Constructing Robust Classifiers using Cryptographic objects

DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Technology
in
Computer Science

by

Shankhadeep De

[Roll No: CS-1828]

under the guidance of

Dr. Debrup Chakraborty
Associate Professor
Cryptology and Security Research Unit

Indian Statistical Institute
Kolkata-700108, India

July 2020

Abstract

The existence of evasion attacks during the test phase of machine learning algorithms represents a significant challenge to their deployment and understanding. These attacks are carried out by adding imperceptible perturbations to the inputs to generate *adversarial examples*. As of now designing good robust classifiers in real life seems very difficult. But so far most of the studies depict the relationship between computational power of adversary and robustness of the classifier. In this report, we have used some of the cryptographic schemes to create robust classifiers and show the dependency of robustness with adversarial budget.

1 Introduction

One of the basic tasks of machine learning is to learn a classifier from a given data set. Let X be the domain, i.e., the set of objects that are required to be classified, in general the elements of X are a suitable encoding of the objects which are represented by vectors, commonly called as a feature vector. Let Y be a set of possible labels. Let D be an unknown distribution over $X \times Y$ and S be a set of m samples drawn i.i.d. from D. Call L as the training set. A learning algorithm takes a input the training set S and a set of hypothesis functions H,and outputs a classifier $h \in H$. The goal is to (approximately) minimize the error

$$\delta = \Pr_{(x,y) \xleftarrow{D} X \times Y} [h(x) \neq y].$$

Classification problems abound in nature and designing good classifiers are important in today's world. There are numerous practical scenarios where a good classifier makes our lives easier, for example, classifying good emails from spams, classifying pixels in a satellite image as land-mines and safe, and many others.

Designing good classifiers using small training sets which can perform well on previously unseen data have been a problem of interest for many years and numerous techniques to design classifiers have been developed like logistic regression, discriminant analysis, nearest neighbor methods, (deep) neural networks etc[8]. There have also been numerous theoretical studies on classification problems which tries to characterize the class of learn-able functions, compute bounds on the training set size etc[10].

Recently there has been an interest in studying classification problems in the adversarial setting. We describe the problem in one such setting with a simple example. Let us consider a classifier h for images which takes in an image and classifies it into two classes say cat and dog signifying whether the image has in it a cat or a dog. We assume that h performs quite good on examples drawn from the true distribution. The goal of the adversary is to fool this classifier h in the following way. Suppose x is an image containing a dog and h classifies it correctly, the adversary perturbs x into a new instance x' which is very close to x, the adversary wins if h classifies x' as cat. Note that a small perturbation of x is unlikely to change the visual information in it but the classifier fails to classify it correctly. It has been shown that several classification algorithms, which are otherwise efficient may

be sensitive to small perturbations and may succumb to the adversarial attack described above [1, 7]. In real world setting there can be scenarios where this adversarial behaviour can render a classifier useless, for example, consider senders of junk email disguising their emails by adding small amount of extra words which forces the classifier to classify as legitimate email.

A learning algorithm is called a *robust learner* if it produces a classifier that still predicts the true label of a input instance x if it is perturbed to a close instance x'. In the last decade there have been intense activity in addressing this problem: both for strategies to design classifiers to thwart such attacks [9] and new attacks [4].

Till date design of robust classifiers for real life problems has proved to be a very difficult task and in the existing race between designers and attackers it seems that the attackers still have a upper hand. Thus an important problem of today is to construct classifiers which are provably robust. Among many directions that are being explored towards solving this problem one significant one asks the question of the relation between computational power and robustness. A class of research shows evidence that constructing a robust classifier from limited training data is information theoretically possible but computationally intractable [3].

Most modern cryptographic constructions have the property that they are secure against computationally bounded adversaries but insecure against information theoretic (computationally unbounded) adversaries. Thus, it is likely that cryptographic objects may help in construction of robust classifiers and there are a line of work which explores this idea [5, 2, 6]. We explore further in this direction.

Our Contribution: In [6] the authors study the problem of robust classification in a fresh way. Their main contribution is to see the robustness of a classifier against the computational power of an adversary. In particular they claim that there exists a classification problem P which has a classifier h_P which is only robust against computationally bounded adversaries but not robust against computationally unbounded adversaries. They use a one-time signature scheme, and error correcting codes in their construction in a novel way. The security definition and the security results that they prove are in the asymptotic setting.

Firstly, we analyze in detail the construction in [6]. Using their basic idea we come up with two different constructions of robust classification problems. Instead of the asymptotic setting we use the concrete security setting and give concrete security bounds for the adversarial risk of our classifiers.

1. In our first construction we use a hash function. For proving security we treat the hash function as a random oracle. Though unrealistic, the random oracle model is a well accepted heuristic in the provable security literature.

2. Our second construction uses block ciphers. We introduce a new (non-standard) assumption regarding the block cipher which we call the Black-Box-non-Reconstructible (BBnR) assumption. Which essentially says that it is not possible for a computationally bounded adversary to reconstruct the description of a block-cipher from its input-output pairs. This assumption can be seen as a stronger version of the assumption that key recovery is difficult from input output pairs. Note that for the BBnR assumption the description of the block cipher is kept hidden from the adversary. We prove our construction to be robust if the BBnR assumption holds.

2 Preliminaries

General Notations: $\{0,1\}^*$ is the set of all binary strings and $\{0,1\}^n$ denotes the set of all n-bit strings. For $x, y \in \{0,1\}^*$, $x||y$ denotes the concatenation of strings x and y. When X is a finite set, by $x \xleftarrow{\$} X$ we mean that x is a uniform random element of X.For $x \in \{0,1\}^n$, $\mathsf{wt}(x)$ is defined as the number of 1's in x. For $x, x' \in \{0,1\}^n$ the Hamming distance between x and x' is denoted by $\mathsf{HD}(x, x') = \mathsf{wt}(x \oplus x')$. If D is a probability distribution over a set X, then by $x \xleftarrow{D} X$ indicates that (x, y) is sampled from D where $x \in X$ and $y \in Y$.

Signature Scheme: A *signature Scheme* is a tuple of three probabilistic ploynomial-time algorithms $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ satisfying the following :

1. The *key-generation algorithm* Gen takes as input a security parameter 1^n and outputs a pair of keys (pk, sk). These are called public key and the private key, respectively. We assume for convenience that pk and sk each have length at least n, and that n can be determined from pk, sk.

2. The *signing algorithm Sign* takes as input a private key sk and a message $m \in \{0,1\}^*$. It outputs a signature σ, denoted as $\sigma \leftarrow \mathsf{Sign}_{sk}(m)$.

3. The deterministic *verification algorithm* Vrfy takes as input a public key pk, a message m, and a signature σ. It outputs a bit b, with $b = 1$ meaning *valid* and $b = 0$ meaning *invalid*. We write this as $b := \mathsf{Vrfy}_{pk}(m, \sigma)$.

Security of a signature scheme is defined by the following. We say a signature scheme is secure, if

1. **Completeness:** It is required that for every n, every (pk, sk) output by $\mathsf{Gen}(1^n)$, and every $m \in \{0,1\}^*$, it holds that,
$$\mathsf{Vrfy}_{pk}(m, \mathsf{Sign}_{sk}(m)) = 1.$$

2. **Unforgeability:** For every positive polynomial s, for every λ and every pair of algorithm (A_1, A_2), which are polynomial in $s(\lambda)$, the following probability is negligible in λ.
$$\begin{aligned} \Pr[&(sk, pk) \leftarrow Gen(1^\lambda); \\ &(m, st) \leftarrow A_1(1^\lambda, pk); \\ &\sigma \leftarrow \mathsf{Sign}_{sk}(m); \\ &(m', \sigma') \leftarrow A_2(1^\lambda, pk, st, m, \sigma); \\ &m \neq m' \wedge \mathsf{Vrfy}_{pk}(\sigma', m') = 1] \\ &\leq \mathsf{negl}(\lambda) \end{aligned} \tag{1}$$

Error Correcting Code: An error correction code with code rate α and error rate β consists of two algorithms *Encode* and *Decode* as follows.

- The encode algorithm *Encode* takes a binary string m and outputs a Boolean string c such that $|c| = |m|/\alpha$
- The decode algorithm *Decode* takes a binary string c and outputs either $\perp$ or a Boolean string m. For all $m \in \{0,1\}^*, c = Encode(m)$ and c' where $HD(c,c') \leq \beta \cdot |c|$, it holds that $Decode(c') = m$.

Hash Functions: A hash function $G : \cup_{i \leq M}\{0,1\}^i \rightarrow \{0,1\}^N$ maps arbitrary long strings to a fixed length string. Many cryptographic protocols uses hash functions. If a hash function G is well designed, it should be the case that the only efficient way to determine the value $G(x)$ for a given x is to actually evaluate the function G at the value x. This should remain true even if many other values $G(x_1), G(x_2)...$ have already been computed.

The *random oracle model* provides a mathematical model of an ideal hash function. In this model, a hash function $G : X \longrightarrow Y$ is considered to be a uniform random element of $F^{X,Y}$, where $F^{X,Y}$ denotes the set of all functions mapping X to Y, and we are only permitted oracle access to the function G. This means that we are not given a formula or an algorithm to compute values of the function G. Therefore the only way to compute the value $G(x)$ is to query the oracle. This can be thought of as looking up the value of $G(x)$ in a giant book of random numbers such that, for each possible x, there is a complete random value $G(x)$. As a consequence of the assumption made in the random oracle model, it is obvious that the following independence property holds:

Property 1 *Suppose that* $G \stackrel{\$}{\leftarrow} F^{X,Y}$ *, and let* $X_0 \subseteq X$*. Suppose that the values* $G(x)$ *have been determined (by querying an oracle for* G*) if and only if* $x \in X_0$*. Then* $\Pr[G(x) = y] = \frac{1}{|Y|}$ *for all* $x \in X_0$ *and all* $y \in Y$

Hoeffding's inequality: In probability theory Hoeffding's inequality provides an upper bound on the probability that the sum of bounded independent random variables deviates from its expected value by more than a certain amount.

Hoeffding's inequality is a generalization of Chernoff bound, which applies only to Bernoulli random variable.

Let $Z_1, Z_2, Z_3, ..., Z_n$ be independent bounded random variables with $Z_i \in [a,b]$ for all i where $-\infty < a \leq b < \infty$. Then

$$P\left(\frac{1}{n}\sum_{i=1}^{n}(Z_i - E[Z_i]) \geq t\right) \leq \exp\left(-\frac{2nt^2}{(b-a)^2}\right) \tag{2}$$

and

$$P\left(\frac{1}{n}\sum_{i=1}^{n}(Z_i - E[Z_i]) \leq -t\right) \leq \exp\left(-\frac{2nt^2}{(b-a)^2}\right) \tag{3}$$

So if $Z_1, Z_2, Z_3, ..., Z_n$ be independent bounded random variables with $Z_i \in \{0,1\}$, and $P(Z_i = 1) = p$ Then

$$P\left(\sum_{i=1}^{n} Z_i \leq (p-t)n\right) \leq \exp\left(-2nt^2\right) \tag{4}$$

One-way Functions :A one-way function f has the property that it is easy to compute, but hard to invert.

Let $f : \{0,1\}^* \to \{0,1\}^*$ be a function. Consider the following experiment defined for any algorithm Alg and any value n for the security parameter.

The inverting experiment: $\mathsf{Invert}_{\mathsf{Alg},f}(n)$

1. Choose input $x \xleftarrow{\$} \{0,1\}^n$. Compute $y := f(x)$.
2. Alg is given 1^n and y as input, and outputs x'
3. The outout of the experiment is defined to be 1 if $f(x') = y$ and 0 otherwise.

Alg need not to find x itself but it suffices for Alg to find any value x' for which $f(x') = y = f(x)$.

A function $f : \{0,1\}^* \to \{0,1\}^*$ is one-way if the following two conditions hold:

1. (Easy to Compute) There exists a polynomial time algorithm M_f computing f; that is $M_f(x) = f(x)$ for all x
2. (Hard to invert) For every probabilistic polynomial time algorithm Alg, there exists a negigible function negl such that
$$\Pr[\mathsf{Invert}_{\mathsf{Alg},f}(n) = 1] \leq \mathsf{negl}(n)$$

3 Robust Classification

A classification problem is defined by $P = (X, Y, D, H)$ where set X is the set of possible instances, Y is the set of possible labels, D is a joint distribution over $X \times Y$, and H is the space of hypothesis. Generally, the hypothesis class contains functions $h : X \to Y$. For our purpose, we assume that each $h \in H$ is a function $h : X \to Y \cup \{*\}$, where $*$ is a special symbol not in Y. This extension of the hypothesis functions was done in [6] for providing ability to a classifier to detect tampering or "out of distribution" instances. In all schemes described here this modified definition of a hypothesis will play an important role. A learning algorithm is an algorithm which outputs a hypothesis $h \in H$ given the training instances set denoted by S.

For a classification problem P, the *risk* or *error* of a hypothesis h is defined as

$$\mathsf{Risk}_P(h) = \Pr_{(x,y) \xleftarrow{D} X \times Y}[h(x) \neq y].$$

For the purpose of defining adversarial perturbations we associate a metric d over the set of instances X. For a metric d over X, we let $\mathbf{d}_b(x) = \{x' | d(x, x') \leq b\}$ be the ball of radius b centered at x under metric d. For the rest of the document we will consider the instance space to be a finite set of binary strings of a fixed length and we will associate the Hamming distance as the metric with X.

In cryptography game based security definitions are common. A game based definition, in general, consists of interactions between two entities, a challenger and an adversary and an adversarial goal is specified. In [6] a game based definition of robust learning was provided which we state next.

Definition 1 (Security game of adversarial robust learning) *Let $P_n = (X_n, Y_n, D_n, H_n)$ be a classification problem where the components are parameterized by n. Let L be a learning algorithm with sample complexity $m = m(n)$ for P_n. Consider the following game between Challenger* Chal *and an adversary A with tampering budget $b = b(n)$.*

1. Chal *samples m i.i.d examples $S \longleftarrow D_n^m$ and gets hypothesis $h \longleftarrow L(S)$ where $h \in H_n$*
2. Chal *then samples a test example $(x, y) \leftarrow D_n$ and sends (x, y) to the adversary A.*
3. *Having oracle access to hypothesis h and a sampler for D_n, the adversary obtains the adversarial instance $x' \longleftarrow A^{h(.), D_n}(x)$ and outputs x'.*

Winning conditions :

1. *In case $x = x'$ the adversary A wins if $h(x) \neq y$.*
2. *In case $x \neq x'$, the adversary wins if all the following hold:*
   (a) $HD(x, x') \leq b$,
   (b) $h(x') \neq y$
   (c) $h(x') \neq *$

Definition 2 (Adversarial Risk of hypotheses and learners:) *Suppose L is a learning algorithm for a problem $P = (X, Y, D, H)$. For a class of attackers $\mathcal{A}$ we define*

$$\mathbf{AdvRisk}_{P,\mathcal{A}}(L) = \sup_{A \in \mathcal{A}} \Pr[A \text{ wins }],$$

where the winning is defined as in Definition 1.

A class of attackers will be generally specified by its running time t /circuit size s, and its perturbation budget b. For a class of attackers with running time t and perturbation budget b we will specify the adversarial risk of L by $\mathbf{AdvRisk}_{P,b,t}(L)$, similarly for a class of attackers with circuit size s and perturbation budget b we will specify the adversarial risk by $\mathbf{AdvRisk}_{P,b,t}(L)$. For computationally unbounded or information theoretic attackers, with no bound in running time but with a perturbation budget b, we specify the adversarial risk by $\mathbf{AdvRisk}_{P,b}(L)$.

Definition 3 (Computationally Robust learners :) *Let $P_n = (X, Y, D, H)$ be a classification problem parameterized by n. A learning algorithm L is a computationally robust learner with risk at most $R = R(n)$ against $b = b(n)$ perturbing adversaries, if for any polynomial $s = s(n)$, there is a negligible function $negl(n) = n^{-\omega(1)}$ such that*

$$AdvRisk_{P_n,b,s}(L) \leq R(n)\text{negl}(n)$$

The above definitions are all restated as in [6]. These definitions are asymptotic in nature and are parameterized by n as is common in cryptographic definitions. One can state the *concrete* version of these definitions where the classification problem is a fixed problem $P(X, Y, D, H)$ without any parameterizations. The concrete version of Definition 3 will be as

Definition 4 ((ϵ, t, b)-robust learner) *Let $P = (X, Y, D, H)$ be a classification problem. A learning algorithm L, associated with P is (ϵ, t, b) robust if the adversarial risk of L against all b perturbing adversaries, which runs for time at most t, is at most ϵ.*

4 A Computationally Robust Classification Problem

In [6] the authors present the construction of a classification problem along with its classifier and claim it to be robust against computationally bounded adversaries but not so against computationally unbounded adversaries. We first describe their construction and their proof.

The construction: Let $Q = (\{0,1\}^d, Y, D, H)$ be a learning problem and $h \in H$ be a classifier for Q such that $\mathsf{Risk}_Q(h) = \alpha$. Construct a family of learning problems P_n (based on the fixed problem Q) with a family of classifiers h_n. In this construction the following objects would be used:

1. A signature scheme $\Psi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ where the bit length of the public key pk is λ and the bit length of the signature is $\ell(\lambda) = \text{polylog}(\lambda)$.
2. An error correction code $(\mathsf{Encode}, \mathsf{Decode})$ with code rate $cr = \Omega(1)$ and error rate $er = \Omega(1)$.

The new problem $P_n = (X_n, Y_n, D_n, H_n)$ is defined as follows:

1. The space of instances for P_n is $X_n = \{0,1\}^{n+d+l(n)}$
2. The set of labels $Y_n = Y$
3. The distribution D_n is defined by the following process:
   - $(x, y) \leftarrow D$
   - $(pk, sk) \leftarrow \mathsf{Gen}(1^{n \cdot cr})$
   - $\sigma \leftarrow \mathsf{Sign}(sk, x)$

- $[pk] \leftarrow \mathsf{Encode}(pk)$
- **output** $((x, \sigma, [pk]), y)$

4. : The classifier $h_n : X_n \to Y_n$ is defined as

$$h_n(x, \sigma, [pk]) = \begin{cases} h(x) & \text{if } \mathsf{Vrfy}(\mathsf{Decode}([vk], x, \sigma) \\ * & \text{otherwise.} \end{cases}$$

Theorem 1 *For family P_n of construction above the family of classifiers h_n is computationally robust with risk at most α against adversaries with budget $er \cdot n$ (er is the error rate of the error correction code). On the other hand h_n is not robust against information theoretic adversaries of budget $b+\ell(n)$, if h is itself not robust to b perturbations:*

$$AdvRisk_{P_n, b+l(n)}(h_n) \geq AdvRisk_{Q,b}(h) \tag{5}$$

Proof:

Claim: For problem P_n,

$$Risk_{P_n}(h_n) = Risk_Q(h) = \alpha$$

$$\begin{aligned} Risk_{P_n}(h_n) &= \Pr[((x, \sigma, [vk]), y) \leftarrow D_n; h_n(x, \sigma, [vk]) \neq y] \\ &= \Pr[(x, y) \leftarrow D; h(x) \neq y] \\ &= Risk_Q(h)[Proved] \end{aligned}$$

Claim: For family P_n,and for any polynomial $s(\cdot)$ there is a negligible function negl such that for all $n \in \mathbb{N}$

$$AdvRisk_{P_n, er \cdot n, s}(h_n) \leq \alpha + negl(n) \tag{6}$$

Proof: Let $A_{\{n\in\mathbb{N}\}}$ be the family of circuits maximizing the adversarial risk for h_n for all $n \in \mathbb{N}$. We build a sequence of circuits $A^1_{\{n\in\mathbb{N}\}}$,$A^2_{\{n\in\mathbb{N}\}}$ such that A^1,A^2 are of size at most $s(n) + poly(n)$. A^1_n just samples a random $(x, y) \leftarrow D$ and outputs (x, y). A^2_n gets x, σ and vk, calls A_n to get $(x', \sigma', vk') \leftarrow A_n((x, \sigma, [vk]), y)$ and outputs (x', σ'). Note that A^2_n can provide alll the oracles needed to run A_n if the sampler from D, h and c are all computable by a circuit of polynomial size. Otherwise, we need to assume that our signature scheme is secure with respect to those oracles and the proof will follow. We have,

$AdvRisk_{P_n, er \cdot n, s}(h_n) = \Pr[((x, \sigma, [vk]), y) \leftarrow D_n; (x', \sigma', vk') \leftarrow A((x, \sigma, [vk]), y);$
$\mathrm{h}(x', \sigma', vk') \in HD_{er \cdot n}(x, \sigma, [vk]) \wedge h_n(x', \sigma', vk') \neq * \wedge h_n(x', \sigma', vk') \neq y]$

Note that $(x', \sigma', vk') \in HD_{er \cdot n}(x, \sigma, [vk])$ inplies that $Decode(vk') = vk$ based on the error rate of the error correcting code. Also $h_n(x', \sigma', vk') \neq *$implies that σ' is a valid signature for x' under verification key vk . Therefore we have,

$AdvRisk_{er\cdot n,s}(h_n)$
$\leq \Pr[(sk, vk) \leftarrow KeyGen(1^n; (x, y) \leftarrow A_1(1^n); \sigma \leftarrow Sign(sk, x); (x', \sigma') \leftarrow A_2(x, \sigma, vk); \text{Verify}(vk, x', \sigma') \wedge h_n(x', \sigma', [vk]) \neq y]$
$\leq \Pr[(sk, vk) \leftarrow KeyGen(1^n; x \leftarrow A_1(1^n); \sigma \leftarrow Sign(sk, x); (x', \sigma') \leftarrow A_2(x, \sigma, vk); \text{Verify}(vk, x', \sigma') \wedge x' \neq x] + Risk_{P_n}(h_n)$

Thus, by unforgeability of the one-time signature scheme we have

$$AdvRisk_{P_n,er\cdot n,s}(h_n) \leq Risk_{P_n}(h_n) + negl(n)$$

which by the above claim implies

$$AdvRisk_{P_n,er\cdot n,s}(h_n) \leq \alpha + negl(n)$$

4.1 Analysis of the construction

It is claimed that the construction is robust against computationally bounded adversaries with linear budget but is not so against information theoretic adversaries with even polylog budget. We analyze these situations in details in light of their construction and the proof.

The goal of the adversary is to produce a new instance $(x', \sigma', [pk]')$ which is close to the instance $(x, \sigma, [pk])$ provided by the challenger. To be successful it is at least required that the new instance is considered as a valid one by the classifier, i.e,

$$\mathsf{Vrfy}(\mathsf{Decode}([pk]'), x', \sigma') = 1.$$

As per the construction, the lengths of the signature σ and the encoded public key $[pk]$ are only dependent on the security parameter n, and length of x is a constant d. Thus the budget, which is also parameterized on n does not provide any restriction on the change of x, i.e., the whole of x could be changed to get a x' without violating the budget.

As specified by the construction, the length of pk is n. pk is the encoded using an error correcting code with code rate cr to obtain $[pk]$, thus the length of $[pk]$ is $cr \cdot n$. As the error rate of the error correcting code is er, thus to change $[pk]$ to $[pk]'$ such that $Decode([pk]') \neq [pk]$ the adversary needs to change more than $er.n$ many bits of $[pk]$, which is beyond the budget of the adversary. Thus any change that that the adversary makes on $[pk]$ within its budget will essentially not change it upon decoding, thus the adversarial strategy would be not to change $[pk]$ at all.

Keeping x fixed, it is not possible for the adversary to change σ, thus it will always be the case that $x \neq x'$. To realize such a change and still be successful the adversary has to forge a signature of x' under the public key $[pk]$. A curious feature of the construction is that the signature length is polylog(n), thus the whole signature can be changed within the perturbation budget.

The main argument in the proof is that a computational adversary will be unable to produce a valid (x', σ') pair as the signature scheme is a secure one. In particular, the success probability of the adversary in forging would be a negligible function in n. Though the argument is valid, but a

signature with length $\ell(n) = \text{polylog}(n)$ is a weak signature. The number of possible signatures of length $\ell(n)$ for a message m is at most $2^{\ell(n)}$, thus the adversary can trivially forge with probability $2^{-\ell(n)} = n^{-\text{polylog}(n)}$. In other words, the adversary will require quasi polynomial time to brute force search over all possible signatures. If $\ell(n) = O(n)$ was used instead then the signature would have been a stronger one, with the adversary requiring exponential time to brute force search over all signatures. Also, most signature schemes in use have signature lengths of $O(n)$.

The authors of [6] do not use a linear length signature to accommodate their strong claim regarding information theoretic adversaries. A computationally unbounded adversary can forge the signature of any length but to allow a lesser perturbation budget of polylog(n), they use a polylog(n) sized signature.

5 Construction with Hash function

Here we provide a new construction which instead of a signature scheme uses a hash function

$$G : \{0,1\}^d \to \{0,1\}^N$$

. **The Construction:** *Let $P = (X, Y, D, H)$ be a learning problem and $h \in H$ be a classifier for P. Construct a family of learning problems P' (based on the fixed problem P) with a family of classifiers h'. In this construction random oracle model is used. Let $G : \{0,1\}^d \to \{0,1\}^N$ be a hash function modelled as a random oracle. First $(x, y) \leftarrow D$ is sampled, then after getting $G(x)$ we use $(x, G(x), y)$ as the input instances to the learning algorithm of P'.*

1. The space of instances for P' is $X' = \{0,1\}^{d+N}$
2. The set of labels $Y' = Y$
3. The distribution D' is defined by the following process: $(x, y) \leftarrow D$, compute $z = G(x)$, output $((x, z), y)$
4. The classifier $h' : X' \to Y'$ is defined as

$$h'(x, z, y) = \begin{cases} h(x) & \text{if } G(x) = z \\ * & \text{otherwise.} \end{cases}$$

Theorem 2 *For a family P' of construction above, the family of classifiers h' is ϵ robust against adversaries with budget K, where $\epsilon = \exp\left\{-\left(\frac{N}{2} + \frac{2K^2}{N}\right)\right\}$ and $K \leq \frac{N}{2}$.*

To prove this we need the following lemma.

Lemma 1: Let $G : \{0,1\}^d \to \{0,1\}^N$be a random oracle, Then for any $x, x' \in \{0,1\}^d$,

$$\Pr[HD(G(x), G(x')) \leq K] \leq \exp\left(-2N\left(\frac{1}{2} - \frac{K}{N}\right)^2\right)$$

Proof: $G(x) \in \{0,1\}^N$ so for two instances of input say x_1, x_2, we are trying to find

$$Pr\left(HD(G(x_1), G(x_2)) \leq K\right)$$

Note as G is modelled as a random oracle, thus $G(x_1), G(x_2)$ are two uniform random elements of $\{0,1\}^N$. Let $D_i \in \{0,1\}$ be a random variable where,

$$D_i = \begin{cases} 1 & \text{if } (G(x_1))_i = (G(x_2))_i \\ 0 & \text{otherwise.} \end{cases}$$

So we have,

$$HD(G(x_1), G(x_2)) = \sum_{i=1}^{N} D_i$$
$$\text{and } \Pr\left(D_i = 1\right) = \frac{1}{2}$$

Using Hoeffding bound[equation (3)], we can write

$$\Pr\left(\sum_{i=1}^{N} D_i \leq (\frac{1}{2} - t)N\right) \leq \exp\left(-2Nt^2\right) \tag{7}$$

Now, putting $t = (\frac{1}{2}) - (\frac{K}{N})$ gives,

$$\Pr\left(\sum_{i=1}^{N} D_i \leq K\right) \quad \leq \quad \exp\left(-2N\left(\frac{1}{2} - \frac{K}{N}\right)^2\right)$$

Hence,

$$\Pr\left(HD(G(x_1), G(x_2)) \leq K\right) \leq \exp\left(-2N\left(\frac{1}{2} - \frac{K}{N}\right)^2\right)$$

□

Proposition 1 *Let $x, x' \in \{0,1\}^d$ be such that $HD(x,x') = b$ and $b \leq K$. Let $z = G(x)$ and $z' = G(x')$, then* $\Pr[HD(x||z, x'||z') \leq K] \leq \exp\left(-2N\left(\frac{1}{2} - \frac{K}{N}\right)^2\right)$.

Proof: When the adversary budget is K, and $HD(x,x') = b$, then $HD(G(x), G(x')) \leq K - b \leq K$. So,

$$\begin{aligned}
\Pr[HD(x||z, x'||z') \leq K] &= \Pr[HD(z,z') \leq K - b] \\
&\leq \Pr[HD(G(x), G(x')) \leq K - b] \\
&\leq \Pr[HD(G(x), G(x')) \leq K] \\
&\leq \exp\left(-2N\left(\frac{1}{2} - \frac{K}{N}\right)^2\right).
\end{aligned}$$

□

Now,

$$\begin{aligned}
& \mathsf{AdvRisk}_{P,b}(h) \\
= \ & \Pr[\text{Adversary finds a } x' \text{ such that } HD((x||G(x)), (x', G(x'))) \leq K] \\
\leq \ & \Pr[\exists x' \text{ such that } HD((x||G(x)), (x', G(x'))) \leq K] \\
\leq \ & \exp\left(-2N\left(\frac{1}{2} - \frac{K}{N}\right)^2\right)
\end{aligned}$$

Theorem 2 follows directly from Lemma 1 and Proposition 1 So the adversarial risk is at most $\epsilon = \exp\left(-2N\left(\frac{1}{2} - \frac{K}{N}\right)^2\right)$ for the construction mentioned above. □

With some simplification it can be shown that, for the scheme to be ϵ robust, where $\epsilon = \exp\left(-2N\left(\frac{1}{2} - \frac{K}{N}\right)^2\right)$, the adversarial budget K can be at most $N\left(\frac{1}{2} - \sqrt{\frac{\log\frac{1}{\epsilon}}{2N}}\right)$

6 A Block Cipher based construction

A bock cipher is a function $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$, where for every $K \in \{0,1\}^k$, $E_K : \{0,1\}^n \to \{0,1\}^n$ is a bijection. For convenience, we will assume throughout that $k = n$. For $K, m \in \{0,1\}^n$ we will sometimes write $E_K(m)$ instead of $E(K,m)$.

We make a nonstandard assumption regarding the function E, which essentially says that no efficient adversary can reverse engineer and construct E given input output examples.

Definition 5 Black-Box not Reconstructible: *An adversary A is given oracle access to E, and it can query it multiple times. Suppose the adversary asks q queries where the i^{th} query is $(K^{(i)}, x^{(i)})$*

and the response is $z^{(i)} = E(K^{(i)}, x^{(i)})$. *Let* $Q = \{(K^{(i)}, x^{(i)}) : 1 \leq i \leq q\}$. *Finally the* A *outputs* $(\tilde{K}, \tilde{x}, \tilde{z})$. *We say* A *is successful if* $(\tilde{K}, \tilde{x}) \notin Q$ *and* $E(\tilde{K}, \tilde{x}) = \tilde{z}$.

We say that E *is* (ϵ, t, q)*-black-box not reconstructible (BBnR) if for all adversaries who run for time at most* t*, asks at most* q *queries to its oracle have its success probability at most* ϵ.

Note that the BBnR assumption can be seen as a weaker version of the fact that key recovery is difficult, i.e. given $\{x_i, y_i\}_{i=1}^{q}$ such that $y_i = E_K(x_i)$, it is difficult to find K. For the BBnR assumption we even hide the description of E from the adversary but we find no way to reduce this property from existing standard assumptions on block ciphers.

Now we describe a construction for a robust classification problem using E

6.1 The construction

As before let $C = (X, Y, D, H)$ be a classification problem, where $X = \{0,1\}^n$, D is a probability distribution over $X \times Y$ and H consists of hypotheses where each $h \in H$ is $h : X \to Y \cup \{*\}$. We convert Q to a classification problem $P = (X', Y, D', H')$, where

1. $X' = \{0,1\}^{3n}$
2. The new distribution on $X' \times Y$ is defined as:

   $(x, y) \xleftarrow{D} X \times Y$

   $K \xleftarrow{\$} \{0,1\}^n$

   $z = E(K, x)$

   output (x, z, K, y)
3. The new hypothesis h' is defined as

$$h'(x, z, K) = \begin{cases} h(x) & \text{if } E_K(x) = z \\ * & \text{otherwise.} \end{cases}$$

Theorem 3 *The classification problem* P *defined above is robust. In particular for an arbitary adversary* B *against robustness of* P *with budget* b *and running time* t*, there exists a BBnR adversary* A *for the block cipher* $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$*, which asks at most* q *query to its oracle such that,*

$$\mathbf{AdvRisk}_P(h) \leq \mathbf{Adv}_E^{bbnr}(A) + q \exp\left(-6n\left(\frac{1}{2} - \frac{b}{3n}\right)^2\right)$$

Let A be an adversary to BBnR. It has the oracle access and it can ask q queries to the oracle.

Let B be an arbitrary adversary. We will construct the adversary A, which will use B and thus will also provide the environment for B. Specifically A would act as the challenger for B and also provides it with the required oracle.

Adversary B's goal is to output (x', z', K') given (x, z, K) such that $HD((x||z||K), (x||z'||K')) \leq b$, where b is the adversarial budget.

1. On a sample query from B, A sends (x, z, K, y) as defined above to B.
2. On a classifier query from B, A sends $h(x)$ or $*$ as mentioned above to B.
3. A outputs what B outputs

Let (x', z', K') is the output from B.
Let us define the following:

$E_1 \equiv E_{K'}(x') = z'$
$E_2 \equiv HD((x||z||K), (x||z'||K')) \leq b$
$E_3 \equiv HD((x||z||K), (x^i||z^i||K^i)) \geq b$, where $(x^i||z^i||K^i) \in Q$ and Q is the query set as defined above
$E_4 \equiv (x||z'||K') \notin Q$

Let A_W defines the event that adversary A wins, similarly Let B_W defines the event that adversary B wins. $\Pr[A_W] = \Pr[E_1 \wedge E_2]$ and $\Pr[B_W] = \Pr[E_1 \wedge E_4]$
Also, $\Pr[A_W] = \mathbf{Adv}_E^{bbnr}(A)$
Also, $\Pr[A_W \wedge E_3] \geq \Pr[B_W \wedge E_3]$
So,

$$\begin{aligned}
\Pr[B_W] - \Pr[A_W] &= \Pr[B_W \wedge E_3] + \Pr[B_W \wedge \overline{E_3}] - \Pr[A_W \wedge E_3] + \Pr[A_W \wedge \overline{E_3}] \\
&\leq \Pr[B_W \wedge \overline{E_3}] - \Pr[A_W \wedge \overline{E_3}] \\
&= \Pr[B_W | \overline{E_3}] \Pr[\overline{E_3}] - \Pr[A_W | \overline{E_3}] \Pr[\overline{E_3}] \\
&= \Pr[\overline{E_3}](\Pr[B_W | \overline{E_3}] - \Pr[A_W | \overline{E_3}]) \\
&\leq \Pr[\overline{E_3}]
\end{aligned}$$

$$
\begin{aligned}
&\Pr[\overline{E_3}] \\
&= \Pr[HD((x||z||K),(x^i||z^i||K^i)) < b] \text{ such that } \exists (x^i||z^i||K^i) \in Q \\
&= \Pr[E_i], \text{ such that } \exists i \text{ and } (x^i||z^i||K^i) \in Q \text{ where } E_i \equiv HD((x||z||K),(x^i||z^i||K^i)) < b \\
&= \Pr\left[\bigcup_{i=1}^{q} E_i\right] \\
&\leq \bigcup_{i=1}^{q} \Pr[E_i] \\
&\leq q \exp\left(-6N\left(\frac{1}{2} - \frac{b}{3N}\right)^2\right) \text{ (Using Lemma 1)}
\end{aligned}
$$

So,

$$
\begin{aligned}
\mathbf{AdvRisk}_P(h) &= \Pr[B_W] \\
&\leq \Pr[A_W] + \Pr[\overline{E_3}] \\
&\leq \mathbf{Adv}_E^{bbnr}(A) + q \exp\left(-6N\left(\frac{1}{2} - \frac{b}{3N}\right)^2\right)
\end{aligned}
$$

So the adversarial risk is at most $\epsilon = \mathbf{Adv}_E^{bbnr}(A) + q \exp\left(-6N\left(\frac{1}{2} - \frac{b}{3N}\right)^2\right)$ for the construction mentioned above. □

References

[1] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Zelezný, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III*, volume 8190 of *Lecture Notes in Computer Science*, pages 387–402. Springer, 2013.

[2] Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya P. Razenshteyn. Adversarial examples from cryptographic pseudo-random generators. *CoRR*, abs/1811.06418, 2018.

[3] Sébastien Bubeck, Yin Tat Lee, Eric Price, and Ilya P. Razenshteyn. Adversarial examples from computational constraints. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 831–840. PMLR, 2019.

[4] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57. IEEE Computer Society, 2017.

[5] Akshay Degwekar, Preetum Nakkiran, and Vinod Vaikuntanathan. Computational limitations in robust classification and win-win results. In Alina Beygelzimer and Daniel Hsu, editors, *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pages 994–1028. PMLR, 2019.

[6] Sanjam Garg, Somesh Jha, Saeed Mahloujifar, and Mohammad Mahmoody. Adversarially robust learning could leverage computational hardness. *CoRR*, abs/1905.11564, 2019.

[7] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[8] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of Statistical Learning*. Springer, 2009.

[9] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[10] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning, From theory to algiorithms*. Cambidge University Press, 2014.