei Xiao, Bo Li, Duane S. Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. *CoRR*, abs/1906.06316, 2019.

[151] Junqing Zhang, Trung Q Duong, Alan Marshall, and Roger Woods. Key generation from wireless channels: A review. *Ieee access*, 4:614–626, 2016.