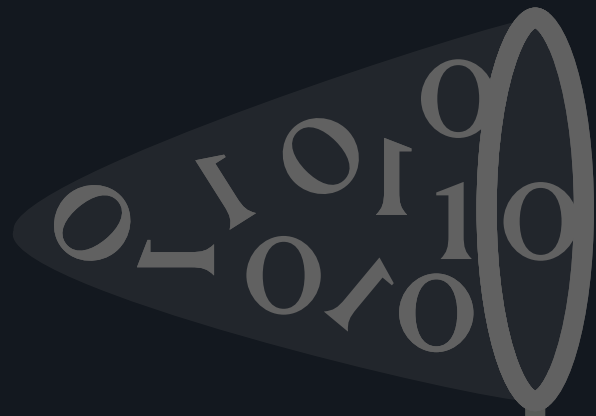


Communicating Research

Doreen Riepel and Thorsten Eisenhofer



10110001110100100011110000111100101100011101001000

CASA

DFG Cluster of Excellence

Cyber Security in the Age of Large-Scale Adversaries



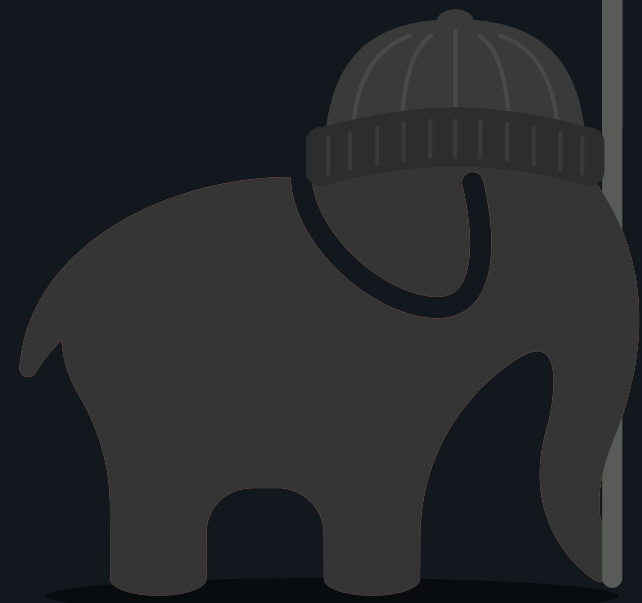
Doreen Riepel

Last year doctoral student
Theoretical Cryptography



Thorsten Eisenhofer

Last year doctoral student
ML & Computer Security



LARGE-SCALE
ADVERSARY

RUHR
UNIVERSITÄT
BOCHUM

RUB

Communicating Research

Beyond academia

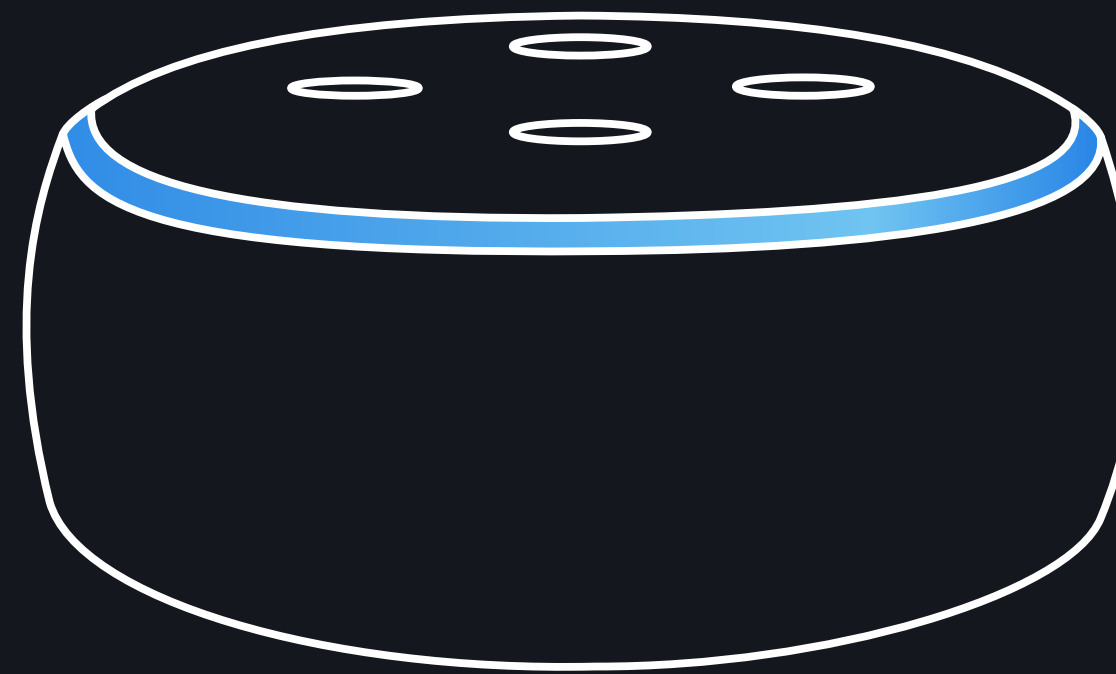
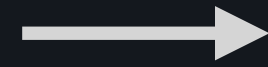
Interdisciplinary communities

Beyond academia

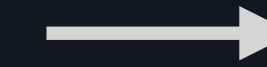
Project about the security of voice assistants



Raw Audio Wave



Voice Assistant



I SOLEMNLY
SWEAR THAT I AM
UP TO NO GOOD

Transcription

Beyond academia

Different abstraction levels

Reduced complexity

Visualization vs. technical details

DOMPTEUR: Taming Audio Adversarial Examples

Thorsten Eisenhofer *Ruhr University Bochum* Lea Schönherr *Ruhr University Bochum* Joel Frank *Ruhr University Bochum*
 Lars Speckemeier *University College London* Dorothea Kolossa *Ruhr University Bochum* Thorsten Holz *Ruhr University Bochum*

Abstract

Adversarial examples seem to be inevitable. These specifically crafted inputs allow attackers to arbitrarily manipulate machine learning systems. Even worse, they often seem harmless to human observers. In our digital society, this poses a significant threat. For example, *Automatic Speech Recognition* (ASR) systems, which serve as hands-free interfaces to many kinds of systems, can be attacked with inputs incomprehensible for human listeners. The research community has unsuccessfully tried several approaches to tackle this problem.

In this paper we propose a different perspective: We accept the presence of adversarial examples against ASR systems, but we require them to be *perceivable* by human listeners. By applying the principles of *psychoacoustics*, we can remove semantically irrelevant information from the ASR input and train a model that resembles human perception more closely. We implement our idea in a tool named *DOMPTEUR*¹ and demonstrate that our augmented system, in contrast to an unmodified baseline, successfully focuses on perceptible ranges of the audible range, while using minimal computational overhead and preserving benign performance. To evaluate our approach, we construct an *adaptive attacker* that actively tries to avoid our augmentations and demonstrate that adversarial examples from this attacker remain clearly perceivable. Finally, we substantiate our claims by performing a hearing test with crowd-sourced human listeners.

1 Introduction

The advent of deep learning has changed our digital society. Starting from simple recommendation techniques [1] or image recognition applications [2], machine-learning systems have evolved to solve and play games on par with humans [3–6], to predict protein structures [7], identify faces [8], or recognize speech at the level of human listeners [9]. These systems are now virtually ubiquitous and are being granted access to

¹The French word for *tamer*

critical and sensitive parts of our daily lives. They serve as our personal assistants [10], unlock our smart homes’ doors [11], or drive our autonomous cars [12].

Given these circumstances, the discovery of *adversarial examples* [13] has had a shattering impact. These specifically crafted inputs can completely mislead machine learning-based systems. Mainly studied for image recognition [13], in this work, we study how adversarial examples can affect *Automatic Speech Recognition* (ASR) systems. Preliminary research has already transferred adversarial attacks to the audio domain [14–19]. The most advanced attacks start from a harmless input signal and change the model’s prediction towards a target transcription while simultaneously *hiding* their malicious intent in the inaudible audio spectrum.

To address such attacks, the research community has developed various defense mechanisms [20–25]. All of the proposed defenses—in the ever-lasting cat-and-mouse game between attackers and defenders—have subsequently been broken [26]. Recently, Shamir et al. [27] even demonstrated that, given certain constraints, we can expect to always find adversarial examples for our models.

Considering these circumstances, we ask the following research question: *When we accept that adversarial examples exist, what else can we do?* We propose a paradigm shift: Instead of preventing *all* adversarial examples, we accept the presence of *some*, but we want them to be audibly changed.

To achieve this shift, we take inspiration from the machine learning community, which sheds a different light on adversarial examples: Illyas et al. [28] interpret the presence of adversarial examples as a disconnection between human expectations and the reality of a mathematical function trained to minimize an objective. We tend to think that machine learning models must learn meaningful features, e.g., a cat has paws. However, this is a human’s perspective on what makes a cat a cat. Machine learning systems instead use *any* available feature they can incorporate in their decision process. Consequently, Illyas et al. demonstrate that image classifiers utilize so-called *brittle features*, which are highly predictive, yet not recognizable by humans.

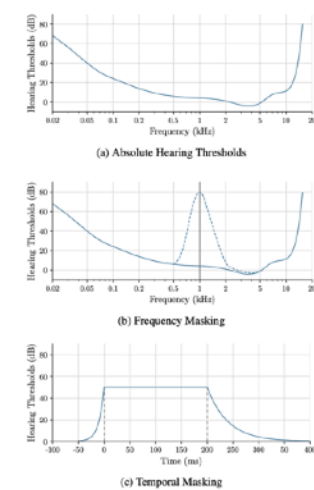


Figure 1: Psychoacoustic allows to describe limitations of the human auditory system. Figure 1a shows the average human hearing threshold in quiet. Figure 1b shows an example of masking, illustrating how a loud tone at 1kHz shifts the hearing thresholds of nearby frequencies and Figure 1c shows how the recovery time of the auditory system after processing a loud signal leads to temporal masking.

Psychoacoustic Modeling. Recent attacks against ASR systems exploit limitations of the human auditory system to create adversarial examples less conspicuous [17,30–41]. Specifically, these attacks utilize limitations of human perception to hide modifications of the input audio signal within inaudible ranges. We use the same effects for our approach to remove inaudible components from the input:

• **Absolute Hearing Thresholds.** Human listeners can only perceive sounds in a limited frequency range, which diminishes with age. Moreover, for each frequency, the sound pressure is important to determine whether the signal component is in the audible range for humans. Mean-

ing the *hearing thresholds*, i.e., the necessary sound pressures for each frequency to be audible to otherwise quiet environments, one can determine the so-called *absolute hearing threshold* as depicted in Figure 1a. Generally speaking, everything above the *absolute hearing threshold* is perceptible in principle by humans, which is not the case for the area under the curve. As can be seen, much more energy is required for a signal to be perceived at the lower and higher frequencies. Note that the described thresholds only hold for cases where no other sound is present.

• **Frequency Masking.** Like frequency masking, temporal masking is also caused by other sounds, but these sounds have the same frequency as the masked tone and are close to it in the time domain, as shown in Figure 1c. Its most cause lies in the fact that the auditory system needs a certain amount of time, in the range of a few hundreds of milliseconds, to recover after processing a higher energy sound event to be able to perceive a new, less energetic sound. Interestingly, this effect does not only occur at the end of a sound but also, although much less distinct, at the beginning of a sound. This seeming causal contradiction can be explained by the processing of the sound in the human auditory system.

• **Temporal Masking.** Like frequency masking, temporal masking is also caused by other sounds, but these sounds have the same frequency as the masked tone and are close to it in the time domain, as shown in Figure 1c. Its most cause lies in the fact that the auditory system needs a certain amount of time, in the range of a few hundreds of milliseconds, to recover after processing a higher energy sound event to be able to perceive a new, less energetic sound. Interestingly, this effect does not only occur at the end of a sound but also, although much less distinct, at the beginning of a sound. This seeming causal contradiction can be explained by the processing of the sound in the human auditory system.

Adversarial Examples. Since the seminal papers by Szegedy et al. [11] and Biggio et al. [13], a field of research has formed around adversarial examples. The basic idea is simple: An attacker starts with a valid input to a machine learning system. Then, they add small perturbations to that input with the ultimate goal of changing the resulting prediction (or in our case, the transcription of the ASR).

More formally, given a machine learning model f and an input prediction pair (x, y) , where $f(x) = y$, we want to find a small perturbation δ s.t.:

$$x' = x + \delta \wedge f(x') \neq f(x)$$

In this paper, we consider a stronger type of attack, a *targeted one*. This has two reasons: First, a targeted attack is in fact a more appealing tool than, for more threatening real-life use cases for adversarial examples. More formally, the attacker wants to perturb an input phrase x (i.e., an audio signal) with a transcription y (e.g., “the Beatles”) in such a way that the transcription

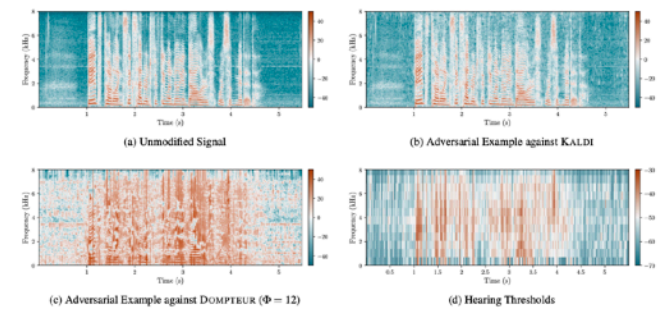


Figure 5: Spectrograms of adversarial examples. Figure 5a shows the unmodified signal. Figure 5b depicts the baseline with an adversarial example computed against KALIN with psychoacoustic hiding. Figure 5c an adversarial example computed with the adaptive attack against DOMPTEUR, and Figure 5d shows the computed hearing thresholds for the adversarial example.

These parameters are chosen such that an attacker needs to introduce ≈ 4.8 phones per second into the target audio, which Schönherr et al. suggests as both effective and efficiently possible [17]. Furthermore, we picked the utterances and target sentence to be easy for an attacker in order to deconvolve the influence on our analysis. Specifically, for these targets the baseline has a very high success rate and low SNRRng (cf. Table 2). Note that the attack is capable of introducing arbitrary target sentences up to a certain length. In Section 4.3.2, we further analyze the influence of the phone rate, and in particular, the influence of the target utterance and sentence on the SNRRng. We compute adversarial examples for different learning rates and a maximum of 2000 iterations. This number is sufficient for the attack to converge, as shown in Figure 4, where the WER is plotted as a function of the number of iterations.

Results. The main results are summarized in Table 2. We report the average SNRRng over all adversarial examples, the best SNRRng_{max}, and the number of successful adversarial examples created.

We evaluate the attack using different learning rates (0.05, 0.10, 0.5, and 1) in our experiments, we observed that while small learning rates generally produce less noisy adversarial examples, they simultaneously get more stuck in local optima. Thus, to simulate an attacker that would run an extensive search and uses the best result we also report the intersection of successful adversarial examples over all learning rates. If

success rate is the primary goal, we recommend a higher learning rate.

By increasing Φ , we can successfully force the attacker into audible ranges while also decreasing the attack’s success rate. When using very aggressive filtering ($\Phi = 14$), we can prevent the creation of adversarial examples completely, albeit with a hit on the benign WER (5.25% \rightarrow 7.93%). Note, however, that the performance of adversarial examples increases with respect to the baselines by up to 24.08 dB (third) and 21.32 dB (music). Equally, the attack’s general success decreases to SNRRng (and 350 (music) successful adversarial examples) based on audio files containing music and bird sounds. The results are presented in Table 3.

We can repeat our observations from the previous experiment. When we utilize a more aggressive filter, we observe that the performance of adversarial examples increases with respect to the baselines by up to 24.08 dB (third) and 21.32 dB (music). Equally, the attack’s general success decreases to SNRRng (and 350 (music) successful adversarial examples).

Note that the SNRRng for music samples are in general higher than that of speech and bird files as these samples have a more dynamic range of signal energy. Hence, potentially added adversarial perturbations have a smaller impact on the calculation of the SNRRng. The absolute amount of added perturbations, however, is similar to that of other content, when listening to the created adversarial examples’ samples are similarly distorted. This is further confirmed in Section 4.4 with our listening test.

4.3.1 Target Phone Rate

The success of the attack depends on the ratio between the length of the audio file and the length of the target text, which we refer to as the *target phone rate*. This ratio describes how

an attacker-chosen transcription y (e.g., “Unlock the front door”). This can be achieved by computing an adversarial example x' based on a small adversarial perturbation δ s.t.:

$$x' = x + \delta \wedge \text{ASR}(x') = y \wedge y \neq x'$$

There exist a multitude of techniques for creating such adversarial examples. We use the method introduced by Schönherr et al. [17] for our evaluation in Section 4. The method can be divided into three parts: In a first step, attackers choose a fixed output matrix of the DNN to maximize the probability of obtaining their desired transcription y . As introduced before, this matrix is used in the ASR system’s decoding step to obtain the final transcription. They then utilize gradient descent to perturb a starting input x , i.e., an audio signal fed into the DNN, to obtain a new input x' , which produces the desired matrix. This approach is generally chosen in white-box attacks [18, 19]. Note that we omit the feature extraction part of the ASR, however, Schönherr et al. have shown that this part can be integrated into the gradient step itself [17]. A third (optional) step is to utilize psychoacoustic hearing thresholds to restrict the added perturbations to inaudible frequency ranges. More technical details can be found in the original publication [17].

3 Modeling the Human Auditory System

We now motivate and explain our design to better align the ASR system with human perception. Our approach is based on the fact that the human auditory system only sees a subset of the information contained in an audio signal to form an understanding of its content. In contrast, ASR systems are not limited to specific input ranges and utilize every available signal—even those inaudible for the human auditory system. Consequently, an attacker can easily hide changes within these ranges. Intuitively, the smaller the changes between these two worlds, the harder it becomes for an attacker to add malicious perturbations that are inaudible to a human listener. This is akin to reducing the attack surface in traditional systems security.

To tackle these issues, we leverage the following two design principles in our approach:

- (i) **Removing inaudible parts:** As discussed in Section 2, audio signals typically carry information impossible to human listeners. Thus, before passing the input to the network, we utilize psychoacoustic modeling to remove these parts.
- (ii) **Restricting frequency access:** The human voice frequency range is limited to a band of approximately 300–3000 Hz [29]. Thus, we implement a band-pass filter between the feature extraction and model stage (cf. Section 2) to restrict the acoustic model to the appropriate frequency.

Table 3: Number of successful Adversarial Examples (AEs) and mean Segmental Signal-to-Noise (SNRRng) ratio for non-speech audio content. For each AE, we selected the least noxious example, from running the attack with learning rates (0.05, 0.1, 0.5, 1). For the SNRRng we only consider successful AEs and report the difference to the baseline (KALIN). We highlight the highest loss in bold.

	Bird		Music	
	AE	SNRRng (dB)	AE	SNRRng (dB)
wildbird	3000	11.81	4000	23.26
nr_singing	3200	17.28	14300	26.06
deutsche	3000	9.34	14300	30.21
$\Phi = 6$	3100	2.11	11300	16.81
$\Phi = 12$	500	-22.01	12400	1.94

Table 4: Attack for different cut-off frequencies of the band-pass filter. We report the number of successful adversarial examples (AEs) and the mean Segmental Signal-to-Noise (SNRRng) ratio. For the SNRRng we only consider successful AEs.

AE	deutsche		music		wildbird	
	AE	SNRRng	AE	SNRRng	AE	SNRRng
Baseline	3000	11.81	4000	23.26	3000	11.81
$\Phi = 6$	3100	2.11	11300	16.81	1100	1.22
$\Phi = 12$	500	-22.01	12400	1.94	100	-24.08

many phones an attacker can hide within one second of audio content.

In our experiments, we used the default rates recommended by Schönherr et al. However, a better rate might exist for our setting. Therefore, to evaluate the effect of the target phone rate, we sample target texts of varying lengths from the WSJ corpus and compute adversarial examples for different target phone rates. We pick phone rates ranging from 1 to 20 and run 20 attacks for each of them for at most 1000 iterations, resulting in 400 attacks.

The results in Figure 6 show that, in general, with increasing phone rates, the SNRRng decreases and stagnates for target phone rate beyond 12. This is expected as the attacker tries to hide more phones and, consequently, needs to change the signal more drastically. Thus, we conclude that the default settings are adequate for our setting.

4.3.2 Band-Pass Cut-off Frequencies

So far, we only considered a relatively wide band-pass filter (200–7000 Hz). We also want to investigate other cut-off frequencies. Thus, we disable the psychoacoustic filtering and compare adversarial examples for different models examined in Section 4.2. We run the attack for each band-pass model with 20 speech samples for at most 1000 iterations.

The results are reported in Table 4. We observe that the energy amount of adversarial perturbations remains relatively constant for different filters, which is expected since the attacker has complete knowledge of the system. As we narrow the frequency band, the attacker adapts and puts more perturbation within those bands.

Apart from the SNRRng, we also observe a decrease in the attack success, especially for small high cut-off frequencies, with only 11/20 (500–3000 Hz) and 12/20 (500–3000 Hz) successful adversarial examples.

3.1 Implementation

In the following, we present an overview of the implementation of our proposed augmentations. We extend the state-of-the-art ASR toolkit KALIN with our augmentations to build a prototype implementation called *DOMPTEUR*. Note that our proposed methods are universal and can be applied to any ASR system.

Psychoacoustic Filtering. Based on the psychoacoustic model of MPEG-1 [13], we use psychoacoustic hearing thresholds to remove parts of the audio that are not perceivable to humans. These thresholds define how dependencies between certain frequencies can mask, i.e., make inaudible, other parts of an audio signal. Intuitively, these parts of the signal should not contribute any information to the recognizer. They do, however, provide space for an attacker to hide adversarial noise.

We compare the absolute values of the complex valued short-time Fourier transform (STFT) representation of the audio signal S with the hearing thresholds H and define a mask M :

$$M(n, k) = \begin{cases} 1 & \text{if } |S(n, k)| \leq H(n, k) + \Phi \\ 0 & \text{else} \end{cases}$$

with $n = 0, \dots, N - 1$ and $k = 0, \dots, K - 1$. We use the parameter Φ to control the effect of the hearing thresholds. For $\Phi = 0$, we use the original hearing threshold, for higher values we use a more aggressive filtering, and for smaller values we use more noise from the original signal. We explore this in detail in Section 4. We then multiply all values of the signal S with the mask M :

$$T = S \odot M$$

to obtain the filtered signal T .

Band-Pass Filter. High and low frequencies are not part of human speech and do not contribute significant information. Yet, they can again provide space for an attacker to hide adversarial noise. For this reason, we remove low and high frequencies of the audio signal in the frequency domain. We apply a band-pass filter after the feature extraction of the system by discarding those frequencies that are smaller or larger than certain thresholds (the so-called cut-off frequencies). Formally, the filtering can be described via

$$T(f, k) = 0 \quad \forall f_{low} < k < f_{high}$$

where f_{low} and f_{high} describe the lower and the upper cut-off frequencies of the band-pass.

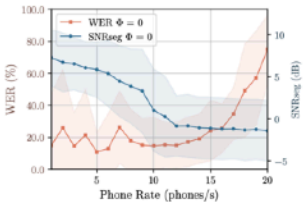


Figure 6: Word Error Rate (WER) and Segmental Signal-to-Noise (SNRRng) ratio for different phone rates. We report the mean and std. deviation for adversarial examples computed for targets with varying length.

Beyond academia

RESEARCH PAPER

NEWS ARTICLE

COMIC

Abstraction

CASA

RUHR-UNIVERSITÄT BOCHUM

RUB

RUBIN

WISSENSCHAFTSMAGAZIN

Schwerpunkt

VIRTUELLE WELTEN



PLUS
AUGMENTED
REALITY

EINFACH APP
»ZAPPAR«
INSTALLIEREN UND
CODES SCANNEN



PSYCHISCH KRANKE AVATARE
GEHEIME BOTSCHAFTEN FÜR ALEXA & CO.
KÜNSTLICHE UN-INTELLIGENZ



29
Jahrgang
Nr. 2 | 2019



https://news.rub.de/sites/default/files/rubin_2019_2_web.pdf

IT-Sicherheit

WIE SPRACH- ASSISTENTEN UNHÖRBARE BEFEHLE BEFOLGEN

Was für einen Menschen nach einem harmlosen Musikstück klingt, kann für eine Maschine die Anweisung sein, eine bestimmte Aktion auszuführen.

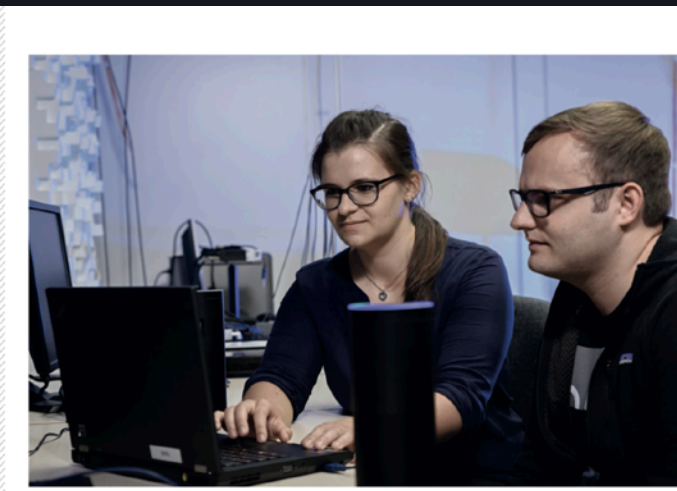
Vielleicht besser als zu den Anfängen der Spracherkennungssysteme versteht Alexa, Siri und Co. heute, was Menschen ihnen sagen. Manchmal verstehen sie sogar Dinge, die der Mensch nicht hören kann. Eine Sicherheitslücke, wie die IT-Spezialisten vom Bochumer Vernetz-Güter-Institut für IT-Sicherheit (VIGI) wissen, ihnen gelang es, beliebige Befehle für Sprachassistenten in unterschiedlichen Arten von Audiosignalen zu verstecken, zum Beispiel in Musik, Sprache oder Vogelgezwitscher. Solange diese Angriffe nur der Forschung dienen, passiert dabei nichts Schlimmes. Ein bösser Angriff könnte aber auf diese Weise über beispielsweise einen Song, der im Radio abgespielt wird, so manipulieren, dass er den Befehl enthält, ein bestimmtes Produkt zu kaufen oder Kontrolle über ein sprachgesteuertes Smart Home zu übernehmen.

In der Fachsprache werden solche Angriffe als Adversarial Examples bezeichnet. Lea Schönberr aus der VIGI-Arbeitsgruppe Kognitive Signalverarbeitung entwickelt sie in ihrer Doktorarbeit im Team von Prof. Dr. Dorothea Kolossa. „Wir nutzen dafür das psycholinguistische Modell des Hörers“, erzählt Lea Schönberr. Wenn das Gehör durch Beschäftigung mit einem Ton einer bestimmten Frequenz zu verzerren, können Menschen für einige Millisekunden andere Töne nicht mehr wahrnehmen. Genau in diesen Bereichen verstecken die Forscherinnen und Forscher die geheimen Befehle für die Maschinen. Für den Menschen klingt die zusätzliche Information wie zufälliges Rauschen, das im Gesamttonfall kaum oder gar nicht auffällt. Für den Sprachassistenten ändert es jedoch den Sinn. Der Mensch hört Aussage A, während die Maschine Aussage B versteht.

Ihre Angriffe testete Lea Schönberr an dem Spracherkennungssystem Kaldi, einem Open-Source-System, welches in Amazons Alexa und vielen anderen Sprachassistenten enthalten ist. Sie versteckte unhörbare Befehle in unterschiedlichen Audiosignalen und überprüfte, welche Informationen Kaldi daraus deutierte. Tatsächlich verstand das Spracherkennungssystem die geheimen Befehle zuverlässig. Zusätzlich funktionierte dieser Angriff nicht über den Luftweg, sondern nur, wenn Lea Schönberr die manipulierten Audiodaten direkt in Kaldi hörte. In der Welt der Mittelstufe können die geheimen Befehle aber auch an, wenn die Forscherin dem Spracherkennungssystem die Audiodaten über einen Lautsprecher vorspielt. „Das ist viel komplizierter“, erklärt sie. „Denn der Raum, in dem die Datei abgespielt wird, beeinflusst den Klang.“ Ein Musikstück hört sich etwa anders an, wenn es in einem Kino erklingt, als wenn es über die Lautsprecherboxen eines Autos gespielt wird. Die Größe des Raums, das Material der Wände und die Position des Lautsprechers im Raum spielen dabei eine Rolle.

Alle diese Parameter muss Lea Schönberr berücksichtigen, wenn sie eine Audiodatei erzeugen will, die ein Sprachassistent in einem bestimmten Raum verstehen können soll. Dabei hilft die sogenannte Raumpunktschätzung. Sie beschreibt, wie ein Raum den Schall reflektiert und so den Klang.

In Sprachassistenten wie Alexa steckt die Spracherkennungswelt von Kaldi. Darin haben Bochumer Forscherinnen und Forscher eine Sicherheitslücke gefunden.



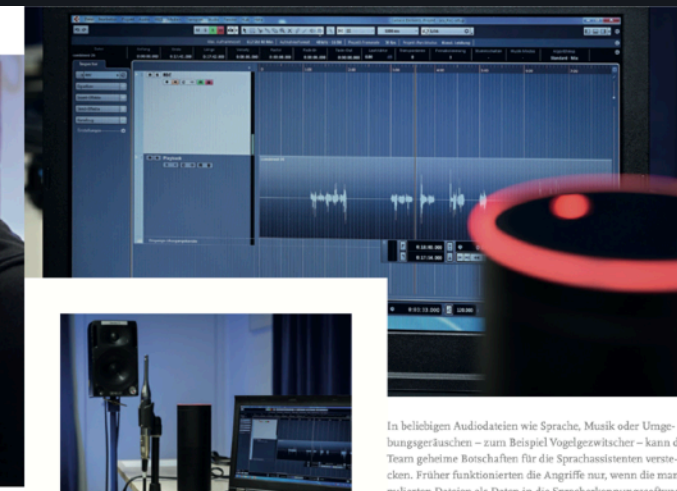
Die Forscherinnen und Forscher manipulierten Audiodaten so, dass Maschinen eine ganz andere Aussage verstehen als Menschen.

versteckt. „Wenn wir wissen, in welchem Raum der Angriff erfolgen soll, können wir die Raumpunktschätzung mit speziellen Computerprogrammen simulieren und beim Erzeugen der manipulierten Audiodaten berücksichtigen“, erklärt Lea Schönberr. Dass das funktioniert, hat die Forscherin bereits gezeigt. Im Team aus der RUB deutierte Kaldi wie gewohnt die geheimen Botschaften, die die Forscherin zuvor in verschiedenen Tonsignalen versteckt hatte.

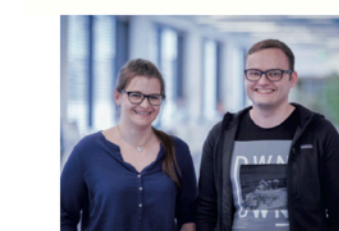
den Lehrstuhl für Systemicherheit von Prof. Dr. Thorsten Holz, dessen Team an Gegenmaßnahmen dazu arbeitet. Die IT-Sicherheitsforscher wollen Kaldi behelligen, für Menschen nicht hörbare Bereiche in Audiosignalen auszuheben und nur das zu hören, was übrig bleibt. „Im Grunde soll die Erkennung der Maschinen nicht wie das menschliche Gehör funktionieren, sodass es schwieriger wird, geheime Botschaften in Audiodaten zu verstecken“, erklärt Thorsten Holz, der in seiner Promotion die Sicherheit von intelligenter Systemen untersucht.

Die Forscher können zwar nicht verhindern, dass Angreifer Audiodaten manipulieren. Wenn diese Manipulation aber in den für Menschen hörbaren Bereichen platziert werden müsste, weil die Sprachassistenten den Rest ausfüllen, so hören sie sich die Angriffe nicht so leicht verstecken. „Wir wissen also, dass der Mensch wenigstens hören kann, wenn mit einer Audiodatei etwas nicht stimmt“, so der Forscher. „Im besten Fall muss ein Angreifer die Audiodatei so weit manipulieren, dass diese mehr wie die versteckte Botschaft klingt als wie das eigentliche Gesagte.“ Die Idee: Wenn der Sprachassistent die für Menschen nicht hörbaren Bereiche eines Audiosignals auswertet, müsste ein Angreifer auf die hörbaren Bereiche ausweichen, um seine Befehle zu platzieren. Um das zu realisieren, nutzt Thorsten Eisenhofer das MFY-Prinzip. MFY-Daten werden komprimiert, indem für Menschen nicht hörbare Bereiche gelöscht werden – genau das ist es, was

die Versteckungstrategie gegen die Adversarial Examples auch versteht. Eisenhofer kombiniert Kaldi dabei mit einem MFY-Encoder, der die Audiodaten zunächst brennt, bevor sie zum eigentlichen Spracherkennung gelangen. Die Tests ergaben, dass Kaldi die geheimen Botschaften tatsächlich nicht mehr versteht, es sei denn sie werden in die für Menschen wahrnehmbaren Bereiche verschoben. „Das veränderte die Audiodatei aber merklich“, berichtet Thorsten Eisenhofer. „Die Störgeräusche in denen die geheimen Befehle versteckt sind, wurden deutlich hörbar.“ Gleichzeitig blieb Kaldi Spracherkennungspersonal trotz der MFY-Bereinigung empfindlicher gegenüber der Spracherkennung für nicht bewusste Daten. Allerdings nur, wenn das System nicht mit MFY-komprimierten Daten trainiert wurde. „In Kaldi selbst ein Machine-Learning-Modell“, erklärt Thorsten Eisenhofer diesen Umstand. Dieses Modell ist notwendig eine künstliche Intelligenz, die mithilfe vieler Audiodaten als Lernmaterial trainiert wird, den Sinn von Tonsignalen zu interpretieren. Nur wenn Kaldi mit MFY-komprimierten Daten trainiert wird, kann es diese später auch verstehen. Mit diesem Training konnte Thorsten Eisenhofer das Spracherkennungssystem dazu bringen, alles zu verstehen, was es verstehen soll – aber eben nicht mehr.



In beliebigen Audiodaten wie Sprache, Musik oder Utensilgeräuschen – zum Beispiel Vogelgezwitscher – kann das Team geheime Botschaften für die Sprachassistenten verstecken. Früher funktionierte die Angriffe nur, wenn die manipulierten Daten zu Daten in die Spracherkennungssysteme geladert wurden. Heute gelangen sie auch, wenn die Audiodaten über Lautsprecher abgespielt werden.



Lea Schönberr und Thorsten Eisenhofer präsentieren am Hörer-Güter-Institut für IT-Sicherheit.

RUHR
UNIVERSITÄT
BOCHUM

RUB

Beyond academia



<https://www.sueddeutsche.de/projekte/artikel/digital/smartspeaker-alexa-und-co-lauschen-versehentlich-e793169>

Beyond academia

Different abstraction levels

Reduced complexity

Visualizations vs. technical details

Caveat: Overselling

Work together and communicate simplifying assumptions

Decline requests when you're not comfortable

Communicating Research

Beyond academia

Interdisciplinary communities

Interdisciplinary communities

Cryptography for trustworthy machine learning

Provably secure implementation of the right to be forgotten

Researchers from different fields



Thorsten Eisenhofer, Doreen Riepel, Varun Chandrasekaran, Esha Ghosh, Olga Ohrimenko, and Nicolas Papernot

“Verifiable and Provably Secure Machine Unlearning”, In Submission



RUHR
UNIVERSITÄT
BOCHUM



Verifiable and Provably Secure Machine Unlearning

ABSTRACT

Machine unlearning aims to remove points from the training dataset of a machine learning model after training, for example when a user requests their data to be deleted. While many machine unlearning methods have been proposed, none of them enable users to audit the procedure. Furthermore, recent work shows a user is unable to verify if their data was unlearned from an inspection of the model alone. Rather than reasoning about model parameters, we propose to view verifiable unlearning as a security problem.

To address these concerns, we propose a *cryptographic approach to verify unlearning*. Rather than trying to verify unlearning by examining changes in the model, we ask the service provider (i.e., the server) to present a cryptographic proof that an agreed-upon unlearning process was executed. This leads us to view unlearning as a security problem that we aim to solve with formal guarantees.

In this paper we propose the *first formal security definition* of verifiable machine unlearning. Our framework describes an iteration-based protocol and requires the server to prove that it has honestly updated the model and dataset in each iteration, either due to training with new data or unlearning previously used data. Only then does the user have sufficient guarantees about deletion of their data. Under this definition, we can instantiate protocols using any unlearning technique and any cryptographic primitives that have appropriate security guarantees.

We identified several challenges while developing the framework that we believe are inherent to unlearning.

- Verifying unlearning cannot be solved by naive one-shot verifiable computation as it requires a user to be able to verify that their data was not re-added at later stages. Hence, the definition has to capture all model updates due to new points added or points being deleted.
- The relationship between an updated model and the evolving dataset needs to be formally captured for verification. For example, a naive way would be to define this relationship as a re-training function, i.e., the updated model is the result of training on the evolved dataset. This can be viewed as “exact unlearning”. However, since other (approximate) unlearning techniques exist, we define this relationship as a set of functions that we call *admissible functions*. This abstraction captures the relationship between models and datasets via initialization, training and unlearning functions.
- As we observe above, the security definition needs to capture consistency of data during training and unlearning, and across model updates and evolving datasets. Though this can be done by passing the whole dataset between the verification steps (training and unlearning) and sending data to the user, we aim to verify consistency in a succinct manner. To this end, we define a strong notion of extractor-based security, capturing that the server must know some underlying dataset in order to compute a valid proof.

Our framework is general and we later demonstrate its applicability to three different unlearning techniques. Notably, none of these have been proved using verifiable computation before. We focus our discussion below on re-training based unlearning, one of

auditors) cannot determine whether a data point is unlearned (or not) by comparing the model’s predictions or parameters before and after claimed unlearning. The complex relationship between training data, models’ parameters, and their predictions make it difficult to isolate the effects of any training point. In fact, prior work [65, 68] demonstrates that a model’s parameters can be identical when trained with or without a data point.

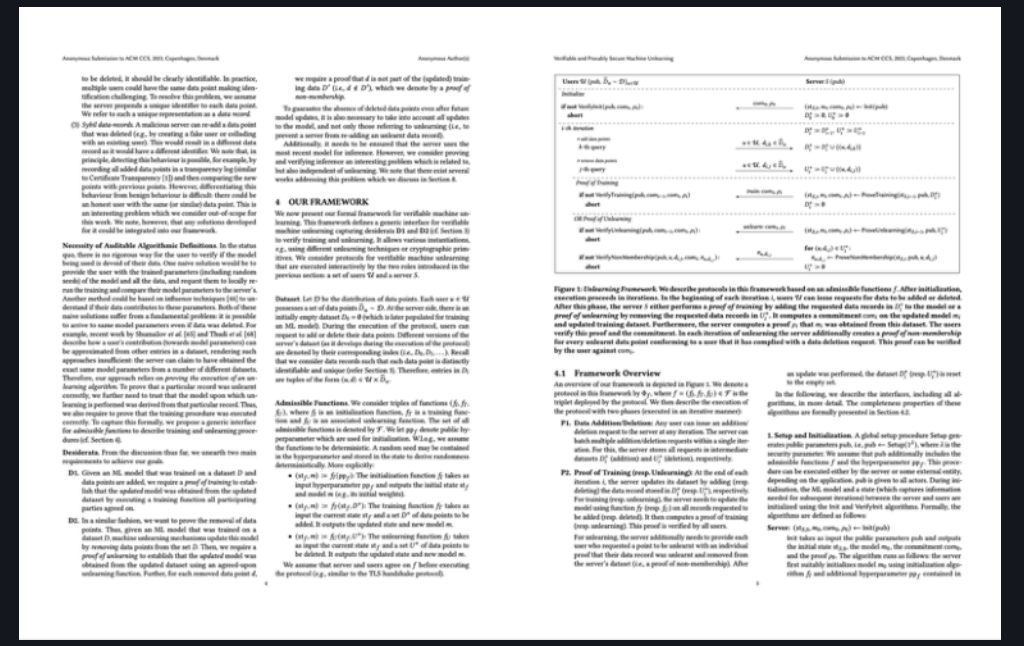
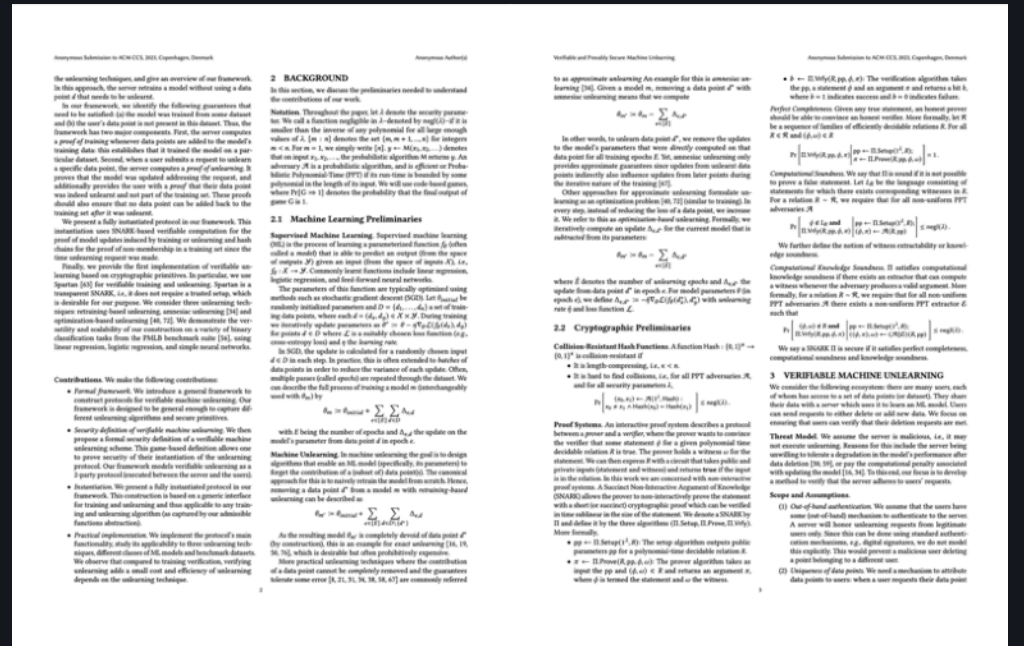
To address these concerns, we propose a *cryptographic approach to verify unlearning*. Rather than trying to verify unlearning by examining changes in the model, we ask the service provider (i.e., the server) to present a cryptographic proof that an agreed-upon unlearning process was executed. This leads us to view unlearning as a security problem that we aim to solve with formal guarantees.

In this paper we propose the *first formal security definition* of verifiable machine unlearning. Our framework describes an iteration-based protocol and requires the server to prove that it has honestly updated the model and dataset in each iteration, either due to training with new data or unlearning previously used data. Only then does the user have sufficient guarantees about deletion of their data. Under this definition, we can instantiate protocols using any unlearning technique and any cryptographic primitives that have appropriate security guarantees.

We identified several challenges while developing the framework that we believe are inherent to unlearning.

- Verifying unlearning cannot be solved by naive one-shot verifiable computation as it requires a user to be able to verify that their data was not re-added at later stages. Hence, the definition has to capture all model updates due to new points added or points being deleted.
- The relationship between an updated model and the evolving dataset needs to be formally captured for verification. For example, a naive way would be to define this relationship as a re-training function, i.e., the updated model is the result of training on the evolved dataset. This can be viewed as “exact unlearning”. However, since other (approximate) unlearning techniques exist, we define this relationship as a set of functions that we call *admissible functions*. This abstraction captures the relationship between models and datasets via initialization, training and unlearning functions.
- As we observe above, the security definition needs to capture consistency of data during training and unlearning, and across model updates and evolving datasets. Though this can be done by passing the whole dataset between the verification steps (training and unlearning) and sending data to the user, we aim to verify consistency in a succinct manner. To this end, we define a strong notion of extractor-based security, capturing that the server must know some underlying dataset in order to compute a valid proof.

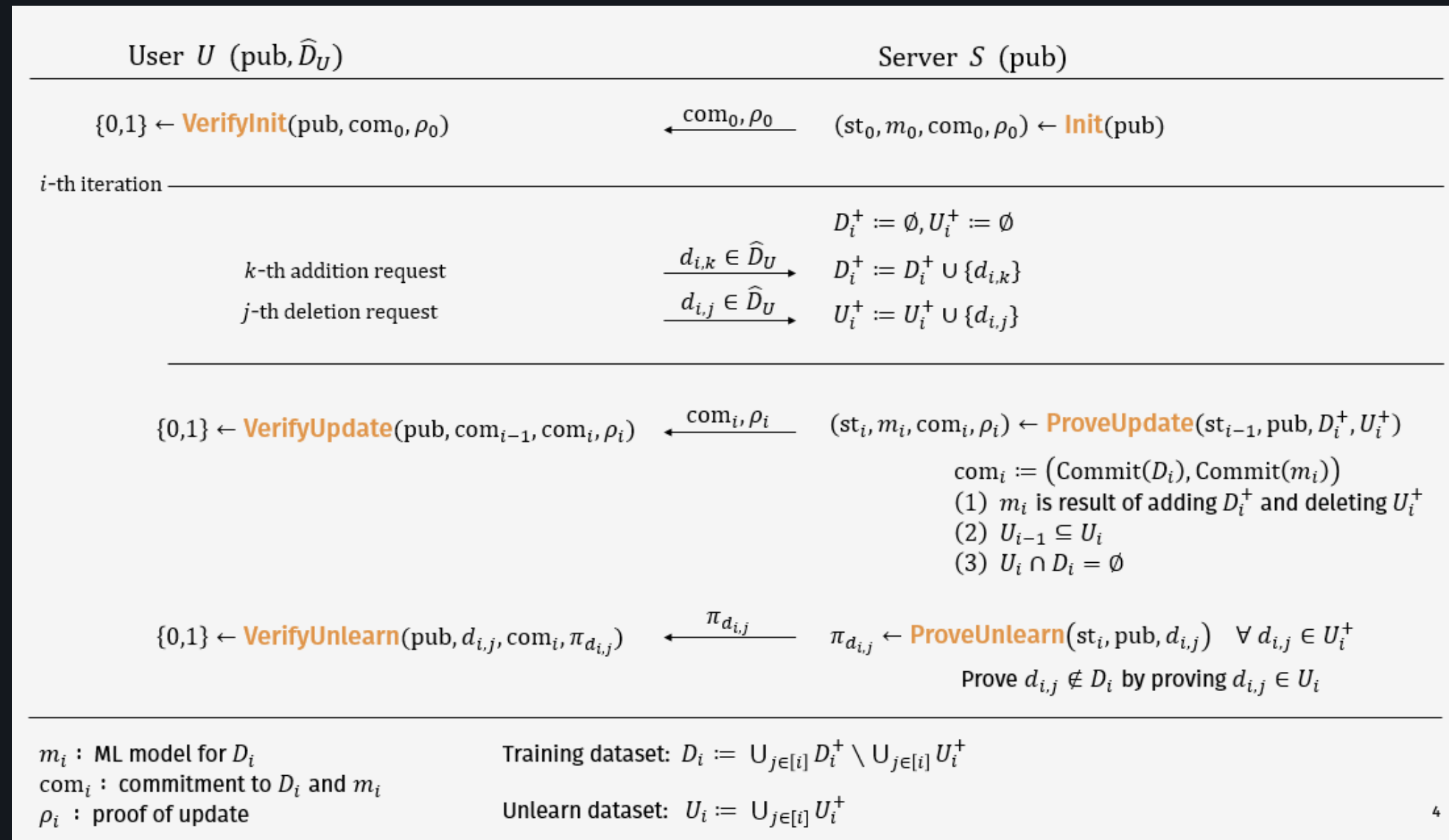
Our framework is general and we later demonstrate its applicability to three different unlearning techniques. Notably, none of these have been proved using verifiable computation before. We focus our discussion below on re-training based unlearning, one of



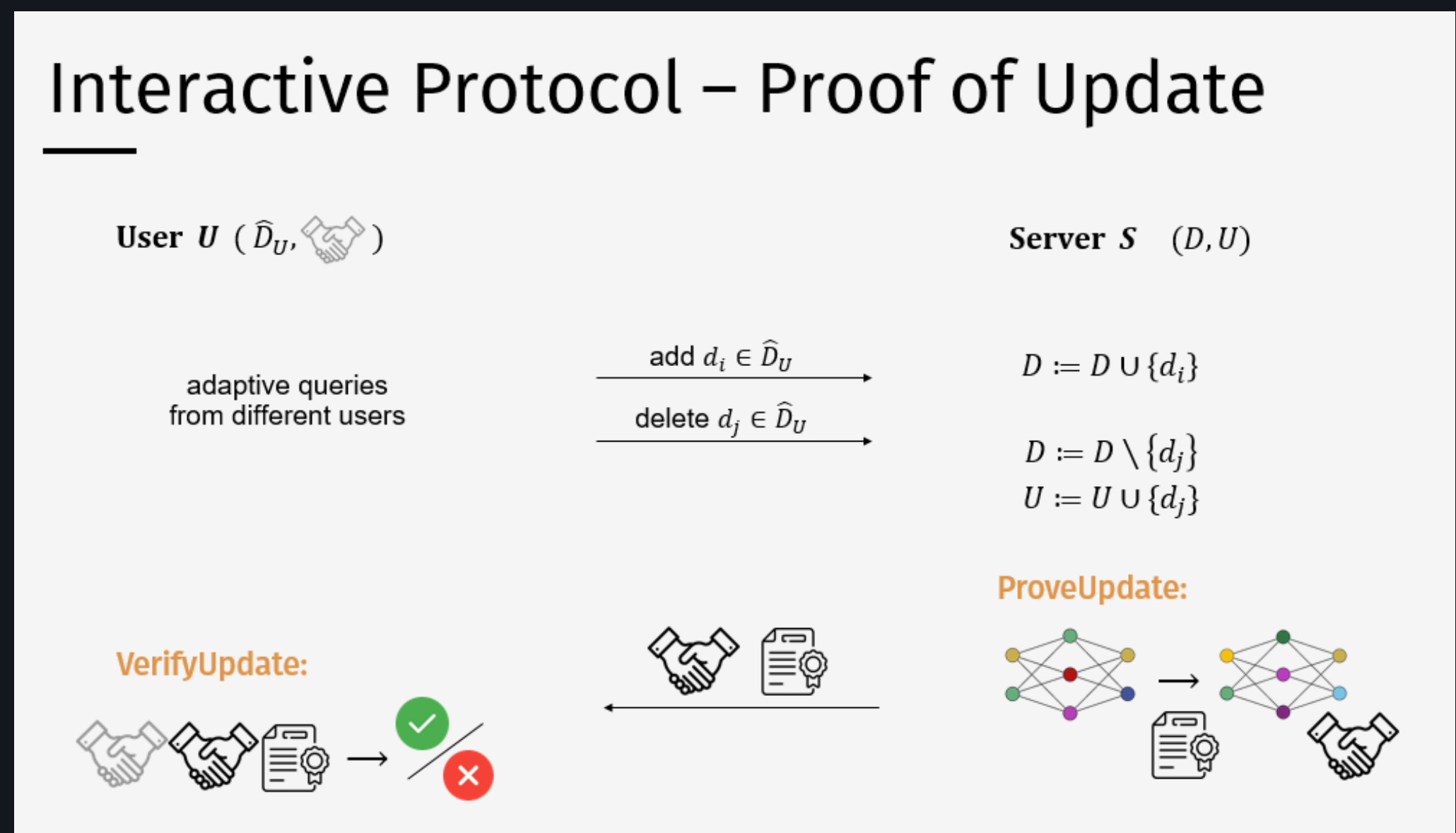
RUHR
UNIVERSITÄT
BOCHUM



Interdisciplinary communities



AUDIENCE:
CRYPTOGRAPHERS



AUDIENCE:
RESEARCHERS IN SECURITY / ML

Interdisciplinary communities

Work on interesting problems on the edge of communities

Develop a common language across communities

Opportunity to work with and learn from experts of different fields

Caveat: Underselling

Solution might be trivial within each community