UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Privacy-preserving Machine Learning for Remote Speech Processing

Francisco Saraiva Sepúlveda Teixeira

Supervisor: Doctor Isabel Maria Martins Trancoso
Co-Supervisors: Doctor Alberto Abad Gareta
Doctor Bhiksha Raj Ramakrishnan

Thesis approved in public session to obtain the PhD Degree in Electrical and Computer Engineering

Jury final classification: Pass with Distinction and Honour

2024

TÉCNICO LISBOA

UNIVERSIDADE DE LISBOA
INSTITUTO SUPERIOR TÉCNICO

Privacy-preserving Machine Learning for Remote Speech Processing

Francisco Saraiva Sepúlveda Teixeira

Supervisor: Doctor Isabel Maria Martins Trancoso
Co-Supervisors: Doctor Alberto Abad Gareta
Doctor Bhiksha Raj Ramakrishnan

Thesis approved in public session to obtain the PhD Degree in Electrical and Computer Engineering

Jury final classification: Pass with Distinction and Honour

Jury

Chairperson: **Doctor** Rui Jorge Morais Tomaz Valadas, Instituto Superior Técnico, Universidade de Lisboa

Members of the Committee:

Doctor Bhiksha Raj Ramakrishnan, Language Technologies Institute, College of Engineering, Carnegie Mellon University, Pennsylvania, EUA

Doctor Miguel Nuno Dias Alves Pupo Correia, Instituto Superior Técnico, Universidade de Lisboa

Doctor Emmanuel Vincent, INRIA Research Center, Université de Lorraine, France

Doctor Tom Bäckström, School of Electrical Engineering, Department of Information and Communications Engineering, Aalto University, Finland

Doctor João Paulo Baptista de Carvalho, Instituto Superior Técnico, Universidade de Lisboa

2024

To my family

Acknowledgments

First and foremost I would like to extend my deepest gratitude to my supervisors, professors Isabel Trancoso, Alberto Abad and Bhiksha Raj. Thank you for giving me this amazing opportunity to learn from you, for passing on your passion for research, for providing me with the best possible learning and growing environment, both at a personal and scientific level, allowing me the freedom to explore while still keeping me tethered. Thank you for all of your continuous encouragement, support, patience and kindness, I can only hope to one day be as good a researcher and as kind a person as all of you.

I extend my thanks to the members of my thesis committee, for accepting to be part of it, and for the invaluable feedback provided since my thesis proposal.

I would also like to thank my PhD "(step-)siblings" Catarina, Mariana, Thomas, Carlos, John and Joana for their friendship and support, our lunches, breakfasts and conversations, both serious and otherwise, provided some of the best moments of my PhD journey. I extend these thanks to all other HLT PhD students, including João, Daniel, Isabel, Patricia, Artur and Diogo. I also thank all other members of HLT, including Rubén and Anna Maria, as well as to David and Teresa for their invaluable work in the group. Thank you all for providing such an amazing work environment.

My gratitude also extends to all who I meet during my stay at CMU, and whom I have interacted since, including professor Rita Singh, as well as Roshan, Hira and Ahmed. I also want to specifically thank my co-authors Raphaël and Karla, for their amazing work and friendship, I hope we can continue to find the time to play *skat*.

I cannot abstain from thanking everyone I have worked with at ISCA-SAC since I first started as a volunteer in 2020. I have learned a lot from being a member of ISCA-SAC and I am extremely proud of the work that we have been able to accomplish, as well as of the work that will continue to be done by the next generation of volunteers.

On a more personal level, I would like to thank my friends, Daniel and Mariana, Margarida and Gonçalo, thank you for always being there. I also want to thank my newly found "compadres", Diogo, João, Rodrigo and Ausenda for their friendship, as well as Matilde and Zé for their friendship and support.

I would like to thank all of my family, all of my "Figueira" cousins who are too many to name, as well as my uncle and aunt Jorge and Gabi, and my cousins Mariana, Guilherme and Pedro. To my late grandparents Lívia and Francisco, and my uncle Francisco, I can only say that you are dearly missed.

I have to express my gratitude to my "parents in-law", Isabel and António, and my siblings-in-law, Mafalda and Miguel. I cannot thank you enough for welcoming me into your family and for all of your constant support.

Finally, to my parents, Miguel and Lígia and my sister Margarida I cannot express how much your encouragement, support and your unconditional love and means to me. To my parents, I thank you for

showing me the importance of always being curious and always striving to learn more about everything, for passing on your passion for science and research, and for always showing me that I should stand up for what I believe in. To Margarida, thank you for being my loving sister, who is always able to show the three of us that there are other ways of looking at the world. Thank you all for pushing me into this journey and always believing in me.
To Mariana and Pluto, thank you for being my chosen family, for always being there, for all the patience during the late nights, all of the support, all of the encouragement and love, without you this thesis would not have been possible.

Obrigado a todos,
Francisco

Abstract

As an increasing number of remote services and applications turn to speech as a means of interaction, authentication, or information extraction, there is a growing demand for privacy-preserving solutions that protect the user's speech data while it is being processed in remote servers. In this thesis, we address this issue by developing new methods to protect user privacy in remote speech processing, based on two main paradigms: cryptographic processing, and privacy-oriented speech manipulation. Initially, we propose cryptographic-based methods for the privacy-preserving detection of Parkinson's disease and Obstructive Sleep Apnea detection, as well as for the extraction of speaker representations for Automatic Speaker Recognition and Diarization. The results obtained for these methods show that, although cryptographic methods provide strong privacy guarantees, they may be too computationally expensive and difficult to adapt to complex speech-processing tasks. However, we argue that cryptographic protocols may be the most adequate solution for tasks where it is difficult to disentangle speaker and task-related information, such as clinical applications, and remain the best solution for scenarios where privacy is paramount.

Our following approach consists of machine-learning-based privacy-oriented speech manipulation methods that are able to remove sensitive speaker-related information, such as the speaker's age and sex. We show that these methods are more computationally lightweight and more independent of downstream tasks than cryptographic protocols. Despite their weaker privacy guarantees, we show that our privacy-oriented speech manipulation methods provide users with finer-grained control over the information that should be kept private, allowing them to trade off privacy for utility in speech applications.

In a final contribution, we explore membership inference in Automatic Speech Recognition and showcase its potential to act as a tool to audit the training data of these models with regard to the unauthorised use of data.

Overall, this thesis contributes with advances in the two main explored paradigms, provides insights into different trade-offs, and opens new avenues for future research in the increasingly important problem of privacy in speech processing.

Keywords

Speech, Machine Learning, Privacy, Cryptography, Remote Processing.

Resumo

Com o aumento do número de serviços e aplicações que funcionam de forma remota e que utilizam a fala como uma forma de interação, autenticação ou extração de informação, tem simultaneamente crescido a necessidade de desenvolver soluções que preservem a privacidade do sinal de fala dos utilizadores destas aplicações. Nesta tese, é abordado o problema da aprendizagem automática privada para processamento da fala. Concretamente, são desenvolvidos métodos que permitem proteger a privacidade da fala de utilizadores de sistemas remotos, tendo por base dois paradigmas: processamento criptográfico e manipulação da fala orientada à privacidade.

Como exemplos do primeiro paradigma, propõem-se métodos criptográficos para a deteção privada de doenças como a doença de Parkinson e a apneia obstrutiva do sono, e para a extração de representações de orador em tarefas de reconhecimento automático e diarização de orador. Os resultados obtidos mostram que, apesar destes métodos criptográficos oferecerem fortes garantias de privacidade, o seu custo computacional poderá ser demasiado alto, dificultando a sua adaptação a tarefas de processamento da fala complexas. No entanto, os protocolos criptográficos poderão ser a solução mais adequada para tarefas onde é difícil separar informação relacionada com o orador da informação relacionada com a tarefa, sendo a melhor solução para situações em que a privacidade é fundamental.

Em alternativa, como segundo paradigma, propõem-se métodos de manipulação da fala orientados à privacidade, com base em aprendizagem automática, que possibilitam a supressão de informação do orador. Os resultados obtidos mostram que estes métodos têm um custo computacional muito inferior ao das abordagens baseadas em protocolos criptográficos, sendo também mais independentes das tarefas a jusante. Apesar de oferecerem garantias de privacidade mais fracas, estes métodos permitem que os utilizadores possam escolher um bom compromisso entre privacidade e usabililidade em aplicações de fala.

Como contribuição final, exploram-se técnicas de inferência de pertença como ferramenta de auditoria de modelos de reconhecimento automático da fala, relativamente ao uso não autorizado de dados de utilizadores.

No seu todo, esta tese pretende contribuir com avanços nos dois principais paradigmas explorados e abrir novas vias para investigação futura sobre o problema cada vez mais premente da privacidade em processamento da fala.

Palavras Chave

Fala, Aprendizagem Automática, Privacidade, Criptografia, Processamento Remoto.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

AHC	Agglomerative Hierarchical Clustering
ASD	Automatic Speaker Diarization
ASV	Automatic Speaker Verification
ASR	Automatic Speech Recognition
AUC	Area Under the ROC Curve
AUPRC	Area Under the Precision-Recall Curve
BIM	Basic Iterative Method
BFV	Brakerski/Fan-Vercauteren
BGV	Brakerski-Gentry-Vaikuntanathan
CKKS	Cheon-Kim-Kim-Song
CCC	Concordance Correlation Coefficient
CCPA	California's Consumer Protection Act
CTC	Connectionist Temporal Classification
DCT	Discrete Cosine Transform
DER	Diarization Error Rate
EER	Equal Error Rate
FGSM	Fast Gradient Sign Method
FHE	Fully Homomorphic Encryption
FPR	False Positive Rate
GAN	Generative Adversarial Network
GC	Garbled Circuits
GCA	Gated Convolutional Autoencoder
GDPR	General Data Protection Regulation
GMM	Gaussian Mixture Model
GMM-UBM	Gaussian Mixture Model - Universal Background Model
GMW	Goldwasser-Micali-Wigderson
HE	Homomorphic Encryption
HIPAA	Health Insurance Portability and Accountability Act

JER	Jaccard Error Rate
JND	Just Noticeable Differences
KL	Kullback-Leibler
LLH	Limited-leakage Hashing
LSH	Locality-Sensitive Hashing
LWE	Learning With Errors
MDCT	Modified Discrete Cosine Transform
MFCC	Mel Frequency Cepstral Coefficients
minDCF	minimum of the Detection Cost Function (minDCF)
MI	Membership Inference
ML	Machine Learning
OT	Oblivious Transfer
PESQ	Perceptual Evaluation of Speech Quality
PCC	Pearson's Correlation Coefficient
PCA	Principal Component Analysis
PHE	Partially Homomorphic Encryption
PLDA	Probabilistic Linear Discriminant Analysis
RBF	Radial Basis Function
RSS	Replicated Secret Sharing
SBE	Secure Binary Embeddings
SHE	Somewhat Homomorphic Encryption
SMC	Secure Multiparty Computation
SMH	Secure Modular Hashing
SNR	Signal-to-Noise Ratio
STFT	Short-time Fourier Transform
SVM	Support Vector Machine
TPR	True Positive Rate
RLWE	Ring Learning With Errors
UAR	Unweighted Average Recall
VAE	Variational Autoencoder
VAD	Voice Activity Detection
VB-HMM	Variational Bayes - Hidden Markov Model
VQ-VAE	Vector Quantised - Variational Autoencoder
WER	Word Error Rate

1

Introduction

> "It seems to me (...) that the advance of civilization is nothing but an exercise in the limiting of privacy."
>
> *Isaac Asimov, Foundation's Edge, 1982.*

Speech is our most natural means of communication and a fundamental part of our everyday lives. Through speech, we interact with each other and communicate our thoughts and emotions.

Speech production is an amazingly complex process that involves our brain, and lungs, along with several muscles, articulators and organs of the vocal tract [289]. Each of these components leaves its mark on speech, making it unique to and reflective of the speaker. As a result, speech conveys a wealth of intrinsic information about the speaker that extends far beyond the communicative information that is generally associated with speech [168, 281].

From a technological perspective, speech is a natural means of human-computer interaction. Its uniqueness and the information it contains about a speaker allows its use in authentication systems and lends it potential as an inexpensive and non-intrusive biomarker for health [66]. Speech and language technologies have seen significant progress in the past decade, supported in large part by the advent of deep learning, the unprecedented availability of massive data sources, as well as access to high-performance computing platforms [360]. This, in combination with the ubiquitousness of smart devices in the modern world, is leading to the deployment of numerous *cloud*-based speech services and applications that leverage machine learning models. Among other speech applications, voice assistants and smart speakers are arguably the most popular, with an estimated 4.2 billion voice assistants having been in use worldwide in 2020 [302], and with the smart-speaker global market share expected to reach 35.5 billion US dollars by 2025 [303].

The developments in speech technology have also resulted in improved automatic systems that are able to infer speaker-related information more accurately than humans and even to obtain information that would be out of reach for non-experts – e.g., the speaker's physical characteristics, personality traits, emotional state and even information regarding the speaker's physical and mental health [289]. All of this information can be considered helpful for service providers, who can use it to improve their services as well as for commercial purposes. From a user's perspective, however, this information can be considered sensitive, and its collection without consent is deemed inappropriate – in other words, it is information that is and should remain *private*.

When conversing with others in in-person settings, we are aware of whom we are sharing information with: we can assess the trust we have in other interlocutors, the number of participants in the conversation, and the setting where the conversation is occurring and adjust what we say and how we say it accordingly. This is in contrast to interactions with *cloud*-based speech services, where users do

not have control over – or possibly even knowledge of – who has access to their speech, how it is used, and the information that can be derived from it. This, combined with the wealth of information that can be automatically derived from speech and the growing use of speech technologies, has raised concerns over users' privacy and demands solutions for the protection of speech data.

As a society, we have seen similar concerns over different types of user data lead to the implementation of several regulations, including the European Union's (EU) General Data Protection Regulation (GDPR) [88], California's Consumer Protection Act (CCPA) [40] and the USA's Health Insurance Portability and Accountability Act (HIPAA) [327]. For instance, the GDPR strives to answer data privacy concerns by introducing restrictions on how service providers are allowed to deal with personal data (cf. section A.1)[1] and encourages the development of privacy-by-design solutions – i.e., systems designed in such a way that they inherently take into account the privacy of user data (cf. section A.2). Although not directly aimed at speech, in light of the definition of personal data provided by the GDPR (cf. section A.3), speech data and the information that can be derived from it may be legally regarded as Personally Identifiable Information (PII), i.e., information that on its own is enough to determine the identity of an individual [209], and consequently a type of data that is protected under this regulation.

These factors have led to a growing interest in the field of privacy in speech processing in recent years [210, 320]. Although traditionally focused on cryptographic techniques for biometric and privacy-preserving speech processing applications [210], this field has expanded largely in the areas of speech anonymisation [320] (i.e., removing speaker identifying characteristics from the speech signal), speaker information minimisation [8, 351], and in the development of methods for privacy in training speech-based machine learning models [145, 348]. Despite the above, privacy in speech processing is still an underrepresented topic in the overarching field of speech science and technology, requiring more solutions and a more comprehensive range of approaches. In addition, the general public still lacks awareness of the possible vulnerabilities individuals may face due to the publishing or sharing of their speech data [158].

With this thesis, we aim to contribute towards the development of solutions to the issue of privacy in speech processing, having as a main foundation the ethical needs to protect users of speech technologies, as well as the knowledge that privacy-preserving technologies are necessary to abide by data protection regulations. The core focus of this thesis rests on a specific scenario: *remote speech processing*, for which we propose cryptographic-based solutions that provide confidentiality guarantees and machine-learning-based solutions that minimise the information contained in representations of the speech signal. In addition to this, we explore an additional aspect of speech privacy: the privacy of speech training data in the context of machine-learning model deployment.

[1] This definition, as well as the following ones can be found in Appendix A.

The present chapter aims to motivate the problem of speech privacy and introduces the base concepts that allow us to define the thesis' target problem. We start by providing a short discussion on privacy, privacy definitions, and their implications for this thesis. This is followed by a description of possible vulnerabilities that come from disclosing speech data, after which we define this thesis' main working scenario. We then provide an overview of existing methods and challenges for privacy in remote speech processing. Finally, we present this thesis's research questions and main contributions.

1.1 Speech data privacy

Privacy is a fundamental human right; it is enshrined in Article 12 of the United Nations' Universal Declaration of Human Rights as [326]:

> *No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.*

Even though it is considered important almost universally, privacy is difficult to define, as it is a concept that varies across cultures and time, being perceived differently by different individuals. Nevertheless, without an understanding of privacy, it is not possible to understand what it means to protect it, as we aim to do in this thesis. For this reason, in this section, we introduce some definitions and conceptualisations of privacy that will later be used to delineate what it means to uphold privacy in remote speech processing. In addition, in this section, we present an overview of possible threats or vulnerabilities that may arise from not protecting speech data. While deontological reasons alone are enough to justify protecting privacy, we consider that it is also necessary to have an understanding of the possible consequences of not doing so.

1.1.1 Defining privacy

Modern views of privacy have their main precursor in an 1890 Harvard Law Review paper by Warren and Brandeis [344], where the authors defended that the right to privacy, defined as "the right to be let alone", is a right in and of itself, on par with the rights to life and property, and argued for its inclusion in criminal law in the United States of America. This stance stemmed from contemporaneous journalistic intrusions, along with the advent of easily accessible photographic cameras. This work was crucial to the development of privacy law in Western society and was fundamental in the establishment of privacy as a human right [298].

Warren and Brandeis' work has since led to the emergence of numerous privacy definitions and conceptualisations. To provide a brief overview of the current views of privacy, we turn to Daniel Solove's categorisation of existing privacy definitions and conceptions [295], which groups them under

six concepts: the right to be let alone, limited access to the self, secrecy, personhood, intimacy and control over information.

Definitions based on the concept of "the right to be let alone" follow directly from Warren and Brandeis' definition and view the right to privacy as a right against intrusions into one's life, physical or otherwise [296]. The concept of limited access to the self can be seen as a refinement of the right to be let alone, wherein privacy is seen as a state where one is protected against any unauthorised intrusion into oneself and one's private affairs [297]. For instance, Sissela Bok defines privacy as "the condition of being protected from unwanted access by others – either physical access, personal information, or attention" [30]. Privacy is also defined as secrecy; Richard Posner describes privacy as one's "right to conceal discreditable facts about himself" [256], whereby privacy is violated by the disclosure of "secret" information [297]. In personhood-based views, privacy is considered a medium through which one can develop and assert oneself as a person [298]. For instance, Jeffrey Reiman states that the right to privacy "protects the individual's interest in becoming, being, and remaining a person" [269]. Similarly, it is also argued that privacy can be defined through what is considered *intimate*, with this view emphasising privacy's role in relationships [297]. Following this view, Julie Inness defines privacy as "the state of possessing control over a realm of intimate decisions, which include decisions about intimate access, intimate information, and intimate actions" [123].

Finally, the group of privacy definitions that is most often used in the fields of data privacy and cloud computing is privacy as control over oneself and one's personal information [198]. Among other control-based definitions, Charles Fried states that privacy is "the control we have over information about ourselves" [100], and Alan Westin defines privacy as "the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others." [346].

Overall, each of these categories focuses on different aspects of privacy, each coming from different points of view. Together, they show that privacy is not limited to a single dimension and that upholding privacy not only prevents certain wrongs in these dimensions but also aids individuals in developing themselves and in the management of their relationships. However, these definitions are considered to be either too narrow – i.e., do not cover all aspects of privacy – or too broad – i.e., cover all aspects of privacy but are so broad that harmless actions can be seen as privacy infringements [297].

As an alternative, Solove [295, 297] and Nissenbaum [218] argue that to understand what it means to uphold privacy or otherwise infringe upon it, it is necessary to define the context under which privacy is being analysed. Furthermore, these authors argue that although they are important from a philosophical point of view, definitions of privacy are often impractical for those who aim to implement privacy protection measures, develop privacy policies and regulations or enforce existing legal frameworks.

The Contextual Integrity theory proposed and developed by Helen Nissenbaum [218, 219] rejects the

idea of a single definition of privacy and, instead, delineates a general framework that can be used to evaluate whether privacy has been kept in any context or situation, advocating that privacy is "a right to the appropriate flow of personal information" [219], based on the assumption that ensuring privacy is dependent on context.

Contextual Integrity defends that to assess if privacy has been violated in a given context, it is necessary to delineate an appropriate *information flow*: how the transfer of personal information between one party to another occurs, including all intermediate steps and parties. If the information flow abides by contextual privacy norms – e.g., social rules and regulations – then it can be said that privacy has been kept. In contrast, if the information flow diverges from the expected norm, then privacy has been violated. For example, consider the case where an individual is feeling unwell and would like to schedule a medical appointment at a clinic. For this purpose, the individual needs to share their health information, as well as their symptoms. To do so, this information goes through the clinic's secretariat, who sends it to a nurse for an initial screening. A doctor is then selected, and this information is shared with this specialist. If the symptom-related information had not been shared with anyone outside this chain, then the information flow would have kept its integrity. However, if this information had been additionally shared with an insurance company without the subject's knowledge or consent, then the privacy of this information would not have been kept, as this would not adhere to the norm expected for this information flow.

The process of defining the context, the information flow and the expected social, ethical or legal norm generalises to any situation where privacy needs to be evaluated, making the theory of Contextual Integrity a versatile tool for privacy practitioners. When comparing this framework to what is common practice in the field of privacy in Machine Learning (ML), and specifically in privacy in speech processing, we can see that the idea of a proper information flow is usually present, although under different terms, underlining the definition of the "attacker", who breaks the normal information flow, and against whom defences should be created to uphold privacy. In this thesis, we follow the same reasoning and provide a detailed definition of remote speech processing and what we consider to be the principal vulnerabilities and possible attackers in this setting in Section 1.2.

1.1.2 Speech data privacy vulnerabilities

As mentioned at the beginning of this chapter, speech carries not only meaning and intent but also information about the speaker. All of this information can be used to the speaker's benefit, but it can also be used to their detriment. With this section, we aim to showcase different types of privacy threats to speech data to further motivate the importance of speech data privacy. A summary of these threats, or vulnerabilities, can be found in Figure 1.1.

Formally, the information comprised in speech can be categorised as *linguistic* (or lexical) content,

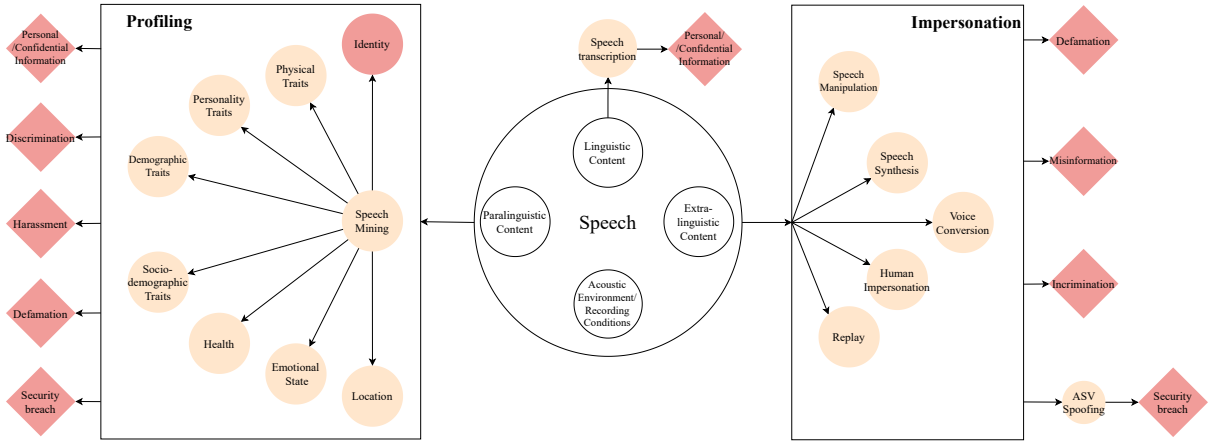


Figure 1.1: Speech data privacy vulnerabilities.

paralinguistic content such as conveyed emotion, and *extralinguistic* content, usually an involuntary result of speech production, reflecting speaker characteristics, i.e., characteristics of speech that go beyond communication [168][2].

The fact that speech incorporates all this information means that from a speech recording, it is possible to manually or automatically infer not only linguistic content but also a vast amount of speaker characteristics, including: the speaker's identity; physical characteristics such as the speaker's facial structure [224] and height [201]; the speaker's emotional state [281]; the speaker's demographic and socio-demographic traits, including the speaker's sex, age range, accent, ethnicity and education level [159]; the speaker's personality traits [249]; as well as the presence of speech affecting diseases such as Parkinson's disease, Alzheimer's disease and Obstructive Sleep Apnea [32, 250, 251], as well as mental disorders, such as depression and anxiety [20, 67], among others. In addition to this, speech recordings also carry information on the recording conditions and characteristics resulting from the acoustic environment where they were recorded [307].

From the standpoint of privacy, gathering all of this information can lead to, without being exhaustive, discrimination by individuals – due to personality, physical or socio-demographic traits – or companies –e.g., companies increasing insurance premiums on account of the individual's socio-demographic characteristics or health state – as well as harassment by others – e.g. re-identification of users across platforms [15]. Some of this information may also be used for defamation, e.g., undermining the credibility of a public figure by exposing their health state. Inferring a speaker's identity might also be, on its own, a breach of privacy and even security, e.g., consider a recording of a whistle-blower. Even inferring characteristics about the recording's acoustic environment can lead to a security breach, as it may provide information on a speaker's location and surroundings [15].

[2]Some authors do not make the distinction between paralinguistic and extralinguistic information, using paralinguistic as an umbrella term for all information that is not linguistic [281].

The linguistic content of speech – which can be obtained via manual or automatic transcription – may also contain private or confidential information. Similar to the above, this information can be used for discrimination, harassment and defamation, among many others. Further, it can also pose a security breach, as the speaker might utter information such as addresses, social security numbers and banking information.

Beyond the information that can be extracted from speech, we also need to consider what can be done with a recording of a speaker's voice, namely speaker *impersonation*. For example, voice-protected systems may be spoofed by replaying speech recordings [148]. Moreover, an individual might learn how to impersonate a speaker by listening to the speech recording. Ill-intended parties might also synthesise new recordings of the speaker's voice or even convert a speech recording from another speaker and change it to sound like the speaker in the original recording [342]. Besides creating security breaches by spoofing voice-protected systems, false recordings of a speaker's voice may also be used to defame, incriminate, or even to spread misinformation [354] – e.g., consider the case where a politician's voice is presented saying something they have not spoken in reality, or the case where a politician's voice is made to sound like they have mental illness or, as has happened in a real case, sound like they are inebriated [192].

Some of these vulnerabilities may seem unrealistic; however, recent cases reported in the media are beginning to demonstrate that this is not the case. For instance, a 2021 article by Forbes [97] reported that, in 2020, a Hong Kong bank manager had been misled by a *deepfake* of the voice of one of his clients, a director at company, resulting in a theft of 35 million dollars. More recently, in a 2023 Forbes article [98], Forbes itself revealed the identity of the owner of a social media account by asking a forensics expert to compare recordings of the owner of the account with public recordings of the voice of someone who was suspected to be the account's actual owner, which the forensics expert then confirmed. Even though Forbes claims to have had reason to reveal the owner's identity, this was not authorised by the individual himself.

In addition to these accounts of speech privacy violations, one can also look at existing patents on the collection of user information from speech to understand that from the point of view of large companies, inferring user attributes from speech is a realistic scenario [122, 134].

1.2 Privacy in remote speech processing

As discussed in the previous section, privacy is highly dependent on context. For speech data, mechanisms that uphold privacy depend on where speech data is stored and processed and the medium over which it is transmitted. Hence, to determine how we can ensure privacy, we need to define the setting under which we want to uphold it.

As mentioned at the beginning of this chapter, cloud-based services and applications are becoming

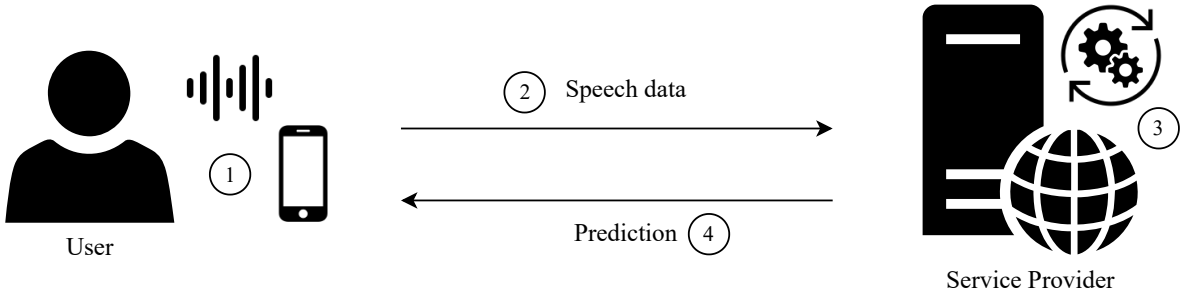


Figure 1.2: Speech processing in a cloud-based setting.

extremely common as a way for companies to provide users with access to their machine-learning models. Developing machine learning models is a time-consuming and costly process that requires high levels of expertise and access to large amounts of data. Machine learning models also require considerable computational power to train and even to perform inference after being deployed [111, 235]. This entails that service providers have an incentive to protect the privacy of their models and to avoid distributing these models to users, as they are often the core of the company's business. Simultaneously, users may not have access to the computational power required to run these models. Through cloud-based applications, service providers are hence able to protect their models while still being able to make them available to users, and, at the same time, users can apply these models to their data without having access to high-performance computing machines.
As stated above, in this thesis, our main focus is this setting applied to speech data. This setting can be summarised as follows (cf. Figure 1.2):

1. A user records their speech through a dedicated application on a device with internet access;
2. The user's recorded speech is sent to a remote server for processing;
3. The remote server receives this data and feeds it through its machine-learning pipeline to obtain the desired result;
4. The remote server sends this result back to the user.

In the ideal version of this setting, the service provider would process the user's data only for the intended purposes – following existing data protection regulations – with the result of this processing being sent back to the user without any additional unauthorised processing or sharing of this data, or information derived from it, with third parties. However, this ideal scenario is strongly dependent on the service providers' trustworthiness. In fact, in this scenario, users effectively lose control over their data, having little guarantee against possible misuse of their data. Among other improper behaviours, an untrustworthy service provider may sell user data, use it for undisclosed purposes, perform

unauthorised processing, and share it with third parties, making users vulnerable to the privacy threats presented in the previous section.

Although some of these privacy concerns may be mitigated through service-user agreements, conformance to these will depend on the service provider's accountability, an assumption that may not hold or even be sufficient for all situations. For instance, consider a medical professional who wants to use a machine learning model provided by a company to evaluate a patient's data concerning a health condition. The medical professional is bound by confidentiality and cannot use this service, as sharing the patient's data with the service provider would break patient-doctor confidentiality, violating the patient's privacy. As such, even the ideal setting described above does not provide sufficient guarantees for this scenario. Health data is the most stark case, but we can also consider internal company data such as company meeting recordings. From the point of view of the company (as a user), using such a service to transcribe or annotate internal meetings would be a risk, as it would mean sharing internal company information with an outside party.

For these reasons, in this thesis, we view the service provider as a possible *attacker* and focus on the development of methods that can protect the privacy of the user's speech data against the service provider.

One could argue that the user may also attack the service provider's model. An untrustworthy user might attempt to gain more information about the model than what is allowed and conduct model inversion attacks [99] that attempt to reconstruct training data points, membership inference attacks [287] that attempt to determine whether specific data records were used to train the model, and model extraction attacks [321] that attempt to reconstruct the model's weights and architecture. However, although the development of mechanisms that protect the service provider's model against user attacks is an important and relevant line of research, this thesis does not focus directly on these attacks. Nevertheless, for the interested reader, we leave here pointers to relevant speech research in this area [3, 50, 248, 283].

Furthermore, even though in this thesis we only focus on the service provider as an attacker, we are aware that there exist other possible attack surfaces, such as intrusions from third parties into the user's device; the service provider's computing platform; physical eavesdroppers that listen to the user recording their voice; and attacks on the communication channel, among others [15]. All of these are active and valid areas of research. However, for simplicity, in the methods developed in this work, we assume that the acoustic environment, the user's device, the remote server and the communication channel are secure against third-party intrusions.

1.3 Methods and challenges for privacy in speech processing

The growing interest in speech data privacy has accelerated the development of privacy-preserving machine learning methods. Having different trade-offs between privacy, utility and computational cost, these methods can be broadly categorised into cryptographic processing, privacy-oriented manipulation, differential privacy and federated learning. Even though this categorisation does not include all existing methods for speech privacy (e.g., slicing [189], speech content filtering [4]), it is intended to provide a general overview of existing techniques.

In this section, we will define the two paradigms that are most relevant to this thesis: cryptographic processing and privacy-oriented manipulation, and discuss their trade-offs when applied to remote speech processing.

It is further important to mention that we will not cover methods that are more directly related to security, such as *anti-spoofing* techniques that protect speech-based authentication systems against attacks [139, 352], or *watermarking* techniques that allow one to verify the ownership of a speech signal to protect copyright and intellectual property [241], or that allow humans to identify synthetic speech [137].

1.3.1 Cryptographic-based processing

Cryptographic-based processing is the backbone of many privacy-preserving machine learning applications [108, 136, 196, 273], wherein cryptographic protocols are used to perform numerical computations privately. Among other cryptographic protocols, Homomorphic Encryption (HE) and Secure Multiparty Computation (SMC) stand out as general-purpose tools that can privately perform most numerical operations, making them suitable for many applications, in particular remote speech processing.

HE is a family of cryptosystems that allow arithmetic operations to be performed over *ciphertexts*, i.e., encrypted values. In protocols based uniquely on HE, users can encrypt their data, send it to a remote server for processing and receive an encryption of the result of this process. Since all operations are performed over encrypted data, the users can be sure that no outside party has had access to their data, ensuring confidentiality and, hence, privacy. On the other hand, the service provider's model is never sent to any user, providing it with some protection – even though the model and its weights are not shared, the model's outputs may provide some information about it. In addition, any protocol using only HE derives its security from underlying computational hardness or information-theoretic guarantees, which, assuming a correct implementation and no decryption of intermediate results, also extend to the complete protocol.

While HE is a perfect solution in theory, in practice, it is limited by the number or type of operations allowed by each scheme and by its high computational cost. Most modern HE schemes allow both

additions and multiplications to be computed (and thus any polynomial function). However, the number of times each operation can be applied to a ciphertext determines the computational cost of the scheme, i.e., being allowed to perform more operations over a ciphertext entails a higher computational cost for the overall protocol. Another critical limitation of HE is the fact that, except for cryptosystems that work directly over binary values [56], most HE schemes cannot perform non-linear operations. The above severely limits the applicability of HE to remote speech processing. In particular, the computational complexity and non-linear nature of state-of-the-art machine learning models make pure HE solutions to this problem extremely inefficient.

An orthogonal but complementary approach to HE is SMC, a family of protocols that allow two or more parties to jointly compute functions over their data with the guarantee that the inputs of each party are kept private. Contrarily to HE, SMC protocols allow the computation of any function. Moreover, SMC allows trading-off security guarantees for computational efficiency, making it much more versatile than HE. Nonetheless, the security of SMC protocols is dependent on whether other parties fit the adversarial behaviour model defined for the protocol. For instance, it is possible to implement very efficient protocols simply by assuming that other parties are trustworthy enough that they will follow the established protocol. However, when a higher level of privacy is required, upholding security means adding extra layers of complexity to verify if other parties are following the protocol, which significantly increases the scheme's computational cost. SMC also requires all parties to be actively involved in the computation, which translates into a constant exchange of data through the computation, entailing a much higher communication cost. Consequently, this also restricts the applicability of SMC protocols in limited bandwidth settings. Nevertheless, in settings where trust is present and where every party has sufficient computational power, SMC can provide moderately efficient solutions to perform inference over, or even train, low to medium complexity machine learning models [69, 335]. It should be noted that hybrid solutions between HE and SMC are also possible, particularly in situations where there is a need to trade off computational and communication costs [136].

Finally, it is also important to mention limited-leakage techniques [133, 253]. Instead of providing perfect secrecy, these information-theoretic constructions allow the leakage of limited amounts of information. Specifically, limited-leakage hashing techniques guarantee that the distance between a pair of hashed vectors is proportional to the distance between the original vectors if this distance between the original vectors is smaller than a threshold, providing no information otherwise. Although not general purpose, this type of technique is extremely lightweight when compared to other cryptographic constructions and is particularly useful for clustering and template-based verification tasks.

Overall, cryptographic constructions have many potential applications in remote processing scenarios, providing confidentiality guarantees in processing, along with formal proofs of security. However, these constructions are still limited in terms of computational and communication costs and require expert

knowledge to be applied. This has limited the adoption of these techniques by the practitioners of specific fields, particularly in state-of-the-art models. In the case of speech processing, few examples exist of complex state-of-the-art pipelines being implemented with cryptographic techniques [109, 339]. Nevertheless, this is a rapidly evolving field of research, with recent works showing that it is becoming possible to implement large neural networks with state-of-the-art architectures [5, 170, 181, 233].

1.3.2 Privacy-oriented manipulation

As the state-of-the-art in speech technology advances, so does the complexity of the deep learning systems behind it. This trend is not exclusive to speech, as similar phenomena can be observed in areas such as image processing and natural language processing [317]. All the while, as stated above, the complexity of private systems based on underlying cryptographic constructions is limited by computational and communication costs, making their adoption for state-of-the-art speech systems challenging. For this reason, machine learning-based privacy-oriented speech manipulation methods have received increased attention as more efficient and flexible alternatives to the problem of privacy. These methods focus on removing, disentangling or obfuscating sensitive information, such as the speaker's identity [300, 320], or discrete traits such as the speaker's sex or perceived emotions [8, 222], while attempting to limit changes in other aspects of the signal to guarantee target-task utility.

The versatility of this type of approach stems from the fact that only specific aspects of the speech signal are privatised, allowing a conscious trade-off between the information that is disclosed to the service provider and the information that should remain hidden. This family of methods also differs from cryptographic methods by the fact that it does not provide formal privacy guarantees. Instead, privacy-oriented manipulation methods mostly base their measures of privacy on empirical and information-theoretic evidence. These solutions are also more user-centred, as the privatisation process is applied directly in the user's device, giving them direct control over the privacy of their information. Moreover, as opposed to cryptographic processing, privacy-oriented speech manipulation methods are independent of the complexity of the downstream task, which allows the server to apply state-of-the-art methods over the user's data.

Techniques used in privacy-oriented speech manipulation approaches for speech privacy can be broadly grouped into four categories: voice anonymisation, speaker information minimisation, adversarial examples and privacy-aware feature extraction.

Probably the most common branch of privacy-oriented speech manipulation, voice anonymisation focuses on modifying the speech signal such that the original speaker cannot be identified, i.e., removing speaker identity, while keeping linguistic and paralinguistic content intact [318]. This type of approach is particularly suited to settings where the speech signal is sent to a remote server to be transcribed or used as part of a dataset to train speech-based machine learning models. Research in this area is in part

motivated by Recital 26 of the GDPR (cf. Appendix A, Section A.5), which states that *anonymised data* does not fall within the scope of the regulation. Anonymising speech data thus allows companies to store and process user data, ensuring that this is done within the regulatory framework.

Building on earlier efforts [18, 91, 300], the work on voice anonymisation has been significantly pushed forward by the Voice Privacy Challenges held in 2020 and 2022 [318, 319]. These challenges introduced a standardised benchmark for evaluation and two baseline systems that have been the foundation of a large number of subsequent works on speaker anonymisation. The first baseline leverages a machine learning approach, where the speech signal is decomposed into three streams of information representing linguistic features, pitch and speaker information. To anonymise the signal, the speaker information is replaced with that of a pseudo speaker, and the signal is reconstructed. The second baseline leverages conventional signal processing techniques to move the relative position of speaker formant frequencies to change the speaker's voice in the reconstructed signal [240]. Besides these two baselines, the applicability of voice conversion techniques to the problem of voice anonymisation has also been investigated [301, 359].

The challenge's first baseline has proven to be most effective in both anonymisation and naturalness [320], however, it has also been shown to be vulnerable to attacks that leverage the invertibility of the speaker representation anonymisation strategy [48, 231], a problem that has been tackled by the generation of synthetic speaker embeddings [193, 325]. Sub-par performances in terms of anonymisation have also been linked to the speaker information contained in linguistic features, with several methods having been proposed to overcome this limitation, such as feature quantization [245], differentially private noise [285], and higher-level phonetic representations [193]. Recent improvements over this baseline have also focused on quality and multilinguality through the use of self-supervised speech representations [194] and generative models [230].

A related and complementary area of research is the development of methods for speaker information minimisation. This is a more fine-grained approach that aims to manipulate or remove specific speaker traits that are considered sensitive from the speech signal or a representation thereof while keeping the remaining information intact [8, 222, 244]. This family of methods abides by the *data minimisation* principle, also contemplated in the GDPR, whereby personal data shall be "adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed" as described in Article 5 (c) of the GDPR (cf. Appendix A, Section A.4). These methods often work through adversarial training, Generative Adversarial Networks (GANs), or neural network architectures that inherently disentangle the signal in intermediate representations (e.g., Variational Autoencoders (VAEs)) [8, 222, 244, 304, 351]. Speaker information minimisation methods have been shown to provide good results in terms of utility. However, it has also been shown that they do not entirely remove sensitive information. Adversarial methods often re-arrange information in the signal to

fool downstream (adversarial) classifiers (even when the objective is the removal of information). Consequently, if an adversary gets hold of a moderate amount of labelled transformed samples, they can train a new classifier that will be able to classify the sensitive information, removing the effect of the adversarial transformation [300, 350, 351]. For this reason, these methods require extensive empirical validation to show that information is being removed.

Adversarial examples correspond to data points to which a perturbation has been added such that, when fed to a machine learning classifier, the model outputs a different result than if it had been applied over the original, non-perturbed input [113, 308]. This change can be *targeted*, meaning the adversarial classifier outputs a specific class or prediction, or *untargeted*, where the signal is changed just so that it can change the classifier's prediction to any other class [44]. Although viewed most often as a measure of the robustness of these classifiers, adversarial examples can also be leveraged for privacy. For example, an adversarial example created to fool an attribute classifier may be enough to prevent an intrusive but unaware service provider from learning the real attribute [329], the identity of the speaker [155, 176], or even to correctly transcribe the speech signal [226]. However, adversarial examples do not remove information from the signal and often do not generalise between classifiers. This means that the use of a different classifier or classification strategy is often enough to overcome the adversarial example and obtain the original result [329]. Nevertheless, for unaware attackers, adversarial examples may be sufficient to provide some level of privacy protection.

A final alternative is *privacy-aware feature extraction*. In this type of approach, instead of focusing on removing specific subsets of information, the goal is to extract particular sets of features that leak little information outside that of the target task. The methods used for this type of approach are similar to those mentioned above for attribute-based privacy, consisting mainly of adversarial approaches such as GANs, variational inference and Siamese Neural Networks [152, 213–216].

When compared to cryptographic approaches, in the case of remote speech processing, privacy-oriented speech manipulation methods offer three main advantages: the first is that the user is given a choice to employ these models or not, and this choice does not demand extra steps from the service provider, being independent of the downstream task; the second advantage is the fact that these methods are particularly lightweight when compared to cryptographic protocols; finally, the adoption and development of these methods is much easier for speech practitioners, as these methods rely on concepts and methods that are commonly used in speech research. Nevertheless, as stated above, privacy-oriented speech manipulation methods are not based on any formal privacy guarantees. Moreover, except for voice anonymisation, privacy evaluation protocols are not standardised, making it hard to properly assess the degree of privacy provided by a given method. In addition, privacy-oriented manipulation methods do not provide confidentiality guarantees, as some of the information contained in the input speech signal will always remain present and unchanged.

1.4 Thesis Statement

The main goal of this thesis is to contribute to ongoing efforts for privacy in remote processing. To this end, we focus on two approaches: cryptographic processing and privacy-oriented speech manipulation. Even though cryptographic processing has been previously explored in speech biometric scenarios, its applicability to other speech processing pipelines remains an open question despite its strong privacy guarantees. Moreover, outside privacy-aware feature extraction (cf. Section 1.3.2), cryptographic techniques seem to be the most adequate solutions to achieve privacy in tasks whose target is intrinsically related to the acoustic content of the speech signal, making it hard to disentangle potentially private information from task-related information. This is the case for tasks such as Speaker Recognition and Automatic Speaker Diarization (ASD), or tasks where the speech signal is analysed with regard to a specific characteristic or condition (e.g. diagnosis of speech-affecting diseases, emotion recognition).
For these reasons, in this thesis, we study the applicability of these techniques to speech processing and explore the following research question:

1. *Can cryptographic techniques guarantee usable privacy in remote speech processing? Specifically, how feasible is the implementation of state-of-the-art speech classifiers and deep learning architectures using cryptographic techniques, and what are the necessary trade-offs between privacy, computational and communication costs, and model performance to do so?*

Although feasible, cryptographic methods incur very high computational and communication costs. Hence, we explored alternative techniques for use cases where it would not be possible to implement them. This prompted the exploration of privacy-oriented speech manipulation methods as a second research topic. As mentioned in Section 1.3, privacy-oriented manipulation methods are much more versatile than cryptographic methods and are independent of the complexity of downstream classifiers, even though their use requires sacrificing the strong privacy guarantees provided by cryptographic methods.
In this line of research, we aim to answer the following research question:

2. *Are speech manipulation methods suited to privacy-preserving remote speech processing? If so, how can we measure privacy in these methods? Is it possible to achieve different levels of privacy?*

1.5 Contributions

In this document, we present the work that was completed during this thesis.
Concerning the first research question, we explored three approaches applied to different levels of target task complexity. As a first approach, we built on our previous work [313, 314] and proposed a method

for privacy-preserving Support Vector Machine-based classification applied to the detection of two speech-affecting diseases: Obstructive Sleep Apnea and Parkinson's disease. This method is based on a combination of HE, SMC and Secure Modular Hashing (SMH). The second work developed in this direction corresponds to a privacy-preserving implementation of the extraction of speaker representations for Automatic Speaker Verification (ASV), using SMC. The goal of this method was to add to existing privacy-preserving techniques for speech biometric verification and privatise this important step in the verification pipeline. Our third work in this direction corresponds to the privacy-preserving implementation of an Automatic Speaker Diarization pipeline through a combination of SMC and SMH. This work is meant as a proof-of-concept on how feasible it is to implement a complex speech processing pipeline with cryptographic techniques.
The work completed in this direction resulted in the following publications:

- **Teixeira, F.**, Abad, A., Trancoso, I. and Raj, B., "Voice Biometrics: Privacy in Paralinguistic and Extra-Linguistic Tasks", in Chapter 4, Voice Biometrics: Technology, trust and security, C. Garcia-Mateo and G. Chollet, Eds. ISBN: 978-1-78561-900-7, IET, 2021 [315];
- **Teixeira, F.**, Abad, A., Raj, B., Trancoso, I., "Towards end-to-end private Automatic Speaker Recognition", Proc. Interspeech 2022, 2798-2802, doi: 10.21437/Interspeech.2022-10672, 2022 [310];
- **Teixeira, F.**, Abad, A., Raj, B., Trancoso, I., "Privacy-preserving Automatic Speaker Diarization", ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10096113, 2023 [311].

Two different approaches were explored concerning the second research question. In the first, we focused on the creation of highly imperceptible adversarial examples against speaker identification. While this work is framed as an adversarial attack on speaker identification – its primary goal – the proposed method could potentially be extended to protect speakers from the automatic collection of recordings of their voices that could be used against them. This work was the outcome of a collaboration with researchers from the Queen Mary University of London and resulted in the following publication:

- Shamsabadi, A. S.*, **Teixeira, F.***[3], Abad, A., Raj, B., Cavallaro, A. and Trancoso, I., "FoolHD: Fooling speaker identification by highly imperceptible adversarial disturbances", in ICASSP 2021 - International Conference on Acoustics, Speech and Signal Processing, pp. 6159–6163, 2021 [286].

Although this work was not directed at privacy, its development allowed us to obtain a better understanding of how privacy-oriented speech manipulation methods should be developed and evaluated. For this reason, it was decided that this work should be included in this thesis.

[3]Shared first-authorship.

In turn, this led to the proposal of a new privacy-oriented speech manipulation method for the removal of speaker age and sex information from speaker representations, which has resulted in the following publication:

- **Teixeira, F.**, Abad, A., Raj, B., Trancoso, I.,. "Privacy-Oriented Manipulation of Speaker Representations", in IEEE Access, vol. 12, pp. 82949-82971, doi: 10.1109/ACCESS.2024.3409067, 2024 [312].

Outside the two main topics of this thesis, we additionally explored Membership Inference (MI). As stated in Section 1.2, Membership Inference (MI) is one possible attack on deployed machine learning models. However, it can also be used as an auditing tool to assess the proper use of customer data when training ML models. As such, and as a final contribution of this thesis, we explore the use of MI as a tool to audit the training data of Automatic Speech Recognition (ASR) models. This work resulted from the collaboration with researchers from Carnegie Mellon University and the Technical University of Munich and has been published as:

- **Teixeira, F.***[4], Pizzi, K.*, Olivier, R.*, Abad, A., Raj, B. and Trancoso, I., "Improving Membership Inference in ASR Model Auditing with Perturbed Loss Features", Trustworthy Speech Processing, Satellite Workshop, ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).

In addition to the above, we have also actively collaborated with other researchers in the following review article:

- Nautsch, A., Jiménez, A., Treiber, A., Kolberg, J., Jasserand, C., Kindt, E., Delgado, H., Todisco, M., Hmani, M. A., Mtibaa, A., Abdelraheem, M. A., Abad, A., **Teixeira, F.**, Matrouf, D., Gomez-Barrero, M., Petrovska-Delacrétaz, D., Chollet, G., Evans, N., Schneider, T., Bonastre, J.F., Raj, B., Trancoso, I., Busch, C. "Preserving Privacy in Speaker and Speech Characterisation", Computer Speech & Language, vol. 58, pp. 441–480, 2019 [210].

Part of the original motivating factors for this thesis were the privacy implications of the use of speech data as a biomarker for health, which, as stated above, is the focus of one of the chapters of this thesis. This was the reason for the author's strong involvement in the research group activities related to health, having participated in the following published works:

- Mendonça, J., **Teixeira, F.**, Trancoso, I., Abad, A. (2020) Analysing Breath Signals for the Interspeech 2020 ComParE Challenge. Proc. Interspeech 2020, 2077-2081, doi: 10.21437/Interspeech.2020-2778, 2020 [190];

[4]Shared first-authorship.

- Solera-Ureña, R., Botelho, C., **Teixeira, F.**, Rolland, T., Abad, A., Trancoso, I. "Transfer Learning-Based Cough Representations for Automatic Detection of COVID-19". Proc. Interspeech 2021, 436-440, doi: 10.21437/Interspeech.2021-1702, 2021 [294];
- J. Correia, **F. Teixeira**, C. Botelho, I. Trancoso and B. Raj, "The in-the-Wild Speech Medical Corpus," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6973-6977, doi: 10.1109/ICASSP39728.2021.9414230, 2021 [63].

1.6 Outline

This thesis is structured in 9 chapters.

- Chapter 1 motivates the topic and introduces concepts that are fundamental to understanding the remainder of the document.
- Chapter 2 provides fundamental cryptographic concepts to allow non-experts to understand methods described later in the document. Anticipating an audience with a speech-processing background, speech-processing concepts are assumed to be familiar to the reader and thus will not be covered. For interested readers, we recommend the following works for in-depth introductions to speech processing [121, 261].
- Chapters 3, 4 and 5 present the work that has been conducted on cryptographic methods for speech processing. Specifically:
  - Chapter 3 presents the work that was conducted on privacy-preserving classification of speech affecting diseases;
  - Chapter 4 targets the privacy-preserving extraction of speaker representations for ASV;
  - Chapter 5 details the work that was done on privacy-preserving ASD.
- Chapter 6 includes the work developed for adversarial examples against speaker identification;
- Chapter 7 contains the work done using privacy-oriented speech manipulation methods, focusing on the removal of *sex* and *age* information from speaker representations.
- Chapter 8 comprises the work that was developed concerning membership inference in ASR.
- Chapter 9 presents a discussion of the work conducted in this thesis, its main conclusions, and an overview of possible future research directions.

This thesis also includes three appendices: Appendix A, which includes definitions taken from the EU's GDPR that help support the motivation of this work, Appendix B, which provides supplementary

material related to the introduction to Secure Multiparty Computation presented in Chapter 2, and Appendix C, which provides additional details regarding the mathematical bases of the mutual information estimators used in Chapter 7.

2

Background

In this chapter, we introduce fundamental concepts of cryptographic primitives and protocols such as Homomorphic Encryption (HE), Secure Multiparty Computation (SMC), as well as Limited-leakage Hashing (LLH), which will allow the reader to understand the methods described in Chapters 3, 4, and 5.

It is important to note that we will not address neither Federated Learning nor Differential Privacy techniques in this chapter. Although important for decentralised learning of machine learning models, Federated Learning [150, 177] algorithms are complementary to privacy-preserving inference techniques but do not function as alternatives. Differential privacy [82, 140] is also most often used as a tool during model training to guarantee that the resulting model provides privacy guarantees with regard to its training data, particularly when combined with Federated Learning. Nevertheless, for speech, it can also be used as a way to randomise specific speaker attributes [60], including speaker identity [285], when these are being to hide the contributions of individual data providers. Another line of research we will not cover is remote inference based on hardware-based security, such as Intel's Security Guard Extensions (SGX) enclave [64]. For in-depth descriptions of these techniques, we direct the reader to [64, 82, 140, 177]. For works that apply these techniques to speech, we refer the reader to [37, 95, 347, 355].

This chapter is organised as follows: Section 2.1 introduces and describes HE; Section 2.2 provides an introduction to SMC protocols and their security models; Section 2.3 describes LLH techniques; finally, Section 2.4 provides a summary of the chapter.

2.1 Homomorphic Encryption

Homomorphic Encryption (HE) is a family of cryptosystems, under which operations performed over *ciphertexts* (i.e. encrypted values) are *homomorphic* with regard to the *plaintexts* (i.e. unencrypted values). In other words, considering the encryption of a value x, $E(x)$ and of a value y, $E(y)$, if a homomorphic operation is performed on the two ciphertexts, the result of this operation will correspond to the equivalent unencrypted operation of the two values, as follows:

$$\begin{aligned} E(x) \otimes E(y) &= E(x \times y), \\ E(x) \oplus E(y) &= E(x + y), \end{aligned} \tag{2.1}$$

Homomorphic Encryption (HE) serves as a building block for many secure protocols, having applications in a wide variety of areas such as data mining, forensics, financial privacy and medicine [14]. Homomorphic Encryption (HE) techniques can be roughly divided into three categories: Partially Homomorphic Encryption (PHE), Somewhat Homomorphic Encryption (SHE) and Fully Homomorphic Encryption (FHE). The former, PHE schemes, are limited by the type of operations that can be performed, while SHE schemes are constrained in the number of times each operation can be

performed over a ciphertext. In FHE schemes, neither of these restrictions apply, and all the allowed operations (usually additions and multiplications) can be performed an unlimited number of times.

Partially Homomorphic Encryption (PHE) schemes such as Paillier [228], and additively and multiplicatively homomorphic cryptosystems like RSA [274] and ElGamal [84], have been extensively used in the literature for privacy-preserving applications.

The first FHE scheme, proposed by Craig Gentry in 2009 [107], introduced *bootstrapping* – i.e., a technique that allows the decryption and re-encryption of a *ciphertext* using HE – as a way to allow for unlimited operations to be performed over ciphertexts. Although this technique was highly inefficient at the time, since then, FHE schemes have been constantly improved, and there are currently much faster implementations of HE using bootstrapping [51, 55].

Nonetheless, FHE implementations are still computationally heavy. For most applications, however, it is not necessary to perform an unlimited number of operations over ciphertexts, as the user knows beforehand the number of operations that will be performed. In these cases, SHE schemes such as Brakerski-Gentry-Vaikuntanathan (BGV) [36], Brakerski/Fan-Vercauteren (BFV) [35, 90] and more recently, Cheon-Kim-Kim-Song (CKKS) [54], allow the user to select the scheme's encryption parameters such that the number of permitted operations over a *ciphertext* matches that of the computation. However, if the number of operations is greater than the pre-established limit, the underlying value's decryption becomes meaningless. In order to perform more operations, it is necessary to use larger encryption parameters, which results in more expensive computations. Hence, in these schemes, one needs to trade-off between computational complexity and multiplicative depth. To compensate for this limitation, SHE cryptosystems like BGV, BFV and CKKS encompass *batching* techniques that allow several messages to be encrypted in the same ciphertext and thus to be operated as Single Instruction Multiple Data (SIMD), effectively reducing the scheme's computational cost [36].

Most SHE schemes also provide the capability of computing operations between ciphertexts and *plaintexts* (i.e., non-encrypted values) at a much lower cost, both computationally and in terms of multiplicative depth, when compared to ciphertext-ciphertext operations [28]. This characteristic is essential for remote processing applications, as this allows the holder of a model to perform operations between the weights of their model and the user's data at a much lower cost than if it was necessary to perform these operations between ciphertexts.

The security of HE schemes stems from their underlying computational hardness assumptions. In particular, SHE schemes, such as BFV and CKKS, derive their security from the Learning With Errors (LWE) [268] and Ring Learning With Errors (RLWE) problems [183], which are assumed to be post-quantum secure [7].

Due to the characteristics mentioned above, the schemes cited here, along with their variations, are currently the prominent choice for privacy-preserving remote speech processing that leverages the use of

HE to build privacy-preserving applications [47, 108, 117, 136]. In this thesis, HE and, in particular, the CKKS scheme, is used in Chapter 3 as part of the privacy-preserving computation of an Support Vector Machine (SVM) for speech-affecting disease detection.

2.2 Secure Multiparty Computation

Secure Multiparty Computation (SMC) is an umbrella term for protocols designed to allow several parties to jointly and interactively compute a function over their data while keeping all inputs private. Among others, SMC protocols range from general purpose primitives such as Oblivious Transfers (OTs) to high-level protocols such as Shamir's Secret Sharing and Arithmetic and Boolean Secret Sharing (i.e., Goldwasser-Micali-Wigderson (GMW)) [25, 110, 284] and Yao's Garbled Circuits (GCs) [293, 356].
Secure Multiparty Computation (SMC) protocols combine underlying cryptographic constructions such as public-key encryption, symmetric encryption, HE, or even other SMC protocols such as OTs to perform specific functionalities under different levels of security, computational and communication costs [38, 70, 178, 217].

2.2.1 Oblivious Transfers

Oblivious Transfers (OTs) are general-purpose cryptographic primitives, being a basic building block for many SMC applications [142, 217], that allow two distrusting parties, a sender $\mathcal{S}$ and a receiver $\mathcal{R}$, to exchange data in a private setting. In 1-out-of-n OT, $\mathcal{S}$ possesses n messages, one of which $\mathcal{R}$ is interested in but does not want $\mathcal{S}$ to find out which, whereas $\mathcal{S}$ is willing to give $\mathcal{R}$ one of the messages, but does not want $\mathcal{R}$ to learn anything about the other messages [59, 280]. A particular case of this formulation, 1-out-of-2 OT, where there are only two messages for $\mathcal{R}$ to choose from, was first proposed by Rabin, M. in 1981 [223]. Optimisations such as OT pre-computations [24] and OT extensions [125] have to be highlighted due to the critical role they play in improving the efficiency of OT-based SMC protocols.
These protocols are generally built over asymmetric (or public-key) cryptography or combinations of asymmetric and symmetric cryptographic primitives [125]. As such, the security of OT protocols is dependent (though not uniquely) on the security of the underlying cryptographic primitives.

2.2.2 Yao's Garbled Circuits

First proposed by Yao, A. et al. in 1986 [356], Garbled Circuits (GCs) are a cryptographic construction that allows two parties, Alice and Bob, to jointly compute a function represented as a boolean circuit, such that their inputs, as well as any intermediate results, are kept private. Only the function's output

is revealed to one or both parties. This construction requires each party to take a role; one must be the *garbler*, which we will assume is Alice, and a second party, in this case, Bob, will be the *evaluator*.

a	b	$z = a \oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

(a) Original truth table.

a	b	$z = a \oplus b$
k_a^0	k_b^0	$\varepsilon_{k_a^0}(\varepsilon_{k_b^0}(k_z^0))$
k_a^0	k_b^1	$\varepsilon_{k_a^0}(\varepsilon_{k_b^1}(k_z^1))$
k_a^1	k_b^0	$\varepsilon_{k_a^1}(\varepsilon_{k_b^0}(k_z^1))$
k_a^1	k_b^1	$\varepsilon_{k_a^1}(\varepsilon_{k_b^1}(k_z^0))$

(b) Garbled truth table.

Figure 2.1: XOR Gate truth tables.

Consider an XOR gate (cf. Table 2.1a), which contains three wires, two inputs and one output. For each of these wires, Alice chooses two random values, one for each bit, obtaining six encryption keys: $k_a^0, k_a^1, k_b^0, k_b^1, k_z^0, k_z^1$. Alice then uses the keys to encrypt the output of each row, using a symmetric encryption scheme, obtaining a garbled truth table (cf. Table 2.1b). Now, we require a way for Bob to *evaluate* the circuit without learning Alice's inputs. Since each bit is a random value, Bob will not be able to learn anything from them. However, the rows of the table need to be permuted so that Bob cannot correspond them with the rows of the original function. Consequently, Alice must also permute the rows of her GC table. Afterwards, Alice can send Bob the encrypted output column of the GC table, as well as her input, which is a random value, and, as such, Bob will not be able to determine the bit it corresponds to. The next step in this process is for Bob to receive his input values from Alice. To this end, Alice and Bob perform a 1-out-of-2 OT. Therefore, Alice does not learn which input bit Bob selected, and Bob will only be able to decrypt the key corresponding to his chosen bit and will not be able to learn anything about the other key. Bob is now able to decrypt the output of the circuit using his and Alice's input keys. Depending on the implementation, this process can occur in different ways. For simplicity, we will assume that, after decryption, it is possible to distinguish the correctly decrypted value from the values decrypted using incorrect keys. Finally, Bob can either keep his output result or share it with Alice. For a more complex circuit, the protocol defined above can be generalised for all gates in the circuit, taking into account that intermediate results will also be keys that will serve as inputs for the subsequent gates.

We can thus summarise Yao's GC protocol as follows [210]:

1. The *garbler*, *Alice*, transforms the function f to be computed into a boolean circuit and generates keys for all wires and gates of the circuit.

2. *Alice* sends the permuted garbled tables and the keys corresponding to their inputs to the *evaluator*, *Bob*.

3. *Bob* obtains his inputs through OT with *Alice* and evaluates each gate using his keys, as well as

Alice's keys.

4. Finally, *Bob* reveals the output of the circuit and decides whether to share it or not with *Alice* according to what was agreed upon by both parties beforehand.

Since it was first proposed, this protocol has been subject to several optimisations, including the *point-and-permute* optimisation, which only requires one decryption per gate, the *half-gates* optimisation which reduces the bandwidth required to compute AND gates [246, 361] and the *free-XOR* technique, which allows the computation of XOR gates without communication and at a very low computational cost [149]. Besides being able to model any function, one of the advantages of Yao's GC protocol is the fact that it requires a constant number of rounds of communication.
In this thesis, GCs will be used in Chapter 3 to compute a non-linear function in the privacy-preserving implementation of an SVM for the classification of speech-affecting diseases.

2.2.3 Secret Sharing

Secret Sharing is a family of protocols that allow parties to represent and share their data with other parties and to interactively compute any operation over their secret data. In Secret Sharing, data is represented in such a way that each of the parties participating in the computation will only have access to a random-looking *share* (here denoted as $\langle \cdot \rangle$) of the original value, being unable to observe the true underlying data. When a value is represented in this way, a single party is not able to reconstruct the *secret* without – at least a subset of – the remaining parties.
Secret sharing schemes have the advantage of being much lighter in terms of computational cost when compared to HE. However, they require the online presence of all the parties involved in the computation and usually have higher costs in terms of communication, as they often require multiple rounds of interaction.
This family of protocols allows the computation of any function, including additions and multiplications and their Boolean counterparts, XOR and AND operations (depending on the base representation of the protocol), and any non-linear function. The protocols used to compute each operation are composable, and there is no limit to the number of operations that can be performed. Moreover, performing more operations does not increase the cost of each operation, as was the case with HE.
Although there exist multiple possible secret-share representations (e.g., notably Shamir's Secret Sharing scheme [284]), we will focus here on additive secret-sharing schemes that are based on the GMW protocol, proposed by Goldreich, O. et al. in 1987 [110], as this representation is the basis for the protocols used in the experiments that will be presented later in this thesis. In particular, additive secret sharing will be used in Chapters 3, 4 and 5 for the privacy-preserving classification of speech-affecting diseases, the private extraction of speaker representations and the privacy-preserving implementation of an ASD pipeline, respectively.

2.2.3.A Additive Secret Sharing

In the general n-party case, a value x, in an additive secret sharing scheme, shared among several parties by a dealer, is defined as:

$$x = \langle x \rangle_1 + \langle x \rangle_2 + ... + \langle x \rangle_n, \tag{2.2}$$

where $\langle x \rangle_1$, ..., $\langle x \rangle_n$ represent random-looking shares of x held by each party, and $+$ can represent either an addition or an XOR, depending on whether arithmetic (addition) or boolean (XOR) sharing is being considered. In fact, the boolean variant of this representation corresponds to the original GMW protocol proposed by Goldreich et al. in 1987 [110]. Each share is generated as $\langle x \rangle_n = x + \sum_{i=1}^{n-1} s_i$, where each s_i is chosen uniformly at random.

Due to their associative property, with this representation, additions can be computed locally by each party. In other words, it can be shown that adding two secret-shared values corresponds to each party adding the shares they hold for the two values, without the need for communication with the other parties[1]. This property makes it such that Additive Secret Sharing protocols have negligible computational and communication costs regarding the computation of additions and subtractions. This representation also allows the computation of multiplications, but these are more expensive in terms of computation and communication, requiring specific sub-protocols to be performed.

2.2.3.B Multiplication Triples

As mentioned above, multiplications require specific constructions to be computed within a secret-sharing protocol. The original GMW protocol relied on an OT-based sub-protocol to perform multiplications [110]. However, *Beaver Triples*, or *Multiplication Triples (MTs)* [23] have since become the standard for secret sharing multiplications.

These values are shares of the form $\langle a \rangle$, $\langle b \rangle$ and $\langle c \rangle$, where $\langle c \rangle = \langle a \rangle \times \langle b \rangle$. To perform a multiplication between shared values x and y, each party sets its shares to $\langle e \rangle_i = \langle x \rangle_i - \langle a \rangle_i$ and $\langle f \rangle_i = \langle y \rangle_i - \langle b \rangle_i$ and exchanges the results with the other parties, so that each party holds e and f. The resulting share is given by [76]:

$$\langle z \rangle_i = e \cdot f + f \cdot \langle a \rangle_i + e \cdot \langle b \rangle_i + \langle c \rangle_i \tag{2.3}$$

It can then be shown that by adding all z_i we obtain $x \times y$. [2]

[1] A short proof can be found in Appendix B, Section B.1.
[2] A detailed proof is provided in Appendix B, Section B.2.

2.2.3.C Replicated Secret Sharing

The secret-sharing construction described above works for any number of parties greater than or equal to two. However, for a number of parties strictly larger than two, it is possible to instantiate more efficient schemes. Replicated Secret Sharing (RSS) schemes [13] are such an example. While in additive secret sharing, each party holds a single share per value in the computation, with RSS, each party holds a set of shares per value.

Considering, for instance, the three-party case and a shared value $y = \sum_{i=1}^{3} \langle y \rangle_i$, party p_1 will hold shares $\langle y \rangle_1, \langle y \rangle_2$, party p_2 will hold shares $\langle y \rangle_2, \langle y \rangle_3$ and party p_3 will hold shares $\langle y \rangle_3, \langle y \rangle_1$. In this case, computing additions will work as before, and each party can perform the operation locally. Multiplication, on the other hand, may work differently.

A possible implementation of the multiplication operation would be for each party to locally multiply the shares it holds for each of the secret shared values. In this way, party p_1 will obtain $z_1 = \langle x \rangle_1 \langle y \rangle_1 + \langle x \rangle_1 \langle y \rangle_2 + \langle x \rangle_2 \langle y \rangle_1$; party p_2, $z_2 = \langle x \rangle_2 \langle y \rangle_2 + \langle x \rangle_2 \langle y \rangle_3 + \langle x \rangle_3 \langle y \rangle_2$ and party p_3, $z_3 = \langle x \rangle_3 \langle y \rangle_3 + \langle x \rangle_3 \langle y \rangle_1 + \langle x \rangle_1 \langle y \rangle_3$. As above, it can be shown that adding the resulting shares will yield the correct result. Still, at the end of the computation, each party only holds a single share of the value, and a *re-sharing* protocol is required so that each party holds the same set of shares as before [336]. This implementation is described with regard to arithmetic operations but also holds for binary computations such as those in the GMW protocol.

2.2.3.D Domain conversion

Performing operations in either the arithmetic or boolean domains may prove to be more efficient for different operations or may even allow performing different functionalities. This makes it helpful to alternate between protocols within the same computation. For instance, non-linear operations (e.g., neural network activation functions) cannot be computed in the arithmetic domain and require converting shares between the arithmetic and boolean domains.

Depending on the SMC protocol, the conversion between domains may take different forms. In some protocols, it is possible to perform the conversion locally, with minimal interaction between parties [69, 141]. However, other protocols may need to use pre-computed values that are shared in both domains, such as *daBits* [276] or *edaBits* [86].

2.2.3.E Fixed-point numbers

An important detail of Secret Sharing schemes is the fact that secret shared values are integers or binary values, whereas, for most real-world applications, values are floating-point numbers. While floating-point representations exist within SMC, fixed-point representations are much more efficient. An example of a fixed-point representation, for the arithmetic domain, is that of [141], where a value x is

represented as $x = y \cdot 2^f$, where y is an integer, and f is the fixed precision. While this approximation does not affect additions, for multiplications, one needs first to multiply the two integers and then truncate by f. This can be implemented as a binary left shift operation [45] or via probabilistic truncation [46, 68, 86].

2.2.4 Offline vs Online phases

The generation of *Multiplication Triples*, *daBits*, *edaBits*, as well as other auxiliary secret shares, requires the participation and interaction of the parties involved in the computation. However, since the generation of these auxiliary shares is not dependent on input data, this step can be moved to what is called an *offline* or *pre-processing* phase. This phase can be performed at any time before the *online* data-dependent phase. Many protocols are hence designed to have the most expensive operations within the *offline* phase, making the *online* phase much more efficient.

2.2.5 Security and computational performance

The shared nature of SMC protocols demands that threat assumptions are made about the parties participating in the computation. The threat model of an SMC protocol is critical as it significantly affects its security and computational performance and, thus, its range of applications. The most common security (or threat) models include the *honest-but-curious* adversary model (also called *semi-honest* adversary or *passive* security) and the *malicious* adversary model (or *active* security).
The *honest-but-curious* model is the simplest model possible, and it is considered to be sufficient for most applications [76, 136, 179, 273]. In this model, the adversaries are assumed to follow the established protocol but are also assumed to pry into data that is visible to them. In this way, there is no need to create additional safeguards outside of the protocol's inherent security, allowing for very efficient implementations. The *honest-but-curious* model is used in applications where all parties are trustworthy (e.g. interaction between hospitals or clinics and companies).
The *malicious* model assumes that adversaries will attempt to thwart the protocol, demanding additional proof that each party is behaving correctly. This can be done in different ways, depending on the protocol and phase of the computation, through Zero-Knowledge proofs [22], cut-and-choose methods [102], and Message Authentication Codes [65, 70, 141], among others. This threat model should be used in settings where parties do not trust each other (e.g. two competing companies that need to perform a computation over their private data). Although more secure, this model significantly increases the computational cost of SMC protocols [65, 70].
Besides the behaviour of individual parties, one can also define the security of the protocol in terms of whether a *majority* of the parties will behave correctly or not – *honest majority* vs *dishonest majority*, and whether a subset of parties might collaborate – or *collude* – to obtain more information than they

are allowed to. If a majority of parties are assumed to be honest, protocols that take into account *malicious* behaviour can be made much more efficient [69, 102].

The highest possible level of security is achieved by assuming malicious adversaries in a dishonest majority. However, this comes at very high computational and communication costs. More complex models exist that take other assumptions into account, e.g., whether the adversary changes behaviour during the protocol's execution and what is the maximum number of corrupted parties allowed before the protocol can no longer be executed securely. However, these fall out of the scope of this chapter, and we instead direct readers who would like to learn more about this topic to the following: [89, 178].

2.3 Limited-leakage Hashing

Limited-leakage Hashing (LLH) is a family of hash functions $H(\cdot)$ that guarantee that, if two vectors are close enough in the input space, the distance between their hashes will be proportional to their distance in the original space. Contrarily, if the two vectors are far apart in the input space, the distance between their hashes will not provide any meaningful information about the distance in the original space [33, 133, 252].

This guarantee makes LLH functions useful in privacy-preserving nearest-neighbour and template comparison applications, where some information is allowed to be leaked – i.e., one can determine the distance between a pair of vectors if they are close and will receive no information otherwise.

In the remainder of this section, we will detail two LLH functions: Secure Binary Embeddings (SBE) and Secure Modular Hashing (SMH). Limited-leakage Hashing (LLH), and specifically SMH, is used in the privacy-preserving methods proposed in Chapters 3 and 5.

2.3.1 Secure Binary Embeddings

Secure Binary Embeddings (SBE) [33] are a type of LLH function that uses band-quantised random projections to convert real-valued vectors into bit sequences, providing information-theoretic security guarantees. This quantisation scheme is based on Universal Scalar Quantisation [34], a quantisation scheme where the quantisation function has non-contiguous bands, making it more efficient when encoding information. SBE is also based on Locality-Sensitive Hashing (LSH) [72], a family of hash functions that guarantee that if two vectors are close enough in the input space, they will hash to the same output value. On the contrary, if their distance is over a threshold, they will hash to different values.

For a vector x with L dimensions, the SBE transformation is a random projection from $\mathbb{R}^L$ into $\mathbb{Z}_2^M$, defined as:

$$Q_{SBE}(x) = \lfloor \Delta^{-1}(Ax + w) \rfloor (\text{mod } 2), \tag{2.4}$$

where Δ is a diagonal matrix with diagonal values equal to δ, $A \in \mathbb{R}^{L \times M} \sim N(0, \sigma^2)$ is a random matrix, and $w \in \mathbb{R}^M \sim \text{unif}[0, \delta]$ is the additive dither; the size of the output hash vector is usually defined in terms of the number of input dimensions as $M = L \times mpc$, where mpc is the number of *measurements per coefficient*.

Boufounos et al. [33], show that their scheme provides information-theoretical security by proving that the mutual information between two hashed vectors $\mathbf{q}, \mathbf{q}'$, obtained from two vectors $\mathbf{x}, \mathbf{x}'$ using equation 2.4, is bounded by:

$$I(\mathbf{q}, \mathbf{q}'|d) \leq 10Me^{-(\frac{\pi\sigma d}{2\delta})^2}, \tag{2.5}$$

where d is the Euclidean, or l_2, distance between $\mathbf{x}, \mathbf{x}'$. Analysing equation 2.5, it is possible to infer several properties of this transformation. In particular, we can observe that the upper bound on the mutual information between the hashes of the two vectors decays exponentially with d, with this rate being controlled by the precision parameter δ. As such, for vectors that are too far apart, comparing the distance between their hashes will not provide meaningful information. Moreover, increasing δ slows the rate of decay and, consequently, increases the maximum d for which the comparison between the hashed vectors is still meaningful. On the other hand, increasing M also affects the upper bound of the mutual information, but it is much less significant than varying δ.

The SBE quantisation function maps real vectors into binary vectors. As such, to compare two quantised vectors, q and $q' \in \mathbb{Z}_2^M$, we should use the normalised Hamming distance, defined as:

$$d_H(\mathbf{q}, \mathbf{q}') = \frac{1}{M} \sum_{i=0}^{M} q_i \oplus q'_i. \tag{2.6}$$

Boufounos et al. [33] show that with probability at most $2e^{-2t^2M}$, d_H is bounded by:

$$\frac{1}{2} - \frac{1}{2}e^{-(\frac{\pi\sigma d}{\sqrt{2}\delta})^2} - t \leq d_H(\mathbf{q}, \mathbf{q}') \leq \frac{1}{2} - \frac{4}{\pi^2}e^{-(\frac{\pi\sigma d}{\sqrt{2}\delta})^2} + t, \tag{2.7}$$

for a control factor t. This means that d_H depends only on the l_2 distance between the original vectors and the parameters of the transformation.

Moreover, the authors also show that the expected value of $d_H(\mathbf{q}, \mathbf{q}') \leq \sqrt{\frac{2}{\pi}}\frac{\sigma d}{\delta}$, for small values of d. This means that for small distances, the d_H between the hashed vectors is linear in d, or, in other words, Hamming distance between the transformed vectors is proportional to the Euclidean distance between the original vectors if this distance is small enough.

Contrarily, for high values of d, the expected value of $d_H(\mathbf{q}, \mathbf{q}') \leq \frac{1}{2} - \frac{4}{\pi^2} e^{-(\frac{\pi\sigma d}{\sqrt{2}\delta})^2}$, which tends to $\frac{1}{2}$ as d approaches infinity. This means that, after a certain threshold in the Euclidean distance, the distance between the hashed vectors *saturates*, and becomes uninformative.

The non-informative region of the transformation shows that this transformation provides information-theoretic security. However, the security of the transformation is also dependent on the secrecy of the transformation parameters A, w, also called the *key* of the transformation. An attacker that only has access to hashed vectors, without knowledge of the transformation key, can only compare the hashes and learn their relative positions in space. On the other hand, if this attacker knows the original values of a subset of the hashed vectors and if these vectors are close enough to the remaining set, then some information will leak, depending on the transformation parameters and the number of vectors the attacker has access to. However, if the attacker has access to the transformation parameters, it can create any new hashed vectors and potentially obtain the true values of every hashed vector. As such, SBE provides information-theoretic security as long as the parameters (A, w) are kept secret from other parties [33, 253].

2.3.2 Secure Modular Hashing

A generalisation of SBE, in SMH [133] the hash transformation Q_{SMH} is a random projection from $\mathbb{R}^N$ into $(\mathbb{Z}/k)^M$, where $\mathbb{Z}/k$ is the set of integers from 0 to $k-1$ and M is the number of hashes, such that [79]:

$$Q_{SMH}(x) = \lfloor Ax + w \rfloor (\text{mod k}), \tag{2.8}$$

with $w \in \mathbb{R}^{\mathbb{N}} \sim \text{unif}[0, k]$ and $A \in \mathbb{R}^{N \times M} \sim N\left(0, \frac{1}{\delta^2} I_N\right)$. The random matrix A and the random vector w are user-generated parameters that, as in SBE, must be treated as the framework's key and, therefore, must be kept secret to ensure its security.

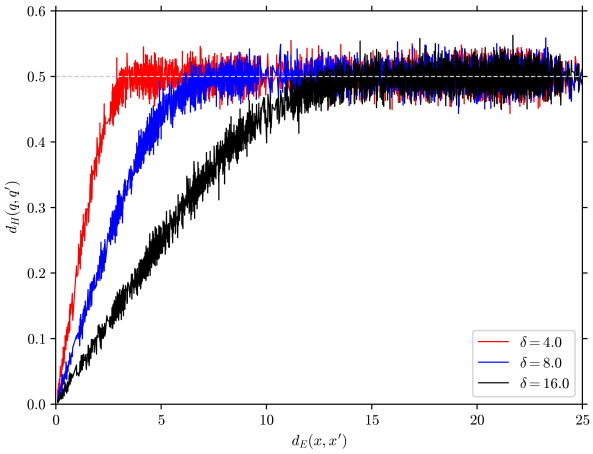
The two main differences of this transformation with regard to SBE are the modularity with regard to k, and the sampling distribution of w. The generalised modularity with regard to k adds an extra parameter that can be used to control the behaviour of the transformation, including the threshold after which the distance between the hashed vectors becomes uninformative. On the other hand, the fact that w is randomly sampled from $unif[0, k]$ adds an extra level of protection to the transformation by hiding the length of the input vector [133]. These two modifications introduce several changes in the results presented for SBE. For instance, for SMH, the mutual information between two hashed vectors is bounded as [133]:

$$I(\mathbf{q}, \mathbf{q}'|d) = \frac{k^2}{3} e^{-2(\frac{\pi d}{\delta k})^2}. \tag{2.9}$$

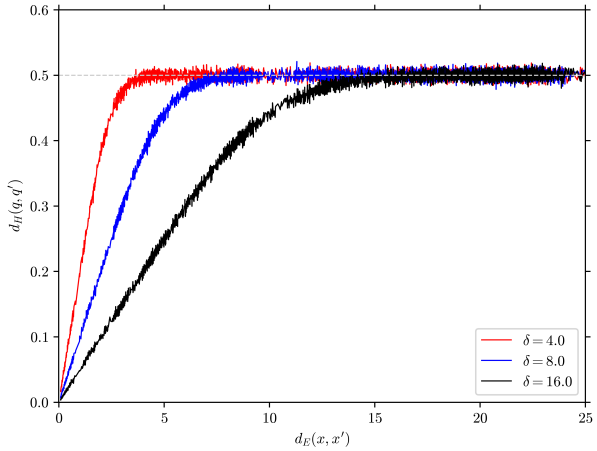
As before, as distance d increases, the mutual information decays exponentially, with the decay rate now being controlled by δ and k. Moreover, Portêlo et al. [133] show that when d tends to infinity, the

Hamming distance $d_H(\mathbf{q}, \mathbf{q}')$ tends to $1 - 1/k$. Similarly to SBE, in this scheme, the tuple (A, w) should be treated as the framework's key and should be kept secret in order to ensure security.

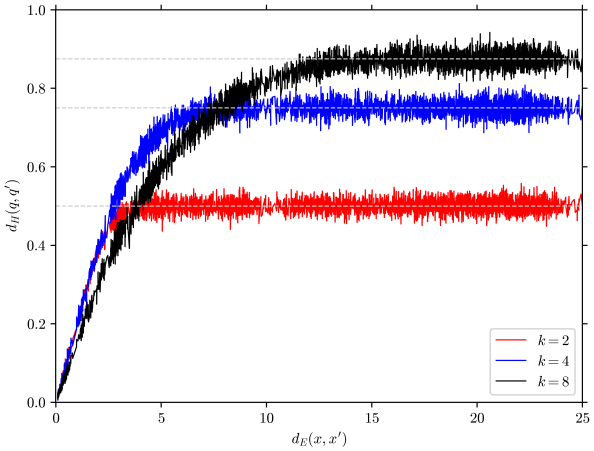
To better illustrate the properties described above, in Figure 2.2, we present an empirical estimate of the relationship between the Euclidean distance $d_E(\mathbf{x}, \mathbf{x}')$ and the Hamming distance $d_H(\mathbf{q}, \mathbf{q}')$. These simulations were performed using 5,000 pairs of vectors of size 256. The transformation parameters correspond to $k = 2$, $mpc = 4$ (resulting in hashed vectors of size $M = L * mpc = 1024$) and $\delta = 4$, unless stated otherwise. In all figures, the dashed grey line corresponds to $1 - 1/k$, the theoretical limit of the Hamming distance for SMH.



(a) $d_E(\mathbf{x}, \mathbf{x}')$ vs. $d_H(\mathbf{q}, \mathbf{q}')$ for varying values of the precision parameter δ.



(b) $d_E(\mathbf{x}, \mathbf{x}')$ vs. $d_H(\mathbf{q}, \mathbf{q}')$ for $mpc = 32$.



(c) $d_E(\mathbf{x}, \mathbf{x}')$ vs. $d_H(\mathbf{q}, \mathbf{q}')$ for varying values of modulus k.

Figure 2.2: Empirical estimates of the relation between the Hamming and Euclidean distances with the SMH transformation for different transformation parameters.

Figure 2.2a shows the difference between SMH transformations with three different values of δ. As was

remarked above, we can observe that increasing δ increases the saturation threshold. Figure 2.2b makes evident the effect of increasing the number of *measurements per coefficient*. When compared to the curves presented in Figure 2.2a, the curves of Fig. 2.2b are much less noisy, and the Hamming distance varies much less with regard to the Euclidean distance. Finally, Figure 2.2c compares the behaviour of the transformation for varying values of the modulus k. As described above, k affects not only the value of the Hamming distance for very large Euclidean distances but also the decay rate of the mutual information between two hashed vectors and, consequently, the transformation's saturation threshold.

2.4 Summary

In this chapter, we provided a brief introduction to the Cryptographic techniques used in the work described in this proposal. We have described cryptographic primitives such as HE, SMC protocols such as OT, GMW, RSS and Yao's GC, as well as LLH techniques, including SBE and SMH. Nonetheless, this chapter is not meant to be exhaustive, and many other cryptographic techniques exist that have not been mentioned, as they were not applied in this thesis. An analysis of the computational performance of each of the described techniques is also missing from this chapter. However, this discussion will appear in the following three chapters, where the necessary results to support it will be presented.

3

Privacy-preserving Speech Processing for Health

As stated in Chapter 1, Section 1.1.2, it is possible to infer numerous speaker traits through the speech signal, including characteristics such as the speaker's sex, age, personality traits, as well as emotional and health states, among others. While this can be seen as a privacy threat when speech is sent to a remote service provider, deriving these traits can also be an application performed by the service provider. Among other speech characteristics, health-related information is extremely valuable, as it makes speech a medium through which several diseases can be remotely detected and monitored. Nonetheless, this information is also sensitive and makes the speaker vulnerable to multiple privacy threats. For this reason, it is necessary to develop solutions that protect user privacy and allow the classification of speech-affecting diseases in remote settings.
In this chapter, we present a privacy-preserving implementation of a Support Vector Machine (SVM) classifier with the Radial Basis Function (RBF) kernel as one possible solution to the problem above. As a proof of concept, this classifier is applied to the speech-based detection of Obstructive Sleep Apnea and Parkinson's disease.

3.1 Introduction

The potential of speech to act as a biomarker for speech-affecting diseases has led to the development of numerous works that seek to detect and assess these disorders automatically, using machine learning classifiers [32, 250, 251, 290]. The fact that speech is ubiquitous and can be acquired non-intrusively makes it an inexpensive modality for this purpose. Speech may be used by clinicians and patients in many scenarios, including clinical facilities or even at patients' homes. Through speech-based methods, it may be possible to monitor the progress of a disease remotely, allowing for rapid interventions and adjustments to patients' medication. Speech-affecting disease classifiers may also serve as screening tools, that alert patients to seek medical assistance.
However, as stated above, remote speech processing raises serious privacy concerns. In a health-related setting, the server will 1) have access to the patient's data and 2) have access to information about the patient's health state. Given its sensitive nature, this information should remain private. For this reason, in this chapter, we explore how to privately classify health-related speech data in a remote processing setting involving two parties, a user and a service provider. Specifically, we will do so using a privacy-preserving implementation of an SVM.
SVMs are powerful and computationally light discriminators that can perform well in various tasks, including those where data is scarce - a frequent scenario in speech analysis for health [66]. There has been a wide variety of works on privacy-preserving SVM inference, several of which implement this classifier with both linear and polynomial kernels [21, 31, 167, 186, 262]. On the other hand, few works have proposed solutions for private SVM inference using the RBF kernel [39, 187, 328]. This can be justified by the combination of two factors: the RBF kernel, which requires the computation of the

Euclidean distance and, most importantly, the computation of the $exp(\cdot)$ function. While computationally heavy, the first can be solved through HE or SMC protocols. The second, however, is more complex, as it requires computing a polynomial approximation of the function.

In [39], the authors propose the use of a variation of the GSHADE [38] protocol to compute the RBF kernel. Nonetheless, the authors only report the computational performance of their method and do not provide values for the results obtained. In [187], the authors used a random sampler to approximate the RBF kernel, removing the need to perform complex functions, with all operations being simplified to linear operations. However, this simplification requires trading off computational performance with the SVM's utility.

In a different approach, [79, 133, 253, 254] take advantage of SMH to mask data vectors. Using SMH's property of proportionality between the Euclidean and Hamming distances, the RBF kernel can be adapted to work with the Hamming distance while providing a level of protection to the input data.

In this chapter, we provide an alternative solution to the private computation of an SVM with the RBF kernel, which combines SMH with Secret Sharing, HE and GCs. Our approach leverages the fact that the computational cost of the E distance is much lower than that of the Euclidean distance. Whereas the Hamming distance corresponds to a sum of XORs of binary values, the Euclidean distance is defined as the square root of a sum of squares of floating point values, making it much cheaper to compute the former than the latter in a privacy-preserving setting. As such, given SMH's proportionality between the Hamming and Euclidean distances, instead of as a privacy method, SMH can be applied as a way to accelerate the privacy-preserving computation of the RBF kernel [132].

To validate our framework, we chose as target tasks the speech-based detection of Obstructive Sleep Apnea and Parkinson's Disease. Obstructive Sleep Apnea is a sleep-related breathing disorder characterised by frequent episodes of upper airway collapses during sleep [259]. Obstructive Sleep Apnea patients report a significant decrease in their quality of life, associated with excessive daytime sleepiness, cognitive impairment, mood and personality changes, relationship discord associated with loud snoring, and depression [229, 259]. Parkinson's disease is the second most common neurodegenerative disorder of mid-to-late life after Alzheimer's disease [251], affecting 1% of people over the age of 65 [138]. Common symptoms include bradykinesia (slowness or difficulty in performing movements), muscular rigidity, rest tremor, as well as postural and gait impairment. 89% of Parkinson's disease patients also develop speech disorders, typically *hypokinetic dysarthria*, which translates into symptoms such as reduced loudness, mono loudness, mono-pitch, hypotonicity, breathy and hoarse voice quality, and imprecise articulation [138, 334].

Our results show that it is possible to implement the proposed classifier in a privacy-preserving way with negligible degradation of disease classification performance. Moreover, the proposed method, in the online phase, takes ∼650ms to perform a single prediction over a feature vector of $\sim 1,500$

dimensions, using 3MB of communication bandwidth for each party (the user and the service provider). This chapter is organised as follows: Section 3.2 presents the state-of-the-art in both privacy-preserving remote processing and privacy-preserving remote speech processing using cryptographic techniques; in Sections 3.3 and 3.4, we detail our proposed method and experimental setup, respectively; in Section 3.5 we present and discuss our results; finally, in Section 3.6 we provide a summary of the chapter.

3.2 Related Work

With the help of breakthroughs in HE and SMC, interest in remote processing frameworks has grown exponentially in recent years. Works such as Cryptonets [108], MiniONN [179], Chameleon [273], Gazelle [136], ABY$^{\mathbf{3}}$ [197] and Delphi [196] have pushed this field forward to a point where private inference over machine learning models for benchmark datasets such as MNIST [169] and CIFAR-10 [157] does not suffer any relevant loss of accuracy when compared to the original *in-the-clear* models. Moreover, open-source libraries such as HELib [116], SEAL [164], ABY [76], and MP-SPDZ [70, 141] have helped make research reproducible, and have allowed non-cryptographers to contribute to this topic with expert knowledge from their fields (e.g., speech processing, machine learning).

To the best of our knowledge, the first contribution to privacy-preserving speech analysis was made by Dias et al. [79]. In an approach similar to [133], the authors of this work applied SMH in combination with an SVM for privacy-preserving emotion recognition. In this work, as in [133], the authors take advantage of the characteristics and privacy guarantees of SMH and modify the RBF kernel to work with Hamming distances between SMH hashes instead of Euclidean distances between feature vectors. For this method to work, the SVM's training data has to be transformed using the same key (A, w) as the user's data to ensure that the distances between the SVM's support vectors and the user's test vectors are meaningful.

Following the work of Gilad-Bachrach et al. [108], and Chabanne et al. [47], Dias et al. [79] also provided a method for privacy-preserving emotion recognition by combining neural networks and the HE Brakerski/Fan-Vercauteren [35, 90] cryptosystem, to build an Encrypted Neural Network. In this method, all operations in the neural network are replaced with their HE counterparts. Given that HE only allows multiplications and additions to be computed, nonlinear activation functions cannot be computed directly and need to be replaced by polynomial approximations. Dias et al. [79] followed the approach of Chabanne et al. [47] and replaced activation functions with their Taylor series expansion at inference, reporting an accuracy degradation of $\sim$2-3% when comparing the private model to the baseline.

Thaine et al. [316] focused on the privacy-preserving extraction of low-level features. In particular, the authors proposed methods to extract Bark Frequency Cepstral Coefficients (BFCCs) and Mel

Frequency Cepstral Coefficients (MFCC) from an encrypted signal using the HE Brakerski/Fan-Vercauteren scheme. The authors report that their method takes ∼47 s to compute Bark Frequency Cepstral Coefficients from 100 frames of length 25. Moreover, the authors argue that it is inefficient to privately compute MFCC features, as it requires more expensive computations (i.e., a logarithm) to be performed. For this task, their method takes between 143-346 s to compute the logarithm of a single encrypted value, depending on the logarithm's desired precision. The authors show that their approach introduces little to no performance degradation on their target task – ASR. Although not directly applied to speech analysis, this method can be included as a first step in the pipeline of privacy-preserving methods for many speech tasks.

In our prior MSc. thesis work [313, 314], we applied variants of the Encrypted Neural Network method of Dias et al. [79] to the detection and assessment of three speech-affecting diseases: depression, Parkinson's disease and the common cold. Instead of using the approach of Chabanne et al. [47], in [313, 314], we followed the approach of Hesamifard et al. [117] and replaced activation functions with a Chebyshev polynomial approximation at both training and inference time. In addition, in [314], the cryptosystem's batching capabilities were used to compute several predictions at the same time, thus amortising the effective cost of each prediction. To this end, every weight in the network had to be converted to integers. Furthermore, to avoid having to scale the network's inputs, these were quantised using μ-law quantisation. For both works, the proposed method yielded negligible accuracy degradation. Nonetheless, it is important to note that the results reported by these two works corresponded to those obtained with the development set, which was used to tune the model's hyper-parameters. For this reason, these results may not reflect the actual performance of these models on unseen test data. In terms of computational performance, [314] achieved ∼4.5 s for a single prediction without the use of batching, and ∼23 s for 16,384 simultaneous predictions, yielding an amortised cost of ∼1.4 ms per prediction.

More recently, Bittner et al. [27] applied SMC to perform privacy-preserving emotion recognition, reporting results corresponding to different SMC protocols, security assumptions and computational power, being able to achieve state-of-the-art results in the RAVDESS dataset [180]. The authors base their approach on the work of [68] and use post-training quantisation to convert the weights of a previously trained model to 8-bit integers. In this way, the authors avoid the need to convert real-valued weights into integers during inference. Their implementation of a 1D Convolutional Neural Network takes an average of 0.26 seconds to perform inference with their best-performing protocol in the three-party passive security setting, with an honest majority reporting no degradation in terms of model performance.

3.3 Method

This work's approach for the privacy-preserving speech-based classification of speech-affecting diseases is based on an SVM with the RBF, having been developed in the early stages of this PhD. This method is intended to be applied in a remote processing scenario, wherein a user wants to use a service provider's model to classify their speech with regard to the presence of a speech-affecting disease. Though this work is a continuation of the research developed in [313, 314], in this work chose to focus on an SVM classifier, instead of on neural networks, as we found the latter to be inadequate for health-related tasks where data is scarce.

Given the sensitive nature of speech data and the application at hand, having the service provider classify the user's speech data without any protection mechanism creates privacy vulnerabilities for the user. On the other hand, running classification on the user's device is also unattractive, as the user may not have the necessary computational resources to do so. Moreover, doing so also threatens the privacy of the service provider's model, as well as of its training data (cf. Chapter 1, Section 1.1.2). For SVMs, the vulnerability of the training data points is particularly stark given that the SVM's parameters consist, in part, of training data points.

To solve this problem, we propose a cryptographic-based method for the privacy-preserving classification of the user's speech data, such that the user's data and the service provider's model always remain protected. Specifically, we use a combination of SMH, Secret Sharing, HE and GC. As stated in Section 3.1, we take advantage of the proportionality characteristics of SMH to simplify the private computation of the RBF kernel. This results in a collaborative pipeline, where the user and service provider interact and jointly compute the SVM over the user's data. This setup is represented in Figure 3.1. For simplicity, in this work, we assume that the user records the audio signal, pre-processes it and extracts the necessary features for classification.

In the remainder of this section, we detail each step of the proposed method to achieve the final privacy-preserving SVM classifier.

3.3.1 Private RBF computation

The original RBF kernel is defined as:

$$\mathrm{k}(x, x_i') = \exp(-\gamma d_E^2(x, x_i')), \tag{3.1}$$

where d_E^2 is the Euclidean distance (cf. Equation 3.2) between x and $x_i^{'}$.

$$d_E(x, y) = ||x_i - y_i|| = \sqrt{\sum_{i=0}^{M}(x_i - y_i)^2} \tag{3.2}$$

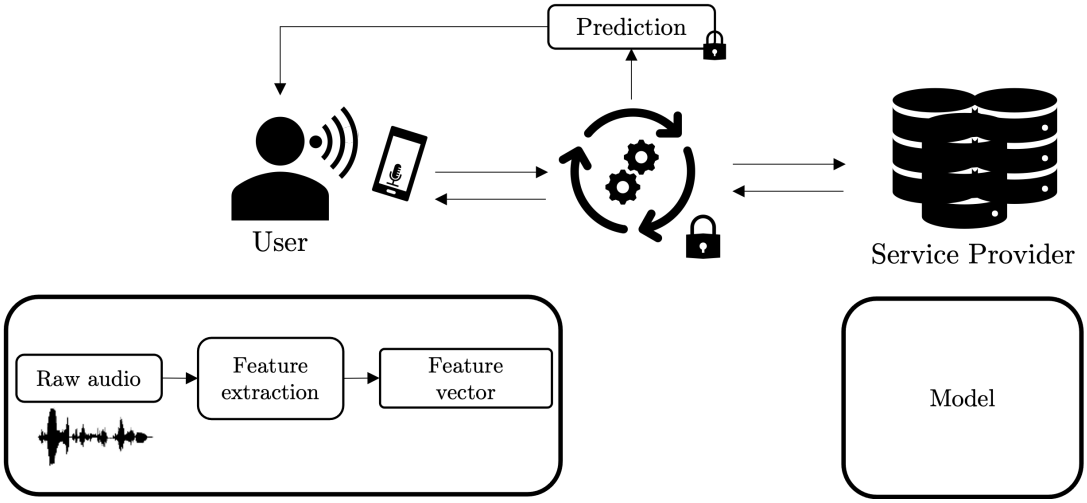


Figure 3.1: Computational setting of privacy-preserving SVM.

Following what was discussed above, it is necessary to transform all data using the SMH transformation to replace the Euclidean distance with the Hamming distance (cf. Equation 3.3).

$$d_H(x, y) = \frac{1}{M} \sum_{i=0}^{M} x_i \oplus y_i \tag{3.3}$$

To do so, the server needs first to generate an SMH key (A, w), sampling each term as follows:

$$\begin{aligned} w &\in \mathbb{R}^{\mathbb{N}} \sim \text{unif}[0, k] \\ A &\in \mathbb{R}^{N \times M} \sim N\left(0, \frac{1}{\delta^2} I_N\right) \end{aligned} \tag{3.4}$$

where k is the modulus of the SMH transformation and δ a scalar that controls the variance of A. The key is shared with the user, and both parties apply the SMH transformation $H(\cdot)$ (eq. 3.5) to their data.

$$H(x) = \lfloor Ax + w \rfloor \pmod{k} \tag{3.5}$$

At the end of this process, the resulting hashes are represented as a list of binary values (each value being composed of k bits), making the resulting vectors comparable through the Hamming distance. Since the Hamming distance is a sum of XORs, it is possible to efficiently compute this distance privately using Boolean Secret Sharing to compute the XOR operations and Arithmetic Secret Sharing to compute the sum (cf. Section 2.2.3).

However, it is not possible to perform exponentiation directly with HE or Secret Sharing in an efficient way. Consequently, it is necessary to compute an approximation of the function. Considering that it is also necessary to multiply and square the Hamming distance before exponentiation (cf. eq. 3.1), instead of computing all of these operations individually, the three operations can be combined into a single

function to approximate. Moreover, to avoid having to perform multiplications using Secret Sharing, the normalising factor $1/M$ can also be moved out of Equation 3.3 and incorporated into the new function:

$$\mathrm{k}_H(x_h, x'_{h,i}) = \exp(-\frac{\gamma}{M^2} d'_H(x_h, x'_{h,i})^2), \tag{3.6}$$

where d'_H is the non-normalised Hamming distance, and M is the size of the vector, x_h is an SMH hashed vector, and $x'_{h,i}$ are the hashed support vectors.

Several options exist on how to approximate the exponential function, including a Taylor series expansion or computing an approximation using Least Squares. Unfortunately, for low-degree polynomials, these approximations have small convergence intervals and diverge quickly out of them. On the other hand, as shown by [117, 313], Chebyshev polynomials can approximate a function within a given interval, which is much more suited to our case. Since it is composed of real-valued coefficients, this polynomial can be evaluated efficiently using the CKKS cryptosystem (cf. Section 2.1) [54]. The resulting kernel can thus be represented as:

$$k_P(x_h, x'_{h,i}) = \sum_{p=0}^{P} a_p d'_H(x_h, x'_{h,i})^p \tag{3.7}$$

where P is the degree of the polynomial, and a_p correspond to its coefficients.

It is important to note that before the computation of the polynomial with HE, the resulting Hamming distances are represented as secret shares, with part of the result being held by each party. To prevent this result from being leaked to either party, the user can encrypt its share and send it to the service provider. In turn, the service provider can use HE operations to add its share to the encrypted share, reconstruct the true value of the Hamming distance, and then proceed with the remaining HE operations.

3.3.2 Private SVM Computation

To complete the full privacy-preserving SVM computation (cf. Equation 3.8), it is necessary to multiply the output of the approximated RBF kernel with the support vector coefficients α_i, accumulate the results and add the intercept term w_0. Since the server has access to the polynomial and α_i coefficients of the SVM, it can pre-multiply them, avoiding an extra multiplication level. Additionally, dividing w_0 by the number of support vectors makes it possible to add it to the constant term of the polynomial to avoid an extra addition. The SVM computation then changes from:

$$\hat{y}(x) = \mathrm{sign}(w_0 + \sum_{i=0}^{n} \alpha_i y_i k_H(x_h, x'_{h,i})), \tag{3.8}$$

where n is the number of support vectors, w_0 is the intercept term, α_i is the weight for support vector x'_i, and y_i is the label corresponding to the same support vector, to:

$$\hat{y}(x) = \text{sign}(\sum_{i=0}^{n} k'_{P,i}(x_h, x'_{h,i})), \tag{3.9}$$

with $k'_{P,i}$ being defined as:

$$k'_{P,i}(x_h, x'_{h,i}) = \frac{w_0}{n} + a_0 + \sum_{p=1}^{P} \alpha_i y_i a_p d'_H(x_h, x'_{h,i})^p. \tag{3.10}$$

Finally, it is necessary to compute the $sign(.)$ function. Since this is a non-linear operation, the most efficient way to perform it would be with SMC. In this case, by comparing the performance of Boolean Secret Sharing and GCs, the latter was selected, as it was reported as more efficient in the framework used to implement our method [76]. However, at the beginning of this step, the result of the previous step is still held by the service provider in encrypted form. To convert it back to a secret-shared form that is compatible with GCs, the service provider needs to create a new secret-share by adding a randomly selected value (under some constraints) to the encrypted value. After, this value is sent to the user, which can decrypt it such that the two parties have the result in a secret-shared form and can continue with the final step of the process.

When the last computation has been performed, the service provider then sends its resulting values to the user so that it can reconstruct the final result of the computation.

The full privacy-preserving SVM computation consists of the steps presented in Algorithm 1.

3.4 Experimental Setup

As stated in the introduction of this chapter, to validate our approach we perform experiments with two speech-affecting diseases, Obstructive Sleep Apnea and Parkinson's disease. In this section, we detail the corpora used for this purpose, along with model training and implementation details.

3.4.1 Corpora

3.4.1.A Obstructive Sleep Apnea

The corpus used for Obstructive Sleep Apnea detection is an extended version of the Portuguese Sleep Disorders (PSD) corpus (a detailed description of the recording protocol and speech tasks can be found in [32]). The corpus includes read and spontaneous speech recordings of 30 (21 male, 9 female) Obstructive Sleep Apnea patients and 30 (11 male, 19 female) control speakers. Each speaker recorded 12 items (1 small text, 10 sentences and one image description). The total duration of the corpus is

Algorithm 1 Steps for the privacy-preserving computation of an SVM classifier with an RBF kernel between two parties, a remote server and a user.

	User		**Server**
1:		1:	
		2:	Sends the user the list of features to be extracted from the speech signal.
2:	Records their voice and extracts the list of features sent by the server.		
		3:	Generates the SMH key and sends it to the user.
3:	Receives the SMH key.		
4:	Applies the SMH transformation to its data.	4:	Applies the SMH transformation to its data and trains the SVM classifier with it.
5:	Secret-shares input data with the server and receives its shares of the support vectors.	5:	Secret-shares support vectors with the user and receives its shares of the user's input data.
6:	Computes the Hamming distance between the support vectors and the input features using Secret Sharing.	6:	Computes the Hamming distance between the support vectors and the input features using Secret Sharing.
7:	Generates the necessary HE keys, encrypts shares resulting from the previous step, and sends them to the server.		
		7:	Receives encrypted shares from the user and reconstructs shares using HE.
		8:	Computes polynomial approximation of the RBF kernel using HE.
		9:	Generates random values for new secret shares, adding them to the result of the previous step and sends the encrypted "shares" to the user.
8:	Receives and decrypts the new secret shares using a decryption key.		
9:	Interacts with server to compute *sign()* function over its secret shares, using GCs.	10:	Interacts with user to compute *sign()* function over its secret shares, using GCs.
		11:	Sends the resulting shares to the user.
10:	Receives the resulting shares from the server, reconstructs true value and obtains final computation result.		

2h09 min. We partitioned the corpus into 4-second-long audio files using overlapping windows, with a shift of 2 seconds, resulting in 1793 and 1702 control and patient samples, respectively. Each sample is represented by a vector of 109 knowledge-based features, as proposed by [32].

3.4.1.B Parkinson's Disease

The corpus used for Parkinson's disease detection corresponds to a subset of the New Spanish Parkinson's Disease Corpus, collected at the Universidad de Antioquia, Colombia [227], composed of read sentences. The corpus includes 50 patients and 50 controls. This subset of the corpus has a duration of 59 min. As with the PSD corpus, if the utterance was longer than 4 seconds, we partitioned it into 4-second-long audio files using overlapping windows with a 2-second shift. This resulted in 661 patient and 655 control samples. Each sample is represented by a 114-dimensional knowledge-based feature vector, as proposed by [251].

3.4.2 Model training and parameters

We compared the performance of three models: *Baseline*, which refers to the baseline model implemented without the privacy-preserving framework; *SVM+SMH* which refers to the SVM combined with the SMH transformation; and *Poly SVM+SMH* which corresponds to the previous model, but where the kernel has been approximated with a Chebyshev polynomial. The parameters for each SVM model and the SMH parameters were optimised through a grid search. All models were trained using leave-one-speaker-out cross-validation. For Obstructive Sleep Apnea's baseline model, our best results were obtained using $C = 10$ and $\gamma = 0.001$. For the corresponding *SVM+SMH* and *Poly SVM+SMH* models we found that $C = 1$, $\gamma = 10$, $k = 4$, $\delta = 32.26$ and $mpc = 32$ yielded the best results. For the Parkinson's disease baseline model, we used $C = 10$, $\gamma = 0.001$. For the Parkinson's disease *SVM+SMH* and *Poly SVM+SMH* models we used $C = 1000$, $\gamma = 0.01$, $k = 2$, $\delta = 1000.0$ and $mpc = 4$. All models were implemented and trained using Python's *scikit-learn* SVC classifier [242]. The SMH transformation and the custom RBF kernel using the Hamming distance were also implemented in Python.

3.4.3 Private SVM implementation details

As stated above, our method requires three steps: computing the Hamming distance between the user's input and the server's support vectors, evaluating the polynomial approximation of the kernel, and computing the $sign(\cdot)$ function.

The first step can be efficiently computed using secret sharing. For this step, we used ABY's [76] implementation of the Arithmetic and Boolean secret sharing protocols, as well as the corresponding conversions between them. In addition, we took advantage of the library's efficient Hamming weight implementation [247], to perform the sum component of the Hamming distance. At the end of this step, both the server and the user hold a random-looking share of the Hamming distance between the user's input and the server's support vectors. For this step, we took advantage of ABY's Single Instruction Multiple Data (SIMD) capabilities and encoded 64 bits in each shared value to speed up computations.

The second step involves approximating the RBF kernel using a polynomial. Since we trained and tested our model using Leave-One-Speaker-Out cross-validation, we computed a different polynomial approximation for each fold using the Hamming distances between every pair of training data vectors. In this way, we emulated real-world conditions where a service provider has fixed training sets. Our experiments showed that a 5^{th} degree Chebyshev polynomial yielded the best trade-off between computational complexity and accuracy. An example of a 5^{th} degree Chebyshev polynomial approximation of equation 3.6 can be found in Figure 3.2.

To perform this step, we used SEAL's [164] implementation of CKKS [54], using a *polynomial modulus* of 8192 and a *coefficient modulus* composed of two 60-bit long and three 40-bit long small primes. We took advantage of CKKS's batching capabilities to encode all Hamming distances into fewer ciphertexts, thus reducing communication and computational costs.

To compute the final $sign(.)$ function, we used ABY's implementation of Yao's GCs [76] to perform a *greater than* operation. For both libraries, we used the default parameters for 128-bit security.

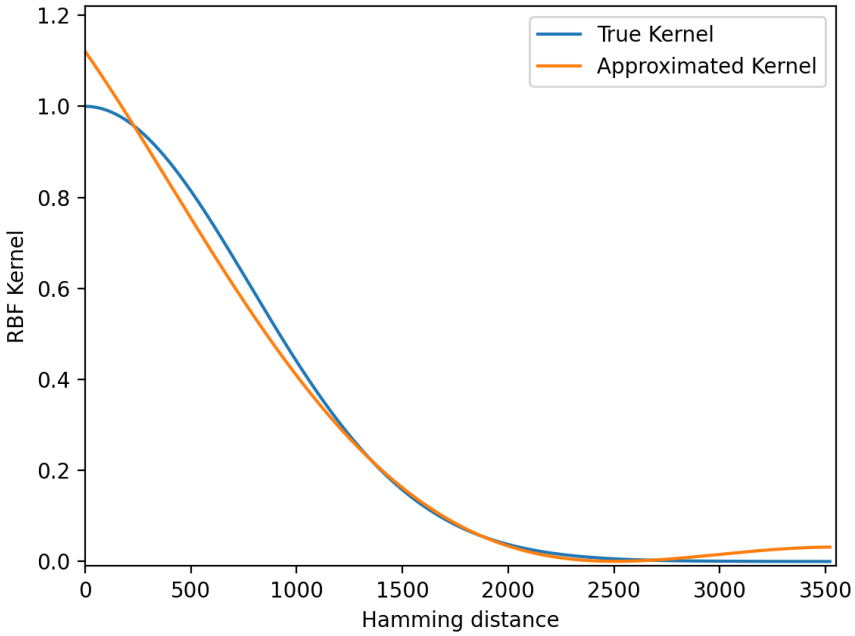


Figure 3.2: 5^{th} degree Chebyshev polynomial approximation of Equation 3.6.

3.4.4 Evaluation metrics

To evaluate performance in terms of disease classification, we report unweighted average precision, recall and F1 scores. In terms of the performance of the privacy-preserving classifier, we report computational costs in *ms* and communication costs in *MB*, obtained on a machine with an Intel Core

Quad-Core i5 CPU @ 1.40GHz and 16GB of RAM.

3.5 Results

Table 3.1: Results achieved for Obstructive Sleep Apnea and Parkinson's disease detection in terms of unweighted average Precision, Recall and F1 Score.

Method	Obstructive Sleep Apnea			Parkinson's Disease		
	Precision(%)	Recall(%)	F1 Score(%)	Precision(%)	Recall(%)	F1 Score(%)
Baseline	69.0	68.9	68.9	78.6	78.6	78.6
SVM+SMH	68.3	68.2	68.2	80.1	79.7	79.6
Poly SVM+SMH	68.4	68.4	68.4	80.1	79.7	79.6

The results for our experiments are presented in Table 3.1, in which the *Baseline* corresponds to an SVM trained with data without any transformation, using the RBF kernel with the Euclidean distance; *SVM+SMH* corresponds to an SVM trained with SMH transformed data and using the RBF kernel with the Hamming distance; finally, *Poly SVM+SMH* corresponds to the results obtained training the SVM in the same way as in *SVM+SMH* and performing inference over the test set with the polynomial approximation.

We can see that the results obtained for Obstructive Sleep Apnea classification in the privacy-preserving framework are slightly worse than the baseline but with a negligible difference (0.5% in terms of UAR). On the other hand, the results achieved for Parkinson's disease classification surpassed the baseline. This may be attributed to the quantisation of the inputs due to the SMH transformation, which, by reducing the variability of each feature in the training dataset, allows the classifier to obtain higher results with this small dataset. Nevertheless, considering the small size of this dataset, it is unclear whether these differences are statistically significant.

3.5.1 Privacy, security and computational performance

To assess the privacy of our protocol, we need to consider the privacy of two components, the model and the user's input.

The privacy of the user's input comes from the fact that it is never held by the server in an unprotected form and the security guarantees of the underlying sub-protocols. Initially, the data is secret-shared with the server, after which it is encrypted with HE, and finally, it is secret-shared again. On the other hand, the server's model is only held by the server, and the user never has access to it. Even though this gives some privacy to the model, it does not protect it from model extraction attacks, that might recover information about the model [49, 321].

The security of the HE portion of our method comes directly from the CKKS protocol [54], which, as stated in 2, Section 2.1, is secure under the assumptions of the hardness of the RLWE problem.

Recently, CKKS has been shown to be vulnerable to key-recovery attacks that are able to reconstruct the secret key of a set of ciphertexts by observing their decryption [171]. Nevertheless, our protocol is secure against this attack, since the decrypted values are only seen by the user, who already knows the key.

In the case of the SMC protocols (for Secret Sharing and GCs), we define our security under the *honest-but-curious* model, where both parties are expected to follow the protocol while trying to learn as much as possible about each other. We chose this assumption as it allows for more efficient models and is considered in the literature to be enough for most applications [136, 273, 323]. It is, nevertheless, possible to transition to the *malicious* model at the cost of efficiency [70], as stated in Section 2.2.3.

Table 3.2: Computational and communication costs for each protocol in the proposed method. Computational costs were averaged over 100 runs.

Sub-protocol	Time (ms)			Communication (MB)		
	Pre-processing	Online	Total	Pre-processing	Online	Total
Hamming distance	698.9 ± 36.9	523.6 ± 32.7	1222.5 ± 34.9	194.8	3.0	197.8
Polynomial kernel	–	126.9 ± 0.9	126.9 ± 0.9	–	0.7	0.7
Sign function	6.9 ± 8.3	0.8 ± 0.2	7.7 ± 5.9	43.5×10^{-4}	1.3×10^{-4}	44.8×10^{-4}
Total	705.8 ± 26.7	651.3 ± 18.9	1357.1 ± 20.4	194.8	3.7	198.5

The computational and communication costs of each sub-protocol of the scheme are presented in Table 3.2. Assuming 109 features, 1432 support vectors, $k = 4$ and $mpc = 32$, resulting in hashed vectors of size 6,976, our implementation takes a total of ∼650 ms for a single prediction during the online phase (excluding communication time), in addition to ∼700 ms for the pre-processing phase. In terms of computation, our protocol uses ∼4MB and ∼195MB of bandwidth during the online and pre-processing phases, respectively. Comparatively, a single prediction using Python's *scikit-learn* SVC implementation [242], under the same conditions, takes on average ∼0.65 ms, making the total of our private implementation almost 2,000 slower than its unencrypted counterpart.

3.6 Summary

In this chapter, we describe a method for the privacy-preserving computation of SVMs using the RBF kernel. We have shown that through the use of SMH, it is possible to replace an otherwise expensive computation, such as the Euclidean distance, with the computation of the Hamming distance, making it much cheaper to compute in a private setting. We further show that, for two health-related speech tasks, Obstructive Sleep Apnea and Parkinson's disease detection, our method does not introduce any relevant accuracy degradation.

The current state-of-the-art in health-related speech tasks indicates how mature this technology is becoming. However, machine learning models require a great deal of expertise and investment to be

developed, making their distribution undesirable from a business point of view. Moreover, deep learning models are often computationally expensive, requiring hardware (i.e. GPUs) that may not be available for many users. Remote processing solves these issues but introduces ethical and legal concerns over patient privacy issues.

The techniques described in Chapter 2 can be used to solve these concerns, but while they have been applied to numerous tasks in other fields, few contributions exist on the topic of privacy for speech-affecting disease detection and monitoring, notwithstanding its relevance. This may be caused by several factors, the most important of which likely being the difficulty in combining state-of-the-art machine learning methods with cryptographic primitives. The fact that speech-based machine learning models have only recently started to obtain good results with health-related data *in-the-clear* explains why research on privacy techniques that are non-essential to obtain good speech-based disease classifiers has, so far, received little attention. Nonetheless, with the importance and sensitive nature of these tasks, and the current societal concerns about privacy, it is essential to increase the efforts to develop privacy-preserving techniques for this end. Furthermore, linguistic cues can also be fundamental to the detection of certain diseases. While this work adds strength to the need to protect raw audio data, it also highlights the importance of the unexplored area of privacy-preserving methods for health based on transcribed speech, making it a promising avenue for future work.

As a final note, it is important to state that, while result degradation is negligible to non-existent, we do not evaluate the statistical significance of our results, something which would have been important given the small size of the datasets. Re-running these early experiments was, however, not considered a priority.

4

Privacy-preserving Speaker Embedding Extraction

The development of privacy-preserving automatic speaker verification systems that allow users to authenticate themselves without risking the privacy of their voices has been the focus of several studies. However, current privacy-preserving methods assume that the template voice representations (or speaker embeddings) used for authentication are extracted locally by the user. This poses two significant issues: first, having knowledge of the speaker embedding extraction model may create security and robustness liabilities for the authentication system; second, from the point of view of a service provider the speaker embedding extraction model is arguably one of the most valuable components in the system and, as such, disclosing it would be highly undesirable.
In this chapter, we show how speaker embeddings can be extracted while keeping both the speaker's voice and the service provider's model private using SMC. Further, we show that it is possible to obtain reasonable trade-offs between security and computational cost. This work is complementary to those showing how authentication can be performed privately and thus can be considered as another step towards fully private automatic speaker recognition.

4.1 Introduction

As described in the introduction of this thesis, recent years have seen an increase in the number of online services and applications that use speech as a means of authentication and interaction. Among other speech technologies, voice-based authentication systems – or ASV systems – are becoming widespread. The uniqueness and ubiquitous nature of speech make its use a straightforward manner to protect and grant access to both local and remote systems. However, in the remote case, ASV systems raise multiple privacy concerns.
Given the sensitive information that speech carries and the possible threats that come from speech data sharing by sending a recording of their voice – or a template thereof – to a remote server, users are risking their privacy and, in the case of authentication systems, their security.
Due to the above, the problem of protecting privacy in the ASV setting has been a precursor for much of the research done on speech privacy. One of the first strides in this direction is the cryptographic-based work of Pathak et al. [239], who adapted a Gaussian Mixture Model (GMM) to work with HE to perform speaker verification and identification. Similarly, Portêlo et al. [255] implemented a privacy-preserving GMM-based speaker verification using GC.
More recently, Nautsch et al. and Treiber et al. applied HE [208] and later SMC [212, 323], to the verification step of an ASV pipeline. Similarly, Cheng et al [52], developed a protocol for privacy-preserving speech verification using SMC. In a different type of approach, Pathak et al. [238], Portêlo et al. [253] and Jiménez et al. [133] explored the applicability of LLH techniques to privacy-preserving ASV, while Mtibaa et al. studied cancelable biometric schemes for ASV [203, 204]. However, these works focus mainly on the security of the speaker templates or on how the verification

step can be performed privately, sharing the assumption that the user locally extracts voice templates. In contrast, we argue that this is highly undesirable for service providers. Specifically, we argue that the model used to extract voice templates, or speaker embeddings, is one of the most valuable components, if not the most valuable, in the speaker verification pipeline. This stems from the fact that speaker embedding extractors require large amounts of data and high levels of expertise to be developed. As such, by sharing this model, ASV service providers would relinquish control over their intellectual property and, consequently, lose the value it holds. Further, as noted by Das et al. [71] and Villalba et al. [333], having knowledge of the speaker embedding extractor model may allow attackers to craft adversarial examples that mislead the ASV system, raising security and robustness concerns.

For this reason, in this work, we show how speaker embeddings can be extracted privately using SMC. Specifically, we focus on the private extraction of *x-vector* speaker embeddings [292]. This not only allows the protection of the speaker's voice, as it is never shared with the ASV provider but also the protection of the speaker embedding extraction model. Moreover, even though we only consider the private extraction of *x-vectors*, our implementation can be directly combined with some of the works mentioned above for private speaker verification [212, 323] to produce a fully end-to-end private speaker verification system that protects both the speaker's voice and the *vendor*'s model.

The remainder of this chapter is organised as follows: Section 4.2 specifies the setting and threat models assumed for our task; in Section 4.3, we describe the experimental setup, while in Section 4.4, we present and discuss the results obtained. Finally, Section 4.5 presents a summary of this chapter, drawing some conclusions and topics for future work.

4.2 Privacy-preserving speaker embedding extraction

Designed for speaker recognition, speaker embeddings are fixed-length representations of variable-length speech signals that capture information about the speakers who uttered them. Traditional speaker embedding extractor systems tried to model how speech was produced by a speaker, relying on generative models such as Gaussian Mixture Model - Universal Background Models (GMM-UBM) [271], Gaussian Mixture Model (GMM) Supervectors [41] and *i-vectors* [75]. Modern neural speaker embedding extractor systems such as *d-vectors* [331] and *x-vectors* [78, 292, 365] instead model the differences between speakers by relying on latent representations extracted from intermediate layers of deep neural network models trained for speaker identification, hence being considered discriminative systems.

We consider two parties to be involved in the extraction of speaker embeddings in the context of ASV: the *user*, who wants to be able to access a given system and the ASV *vendor*[1], who provides the

[1] In this chapter we use the terminology commonly used in speech biometrics and refer to the server/service provider as *vendor*.

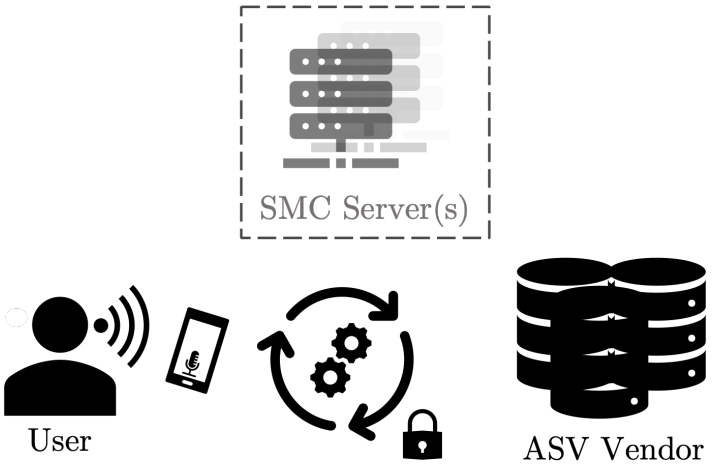


Figure 4.1: Privacy-preserving extraction of speaker embeddings.

authentication system as a service.

If we were considering a complete ASV system, we would also need to consider the *ASV Controller*, the party who holds the set of speaker templates who are allowed to access the system. However, since we only focus on the extraction of speaker embeddings in this work, we will not take this party into account. Nonetheless, in some cases, the *vendor* and *controller* may be the same party.

4.2.1 Threat models

Our goal is to have the *vendor* and *user* collaborate to privately extract a speaker embedding from a speech sample belonging to the *user*, using an extraction model belonging to the *vendor*. We consider that both the *user* and *vendor* are interested in protecting the privacy of their data – the *user* for the sensitive nature of their speech data, and the *vendor* due to the value of its model, and the security of its service. Furthermore, the resulting *speaker embedding* should only be accessed by the *user*, if it is to be used in an external application, or by no party, if this protocol is an intermediate computation in a larger pipeline. A diagram of the overall computational setting considered in this work can be found in Figure 4.1.

As stated in Chapter 2, Section 2.2, when using SMC, it is necessary to define the threat model of the computation. In this work, we consider four different scenarios.

In the first scenario, the *vendor* and *user* are the only parties involved in the private extraction of the speaker embedding and are both assumed to be *semi-honest*. This is the weakest security model, as either the *vendor* or the *user* might thwart the protocol to obtain information about the other's data.

In the second scenario, we consider adding a trusted non-colluding SMC server to the computation. In a real-world setting, this party would correspond to a company providing servers for SMC. Since such a company would need to rely on its reputation for its business, we argue that it would always follow

protocol and would never collude with any other party involved in the computation [29]. By adding this trusted non-colluding server, and since the *user* and *vendor* have no incentive to collude with each other – the SMC server does not have data to be stolen – this allows us to instantiate the honest majority 3-party RSS SMC protocol of Araki et al. [13]. As detailed in Section 2.2.3.C, RSS schemes are much more efficient than additive secret-sharing protocols while keeping the same level of security [13].

For our third scenario, we consider adding a second trusted non-colluding SMC server. This allows us to instantiate a 4-party honest-majority RSS SMC protocol. Specifically, we can instantiate the 4-party protocol of Dalskov et al. [69], which is secure against one malicious party. In this way, the protocol will abort if either the *user* or the *vendor* behave maliciously. Since the *user* and the *vendor* will not collude, and the SMC servers are assumed to be trusted, this setting will be more secure than the previous one. However, in this case, the non-collusion assumption of the SMC server is much stronger.

In our fourth scenario, we return to the 2-party setting and assume that either the *vendor* or the *user* might behave maliciously. This is the setting with the highest level of security, but it will also incur the highest computational and communication costs.

4.2.2 Privacy-preserving *x-vector* extraction using SMC

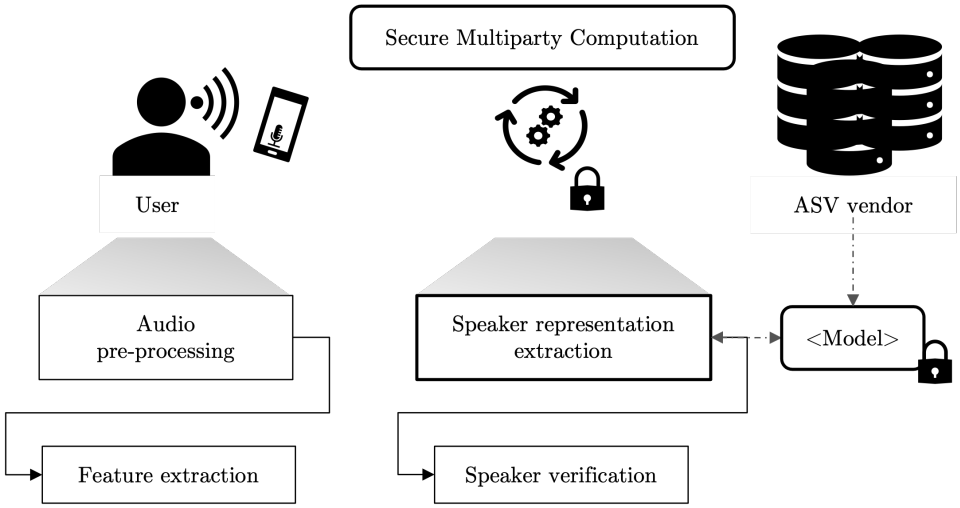


Figure 4.2: Proposed privacy-preserving speaker embedding extraction system.

Originally proposed by Snyder et al. [292] as an alternative to *i-vectors* [75] – speaker representations that aimed to represent the total speaker and channel variability – *x-vectors* aim to model characteristics that discriminate between speakers. The *x-vector* architecture is a neural network trained to discriminate between a large number of speakers. In this context, *x-vectors* correspond to latent representations extracted from an intermediate layer of the network. This network is composed of three main blocks: the first block is a set of time-delay neural network (TDNN) layers that operate at

the frame level with a small temporal context. These layers work as 1D dilated convolutions, with a kernel size corresponding to the temporal context, which alternate with ReLU activation functions; the second block, a statistical pooling layer, aggregates the information across the time dimension and outputs the per-feature mean and standard deviation for the entire speech segment; the third block is a set of fully connected layers, from which *x-vector* embeddings are extracted after the network is trained for speaker classification. The *x-vector* network takes as input Mel-Filterbank Energies (FBanks) extracted from small, overlapping speech frames. In this work, for simplicity, this step, along with prior audio pre-processing steps, is considered to be performed by the user.

To implement this extractor with SMC, we need to take into account the type of operations required by each layer in the network. 1D dilated convolutional layers are linear transformations and can be implemented using either the basic arithmetic operations of the SMC protocol or with specific protocols for inner product computations [141]. ReLU activation functions require the computation of a comparison, which can be done through the secure comparison protocol of [46]. The Statistical Pooling layer involves computing the mean and standard deviation of the input. To compute the standard deviation, we will need to compute a square root, which can be done through the protocol of [11]. A representation of the complete process is presented in Figure 4.2

All the protocols mentioned above work for the fixed-point number representation described in Section 2.2.3.E, making them directly compatible with the weights and inputs of neural networks after these have also been converted to a fixed-point representation.

4.3 Experimental Setup

4.3.1 Corpora

The Voxceleb corpus was used to train the *x-vector* extractor and the Probabilistic Linear Discriminant Analysis (PLDA) model described below. This corpus includes recordings of 7,363 speakers of multiple ethnicities, accents, occupations and age groups. It is composed of short clips taken from interviews uploaded to YouTube [61, 207]. The corpus is composed of two parts, *VoxCeleb 1 and 2*, both subdivided into *dev* and *test* sets.

4.3.2 Speaker embeddings

For our experiments, we used the pre-trained *x-vector* model made available by SpeechBrain [265]. This model follows the architecture of [292]. A description of the layers used for extraction can be found in Table 4.1. The model was trained using the *dev* partitions of Voxceleb 1 and 2, amounting to 7,205 speakers. As a baseline reference for computational cost, extracting a single *x-vector* from a 3-second long speech sample with this model, using a CPU, takes ∼0.03s.

Table 4.1: *x-vector* extractor architecture.

#	Layer	Input	Output	Kernel	Dilation
1	TDNN 1	24	512	5	1
2	TDNN 2	512	512	3	2
3	TDNN 3	512	512	3	3
4	TDNN 4	512	512	1	1
5	TDNN 5	512	1500	1	1
6	Statistics Pooling	1500	3000	-	-
7	Linear	3000	512	-	-

A PLDA model was used to score pairs of *x-vectors* when performing verification [143]. The full pipeline achieves 3.2% Equal Error Rate (EER) on the Voxceleb 1 test set (Cleaned) [61, 265]. For reference, the EER achieved for VoxCeleb 1 test set (Cleaned) using cosine-similarity scores, and 3-second long samples, corresponds to 11.24% EER.

4.3.3 Privacy-preserving implementation

The network described in the previous subsection was implemented using the MP-SPDZ library [141]. We tested our implementation using four different protocols with different levels of security, as detailed in Section 4.2.1. Specifically, we tested our implementation over the following protocols: the 2-party *semi-honest* (SH) version of the SPDZ_{2^k} scheme for a *dishonest majority* (DM), denoted as Semi_{2^k} [65]; the 3-party RSS scheme described in [13], which provides *semi-honest* security, in the *honest majority* (HM) setting; the 4-party RSS scheme of [69], which provides *malicious* (Mal) security in the *honest majority* setting against one corrupted party; and the 2-party *malicious* version of the SPDZ_{2^k} scheme [65].

For 3 and 4-party RSS and the *semi-honest* version of SPDZ_{2^k}, we used local share conversions [69] to improve efficiency. For 3 and 4-party RSS, we also used probabilistic truncation as proposed by [68, 69] instead of regular truncation to further improve efficiency. Our experiments assume the default security parameters for each protocol, namely 40-bit security for 3 and 4-party RSS, and Semi_{2^k}, and 64-bit security for SPDZ_{2^k}. We used the library's fixed-point number representation, adopting the default configuration of 15 bits for the decimal part and 16 bits for the fractional part. All tested protocols perform computations modulo 2^k, where $k = 64$. Experiments were performed on a machine with 24 Intel(R) Xeon(R) CPU E5-2630 v2 @ 2.60GHz processors and 250GB of RAM.

4.3.4 Evaluation metrics

We evaluate the performance of the privacy-preserving implementation of the *x-vector* extraction in terms of computational costs in seconds and in terms of communication costs in *MB*. On the other hand, utility is measured in terms of Mean Squared Error between the original and privately extracted

x-vectors, as well as the EER of the privately-extracted *x-vectors* computed over VoxCeleb 1's test set (Cleaned). The EER is computed using cosine-similarity scores, instead of PLDA scores.

4.4 Results

Table 4.2: Results obtained for each protocol in terms of computational performance measured in seconds.

Protocol	Security Model	Pre-processing	Online	Total
2-party Semi$_{2^k}$ [65]	DM/SH	8,423.00 ± 165.36*	18.92 ± 0.22	8,441.93 ± 165.36
3-party RSS [13]	HM/SH	0.18 ± 0.15*	10.68 ± 0.15	10.85 ± 0.14
4-party RSS [69]	HM/Mal	1.21 ± 0.21*	16.76 ± 0.21	17.97 ± 0.21
2-party SPDZ$_{2^k}$ [65]	DM/Mal	147,799.68 ± 1,016.82*	126.32 ± 1.16	147,926.00 ± 1,016.82

Table 4.3: Results obtained for each protocol in terms of communication costs measured in MB.

Protocol	Security Model	Pre-processing	Online	Total
2-party Semi$_{2^k}$ [65]	DM/SH	1,662,300.60*	12,919.40	1,675,220.00
3-party RSS [13]	HM/SH	15.04*	118.02	133.06
4-party RSS [69]	HM/Mal	27.14*	333.16	360.30
2-party SPDZ$_{2^k}$ [65]	DM/Mal	21,870,489.60*	27,810.40	21,898,300.00

Tables 4.2 and 4.3 include the results obtained for our experiments in terms of computational performance and communication cost. All results correspond to the extraction of a single *x-vector* using a 3-second long speech sample. Online results for all protocols and the full computation results for 3 and 4-party RSS were obtained by averaging over 100 runs. Full computation results for Semi2k and SPDZ2k were computed over 10 runs due to their high computational cost. Values denoted with * were estimated by computing the difference between the full protocol and the online phase.

Our results show that RSS schemes significantly outperform the *semi-honest* and *malicious* versions of SPDZ$_{2^k}$, both in terms of computational and communication performance. Further, since for the *semi-honest* version of SPDZ$_{2^k}$, pre-processing takes >2h and >1TB of data, and since the pre-processing for the malicious version takes >41h and >20TB of data, it is clear that the private extraction of *x-vectors* with these protocols, particularly for a high level of security, is currently infeasible. Contrarily, the results for the RSS schemes can be deemed feasible, particularly when considering that no modifications were made to reduce the size of the *x-vector* extraction network.

When comparing the 3 and 4-party RSS protocols, while the 3-party *semi-honest* version is more efficient in terms of computational cost and communication, we argue that the added security of the 4-party RSS protocol is a reasonable trade-off for the additional ∼7s and ∼230MB in the total computational and communication costs[2]. Still, it is necessary to consider that to implement the 4-party

[2]This difference in computational cost was found to be statistically significant, with $p < 0.001$, through an independent samples *t-test*, and assuming a normal distribution of residuals.

RSS protocol, one needs to have strong assumptions about the honest behaviour of the SMC servers. Finally, our experiments showed that the SMC implementation yielded negligible degradation, with the average Mean Squared Error distance between 100 *x-vectors* extracted with the original and SMC implementations being just $\sim 1\%$ of the total magnitude of the vector. Moreover, the *x-vectors* extracted with the cryptographic protocols achieve an EER of 11.26%, which we consider to be a negligible degradation when compared to the 11.24% EER of the original *x-vectors*.

4.5 Summary

In this chapter, we have shown that it is possible to extract *x-vector* speaker embeddings at a reasonable level of security and computational and communication costs while protecting the privacy of both the user's data and the ASV *vendor*'s model, using SMC, particularly when deploying on 3 and 4-party RSS protocols. This problem had been unexplored so far, as other privacy-preserving works for ASV assumed that speaker embeddings are to be extracted by the user. This makes this work complementary to others in the literature and another step towards fully private ASV pipelines. Given the current computational cost of the system, it would be important to extend this work by finding ways to modify the *x-vector* extraction network to improve efficiency, such as using weight quantisation techniques, replacing activation functions with polynomial approximations and removing expensive operations, e.g., the square-root operation when applying the statistics pooling layer. Moreover, it would be interesting to also consider protocols following the *covert* security model, wherein adversaries may have a malicious behaviour, but where there is a probability that in doing so, they may be discovered.

5

Privacy-preserving Automatic Speaker Diarization

Automatic Speaker Diarization (ASD) is an enabling technology with numerous applications, which deals with recordings of multiple speakers, raising particular concerns in terms of privacy. In remote ASD settings, where recordings are shared with a server, users relinquish not only the privacy of their conversations but also all the information that can be inferred from their voices. However, to the best of our knowledge, the development of privacy-preserving ASD systems has been overlooked thus far. In this chapter, we build on the work developed in Chapter 4 and tackle this problem using a combination of SMC and SMH, applying them to the two main steps of a cascaded ASD system: speaker embedding extraction and agglomerative hierarchical clustering. Our system is able to achieve a reasonable trade-off between performance and efficiency, presenting real-time factors of 1.1 and 1.6 for two different SMC security settings.

5.1 Introduction

Automatic Speaker Diarization (ASD) is an enabling technology for many speech-based applications. When combined with Automatic Speech Recognition systems, ASD can provide additional context to transcriptions and be used to perform speaker adaptation. On its own, ASD also allows users to search for and filter segments that correspond to specific speakers or, in the case of audio diarization, specific audio events. This filtering may be particularly important in multi-speaker audio streams where the target is a single speaker. In security applications, this speaker may be a potential blacklisted criminal. In clinical interviews, it may be the patient. In language acquisition recordings, it may be the child whose linguistic skills are being assessed. The list of potential ASD scenarios is pervasive, ranging from courtrooms to meetings, socio-linguistic interviews and broadcast news, among others [236, 322].

When dealing with large amounts of speech data, when ASD is used as part of a larger system, or even due to the lack of computational resources, it may be useful to delegate this task to an external service. However, this setting creates a significant privacy challenge: the server will have direct access to the user's data. This means that the voices present in the recording and what is being said will be available to the server, giving it a vast repository of potentially sensitive information [289], which the speakers may want to keep private.

The alternative of running the diarization process on the user's device is also unattractive, as it would require the service provider to share their model with the user. Considering that ASD models require large amounts of data and high levels of expertise to be developed, sharing them with users would make the service provider potentially lose the value that the model holds. In cascaded ASD models, this is particularly true for the speaker embedding extraction model, as stated in Chapter 4. This makes this (mostly) unexplored problem – with the notable exception of [237] – particularly interesting.

In this chapter, we build on the work developed in the previous chapter on the privacy-preserving extraction of *x-vector* embeddings using SMC and extend it to the setting of ASD. Specifically, we

propose a system that performs the extraction of speaker embeddings and the clustering step in a privacy-preserving way by leveraging two cryptographic techniques: SMC and SMH. The combination of these techniques allows us to protect the service provider's model, particularly the speaker embedding extraction model, while also keeping the speakers' data hidden from the server.
The remainder of this chapter is organised as follows: Section 5.3 describes the ASD baseline model and our privacy-preserving system; in Section 5.4 we describe the experimental setup; in Section 5.5 we present and discuss the results obtained; finally, Section 5.6 summarises the chapter, presenting some conclusions and topics for future work.

5.2 Automatic Speaker Diarization

Traditional ASD systems, also known as clustering-based systems, comprise several independent if sometimes overlapping, modules.
The first steps in this pipeline correspond to audio pre-processing (e.g., speech enhancement, dereverberation, or separation). This is usually followed by a Voice Activity Detection (VAD) stage, where the signal is filtered for non-speech segments. The simplest forms of VAD are based on frame-level energy. These provide very good results for recordings under controlled conditions, however, they do not perform well in highly variable environments [12]. This problem can be solved by discriminating between speech and non-speech segments using classifiers such as Gaussian Mixture Model (GMM) or deep learning models [364].
VAD can be followed by an optional speaker segmentation stage, where the already segmented signal is further divided where speaker changes occur. Speaker segmentation methods can be divided into two categories: implicit and explicit. Implicit methods either assume that the speech segments acquired in the VAD stage are single speakers or split them into small, overlapping, speech frames, that can be considered to have been uttered by a single speaker. Explicit approaches rely on the computation of metrics between speaker discriminative representations of consecutive segments. While explicit segmentation was common in early works, more recent works favour implicit segmentation [106, 358]. Each of the segments resulting from this stage is then assumed to correspond to a single speaker.
Following VAD and speaker segmentation, speaker discriminative representations are extracted from each of the segments. Traditionally, Gaussian Mixture Model - Universal Background Model (GMM-UBM) were the predominant technique for speaker modelling. Nevertheless, following the progress made in Speaker Verification, GMM-UBM have been replaced with discriminative speaker embeddings, such as i-vectors [75], d-vectors [331] and x-vectors [292]. The latter are the current state-of-the-art for both ASV and ASD [19, 236, 282].
Speaker representations are then clustered and assigned to a set of speakers. Commonly used clustering techniques include Agglomerative Hierarchical Clustering (AHC), Spectral Clustering and *k-means*.

AHC is by far the most used technique. When dealing with speaker representations that highly depend on the training data, however, Spectral Clustering has been proven to obtain better results. On the other hand, *k-means* clustering has been show to perform better in tasks where the number of speakers is known beforehand [106, 236, 338]. Finally, one can use the output of the clustering step to create new speaker representations and use them to re-segment the signal, to obtain a more refined result [12, 322].
The above-mentioned steps are common to many clustering-based approaches. However, there are many variations of this pipeline. For instance, several of the steps can be merged into a single module. An example of this is the Variational Bayes - Hidden Markov Model framework [81], which jointly models segmentation and clustering. Additional steps can also be introduced. For instance, overlap detection [160] can be introduced as an extra branch of the diarization pipeline, to address the issue of speaker overlap, an issue which clustering-based methods are mostly incapable of addressing [263]. On the other hand, the outputs of different models can be fused to improve the final diarization result [263, 306].
Notwithstanding its status as the predominant method, clustering-based ASD presents several shortcomings, chiefly: the inherent inability to handle speaker overlap without external components; most clustering-based ASD systems are composed of different modules that are individually optimised to perform a specific function, warranting careful calibration to ensure the best possible performance. For this reason, end-to-end ASD systems have started to receive growing attention. As opposed to modular clustering-based systems, end-to-end frameworks aim to solve the speaker diarization problem using a single neural network [101]. While still not fully matured, this type of approach has the advantage of being a single system, fully optimised for the diarization objective. Moreover, end-to-end systems are inherently able to deal with overlapped speech [236].
Besides the aforementioned challenges, one of the most prevalent challenges for speaker diarization systems is domain mismatch. ASD systems are often incapable of generalising to out-of-domain data, and only perform well in specific conditions. To bring forward new developments to overcome this obstacle, the DIHARD – *Diarization is Hard* – challenge series was introduced [277–279]. By providing the community with standardised evaluation over data taken from multiple domains, the DIHARD challenges have been one of the driving forces behind the significant progress made for ASD.

5.3 Privacy-preserving ASD

5.3.1 Baseline system

As a baseline system for our work, we adopted the DIHARD III challenge's baseline [279] – Fig. 5.1. Even though better performing systems were later submitted to this challenge (e.g., [118, 165, 343]), its relative simplicity and the modular nature of this cascaded approach is advantageous to its

privacy-preserving implementation as it easily allows the use of different cryptographic methods for different system modules.

The first step in this system is the extraction of MFCC features from short, overlapping speech frames. An implicit segmentation is assumed (the speech signal is partitioned in uniform and overlapping segments), and *x-vector* embeddings [292] are extracted from the resulting speech segments. The following step is to perform dimensionality reduction by applying Principal Component Analysis (PCA) over the zero-centred, whitened, and length-normalised speaker embeddings. This is followed by PLDA scoring. Agglomerative hierarchical clustering is then performed using the resulting scores. The baseline system includes a final re-segmentation stage, using a Variational Bayes - Hidden Markov Model (VB-HMM) [81].

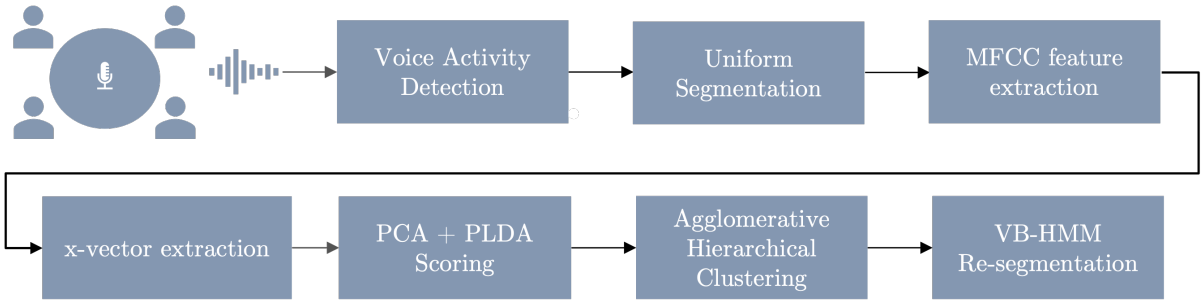


Figure 5.1: Baseline ASD system.

5.3.2 Simplified baseline system

To implement the system described in the previous section in a privacy-preserving way, it is first necessary to assess the computational cost and performance contribution of each block. Considering that this work represents a first approach to privacy-preserving ASD, we consider that it is reasonable to remove blocks with very high computational costs or limited contributions to the utility of the system to achieve a simpler pipeline that is feasible to implement with cryptographic techniques. Following this reasoning, the re-segmentation stage is dropped from the pipeline. This was done due to the computational complexity of this step and because it only provides limited improvement to the baseline system [279]. Similarly, the PCA+PLDA-based comparison of speaker embeddings is also considered to be too computationally expensive to be implemented with cryptographic techniques for this task. An implementation similar to that of Treiber et al. [323] could have been used. However, even with a dedicated system such as the one proposed in Treiber et al.'s work, comparing two vectors with a small-sized speaker embedding (e.g. 150 features) takes ~ 0.1 s. Though this value may seem low, for speaker diarization, it may represent a very high computational cost. For instance, for a short 2-minute speech recording to be diarized, assuming a uniform segmentation of 1.5 second-long frames with 0.25 s

overlap, we will have 480 speaker embeddings that need to be compared to each other. This amounts to a total of 114,960 comparisons, the equivalent of 11,496 s (or 191 minutes) to perform this step. As such, we replace PCA+PLDA scoring by the Euclidean distance between pairs of vectors. In addition to the above, for simplicity, in this work, we only consider Oracle Voice Activity Detection, a setting that corresponds to Track 1 of the DIHARD III challenge [279]. The final simplified system is represented in Figure 5.2, where grey boxes represent steps that are left out of the pipeline, the blue box represents the components computed by the user, the red box the components to be computed using SMC, and finally, the gold boxes represent the components to be computed by the service provider alone.

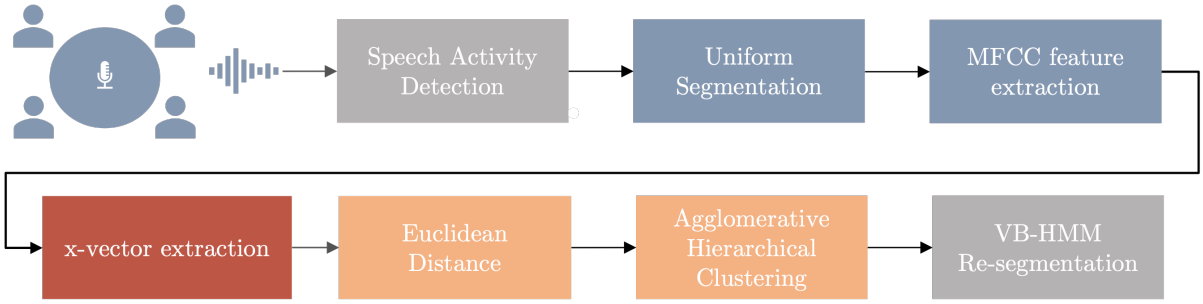


Figure 5.2: Simplified baseline ASD system.

5.3.3 Privacy-preserving system

Our final privacy-preserving system will consist of five steps: uniform segmentation and feature extraction, speaker embedding extraction, pairwise speaker embedding comparison, and agglomerative hierarchical clustering.

Similar to the previous two chapters, we assume that the user can perform steps 1 and 2 in the clear with limited computational cost. The reasoning for this decision rests on the high computational cost that would be involved in a remote privacy-preserving feature extraction process [316].

For step 3, we follow the work developed in Chapter 4 and assume that the speaker embedding extraction step is performed collaboratively between the user and server, using SMC, to protect the speaker embedding model from the user, and the user's data from the server. We also assume the existence of trusted parties that will participate in the computation to improve its efficiency.

Steps 4 and 5 in our pipeline correspond to the pairwise comparison of speaker embeddings and the clustering algorithm. We assume that these steps should be performed by the server to minimise the computational cost on the user's side and to allow for a level of flexibility in the overall diarization pipeline (i.e., the server can change the clustering algorithm without having to communicate with the user). However, to keep the privacy guarantees provided by Step 2 – the user does not have access to information about the model, and the server does not have access to information about the user's data

– we need to ensure that neither the user nor the server has access to the *x-vectors*. As such, to allow the server to be able to compare and cluster these vectors, we consider the use of SMH.

Considering that SMH is a non-invertible transformation, assuming that the SMH *key* is kept secret by the user and is not re-used and that the server does not have access to any non-transformed vector in the set, this would allow us to share the set of SMH vectors knowing that: 1) vectors cannot be meaningfully compared to vectors outside the set; 2) the only information the server can obtain is the spatial configuration of the set of vectors with regard to each other. To achieve the above, we need to ensure that the user does not have access to the *x-vectors* and that the server does not have access to the SMH key. To solve this, we can apply the SMH transformation using SMC, where the SMH key is secret-shared with the server and eq. 2.8 is collaboratively applied between the user and server to the already secret-shared *x-vectors*. The user can then send its resulting shares to the server, which can proceed with the comparison and clustering steps. The use of SMH will, however, require that the speaker embeddings are compared directly with the Hamming distance instead of the Euclidean distance, which will introduce some degradation. Moreover, further degradation may be introduced by cases where the Euclidean distance between pairs of vectors is larger than the SMH transformation saturation threshold.

The final privacy-preserving system is represented in Figure 5.3, where the blue box corresponds to the step to be performed by the user, the red box corresponds to the steps that are performed jointly between the user and service provider using SMC and the gold boxes correspond to steps being performed by the service provider.

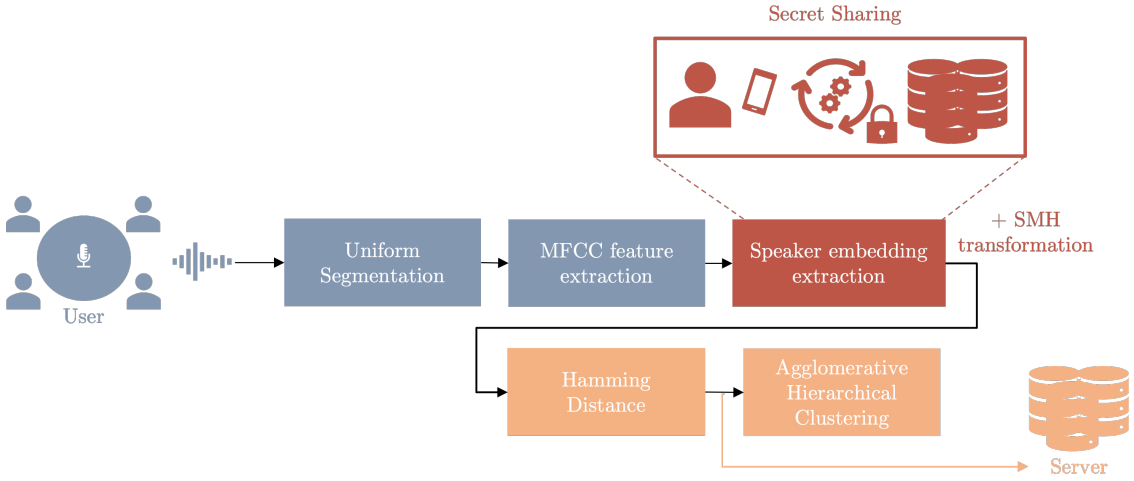


Figure 5.3: Final privacy-preserving ASD system.

5.4 Experimental Setup

5.4.1 DIHARD III corpus

The DIHARD III challenge dataset is a multi-domain dataset, with development (*dev*) and evaluation (*eval*) partitions, consisting of recordings of 5-10 minute-long samples drawn from 11 domains. These domains were selected to reflect variation in terms of recording equipment, recording environment, ambient noise, number of speakers, and speaker demographics. The simplest domain is the read audio-book domain. On the opposite extreme, we have conversations in noisy restaurant scenarios, with overlapping speech.

The *dev* partition includes 254 recordings, while the *eval* partition includes 260 recordings. In the experiments described in this work, we report our results with regard to the *core* subset of each of these partitions, using the metrics provided for this challenge [279].

5.4.2 Evaluation metrics

To evaluate the performance of each diarization system, we adopt the two metrics used in the DIHARD III challenge [279], Diarization Error Rate (DER) and Jaccard Error Rate (JER):

- DER is the sum of false alarm speech (FA) – the total system speaker time not attributed to a reference speaker; missed speech ($MISS$) – the total reference speaker time not attributed to a system speaker; speaker misclassification error ($ERROR$) – the total reference speaker time attributed to the wrong speaker; divided by the total reference speaker time ($TOTAL$) [278]:

$$DER = \frac{FA + MISS + ERROR}{TOTAL} \tag{5.1}$$

- JER is based on the Jaccard similarity index [277]. The goal of this metric is to provide an equal measure of the system's performance for each speaker in the recording. JER is computed for and averaged across all speakers. It is defined as the sum of false alarm (FA_i) and missed speech ($MISS_i$) for a given speaker, divided by the union between the reference speaker's total speaking time and the hypothesis speaker's total speaking time ($TOTAL_i$). Assuming N speakers [236]:

$$JER = \frac{1}{N}\sum_{i}^{N}\frac{FA_i + MISS_i}{TOTAL_i} \tag{5.2}$$

For computational and communications performance, we report inference times in seconds and communication in MB.

5.4.3 Speaker embedding extraction

Similar to what was done in Chapter 4, we used SpeechBrain's pre-trained *x-vector* model [265] for our experiments, which follows the architecture of [292], having been trained using the *dev* sets of Voxceleb 1 and 2 [61, 207], and achieving 3.2% EER on VoxCeleb 1's test set (Cleaned); *x-vectors* were extracted using 1.5 s windows and 0.25 s shift.

5.4.4 Privacy-preserving implementation

The speaker embedding extraction network and SMH transformation were implemented with the MP-SPDZ library [141], using two protocols: the 3-party *semi-honest* RSS scheme of [13] and the 4-party RSS scheme of [69], which provides *malicious* (Mal) security in the *honest majority* setting against one corrupted party. We selected these protocols as they were found to be the most efficient to extract *x-vectors* in Chapter 4. For both protocols, we used local share conversions and probabilistic truncation to improve efficiency [68, 69].

Considering the fact that we need to extract multiple *x-vectors* from each file to be diarized, we performed experiments using different batch sizes of *x-vectors*: 256, 1024 and 2048. These values were selected based on the statistics of the number of *x-vectors* extracted per file in the DIHARD III *dev* set, encompassing an interval covering roughly 85% of the data (i.e., 85% of the considered recordings contain 2048 *x-vectors* or fewer. However, due to implementation limitations of the cryptographic library being used, we were unable to extract batch sizes larger than 700 directly. As such, the values corresponding to 1024 and 2048 were linearly estimated from the cost of extracting a batch size of 700. For the SMH transformation, given that the overall computation is lighter, we were able to compute the cost for these batch sizes directly.

Our experiments assume the default parameters for 40-bit security. We used the library's fixed-point number representation, adopting the default configuration of 16 bits for the decimal part and 15 bits for the fractional part. All tested protocols perform computations modulo 2^k, where $k = 64$. For the SMH transformation, we use the following parameters: $k = 2$, $\delta = 225.0$ and $mpc = 4$. Experiments were performed on a machine with 24 Intel(R) Xeon(R) CPU E5-2630 v2 @ 2.60GHz processors and 250GB of RAM.

5.5 Results

5.5.1 Computational and communication costs

Table 5.1 presents the computational and communication costs for both the extraction of *x-vectors* and the SMH transformation.

Table 5.1: Computational and communication costs obtained for the extraction of *x-vectors* and SMH transformation. Values denoted with $^{\$}$ were linearly estimated from a batch size of 700. All results were obtained by averaging over 100 runs.

Protocol	Security Model	Batch Size	*x-vector* extraction		SMH transformation	
			Time (s)	Comm. (MB)	Time (s)	Comm. (MB)
3-party RSS [13]	HM/SH	256	73.27 ± 0.41	1,691.03	0.90 ± 0.04	12.92
		1024	280.05 ± 2.77$^{\$}$	6,564.95$^{\$}$	2.88 ± 0.04	72.1
		2048	560.11 ± 5.53$^{\$}$	13,129.90$^{\$}$	4.90 ± 0.05	135.63
4-party RSS [69]	HM/Mal	256	108.50 ± 1.34	5,098.15	2.03 ± 0.04	62.89
		1024	404.12 ± 5.25$^{\$}$	19,890.61$^{\$}$	4.53 ± 0.04	176.84
		2048	808.25 ± 10.50$^{\$}$	39,781.23$^{\$}$	7.20 ± 0.05	331.32

Concerning the *x-vector* extraction, we can see that, in the average case, for 1024 *x-vectors*, our method takes ∼5 mins and ∼6.5 GB for the 3-party setting, and ∼7 mins and ∼19.5 GB, for the 4-party setting[1]. Taking into account that a single *x-vector* represents ∼0.25 seconds of speech, 1024 *x-vectors* represent roughly 4 mins of speech, without any silence, corresponding to real-time factors of 1.1 and 1.6, for 3- and 4-party RSS, respectively.

When extracting 2048 *x-vectors*, in the 3-party setting, our implementation takes an average of ∼ 9 minutes and requires a total of ∼12 GB of communication per party. For the 4-party setting, our method takes ∼13.5 minutes and ∼38 GB of data. In this case, we see that the 4-party protocol starts to become very inefficient.

In terms of the SMH transformation, similar to the *x-vector* extraction, the 3-party RSS setting is more efficient. However, in this case, the added cost of the 4-party setting can be deemed acceptable – though statistically significant, with $p < 0.001$, following the same assumptions as above – particularly if we consider that the overall cost of the SMH transformation is smaller than that of the *x-vector* extraction by close to 2 orders of magnitude, making it negligible in comparison.

5.5.2 Diarization results

Having discussed how to extract *x-vectors* and apply SMH, our last step is to compare the transformed vectors using the Hamming distance and cluster them using agglomerative hierarchical clustering. As stated in Section 5.3.1, the original baseline system uses PCA reduction, PLDA scoring and re-segmentation. In contrast, our system directly clusters the SMH-transformed *x-vectors*. Because of this, and to better assess the degradation introduced by SMH, we provide results with and without these components.

These results can be found in Table 5.2, wherein the DIHARD III baseline corresponds to our baseline system, the *Simplified baseline* corresponds to the baseline system without PCA, PLDA and

[1]This added cost was found to be statistically significant, with $p < 0.001$, using the values obtained for a batch size of 256, through an independent samples *t-test*, and assuming a normal distribution of residuals of each measurement regarding the overall sample mean.

re-segmentation steps, and PP-ASD denotes our privacy-preserving system.

From the table, we can see that for the *dev* set, the simplified baseline introduces a degradation of ∼4% in terms of DER and ∼9% in terms of JER. Further, transforming the *x-vectors* with SMH causes an additional degradation of ∼2.5% DER and ∼10% JER, most likely due to the saturation property of SMH. Our final system thus introduces a total degradation of ∼7.5% DER and ∼19% JER when compared to the baseline.

Table 5.2: Results obtained for each ASD system.

System	Development		Evaluation	
	DER (%)	JER (%)	DER (%)	JER (%)
DIHARD III baseline [279]	20.25	46.02	20.65	47.74
Simplified baseline	25.36	55.75	25.15	54.57
PP-ASD	27.95	65.52	29.58	67.72

When looking at the *eval* results, while the degradation remains similar from the original to the simplified baseline, we observe a more substantial degradation – ∼ 9% DER and ∼ 20% JER – when applying SMH. We hypothesise that this is due to the non-linear relation between the Euclidean and Hamming distances, which makes our system more sensitive to the clustering threshold, which was optimised for the *dev* set.

5.5.3 Per-domain analysis

Table 5.3: Results for the Clinical and MapTask domains for the baseline and privacy-preserving systems using task-specific thresholds selected for the *dev* set. Values on the left-hand (resp. right-hand) side of → indicate the result obtained for the original (resp. adapted) thresholds.

System	**Domain**	Development		Evaluation	
		DER (%)	JER (%)	DER (%)	JER (%)
Simplified baseline	Clinical	15.28→14.7	21.71→21.0	15.82→14.02	25.94→25.27
	MapTask	11.01→10.64	18.48→19.07	8.66→8.53	15.55→15.45
PP-ASD	Clinical	37.56→16.65	67.36→25.73	39.53→19.17	68.71→35.46
	MapTask	24.34→12.72	54.31→26.99	23.61→15.81	59.64→33.30

To have a better understanding of the effects of SMH with regard to the simplified baseline system's performance, we also decided to look into the system's results on a per-domain basis. We found that even though the overall degradation is close to 2.5% in terms of DER and 10% in terms of JER for most domains, for some domains, namely for the Clinical – recordings of autism-screening interviews – and MapTask – recordings of pairs of sleep-deprived individuals who have to collaborate to reproduce the path shown in a map held by one of the participants, on the other's map – the degradation introduced by SMH was disproportionate with regard to the baseline: an absolute degradation of 22% (24%) DER

and 46% (43%) JER for the Clinical domain, and of 13% (15%) DER and 36% (44%) JER for the MapTask domain, in the *dev* (*eval*) set.
We again claim that this is a direct effect of the sensitivity to the threshold selection due to the saturation effect introduced by SMH and the fact that a single threshold was selected for all domains. To verify this hypothesis, we decided to adjust the threshold for each of the *dev* set domains individually. This was done for both our simplified baseline system and for our final private system to provide a fair comparison.
The results related to this experiment are provided in Table 5.3 for the two domains mentioned above for the sake of space. We can see that by adjusting the threshold, our simplified baseline results show only small levels of improvement. On the other hand, for the privacy-preserving system, the results highly improve: ∼21% (20%) DER and ∼42% (33%) JER for the Clinical domain and ∼12% (8%) DER and ∼28% (26%) JER for the MapTask domain, for the *dev* (*eval*) set. When comparing these results to the new baseline results, we see that the SMH-introduced degradation is close to 2.5% DER and less than 10% JER in both settings, which is in line with the average degradation, thus proving our hypothesis.
The above shows that the privacy-preserving system is more sensitive to domain changes, making it less reliable than the baseline system. However, in a real-world application, it is reasonable to assume that one can ask a potential user to define the setting in which the recording under evaluation was made so that the system can use a domain-specific threshold to optimise performance.

5.6 Summary

In this chapter, we presented the first implementation of a privacy-preserving speaker diarization system using existing cryptographic techniques. The contributions of this work are not limited to the setting of ASD, as we introduce an approach to apply SMH in a privacy-preserving way, using SMC, which has potential applications in the area of template protection.
This system still has limitations, both in terms of ASD performance and computational cost, but we foresee many possible improvements. Considering that the computational bottleneck is the *x-vector* extraction, reducing the size of this model could help improve efficiency. At the same time, exploring other techniques for the SMH-based clustering could help mitigate the degradation of the results. Future work should also deal with privacy-preserving feature extraction, voice activity detection, re-segmentation or overlap detection algorithms, all of which are necessary to implement end-to-end private speaker diarization pipelines. Moreover, tasks such as voice activity detection and feature extraction extend beyond ASD to numerous speech tasks.

6

Adversarial Examples against Speaker Identification

In this chapter, we propose a white-box adversarial attack to fool speaker identification using highly imperceptible adversarial perturbations, an attack that we dub as *FoolHD*. Our approach uses a Gated Convolutional Autoencoder (GCA) that operates in the Modified Discrete Cosine Transform (MDCT) domain, being trained with a multi-objective loss function to generate and conceal adversarial perturbations within speech signals. In addition to hindering speaker identification performance, this multi-objective loss accounts for human perception through a frame-wise cosine similarity of MFCC feature vectors extracted from the original and adversarial signals.

6.1 Introduction

Machine learning models have been shown to be vulnerable to adversarial attacks – i.e. perturbations to the inputs of these models that cause them to output wrong predictions [308]. Adversarial perturbations can be generated under different assumptions and with different objectives. For instance, in the case of a speaker identification model, an *untargeted* attack pushes the model to misidentify the speaker for a given speech sample. In contrast, a *targeted* attack attempts to force the model to identify a specific speaker chosen by the attacker. The attacker's knowledge about the speaker identification model can also vary [1]. In a *white-box* setting, all the model-related information, such as its architecture and parameters, is available to the attacker (e.g. open-source models). Contrarily, in a *black-box* setting, the attacker's knowledge of the classifier consists – at most – of the model's outputs (e.g. public APIs). Many adversarial example-creation methods also have the secondary goal of making adversarial perturbations imperceptible to humans, to prevent listeners from detecting them, and from being able to distinguish the adversarial examples from the original signal [1].

In this chapter, we propose a novel adversarial attack to generate imperceptible adversarial speech perturbations against speaker identification by leveraging steganography techniques. As stated by Ian Goodfellow, the creation of adversarial examples can be seen as "*accidental steganography*" [113]. Even though adversarial examples and steganography have different goals, both try to hide a message in a carrier such that this message does not perceptually affect the carrier.

To this end, our proposed attack, FoolHD, adapts the speech steganography method proposed by [156], wherein a frequency-domain GCA [73] is exploited to embed one or more speech samples (i.e. messages) in another speech sample (i.e. carrier). In particular, we train the GCA to generate adversarial speech signals whose perturbations are imperceptible to the human auditory system against a *white-box* speaker identification model. We achieve these contrasting objectives with a multi-objective loss function that combines a perceptual loss function and an adversarial loss function. The former tries to make the adversarial signals perceptually close to the original signals, whereas the latter aims to mislead the speaker identification model in both *untargeted* and *targeted* settings. We validate the effectiveness of FoolHD using a 250-speaker identification x-vector network, trained with

VoxCeleb [206], in terms of accuracy, success rate, and imperceptibility.

This chapter is organised as follows: Section 6.2 provides an overview of the related work; Section 6.3 introduces our proposed method; Section 6.4 provides details on our experimental setup. In Section 6.5 we present and discuss the obtained results. Finally, Section 6.6 provides a summary of the chapter.

6.2 Related Work

Most of the existing adversarial attacks [129] against speaker identification models exploit state-of-the-art methods originally developed for image classification, focusing mainly on the Fast Gradient Sign Method (FGSM) [113]. The FGSM attack tries to find a small perturbation δ_x, such that for an example x, and a neural network classifier f with parameter set θ:

$$f_\theta(x + \delta_x) \neq f_\theta(x), \tag{6.1}$$

In other words, FGSM tries to find a perturbation δ_x that changes the output of the classifier when added to x. To do so, FGSM selects δ_x as follows [155]:

$$\delta_x = \epsilon \times \text{sign}(\nabla_x \mathcal{L}(f_\theta(x), y)), \tag{6.2}$$

where $\mathcal{L}$ corresponds to the training loss of the classifier, y the label of the example, sign corresponds to the *sign* function applied to the gradient of the loss function, and ϵ controls the magnitude of the perturbation. In essence, to change the output of the classifier for example x, FGSM selects a perturbation δ_x that moves x in the opposite direction of the minimum gradient.

Using this attack, Kreuk et al. [155] and Li et al. [175] explored the adversarial vulnerability of *x-vector* and *i-vector* -based speaker verification models, respectively. Li et al. [176] additionally integrate an estimate of room impulse responses with FGSM to generate adversarial speech signals that are still effective when played over-the-air against an *x-vector*-based speaker verification system. Differently, Li et al. [174] tried to learn universal adversarial perturbations by adversarially training a perturbation generator against a SincNet-based speaker identification model.

However, even though these attacks achieve high success rates in misleading classifiers, most present high levels of distortion, neglecting the impact that adversarial perturbations have on human perception. For example, the Perceptual Evaluation of Speech Quality (PESQ) [120] score of the adversarial speech samples generated by the universal adversarial perturbation of [174] is only 3 out of 5. Recently, Wang et al. [340] improved the imperceptibility of an FGSM-based attack against an *x-vector* speaker identification model by exploiting frequency masking. However, their method trades off imperceptibility against the success rate of their adversarial examples (i.e. at the highest

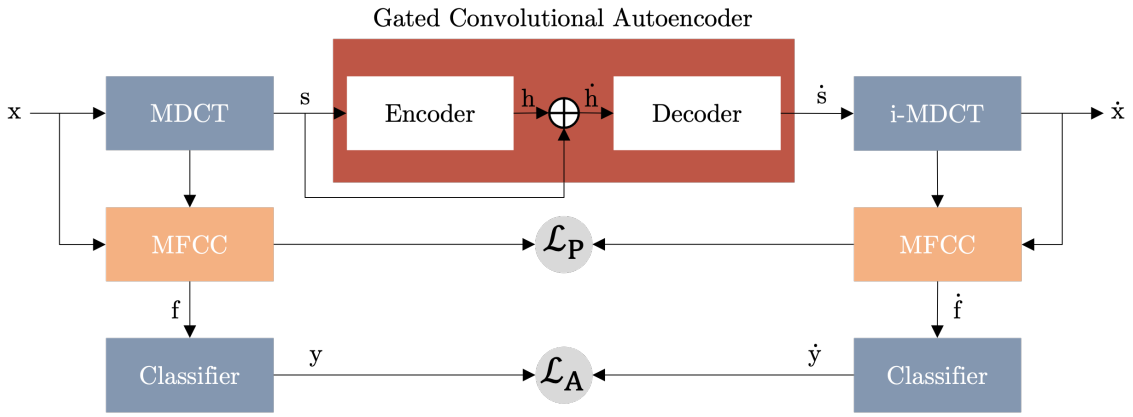


Figure 6.1: Overview of proposed attack.

imperceptibility score of 4.23, as measured by PESQ, the success rate is ∼73%).

6.3 Method

Let $\mathbf{x} \in \mathbb{R}^{1\times D}$ be an (original) speech signal and $f(\cdot)$ be an N-speaker identification model that predicts the most likely speaker

$$y = f(\mathbf{x}) = \underset{i=1,\ldots,N}{\arg\max}\, p_i, \tag{6.3}$$

where p_i is the predicted probability of speaker i, computed by normalising the i-th predicted logits, z_i, using a softmax operation:

$$p_i = \frac{e^{z_i}}{\sum_{n=1}^{N} e^{z_n}}. \tag{6.4}$$

We aim to generate an adversarial speech signal, $\dot{\mathbf{x}} \in \mathbb{R}^{1\times D}$, by perturbing $\mathbf{x}$ such that it *misleads* the speaker identification model while ensuring the perturbed signal remains perceptually similar to the original one (*imperceptible perturbation*).

To this end, we propose the use of a frequency-domain GCA to generate and embed the desired adversarial perturbations in the input signal (see Figure 6.1).

The architecture of our model is inspired by the steganographic method of [156]. Specifically, the proposed model comprises an encoder and a decoder. The encoder, $\mathrm{E}(\cdot)$, creates a latent representation, $\mathbf{h} = \mathrm{E}(\mathbf{s})$, of the spectral representation of the input signal $\mathbf{s}$, using three gated convolutional layers. This latent representation, $\mathbf{h}$, is concatenated with the original input using a skip connection to obtain the joint representation $\dot{\mathbf{h}} = [\mathbf{h}; \mathbf{s}]$ where ; denotes the concatenation operation. In turn, the decoder, $\mathbf{D}(\cdot)$, takes $\dot{\mathbf{h}}$ and creates an adversarial spectral representation of the speech signal as $\dot{\mathbf{s}} = \mathrm{D}(\dot{\mathbf{h}})$, feeding the latent representation through four gated convolutional layers. Similar to [156], each gated convolutional layer of both the encoder and the decoder is composed of 64, 3×3 kernels, followed by a

batch normalisation and a dropout layer.

Our model has two main differences from [156]. First, it operates over a variation of the Discrete Cosine Transform (DCT) type IV, called the MDCT [258], as opposed to the Short-time Fourier Transform (STFT) domain. The STFT is a complex-valued transformation that represents a speech signal by its magnitude (real component) and phase (complex component). Therefore, the phase and magnitude of the signal are processed separately, which may result in reconstruction errors of the processed signal during the STFT inversion due to magnitude and phase mismatches [156]. This reconstruction error can be avoided by using the MDCT, as the MDCT is a real-valued transform whose frequency coefficients encode both the phase and magnitude of the signal [363].

Second, instead of providing an external steganographic message to the model [156], our model will *learn* the adversarial perturbation and hide it within the input audio file. Specifically, instead of optimising a perturbation to fool the downstream classifier, we optimise the model's weights such that the adversarial perturbation is embedded in the input signal by feeding it through the model.

The input of the model is normalised to zero mean and unit variance to prevent amplitude mismatches between original and adversarial signals. To do so, we save the input's statistics to re-normalise the model's output to ensure it has the same mean and standard deviation as the input.

We train our model end-to-end, by back-propagating errors captured by two loss functions: a perceptual loss, which accounts for the perceptual differences between the original and adversarial audio files, and an adversarial loss, which induces speaker misclassification.

6.3.1 Perceptual and adversarial losses

It is well known that humans cannot perceive specific changes introduced to speech and audio signals [144]. Since the goal of this work is to create imperceptible perturbations, we have to design our model to take human perception into account. To do so, we propose a perceptually inspired loss that exploits the perceptually inspired MFCC features, which were designed to mimic the human auditory system [1, 288]. Specifically, we propose a loss that compares the model's input and output signals by computing the pairwise similarity of MFCC feature frames between the two signals extracted from the reconstructed time-domain signal. This perceptual loss is defined as follows:

$$\mathcal{L}_P(\mathbf{x}, \dot{\mathbf{x}}) = \sum_{t=1}^{T} 1 - S_{\cos}(\mathbf{f}_t, \dot{\mathbf{f}}_t), \tag{6.5}$$

where $S_{\cos}(\cdot, \cdot)$ is the cosine similarity, defined as:

$$S_{\cos}(\mathbf{f}_t, \dot{\mathbf{f}}_t) = \frac{\mathbf{f}_t \cdot \dot{\mathbf{f}}_t}{\|\mathbf{f}_t\| \|\dot{\mathbf{f}}_t\|} = \frac{\sum_{i=1}^{F} f_i \dot{f}_i}{(\sum_{i=1}^{F} f_i^2)^{1/2} (\sum_{i=1}^{F} \dot{f}_i^2)^{1/2}}, \tag{6.6}$$

and $\mathbf{f}_t \in \mathbb{R}^{1\times F}$ and $\dot{\mathbf{f}}_t \in \mathbb{R}^{1\times F}$ are MFCC feature vectors extracted from the original and perturbed signals, respectively, at time frame t. We use the cosine similarity rather than the negated or inverted Euclidean distance since the latter is sensitive to magnitude differences between the signals, while the former only focuses on differences in the spectral structures of the two signals.

To train the proposed model to mislead the speaker identification classifier, we use the Carlini-Wagner (CW) attack, wherein the adversarial loss takes the errors of the output logits of the classifier with regard to the type of attack (i.e., untargeted or targeted attack) [44]. In particular, for an untargeted attack, the adversarial loss, $\mathcal{L}_{\mathrm{A}_{\mathrm{untarg}}}(\cdot,\cdot)$, aims to lower the values of logits, $\dot{z}_y$, that correspond to the true speaker y of the speech sample x,

$$\mathcal{L}_{\mathrm{A}_{\mathrm{untarg}}}(\mathbf{x},\dot{\mathbf{x}}) = \dot{z}_y - \max_{\substack{i=1,\ldots,N \\ i\neq y}} \dot{z}_i, \tag{6.7}$$

where, $\dot{y}$ will correspond to any other speaker:

$$\dot{y} = f(\dot{\mathbf{x}}) \neq y. \tag{6.8}$$

with C being the speaker classifier.

The targeted version of our method, FoolHD-t, aims to force the speaker identification model to predict a specific target class, y_{targ}, chosen by the attacker:

$$\dot{y} = y_{\mathrm{targ}} = f(\dot{\mathbf{x}}) \neq y. \tag{6.9}$$

To this end, we use the targeted adversarial loss, $\mathcal{L}_{\mathrm{A}_{\mathrm{targ}}}(\cdot,\cdot)$, that increases the value of the logit, $\dot{z}_{y_{\mathrm{targ}}}$ that corresponds to the target speaker, y_{targ},

$$\mathcal{L}_{\mathrm{A}_{\mathrm{targ}}}(\mathbf{x},\dot{\mathbf{x}}) = \max_{\substack{i=1,\ldots,N \\ i\neq y_{\mathrm{targ}}}} \dot{z}_i - \dot{z}_{y_{\mathrm{targ}}}. \tag{6.10}$$

The target speaker is selected randomly [162] among all other speakers.

Our final objective function $\mathcal{L}(\cdot,\cdot)$ thus accounts for both the perceptual loss, $\mathcal{L}_{\mathrm{P}}(\cdot,\cdot)$, and (untargeted or targeted) adversarial loss, $\mathcal{L}_{\mathrm{A}}(\cdot,\cdot)$:

$$\mathcal{L}(\mathbf{x},\dot{\mathbf{x}}) = \mathcal{L}_{\mathrm{P}}(\mathbf{x},\dot{\mathbf{x}}) + \alpha\mathcal{L}_{\mathrm{A}}(\mathbf{x},\dot{\mathbf{x}}). \tag{6.11}$$

where the hyper-parameter α is a weight chosen empirically. The proposed model then generates adversarial examples by minimising this objective function over M iterations, with the value of M being set experimentally (see Section 6.4.3).

6.4 Experimental Setup

In this section, we describe the dataset used for our experiments, the speaker identification network under attack, and the implementation details of the GCA[1].

6.4.1 Dataset

The VoxCeleb dataset [207], as described in previous chapters, contains speech from 7,363 speakers of multiple ethnicities, accents, occupations and age groups. Among these, we randomly chose 250 speakers for our experiments, for consistency with the state-of-the-art [129, 176, 353]: 125 female speakers, and 125 male speakers. All speech samples were down-sampled to 8 kHz to match the sampling rate of our pre-trained speaker identification model. Moreover, all files were split into 4 seconds-long segments. Our test set is composed of 10 segments per speaker, amounting to a total of 2,500 segments.

6.4.2 Speaker identification model architecture and training

We conduct our white-box attack against an *x-vector* speaker identification network, following the architecture of [292]. This network is composed of three blocks: a first block that operates at the frame level and two others that operate at the utterance level. The first block is composed of five time-delay layers with a small temporal context. These layers work as a 1-dimensional convolution, with a kernel size corresponding to the temporal context. In the second block, an attentive statistical pooling layer [225] weighs each time frame according to its importance and then computes utterance-level statistics (mean and standard deviation) across the time dimension, providing a summary for the entire speech file. The third and final block takes this summary and propagates it through a set of three fully connected layers, followed by a Softmax layer from which a prediction is obtained. All time delay and fully connected layers were followed by a batch normalisation layer, a ReLU activation layer and a dropout layer. The network's input features follow Kaldi's *x-vector* recipe [257] feature configuration, corresponding to 29 MFCCs + 1 log-energy feature, extracted over 25 ms-long windows with 10 ms shift, from each 4 seconds-long file, resulting in 400 frame-long sequences of 30 feature coefficients. Each frame is mean-normalised using a sliding window. Non-speech frames are removed via an energy-based voice activity detection module.

This model was implemented in Pytorch and trained over the complete dev set of VoxCeleb 1 and 2 plus VoxCeleb 2's test set, amounting to 7,323 speakers. The model was trained for 100 epochs, with a learning rate of $1e^{-3}$, a learning rate decay of $5e^{-2}$ with a period of 30 epochs, a dropout value of $1e^{-3}$ and the Adam optimiser. These hyper-parameters were optimised for a dev subset consisting of $\sim 30\%$ of the dataset. After these parameters were selected, the model was re-trained using all available data.

[1] Source code and audio samples are available at https://fsepteixeira.github.io/FoolHD/

Training samples were augmented with randomly selected Room Impulse Responses and sounds taken from the MUSAN corpus [292]. This network was further adapted to our 250 test speakers without data augmentation. To this end, all the neurons in the network's classification layer that did not correspond to this set of 250 speakers were dropped. The network was trained for another 100 epochs, with a learning rate of $1e^{-5}$, using training data from the 250 speakers without overlap with our test samples, achieving a final accuracy of 98.3% on our test set.

6.4.3 Adversarial attack implementation

We implemented and trained our model in Pytorch using the Adam optimiser with a learning rate of $1e^{-3}$, similarly to [156]. We also used a weight decay of $1e^{-5}$ and a dropout value of $1e^{-3}$ to help the network converge. The number of training iterations used to generate the adversarial examples, M, was set experimentally as a trade-off between the running time and multi-objective losses. For the untargeted attack, we set $M = 500$. However, we increase the value of M to $1,000$ for our targeted attack. While $M = 500$ is sufficient to generate imperceptible perturbations that can mislead the classifier for the untargeted attack, in the targeted setting, the target speaker is potentially far from the original speaker, requiring more iterations to obtain similar results. Within this number of iterations, we select as an adversarial example the sample with the lowest perceptual loss that can satisfy our target task (i.e., fool the network). The weight α was set to 1, as preliminary experiments showed that for the selected number of iterations, this value already guaranteed the success and quality of the adversarial examples.

6.5 Results

We evaluate the performance of FoolHD in terms of effectiveness (i.e., model accuracy and attack success rate) and imperceptibility of the adversarial examples for both untargeted and targeted attacks. The *untargeted success rate* (S) is defined as the ratio between the number of adversarial examples that successfully mislead the speaker identification model and the total number of adversarial examples. The *targeted success rate* (S-t) is defined as the ratio between the number of adversarial examples that successfully induce the speaker identification model into predicting the target speaker y_{targ}, and the total number of adversarial examples. Since our speaker identification model is not 100% accurate when classifying our test files, we also report the accuracy (i.e. ratio between correctly predicted files with regard to the ground truth and total number of files) of the speaker identification model for both the original samples and the adversarial examples.

To assess the imperceptibility of the adversarial attacks, we use two perceptual audio metrics: Perceptual Evaluation of Speech Quality (PESQ) [120] and Just Noticeable Differences (JND) [188].

Attack	Effectiveness			Imperceptibility		
	Acc. (%) ↓	S (%) ↑	S-t (%) ↑	JND ↓	PESQ ↑	WER (%) ↓
Baseline	98.3	-	-	-	-	-
FoolHD-MSE	0.1	99.9	-	2.93 ± 1.19	1.44 ± 0.55	-
FoolHD-noSkip	1.2	99.5	-	0.97 ± 0.75	4.34 ± 0.10	-
FoolHD	1.2	99.6	-	0.97 ± 0.77	4.37 ± 0.08	21.7
FoolHD-t	0.1	99.9	.992	1.20 ± 0.86	4.30 ± 0.10	28.0

Table 6.1: Impact of the proposed perceptual loss, skip-connection and targeted FoolHD (FoolHD-t) on the effectiveness – Accuracy (Acc.), Success rate (S) and targeted Success rate (S-t) – and imperceptibility – JND, PESQ, WER – of the resulting adversarial examples.

PESQ scores cover a scale from 1 (bad) to 5 (excellent) [120]. JND is defined as the l_1 norm of the difference between the representation of the original and adversarial audio files, computed by a neural network trained using pairs of audio files whose similarity was judged by humans [188]. To have a better understanding of the range of JND, we evaluated this metric using four scenarios: *sample against itself*, *sample vs all zeros*, *sample vs Gaussian noise*, *sample versus another sample*. With this study, we concluded that JND scores are bounded between 0.0 (*sample vs itself*) and $\sim$ 5.0 (*sample versus random noise*).

We also evaluate how our method affects the performance of an ASR system as a way to measure its utility in other tasks and as an additional metric for imperceptibility. To this end, we used an end-to-end ASR system (i.e., ESPNet [345]) to transcribe the perturbed samples. The results are reported in terms of Word Error Rate (WER) (%). Since VoxCeleb does not include transcriptions, the transcriptions of the original samples were used as the gold standard. Comparing the WER between the transcriptions of the original and perturbed samples will allow us to have an additional view of the degradation caused by the adversarial perturbations.

6.5.1 Ablation study and analysis

To validate the effectiveness of our perceptual loss and the model's skip connection, we present two analyses of FoolHD (untargeted). We first consider FoolHD-MSE, a modification of the proposed method, where the perceptual loss is replaced by the mean square error (MSE) between the input audio file and the adversarial audio file. Secondly, we test our model without any skip connection from the input to the output of the encoder – FoolHD-noSkip. In addition to the above, we further validate the performance of the targeted version of our attack, FoolHD-t.

Table 6.1 shows the effect of generating adversarial examples using the proposed perceptual loss and the skip connection. In general, the success rate of adversarial examples generated by FoolHD, FoolHD-MSE and FoolHD-noSkip are similar and above 99%. However, the perceptual loss of FoolHD improves the imperceptibility of the adversarial examples. For instance, the average PESQ scores of the

Attack	Effectiveness		Imperceptibility		
	Acc. (%) ↓	S (%) ↑	JND ↓	PESQ ↑	WER (%) ↓
FGSM [113]	36.9	63.6	1.29 ± 1.01	3.21 ± 0.63	53.7
BIM [161]	0.4	100.0	1.05 ± 0.88	3.36 ± 0.61	51.5
FoolHD	1.2	99.6	0.97 ± 0.77	4.37 ± 0.08	21.7

Table 6.2: Comparing the effectiveness – Accuracy (Acc.) and Success rate (S) – and imperceptibility of FoolHD (untargeted) with Fast Gradient Sign Method (FGSM) and Basic Iterative Method (BIM).

FoolHD and FoolHD-MSE are 4.37 and 1.44, respectively.

When comparing the results of FoolHD and FoolHD-noSkip, we observe that the skip connection provides very slight improvements. We hypothesise that the skip connection might prevent vanishing gradients in the backward pass and provide the decoder with information about the original input during the forward pass. Nonetheless, these results are inconclusive and cannot be considered statistically significant (differences in the success rate correspond to the model incorrectly predicting 2 additional samples), and further experimentation is required to assess the skip connection's contribution.

Table 6.1 also shows that FoolHD-t has a 99.2% success rate in misleading the speaker identification model into predicting any arbitrary speaker. In addition, FoolHD-t drops the accuracy of the speaker identification model to very close to zero. However, this may be due to the fact that FoolHD-t is trained for 1,000 iterations, whereas FoolHD only uses 500. However, the imperceptibility of FoolHD-t as measured by PESQ, JND and WER is slightly worse than that of FoolHD. We hypothesise that this is because FoolHD-t needs to introduce more drastic perturbations in the original signal than FoolHD, given that this version of the attack needs to not only mislead identification but also reach specific target classes which may be far from the original classes.

6.5.2 Comparison with other adversarial attacks

We compare FoolHD with two baseline attacks: FGSM [113] and Basic Iterative Method (BIM) [161] (we excluded the method proposed by Wang et al. [340] from this comparison, as no source code was available at the time of writing). We selected ϵ=0.004 for both methods to trade-off between the effectiveness and imperceptibility of adversarial audio files. We used 10 iterations for BIM.

Table 6.2 compares the effectiveness and imperceptibility of FoolHD with FGSM and BIM. FGSM only achieves a 63.6% success rate in misleading the speaker identification model. However, BIM improves the success rate of FGSM to 100% by iteratively tailoring the adversarial perturbations towards misleading the speaker identification model. In comparison, the success rate of FoolHD is 99.6%. While the speaker identification model can still recognise 36.9% of the FGSM adversarial speech samples correctly, only 0.4% and 1.2% of BIM's and FoolHD's adversarial samples are correctly classified by the speaker identification model.

The imperceptibility scores of FGSM and BIM in Table 6.2 show that bounding the l_∞ norm of adversarial perturbations by ϵ is not enough for having perceptual similarities between original and adversarial audio files. FoolHD achieves very remarkable improvements in terms of PESQ scores when compared to FGSM and BIM, not only in average but also the standard deviation is significantly lower, showing that there is little variability in the (high) quality of our generated audio files. On the other hand, in terms of JND, although less stark, we are still able to see some improvement in both the average and standard deviation of the results. In terms of WER, the relative improvement provided by FoolHD amounts to ∼60% with regard to the results obtained for FGSM and BIM. This adds to the evidence that FoolHD can provide highly imperceptible adversarial examples and that the adversarial perturbation does not substantially affect the utility of the sample for other tasks.

6.5.3 Robustness experiments

A possible extension of the work developed in this chapter is privacy protection. When publishing speech files to online services and applications, similar methods could be used to help deter the automatic collection of speech data for a given individual. However, to reach this goal, this method would need to be extended to fulfil a set of necessary properties. Specifically, the method should be robust to signal perturbations such as compression algorithms, that are commonly used to transmit and store speech signals. The method should also be robust to room impulse responses, that mimic over-the-air play.

Having this application in mind, additional experiments were performed to evaluate the robustness of our method against Room Impulse Responses (which simulate over-the-air play) and MP3 compression. These experiments were performed over smaller subsets of data and, thus, do not match the conditions of the remaining results presented for our method. For this reason, we do not report them as comprehensively as the experiments above and instead focus on analysing them qualitatively.

Concretely, we augmented the adversarial samples with room impulse responses for small, medium and large rooms and compressed them using MP3, with varying levels of quality. Overall, each of these individual transformations created a drop of ∼60% in the success rate of the attack, showing that it cannot be considered robust to them.

6.6 Summary

In this chapter, we presented a steganography-inspired method to generate adversarial examples against speaker identification. To this end, we trained a GCA using a perceptually motivated multi-objective loss. We showed that our method is capable of generating imperceptible adversarial examples that are highly successful in attacking a speaker identification model for both untargeted and targeted scenarios.

As mentioned Section 6.5.3, a possible extension of the work developed in this chapter is privacy protection. To reach this goal, the method would need to be made robust to compression algorithms and to room impulse responses that mimic over-the-air play, however, the preliminary experiments reported in this section have shown that the method is not yet robust to these transformations. As such, it would be interesting to explore methods that would improve the robustness of the proposed method. In addition to robustness guarantees, the method would also need to be made transferable to other speaker identification models. In a real-world scenario, the true speaker identification model is likely to be at least partially unknown to the adversarial "attacker". Finally and importantly, the proposed method would need to fool not only speaker identification models but also real-world speaker verification systems. We have shown that our method can fool speaker identification. Still, there is no guarantee that it will be able to fool a speaker verification system working with template embeddings on an open set of speakers. All of these ideas are interesting extensions of this method that may be worth exploring as future work.

Contrarily to other chapters in this thesis, this work does not represent a direct method to protect speech privacy in remote processing. However, its development prompted a reflection on the required properties of machine learning methods for privacy. For instance, the informal robustness experiments reported in this chapter show that adversarial perturbations are not always robust to simple signal manipulations. In contrast, methods that aim at upholding privacy should be robust to most manipulations. Moreover, the adversarial examples developed in this work are targeted at a single classifier. At the same time, a method that is intended to be used for privacy should not be dependent on the downstream classifier. These considerations thus led us to develop the work presented in the next chapter.

7

Privacy-oriented Manipulation of Speaker Embeddings

Speaker embeddings are ubiquitous, with applications ranging from speaker recognition and diarization to speech synthesis and voice anonymization. The amount of information held by these embeddings lends them versatility but also raises privacy concerns. Speaker embeddings have been shown to contain sensitive information, including the speaker's age, sex, health state and more – in other words, information that speakers may want to keep private, especially when it is not required for the target task. In this work, we propose a method for removing and manipulating private attribute information in speaker representations that leverages a Vector-Quantized Variational Autoencoder architecture combined with an adversarial classifier and a novel mutual information loss. We validate our model on two attributes, sex and age, and perform experiments to remove or manipulate this information using ignorant and informed attackers. The model is tested with in-domain and out-of-domain data to assess its robustness, and the resulting speaker representations are used in a speaker verification scenario to validate their utility. Our results show that our model obtains a strong trade-off between utility and privacy, achieving age and sex classification results near chance level for both attackers and yielding little impact on speaker verification performance.

7.1 Introduction

Speaker representations, or embeddings – vector representations that model speakers' voices – are a key component in speech technologies. Originally developed for speaker recognition [41, 75, 292], i.e., the task of identifying or verifying the identity of a speaker, speaker embeddings are applied to a multitude of tasks that extend far beyond their original purpose, ASV, as considered in Chapter 4. Applications of modern neural speaker embeddings [78, 365] – latent representations taken from intermediate layers of neural networks trained to classify large sets of speakers – range from speaker diarization [166] – as seen in Chapter 5 – to text-to-speech synthesis [62], voice anonymization [318], and even detection of speech-affecting diseases [243].

This versatility is a testament to the wealth of information that is encoded by neural speaker embeddings, including (i) linguistic information [260, 264]; (ii) paralinguistic information [168], i.e., non-linguistic, but communicative information, such as affective, attitudinal and emotional information [135, 234]; and (iii) extra-linguistic information [168], i.e. non-communicative information about the speaker that is carried by the speech signal, such as the speaker's age and sex [163], accent [264], as well as the speaker's health state (i.e., the presence of speech-affecting diseases such as Parkinson's disease or Obstructive Sleep Apnea, among others) [199, 243]. However, whereas this information renders speaker representations particularly useful, it also raises questions of privacy and even adherence to data protection regulations when speaker representations are processed remotely, outside users' devices.

The work presented in Chapters 3, 4 and 5, along with others in the literature [208, 323, 339], have

shown how increasingly complex systems can be implemented with cryptographic techniques. However, the computational and communication costs of the resulting methods are still high and are limited by the efficiency of the underlying cryptographic constructions. Moreover, the computational performance of these methods depends on the complexity of the target task, making them difficult to apply to state-of-the-art systems that leverage machine learning models that require billions of operations.

Alternatively, we can consider privacy-oriented speech manipulation methods. Instead of providing confidentiality during the computation, these methods are applied before the data is processed and aim to remove or sanitise information that is considered private and not relevant to the target task [8, 222, 318]. This allows for a conscious trade-off between the information that is disclosed and the information that should remain hidden, or in other words, a trade-off between privacy and utility. These solutions are also more user-centred, as the privatisation process may be applied directly in the users' device [8, 351].

Speech manipulation methods also go in line with the *data minimisation* principle mentioned in Article 25 of the GDPR and defined in Article 5 (cf. Appendix A, Section A.4) of the GDPR, whereby personal data should be "adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed" [88].

These methods have the advantage of being independent of the downstream task's complexity, though not necessarily of the task itself. This is an advantage over cryptographic protocols as it allows the downstream adoption of arbitrarily complex state-of-the-art methods. However, unlike cryptographic constructions, this family of methods does not provide any formal privacy guarantees. This means that the evaluation of these methods, which is usually done empirically, needs to be thorough and well-designed in order to support privacy claims correctly.

Privacy-oriented speech manipulation methods follow three main trends. The first is voice anonymisation [318], where the goal is to modify the speech signal to hide the identity of the true speaker but keep linguistic and paralinguistic content intact, such that the speech signal is considered anonymised under the GDPR, allowing its storage and use in the training of speech-based machine learning applications, or even in remote inference scenarios, where only linguistic or paralinguistic content are necessary for the task at hand. The second trend is privacy-oriented feature extraction [214, 337], where the goal is to obtain feature vectors from which all the information that is not related to the target task is removed and where particular focus is given to the removal of speaker-identity-related information. The third trend consists of attribute disentanglement, manipulation, or removal methods. This is a more fine-grained approach that aims to remove specific speaker traits that are considered sensitive from the speech signal or a representation thereof while keeping the remaining information intact [8, 222, 244].

In this work, we focus on the third trend and propose a method for attribute manipulation and removal

in speaker embeddings. As mentioned at the beginning of this section, neural speaker representations have a vast number of applications. Consequently, modifying these representations to promote privacy will indirectly lend a level of privacy to downstream applications. For instance, removing demographic attributes from speech (or speech representations) can potentially avoid negative biases or even discrimination on the part of the service provider. Moreover, as shown by [220, 244], privatised speaker representations can be used to perform voice anonymisation to a certain extent.

Notwithstanding other possible applications, the primary purpose of speaker embeddings is to perform ASV, the process of verifying an individual's identity through their voice – a process which is performed mainly in remote settings. Privatised representations that hide sensitive speaker attributes will directly prevent speaker verification vendors (remote servers) from inferring sensitive information, again providing a level of privacy to this task [60, 221, 222]. Given that ASV is the main application of speaker embeddings and that measuring ASV performance using privatised vectors provides an estimate of how much the original (non-private) content of the vectors was changed, we consider ASV as both our target task and measure of utility.

The contributions of this chapter can be summarised as follows:

- We propose a new method for the privacy-oriented removal and manipulation of age and sex information in speaker representations. To the best of our knowledge, this work is the first to consider the removal of age information from speaker representations.
- Our method is based on a combination of a Vector Quantised - Variational Autoencoder (VQ-VAE), an adversarial classifier and a novel mutual information loss.
- For each attribute we evaluate our method with to two competing aspects: privacy and utility.
  - Privacy is assessed using as a proxy the attribute classification performance of two types of attackers, an ignorant attacker and an informed attacker.
  - Utility is evaluated in terms of ASV performance.
  - We perform an ablation study to assess the privacy and utility contributions of each component of our method.
  - We evaluate the attribute manipulation performance of the proposed methods, to understand whether they are versatile enough to be applied in tasks that are not related to privacy.
- For the sex attribute:
  - We evaluate our method through its performance on out-of-domain data, to assess its transferability to new domains.

- Overall, our results show that the proposed mutual information loss improves both privacy and utility when combined with the adversarial classifier, with their combination being able to reach near chance-level classification for both attributes and types of attackers. The proposed model is also shown to transfer to new domains and to be able to successfully manipulate attribute information within the speaker representations.

The remainder of this chapter is organised as follows: Section 7.1.1 describes the relevant literature; in Section 7.2, we describe the problem at hand; Section 7.3 describes the proposed method and each of its components; Section 7.4 details the experiments that were conducted along with the corresponding datasets and parameters; in Section 7.5, we present and discuss our results, and in Section 7.6, we summarise this chapter, provide closing statements and propose topics for future work.

7.1.1 Related Work

Modifying or suppressing speaker attributes within the speech signal, or representations thereof, is a growing area of research. Several studies do so to ensure that classifiers are invariant with regard to certain traits [182, 243], or to create control mechanisms for speech synthesis and voice conversion algorithms [26]. In addition to this, and more relevant to the present work, privacy-related approaches have also seen a surge in recent years.

An early example of attribute suppression for privacy is the work by Aloufi et al. [8], where the authors apply a CycleGAN to convert emotional speech to neutral speech as a way to remove sensitive, emotional information from the speech signal. In [9, 10], the same authors proposed two methods to protect the privacy of speaker identity, emotional content, sex, and accent/language information. This is done to protect the user's privacy for ASR. The methods are based on encoder-decoder architectures, whose encoders comprise two branches, one encoding linguistic information and another encoding speaker or paralinguistic information. By selecting the branches that are fed to the decoder, the authors are able to select the information present in the output signal. In [10], the authors evaluate their model in terms of efficiency to assess their usability in the context of mobile computing.

Jaiswal et al. [126] develop a neural network for emotion classification using speech and text data. This network includes an adversarial classifier with a gradient reversal layer [103] that promotes the learning of latent representations that are invariant to sex, making them private with regard to this attribute. The authors show that their method has little impact on emotion classification performance while improving privacy protection, to varying degrees, with regard to sex information. The authors also study how their sex-invariant representations affect an attacker's ability to perform membership inference (i.e., classify whether a sample was seen or not during the model's training).

Ericsson et al. [85] proposed a model to remove sex information from speech and validate their model for spoken digit classification. Similarly to [9, 10], this method is based on an encoder-decoder network,

where the encoder acts as a filter to the sensitive attribute, and the decoder takes this sanitised representation and reconstructs the speech signal using a fake, externally provided attribute. To promote the removal of sex information, the filter is trained adversarially against the attribute classifier.

Stoidis and Cavallaro [304] focused on disentangling and manipulating sex and speaker identity from the speech signal for privacy using a VQ-VAE and evaluated the utility of their method with regard to ASR performance. Later, the same authors developed a method based on their prior work and the work of Ericsson et al. [85], with the goal of generating gender-ambiguous voices (i.e. voices that are not strongly related to any gender) for ASR [305].

Wu et al. [351] explore and compare multiple methods to remove sex and accent from speech, including pitch standardisation, a Variational Autoencoder, and a version of the same work combined with a Generative Adversarial Network for improved speech reconstruction quality. The latter was found to be the best-performing model for privacy protection.

Differently, Bemmel et al. [329] study the protection provided by adversarial examples created against sex classification neural networks. The authors show that combining a simple Support Vector Machine with knowledge-based features for sex classification is sufficient to overcome the adversarial perturbation and successfully classify sex. The authors also propose the use of different vocal adaptations (e.g. whispering, monotonality, high pitch) as protection against sex classifiers that use knowledge-based features.

Whereas the approaches above have focused on removing information from or hiding information contained in the speech signal itself, other works have instead focused on removing information from speaker representations or knowledge-based feature vectors.

In Noé et al. [222], this is done through the use of an autoencoder trained adversarially with regard to a sex classifier where similar to [85], the decoding part of the network is conditioned on an externally provided attribute.

Similarly, Ali et al. [6] propose the use of an autoencoder architecture with an adversarial branch, using a gradient reversal layer, so that the encoder learns to remove sex, language, and speaker information from a set of speech features while keeping the remaining content intact. This approach is then applied to remote emotion recognition.

In [221], the same authors of [222], propose the use of a Normalising Flow-based architecture that disentangles sex information and aggregates it in a single component in a latent representation of the speaker embedding. To remove sex information, the component in the latent representation is set to zero, and the vector is reconstructed. In the same paper, [221], the authors also argue that to assess how well an attribute is removed, attacker classifiers should be trained over protected representations.

Feng and Narayanan [94], in a similar line to that of [8], develop a model to transform the emotional content of a knowledge-base feature vector into a neutral emotion, in case the corresponding emotion is

deemed sensitive (e.g. anger). The resulting transformed vector is then used to infer non-sensitive emotions (e.g. sadness). An adversarial classifier is further added to remove sex information from the feature vector. Later, within the same emotion recognition context, Feng et al. [93] used a multi-objective mutual information-based feature selection approach, to select the set of features that were most relevant for emotion classification and least informative regarding speaker sex. This approach also included the addition of Gaussian noise tailored to the masking of sex information, in addition to an adversarial classifier that was added to remove sex information from the resulting features.

Similar to [221, 222], Perero-Codosero et al. [244] propose the use of an adversarial autoencoder, based on their prior work [243], to remove speaker identity, sex and accent information from speaker representations. To remove each of these, an adversarial classifier with a gradient reversal layer is added and applied over the latent representations of the autoencoder. The privatised speaker representations are subsequently used as part of a voice anonymisation framework.

Recently, Chouchane et al. [60], basing their approach on the work of Noé et al. [222], proposed a method where differentially private noise is added to an autoencoder's latent representation, to remove sex information from a speaker representation. The authors show that, by controlling the level of noise, they are able to achieve different trade-offs between privacy and utility (i.e. speaker verification performance).

As mentioned in Section 7.1, one of the main trends of privacy-oriented speech manipulation is privacy-aware feature extraction. The main goal in this research line is to remove all of the information that is not necessary to the target task, while simultaneously optimising the representation for the target task. Although this goal differs from ours, it is worth mentioning some works related to this trend, as they share many of the methods used for attribute suppression.

For instance, Nelus and Martin [213] proposed an adversarial training architecture to remove speaker information from a feature representation used to classify speaker sex. In a later work [214], the same authors apply the concept of a variational information bottleneck and minimise the mutual information between the input and output representations of a neural network trained for sex classification. This is done to minimise the amount of information contained by the feature representation that is not relevant to the target task. It is then shown that this reduces the amount of information related to speaker identity. Building on their two prior works, in [216], Nelus and Martin train a neural network for sex classification using a Siamese architecture trained with a contrastive loss, to bring feature vectors that belong to speakers from the same sex closer together, and vice-versa. The authors show that the latter approach obtains improved results both in terms of utility and speaker privacy when compared to the two previous works.

Similarly, the work of Wang et al. [341] focuses on the removal of all target-task irrelevant information, as opposed to the removal of selected attributes. To this end, the authors leverage a CycleGAN

"obfuscator", trained to minimise a target task loss (e.g. sex or speaker classification), while simultaneously being trained adversarially against a "deobfuscator" that attempts to reconstruct the true signal from the obfuscated signal. This combination is then expected to elicit the model to remove all information that is unnecessary to the target task.

The works of Ravi et al. [266, 267] and Wang et al. [337] focus on the development of privacy-aware feature extraction methods for the classification of depression, while removing all non-depression-related speaker information, using adversarial training. Whereas Ravi et al. [266] focus solely on adversarial training, in [267] the authors expand their previous work, testing several models and different adversarial loss functions. Although the three works leverage a GRL, Wang et al. [337] propose a variation of the work of [266] by assigning different gradient weights to different layers, which is shown to improve the trade-off between target task performance and privacy.

Though not related to privacy, the works of Janbakhshi and Kodrasi [127], Mun et al. [205], and Li et al. [173] are also worth mentioning, due to their use of mutual information-based losses for information disentanglement. Specifically, Janbakhshi and Kodrasi [127] propose a method for the detection of dysarthric speech that aims to be invariant with respect to speaker information. To this end, the authors use an AE architecture, trained to reconstruct the input signal, using two branches, one to encode task-related information, and a second to encode speaker information. Both encoders are trained to classify the information they are meant to encode. To promote information independence between the two branches, the authors add a mutual information minimisation loss which is based on the CLUB mutual information upper bound [53]. Similar approaches have been used by Mun et al. [205] and Li et al. [173] to disentangle speaker information and domain conditions for improved domain generalisation in speaker recognition tasks.

It is also important to note that there are template protection mechanisms that can perform privacy-preserving enrolment and authentication in ASV, concealing all of the user's information [133, 202, 203]. These mechanisms correspond to transformations of the input, such that the original values cannot be recovered from the transformed ones. This makes these schemes secure, as any party can hold the transformed vector without being able to learn any information about it. Moreover, vectors transformed in the same way (i.e., using the same secret key) can be meaningfully compared. Although such schemes are important to biometric verification, they are not directly applicable to tasks other than verification, retrieval or clustering. In contrast, the method developed in this work extends to any downstream task, even though it does not provide confidentiality.

7.2 Formal problem definition

As mentioned in Section 7.1, in this work, we consider a remote Automatic Speaker Verification scenario, where a user wants to be able to authenticate through a remote ASV service provider (or

vendor). To do so, the user first needs to enrol into the system by sending a speaker embedding to be used as a template. Later, for authentication, the same user generates a new embedding of their voice and sends it to the vendor so that the vendor can compare it to the stored template.

In this scenario, we assume that the speaker representation is extracted on the user's device, whereas verification is performed remotely. We also assume that the user does not fully trust the service provider with their information and wants to hide sensitive attributes contained in the speaker representations, such that the service provider or any other entity that is able to obtain the user's speaker representation (e.g., via a data breach, or directly shared by the ASV vendor), is not able to infer the sensitive information from it

ASV was chosen as our target task as it represents a simple setting where we can test the utility and privacy of the transformed speaker representations.

The scenario described above can be simplified as an adversarial game, where we have a user trying to protect sensitive attribute information about themselves and an attacker who wants to obtain this information. As such, we want to develop a method of hiding a sensitive attribute from a speaker representation so that an attacker cannot obtain this attribute just by observing the transformed representation. This method should be applied in the user's device after the speaker representation has been extracted.

For a given input speaker embedding x with private attribute y_a, *discrete or continuous*, coming from a dataset $\mathcal{D}$, our goal is to learn a function F_a that removes attribute information y_a. Moreover, for versatility, we want our method to not only remove attribute information but also to be able to manipulate it. As such, we want to develop a function F_a that removes y_a and replaces it with external information $\hat{y_a}$:

$$\hat{x} = F_a(x|\hat{y_a}) \tag{7.1}$$

To ensure the attacker is not able to learn anything about the attribute, we should select $\hat{y_a}$ such that it provides the least amount of information – e.g., using the expected value of y_a. Nevertheless, defining our model as dependent on the conditioning of the decoder allows us to choose the best strategy to undermine a possible attacker.

To ensure utility, we also want F_a to guarantee the same discriminability shown by the original vectors. In other words, transformed vectors that belong to different speakers should be far apart, whereas those that belong to the same speaker should be as close as possible. To measure this, we can compute the distance of the same- and different-speaker pairs of vectors after transformation and measure how discriminative this distance is with regard to speaker identity.

To measure the level of privacy provided by F_a, we need to assess how well an attacker can recover the original attribute y_a. However, an attacker can take different forms. Here, we consider two types of

attackers with different levels of knowledge about the protection mechanism: an *ignorant attacker* and an *informed attacker*.

We assume that the weakest possible attacker, the *ignorant attacker*, will try to infer the original attribute directly, having no knowledge of the privatisation mechanism. We assume that an *ignorant attacker*, will hold an attribute classifier $C_{\mathcal{A}}$, trained on a dataset $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), ...(x_n, y_n)\}$ of non-transformed data, with probability $\mathbb{P}(C_{\mathcal{A}}(x) = y_a)$ as close to 1 as possible.

In the case of classification, to guarantee privacy with regard to y_a, the following should hold for any pair (x, y_a):

$$\mathbb{P}(C_{\mathcal{A}}(F_a(x|\hat{y}_a)) = y_a) = \frac{1}{n_a}, \tag{7.2}$$

with n_a as the number of classes of attribute a.

To encompass the possibility of F_a allowing the manipulation of the attribute y_a within the speaker embedding, we also want that $\mathbb{P}(C_{\mathcal{A}}(\hat{x}) = \hat{y_a})$ be as high as possible. This means that an attacker holding any classifier trained on non-transformed data should not be able to obtain any information about attribute y_a by observing $\hat{x}$ unless the fake attribute $\hat{y}_a$ is the same as the true attribute y_a:

$$C_{\mathcal{A}}(F_a(x|\hat{y}_a)) = y_a \leftrightarrow y_a = \hat{y}_a. \tag{7.3}$$

Still, to ensure that the information is fully protected, we need to account for the possibility of an attacker being aware of the transformation that was applied to the speaker representation. As such, we consider as a stronger attacker, the *informed attacker*. This attacker not only knows that a privacy transformation was put in place but is also able to apply this transformation to its data, for which the true labels are known, and train a classifier using the privatised representations. In a way, this attacker will develop a classifier to try to infer the sensitive attribute, using the residual information that is still encoded by the privatised representations. We assume that this attacker will hold an attribute classifier $\hat{C}_{\mathcal{A}}$, trained on a dataset $\hat{\mathcal{D}} = \{(\hat{x}_1, y_1), (\hat{x}_2, y_2), ...(\hat{x}_n, y_n)\}$ of data transformed as $\hat{x} = F_a(x|\hat{y}_a)$. In this situation, our goal is that the attribute classifier trained by the *informed attacker* is not able to generalise beyond the training data such that, for unseen data, $\mathbb{P}(\hat{C}_{\mathcal{A}}(\hat{x}) = y_a) = \frac{1}{n_a}$.

To summarise the above, the goal of this work is to develop a method that achieves the following under the two attack scenarios:

- Allows the suppression of attribute information from speaker representations and enforces privacy with regard to this subset of information (cf. eq. 7.2);
- It not only removes attribute information but manipulates it within the speaker embedding (cf. eq. 7.3);
- Keeps the utility of the transformed vectors for speaker verification.

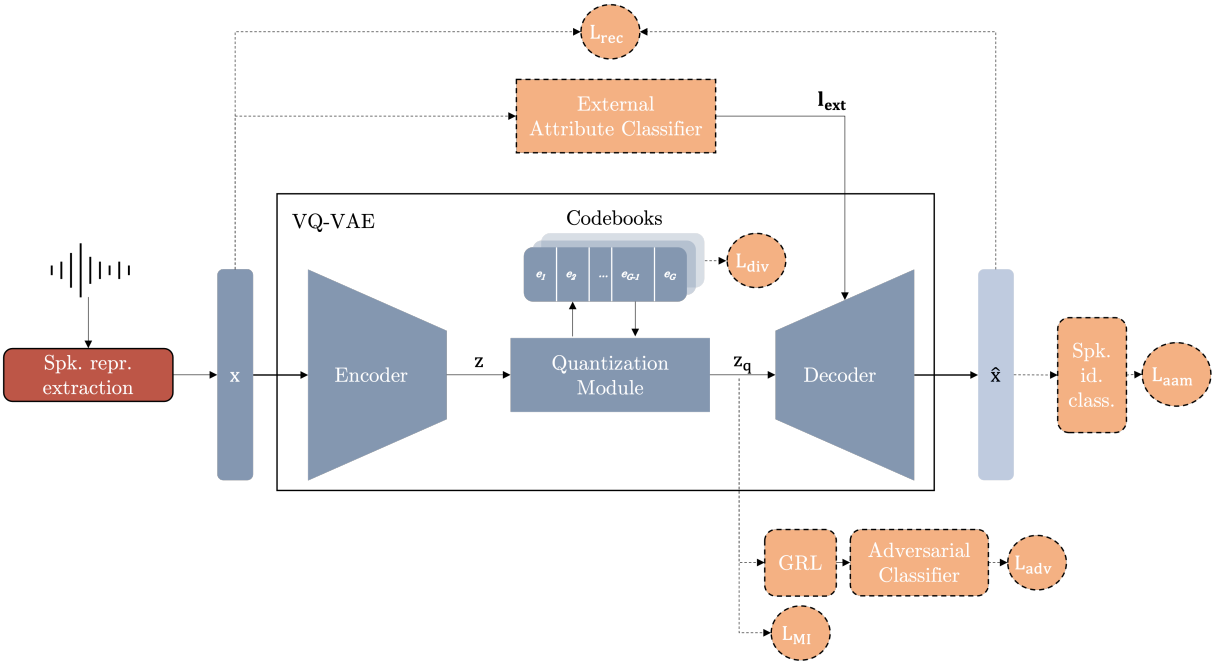


Figure 7.1: Block diagram of the proposed method. Dashed boxes and lines represent components that are only necessary during training and that are dropped at inference time.

7.3 Method

To achieve the objectives summarised in the previous section, we propose a combination of five components: a Vector-Quantized Variational AutoEncoder (VQ-VAE); an external speaker identification classifier; an external attribute classifier C_{ext}; an adversarial attribute classifier, C_{adv}; and a Mutual Information (MI) loss L_{MI}. In the remainder of this section, we will detail each of these components and their role in removing information from speaker representations.

7.3.1 Vector-Quantized Variational Autoencoder

The main basis of our method is a Vector Quantised - Variational Autoencoder (VQ-VAE). VQ-VAEs have been shown to perform well for several speech tasks [16, 17, 330], revealing a solid capability for information disentanglement [58, 330, 349]. In this section, we briefly introduce the concept of VQ-VAEs and detail the importance of this model in our overall method.

VAEs [147] are a family of generative models that have been widely used for synthetic data generation, representation learning and disentanglement. VAEs models follow a general autoencoder architecture, being composed of an *encoder* and a *decoder*. Specifically, the encoder creates a latent representation from the input, while the decoder uses this representation to reconstruct the input. During training, the encoder learns to map the input to the parameters of a prior distribution – usually, a normal distribution parameterised by a mean vector and a covariance matrix – while the decoder learns to

reconstruct the input by sampling from this distribution. This, together with its specific loss function, regularises the latent space, imposing a structure on the model's latent representations. This property makes it possible to use the decoder as a generator by sampling from the latent space. In addition, the structured latent space will be composed of independent, or disentangled, factors, allowing for an easier manipulation of the input signal when represented in this form.

To address these issues, van den Oord et al. [330] proposed a vector quantised version of VAEs (VQ-VAE). In this version, instead of being modelled by a continuous prior distribution, the latent space is modelled by a learnable set of discrete codes. To perform inference, this set of codes, the *codebook*, is indexed by the output of the encoder, which selects the sub-set of codes that best models the input. The decoder then takes this sub-set of codes and reconstructs the input.

This poses several advantages over the original VAE, namely avoiding the problem of posterior collapse, by having a function of the input select the codes that best model it, and improves reconstruction quality, by the fact that the latent space is no longer static, being trainable, and thus more adjusted to the training data distribution. Moreover, the discrete nature of the codebook also helps in the disentanglement of information, as each entry in the codebook will correspond to an aspect of the input signal.

When considering our target task, the removal and manipulation of information within a speaker representation for privacy, VQ-VAEs appear as an attractive solution. This comes from the fact that all of the information that is necessary to reconstruct the input signal is obtained from the quantization module and that this information is inherently disentangled, making it easier to manipulate or remove.

Formally, a VQ-VAE is defined as follows [330]: assume we have an encoder $E : \mathbb{R}^n \to \mathbb{R}^h$, a decoder $D : \mathbb{R}^f \to \mathbb{R}^n$ and a quantization module $Q : \mathbb{R}^h \to \mathbb{R}^q$. For an input vector (in our case a speaker embedding) $x \in \mathbb{R}^n$, we start by feeding it through the encoder E to obtain a latent representation $\mathbf{z} \in \mathbb{R}^h$; this vector is passed through the quantization module, where we obtain the quantized representation $\mathbf{z}_q \in \mathbb{R}^q$; $\mathbf{z}_q$ is in turn fed to decoder D, such that the original input is reconstructed.

Our setting differs from a regular VQ-VAE because we want the output to differ from the input. However, we do not have access to embeddings of the same speaker presenting different versions of each attribute. As such, to be able to train the VQ-VAE and promote attribute disentanglement, we turn to the solution of Noé et al. [222] and condition the decoder with the output of an external pre-trained attribute classifier. Specifically, we take the output logits l_{ext} of an external classifier $C_{ext} : \mathbb{R}^n \to \mathbb{R}^{c_{\text{attr}}}$ – where c_{attr} corresponds to the number of classes[1] – obtained for the original input, to which we apply a linear transformation $h_{attr} : \mathbb{R}^{c_{\text{attr}}} \to \mathbb{R}^w$ and concatenate this representation with the output of the quantization module, $\mathbf{z}_q$, obtaining:

$$\hat{\mathbf{z}}_q = [\mathbf{z}_q \mid h_{attr}(l_{ext})], \tag{7.4}$$

[1] $c_{\text{attr}} = 1$ for regression tasks.

where | represents the concatenation operator; $\hat{\mathbf{z}}_q$ is then feed as input to the decoder D.
This enables the VQ-VAE to reconstruct the original input signal during training while also allowing us to manipulate the attribute information at test time by changing the values used to condition the decoder. Moreover, it also provides an implicit level of disentanglement, as the decoder will not require as much information about the attribute from the latent representation, since it has direct access to it from the conditioning logits.

7.3.1.A Quantization Module

Our implementation of the quantization module of the VQ-VAE corresponds to the product quantization approach of Baevski et al. [17, 131]. In [17], the quantization module is defined as a tensor $Q \in \mathbb{R}^{G \times V \times e/G}$, with G being the number of codebooks, and V the number of codewords $v \in \mathbb{R}^{e/G}$ within each codebook. To quantize a latent vector $\mathbf{z} = E(x)$, we select an entry v from the V entries of each codebook G to obtain a set of codewords $v_1, ..., v_G$. To this end, first, a linear transformation is applied $\mathbb{R}^h \rightarrow \mathbb{R}^{G*V}$, to obtain $\hat{\mathbf{z}} \in \mathbb{R}^{G*V}$, after which $\hat{\mathbf{z}}$ is reshaped to $\mathbb{R}^{G \times V}$, giving us G logit vectors $l_g \in \mathbb{R}^V$ (one logit per codeword per codebook). To choose entries v at inference time, the largest index i of each l_g is selected. During training, to ensure the selection is fully differentiable, a straight-through estimator of the Gumbel-Softmax is used [16, 17, 128]:

$$p_{g,v} = \frac{\exp(l_{g,v} + \eta_v)/\tau}{\sum_{k=1}^{V} \exp(l_{g,k} + \eta_k)/\tau}, \tag{7.5}$$

where each $p_{g,v}$ corresponds to the probability of selecting entry v of codebook g; $\eta_v = -\log(-\log(u_v))$, with u_v uniformly sampled from $\mathcal{U}(0, 1)$; and τ is a non-negative temperature. During the forward pass, the codeword is selected by index $i = \text{argmax}_j p_{g,j}$, whereas in the backward pass, the true gradient of eq. 7.5 is used. After $v_1, ..., v_G$ have been selected, a final linear transformation is applied, $\mathbb{R}^e \rightarrow \mathbb{R}^q$, to obtain $\mathbf{z}_q \in \mathbb{R}^q$.

7.3.1.B Training losses

The VQ-VAE is trained with several losses. The first loss we consider is the reconstruction Mean Squared Error loss, or L_{rec}, defined as:

$$L_{\text{rec}} = \|x - F(x|l_{ext})\|_2^2, \tag{7.6}$$

with $F(\cdot)$ corresponding to the VQ-VAE, and l_{ext} corresponding to output logits of the external attribute classifier C_{ext} with regard to input x, that are used to condition the decoder of F.
To encourage a more diverse selection of codewords,and to prevent codebook collapse (i.e., a state where only a subset of codewords are ever selected for any input), we also add a *codebook diversity* loss,

L_{div}, as proposed by [17, 80]:

$$L_{\text{div}} = \frac{1}{GV} \sum_{g=1}^{G} \sum_{v=1}^{V} \overline{p}_{g,v} \log \overline{p}_{g,v}, \tag{7.7}$$

with V corresponding to the number of entries per codebook, and G corresponding to the number of codebooks in the quantisation module; $\overline{p}_{g,v}$ corresponds to the per-batch average of probabilities $p_{g,v}$, defined in eq. 7.5.

Finally, to promote target-task performance, we train the VQ-VAE for speaker identification, using a pre-trained, frozen, speaker classification layer combined with an Additive Angular Margin loss [77], L_{aam}, defined as:

$$L_{\text{aam}} = \frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{\zeta \ cos(\theta_{y_i,i}+a)}}{\mathcal{Z}}, \tag{7.8}$$

where $\mathcal{Z}$ is defined as:

$$\mathcal{Z} = e^{\zeta \ cos(\theta_{y_i,i}+a)} + \Sigma_{j=1,j\neq i}^{c_{spk}} e^{\zeta \ cos(\theta_j,i)}, \tag{7.9}$$

and where N is the number of samples in the batch; c_{spk} is the number of speaker classes; a is the angular margin; ζ is a scale factor; θ_y is the output of the speaker classification layer for a sample x_i.

The full VQ-VAE loss is then defined as:

$$L_{\text{VQ-VAE}} = \alpha L_{\text{rec}} + \beta L_{\text{div}} + \gamma L_{\text{aam}}, \tag{7.10}$$

where α, β and γ are weights for each of the loss functions. This system is represented in Fig. 7.1, corresponding to the blue boxes. Dashed blocks correspond to components of the method that are removed at inference time.

Even though the current method, as it stands, may already have some ability to disentangle information, it does not yet explicitly promote the removal of private information. In the following sections, we detail the two approaches we use to achieve this goal: an adversarial classifier and a mutual information minimisation loss.

7.3.2 Adversarial Classifier

Following what was stated above, to promote the explicit removal of the sensitive attributes, we consider adding an adversarial classifier C_{adv} [103, 112].

The goal of this adversarial classifier is to predict the sensitive attribute from a latent representation of the VQ-VAE. If it can predict the attribute, then it means that the model is not removing this information. We want to incorporate this information when training the VQ-VAE to improve its removal ability. To this end, we train the adversarial classifier and the VQ-VAE in tandem, wherein the

former will try to obtain information about the protected attribute, and the latter will try to provide as little information about it as possible. This can be seen as a minmax game, where the VQ-VAE is trying to minimise its target loss and maximise the loss of the adversarial classifier, and the adversarial classifier is trying to minimise its own loss.

Concretely, the adversarial classifier is trained to predict the attribute from the latent representation $\mathbf{z}_q$, whereas the VQ-VAE will be trained to prevent C_{adv} from being able to correctly predict the attribute from this latent representation. To do so, we use a gradient reversal layer (GRL) [103], such that C_{adv} is optimised jointly with the VQ-VAE, but where the gradient corresponding to its loss is multiplied by a negative constant before being backpropagated through to the VQ-VAE. This means that the weights of C_{adv} will be adjusted to better predict the attribute, whereas the negated gradient that is passed to the VQ-VAE will adjust the weights such that it is more difficult for C_{adv} to predict the attribute, and, therefore, this attribute will be hidden or absent in the latent representation of the model.

Since the attribute information will be externally fed to the decoder, adding the adversarial classifier will compel the network to learn attribute-invariant codebooks, forcing the VQ-VAE to use the external information that is fed to the decoder.

For discrete attributes, the adversarial classifier is trained using the cross-entropy loss:

$$L_{adv} = -\frac{1}{c_{\text{attr}}} \sum_{i=1}^{c_{\text{attr}}} y_{attr_i} \log(p_i), \tag{7.11}$$

where c_{attr} corresponds to the number of adversarial classes, y_{attr_i} to the attribute label, and p_i, the output soft-probability for class i of the adversarial classifier obtained for the latent representation yielded by the quantization module, $\mathbf{z}_q$.

For continuous attributes, the MSE loss is used instead:

$$L_{adv} = \|y_{attr} - C_{adv}(\mathbf{z}_q)\|_2^2 . \tag{7.12}$$

The gradient reversal layer, adversarial classifier and adversarial loss are represented by dashed boxes in Fig. 7.1.

7.3.3 Mutual Information Loss

Adversarial networks have been shown to create seemingly invariant representations during adversarial training. However, these have been shown to fail to generalise to unseen data and new classifiers trained over the new adversarial representations [83, 221, 300]. There are several possible reasons for this to happen. For instance, during training, the adversarial classifier may no longer be able to infer the protected attribute, whereas the main network performs well for the target task. This may seem to indicate that the goal of removing the attribute was achieved. However, the adversarial network may

lack the capacity (i.e., may be too simple or have too few parameters) to infer the attribute from an "obfuscated" latent representation, where the attribute information is hidden, thus achieving the training loss objectives without being able to actually remove information. On the other hand, one may also see adversarial training as a way of inadvertently creating *adversarial examples*, i.e., data points that have suffered minute changes, but that can change a neural network's predictions [113].

For these reasons, in this work, we explore the usage of non-parametric nearest-neighbour-based mutual information (MI) estimators [105, 154, 275] as companion losses to the adversarial network. The goal of these losses is to minimise the amount of information shared between the output of the quantization module $\mathbf{z}_q$ and the target attribute label y. We hypothesise that, given their non-parametric nature, these losses should promote the learning of representations that are invariant to the target attribute and not simply representations that are able to "fool" the adversarial classifier.

To this end, we leverage two mutual information estimators: (1) the mutual information estimator proposed by B. Ross [275] for mixtures of discrete and continuous random variables and (2) the Kraskov, Stögbauer and Grassberger (KSG) [105, 154] estimator to estimate the mutual information between two continuous random variables.

The first estimator will be used as the loss between the latent representation $\mathbf{z}_q$ and a discrete attribute label y, which, in this work, corresponds to sex information. The second estimator will be used to compute the mutual information loss between $\mathbf{z}_q$ and y, in the case where y is a continuous attribute (e.g., age). In this section, we present only a high-level overview of these estimators. For further details we direct the reader to Appendix C, and to [105, 153, 154, 275].

7.3.4 Mutual information estimator for discrete and continuous random variables

The mutual information $I(Z, Y)$ between two variables Z and Y can be expressed in terms of the individual differential entropies and the entropy between the two random variables:

$$I(Z, Y) = H(Z) + H(Y) - H(Z, Y). \tag{7.13}$$

Given a set of N observations taken from dataset $\mathcal{B}$ of the joint variable $M = (Z, Y)$, $m_i = (z_i, y_i)$, with $i \in 1 \dots N$,

the goal of a mutual information estimator is to use these observations to obtain $I(Z, Y)$.

The continuous-discrete mutual information estimator proposed by Ross [275] shows that for a discrete variable Y and a continuous variable Z, the mutual information estimator can be obtained through a

combination of nearest-neighbour entropy estimators [153], such that:

$$\hat{I}(z_i, y_i) = \psi(N) + \psi(k) - \psi(N_{y_i}) - \psi(n_{z_i}), \tag{7.14}$$

where $I(z_i, y_i)$ is the mutual information for a single observation (z_i, y_i); ψ corresponds to the *digamma* function [2]; k is a pre-specified number of neighbours; N_{y_i} corresponds to number of samples in $\mathcal{B}$ with the same discrete value y_i; and n_{z_i} is the number of samples between the continuous observation z_i and its k^{th} neighbour, sharing the same value y_i, computed using the euclidean distance.

To obtain the mutual information for the full set of samples, we compute the average of all $I_i(z_i, y_i)$:

$$\hat{I}(X, Y) = \psi(N) + \psi(k) - \langle \psi(N_y) \rangle - \langle \psi(n_z) \rangle, \tag{7.15}$$

where $\langle ... \rangle = \frac{1}{N} \sum_{i=1}^{N} ...$ is the average operator.

In summary, to compute the mutual information $I(z_i, y_i)$ between a vector z_i and its discrete label y_i, we need to find z_i's k^{th} neighbour in a set B, sharing the same discrete variable. We then count the number of vectors z (n_{z_i}) in $\mathcal{B}$, that correspond to all other discrete variables $Y \neq y_i$, that are within the distance between z_i and its k^{th} nearest neighbour, and the total number of observations n_{y_i} with discrete value $Y = y_i$.

For a high-level intuition of this estimator, consider the following. From equation (7.14) we can see that the MI between a vector X (i.e., a speaker representation) and its discrete counterpart, Y (i.e., a class label) will be lower if n_z is high, and vice-versa. Note that n_z is the number of samples that are not from the same class as X, but which are closer to X than X is to its k^{th} nearest-neighbour belonging to the same class. Taking this into account, the MI can be seen as a measure of how well the speaker representations from each class are separated in space. If the MI is high, the vectors of each class are well separated from the other classes, and if the MI is low, then the vectors belonging to different classes will be intermixed. Thus, using the MI as a loss will prompt the VQ-VAE to learn to create latent representations that are closer together in space independently of their attribute classes, and that do not provide discriminative information concerning their attribute classes Y.

This estimator is presented in pseudo-code in Algorithm 2.

7.3.4.A Mutual information estimator for continuous random variables

For the second mutual information loss, between a continuous vector and a continuous attribute, we consider the use of a variant of the Kraskov, Stögbauer and Grassberger mutual information estimator [154] (Algorithm 2), proposed by Gao et al. [105], where the mutual information is estimated through:

Algorithm 2 Pseudo-code to compute $\hat{\text{I}}$(Z,Y) using eq. 7.15

Input: batch $\mathcal{B}=(Z,Y)$ of size N, neighbours k, pairwise euclidean distance matrix (edm) function $\text{pdist}_{l2}(\cdot)$, bottom_k$(\cdot)$ to obtain the k^{th} lowest value, row-wise.
$\text{edm}_Z \leftarrow \text{pdist}_{l2}(Z)$
$N_y \leftarrow [\,]$, k_dists $\leftarrow [\,]$
for $y \in \{Y\}$ **do**
  $N_y[y] \leftarrow \#\mathcal{B}_{Z|Y=y}$
  k_dists$[Y=y] \leftarrow$ bottom_k$(\text{edm}_{Z|Y=y})$
end for
$n_z \leftarrow [\,]$
for $i \in N$ **do**
  $n_z[i] \leftarrow 0$
  for $j \in N$ **do**
    $n_z[i]$ += 1 **if** $\text{edm}_z[i,j] \leq$ k_dists$[i]$
  end for
end for
mi $\leftarrow \psi(N) + \psi(k) - \langle\psi(N_y)\rangle - \langle\psi(n_z)\rangle$
return mi

$$\begin{aligned}\hat{I}(Z,Y) = \log(N) + \psi(k) + \log\frac{v_z v_y}{v_z + v_y} \\ - \langle\log(n_z) + \log(n_y)\rangle.\end{aligned} \tag{7.16}$$

Here, n_z and n_y correspond to the number of points between observation $m_i = (z_i, y_i)$ and its k^{th} neighbour in each marginal space (Z or Y), being defined as the k^{th} observation that is closest to the joint observation m_i, obtained using the euclidean distance. The values v_z and v_y correspond to the volumes of the d_z and d_y-dimensional unit-ball, for the marginal spaces z and y, being defined as $v = \pi^{\frac{d}{2}}/\Gamma(\frac{d}{2}+1)$, with Γ the *gamma* function [2].

In other words, for each pair (z_i, y_i) in $\mathcal{D}$, we count the number of points (n_z and n_y) for each random variable, that are within distances ϵ_{z_j} and ϵ_{y_j}, which correspond to the distances in each marginal space between the joint observation m_i and its k_{th} neighbour. As before, this estimator is described in pseudo-code in Algorithm 3.

7.3.4.B Differentiability of the estimators

To turn $\hat{I}(Z,Y)$ into a loss, we need to ensure that all steps in its computation are differentiable. Determining the k^{th} closest neighbour and counting the number of data points inside a given radius are not differentiable operations.

For simplicity, we assume that in the top-k operation (to determine the k^{th} closest neighbour), gradients are only passed through to the top-k elements. In contrast, for other elements, gradients are set to zero. On the other hand, the less or equal than comparison is implemented using a

Algorithm 3 Pseudo-code to compute $\hat{\mathrm{I}}$(Z,Y) using eq. 7.16

Input: batch $\mathcal{B}=(Z,Y)$ of size N, neighbours k, pairwise euclidean distance matrix (edm) function $\mathrm{pdist}_{l2}(\cdot)$, bottom_k_idx$(\cdot)$ to obtain the row-wise index of the k^{th} lowest value.
$v_z \leftarrow \pi^{\frac{d_Z}{2}}/\Gamma(\frac{d_Z}{2}+1)$, $v_y \leftarrow \pi^{\frac{d_Y}{2}}/\Gamma(\frac{d_Y}{2}+1)$
$\mathrm{edm}_Z \leftarrow \mathrm{pdist}_{l2}(Z)$
$\mathrm{edm}_Y \leftarrow \mathrm{pdist}_{l2}(Y)$
$\mathrm{edm}_{ZY} \leftarrow \mathrm{pdist}_{l2}((Z,Y))$
k_dists_idx $\leftarrow$ bottom_k_idx(edm_{ZY})
$n_x \leftarrow [\,]$, $n_y \leftarrow [\,]$
for $i \in N$ **do**
  $n_z[i] \leftarrow 0$, $n_y[i] \leftarrow 0$
  for $j \in N$ **do**
    $n_z[i]$ += 1 **if** $\mathrm{edm}_Z[i,j] \leq \mathrm{edm}_Z[i,k_dists_idx[i]]$
    $n_y[i]$ += 1 **if** $\mathrm{edm}_Y[i,j] \leq \mathrm{edm}_Y[i,k_dists_idx[i]]$
  end for
end for
mi $\leftarrow \log N + \psi(k) + \log \frac{v_z v_y}{v_z+v_y} - \langle \log n_z + \log n_y \rangle$
return mi

straight-through estimator of the Heaviside function:

$$(d_i \leq d_{\mathrm{kth}}) = \mathrm{STHeaviside}(d_{\mathrm{kth}} - d_i). \tag{7.17}$$

These two adaptations allow us to use $L_{\mathrm{MI}} = I(Z,Y)$ in combination with our model. We positioned the loss in the same place as the adversarial classifier at the output of the quantization module.

The mutual information loss is represented at the bottom of Fig. 7.1 by a dashed circle, completing the method.

7.3.5 Full training loss

The simplest form of our model, the VQ-VAE by itself, uses as a training loss eq. 7.10.

To use the adversarial classifier and loss described above, we add L_{adv} to the training loss, multiplied by a weight δ. Similarly, to use the mutual information loss (cf. eqs. 7.15 and 7.16), we weight it with a constant value ϵ and add it to the remaining training losses, with the full loss becoming:

$$\begin{aligned} L_{\mathrm{Total}} &= L_{\text{VQ-VAE}} + +\delta L_{adv} + \epsilon L_{\mathrm{MI}} \\ &= \alpha L_{\mathrm{rec}} + \beta L_{\mathrm{div}} + \gamma L_{\mathrm{aam}} + \delta L_{adv} + \epsilon L_{\mathrm{MI}}. \end{aligned} \tag{7.18}$$

7.4 Experimental Setup

7.4.1 Experiments

As mentioned in the Introduction, two speaker attributes are considered, sex and age, which should be removed from speaker representations using the method described in the previous section. This is done with two different models, one for each attribute, each trained using the losses that are appropriate to discrete (i.e., sex) or continuous labels (i.e., age).

For the proposed method to be validated, it is required that we show that it fulfils the objectives detailed at the beginning of the Section 7.3: a) the method should be able to remove and manipulate attribute information, and b) the method should have little impact on the target task (speaker verification).

To validate both of these conditions, we conduct an extensive set of experiments:

1. An ablation study is conducted to compare the performance of a simple VQ-VAE with versions of the same VQ-VAE to which the adversarial loss L_{adv} or the mutual information loss, L_{MI}, were added, and finally, when both losses are used in combination. This study concerns both the sex and age attributes, and we report results in terms of privacy (i.e., the ability to remove the attribute) and utility (i.e., speaker verification performance).

2. The results that were obtained for the sex attribute are compared to the method of Noé et al. [221], the Normalising Flow zero Log-Likelihood Ratio (NFzLLR). This method was selected because it is a good representative of the state-of-the-art for attribute removal from speaker representations and because it is the work that has the closest evaluation methodology to our own.

3. We perform cross-domain experiments to understand how robust the proposed method is to domain changes. To do so, we use an out-of-domain dataset with which we replace (1) the test data, (2) the training data of the attribute classifier and (3) the training data of the VQ-VAE itself.

4. We test the manipulation capabilities of our method for both attributes. To this end, we treat the externally provided attribute information as the true labels and measure the performance of pre-trained (i.e., trained on unprotected data) sex and age classifiers in classifying the false information. This way, we are able to obtain an indication of whether the proposed method did indeed replace the true attribute with the fake one.

5. Due to a lack of age-labelled speech data sources, the cross-domain experiments are only applied to the sex information removal models.

All experiments are reported for both *ignorant* and *informed* attackers, with the exception of the attribute manipulation experiment, where we only consider the *ignorant* scenario.

7.4.2 Corpora

Four datasets are used in our experiments: VoxCeleb [206]; LibriTTS [362]; an age annotated partition of VoxCeleb named AgeVoxCeleb [309]; and a Portuguese version of the VoxCeleb corpus, VoxCelebPT [191], which contains annotations on both the speakers' sex and ages. Next, we describe each of these datasets and how they are used for the experiments described above.

7.4.2.A VoxCeleb

Table 7.1: Data partitions for the VoxCeleb and LibriTTS datasets.

Source dataset	Partition	#Speakers			#Utterances		
		Male	Female	Total	Male	Female	Total
VoxCeleb	train_vox_spk	4,347	2,858	7,205	1,459,045	887,649	2,346,694
	train_vox_vq	2,572	2,572	5,144	467,870	412,225	880,095
	train_vox_att	209	191	400	37,444	29,835	67,279
	test_vox_att	91	46	137	24,598	9,511	34,109
LibriTTS	train_libri_vq	600	560	1,160	100,364	104,680	205,044
	train_libri_att	474	430	904	55,619	60,881	116,500
	test_libri_att	164	162	326	20,274	23,536	43,810
Vox+Libri	train_vox_libri_vq	3173	3132	6305	209,286	200,623	409,909

VoxCeleb [206] is the primary source of data for the experiments presented in this work. As reported in previous chapters, this corpus includes recordings of 7,363 speakers of multiple ethnicities, accents, occupations, age groups and languages, having English as the most prevalent language. It is composed of short clips taken from interviews uploaded to YouTube. The corpus is composed of two parts, *VoxCeleb 1 and 2*, both subdivided into *dev* and *test*.

We use four data partitions, described in detail in Table 7.1, three of which are used for training the different components of our method, and the fourth is used for testing.

The first partition – *train_vox_spk* – corresponds to the data used to train the speaker embedding extraction model and corresponds to the full *dev* set of VoxCeleb (1+2), with 7,205 speakers.

The second partition – *train_vox_vq* – is used to train the VQ-VAE for the sex attribute. It uses a subset of 5,144 speakers (balanced by sex), taken from the *dev* set of VoxCeleb (1+2). This partition is also used to train the external sex classifier, from which we extract the logits used to condition the VQ-VAE's decoder.

The third partition – *train_vox_att* – is composed of a second set of 400 speakers, also taken from the *dev* set of VoxCeleb, having no speaker overlap with the partition used to train the VQ-VAE. This

Table 7.2: In-domain and cross-domain experiments.

Partition	Train VQ-VAE	Train C_{att}	Test C_{att}
Domain	VoxCeleb	VoxCeleb	VoxCeleb
			LibriTTS
		LibriTTS	VoxCeleb
			LibriTTS
	LibriTTS	VoxCeleb	VoxCeleb
			LibriTTS
		LibriTTS	VoxCeleb
			LibriTTS

partition is used to train the sex classifiers that evaluate the privacy capabilities of our method.

All sex attribute-related experiments are evaluated using a combination of the *test* sets of VoxCeleb 1 and 2 – *test_vox_att*. However, Nagrani et al. [206] warn that there may be a speaker overlap between the VoxCeleb 1 *dev* and *test* partitions with VoxCeleb 2 *test*. We manually checked the speakers in VoxCeleb 2 *test* and found 21 speakers that were present in VoxCeleb 1. These speakers were removed from the test set to avoid contamination from the training data. This resulted in a final set of 137 test speakers.

Speaker verification performance is evaluated using VoxCeleb 1's original trial pairs, taken from VoxCeleb 1's test partition, corresponding to a set of 40 speakers, 4,874 utterances and a total of 37,720 trials.

7.4.2.B LibriTTS

Our second main source of data is LibriTTS [362]. This dataset is an adaptation of the LibriSpeech corpus – a corpus of read speech, fully in English, taken from audiobooks – wherein the data was processed to be suitable for text-to-speech tasks. The complete LibriTTS corpus amounts to a total of 586.5 hours, containing 2,456 speakers.

In our cross-domain study for the sex attribute, we use this dataset to assess how well our model generalises to unseen domains. LibriTTS is comprised of read speech, recorded under controlled conditions, which makes it starkly different from VoxCeleb, where the speech recordings are noisy and contain spontaneous speech, making this dataset an ideal source of out-of-domain data. The motivation for this experiment comes partly from the fact that the VQ-VAE, the sex attribute classifier, and the speaker embedding extraction model are all trained on VoxCeleb, possibly giving us biased results.

For the above, to assess the impact of domain changes, we perform a total of 8 experiments using different combinations of VoxCeleb and LibriTTS. These include replacing the data used to train the

VQ-VAE, the data used to train the attribute classifier and the test data. These experiments, as well as the in-domain experiments, are summarised in Table 7.2, where each line corresponds to one experiment, and each column corresponds to the different tasks for which the data is used.

To perform these experiments, we use three LibriTTS partitions: *train_libri_vq*, *train_libri_att* and *test_libri_att*. The first is used to train the VQ-VAE, the second is used to train attribute classifiers, and the third is used as a test set. The *train_libri_vq* partition comprises data taken from LibriTTS' train-other-500 partition; *train_libri_att* uses data taken from train-clean-360 and, *test_libri_att* combines data taken from train-clean-100, dev-clean and test-clean. Each speaker is present only in a single partition.

Finally, we use *train_vox_libri_vq* to train the VQ-VAE, in one of the cross-domain scenarios, where 50% of the VQ-VAE's training partition is composed of data taken from LibriTTS, and 50% is taken from VoxCeleb. Specifically, the subset of LibriTTS data corresponds to *train_libri_vq*, and the subset of VoxCeleb corresponds to *train_vox_vq*, with the number of samples downsampled to match the size of *train_libri_vq*.

In-depth details for all partitions can be found in Table 7.1.

7.4.2.C AgeVoxceleb & VoxCelebPT

For our age-related experiments, we use two datasets: AgeVoxCeleb [309] and VoxCelebPT [191]. The full details of the partitions used in our experiments can be found in Table 7.3.

AgeVoxCeleb is a subset of VoxCeleb 2 that has been annotated with speaker age labels, obtained by cross-checking birth years found online, with video recording and broadcasting dates. This dataset is composed of 4,976 speakers and 21,707 utterances, with several speakers having multiple utterances at different ages. It is, to the best of our knowledge, the largest publicly available age-labelled speech corpus. This, and the fact that it is a subset of VoxCeleb 2, prompted us to select this dataset for our age-related experiments.

VoxCelebPT [191] is a Portuguese version of VoxCeleb, containing recordings of 51 Portuguese celebrities obtained online. This corpus amounts to a total of 26,736 utterances, manually annotated with sex and age labels. In this work, we use a subset of this corpus, containing 25,929 utterances with a minimum length of 1s.

In our experiments, we used AgeVoxCeleb – train_agevox – as the training data for the VQ-VAE and the age classifier. Given the small size of this dataset, when compared to the one used for sex classification, we decided to use the same partition for both the VQ-VAE and the attribute classifier, as our preliminary experiments with smaller partitions showed poor performance for age regression.

VoxCelebPT is used as held-out test data – test_voxpt. Even though it is also comprised of interviews, under a wide variety of recording conditions – the reason for which it was selected – this dataset can

Table 7.3: Data partitions for AgeVoxCeleb and VoxCelebPT.

Source dataset	Partition	Utt./Spk.	<=20	30-39	40-49	50-59	60-69	>=70	Total
AgeVoxCeleb	train_agevox	#Speakers	1,531	1,773	1,292	921	567	217	4,220
		#Utterances	26,970	34,856	30,548	25,751	17,686	5,757	141,568
VoxCelebPT	test_voxpt	#Speakers	7	12	14	7	6	5	51
		#Utterances	3,855	6,610	7,722	3,402	3,034	2,113	26,736

also be considered out-of-domain data, as it only contains recordings of European Portuguese.

7.4.3 Evaluation

To evaluate the performance of our method in terms of privacy concerning sex information, we use two binary classification metrics: Unweighted Average Recall (UAR) and Area Under the Precision-Recall Curve (AUPRC). The UAR reflects the performance of a classifier on a fixed threshold, whereas the AUPRC reports the average classifier performance over all possible classification thresholds. Both have a chance level of 50% for binary classification with imbalanced datasets. These metrics should be as close to 50% as possible for privatised speaker embeddings and as close to 100% as possible for the original, non-protected vectors.

For comparison with the work of [221], we also report two Privacy Zebra metrics [211]. The first Zebra metric is D_{ECE}, the *expected privacy disclosure* which compares the amount of information provided by the oracle-calibrated output log-probabilities of a classifier and that of a non-informative posterior. The second Zebra metric we consider is the llr_{max}, which measures the worst-case privacy disclosure among the test data by selecting the highest log-likelihood ratio for a single sample over oracle calibrated log-probabilities. For both metrics, values close to zero correspond to better privacy protection.

For age, we use the Concordance Correlation Coefficient (CCC) and Pearson's Correlation Coefficient (PCC) as metrics. The CCC measures whether the classifier's output exactly matches the provided labels, being a conservative estimate of the classifier's performance. On the other hand, the PCC measures correlation up to a linear transformation, corresponding to a more optimistic view of the classifier's performance.

Speaker verification performance is evaluated in terms of EER and of the minimum of the Detection Cost Function (minDCF) (minDCF). We use the cosine similarity between two embeddings as the scoring method.

7.4.4 Implementation details

We use SpeechBrain's pre-trained ECAPA-TDNN [78, 265] as our speaker embedding extractor. This model was trained on the development set VoxCeleb 1+2, as described in Data. Speaker embeddings

extracted from the ECAPA-TDNN have a size of 192. The complete architecture of this network can be found in [78].

The encoder and decoder modules of the VQ-VAE (for both attributes) are composed of 3 hidden layers, all of size 512, except for the 3$^{\text{rd}}$ layer of the encoder, which has size $h = 128$, to create a bottleneck. The decoder has an output layer of size $n = 192$ to match the input embeddings. The quantization module is composed of $G = 64$ codebooks, with $V = 128$ entries of size $(e/G) = 4$. The quantisation module linear transformation layer has dimension $q = 256$, whereas the external logits linear layer has size $w = 4$ to match the size of the codewords. In total, our model amounts to $\sim$ 1M parameters.

Attribute classifiers are composed of 2 hidden layers of size 128 and an output layer of size c_{attr}, corresponding to the number of classes of the attribute at hand – 2 for sex and 1 for age. The adversarial classifier is composed of an input Batch Normalisation (BN) layer [124], 3 hidden layers of size 128, and an output layer of size c_{attr}. All hidden layers consist of a linear layer, a Leaky-ReLU activation, and a BN layer. To compute the L_{aam} loss, speaker classification is performed with a linear layer, pre-trained with the same data used to train the VQ-VAE. This layer is frozen to force the model to ensure perfect reconstruction.

All models were trained with Adam [146], using a one-cycle learning rate (lr) policy [291]. VQ-VAE models were trained for 100 epochs, using a start lr of $8{\times}10^{-4}$, and a maximum of 0.01, dropout probability of 0.1 and a batch size of 128; attribute classifiers were trained for 20 epochs, with a start lr of 10^{-5}, and a maximum of $5{\times}10^{-5}$, a dropout probability of 0.3 and a batch size of 64. When training the VQ-VAE for the sex attribute, we ensure batches are always balanced in terms of sex, per sample.

For all experiments, except for the manipulation experiment, when testing the VQ-VAE, the decoder is fed with the same *fake* attribute. This fake attribute corresponds to the mean value of the logits outputted by the pre-trained external attribute classifier, computed over the full training set. The reasoning behind this selection is that, by providing the mean logits for the attribute, we are providing a possible attacker with the least possible amount of information [222].

When performing the attribute manipulation experiment, the VQ-VAE is fed random attribute logits that follow a simple Gaussian distribution to ensure they fall within the observed range of logit values. We select random attribute logits in this experiment to ensure that there is sufficient coverage of possible attribute values when testing the performance of the pre-trained classifier over these *fake* attributes.

Both mutual information losses use $k = 4$ neighbours and the l^2-norm as the distance metric. L_{aam} has a margin of $m = 0.2$ and a scale factor of $s = 30$.

For all VQ-VAE models, the reconstruction loss L_{rec} has weight $\alpha = 1.0$, the codebook diversity loss L_{div} has weight $\beta = 0.1$, and the Additive Angular Margin loss L_{aam} has weight $\gamma = 1.0$.

For the sex attribute, the VQ-VAE is trained with $\delta = 1000$ when using only the adversarial classifier,

with $\epsilon = 100$ when using only the mutual information loss, and $\delta = \epsilon = 10$ when both losses are used. For the age attribute, the VQ-VAE is trained with $\delta = 1$ when using only the adversarial classifier, with $\epsilon = 100$ when using only the mutual information loss, and $\delta = 10$, $\epsilon = 1$ when the two losses are used in combination. This selection was made through a hyper-parameter search, using powers of ten in the range of $[0.1, 1000]$ as the weights for each loss.

To train the NFzLLR model, we use the authors' original implementation [221], available online[2]. We use the same data partitions that we use to train and test our models. Since a hyper-parameter search for this model was out of the scope of this work, we tried the two hyper-parameter configurations used by the authors in [220, 221]. By comparing the results for both configurations, we determined that the hyper-parameters used in [220] provided the best results in terms of privacy. Moreover, these hyper-parameters were selected for ECAPA-TDNN speaker embeddings, the same as the one used in this work. Nonetheless, in our experiments, the hyper-parameter configuration of [221] provided better results in terms of speaker verification.

All attribute classification (or regression) results were obtained by training the attribute classifiers 25 times, with different random initialisations. All privacy metrics are reported as the mean ± standard deviation, computed over all runs. Speaker verification results are obtained over a single run, as there is no source of randomness in this experiment.

7.5 Results

This section provides the results of our experiments. In the first two subsections, we report results for the sex and age removal experiments (experiments 1 and 2). After, we report the results of the experiments regarding the manipulation of sex information (experiment 3) and the cross-domain experiments (experiment 4).

7.5.1 Removal of sex information

The results for the removal of sex information can be found in Table 7.4 for the ignorant attacker and in Table 7.5 for the informed attacker. In both tables, down-pointing arrows mean that lower values are better.

In each table, we report sex classification results for the *Original* (i.e., non-transformed) speaker embeddings, as well as the results obtained for *NFzLLR* [221]. This is followed by the results of the ablation study, where we include results for the VQ-VAE trained without any adversarial loss, for the combination of the VQ-VAE with either the mutual information or the adversarial loss, and for the complete method, using a combination of both losses.

[2] https://github.com/LIAvignon/bridge-features-evidence

From Tables 7.4,7.5, we can observe that each component of our method provides consistent improvements over the simple VQ-VAE. By adding the mutual information loss to the method, we observe a sex classification performance degradation of more than 15% for UAR and AUPRC when compared to the VQ-VAE for both attacker settings. When adding the adversarial classifier and loss, we see a similar improvement to that of the mutual information loss for the *ignorant attacker* setting. However, for the *informed attacker*, the degradation is much more pronounced, over 20% UAR and AUPRC, showing that the adversarial classifier provides a better ability to remove sex information. This is to be expected, as the adversarial loss is parametric – it is based on a classifier – whereas the mutual information loss is non-parametric.

Notably, the results show that combining the adversarial classifier with the mutual information loss also yields the best overall performance in terms of privacy protection. This proves that these two approaches complement each other with regard to information removal, validating our method. In terms of the Zebra metrics, the results follow a similar trend, with each component providing consistent improvements over the baseline.

One should also note that none of the considered methods is able to remove sex information entirely. This can be seen in the results for the *informed attacker*, where the sex classification performance reaches values close to 60% UAR and AUPRC.

For the target task, speaker verification, the results show that the proposed method introduces an absolute degradation of 1.2% and 1.6% EER for the VQ-VAE trained with the mutual information loss and the adversarial loss, respectively, when compared to the original vectors. On the other hand, the combination of the two losses introduces a degradation of only 0.6% EER. A possible reason for this is the fact that, for this model, the weights of both losses are set to 10.0, whereas for the mutual information or adversarial-only models, the corresponding weights are 100.0 and 1000.0. For this reason, these losses will have a much higher impact with regard on the MSE and L_{aam} losses, where the weights are set to 1.0 and 0.1. This set of weights was chosen because it provided the best performance in terms of privacy.

When comparing our approach to that of [221], we see that our complete method (VQ-VAE+ADV+MI) is on par with the NFzLLR for privacy protection for the ignorant attacker, in terms of the classification metrics, whereas for the Zebra metrics, our method provides worse privacy results. This may be because the NFzLLR model was specifically developed to minimise the amount of information disclosed to an attacker – the log-likelihood ratio between the two classes is set precisely to 0 – which is exactly what is measured by the Zebra metrics. In our model, we are providing the mean "attribute" for all samples, which does not necessarily carry zero information about any class, i.e., pre-trained classifiers may interpret the mean as one class instead of no class.

Contrarily, considering the informed attacker, our method shows a much better ability to protect sex

Table 7.4: Results regarding the removal of sex information for ignorant attackers.

Model	Speaker Verification Metrics		Sex Classification Metrics		Sex Privacy Metrics	
	EER (%) ↓	**minDCF** ↓	**AUPRC (%)** ↓	**UAR (%)** ↓	$\mathbf{D_{ECE}}$↓	$\mathbf{llr_{max}}$↓
Original data	0.88	0.0011	99.40 ± 0.11	97.74 ± 0.28	0.649 ± 0.007	3.444 ± 0.176
NFzLLR [221]	4.89	0.0043	**51.29 ± 0.96**	51.72 ± 0.66	**0.002 ± 0.001**	**0.633 ± 0.245**
VQ-VAE	**1.44**	0.0021	82.35 ± 1.09	73.82 ± 1.35	0.218 ± 0.014	2.262 ± 0.227
VQ-VAE + MI	2.12	0.0026	60.54 ± 1.30	56.11 ± 1.31	0.039 ± 0.009	1.690 ± 0.394
VQ-VAE + ADV	2.45	0.0029	56.72 ± 0.84	54.76 ± 0.78	0.016 ± 0.004	0.883 ± 0.327
VQ-VAE + ADV + MI	1.48	**0.0019**	52.92 ± 0.92	**50.91 ± 0.60**	0.005 ± 0.002	0.761 ± 0.289

Table 7.5: Results regarding the removal of sex information for informed attackers.

Model	Sex Classification Metrics		Sex Privacy Metrics	
	AUPRC (%) ↓	**UAR** (%) ↓	$\mathbf{D_{ECE}}$ ↓	$\mathbf{llr_{max}}$ ↓
Original data	99.40 ± 0.11	97.74 ± 0.28	0.649 ± 0.007	3.444 ± 0.176
NFzLLR [221]	74.59 ± 0.85	71.36 ± 0.68	0.138 ± 0.008	1.839 ± 0.177
VQ-VAE	90.89 ± 0.68	85.67 ± 0.70	0.367 ± 0.013	2.844 ± 0.158
VQ-VAE + MI	72.78 ± 1.09	70.31 ± 0.89	0.132 ± 0.010	2.345 ± 0.197
VQ-VAE + ADV	63.18 ± 0.84	62.62 ± 0.69	0.052 ± 0.005	1.474 ± 0.195
VQ-VAE + ADV + MI	**57.41 ± 0.67**	**57.71 ± 0.87**	**0.021 ± 0.004**	**1.145 ± 0.255**

information, with a difference of more than 10% for the classification metrics. For the Zebra metrics, our method also shows a marked improvement over the NFzLLR. In addition, the NFzLLR shows a much higher degradation for speaker verification, being close to 5% EER, as opposed to our 1.5%. However, these results differ from those provided in [221], where the model had much better behaviour against informed attackers and where the degradation introduced by the model was much lower. One possible explanation for the privacy results may be the fact that in [221], only 71 speakers and 17,735 utterances were used to train the attribute classifier, whereas, in this work, we use 400 speakers and 67,279 utterances. For the results in terms of speaker verification, a possible reason may be the fact that, unlike [221], we use cosine scoring instead of PLDA scoring to perform speaker verification. Nevertheless, it is necessary to state that no hyper-parameter tuning was made for the NFzLLR and that better results could be obtained by performing a hyper-parameter search.

7.5.2 Removal of age information

The results regarding the removal of age information can be found in Table 7.6. Similar to the sex attribute experiment, we observe a consistent improvement with each loss being added to the model, with the combination of the mutual information and adversarial losses providing the best results in both attacker settings.

In particular, we observe a 90% relative improvement in terms of privacy for both correlation metrics in the ignorant attacker, a value that is reduced to between 80-85% for the informed attacker. When compared to the results for sex, this improvement is much higher. For the sex attribute, the relative improvement was close to 40% AUPRC and UAR for the ignorant attacker and close to 45% for the informed attacker. This shows that our method is able to generalise to continuous attributes successfully.

Nevertheless, for this attribute, the informed attacker does not provide a performance improvement over the ignorant attacker, as was observed for the sex information, for the cases where the VQ-VAE is only combined with one of the two losses. Moreover, we must also note that for the best privacy model, the ASV performance suffers from a degradation of 3.4% EER, which is much larger than for the sex attribute, where the degradation was kept at 0.6%.

A possible reason for these two phenomena may be the amount of data used to train the VQ-VAE in this experiment, which corresponds to about one-eighth of the amount of data used for the sex attribute experiment. The degradation of the speaker representations that is indicated by the poor ASV performance may also affect the age regression model, such that even when it is trained over the transformed representations, it is not able to generalise properly to unseen data.

As such, we hypothesise that observing such a lower amount of data during training may have prevented the model from achieving a better trade-off between privacy and utility, with the model

Table 7.6: Results for age regression for both ignorant and informed attackers.

Model	Speaker Verification Metrics		Age Regression Metrics			
			Ignorant Attacker		Informed Attacker	
	EER (%) ↓	**minDCF ↓**	**CCC ↓**	**PCC ↓**	**CCC ↓**	**PCC ↓**
Original data	0.88	0.0011	0.681 ± 0.005	0.753 ± 0.003	0.681 ± 0.005	0.753 ± 0.003
VQ-VAE	**1.74**	**0.0018**	0.194 ± 0.009	0.370 ± 0.015	0.198 ± 0.013	0.315 ± 0.021
VQ-VAE + MI	1.97	0.0024	0.147 ± 0.011	0.279 ± 0.020	0.160 ± 0.012	0.259 ± 0.018
VQ-VAE + ADV	2.68	0.0027	0.117 ± 0.010	0.229 ± 0.020	0.119 ± 0.011	0.184 ± 0.017
VQ-VAE + ADV + MI	4.24	0.0039	**0.042 ± 0.009**	**0.084 ± 0.018**	**0.101 ± 0.012**	**0.165 ± 0.020**

Table 7.7: Results for the proposed methods for sex and age information manipulation within the speaker representations.

Model	Speaker Verification		Sex Classification		Speaker Verification		Age Regression	
	EER (%) ↓	**minCLLR ↓**	**AUPRC (%) ↑**	**UAR (%) ↑**	**EER (%) ↓**	**minCLLR ↓**	**CCC ↑**	**PCC ↑**
Original data	0.88	0.0011	99.40 ± 0.11	97.74 ± 0.28	0.88	0.0011	0.681 ± 0.005	0.753 ± 0.003
VQ-VAE	1.13 ± 0.04	0.0016 ± 0.0001	91.94 ± 0.34	85.09 ± 0.85	**1.56 ± 0.03**	0.0015 ± 0.0001	0.883 ± 0.007	0.889 ± 0.003
VQ-VAE + MI	1.24 ± 0.05	0.0016 ± 0.0001	95.13 ± 0.74	86.98 ± 0.84	1.72 ± 0.02	0.0021 ± 0.0001	0.898 ± 0.008	0.908 ± 0.003
VQ-VAE + ADV	1.65 ± 0.05	0.0022 ± 0.0002	96.94 ± 0.15	**90.97 ± 0.83**	2.41 ± 0.04	0.0024 ± 0.0001	**0.915 ± 0.007**	0.926 ± 0.002
VQ-VAE + ADV + MI	**1.03 ± 0.04**	**0.0014 ± 0.0001**	**97.23 ± 0.18**	90.23 ± 0.68	3.71 ± 0.04	0.0034 ± 0.0001	0.914 ± 0.014	**0.934 ± 0.002**

degrading the signal more in favour of privacy.

7.5.3 Attribute manipulation results

To fully validate our model, it is also necessary to understand how well it incorporates the information that is fed into the decoder and, consequently, how well it can manipulate attribute information within the speaker embedding.

To do so, we performed a set of experiments using the models trained for each attribute, where pre-trained classifiers are tested with regard to the "fake" attribute labels fed to the model's decoder. Differently from the prior experiments, here, the "fake" attribute is random for every sample, as we want to cover both classes, for sex, and a widespread range of values for age. Specifically, we generate random logits using a distribution trained over the output logits of the external classifier for the training set. In the case of the sex classification model, to obtain the label of each vector of logits, we take the argmax and use the corresponding index.

We also test the performance with regard to ASV performance, wherein the same information is used to condition both samples in same-speaker trials. For different speaker trials, different attribute information is used for either sample.

The results for this experiment are presented in Table 7.7. We do not report here Zebra metrics, as they measure information disclosure and, thus, are not relevant for this task.

Contrary to prior experiments, in this experiment, for sex information the full model does not clearly improve in terms of classification metrics over the adversarial loss-only model, with only small differences observed for the AUPRC (higher for the full model) and UAR (higher for the adversarial-only model). Nevertheless, in terms of ASV performance, the full model outperforms all models.

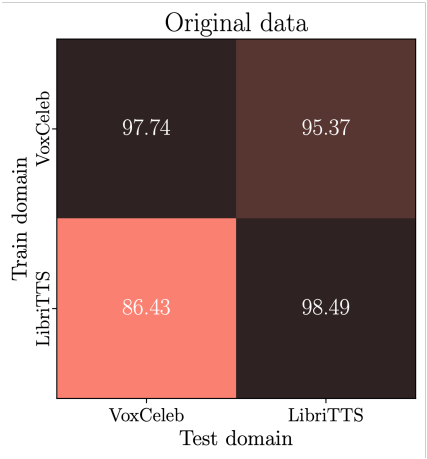
In the case of the age manipulation experiments, we observe a similar pattern, with the full and adversarial-only models showing only slight differences for CCC (higher for the adversarial-only model) and PCC (higher for the full model). For age, we also observe that the values obtained in terms of CCC and PCC are much higher (and improvement of $\sim$0.2) than those obtained for the original data, as opposed to what was shown by the sex information manipulation experiments, where the classification metrics presented some degradation when compared to the original data. We hypothesise that, in the case of sex information, some logit configurations may be very close to the classification boundary between the two classes, whereas for age, given that it is a regression task, this may happen less often.

The fact that the best models are able to achieve a 90% UAR and 0.91 CCC for "fake" attribute prediction with pre-trained classifiers shows that our model is capable of manipulating the attribute information within the speaker embedding. Moreover, the performance in terms of speaker verification is better than the performance obtained for the original experiments (cf. results in Tables 7.4 and 7.6), presenting a degradation of only 0.15% EER when compared to the original data, for the sex

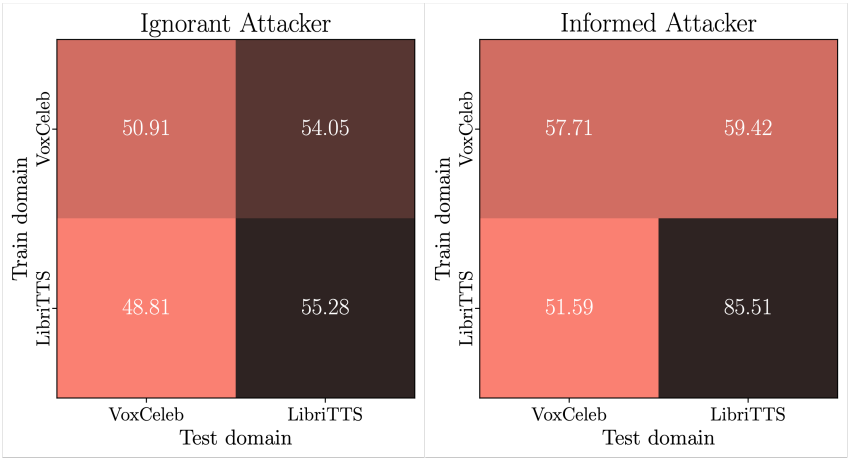
manipulation model, and a ∼3% EER degradation for the age manipulation model. The likely reason for this is that the same attribute information is being used for same-speaker trials, and different information is being used for different-speaker trials. In other words, embeddings corresponding to the same speaker will be transformed with the same ”fake” information (i.e., the same random logits), bringing them closer together. Conversely, pairs of different speakers will be further apart, as the random logits will be different for each vector. This will make the pairs more discriminative and hence improve the speaker verification results.

7.5.4 Cross-domain results

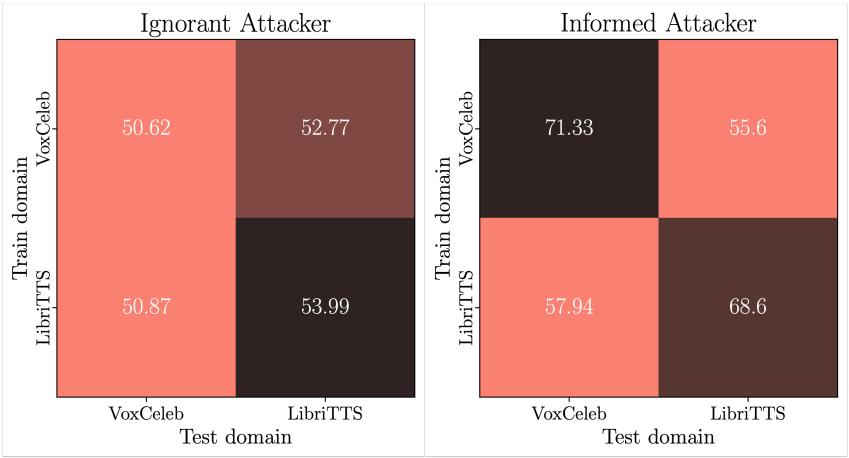
In this section, we discuss the cross-domain experiments for the sex attribute. These experiments aim to provide an understanding of how well our models can generalise their ability to remove attributes to unseen domains. As stated in Section 7.4.3, we perform a total of 8 experiments (cf. Table 7.2), using two datasets (VoxCeleb and LibriTTS) to train the VQ-VAE and to train and test the attribute classifier. These experiments are performed with the two types of attackers, ignorant and informed, as well as for the original non-manipulated data. In total, this results in 28 experiments, the results of which can be found in Fig. 7.2. For conciseness, this figure only reports results in terms of mean UAR. For every sub-figure, the Y-axis corresponds to the domain used to train the attribute classifier, whereas the X-axis corresponds to the domain of the test data. Darker colours indicate higher UAR values, and conversely, lighter colours indicate lower UAR values.



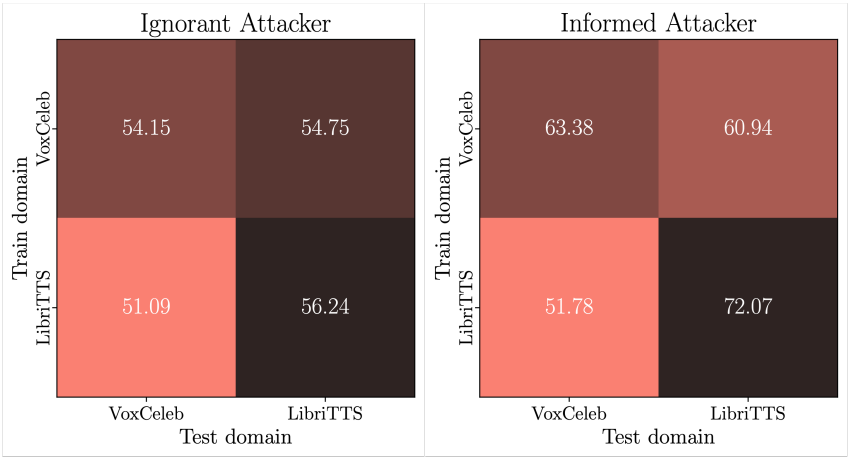
(a) Original data.



(b) VQ-VAE trained on VoxCeleb.



(c) VQ-VAE trained on LibriTTS.



(d) VQ-VAE trained on 50% VoxCeleb and 50% LibriTTS.

Figure 7.2: Results for the cross-dataset experiments.

Regarding the cross-domain results for the original data, shown in Fig. 7.2a, we can observe that each domain tested against itself (diagonal squares) provides very high results, with the highest UAR for sex classification corresponding to attribute classifiers trained and tested on LibriTTS. In the values in the counter-diagonal, whereas the classifier trained on VoxCeleb and tested on LibriTTS provides good results, around 95% UAR, the opposite shows a UAR of around 86.5%, amounting to an absolute degradation of almost 10%. This trend is observed in most of the remaining experiments, showing that sex attribute classifiers trained on LibriTTS do not generalise well to VoxCeleb. A possible reason for this is the fact that LibriTTS contains samples of read speech under very controlled conditions (Audiobooks). In contrast, VoxCeleb is composed of interviews recorded in very diverse and noisy conditions, making it easy for the classifier trained on VoxCeleb to obtain good results in the clean conditions of LibriTTS, and the opposite much harder.

For the data manipulated using the VQ-VAE model trained on VoxCeleb, in Fig. 7.2b, we observe the same effects of training the attribute classifier on LibriTTS and testing it on VoxCeleb. However, considering the LibriTTS testing results, we can see that our model is not able to perform as well as for VoxCeleb for both attackers. This is most evident for the informed attacker, where the sex classifier trained and tested on LibriTTS achieves an 85% UAR, showing that the model is somewhat domain-specific.

To understand the source of the domain dependence in our method, we trained a VQ-VAE with LibriTTS and performed the same cross-domain experiments. In Fig. 7.2c, we see that the performance for the attribute classifier trained and tested on LibriTTS is much better for privacy, dropping around 17% UAR, for the informed attacker, when compared to the VQ-VAE trained with VoxCeleb. Moreover, for the informed attacker, we observe almost equal performance when training and testing the attribute classifiers on the same domain or in cross-domain settings. Nonetheless, the performance of the VQ-VAE for LibriTTS in the informed attacker scenario is not on par with the model trained on VoxCeleb. One of the reasons may be the fact that the model was trained with much less data: $\sim$205,000 utterances for LibriTTS versus $\sim$880,000 utterances for VoxCeleb.

Finally, we also explore the behaviour of our model when trained on both domains. To do so, we use the same amount of data taken for both datasets. In this case, we observe a degradation of the results when testing in the original VQ-VAE training domain. However, when the model is tested across training domains (e.g., the VQ-VAE is trained on VoxCeleb and tested for privacy on LibriTTS), it performs better than the VQ-VAEs trained for individual domains.

Specifically, in the scenario where the attribute classifier was trained and tested on VoxCeleb, the result for the informed attacker presented in Fig. 7.2d, shows a degradation of $\sim$5.5% UAR when compared to the in-domain value presented in Fig. 7.2b. Moreover, when considering the attribute classifiers trained and tested on LibriTTS, the result shown in Fig. 7.2d presents a degradation of $\sim$ 3.5% UAR, when

compared to the in-domain result of Fig. 7.2c. Contrarily, the attribute classifier trained and tested on LibriTTS, obtained using the out-of-domain VQ-VAE trained on VoxCeleb (cf. Fig. 7.2b), the model trained on both datasets shows an improvement of ∼13% UAR. In addition, the attribute classifier trained and tested on VoxCeleb shows an improvement of ∼8% UAR, when compared to the out-of-domain VQ-VAE trained on LibriTTS (cf. Fig. 7.2c). This supports the argument that combining multiple domains in the training data helps make the model more robust to those domains. For the ignorant attacker, the performance is stable across the three experiments, with the results obtained for the VoxCeleb test set being close to chance level, and for the LibriTTS test set averaging around 54.5%.

Overall, the results of these experiments for the informed attacker indicate that the performance of the VQ-VAEs is dependent on the domain of data they were trained on. On the other hand, for the ignorant attacker, the models' performance appears to be independent of the data used to train and test the attribute classifiers. Moreover, the general approach in itself seems to be independent, with our results showing that different models can be trained on data from specific domains to obtain better results in these domains.

7.5.5 Limitations

The results detailed in the previous sections show that the proposed method fulfils the objectives set at the end of Section 7.2. Specifically, the trained models allow the suppression of the two target attributes, sex and age, achieving privacy results close to chance level in in-domain settings, as well as in several cross-domain settings. Moreover, our experiments regarding sex information have shown that the proposed method is in fact able to manipulate attribute information, instead of simply removing it. Nevertheless, the proposed method still presents some limitations. For instance, the sex and age attribute classification results show that our method is still unable to remove all attribute information. This means that, for stronger attackers, it may still be possible to recover this information. On the other hand, the measure of the utility of the proposed method rests solely on ASV performance. To fully understand the impact of the proposed method, it would be important to evaluate its effects on the detection of other speaker traits or conditions which may be important for other downstream tasks. In addition, the proposed method does not provide a clear way to trade off utility and privacy. For instance, the results pertaining to the age attribute that are shown in Table 7.6 indicate that as each component of the method is added, the speaker verification results degrade, whereas privacy improves. However, for sex information, this is not the case, and only the baseline VQ-VAE is able to achieve a better ASV result when compared to the full method (VQ-VAE + ADV + MI). One could also consider changing the weights of each loss to manipulate this trade-off. However, our preliminary experiments – wherein the weights for each loss were varied logarithmically between 0.1 and 1000 – showed that this

relation was not linear, i.e., increasing the losses' weights did not always correlate with either more privacy or less utility. We consider that making this trade-off clearer and easier to control is an important objective for future study.

7.6 Summary

In this chapter, we propose the use of a combination of a VQ-VAE, an adversarial classifier, and a mutual information loss to remove or manipulate sex and age information in speaker representations. Our model was tested in an ASV setting, where both the speaker representation extraction step and the application of our model are assumed to be performed in the user's device. Our model is much smaller (∼1M parameters) than the speaker representation extraction model (∼14M parameters), corresponding to a small additive cost in terms of the overall computational cost of the ASV pipeline. The experiments that were conducted prove the validity of the proposed method and show that our model is able to drop the classification or estimation performance of both attributes to close to the chance level while keeping the utility of the speaker representations for ASV. The proposed models were also successfully validated with regard to the manipulation of both attributes. In addition, a cross-domain study showed that our method generalises to a different domain, for ignorant attackers, and, even though its results suffer some degradation when considering the informed attacker, re-training the model with out-of-domain data, or a mixture of in- and out-of-domain data helps improve these results, showing that our approach generalises to different domains. Moreover, to the best of our knowledge, this work is the first to consider the removal of age information from speaker representations. The avenues for future work are vast, with numerous topics worth exploring. In terms of privacy, the proposed method could be tested for the removal of multi-class attributes such as accent information. Other paralinguistic traits, such as emotional information could also be worth exploring. Another possible extension of this work would be its application to domain generalisation, i.e., minimising the amount of domain information contained in speaker representations [173]. Alternatively, one could also explore the cross-attribute effect of each of the attribute models, for instance, by measuring the effect of the age removal model on sex classification performance and vice versa. This would allow a more in-depth understanding of the effects of attribute removal models. A similar line of work would be the application of each of the models in sequence to understand whether it is possible to remove both age and sex information from the same speaker embedding with the proposed methods.
Another potentially very relevant research line would be the use of the proposed model in voice conversion and text-to-speech tasks as a way to manipulate and control speaker traits, as well as to anonymise speech to some extent [220]. Training our model for these tasks would also show the applicability of our model to different speaker representation extractors, as well as its robustness to different downstream applications.

8

Membership Inference in ASR Model Auditing

Membership Inference (MI) – the task of determining whether a data point is part of an ML model's training dataset – poses a significant threat to the privacy of the training data of ASR systems. At the same time, MI also offers an opportunity to audit these models concerning the potentially unauthorised use of data, as well as the level of privacy of their training data. The work presented in this chapter explores MI within this auditing scenario, focusing specifically on the effectiveness of loss-based features in combination with Gaussian and adversarial perturbations to perform MI in ASR models, something that, to the best of our knowledge, has not been explored by previous works. This is done at two levels: *sample level*, where the goal is to determine the training data membership of a specific speech sample, and *speaker level*, where the goal is to determine if data from a specific speaker was part of the model's training dataset. We compare the proposed features with commonly used error-based features, finding that they are able to enhance performance for sample-level MI. For speaker-level MI, these features improve results, though by a smaller margin, as error-based features already obtained a high performance for this task. Our findings emphasise the importance of considering different feature sets and levels of access to target models for effective MI attacks in ASR systems and provide valuable insights for auditing such models.

8.1 Introduction

Automatic Speech Recognition (ASR) systems are revolutionising the way we interact with technology. The recent progress of ASR systems has led to the deployment of numerous cloud-based services and applications that leverage speech as a means of human-computer interaction. As stated in Chapter 1, an estimated 4.2 billion voice assistants were in use worldwide in 2020 [302], and the smart-speaker global market share is expected to reach 35.5 billion US dollars by 2025 [303]. The use of these and other speech systems has given rise to concerns regarding user privacy, as was discussed in previous chapters of this thesis. However, their deployment has also prompted concerns about the privacy of the systems' training data subjects [92].

Of particular concern are Membership Inference (MI) attacks, which exploit the susceptibility of ML models to attacks that allow one to infer if specific individuals were included in the model's training dataset, thereby disclosing potentially sensitive information [287]. For instance, if one knows specific characteristics of the population that make up the training set – e.g., a training dataset that consists only of individuals affected by a particular illness – it follows that an individual that is part of this dataset will share these potentially sensitive characteristics [287].

Even though MI is most often considered an attack on the privacy of learning data, it may equally be seen as a tool to protect data donors and service providers. As an auditing tool, MI can provide evidence that available models do not leak information about their training data and show that these models are in adherence to data protection regulations [357], such as the European Union's GDPR [88],

or CCPA [40]. Membership Inference (MI) can also be used to audit service providers' use of customer data. Specifically, a service provider that trains an ML model with user data without adequate consent may violate data protection regulations. In this case, MI can be used to audit the model and to assert whether or not a data sample was used during training, protecting both users and service providers [172, 195, 299].

MI is thus an essential aspect of trustworthy machine learning that should be studied in all its facets and for all types of data. However, while MI has been extensively studied in the domains of image and text data [119], the focus on speech data [50, 151], and particularly in what concerns ASR models [172, 195, 283, 324], remains limited. Most of the scarce literature on MI in ASR has focused on the use of transcription errors [283], transcription-reference similarity scores [195], or both [172], as features to classify membership. The work of Tseng et al. [324] is an exception to this and explores MI in self-supervised speech models using frame-similarity scores. MI in ASR has also been considered under different target use cases: Shah et al. [283] and Tseng et al. [324] view their work as a traditional MI attack, targeted at understanding the vulnerabilities of ASR models, whereas Miao et al. [195] and Li et al. [172] pose their work from an auditing perspective, where MI is a tool to check for unauthorised use of data.

Nevertheless, all of these works have a strict adherence to *black-box* scenarios, which, in the case of [172, 195, 283] means that only processed (i.e., decoded) model outputs are available, and consequently, only error-based features are used. Contrarily, we argue that having access to the model's output logits is a reasonable assumption that should be explored. We consider this to be particularly true in auditing scenarios, where service providers are under scrutiny for potentially having trained their model on user data without consent and are required to provide some level of model access to the auditor.

In this study, we focus on the auditing scenario for ASR models. We consider grey- to white-box access to the model, specifically, access to the raw output of the ASR model and some knowledge of the training data distribution (grey-box), as well as the ability to back-propagate through the model (white-box). Our focus also extends beyond sample-level MI to include speaker-level MI, i.e., inferring whether an individual's data was part of the model's training data without knowing the exact samples that were used for this purpose.

Under these assumptions, we explore loss information (i.e., Kullback-Leibler (KL) divergence and Connectionist Temporal Classification (CTC) loss) when performing MI, which, to the best of our knowledge, no previous work on the topic of MI for ASR has used. To gain more information about the decision boundary surrounding a given utterance, we further enrich these features by computing the losses over two types of input perturbations: Gaussian noise and adversarial noise. Similar perturbations have been explored in other domains but using different protocols [57, 272].

We conduct our experiments with Transformer [332] and Conformer [115] models, trained on subsets of LibriSpeech [232]. We observe that loss features outperform error features at sample-level MI, particularly when combined with the proposed perturbations. At speaker-level MI, we observe closer results for both sets of features, with loss features still being able to achieve higher performances.

The remainder of this chapter is organised as follows: in Section 8.2, we describe our methodology and the proposed features and perturbations; in Section 8.3, we describe the experimental setup; and in Section 8.4 we present and discuss the results obtained. Finally, Section 8.5 summarises this chapter, drawing some conclusions and presenting topics for future work.

8.2 Methodology

To perform MI, we apply a similar methodology to previous works [172, 283, 287]. Given a *target* model to perform MI, we first train a *shadow* model on a dataset that is disjoint from the one used to train the target model. We then build a balanced binary classification dataset of input utterances labelled positively *iff* they are in the shadow model's training set and negatively otherwise. The set of speakers for positive and negative samples is the same to ensure our classifier is distinguishing between seen and unseen samples and not between seen and unseen speakers. Next, we train a binary classifier for MI on this dataset, using the features that will be described in the remainder of this section. As a final step, this binary classifier is used to evaluate a test set of utterances, labelled as above, but with regard to their membership in the training set of the target model. We refer to this process as *sample-level* MI.

To perform *speaker-level*, utterances are labelled positively *iff* their *speaker* was in the training set and negatively otherwise. To ensure that the MI classifier is recognising speaker membership and not sample membership, we ensure that positive samples are not part of the ASR model's training data. In what follows, we present the three feature categories that were used in our MI framework.

8.2.1 Baseline: error features

Our baseline feature extractor corresponds to a set of errors computed between the target and output transcriptions of the ASR model, combined with the model's confidence for these transcriptions, being inspired by the best-performing set of the features evaluated in [283].

Specifically, we use the WER; the length-normalised counts for edits, substitutions, insertions and deletions; the length ratio between the prediction and target transcription; and the confidence of the model regarding the transcription. We compute all these features for the top-4 transcription hypotheses of the model, obtaining 24 features, and dub their combination as the *errors* feature set.

8.2.2 Loss-based features

The main focus of this work is the set of features that can be computed from the non-processed (i.e., non-decoded) output logits of the model. We consider that these features contain a higher amount of information on membership than features computed from a post-processed output, as long as they are correctly modelled [42]. As loss-based features, we consider the losses used to train a transformer-based ASR model: the attention loss, which corresponds to the KL divergence between the output log-probabilities and the target transcription, and the CTC loss [114].

8.2.3 Perturbed features

To characterise the decision boundary around a given data point and potentially improve the MI decision, we extend the loss features by perturbing the input signal using Gaussian and adversarial noise.

Gaussian noise Inspired by [57, 130], we perturb input data with random Gaussian noise. Gaussian perturbations are agnostic to the model and data and let us evaluate the model's "average" behaviour when getting further away from the input in arbitrary directions. We use decreasing levels of the Signal-to-Noise Ratio (SNR), moving the perturbed signal away from the original input. This set of perturbed signals is then fed to the ASR model, from whose output we compute our set of MI features. Since using a single perturbation per SNR value would only give us information on the decision boundary regarding one random direction, for each SNR value, we select multiple random perturbations. The MI features computed from these random perturbations for the same value of SNR are then summarised by their mean and standard deviation. This procedure is summarised in Algorithm 4.

Algorithm 4 Gaussian noise-based feature computation.

Require: Input x, set of SNRs S, #runs N, model $M(\cdot)$, target transcription y, feature extractor $F(\cdot)$

feats $\leftarrow [\,]$
for snr $\in S$ **do**
  $\text{feats}_{\text{snr}} \leftarrow [\,]$
  for $n \leq N$ **do**
    $\delta \sim \mathcal{N}(0, I)$ {Sample Gaussian noise}
    $\delta_{\text{snr}} \leftarrow \sqrt{\frac{\|x\|_2^2}{\text{snr} \times \|\delta\|_2^2}} \times \delta$ {Scale noise to SNR}
    $\text{feats}_{\text{snr}}[n] \leftarrow F(M(x + \delta_{\text{snr}}), y)$
  end for
  $\text{feats}[\text{snr}] \leftarrow (\text{mean}(\text{feats}_{\text{snr}}), \text{stddev}(\text{feats}_{\text{snr}}))$
end for

return feats

Adversarial noise In addition to random perturbations, we propose to explore "worst-case" directions for which the decision boundary is near the data point. Contrary to [272, 366], we do not estimate the "distance to the decision boundary" (an ambiguous notion for transduction tasks) but instead find directions of maximal error given a fixed perturbation budget. To do this, we run a panel of adversarial attacks, i.e., algorithms that find small perturbations of inputs that can fool ML models into changing their decisions. Details about adversarial attacks can be found in [113], and in [1] for ASR in particular. We focus on the untargeted Projected Gradient descent attack [185] in the L_∞ norm, a standard for adversarial perturbations. Given a radius ϵ, we compute N gradient steps of step size η, and at every step, clip the perturbation so that $\|\delta\|_\infty \leq \epsilon$. We apply this attack with different radii and compute features for all returned perturbations. We detail this procedure in Algorithm 5. Since it is necessary to perform back-propagation through the model to create the adversarial perturbations, the use of these features entails white-box model access to the target model.

Algorithm 5 Adversarial-based feature computation.

Require: Input x, set of radii $\mathcal{E}$, number of steps N, step size η, model $M(\cdot)$, target transcription y, feature extractor $F(\cdot)$

feats $\leftarrow [\,]$
for $\epsilon \in \mathcal{E}$ **do**
  $\delta_\epsilon \sim \mathcal{U}(-\epsilon I, \epsilon I)$
  for $n \leq N$ **do**
    $g = \text{sign}\left(\frac{d}{d\delta_\epsilon} L(M(x+\delta_\epsilon), y)\right)$ {Gradient}
    $\delta_\epsilon \leftarrow \text{clip}_\epsilon(\delta_\epsilon + \eta g)$ {Optimisation & projection}
  end for
  feats$[\epsilon] \leftarrow F(M(x+\delta_\epsilon), y)$
end for

return feats

8.3 Experimental setting

8.3.1 Experiments

For the experiments of this work, we trained three ASR models, varying in training data and architecture:

- An encoder-decoder transformer model (**T1**) [265];
- A transformer model trained on a disjoint set of data coming from the same distribution as the data used to train **T1** (**T2**);
- A conformer model [115] trained on the same data as **T2** (**C1**).

In our experiments, model **T1** always corresponds to the target model.

To validate our hypothesis – *loss-based features improve upon error-based features* – we performed a comparative and ablative study over the feature sets described in the previous section. Specifically, we compared the performance of the *errors* feature set with the *loss* feature set and with the combination of the losses with the Gaussian perturbations, adversarial perturbations, and both, as well as for the combination of all the features. In these experiments, the shadow and target models are the same to have an upper bound on the performance of each feature set. Specifically, we used model **T1** as both the shadow and target model.

In addition to the above, when performing MI, it is reasonable to consider that different model architectures and models trained on different datasets will behave differently regarding the training losses and output errors. As such, we performed two additional experiments, using as shadow models **T2** and **C1** The experiment where **T2** is used as the shadow model corresponds to the case where the model's architecture is known, while the experiment using **C1** as the shadow model corresponds to the case where the model's architecture is not known. In both cases, the training data of the shadow models (**T2** and **C1**) is different from that of the target model **T1** but comes from the same data distribution. These two experiments emulate auditing settings where access to and knowledge of the target model is limited, providing information about the behaviour of the proposed features in these more challenging but more realistic scenarios.

8.3.2 Corpora

The datasets used to train the ASR target and shadow models, as well as to train the MI classifiers, are built from data taken from LibriSpeech (LS) [232].

Model	Sources	#Hours	#Speakers	#Samples
Transformer ASR **T1**	*train-clean-360*	300	2,097	85,317
Transformer ASR **T2**	*train-clean-100*	80	585	23,408
Conformer ASR **C1**	*train-clean-100*	80	585	23,408

Table 8.1: Partitions used to train each ASR model.

More specifically, the dataset used to train **T1**, our target ASR model, was composed of 300h from LibriSpeech's *train-clean-360* partition.Similarly, the dataset used to train models **T2** and **C1** was composed of 80h from LibriSpeech's *train-clean-100* partition. Additional details about the training sets can be found in Table 8.1.

The datasets used to train the MI classifier were composed of 5,000 utterances, the sample-level test set contained 2,000 utterances, while the speaker-level test set contained 1,000 utterances. All datasets were balanced in terms of positive and negative samples. In all cases, the positive samples were taken from the ASR models' training partitions. For sample level MI, the negative samples were taken from

train-clean-360 for **T1**, and from *train-clean-100* for **T2** and **C1**. For speaker level MI, the negative samples were taken from LibriSpeech's *dev-clean* and *test-clean* partitions. Further details regarding each partition can be found in Table 8.2.

Level	Partition	Shadow model	Sources	#Speakers	#Samples
Sample	Train	**T1**	*train-clean-360*	1,872	5,000
		T2 & **C1**	*train-clean-100*	585	5,000
	Test	**All**	*train-clean-360* & *dev-clean* & *test-clean*	1,308	2,000
Speaker	Train	**T1**	*train-clean-360* & *dev-clean* & *test-clean*	190	5,000
		T2 & **C1**	*train-clean-360* & *dev-clean* & *test-clean*	190	5,000
	Test	**All**	*train-clean-360* & *dev-clean* & *test-clean*	79	1,000

Table 8.2: Partitions used to train and test MI classifiers.

8.3.3 Attack perturbations

We experimented with several choices of hyper-parameters for the perturbations detailed in Algorithms 4 and 5. The parameters reported in this section correspond to those that performed the best for held-out data. In the results we report, for Gaussian perturbations, we use 8 different SNRs linearly spaced between 0 and 50dB. For each SNR, we perturb the signal 4 times, after which we take the mean and standard deviation of the resulting features. For adversarial perturbations, we use 16 adversarial radii ϵ: nine evenly spaced from 0.001 to 0.009 and seven from 0.01 to 0.07. We fix $\eta = 1$ and $N = 1$, which we find to be as effective for MI as more computationally expensive hyper-parameter configurations that use higher numbers of steps. Both perturbations result in a feature set of 32 features, to which we add the two unperturbed loss features, arriving at 34 features in total.

8.3.4 Evaluation metrics

To allow our work to be compared to other approaches in the literature, our main metrics of evaluation are Accuracy (Acc.) and Area Under the ROC Curve (AUC). However, as suggested by [42], in MI in general, to correctly assess the strength of an MI system it is necessary to consider its True Positive Rate (TPR) performance at very low False Positive Rate (FPR) values, a metric which shows if the system is able to identify members with high confidence. In auditing scenarios, it is possible to argue for a similar case, where the "cost" of deciding that a sample is in the training set while it should not be in it (a false alarm), can be much higher than deciding that the sample is not in the training set (a missed detection). For instance, the possibility that a user's data has been wrongfully used to train the model may incur legal consequences for the party responsible for handling the user's data, and for the party that developed and trained the model. As such, similarly to what happens with forensic evidence,

to prove membership the system needs to have a very low false alarm rate. On the other hand, as argued by [42], an MI system with very high confidence on only a few samples can still incur in privacy violations for the members of the machine learning model's training dataset. In line with this argument, we also report the performance of our classifiers in terms of the TPR obtained for two very low FPRs: 0.1 and 0.01.

8.3.5 Implementation details

All ASR models were trained using SpeechBrain [265] and followed the default configurations and training parameters, except for the number of epochs = 60; batch size = 16; gradient accumulation factor = 2. All experiments were performed without the use of a language model. **T1** obtains a WER of 5.45% for LibriSpeech's test-clean and 15.17% for LibriSpeech's test-other partitions; **T2** obtains WERs of 10.32% and 24.70%; and **C1** obtains WERs of 6.23% and 16.77%. When computing error features, decoding was performed with a beam size of 30. Experiments using adversarial noise were built with the *robust-speech* package [226]. Membership Inference (MI) is performed using Scikit-Learn's Random Forest classifier [242] with 100 estimators for all experiments. Prediction scores correspond to the mean of the predicted class probabilities for all decision trees in the Random Forest; predictions are made with a 0.5 threshold. The results reported in Section 8.4, correspond to the average of the metrics obtained for 10 random initialisations of the classifiers. The statistical significance values regarding pairwise comparisons of these results were obtained by bootstrapping the outcomes of each random initialisation for each system 1,000 times, for a confidence interval of 95%, and averaging the resulting metrics, following the recommendations and using the codebase of Ferrer et al. [96].

8.4 Results

8.4.1 Ablation study

The results for all experiments are aggregated in Table 8.3, with Lines 1–6 corresponding to the baseline ablation study.

Line 1 shows the results regarding the performance of the error features, as used in [283], for both sample-level (left) and speaker-level MI (right). The error features correspond to a *black-box* scenario, wherein an auditor cannot access model weights or unprocessed outputs. The results obtained for sample-level MI with error features have the lowest accuracy among the considered feature sets, at approximately 70%.

On the other hand, as lines 2–3 demonstrate, allowing access to the output logits of the model and incorporating loss information enhances all success metrics compared to the black-box scenario. For

#	Shadow Model	Features	Sample				Speaker			
			Accuracy	**AUC**	**TPR$_{FPR=0.1}$**	**TPR$_{FPR=0.01}$**	**Accuracy**	**AUC**	**TPR$_{FPR=0.1}$**	**TPR$_{FPR=0.01}$**
1	**T1**	Errors	69.8 ± 0.4	76.0 ± 0.2	30.6 ± 1.1	4.6 ± 1.2	77.7 ± 0.1	83.0 ± 0.3	56.1 ± 1.0	11.8 ± 3.0
2	**T1**	Losses	86.8 ± 0.2	93.3 ± 0.1	78.9 ± 0.8	$\mathbf{24.2 \pm 4.0}$	75.9 ± 0.4	82.1 ± 0.2	53.8 ± 2.3	9.2 ± 2.9
3	**T1**	Losses + GF	87.3 ± 0.3	92.8 ± 0.1	73.8 ± 0.9	15.2 ± 2.5	$\mathbf{79.8 \pm 0.3}$	$\mathbf{84.8 \pm 0.2}$	$\mathbf{63.4 \pm 1.4}$	13.4 ± 3.5
4	**T1**	Losses + AF	$\mathbf{88.3 \pm 0.3}$	$\mathbf{94.2 \pm 0.1}$	$\mathbf{81.6 \pm 1.0}$	22.5 ± 4.1	74.9 ± 0.7	80.6 ± 0.2	50.1 ± 1.9	14.6 ± 2.5
5	**T1**	Losses + GF + AF	88.1 ± 0.2	93.9 ± 0.1	79.2 ± 1.5	17.9 ± 2.8	78.1 ± 0.4	83.5 ± 0.2	62.6 ± 1.1	14.9 ± 2.9
6	**T1**	All features	87.9 ± 0.2	93.9 ± 0.1	78.9 ± 0.4	19.5 ± 2.2	78.3 ± 0.4	83.7 ± 0.3	63.3 ± 0.8	$\mathbf{15.8 \pm 4.2}$
7	**T2**	Errors	70.1 ± 0.2	77.3 ± 0.2	33.2 ± 1.7	5.6 ± 1.5	77.1 ± 0.4	82.8 ± 0.3	52.3 ± 1.5	8.5 ± 2.4
8	**T2**	Losses	86.0 ± 0.2	92.3 ± 0.2	75.2 ± 1.4	0.0 ± 0.0	76.1 ± 0.4	81.5 ± 0.2	49.7 ± 1.7	6.2 ± 2.7
9	**T2**	Losses + GF	85.9 ± 0.2	92.2 ± 0.2	72.6 ± 0.7	1.6 ± 4.9	79.0 ± 0.3	83.9 ± 0.2	59.3 ± 1.9	7.4 ± 2.0
10	**T2**	Losses + AF	$\mathbf{87.3 \pm 0.3}$	$\mathbf{93.4 \pm 0.1}$	$\mathbf{76.0 \pm 0.6}$	$\mathbf{18.1 \pm 2.9}$	76.7 ± 0.4	81.1 ± 0.3	49.7 ± 2.4	7.3 ± 2.9
11	**T2**	Losses + GF + AF	86.6 ± 0.1	93.0 ± 0.1	75.7 ± 0.7	12.8 ± 6.6	$\mathbf{79.7 \pm 0.4}$	$\mathbf{84.2 \pm 0.3}$	59.7 ± 1.8	7.9 ± 2.3
12	**T2**	All features	86.4 ± 0.1	92.9 ± 0.1	75.7 ± 0.9	13.9 ± 7.8	79.2 ± 0.5	84.0 ± 0.3	$\mathbf{61.1 \pm 1.6}$	$\mathbf{11.5 \pm 2.8}$
13	**C1**	Errors	61.3 ± 1.6	67.7 ± 1.9	22.1 ± 1.4	2.6 ± 1.0	64.3 ± 4.7	74.8 ± 2.6	40.0 ± 4.6	$\mathbf{6.6 \pm 2.3}$
14	**C1**	Losses	57.0 ± 0.6	80.4 ± 1.3	24.2 ± 7.9	0.0 ± 0.0	$\mathbf{74.9 \pm 0.7}$	$\mathbf{78.6 \pm 0.7}$	36.2 ± 2.0	1.6 ± 0.7
15	**C1**	Losses + GF	69.1 ± 2.3	73.3 ± 1.9	22.0 ± 2.5	2.3 ± 0.7	63.1 ± 2.1	75.7 ± 1.3	$\mathbf{40.2 \pm 2.9}$	4.9 ± 1.3
16	**C1**	Losses + AF	64.6 ± 2.9	81.1 ± 2.0	35.6 ± 3.7	1.8 ± 0.7	57.7 ± 5.4	65.9 ± 6.1	31.9 ± 7.9	3.4 ± 1.9
17	**C1**	Losses + GF + AF	69.8 ± 1.7	$\mathbf{81.3 \pm 1.9}$	$\mathbf{36.8 \pm 1.3}$	$\mathbf{6.0 \pm 1.9}$	61.8 ± 3.6	65.0 ± 5.8	32.6 ± 7.3	5.7 ± 2.2
18	**C1**	All features	$\mathbf{73.2 \pm 1.8}$	80.7 ± 1.5	35.7 ± 1.6	5.5 ± 0.9	57.5 ± 8.1	60.6 ± 10.8	18.3 ± 9.6	3.1 ± 2.5

Table 8.3: Results for MI performance for shadow models (**T1**, **T2**, **C1**), for target model **T1**, per feature set at both sample and speaker-level.

sample-level MI, using only the losses, results improve by over 15% for both Acc. and AUC when compared to the error features, a result that is statistically significant.

When combining the loss features with each of the perturbations, lines 3–4, and their combination, line 5, we observe that all feature sets bring a similar level of improvement, reaching values close to 88% and 94% for accuracy and AUC, respectively, at the sample level. While the Gaussian features (**GF**) are cheaper to compute and can be computed with *grey-box* access, the features based on adversarial samples (**AF**) require *white-box* access to the model to perform back-propagation. Though the adversarial features slightly outperform the Gaussian-based features, this difference is not statistically significant, with the loss and Gaussian-based features providing a very close performance. This can be advantageous when computational resources are limited and generating adversarial perturbations is not feasible.

Line 6 additionally shows that combining the loss and perturbed loss features with the error features yields no improvement, supporting the claim that the loss-based features already carry most of the relevant information for this task on their own.

When considering speaker-level MI, the results are relatively different. In this case, the loss features alone slightly underperform, though not statistically significantly, when compared to the error-based features that achieve an accuracy $\sim 78\%$. The perturbation-based methods are only able to improve upon these results by a margin of $\sim$2%, a difference that is statistically significant. A possible reason for this contrast is the fact that while ASR models are trained to minimise the loss of specific samples, the model's training process does not explicitly account for speakers. Consequently, loss and error features likely carry similar information regarding the membership of specific speakers.

One might question why the error-based features provide much better results for speaker-level MI than for sample-level MI. We hypothesise that this is due to how we set up our MI dataset. At the sample level, all utterances belong to speakers that are present in the model's training set, making them more challenging to distinguish. In this sense, the results we obtain in this work for these features are similar to those obtained by [283]. This is in contrast to other works, where this distinction is not made and where negative samples always correspond to unseen speakers, thus simplifying the task and achieving better performances [172].

8.4.2 Shadow model performance

Lines 7–18 provide the results for the more realistic scenarios where the shadow models are not based on the same dataset as the target model (models **T2** and **C1**). For the experiments performed with the transformer shadow model (**T2**), lines 7–12, the results follow a similar trend to the above, with the combination of the loss- and perturbation-based features providing the best overall sample-level results, achieving an accuracy close to 87%. At the speaker level, for lines 7-8 we observe a similar trend to

lines 1-2, with the error features providing slightly better results, though not significantly so. In this case, the best results in terms of Acc. and AUC, are achieved with the combination of all perturbed loss features, having statistical significance for Acc when compared to the error-based feature set. However, when the shadow model is based on a different architecture (model **C1**, a conformer model instead of a transformer model), the accuracy for sample-level MI deteriorates to roughly 70%. Nevertheless, in terms of the AUC, the combination of all perturbed losses (line 17) provides a statistically significant improvement of nearly 10% when compared to the error-based features. At the speaker level, the best-performing feature set is the set of loss features, with statistical significance for Acc., whereas the combination of all features achieves the lowest results for most metrics, being closely followed by the combination of the perturbed features. The fact that the perturbed features do not improve results, in this case, may be caused by the large differences between architectures, such that the behaviour of the decision boundary around unseen data points is not comparable between models, making these perturbations ill-adjusted to this case.

8.4.3 Performance at low FPR operating points

In lines 1–6, at the sample level, for a maximum FPR of 10%, most of the proposed loss- and perturbation-based features reach values above 75% TPR, more than 40% above error features, with this difference being statistically significant. Similarly, for the very low value of 1% FPR, at the sample level, the proposed features, namely the **AF**, significantly outperform error-based features, although with much lower absolute TPRs. A similar behaviour is observed for shadow model **T2** in lines 7–12. At the speaker level, the improvement of the loss features over the error features is much smaller. In this case, the best results for **T1** and **T2** as shadow models, in lines 1–12, correspond to those obtained with the **GF** (for **T1**) or the combination of all of the considered features (for **T1** and **T2**), depending on the value of the FPR. On the other hand, for **C1**, the best values for each metric are observed for **GF** and error features. However, these differences are not statistically significant, and further study is necessary to assess the impact of each feature set in very low FPR scenarios, at the speaker level. Nevertheless, the results at the sample level show that low-FPR operating points, which are particularly relevant to MI auditing, also benefit from loss- and perturbation-based features.

8.5 Summary

In the work presented in this chapter, we explored the use of loss-based features, together with Gaussian and adversarial perturbations, to perform membership inference in ASR models. This work was framed as an auditing setting, as a way to determine if users' speech data was used during model training without their consent. To assess the proposed features, we conducted several experiments, considering

various levels of access to model outputs, knowledge of the distribution of the target model's training data, and knowledge of the model's architecture, and performed sample- and speaker-level experiments. At the sample level, the proposed features greatly outperform previously proposed error features. This occurs even for very low FPRs that take into account the importance of wrong decisions in model auditing. At the speaker level, the proposed features obtain similar or improved results when compared to the original error features, depending on the feature configuration. Overall, our results show that easy and computationally cheap features improve MI performance in ASR, particularly for auditing scenarios.

This work is a first step in the exploration of the use of these features for MI in ASR, and there are many possible avenues for further research. For instance, it remains to be understood if loss-based features can be applied to shadow models trained with different loss functions. Similarly, shadow and target models with very different architectures may have very different loss distributions, making it important to explore techniques that minimise this mismatch. Exploring the impact of the differences in the recording conditions and speaking styles between the shadow and target model's training datasets would also be of interest, as this would emulate realistic MI scenarios. It would also be interesting to explore methods to improve TPR scores obtained at very low FPRs, particularly if MI should be used for model auditing.

From the perspective of fairness, we also consider that future work should explore how well our methodology performs when tested on different sub-groups (e.g., different sexes, ages or accents) of the population. In addition, while differential privacy might serve as a theoretical protection against membership inference [119], it also limits the possibility of auditing a model's training data. As such, it would be worth exploring the trade-off between what can be considered conflicting goals.

9

Conclusions

The work conducted in this thesis addressed the problem of privacy in remote speech processing, a scenario that is becoming ever more common with the progress of machine learning and speech processing technologies. The main goal of this thesis was to focus on the study and development of methods based on cryptographic techniques and privacy-oriented speech manipulation methods while providing insights into the usability of these techniques, the trade-offs they require and their limitations. Accordingly, Chapters 3–5 focus on the application of the cryptographic techniques introduced in Chapter 2, to remote speech processing applications, namely the privacy-preserving implementation of an SVM for speech-affecting disease detection, the private extraction of speaker embeddings for ASV, and the privacy-preserving implementation of an ASD pipeline. Chapter 6 presents a method for generating imperceptible adversarial examples against speaker identification. Although not directly related to privacy, this work later guided the development of the privacy-oriented speech manipulation method proposed in Chapter 7, for the removal of sex and age information from speaker representations. Differently, but complementary to the work presented in prior chapters, the work developed in Chapter 8 relates to the privacy of the speakers included in the training dataset of a speech recognition model, introducing a method to perform membership inference for ASR model auditing.
This chapter completes this thesis, with Section 9.1 summarising the work presented in the preceding chapters, and Section 9.2 discussing its advantages and disadvantages, and proposing possible future research directions that may build upon the work contained in this document.

9.1 Thesis summary

In Chapter 1, we defined the main working scenario for this thesis: a user and a service provider want to interact to apply the service provider's machine learning model over the user's speech data, such that the service provider does not learn anything about the user, whose data is kept *private*, and the user learns as little as possible about the service provider's model. In this scenario, the service provider is seen as the main attacker, who will try to gain as much information about the user as possible. The main goal of this thesis was, therefore, set to the development of methods that ensure user privacy in this setting, with a main focus on methods based on cryptographic processing and privacy-oriented speech manipulation.
In Chapter 3, we proposed a method for the privacy-preserving detection of two speech-affecting diseases, Parkinson's disease and Obstructive Sleep Apnea. As argued in Chapter 1, speech can reveal very sensitive information about a speaker, including numerous speaker traits and states, making it necessary to protect speech signals or representations thereof. In pathological speech detection and assessment tasks, this need becomes even more important, as the possibility of an attacker (in the scenario considered in this thesis, the service provider) gaining sensitive information about a speaker is no longer hypothetical: the task is to infer this sensitive information. The work of Chapter 3 tackled

this issue using a combination of three cryptographic techniques – HE, SMC and SMH – to implement an SVM classifier with the RBF kernel. These techniques allow us to protect not only the speech features, but also the corresponding classification result. This method is shown to attain minimal degradation in terms of model performance at the moderate computational cost of $\sim 650ms$, for the online phase of the computation, per prediction, using 200MB of bandwidth per party in the computation. Given the importance of health-related speech tasks, we consider these costs to be an acceptable and usable trade-off for the added level of privacy.

Chapter 4 focused on the extraction of speaker representations, specifically *x-vector* speaker embeddings, in the context of privacy-preserving ASV. While most works on privacy for ASV have focused on the privacy of the verification step, in this chapter, we argued that the speaker embedding extraction model is likely the most valuable component of an ASV pipeline and therefore should be protected, and not shared with the user. To this end, and to protect the user's data, we showed how an *x-vector* extractor network could be implemented using SMC. Comparing different SMC protocols, each presenting different levels of security, we showed that, in a three-party honest-majority, semi-honest SMC setting, we were able to extract a speaker embedding with a computational cost of $\sim 11s$, using $\sim$ 133MB of bandwidth, with negligible performance degradation. We also found that it was possible to operate in a stronger security setting, using a four-party honest-majority protocol with security against one malicious adversary – for an additional computational cost of 7 seconds and $\sim$ 230MB. However, our results showed that it is infeasible to implement this network in two-party settings, with computational costs going over $2h$ ($>$ 1TB of bandwidth) in the semi-honest setting, and $41h$ ($>$ 20TB of bandwidth) in the malicious setting.

In Chapter 5, we built on the work developed in Chapter 4 and proposed a method to perform privacy-preserving ASD, using a combination of SMC and SMH. Concretely, we leveraged SMC to first extract speaker embeddings, as described in Chapter 4, and to apply SMH to the extracted speaker embeddings. Since both steps are performed with SMC, no party has access to intermediate results and both the user's data and the service provider's model are kept private. Moreover, the use of the SMH transformation (with an SMH key generated by the user, which was kept private) guaranteed that the resulting SMH vectors could be shared with the service provider, without it being able to recover the original *x-vectors*. This allowed the service provider to perform the ASD clustering step, completing the ASD pipeline. Using the best-performing security settings observed in Chapter 4, a three-party honest majority semi-honest and a four-party honest majority with malicious security against one party, the proposed system took 5 (three-party setting) and 7 minutes (four-party setting) to perform diarization over 4 minutes of speech, representing real-time factors of 1.1 and 1.6, respectively, and using $\sim$ 6.5 and 19.5GB of bandwidth. On the other hand, we observed a total performance degradation close to 9% and 20% in terms of DER and JER, respectively. This degradation came from the use of a bare-bones

system – which minimised the system's computational cost – as well as the introduction of the SMH transformation. Nevertheless, our system was evaluated with a particularly challenging mixed-domain dataset and, as shown in the chapter's final experiments, it is possible to improve results when considering individual domains. Even though the work presented in this chapter involves high computational and communication costs, it is – to the best of our knowledge – the first implementation of a privacy-preserving ASD pipeline using cryptographic techniques.

In Chapter 6, we focused on a research direction that was orthogonal to the prior chapters: the creation of adversarial examples against a speaker identification model. In particular, we showed that it is possible to imperceptibly perturb speech (PESQ values close to ~ 4.3) while being able to fool a speaker identification system (obtaining success rates very close to 100%), for both untargeted and targeted attacks. In this chapter, we argued that this approach could potentially be used to protect speakers from the automatic collection of speech data. This would require that our method be extended to fool speaker verification systems (i.e. systems that authenticate specific speakers based on pre-trained models). In addition, the proposed method would need to generate adversarial examples that are robust to over-the-air-play and compression algorithms, and most importantly to transfer to other classifiers. However, robustness experiments showed us that, as could be anticipated, the attack's success rate dropped by $\sim 60\%$ when either room impulse responses (which mimic over-the-air-play) or MP3 compression were applied to the adversarial speech sample, indicating that the proposed system was not yet able to provide privacy protections.

Notwithstanding the limitations of this adversarial approach, we acquired a new perspective on the potential of speech manipulation methods for privacy through it. This experience, together with the recent advances in this topic, and the limitations of cryptographic approaches, namely the high computational costs, led us to consider privacy-oriented speech manipulation methods as an alternative research direction for privacy in speech. As stated in Section 1.3.2, as opposed to cryptographic-based processing methods, speech manipulation methods are independent of the complexity of downstream tasks. On the one hand, cryptographic systems can be seen as a "one-size fits all" solution, i.e., each cryptographic method has a set of characteristics independent of the task to which they are being applied, to which the privacy-preserving implementation needs to be adapted. Contrarily, privacy-oriented speech manipulation methods can be made general enough to be used in any downstream task, without requiring adaptation to the target task.

The work presented in Chapter 7 represents a step in this direction. Its main objective was to provide a controllable mechanism that enforced privacy with regard to speaker attributes (i.e., age and sex), by allowing the removal or manipulation of these attributes. The proposed method was based on a combination of a VQ-VAE, an adversarial classifier and a novel mutual information loss. The experiments of this chapter showed that the method achieved attribute classification results at chance

level for an ignorant attacker, and close to chance level, $\sim 57\%$ UAR and ~ 0.1 CCC, for sex and age, respectively, when considering an informed attacker. Attribute manipulation experiments also showed how the proposed method is able to modify the attribute information contained within the speaker embeddings, with pre-trained classifiers achieving $\sim 90\%$ UAR and ~ 0.91 CCC, for sex and age, respectively when evaluated with regard to the target attributes. The model was additionally evaluated in cross-domain scenarios. For ignorant attackers, our results showed that the model was able to generalise, having a similar performance for in-domain and out-of-domain data. However, for informed attackers, a substantial performance degradation was observed, particularly when both the attacker's attribute classifier and the test data were out-of-domain. Nonetheless, it was also shown that retraining the model with the out-of-domain data, or a mixture of in- and out-of-domain data helped the model improve its performance for the out-of-domain data. In terms of utility, measured in ASV performance, it was shown that for the sex information removal model, the degradation was limited to 0.6% EER, while for the age removal model, the degradation rose to 3.4% EER, meaning that the proposed approach was able to manipulate the intended attributes without having a strong degradation effect on other subsets of speaker discriminative information. The difference between the two results was attributed to the difference in the amount of data used to train each model, with the sex information removal model having been trained with eight times the amount of data used to train the same model for age. These results showed us that privacy-oriented speech manipulation methods such as the one proposed in this chapter, are a promising alternative to cryptographic constructions.

Chapter 8 contains work that can be considered complementary to the above, but which does not fall directly within the main goals set for this thesis. This chapter focused on exploring membership inference – the task of determining whether a given sample or speaker has been used in model training – for ASR model auditing. Specifically, the goal of this chapter was to understand how membership inference can be used as a tool to audit ASR models with regard to the potentially unauthorised use of data, as well as to evaluate the level of privacy of the training data of these models. To do so, different sets of features were considered, including features related to the transcription errors of the ASR model (the most commonly used feature set in the literature), and features related to the model's training loss functions – which had been previously unexplored in the literature. Our work also investigated variations of these features which leveraged inputs perturbed with Gaussian noise and adversarial perturbations. Using a fixed target model – a transformer model – for all experiments, the proposed features were tested for three different proxy – or shadow – model scenarios: (1) the shadow and target models corresponded to the same model; (2) the shadow model had the same architecture as the target model, but a different training dataset, drawn from the same domain as the target model's training data; (3) the shadow model had a different architecture and training dataset (equal to the one used in (2)) than the target model. Moreover, we considered two types of membership inference, at sample and

speaker levels. For the first two scenarios, for sample-level membership inference, the loss-based feature sets consistently outperformed the error-based features, with the best-performing feature sets achieving results close to 88% and $\sim$ 94% in terms of accuracy and AUC, corresponding to an improvement of more than 15% when compared to error features. For the third scenario, this performance dropped, but these results still improved over the error-based features. In terms of speaker-level membership inference, the best-performing feature sets also corresponded to loss-based features, which, for the first scenario achieved results close to $\sim$ 80% accuracy and $\sim$ 85% AUC, with similar results having been observed for the second and third scenarios.

The proposed feature sets were also evaluated using the TPR achieved for very low FPRs (i.e., FPR equal to 10% and 1%). These metrics were used to emulate a setting where the cost of *false alarms* is much higher than that of *missed detection*. For the strictest setting (FPR=1%), we observed that, at the sample level, the best-performing feature sets all corresponded to loss features, achieving a maximum TPR of 24%, for the first scenario. At the speaker level, the best-performing feature sets for all scenarios corresponded to loss features or the combination of the loss and error features, except when the shadow model has a different architecture, in which case the best results are obtained with error features.

While not focused on a method to provide privacy in remote speech processing, this chapter highlighted the fact that MI has the potential to be used as a tool for auditing speech-based systems, and consequently as a way to enforce user's privacy rights. On the other hand, these experiments also highlighted the potential privacy vulnerabilities that arise from distributing speech-based machine learning models, a concern that is exacerbated by the knowledge that membership information can be used to help perform training data extraction attacks [42, 43].

9.2 Future directions

The work conducted throughout this thesis focused on two main approaches to privacy in remote speech processing: cryptographic approaches and privacy-oriented speech manipulation.

The results of Chapters 3–5 show that cryptographic approaches entail computational and communication costs that may still be considered too high for real-world applications, particularly when task-related pipelines go beyond simple classifiers. Nonetheless, as mentioned in Chapter 1, cryptographic approaches are particularly suited for tasks where it is hard to disentangle task-related and private information (i.e., speech analysis tasks, speaker recognition, speaker diarization) and in contexts that demand strong privacy guarantees. In addition, it is important to state that the computational cost of cryptographic tools has been continuously decreasing in the last few years and is expected to continue decreasing as cryptographic primitives become more efficient and take advantage of more powerful or dedicated hardware (e.g., GPUs [200], [74]). Such improvements seem to be the

main factor affecting the performance of cryptographic-based privacy-preserving methods for machine-learning-based speech processing. However, these improvements demand expert knowledge in cryptography. As such, the successful development of cryptographic-based privacy-preserving speech processing seems to lie in stronger collaborations and knowledge exchange between speech and cryptography researchers, where both sides strive towards improved compatibility between speech-processing pipelines and cryptographic techniques. Nevertheless, from the point of view of speech processing research, the study of methods that improve the efficiency of speech processing methods in combination with cryptographic techniques is an interesting area for future research. This can include network quantisation [68, 184] and pruning [270], as well as the adaptation of neural networks to the limitations of cryptographic processing, e.g., limiting the number of non-linear activation functions would have a positive impact on the efficiency of cryptographic-based methods. In addition, pushing the limits of cryptographic protocols by implementing even larger and more complex models, i.e., ASR models, remains an interesting and open challenge, that could potentially provide insights for more efficient implementations of simpler classifiers.

Differently, the attribute manipulation models proposed in Chapter 7 have comparatively low computational costs and entail only small additive costs if added to an ASV pipeline, the use case of this chapter, but also to any downstream task that uses speaker representations. Moreover, these methods require no changes to the downstream tasks. For an example use case of these models, we can consider a user who extracts a speaker representation from a recording of their voice in a local device. The user would apply an attribute manipulation model over the speaker representation and obtain a representation of their voice that can be considered private with regard to some attribute. The user could then share this representation with a service provider, knowing that the service provider will not be able to obtain the information that was removed. Given that the use of this type of model is only dependent on the user, and that it does not need to be taken into account by service providers, these models therefore shift the responsibility and decision-making from the service provider to users, who become able to decide whether to disclose less information, potentially losing some utility, in exchange of more privacy, and vice-versa. The combination of several of these "filters" would be a very interesting extension of the work of Chapter 7, as it would provide users with more fine-grained control over the information disclosed. Moreover, using this method as a mechanism to control voice conversion or text-to-speech synthesis is also of interest. As evidenced by [87, 220], manipulating speaker attributes can contribute to speech anonymisation efforts. Exploring the use of this method for this purpose is therefore an important avenue for future work.

The development of methods that hide speaker attributes also raises the question of which attributes are more related to speaker identity, or which can considered more sensitive. One could ask if hiding age provides more privacy than hiding the speaker's sex, or if it would be more important to hide other

speaker traits. In a real-world scenario, it would be important to inform the user of not only the utility degradation introduced by the removal of certain attributes, but also of the possible privacy protections that can be achieved by hiding each specific attribute. We consider that this would also be an interesting line of future research.

Notwithstanding their potential, privacy-oriented speech manipulation methods still pose some disadvantages with regard to cryptographic constructions. In the use case described above, the service provider's speaker representation extraction model needs to be shared with the user, which means that the service provider can potentially lose some of its value and put at risk the privacy of its training data (e.g., consider membership inference or extraction attacks). The result of the target task is also revealed to the service provider, as opposed to what happens with cryptographic constructions. In addition, cryptographic methods provide confidentiality with theoretical guarantees, whereas speech manipulation methods often do not remove all sensitive information, and only provide empirical privacy guarantees. Therefore, the choice between one avenue or the other will require taking the specific use case and context into account, in order to select the most appropriate solution, with all of its advantages and drawbacks.

Finally, one aspect that neither approach is able to take into account is the privacy of the service provider's model's training data. As shown in Chapter 8, it is possible to infer the membership of individual samples, or speakers, in model training datasets with relatively high levels of accuracy. This can be used to enforce user privacy, through model auditing, and to evaluate the privacy conferred to members of machine-learning models' training datasets. Moreover, MI can also be used as a first step for machine unlearning (i.e., removing information regarding a specific training sample from an ML model), an important task in the lifetime of ML models, which can be used to uphold a user's right to be forgotten. However, this type of system can also be used as an attack. In a hypothetical situation, where the full training dataset was collected in a hospital, determining the membership of a speaker in this dataset discloses the potentially sensitive information that the individual is being followed in that specific hospital. Moreover, as stated in the previous section, membership information can be used to conduct stronger extraction attacks, that recover training data samples. As large speech language models become more common, and the risks of these vulnerabilities grow, we consider that research in these under-explored topics for speech should be a paramount goal of future work.

This thesis began with a quote from a character in Isaac Asimov's Foundation's Edge, first published in 1982, where the character states "It seems to me (...) that the advance of civilization is nothing but an exercise of the limiting of privacy." One cannot help but reflect on the truth of this sentence, particularly when considering the technological developments of the past century, or even just the past

thirty years. The overwhelming use of communication technologies, social media applications and other online services and applications, many of which leverage machine learning models, translates into a virtually infinite amount of information being shared at every moment, with a large share of this information consisting of personal information. Paradoxically, privacy is a right that was achieved by the development of civilisation. Now, through its technological advancements, it seems to be withering. However, as individuals, we continuously choose to exercise privacy. As a society, we choose to protect it through legislation, electing it as a human right, showing its fundamental role in our democratic society. Contrary to what some believe, privacy has not disappeared. It faces challenges, as do most aspects of society when faced with changes, but the collective effort put into protecting it shows that it will not be renounced easily. Research towards privacy in speech technologies plays an important part in the overall effort to protect the right to privacy, being fundamental for three reasons: the sheer amount of information speech conveys about each individual; the ubiquitousness of speech and its growing role as a means of human-machine interaction, and as a target of information extraction applications; the lack of public awareness to the sensitive nature of the information conveyed by speech. This thesis is a very small part of this important effort, but one which we hope can positively contribute to it.

Bibliography

[1] Abdullah, H., Warren, K., Bindschaedler, V., Papernot, N., and Traynor, P. SoK: The Faults in our ASRs: An Overview of Attacks against Automatic Speech Recognition and Speaker Identification Systems. In *IEEE Symposium on Security and Privacy (IEEE S&P)* (2021).

[2] Abramowitz, M., Stegun, I. A., and Romer, R. H. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. American Association of Physics Teachers, 1988.

[3] Afonja, T., Bourtoule, L., Chandrasekaran, V., Oore, S., and Papernot, N. Generative extraction of audio classifiers for speaker identification. *arXiv preprint arXiv:2207.12816* (2022).

[4] Ahmed, S., Chowdhury, A. R., Fawaz, K., and Ramanathan, P. Preech: A System for Privacy-Preserving Speech Transcription. In *29th USENIX Security Symposium (USENIX Security 20)* (Aug. 2020), USENIX Association, pp. 2703–2720.

[5] Akimoto, Y., Fukuchi, K., Akimoto, Y., and Sakuma, J. Privformer: Privacy-preserving transformer with mpc. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)* (2023), IEEE, pp. 392–410.

[6] Ali, H. S., ul Hassan, F., Latif, S., Manzoor, H. U., and Qadir, J. Privacy enhanced speech emotion communication using deep learning aided edge computing. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops)* (2021), IEEE, pp. 1–5.

[7] Alkim, E., Ducas, L., Pöppelmann, T., and Schwabe, P. Post-quantum key exchange: a new hope. In *Proceedings of the 25th USENIX Conference on Security Symposium* (2016), pp. 327–343.

[8] Aloufi, R., Haddadi, H., and Boyle, D. Emotion filtering at the edge. In *Proceedings of the 1st Workshop on Machine Learning on Edge in Sensor Systems* (New York, NY, USA, 2019), SenSys-ML 2019, Association for Computing Machinery, p. 1–6.

[9] Aloufi, R., Haddadi, H., and Boyle, D. Privacy-preserving voice analysis via disentangled representations. In *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop* (2020), CCSW'20, p. 1–14.

[10] Aloufi, R., Haddadi, H., and Boyle, D. Paralinguistic privacy protection at the edge. *ACM Transactions on Privacy and Security 26*, 2 (2023).

[11] Aly, A., and Smart, N. P. Benchmarking privacy preserving scientific operations. In *International Conference on Applied Cryptography and Network Security* (2019), Springer, pp. 509–529.

[12] Anguera, X., Bozonnet, S., Evans, N., Fredouille, C., Friedland, G., and Vinyals, O. Speaker diarization: A review of recent research. *IEEE Transactions on audio, speech, and language processing 20*, 2 (2012), 356–370.

[13] Araki, T., Furukawa, J., Lindell, Y., Nof, A., and Ohara, K. High-throughput semi-honest secure three-party computation with an honest majority. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016), pp. 805–817.

[14] Armknecht, F., Boyd, C., Carr, C., Gjøsteen, K., Jäschke, A., Reuter, C. A., and Strand, M. A guide to fully homomorphic encryption. *IACR Cryptology ePrint Archive 2015* (2015), 1192.

[15] Bäckström, T. Privacy in speech technology. *Proceedings of the IEEE* (2023).

[16] Baevski, A., Schneider, S., and Auli, M. vq-wav2vec: Self-supervised learning of discrete speech representations. In *8th International Conference on Learning Representations, ICLR, 2020* (2020), pp. –.

[17] Baevski, A., Zhou, Y., Mohamed, A., and Auli, M. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems 33* (2020), 12449–12460.

[18] Bahmaninezhad, F., Zhang, C., and Hansen, J. Convolutional Neural Network Based Speaker De-Identification. In *Proc. The Speaker and Language Recognition Workshop (Odyssey 2018)* (2018), pp. 255–260.

[19] Bai, Z., and Zhang, X.-L. Speaker recognition based on deep learning: An overview. *Neural Networks* (2021).

[20] Baird, A., Cummins, N., Schnieder, S., Krajewski, J., and Schuller, B. W. An Evaluation of the Effect of Anxiety on Speech — Computational Prediction of Anxiety from Sustained Vowels. In *Proc. Interspeech 2020* (2020), pp. 4951–4955.

[21] Barnett, A., Santokhi, J., Simpson, M., Smart, N. P., Stainton-Bygrave, C., Vivek, S., and Waller, A. Image Classification using non-linear Support Vector Machines on Encrypted Data. *IACR Cryptology ePrint Archive 2017* (2017), 857.

[22] Baum, C., Cozzo, D., and Smart, N. P. Using TopGear in overdrive: a more efficient ZKPoK for SPDZ. In *International Conference on Selected Areas in Cryptography* (2019), Springer, pp. 274–302.

[23] Beaver, D. Efficient multiparty protocols using circuit randomization. In *Advances in Cryptology—CRYPTO'91: Proceedings 11* (1992), Springer, pp. 420–432.

[24] Beaver, D. Precomputing oblivious transfer. In *Annual International Cryptology Conference* (1995), Springer, pp. 97–109.

[25] Ben-Or, M., Goldwasser, S., and Wigderson, A. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *20th annual ACM symposium on Theory of computing* (1988), ACM, pp. 1–10.

[26] Benaroya, L., Obin, N., and Roebel, A. Manipulating voice attributes by adversarial learning of structured disentangled representations. *Entropy 25*, 2 (2023).

[27] Bittner, K., De Cock, M., and Dowsley, R. Private emotion recognition with secure multiparty computation, 2021.

[28] Boemer, F., Lao, Y., Cammarota, R., and Wierzynski, C. NGraph-HE: A Graph Compiler for Deep Learning on Homomorphically Encrypted Data. In *Proceedings of the 16th ACM International Conference on Computing Frontiers* (New York, NY, USA, 2019), CF '19, Association for Computing Machinery, p. 3–13.

[29] Bogdanov, D., Laur, S., and Willemson, J. Sharemind: A framework for fast privacy-preserving computations. In *Computer Security-ESORICS 2008: 13th European Symposium on Research in Computer Security, 2008. Proceedings 13* (2008), Springer, pp. 192–206.

[30] Bok, S. *Secrets: On the Ethics of Concealment and Revelation*. Oxford University Press, New York, 1982.

[31] Bost, R., Popa, R. A., Tu, S., and Goldwasser, S. Machine learning classification over encrypted data. In *NDSS* (2015), vol. 4324, p. 4325.

[32] Botelho, M. C., Trancoso, I., Abad, A., and Paiva, T. Speech as a biomarker for obstructive sleep apnea detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), pp. 5851–5855.

[33] Boufounos, P., and Rane, S. Secure Binary Embeddings for Privacy Preserving Nearest Neighbors. In *2011 IEEE International Workshop on Information Forensics and Security* (2011), IEEE, pp. 1–6.

[34] Boufounos, P. T. Universal rate-efficient scalar quantization. *IEEE Transactions on Information Theory 58*, 3 (2011), 1861–1872.

[35] Brakerski, Z. Fully Homomorphic Encryption without modulus Switching from classical GapSVP. In *Annual Cryptology Conference* (2012), Springer, pp. 868–886.

[36] Brakerski, Z., Gentry, C., and Vaikuntanathan, V. (Leveled) Fully Homomorphic Encryption without Bootstrapping. *ACM Transactions on Computation Theory (TOCT) 6*, 3 (2014), 1–36.

[37] Brasser, F., Frassetto, T., Riedhammer, K., Sadeghi, A.-R., Schneider, T., and Weinert, C. VoiceGuard: Secure and Private Speech Processing. In *Proc. Interspeech 2018* (2018), pp. 1303–1307.

[38] Bringer, J., Chabanne, H., Favre, M., Patey, A., Schneider, T., and Zohner, M. GSHADE: faster privacy-preserving distance computation and biometric identification. In *2nd ACM workshop on Information hiding and multimedia security* (2014), pp. 187–198.

[39] Bringer, J., El Omri, O., Morel, C., and Chabanne, H. Boosting GSHADE capabilities: New applications and security in malicious setting. In *21st ACM on Symposium on Access Control Models and Technologies* (2016), pp. 203–214.

[40] California Civil Code, S. o. C. The California Consumer Privacy Act (CCPA), 2018.

[41] Campbell, W. M., Sturim, D. E., and Reynolds, D. A. Support Vector Machines using GMM Supervectors for Speaker Verification. *IEEE Signal Processing Letters 13*, 5 (2006), 308–311.

[42] Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., and Tramer, F. Membership Inference Attacks from First Principles. In *2022 IEEE Symposium on Security and Privacy (SP)* (2022), IEEE, pp. 1897–1914.

[43] Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)* (2021), pp. 2633–2650.

[44] Carlini, N., and Wagner, D. Towards evaluating the robustness of neural networks. In *Proc. of the IEEE Symposium on Security and Privacy* (San Jose, California, USA, May 2017).

[45] Catrina, O., and De Hoogh, S. Improved primitives for secure multiparty integer computation. In *Security and Cryptography for Networks: 7th International Conference, SCN 2010, Amalfi, Italy, September 13-15, 2010. Proceedings 7* (2010), Springer, pp. 182–199.

[46] Catrina, O., and De Hoogh, S. Secure multiparty linear programming using fixed-point arithmetic. In *Proc. of Computer Security–ESORICS 2010: 15th European Symposium on Research in Computer Security, 2010.* (2010), Springer, pp. 134–150.

[47] Chabanne, H., de Wargny, A., Milgram, J., and al., e. Privacy-Preserving Classification on Deep Neural Network. *IACR Cryptology ePrint Archive* (2017), 35.

[48] Champion, P., Thebaud, T., Le Lan, G., Larcher, A., and Jouvet, D. On the invertibility of a voice privacy system using embedding alignment. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (2021), pp. 191–197.

[49] Chandrasekaran, V., Chaudhuri, K., Giacomelli, I., Jha, S., and Yan, S. Exploring connections between active learning and model extraction. In *29th USENIX Security Symposium (USENIX Security 20)* (2020), pp. 1309–1326.

[50] Chen, G., Zhang, Y., and Song, F. SLMIA-SR: Speaker-Level Membership Inference Attacks against Speaker Recognition Systems. In *31st Network and Distributed System Security Symposium (NDSS)* (2024).

[51] Chen, H., Chillotti, I., and Song, Y. Improved bootstrapping for approximate homomorphic encryption. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (2019), Springer, pp. 34–54.

[52] Cheng, N., Önen, M., Mitrokotsa, A., Chouchane, O., Todisco, M., and Ibarrondo, A. Nomadic: Normalising maliciously-secure distance with cosine similarity for two-party biometric authentication. Cryptology ePrint Archive, Paper 2023/1684, 2023.

[53] Cheng, P., Hao, W., Dai, S., Liu, J., Gan, Z., and Carin, L. Club: a contrastive log-ratio upper bound of mutual information. In *Proceedings of the 37th International Conference on Machine Learning* (2020), pp. 1779–1788.

[54] Cheon, J. H., Kim, A., Kim, M., and Song, Y. Homomorphic encryption for arithmetic of approximate numbers. In *International Conference on the Theory and Application of Cryptology and Information Security* (2017), Springer, pp. 409–437.

[55] Chillotti, I., Gama, N., Georgieva, M., and Izabachène, M. Faster packed homomorphic operations and efficient circuit bootstrapping for tfhe. In *International Conference on the Theory and Application of Cryptology and Information Security* (2017), Springer, pp. 377–408.

[56] Chillotti, I., Gama, N., Georgieva, M., and Izabachène, M. Tfhe: fast fully homomorphic encryption over the torus. *Journal of Cryptology 33*, 1 (2020), 34–91.

[57] Choquette-Choo, C. A., Tramer, F., Carlini, N., and Papernot, N. Label-only Membership Inference Attacks. In *International conference on machine learning* (2021), PMLR, pp. 1964–1974.

[58] Chorowski, J., Weiss, R. J., Bengio, S., and Van Den Oord, A. Unsupervised speech representation learning using wavenet autoencoders. *IEEE/ACM transactions on audio, speech, and language processing 27*, 12 (2019), 2041–2053.

[59] Chou, T., and Orlandi, C. The simplest protocol for oblivious transfer. In *4th International Conference on Progress in Cryptology* (2015), vol. 9230, pp. 40–58.

[60] Chouchane, O., Panariello, M., Zari, O., Kerenciler, I., Chihaoui, I., Todisco, M., and Önen, M. Differentially private adversarial auto-encoder to protect gender in voice biometrics. In *Proceedings of the 2023 ACM Workshop on Information Hiding and Multimedia Security* (2023), IH&MMSec '23, p. 127–132.

[61] Chung, J. S., Nagrani, A., and Zisserman, A. Voxceleb2: Deep speaker recognition. In *Proc. Interspeech 2018* (2018).

[62] Cooper, E., Lai, C.-I., Yasuda, Y., Fang, F., Wang, X., Chen, N., and Yamagishi, J. Zero-shot multi-speaker text-to-speech with state-of-the-art neural speaker embeddings. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP* (2020), IEEE, pp. 6184–6188.

[63] Correia, J., Teixeira, F., Botelho, C., Trancoso, I., and Raj, B. The in-the-wild speech medical corpus. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), pp. 6973–6977.

[64] Costan, V., and Devadas, S. Intel sgx explained. *IACR Cryptology ePrint Archive 2016*, 086 (2016), 1–118.

[65] Cramer, R., Damgård, I., Escudero, D., Scholl, P., and Xing, C. SPD$\mathbb{Z}_{2^k}$: Efficient MPC mod 2^k for Dishonest Majority. *IACR Cryptol. ePrint Arch.* (2018), 482.

[66] Cummins, N., Baird, A., and Schuller, B. W. Speech analysis for health: Current state-of-the-art and the increasing impact of deep learning. *Methods 151* (2018), 41–54.

[67] Cummins, N., Scherer, S., Krajewski, J., Schnieder, S., Epps, J., and Quatieri, T. F. A review of depression and suicide risk assessment using speech analysis. *Speech communication 71* (2015), 10–49.

[68] Dalskov, A., Escudero, D., and Keller, M. Secure Evaluation of Quantized Neural Networks. *Proc. of Privacy Enhancing Technologies 4* (2020), 355–375.

[69] Dalskov, A., Escudero, D., and Keller, M. Fantastic four: Honest-majority four-party secure computation with malicious security. In *30th {USENIX} Security Symposium ({USENIX} Security 21)* (2021).

[70] Damgård, I., Keller, M., Larraia, E., Pastro, V., Scholl, P., and Smart, N. P. Practical covertly secure MPC for dishonest majority – or: breaking the SPDZ limits. In *European Symposium on Research in Computer Security* (2013), Springer, pp. 1–18.

[71] Das, R. K., Tian, X., Kinnunen, T., and Li, H. The Attacker's Perspective on Automatic Speaker Verification: An Overview. In *Proc. Interspeech 2020* (2020), pp. 4213–4217.

[72] Datar, M., Immorlica, N., Indyk, P., and Mirrokni, V. S. Locality-sensitive hashing scheme based on p-stable distributions. In *20th Annual Symposium on Computational geometry* (2004), ACM, pp. 253–262.

[73] Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. Language modeling with gated convolutional networks. In *Proc. of the International Conference on Machine Learning (ICML)* (2017).

[74] de Castro, L., Agrawal, R., Yazicigil, R., Chandrakasan, A., Vaikuntanathan, V., Juvekar, C., and Joshi, A. Does Fully Homomorphic Encryption Need Compute Acceleration? *Cryptology ePrint Archive* (2021).

[75] Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Ouellet, P. Front-End Factor Analysis for Speaker Verification. *IEEE Transactions on Audio, Speech, and Language Processing 19*, 4 (May 2011), 788–798.

[76] Demmler, D., Schneider, T., and Zohner, M. ABY - A Framework for Efficient Mixed-Protocol Secure Two-Party Computation. In *NDSS* (2015), pp. –.

[77] Deng, J., Guo, J., Xue, N., and Zafeiriou, S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).* (2019), pp. 4685–4694.

[78] Desplanques, B., Thienpondt, J., and Demuynck, K. ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN Based Speaker Verification. In *Proc. Interspeech* (2020), pp. 3830–3834.

[79] Dias, M., Abad, A., and Trancoso, I. Exploring Hashing and Cryptonet Based Approaches for Privacy-Preserving Speech Emotion Recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), pp. 2057–2061.

[80] Dieleman, S., van den Oord, A., and Simonyan, K. The challenge of realistic music generation: modelling raw audio at scale. *Advances in Neural Information Processing Systems 31* (2018).

[81] Diez, M., Burget, L., and Matejka, P. Speaker Diarization based on Bayesian HMM with Eigenvoice Priors. In *Odyssey* (2018), pp. 147–154.

[82] Dwork, C. Differential privacy. In *International colloquium on automata, languages, and programming* (2006), Springer, pp. 1–12.

[83] Elazar, Y., and Goldberg, Y. Adversarial removal of demographic attributes from text data. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Brussels, Belgium, Oct.-Nov. 2018), Association for Computational Linguistics, pp. 11–21.

[84] Elgamal, T. A Public Key Cryptosystem and a Signature Scheme based on Discrete Logarithms. *IEEE Transactions on Information Theory 31*, 4 (1985), 469–472.

[85] Ericsson, D., Östberg, A., Zec, E. L., Martinsson, J., and Mogren, O. Adversarial representation learning for private speech generation. In *ICML 2020 Workshop on Self-supervision in Audio and Speech* (2020), pp. –.

[86] Escudero, D., Ghosh, S., Keller, M., Rachuri, R., and Scholl, P. Improved primitives for mpc over mixed arithmetic-binary circuits. In *Advances in Cryptology–CRYPTO 2020: 40th Annual International Cryptology Conference, CRYPTO 2020, Santa Barbara, CA, USA, August 17–21, 2020, Proceedings, Part II 40* (2020), Springer, pp. 823–852.

[87] Espinoza-Cuadros, F. M., Perero-Codosero, J. M., Antón-Martín, J., and Hernández-Gómez, L. A. Speaker de-identification system using autoencoders and adversarial training. *arXiv preprint arXiv:2011.04696* (2020).

[88] European Parliament and Council, . On the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). *Regulation 2016/679* (April 2016).

[89] Evans, D., Kolesnikov, V., and Rosulek, M. A pragmatic introduction to secure multi-party computation. *Foundations and Trends® in Privacy and Security 2*, 2-3 (2017).

[90] Fan, J., and Vercauteren, F. Somewhat Practical Fully Homomorphic Encryption. *IACR Cryptology ePrint Archive 2012* (2012), 144.

[91] Fang, F., Wang, X., Yamagishi, J., Echizen, I., Todisco, M., Evans, N., and Bonastre, J.-F. Speaker anonymization using x-vector and neural waveform models. In *10th ISCA Workshop on Speech Synthesis (SSW 10)* (2019), ISCA.

[92] Feng, T., and al. A review of speech-centric trustworthy machine learning: Privacy, safety, and fairness. *APSIPA Transactions on Signal and Information Processing 12*, 3 (2023).

[93] Feng, T., Hashemi, H., Annavaram, M., and Narayanan, S. S. Enhancing privacy through domain adaptive noise injection for speech emotion recognition. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2022), pp. 7702–7706.

[94] Feng, T., and Narayanan, S. Privacy and utility preserving data transformation for speech emotion recognition. In *2021 9th International Conference on Affective Computing and Intelligent Interaction (ACII)* (2021), pp. 1–7.

[95] Feng, T., Peri, R., and Narayanan, S. User-Level Differential Privacy against Attribute Inference Attack of Speech Emotion Recognition on Federated Learning. In *Proc. Interspeech 2022* (2022), pp. 5055–5059.

[96] Ferrer, L., and Riera, P. Confidence intervals for evaluation in machine learning.

[97] Forbes. Fraudsters cloned company director's voice in $35 million heist, police find. https://www.forbes.com/sites/thomasbrewster/2021/10/14/huge-bank-fraud-uses-deep-fake-voice-tech-to-steal-millions/, 2021. Accessed: 2024-01-12.

[98] Forbes. Who Is @BasedBeffJezos, the leader of the tech elite's "E/Acc" movement? https://www.forbes.com/sites/emilybaker-white/2023/12/01/who-is-basedbeffjezos-the-leader-of-effective-accelerationism-eacc/, 2023. Accessed: 2024-01-12.

[99] Fredrikson, M., Jha, S., and Ristenpart, T. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (2015), Association for Computing Machinery, p. 1322–1333.

[100] Fried, C. Privacy. *Yale Law Journal 77* (1968), 21.

[101] Fujita, Y., Kanda, N., Horiguchi, S., Xue, Y., Nagamatsu, K., and Watanabe, S. End-to-end neural speaker diarization with self-attention. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (2019), IEEE, pp. 296–303.

[102] Furukawa, J., Lindell, Y., Nof, A., and Weinstein, O. High-throughput secure three-party computation for malicious adversaries and an honest majority. In *Annual international conference on the theory and applications of cryptographic techniques* (2017), Springer, pp. 225–255.

[103] Ganin, Y., and Lempitsky, V. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning (ICML)* (2015), PMLR, pp. 1180–1189.

[104] Gao, S., Ver Steeg, G., and Galstyan, A. Efficient estimation of mutual information for strongly dependent variables. In *Artificial intelligence and statistics* (2015), PMLR, pp. 277–286.

[105] Gao, W., Oh, S., and Viswanath, P. Demystifying fixed k -nearest neighbor information estimators. *IEEE Transactions on Information Theory 64*, 8 (2018), 5629–5661.

[106] Garcia-Romero, D., Snyder, D., Sell, G., Povey, D., and McCree, A. Speaker diarization using deep neural network embeddings. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), pp. 4930–4934.

[107] Gentry, C. *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford University, 2009.

[108] Gilad-Bachrach, R., Dowlin, N., Laine, K., and al., e. CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy. In *ICML* (2016), vol. 48 of *JMLR Workshop and Conference Proceedings*, pp. 201–210.

[109] Glackin, C., Chollet, G., Dugan, N., Cannings, N., Wall, J., Tahir, S., Ray, I. G., and Rajarajan, M. Privacy preserving encrypted phonetic search of speech data. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), pp. 6414–6418.

[110] Goldreich, O., Micali, S., and Wigderson, A. How to play any mental game. In *19th Annual ACM Symposium on Theory of Computing* (New York, NY, USA, 1987), STOC '87, ACM, pp. 218–229.

[111] Gondi, S., and Pratap, V. Performance and Efficiency Evaluation of ASR Inference on the Edge. *Sustainability 13*, 22 (2021).

[112] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in Neural Information Processing Systems 27* (2014).

[113] Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings* (2015), Y. Bengio and Y. LeCun, Eds.

[114] Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proceedings of the 23rd international conference on Machine learning* (2006), pp. 369–376.

[115] Gulati, A., and al. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Proc. Interspeech 2020* (2020), pp. 5036–5040.

[116] Halevi, S., and Shoup, V. Algorithms in helib. In *Advances in Cryptology* (2014), J. A. Garay and R. Gennaro, Eds., Springer, pp. 554–571.

[117] Hesamifard, E., Takabi, H., and Ghasemi, M. Cryptodl: Deep neural networks over encrypted data. *arXiv preprint 1711.05189* (2017).

[118] Horiguchi, S., Yalta, N., Garcia, P., Takashima, Y., Xue, Y., Raj, D., Huang, Z., Fujita, Y., Watanabe, S., and Khudanpur, S. The Hitachi-JHU DIHARD III system: Competitive end-to-end neural diarization and x-vector clustering systems combined by DOVER-Lap. *arXiv preprint arXiv:2102.01363* (2021).

[119] Hu, H., Salcic, Z., Sun, L., Dobbie, G., Yu, P. S., and Zhang, X. Membership Inference Attacks on Machine Learning: A survey. *ACM Computing Surveys (CSUR) 54*, 11s (2022), 1–37.

[120] Hu, Y., and Loizou, P. C. Evaluation of objective quality measures for speech enhancement. *IEEE Transactions on audio, speech, and language processing 16*, 1 (2007), 229–238.

[121] Huang, X., Acero, A., Hon, H.-W., and Reddy, R. *Spoken language processing: A guide to theory, algorithm, and system development.* Prentice hall PTR, 2001.

[122] Hulaud, S. Identification of taste attributes from an audio signal, Apr. 3 2018. US Patent 9,934,785.

[123] Inness, J. C. *Privacy, intimacy, and isolation.* Oxford University Press, USA, 1992.

[124] Ioffe, S., and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML* (2015), PMLR, pp. 448–456.

[125] Ishai, Y., Kilian, J., Nissim, K., and Petrank, E. Extending oblivious transfers efficiently. In *Advances in Cryptology* (2003), Springer, pp. 145–161.

[126] Jaiswal, M., and Provost, E. M. Privacy enhanced multimodal neural representations for emotion recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2020), vol. 34, pp. 7985–7993.

[127] Janbakhshi, P., and Kodrasi, I. Adversarial-Free Speaker Identity-Invariant Representation Learning for Automatic Dysarthric Speech Classification. In *Proc. Interspeech 2022* (2022), pp. 2138–2142.

[128] Jang, E., Gu, S., and Poole, B. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).

[129] Jati, A., Hsu, C.-C., Pal, M., Peri, R., AbdAlmageed, W., and Narayanan, S. Adversarial attack and defense strategies for deep speaker recognition systems. *Computer Speech & Language 68* (2021), 101199.

[130] Jayaraman, B., Wang, L., Knipmeyer, K., Gu, Q., and Evans, D. Revisiting Membership Inference Under Realistic Assumptions. *Proceedings on Privacy Enhancing Technologies 2021*, 2 (2021).

[131] Jegou, H., Douze, M., and Schmid, C. Product quantization for nearest neighbor search. *IEEE transactions on pattern analysis and machine intelligence 33*, 1 (2010), 117–128.

[132] Jiménez, A., and Raj, B. Privacy preserving distance computation using somewhat-trusted third parties. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2017), pp. 6399–6403.

[133] Jiménez, A., Raj, B., Portêlo, J., and Trancoso, I. Secure Modular Hashing. In *Information Forensics and Security (WIFS), 2015 IEEE International Workshop* (2015), IEEE, pp. 1–6.

[134] Jin, H., and Wang, S. Voice-based determination of physical and emotional characteristics of users, Oct. 9 2018. US Patent 10,096,319.

[135] Julião, M., Abad, A., and Moniz, H. Exploring Text and Audio Embeddings for Multi-Dimension Elderly Emotion Recognition. In *Proc. Interspeech* (2020), pp. 2067–2071.

[136] Juvekar, C., Vaikuntanathan, V., and Chandrakasan, A. GAZELLE: A low latency framework for secure neural network inference. In *USENIX Security Symposium* (2018), pp. 1651–1669.

[137] Juvela, L., and Wang, X. Collaborative watermarking for adversarial speech synthesis. *arXiv preprint arXiv:2309.15224* (2023).

[138] Kalia, L. V., and Lang, A. E. Parkinson's disease. *Current neurology and neuroscience reports* (2015).

[139] Kamble, M. R., Sailor, H. B., Patil, H. A., and Li, H. Advances in anti-spoofing: from the perspective of asvspoof challenges. *APSIPA Transactions on Signal and Information Processing 9* (2020), e2.

[140] Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. What can we learn privately? *SIAM Journal on Computing 40*, 3 (2011), 793–826.

[141] Keller, M. Mp-spdz: A versatile framework for multi-party computation. *Cryptology ePrint Archive Report 2020/521* (2020).

[142] Keller, M., Orsini, E., and Scholl, P. Mascot: faster malicious arithmetic secure computation with oblivious transfer. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (2016), pp. 830–842.

[143] Kenny, P., Stafylakis, T., Ouellet, P., Alam, M. J., and Dumouchel, P. Plda for speaker verification with utterances of arbitrary duration. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (2013), IEEE, pp. 7649–7653.

[144] Kim, D. S. Perceptual phase redundancy in speech. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)* (Barcelona, Spain, May 2000).

[145] Kim, M., Günlü, O., and Schaefer, R. F. Federated learning with local differential privacy: Trade-offs between privacy, utility, and communication. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021* (2021), IEEE, pp. 2650–2654.

[146] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[147] Kingma, D. P., and Welling, M. Auto-encoding variational bayes, 2014.

[148] Kinnunen, T., Sahidullah, M., Delgado, H., Todisco, M., Evans, N., Yamagishi, J., and Lee, K. A. The ASVspoof 2017 Challenge: Assessing the Limits of Replay Spoofing Attack Detection. In *Proc. Interspeech 2017* (2017), pp. 2–6.

[149] Kolesnikov, V., and Schneider, T. Improved garbled circuit: Free xor gates and applications. In *International Colloquium on Automata, Languages, and Programming* (2008), Springer, pp. 486–498.

[150] Konečný, J., McMahan, H. B., Yu, F. X., Richtarik, P., Suresh, A. T., and Bacon, D. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning* (2016), pp. –.

[151] Kong, F., Duan, J., Ma, R., Shen, H. T., Zhu, X., Shi, X., and Xu, K. An efficient membership inference attack for the diffusion model by proximal initialization. In *The Twelfth International Conference on Learning Representations (ICLR)* (2024).

[152] Koppelmann, T., Nelus, A., Schönherr, L., Kolossa, D., and Martin, R. Privacy-Preserving Feature Extraction for Cloud-Based Wake Word Verification. In *Proc. Interspeech 2021* (2021), pp. 876–880.

[153] Kozachenko, L., and Leonenko, N. A statistical estimate for the entropy of a random vector. *Problems of Information Transmission* (1987), 9–16.

[154] Kraskov, A., Stögbauer, H., and Grassberger, P. Estimating mutual information. *Phys. Rev. E 69* (Jun 2004), 066138.

[155] Kreuk, F., Adi, Y., Cisse, M., and Keshet, J. Fooling end-to-end speaker verification with adversarial examples. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Calgary, AB, Canada, April 2018).

[156] Kreuk, F., Adi, Y., Raj, B., Singh, R., and Keshet, J. Hide and speak: Towards deep neural networks for speech steganography. In *Proc. of Interspeech* (Shanghai, China, October 2020).

[157] Krizhevsky, A., and Hinton, G. Learning multiple layers of features from tiny images. Tech. rep., University of Toronto, 2009.

[158] Kröger, J. L., Gellrich, L., Pape, S., Brause, S. R., and Ullrich, S. Personal information inference from voice recordings: User awareness and privacy concerns. *Proc. Priv. Enhancing Technol. 2022*, 1 (2022), 6–27.

[159] Kröger, J. L., Lutz, O. H.-M., and Raschke, P. Privacy implications of voice and speech analysis – information disclosure by inference. *Privacy and Identity Management. Data for Better Living: AI and Privacy: 14th IFIP WG 9.2, 9.6/11.7, 11.6/SIG 9.2. 2 International Summer School, 2019, Revised Selected Papers 14* (2019), 242–258.

[160] Kunešová, M., Hrúz, M., Zajíc, Z., and Radová, V. Detection of overlapping speech for the purposes of speaker diarization. In *International Conference on Speech and Computer* (2019), Springer, pp. 247–257.

[161] Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial examples in the physical world. In *Proc. of the International Conference on Learning Representations (ICLR) Workshop Track* (April 2017).

[162] Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial machine learning at scale. In *Proc. of the International Conference on Learning Representations (ICLR)* (2017).

[163] Kwasny, D., and Hemmerling, D. Joint gender and age estimation based on speech signals using x-vectors and transfer learning. *arXiv preprint arXiv:2012.01551* (2020).

[164] Laine, K. Microsoft SEAL (release 3.5). Tech. rep., Microsoft Research, Redmond, WA., Apr. 2020.

[165] Landini, F., Lozano-Diez, A., Burget, L., et al. BUT system description for the third DIHARD speech diarization challenge. In *Proc. 3rd DIHARD Speech Diarization Challenge Workshop* (2021).

[166] Landini, F., Profant, J., Diez, M., and Burget, L. Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: Theory, implementation and analysis on standard tasks. *Computer Speech & Language 71* (2022), 101254.

[167] Laur, S., Lipmaa, H., and Mielikäinen, T. Cryptographically private support vector machines. In *12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2006), ACM, pp. 618–624.

[168] Laver, J. *Principles of phonetics*. Cambridge university press, 1994.

[169] LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based Learning Applied to Document Recognition. *IEEE 86*, 11 (1998), 2278–2324.

[170] Lee, G., Kim, M., Park, J. H., Hwang, S. W., and Cheon, J. H. Privacy-preserving text classification on bert embeddings with homomorphic encryption. In *2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022* (2022), Association for Computational Linguistics (ACL), pp. 3169–3175.

[171] Li, B., and Micciancio, D. On the security of homomorphic encryption on approximate numbers. In *Advances in Cryptology–EUROCRYPT 2021: 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, October 17–21, 2021, Proceedings, Part I 40* (2021), Springer, pp. 648–677.

[172] Li, H., and Zhao, X. Membership information leakage in well-generalized auto speech recognition systems. In *2023 International Conference on Data Science and Network Security (ICDSNS)* (2023), IEEE, pp. 1–7.

[173] Li, J., Han, J., Deng, S., Zheng, T., He, Y., and Zheng, G. Mutual Information-based Embedding Decoupling for Generalizable Speaker Verification. In *Proc. Intershpeech 2023* (2023), pp. 3147–3151.

[174] Li, J., Zhang, X., Jia, C., Xu, J., Zhang, L., Wang, Y., Ma, S., and Gao, W. Universal adversarial perturbations generative network for speaker recognition. In *Proc. of the IEEE International Conference on Multimedia and Expo (ICME)* (July 2020).

[175] Li, X., Zhong, J., Wu, X., Yu, J., Liu, X., and Meng, H. Adversarial Attacks on GMM I-Vector Based Speaker Verification Systems. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020).

[176] Li, Z., Shi, C., Xie, Y., Liu, J., Yuan, B., and Chen, Y. Practical adversarial attacks against speaker recognition systems. In *Proc. of the International Workshop on Mobile Computing Systems and Applications* (2020).

[177] Lim, W. Y. B., Luong, N. C., Hoang, D. T., Jiao, Y., Liang, Y.-C., Yang, Q., Niyato, D., and Miao, C. Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials 22*, 3 (2020), 2031–2063.

[178] Lindell, Y. Secure multiparty computation (mpc). *IACR Cryptology ePrint Archive 2020* (2020), 300.

[179] Liu, J., Juuti, M., Lu, Y., and Asokan, N. Oblivious neural network predictions via miniONN transformations. In *ACM SIGSAC Conference on Computer and Communications Security* (2017), pp. 619–631.

[180] Livingstone, S. R., and Russo, F. A. The ryerson audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PloS one 13*, 5 (2018), e0196391.

[181] Luo, J., Zhang, Y., Zhang, J., Mu, X., Wang, H., Yu, Y., and Xu, Z. Secformer: Towards fast and accurate privacy-preserving inference for large language models. *arXiv preprint arXiv:2401.00793* (2024).

[182] Luu, C., Renals, S., and Bell, P. Investigating the contribution of speaker attributes to speaker separability using disentangled speaker representations. In *Proc. Interspeech 2022* (2022), pp. 610–614.

[183] Lyubashevsky, V., Peikert, C., and Regev, O. On ideal lattices and learning with errors over rings. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (2010), Springer, pp. 1–23.

[184] Ma, S., Wang, H., Ma, L., Wang, L., Wang, W., Huang, S., Dong, L., Wang, R., Xue, J., and Wei, F. The era of 1-bit llms: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764* (2024).

[185] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR 2018, Conference Track Proceedings* (2018).

[186] Makri, E., Rotaru, D., Smart, N. P., and Vercauteren, F. PICS: Private Image Classification with SVM. *IACR Cryptology ePrint Archive 2017* (2017), 1190.

[187] Makri, E., Rotaru, D., Smart, N. P., and Vercauteren, F. Epic: efficient private image classification (or: learning from the masters). In *Cryptographers' Track at the RSA Conference* (2019), Springer, pp. 473–492.

[188] Manocha, P., Finkelstein, A., Zhang, R., Bryan, N. J., Mysore, G. J., and Jin, Z. A differentiable perceptual audio metric learned from just noticeable differences. In *Proc. of Interspeech* (Shanghai, China, October 2020).

[189] Maouche, M., Srivastava, B. M. L., Vauquier, N., Bellet, A., Tommasi, M., and Vincent, E. Enhancing Speech Privacy with Slicing. In *Proc. Interspeech 2022* (2022), pp. 5025–5029.

[190] Mendonça, J., Teixeira, F., Trancoso, I., and Abad, A. Analyzing Breath Signals for the Interspeech 2020 ComParE Challenge. In *Proc. Interspeech 2020* (2020), pp. 2077–2081.

[191] Mendonça, J., and Trancoso, I. VoxCeleb-PT – a dataset for a speech processing course . In *Proc. IberSPEECH 2022* (2022), pp. 71–75.

[192] Mervosh, S. Distorted videos of nancy pelosi spread on facebook and twitter, helped by trump. *The New York Times* (2019). [Online: https://www.nytimes.com/2019/05/24/us/politics/pelosi-doctored-video.html; Accessed 18-01-2022].

[193] Meyer, S., Tilli, P., Denisov, P., Lux, F., Koch, J., and Vu, N. T. Anonymizing speech with generative adversarial networks to preserve speaker privacy. In *2022 IEEE Spoken Language Technology Workshop (SLT)* (2023), IEEE, pp. 912–919.

[194] Miao, X., Wang, X., Cooper, E., Yamagishi, J., and Tomashenko, N. Language-independent speaker anonymization approach using self-supervised pre-trained models. In *Proc. The Speaker and Language Recognition Workshop (Odyssey 2022)* (2022), pp. 279–286.

[195] Miao, Y., Xue, M., Chen, C., Pan, L., Zhang, J., Zhao, B. Z. H., Kaafar, D., and Xiang, Y. The Audio Auditor: User-Level Membership Inference in Internet of Things Voice Services. In *Proc. Priv. Enhancing Technol.* (2021), pp. 209–228.

[196] Mishra, P., Lehmkuhl, R., Srinivasan, A., Zheng, W., and Popa, R. A. DELPHI: A Cryptographic Inference Service for Neural Networks. In *29th USENIX Security Symposium* (2020), pp. –.

[197] Mohassel, P., and Rindal, P. ABY3: A mixed protocol framework for machine learning. In *ACM SIGSAC Conference on Computer and Communications Security* (2018), pp. 35–52.

[198] Moore, A. D. Defining privacy. *Journal of Social Philosophy 39*, 3 (2008), 411–428.

[199] Moro-Velazquez, L., Villalba, J., and Dehak, N. Using x-vectors to automatically detect parkinson's disease from speech. In *Proc. ICASSP* (2020), IEEE, pp. 1155–1159.

[200] Morshed, T., Aziz, M., and Mohammed, N. CPU and GPU Accelerated Fully Homomorphic Encryption. In *2020 IEEE International Symposium on Hardware Oriented Security and Trust (HOST)* (2020), IEEE Computer Society, pp. 142–153.

[201] Mporas, I., and Ganchev, T. Estimation of unknown speaker's height from speech. *International Journal of Speech Technology 12* (2009), 149–160.

[202] Mtibaa, A. *Towards robust and privacy-preserving speaker verification systems.* PhD thesis, Institut polytechnique de Paris, 2022.

[203] Mtibaa, A., Petrovska-Delacretaz, D., and Ben Hamida, A. Cancelable speaker verification system based on binary gaussian mixtures. In *2018 4th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP)* (2018), pp. 1–6.

[204] Mtibaa, A., Petrovska-Delacrétaz, D., Boudy, J., and Ben Hamida, A. Privacy-preserving speaker verification system based on binary i-vectors. *IET Biometrics 10*, 3 (2021), 233–245.

[205] Mun, S. H., Han, M. H., Kim, M., Lee, D., and Kim, N. S. Disentangled speaker representation learning via mutual information minimization. In *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* (2022), pp. 89–96.

[206] Nagrani, A., Chung, J. S., Xie, W., and Zisserman, A. Voxceleb: Large-scale speaker verification in the wild. *Computer Speech & Language 60* (2020), 101027.

[207] Nagrani, A., Chung, J. S., and Zisserman, A. Voxceleb: A large-scale speaker identification dataset. In *Proc. Interspeech 2017* (2017).

[208] Nautsch, A., Isadskiy, S., Kolberg, J., Gomez-Barrero, M., and Busch, C. Homomorphic Encryption for Speaker Recognition: Protection of Biometric Templates and Vendor Model Parameters . In *Proc. The Speaker and Language Recognition Workshop (Odyssey 2018)* (2018), pp. 16–23.

[209] Nautsch, A., Jasserand, C., Kindt, E., Todisco, M., Trancoso, I., and Evans, N. The GDPR & Speech Data: Reflections of Legal and Technology Communities, First Steps Towards a Common Understanding. In *Proc. Interspeech 2019* (2019), pp. 3695–3699.

[210] Nautsch, A., Jiménez, A., Treiber, A., Kolberg, J., Jasserand, C., Kindt, E., Delgado, H., Todisco, M., Hmani, M. A., Mtibaa, A., et al. Preserving privacy in speaker and speech characterisation. *Computer Speech & Language 58* (2019), 441–480.

[211] Nautsch, A., Patino, J., Tomashenko, N., Yamagishi, J., Noé, P.-G., Bonastre, J.-F., Todisco, M., and Evans, N. The Privacy ZEBRA: Zero Evidence Biometric Recognition Assessment. In *Proc. Interspeech 2020* (2020), pp. 1698–1702.

[212] Nautsch, A., Patino, J., Treiber, A., Stafylakis, T., Mizera, P., Todisco, M., Schneider, T., and Evans, N. Privacy-Preserving Speaker Recognition with Cohort Score Normalisation. In *Proc. Interspeech 2019* (2019), pp. 2868–2872.

[213] Nelus, A., and Martin, R. Gender discrimination versus speaker identification through privacy-aware adversarial feature extraction. In *Speech Communication; 13th ITG-Symposium* (2018), pp. 1–5.

[214] Nelus, A., and Martin, R. Privacy-aware feature extraction for gender discrimination versus speaker identification. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2019), pp. 671–674.

[215] Nelus, A., and Martin, R. Privacy-preserving audio classification using variational information feature extraction. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 29* (2021), 2864–2877.

[216] Nelus, A., Rech, S., Koppelmann, T., Biermann, H., and Martin, R. Privacy-Preserving Siamese Feature Extraction for Gender Recognition versus Speaker Identification. In *Proc. Interspeech 2019* (2019), pp. 3705–3709.

[217] Nielsen, J. B., Nordholt, P. S., Orlandi, C., and Burra, S. S. A new approach to practical active-secure two-party computation. In *Advances in Cryptology – CRYPTO 2012* (2012), R. Safavi-Naini and R. Canetti, Eds., Springer, pp. 681–700.

[218] Nissenbaum, H. Privacy as contextual integrity. *Wash. L. Rev. 79* (2004), 119.

[219] Nissenbaum, H. *Privacy in context: Technology, policy, and the integrity of social life*. Stanford University Press, 2020.

[220] Noé, P.-G., Miao, X., Wang, X., Yamagishi, J., Bonastre, J.-F., and Matrouf, D. Hiding speaker's sex in speech using zero-evidence speaker representation in an analysis/synthesis pipeline. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2023), IEEE, pp. 1–5.

[221] Noé, P.-G., Nautsch, A., Matrouf, D., Bousquet, P.-M., and Bonastre, J.-F. A bridge between features and evidence for binary attribute-driven perfect privacy. In *Proc. ICASSP* (2022), IEEE, pp. 3094–3098.

[222] Noé, P.-G., Mohammadamini, M., Matrouf, D., Parcollet, T., Nautsch, A., and Bonastre, J.-F. Adversarial Disentanglement of Speaker Representation for Attribute-Driven Privacy Preservation. In *Proc. Interspeech 2021* (2021), pp. 1902–1906.

[223] O. Rabin, M. How to exchange secrets with oblivious transfer. *IACR Cryptology ePrint Archive 2005* (01 2005), 187.

[224] Oh, T.-H., Dekel, T., Kim, C., Mosseri, I., Freeman, W. T., Rubinstein, M., and Matusik, W. Speech2face: Learning the face behind a voice. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2019), pp. 7539–7548.

[225] Okabe, K., Koshinaka, T., and Shinoda, K. Attentive Statistics Pooling for Deep Speaker Embedding. In *Proc. Interspeech 2018* (2018), pp. 2252–2256.

[226] Olivier, R., and Raj, B. Recent improvements of ASR models in the face of adversarial attacks. In *Proc. Interspeech 2022* (2022), pp. 4113–4117.

[227] Orozco-Arroyave, J. R., Arias-Londoño, J. D., Bonilla, J. F. V., Gonzalez-Rátiva, M. C., and al., e. New Spanish Speech Corpus Database for the Analysis of People Suffering from Parkinson's Disease. In *LREC* (2014), pp. 342–347.

[228] Paillier, P. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology* (1999), vol. 1592 of *Lecture Notes in Computer Science*, pp. 223–238.

[229] Paiva, T., Andersen, M., and Tufik, S. Sono e a medicina do sono. 1ª edição, 2014.

[230] Panariello, M., Nespoli, F., Todisco, M., and Evans, N. Speaker anonymization using neural audio codec language models. *arXiv preprint arXiv:2309.14129* (2023).

[231] Panariello, M., Todisco, M., and Evans, N. Vocoder drift in x-vector–based speaker anonymization. In *Proc. INTERSPEECH 2023* (2023), pp. 2863–2867.

[232] Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2015), pp. 5206–5210.

[233] Pang, Q., Zhu, J., Möllering, H., Zheng, W., and Schneider, T. Bolt: Privacy-preserving, accurate and efficient inference for transformers. In *2024 IEEE Symposium on Security and Privacy (SP)* (2024), IEEE Computer Society, pp. 130–130.

[234] Pappagari, R., Wang, T., Villalba, J., Chen, N., and Dehak, N. x-vectors meet emotions: A study on dependencies between emotion and speaker recognition. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), IEEE, pp. 7169–7173.

[235] Parcollet, T., and Ravanelli, M. The Energy and Carbon Footprint of Training End-to-End Speech Recognizers. In *Proc. Interspeech 2021* (2021), pp. 4583–4587.

[236] Park, T. J., Kanda, N., Dimitriadis, D., Han, K. J., Watanabe, S., and Narayanan, S. A review of speaker diarization: Recent advances with deep learning. *Computer Speech & Language 72* (2022), 101317.

[237] Parthasarathi, S. H. K., Bourlard, H., and Gatica-Perez, D. Wordless sounds: Robust speaker diarization using privacy-preserving audio representations. *IEEE transactions on audio, speech, and language processing 21*, 1 (2012), 85–98.

[238] Pathak, M. A., and Raj, B. Privacy-preserving speaker verification as password matching. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2012), pp. 1849–1852.

[239] Pathak, M. A., and Raj, B. Privacy-Preserving Speaker Verification and Identification Using Gaussian Mixture Models. *IEEE Transactions on Audio, Speech, and Language Processing 21*, 2 (Feb 2013), 397–406.

[240] Patino, J., Tomashenko, N., Todisco, M., Nautsch, A., and Evans, N. Speaker Anonymisation Using the McAdams Coefficient. In *Proc. Interspeech 2021* (2021), pp. 1099–1103.

[241] Pavlović, K., Kovačević, S., Djurović, I., and Wojciechowski, A. Robust speech watermarking by a jointly trained embedder and detector using a dnn. *Digital Signal Processing 122* (2022), 103381.

[242] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research 12* (2011), 2825–2830.

[243] Perero-Codosero, J. M., Espinoza-Cuadros, F., Antón-Martín, J., Barbero-Alvarez, M. A., and Hernández-Gómez, L. A. Modeling obstructive sleep apnea voices using deep neural network embeddings and domain-adversarial training. *IEEE Journal of Selected Topics in Signal Processing 14*, 2 (2019), 240–250.

[244] Perero-Codosero, J. M., Espinoza-Cuadros, F. M., and Hernández-Gómez, L. A. X-vector anonymization using autoencoders and adversarial training for preserving speech privacy. *Computer Speech & Language 74* (2022), 101351.

[245] Pierre, C., Larcher, A., and Jouvet, D. Are disentangled representations all you need to build speaker anonymization systems? In *Proc. Interspeech 2022* (2022), pp. 2793–2797.

[246] Pinkas, B., Schneider, T., Smart, N. P., and Williams, S. C. Secure two-party computation is practical. In *International Conference on the Theory and Application of Cryptology and Information Security* (2009), Springer, pp. 250–267.

[247] Pinkas, B., Schneider, T., Weinert, C., and Wieder, U. Efficient circuit-based psi via cuckoo hashing. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques* (2018), Springer, pp. 125–157.

[248] Pizzi, K., Boenisch, F., Sahin, U., and Böttinger, K. Introducing model inversion attacks on automatic speaker recognition. *arXiv preprint arXiv:2301.03206* (2023).

[249] Polzehl, T., Möller, S., and Metze, F. Automatically assessing personality from speech. In *2010 IEEE fourth international conference on semantic computing* (2010), IEEE, pp. 134–140.

[250] Pompili, A., Abad, A., de Matos, D. M., and Martins, I. P. Pragmatic aspects of discourse production for the automatic identification of alzheimer's disease. *IEEE Journal of Selected Topics in Signal Processing* (Jan. 2020), 1–11.

[251] Pompili, A., Abad, A., Romano, P., Martins, I. P., Cardoso, R., Santos, H., Carvalho, J., Guimaraes, I., and Ferreira, J. J. Automatic detection of parkinson's

disease: an experimental analysis of common speech production tasks used for diagnosis. In *International Conference on Text, Speech, and Dialogue* (2017), Springer, pp. 411–419.

[252] Portêlo, J. *Privacy-Preserving Frameworks for Speech Mining.* PhD thesis, Instituto Superior Técnico, 2015.

[253] Portêlo, J., Abad, A., Raj, B., and Trancoso, I. Secure Binary Embeddings of Front-end Factor Analysis for Privacy Preserving Speaker Verification. In *Proc. Interspeech 2013* (2013), pp. 2494–2498.

[254] Portêlo, J., Abad, A., Raj, B., and Trancoso, I. Privacy-preserving Query-by-Example Speech Search. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2015), IEEE, pp. 1797–1801.

[255] Portêlo, J., Raj, B., Abad, A., and Trancoso, I. Privacy-preserving speaker verification using garbled GMMs. In *2014 22nd European Signal Processing Conference (EUSIPCO)* (2014), pp. 2070–2074.

[256] Posner, R. A. *Economic analysis of law*, ninth ed. Aspen Publishing, 2014.

[257] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. The kaldi speech recognition toolkit. In *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding* (Big Island, Hawaii, US, December 2011).

[258] Princen, J., and Bradley, A. Analysis/synthesis filter bank design based on time domain aliasing cancellation. *IEEE Transactions on Acoustics, Speech, and Signal Processing 34*, 5 (1986), 1153–1161.

[259] Punjabi, N. The epidemiology of adult obstructive sleep apnea. *American Thoracic Society 5*, 2 (2008), 136–143.

[260] Quintas, S., Mauclair, J., Woisard, V., and Pinquier, J. Automatic Assessment of Speech Intelligibility using Consonant Similarity for Head and Neck Cancer. In *Proc. Interspeech* (2022), pp. 3608–3612.

[261] Rabiner, L. R., Schafer, R. W., et al. Introduction to digital speech processing. *Foundations and Trends® in Signal Processing 1*, 1–2 (2007), 1–194.

[262] Rahulamathavan, Y., Phan, R. C.-W., Veluru, S., and al., e. Privacy-preserving multi-class support vector machine for outsourcing the data classification in cloud. *IEEE Transactions on Dependable and Secure Computing 11*, 5 (2013), 467–479.

[263] Raj, D., Garcia-Perera, L. P., Huang, Z., Watanabe, S., Povey, D., Stolcke, A., and Khudanpur, S. Dover-lap: A method for combining overlap-aware diarization outputs. In *2021 IEEE Spoken Language Technology Workshop (SLT)* (2021), IEEE, pp. 881–888.

[264] Raj, D., Snyder, D., Povey, D., and Khudanpur, S. Probing the Information Encoded in X-Vectors. In *Proc. ASRU* (2019), pp. 726–733.

[265] Ravanelli, M., Parcollet, T., Plantinga, P., Rouhe, A., Cornell, S., Lugosch, L., Subakan, C., Dawalatabad, N., Heba, A., Zhong, J., Chou, J.-C., Yeh, S.-L., Fu, S.-W., Liao, C.-F., Rastorgueva, E., Grondin, F., Aris, W., Na, H., Gao, Y., Mori, R. D., and Bengio, Y. SpeechBrain: A General-Purpose Speech Toolkit, 2021. arXiv:2106.04624.

[266] Ravi, V., Wang, J., Flint, J., and Alwan, A. A Step Towards Preserving Speakers' Identity While Detecting Depression Via Speaker Disentanglement. In *Proc. Interspeech 2022* (2022), pp. 3338–3342.

[267] Ravi, V., Wang, J., Flint, J., and Alwan, A. Enhancing accuracy and privacy in speech-based depression detection through speaker disentanglement. *Computer Speech & Language 86* (2024), 101605.

[268] Regev, O. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM) 56*, 6 (2009), 1–40.

[269] Reiman, J. H. Privacy, intimacy, and personhood. *Philosophy and Public Affairs 6*, 1 (1976).

[270] Ren, L., Liu, Z., Li, F., Liang, K., Li, Z., and Luo, B. Privdnn: A secure multi-party computation framework for deep learning using partial dnn encryption. *Proceedings on Privacy Enhancing Technologies 3* (2024), 1–18.

[271] Reynolds, D. A., Quatieri, T. F., and Dunn, R. B. Speaker verification using adapted gaussian mixture models. *Digital signal processing 10*, 1-3 (2000), 19–41.

[272] Rezaei, S., and Liu, X. On the Difficulty of Membership Inference Attacks. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 7888–7896.

[273] Riazi, M. S., Weinert, C., Tkachenko, O., Songhori, E. M., Schneider, T., and Koushanfar, F. Chameleon: A hybrid secure computation framework for machine learning applications. In *Asia Conference on Computer and Communications Security* (2018), ACM, pp. 707–721.

[274] Rivest, R. L., Shamir, A., and Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM 21*, 2 (1978), 120–126.

[275] Ross, B. C. Mutual information between discrete and continuous data sets. *PloS one 9*, 2 (2014), e87357.

[276] Rotaru, D., and Wood, T. Marbled circuits: Mixing arithmetic and boolean circuits with active security. In *International Conference on Cryptology in India* (2019), Springer, pp. 227–249.

[277] Ryant, N., Church, K., Cieri, C., Cristia, A., Du, J., Ganapathy, S., and Liberman, M. The second dihard diarization challenge: Dataset, task, and baselines.

[278] Ryant, N., Church, K., Cieri, C., Cristia, A., Du, J., Ganapathy, S., and Liberman, M. First dihard challenge evaluation plan. Tech. rep., Linguistic Data Consortium (LDC), 2018.

[279] Ryant, N., Singh, P., Krishnamohan, V., Varma, R., Church, K., Cieri, C., Du, J., Ganapathy, S., and Liberman, M. The third dihard diarization challenge, 2021.

[280] Schneider, T., and Zohner, M. GMW vs. Yao? Efficient Secure Two-Party Computation with Low Depth Circuits. In *Financial Cryptography and Data Security* (2013), Springer, pp. 275–292.

[281] Schuller, B., and Batliner, A. *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*. John Wiley & Sons, 2013.

[282] Sell, G., Snyder, D., McCree, A., Garcia-Romero, D., Villalba, J., Maciejewski, M., Manohar, V., Dehak, N., Povey, D., Watanabe, S., and Khudanpur, S. Diarization is Hard: Some Experiences and Lessons Learned for the JHU Team in the Inaugural DIHARD Challenge. In *Proc. Interspeech 2018* (2018), pp. 2808–2812.

[283] Shah, M. A., Szurley, J., Mueller, M., Mouchtaris, A., and Droppo, J. Evaluating the Vulnerability of End-to-End Automatic Speech Recognition Models to Membership Inference Attacks. In *Proc. Interspeech 2021* (2021), pp. 891–895.

[284] Shamir, A. How to share a secret. *Communications of the ACM 22*, 11 (1979), 612–613.

[285] Shamsabadi, A. S., Srivastava, B. M. L., Bellet, A., Vauquier, N., Vincent, E., Maouche, M., Tommasi, M., and Papernot, N. Differentially private speaker anonymization. *Proceedings on Privacy Enhancing Technologies 1* (2023), 98–114.

[286] Shamsabadi, A. S., Teixeira, F. S., Abad, A., Raj, B., Cavallaro, A., and Trancoso, I. Foolhd: Fooling speaker identification by highly imperceptible adversarial disturbances. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), IEEE, pp. 6159–6163.

[287] Shokri, R., Stronati, M., Song, C., and Shmatikov, V. Membership Inference Attacks against Machine Learning Models. In *2017 IEEE symposium on security and privacy (SP)* (2017), IEEE, pp. 3–18.

[288] Sigurdsson, S., Petersen, K. B., and Lehn-Schiøler, T. Mel frequency cepstral coefficients: An evaluation of robustness of mp3 encoded music. In *International Society for Music Information Retrieval Conference (ISMIR)* (2006), pp. 286–289.

[289] Singh, R. *Profiling humans from their voice*. Springer, 2019.

[290] Singh, R., Shah, A., and Dhamyal, H. An overview of techniques for biomarker discovery in voice signal. *arXiv* (2021).

[291] Smith, L. N., and Topin, N. Super-convergence: Very fast training of neural networks using large learning rates. *Artificial intelligence and machine learning for multi-domain operations applications 11006* (2019), 369–386.

[292] Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (April 2018), pp. 5329–5333.

[293] Snyder, P. Yao's garbled circuits: Recent directions and implementations. *Literature review, Dept. of Computer Science, University of Illinois at Chicago* (2014).

[294] Solera-Ureña, R., Botelho, C., Teixeira, F., Rolland, T., Abad, A., and Trancoso, I. Transfer learning-based cough representations for automatic detection of covid-19. In *Proc. Interspeech 2021* (2021), pp. 4336–4340.

[295] Solove, D. J. Conceptualizing privacy. *California law review* (2002), 1087–1155.

[296] Solove, D. J. A taxonomy of privacy. *University of Pennsylvania law review 154*, 3 (2006), 477–560.

[297] Solove, D. J. *Understanding privacy*. Harvard university press, 2010.

[298] Solove, D. J., and Schwartz, P. M. *Information privacy law*. Aspen Publishing, 2020.

[299] Song, C., and Shmatikov, V. Auditing data provenance in text-generation models. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (2019), pp. 196–206.

[300] Srivastava, B. M. L., Bellet, A., Tommasi, M., and Vincent, E. Privacy-preserving adversarial representation learning in asr: Reality or illusion? In *INTERSPEECH 2019-20th Annual Conference of the International Speech Communication Association* (2019).

[301] Srivastava, B. M. L., Vauquier, N., Sahidullah, M., Bellet, A., Tommasi, M., and Vincent, E. Evaluating voice conversion-based privacy protection against informed attackers. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2020), IEEE, pp. 2802–2806.

[302] Statista. Number of digital voice assistants in use worldwide from 2019 to 2024. https://www.statista.com/statistics/973815/worldwide-digital-voice-assistant-in-use/, 2022. Accessed: 2023-12-29.

[303] Statista. Smart speaker market revenue worldwide from 2014 to 2025. https://www.statista.com/statistics/1022823/worldwide-smart-speaker-market-revenue/, 2022. Accessed: 2023-12-29.

[304] Stoidis, D., and Cavallaro, A. Protecting Gender and Identity with Disentangled Speech Representations. In *Proc. Interspeech 2021* (2021), pp. 1699–1703.

[305] Stoidis, D., and Cavallaro, A. Generating gender-ambiguous voices for privacy-preserving speech recognition. In *Proc. Interspeech 2022* (2022), pp. 4237–4241.

[306] Stolcke, A., and Yoshioka, T. Dover: A method for combining diarization outputs. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (2019), IEEE, pp. 757–763.

[307] Stowell, D., Giannoulis, D., Benetos, E., Lagrange, M., and Plumbley, M. D. Detection and classification of acoustic scenes and events. *IEEE Transactions on Multimedia 17*, 10 (2015), 1733–1746.

[308] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *Proc. of the International Conference on Learning Representations (ICLR)* (Banff, Canada, April 2014).

[309] Tawara, N., Ogawa, A., Kitagishi, Y., and Kamiyama, H. Age-vox-celeb: Multi-modal corpus for facial and speech estimation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2021), IEEE, pp. 6963–6967.

[310] Teixeira, F., Abad, A., Raj, B., and Trancoso, I. Towards End-to-End Private Automatic Speaker Recognition. In *Proc. Interspeech* (2022), pp. 2798–2802.

[311] Teixeira, F., Abad, A., Raj, B., and Trancoso, I. Privacy-preserving automatic speaker diarization. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2023), pp. 1–5.

[312] Teixeira, F., Abad, A., Raj, B., and Trancoso, I. Privacy-oriented manipulation of speaker representations. *IEEE Access 12* (2024), 82949–82971.

[313] Teixeira, F., Abad, A., and Trancoso, I. Patient privacy in paralinguistic tasks. In *Proc. Interspeech, 2018* (2018), pp. 3428–3432.

[314] Teixeira, F., Abad, A., and Trancoso, I. Privacy-preserving paralinguistic tasks. In *ICASSP* (May 2019), pp. 6575–6579.

[315] Teixeira, F., Abad, A., Trancoso, I., and Raj, B. Voice Biometrics: Privacy in Paralinguistic and Extra-Linguistic Tasks. In *Voice Biometrics: Technology, trust and security*, C. Garcia-Mateo and G. Chollet, Eds. IET, 2021, ch. 4. ISBN: 978-1-78561-900-7.

[316] Thaine, P., and Penn, G. Extracting mel-frequency and bark-frequency cepstral coefficients from encrypted signals. *Interspeech* (2019), 3715–3719.

[317] Thompson, N. C., Greenewald, K., Lee, K., and Manso, G. F. The computational limits of deep learning, 2020.

[318] Tomashenko, N., Srivastava, B. M. L., Wang, X., Vincent, E., Nautsch, A., Yamagishi, J., Evans, N., Patino, J., Bonastre, J.-F., Noé, P.-G., et al. Introducing the voiceprivacy initiative. In *Proc. Interspeech 2020* (2020).

[319] Tomashenko, N., Wang, X., Miao, X., Nourtel, H., Champion, P., Todisco, M., Vincent, E., Evans, N., Yamagishi, J., and Bonastre, J.-F. The voiceprivacy 2022 challenge evaluation plan. *arXiv preprint arXiv:2203.12468* (2022).

[320] Tomashenko, N., Wang, X., Vincent, E., Patino, J., Srivastava, B. M. L., Noé, P.-G., Nautsch, A., Evans, N., Yamagishi, J., O'Brien, B., Chanclu, A., Bonastre, J.-F., Todisco, M., and Maouche, M. The voiceprivacy 2020 challenge: Results and findings, 2021.

[321] Tramèr, F., Zhang, F., Juels, A., Reiter, M. K., and Ristenpart, T. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)* (2016), pp. 601–618.

[322] Tranter, S. E., and Reynolds, D. A. An overview of automatic speaker diarization systems. *IEEE Transactions on audio, speech, and language processing 14*, 5 (2006), 1557–1565.

[323] Treiber, A., Nautsch, A., Kolberg, J., Schneider, T., and Busch, C. Privacy-preserving PLDA speaker verification using outsourced secure computation. *Speech Communication 114* (2019), 60–71.

[324] Tseng, W.-C., Kao, W.-T., and yi Lee, H. Membership Inference Attacks Against Self-supervised Speech Models. In *Proc. Interspeech 2022* (2022), pp. 5040–5044.

[325] Turner, H., Lovisotto, G., and Martinovic, I. Generating identities with mixture models for speaker anonymization. *Computer Speech & Language 72* (2022), 101318.

[326] United Nations. Universal declaration of human rights, Dec. 1948.

[327] US Public Law 104-191. Health insurance portability and accountability act of 1996, 1996.

[328] Vaidya, J., Yu, H., and Jiang, X. Privacy-preserving SVM classification. *Knowledge and Information Systems 14*, 2 (2008), 161–178.

[329] van Bemmel, L., Liu, Z., Vaessen, N., and Larson, M. Beyond neural-on-neural approaches to speaker gender protection. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2023), pp. 1–5.

[330] Van Den Oord, A., Vinyals, O., et al. Neural discrete representation learning. *Advances in neural information processing systems 30* (2017).

[331] Variani, E., Lei, X., McDermott, E., Moreno, I. L., and Gonzalez-Dominguez, J. Deep neural networks for small footprint text-dependent speaker verification. In *2014 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (2014), IEEE, pp. 4052–4056.

[332] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention Is All You Need. *Advances in neural information processing systems 30* (2017).

[333] Villalba, J., Zhang, Y., and Dehak, N. x-Vectors Meet Adversarial Attacks: Benchmarking Adversarial Robustness in Speaker Verification. In *Proc. Interspeech 2020* (2020), pp. 4233–4237.

[334] Vásquez-Correa, J., Orozco-Arroyave, J. R., and Nöth, E. Convolutional neural network to model articulation impairments in patients with parkinson's disease. In *Interspeech* (2017), pp. 314–318.

[335] Wagh, S., Gupta, D., and Chandran, N. SecureNN: Efficient and Private Neural Network Training. *IACR Cryptol. ePrint Arch. 2018* (2018), 442.

[336] Wagh, S., Tople, S., Benhamouda, F., Kushilevitz, E., Mittal, P., and Rabin, T. Falcon: Honest-majority maliciously secure framework for private deep learning. *Proceedings on Privacy Enhancing Technologies 1* (2021), 188–208.

[337] Wang, J., Ravi, V., and Alwan, A. Non-uniform Speaker Disentanglement For Depression Detection From Raw Speech Signals. In *Proc. Interspeech 2023* (2023), pp. 2343–2347.

[338] Wang, Q., Downey, C., Wan, L., Mansfield, P. A., and Moreno, I. L. Speaker diarization with LSTM. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), IEEE, pp. 5239–5243.

[339] Wang, Q., Feng, C., Xu, Y., Zhong, H., and Sheng, V. S. A novel privacy-preserving speech recognition framework using bidirectional lstm. *Journal of Cloud Computing 9* (2020), 1–13.

[340] Wang, Q., Guo, P., and Xie, L. Inaudible Adversarial Perturbations for Targeted Attack in Speaker Recognition. In *Proc. Interspeech 2020* (2020), pp. 4228–4232.

[341] Wang, W.-C., De Coninck, S., Leroux, S., and Simoens, P. An opt-in framework for privacy protection in audio-based applications. *IEEE Pervasive Computing 21*, 4 (2022), 17–24.

[342] Wang, X., Yamagishi, J., Todisco, M., Delgado, H., Nautsch, A., Evans, N., Sahidullah, M., Vestman, V., Kinnunen, T., Lee, K. A., et al. Asvspoof 2019: A large-scale public database of synthesized, converted and replayed speech. *Computer Speech & Language 64* (2020), 101114.

[343] Wang, Y., He, M., Niu, S., Sun, L., Gao, T., Fang, X., Pan, J., Du, J., and Lee, C.-H. USTC-NELSLIP System Description for DIHARD-III Challenge. *arXiv preprint arXiv:2103.10661* (2021).

[344] Warren, S. D., and Brandeis, L. D. The right to privacy. *Harvard Law Review 4*, 5 (1890), 193–220.

[345] Watanabe, S., Hori, T., Karita, S., Hayashi, T., Nishitoba, J., Unno, Y., Enrique Yalta Soplin, N., Heymann, J., Wiesner, M., Chen, N., Renduchintala, A., and Ochiai, T. ESPnet: End-to-end speech processing toolkit. In *Proc. of Interspeech, 2018* (2018), pp. 2207–2211.

[346] Westin, A. F. Privacy and freedom. *Washington and Lee Law Review 25*, 1 (1968), 166.

[347] Woubie, A., and Bäckström, T. Federated learning for privacy preserving on-device speaker recognition. In *Proc. 2021 ISCA Symposium on Security and Privacy in Speech Communication* (2021), pp. 1–5.

[348] Woubie, A., and Bäckström, T. Federated learning for privacy-preserving speaker recognition. *IEEE Access 9* (2021), 149477–149485.

[349] Wu, D.-Y., and Lee, H.-y. One-shot voice conversion by vector quantization. In *Proc. ICASSP* (2020), IEEE, pp. 7734–7738.

[350] Wu, H., Zhang, Y., Wu, Z., Wang, D., and yi Lee, H. Voting for the Right Answer: Adversarial Defense for Speaker Verification. In *Proc. Interspeech 2021* (2021), pp. 4294–4298.

[351] Wu, P., Liang, P. P., Shi, J., Salakhutdinov, R., Watanabe, S., and Morency, L.-P. Understanding the tradeoffs in client-side privacy for downstream speech tasks. In *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)* (2021), IEEE, pp. 841–848.

[352] Wu, Z., Evans, N., Kinnunen, T., Yamagishi, J., Alegre, F., and Li, H. Spoofing and countermeasures for speaker verification: A survey. *Speech Communication 66* (2015), 130–153.

[353] Xie, Y., Shi, C., Li, Z., Liu, J., Chen, Y., and Yuan, B. Real-time, universal, and robust adversarial attacks against speaker recognition systems. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (May 2020).

[354] Yamagishi, J., Wang, X., Todisco, M., Sahidullah, M., Patino, J., Nautsch, A., Liu, X., Lee, K. A., Kinnunen, T., Evans, N., et al. ASVspoof 2021: accelerating progress in spoofed and deepfake speech detection. In *ASVspoof 2021 Workshop-Automatic Speaker Verification and Spoofing Countermeasures Challenge* (2021).

[355] Yang, C.-H. H., Chen, I.-F., Stolcke, A., Siniscalchi, S. M., and Lee, C.-H. An experimental study on private aggregation of teacher ensemble learning for end-to-end speech recognition. In *2022 IEEE Spoken Language Technology Workshop (SLT)* (2023), pp. 1074–1080.

[356] Yao, A. C. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)* (1986), pp. 162–167.

[357] Ye, J., and al. Enhanced membership inference attacks against machine learning models. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (2022), pp. 3093–3106.

[358] Yin, R., Bredin, H., and Barras, C. Speaker Change Detection in Broadcast TV Using Bidirectional Long Short-Term Memory Networks. In *Proc. Interspeech 2017* (2017), pp. 3827–3831.

[359] Yoo, I.-C., Lee, K., Leem, S., Oh, H., Ko, B., and Yook, D. Speaker anonymization for personal information protection using voice conversion techniques. *IEEE Access 8* (2020), 198637–198645.

[360] Yu, D., Gong, Y., Picheny, M. A., Ramabhadran, B., Hakkani-Tür, D., Prasad, R., Zen, H., Skoglund, J., Černockỳ, J. H., Burget, L., et al. Twenty-five years of evolution in speech and language processing. *IEEE Signal Processing Magazine 40*, 5 (2023), 27–39.

[361] Zahur, S., Rosulek, M., and Evans, D. Two halves make a whole: Reducing data transfer in garbled circuits using half gates. *Cryptology ePrint Archive, Report 2014/756* (2014).

[362] Zen, H., Dang, V., Clark, R., Zhang, Y., Weiss, R. J., Jia, Y., Chen, Z., and Wu, Y. LibriTTS: A Corpus Derived from LibriSpeech for Text-to-Speech. In *Proc. Interspeech 2019* (2019), pp. 1526–1530.

[363] Zhang, S., Dou, W., and Yang, H. MDCT sinusoidal analysis for audio signals analysis and processing. *IEEE Transactions on Audio, Speech, and Language Processing 21*, 7 (2013), 1403–1414.

[364] Zhang, X.-L., and Wang, D. Boosting contextual information for deep neural network based voice activity detection. *IEEE/ACM Transactions on Audio, Speech, and Language Processing 24*, 2 (2015), 252–264.

[365] Zhang, Y., Lv, Z., Wu, H., Zhang, S., Hu, P., Wu, Z., yi Lee, H., and Meng, H. MFA-Conformer: Multi-scale Feature Aggregation Conformer for Automatic Speaker Verification. In *Proc. Interspeech 2022* (2022), pp. 306–310.

[366] Zhang, Z., Yu Zhang, L., Zheng, X., Hussain Abbasi, B., and Hu, S. Evaluating Membership Inference Through Adversarial Robustness. *The Computer Journal 65*, 11 (2022), 2969–2978.

A

General Data Protection Regulation – Relevant Articles and Definitions

A.1 Recital 40 - Lawfulness of data processing

In order for processing to be lawful, personal data should be processed on the basis of the consent of the data subject concerned or some other legitimate basis, laid down by law, either in this Regulation or in other Union or Member State law as referred to in this Regulation, including the necessity for compliance with the legal obligation to which the controller is subject or the necessity for the performance of a contract to which the data subject is party or in order to take steps at the request of the data subject prior to entering into a contract.

A.2 Recital 78 - Appropriate technical and organisational measures

When developing, designing, selecting and using applications, services and products that are based on the processing of personal data or process personal data to fulfil their task, producers of the products, services and applications should be encouraged to take into account the right to data protection when developing and designing such products, services and applications and, with due regard to the state of the art, to make sure that controllers and processors are able to fulfil their data protection obligations.

A.3 Article 4. – Definition of personal data

(...) 'personal data' means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the ***physical***, ***physiological***, *genetic,* ***mental***, *economic,* ***cultural*** *or* ***social identity*** *of that natural person;*

A.4 Article 5. – Definition of personal data

1. *Personal data shall be*:
   (a) *processed lawfully, fairly and in a transparent manner in relation to the data subject ('lawfulness, fairness and transparency')*;
   (b) *collected for specified, explicit and legitimate purposes and not further processed in a manner that is incompatible with those purposes; further processing for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes shall, in accordance with Article 89(1), not be considered to be incompatible with the initial purposes ('purpose limitation')*;
   (c) *adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed ('data minimisation')*;
   (d) *accurate and, where necessary, kept up to date; every reasonable step must be taken to ensure that personal data that are inaccurate, having regard to the purposes for which they are processed, are erased or rectified without delay ('accuracy')*;

(e) *kept in a form which permits identification of data subjects for no longer than is necessary for the purposes for which the personal data are processed; personal data may be stored for longer periods insofar as the personal data will be processed solely for archiving purposes in the public interest, scientific or historical research purposes or statistical purposes in accordance with Article 89(1) subject to implementation of the appropriate technical and organisational measures required by this Regulation in order to safeguard the rights and freedoms of the data subject ('storage limitation')*;

(f) *processed in a manner that ensures appropriate security of the personal data, including protection against unauthorised or unlawful processing and against accidental loss, destruction or damage, using appropriate technical or organisational measures ('integrity and confidentiality')*.

2. *The controller shall be responsible for, and be able to demonstrate compliance with, paragraph 1 ('accountability')*.

A.5 Recital 26 - Not Applicable to Anonymous Data

[1] The principles of data protection should apply to any information concerning an identified or identifiable natural person. [2] Personal data which have undergone pseudonymisation, which could be attributed to a natural person by the use of additional information should be considered to be information on an identifiable natural person. [3] To determine whether a natural person is identifiable, account should be taken of all the means reasonably likely to be used, such as singling out, either by the controller or by another person to identify the natural person directly or indirectly. [4] To ascertain whether means are reasonably likely to be used to identify the natural person, account should be taken of all objective factors, such as the costs of and the amount of time required for identification, taking into consideration the available technology at the time of the processing and technological developments. [5] The principles of data protection should therefore not apply to anonymous information, namely information which does not relate to an identified or identifiable natural person or to personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable. [6] This Regulation does not therefore concern the processing of such anonymous information, including for statistical or research purposes.

B

MPC Proofs

B.1 Local Secret Sharing Addition

Proof. Assume a set of two parties, p_1 and p_2, that want to add the input values x and y to obtain $z = x + y$, with x being held by party p_1 and y by p_2. Each party generates a random value r_i, with i the number of the party. To secret-share these values, p_1 sets $\langle x \rangle_1 = r_1$ and $\langle x \rangle_2 = x - r_1$. Similarly, p_2 sets $\langle y \rangle_2 = r_2$ and $\langle y \rangle_1 = y - r_2$. The two parties then exchange the shares corresponding to the other party, such that, after this step, p_1 holds $\langle x \rangle_1 = r_1$ and $\langle y \rangle_1 = y - r_2$, and p_2 holds $\langle y \rangle_2 = r_2$, $\langle x \rangle_2 = x - r_1$.

To add the two shares, each party just needs to add the shares of the two values that they hold. Specifically, p_1 obtains $\langle z \rangle_1 = r_1 + y - r_2$, and p_2 obtains $\langle z \rangle_2 = r_2 + x - r_1$.

To reconstruct z, each parties require the other's share of $\langle z \rangle$. Assuming these shares are exchanged,

each party can reconstruct z by adding the two resulting shares. This would correspond to:

$$\begin{aligned}\langle z\rangle &= \langle z\rangle_1 + \langle z\rangle_2 \\ &= r_1 + y - r_2 + r_2 + x - r_1 \\ &= x + y + \cancel{(r_1 + r_2)} - \cancel{(r_1 + r_2)} \\ &= x + y\end{aligned}$$

□

Even though this proof corresponds to the 2-party case, it is easily generalisable to the n-party case. Moreover, this example is only a toy problem. If one party has one of the terms of the addition and the resulting value, it can learn the other party's input value, and this protocol will not provide any security or privacy guarantees. However, since secret-sharing operations are composable, the full computation might not reveal the individual value of each input.

B.2 Multiplication Triples

Proof. Assume a set of N parties that want to multiply values x and y, secret-shared as $\langle x\rangle$ and $\langle y\rangle$, respectively. Further assume we have pre-computed shares $\langle a\rangle$, $\langle b\rangle$ and $\langle c\rangle = \langle a\rangle \times \langle b\rangle$, so called Multiple Triples (MT). All parties start by setting $\langle e\rangle_i = \langle x\rangle_i - \langle a\rangle_i$ and $\langle f\rangle_i = \langle y\rangle_i - \langle b\rangle_i$, and exchange the results with the other parties, such that each party holds:

$$f = \sum_{i=1}^{N}\langle f\rangle_i = \sum_{i=1}^{N}\langle y\rangle_i - \langle b\rangle_i = \sum_{i=1}^{N}\langle y\rangle_i - \sum_{i=1}^{N}\langle b\rangle_i = y - b \tag{B.1}$$

and

$$e = \sum_{i=1}^{N}\langle e\rangle_i = \sum_{i=1}^{N}\langle x\rangle_i - \langle a\rangle_i = \sum_{i=1}^{N}\langle x\rangle_i - \sum_{i=1}^{N}\langle a\rangle_i = x - a \tag{B.2}$$

Each party will then set their share of $z = x \times y$ as:

$$\langle z\rangle_i = \mathbb{1}_{[i=1]} e \cdot f + f \cdot \langle a\rangle_i + e \cdot \langle b\rangle_i + \langle c\rangle_i \tag{B.3}$$

where $\mathbb{1}_{[i=1]}$ is the indicator function, to denote that the term in question is only added to the secret share by one party.

By adding all $\langle z\rangle_i$, we can then show:

$$
\begin{aligned}
z = \sum_{i=1}^{N} \langle z \rangle_i &= e \cdot f + \sum_{i=1}^{N} f \cdot \langle a \rangle_i + e \cdot \langle b \rangle_i + \langle c \rangle_i \\
&= e \cdot f + f \cdot \sum_{i=1}^{N} \langle a \rangle_i + e \cdot \sum_{i=1}^{N} \langle b \rangle_i + \sum_{i=1}^{N} \langle c \rangle_i \\
&= e \cdot f + f \cdot a + e \cdot b + c \\
&= (x - a) \cdot (y - b) + (y - b) \cdot a + (x - a) \cdot b + a \cdot b \\
&= x \cdot y - x \cdot b - y \cdot a + a \cdot b + y \cdot a - a \cdot b + x \cdot b - a \cdot b + a \cdot b \\
&= x \cdot y - \cancel{x \cdot b} - \cancel{y \cdot a} + \cancel{a \cdot b} + \cancel{y \cdot a} - \cancel{a \cdot b} + \cancel{x \cdot b} - \cancel{a \cdot b} + \cancel{a \cdot b} \\
&= x \cdot y
\end{aligned}
$$

□

Although the present proof is shown with regard to the arithmetic domain, it generalises to the boolean domain, being only necessary to replace additions and subtractions with the XOR operator, and multiplications with the AND operator.

B.3 Replicated Secret Sharing - Local Multiplication

Proof. Assume a set of three parties, that want to multiply values x and y, secret-shared under replicated secret sharing scheme as $\langle x \rangle$ and $\langle y \rangle$, respectively.
A possible implementation of the multiplication operation would be for each party to locally multiply the shares it holds for each of the secret shared values.
In this way, party p_1 will obtain $\langle z \rangle_1 = \langle x \rangle_1 \langle y \rangle_1 + \langle x \rangle_1 \langle y \rangle_2 + \langle x \rangle_2 \langle y \rangle_1$; party p_2, $\langle z \rangle_2 = \langle x \rangle_2 \langle y \rangle_2 + \langle x \rangle_2 \langle y \rangle_3 + \langle x \rangle_3 \langle y \rangle_2$ and party p_3, $\langle z \rangle_3 = \langle x \rangle_3 \langle y \rangle_3 + \langle x \rangle_3 \langle y \rangle_1 + \langle x \rangle_1 \langle y \rangle_3$. After these values are obtained for each party, to return to the previous state where each party holds two shares, each party simply sends its share to one of the remaining two parties.
We can then show that by adding the shares of each party we obtain $z = x \times y$:

$$
\begin{aligned}
z &= \langle z \rangle_1 + \langle z \rangle_2 + \langle z \rangle_3 \\
&= \langle x \rangle_1 \langle y \rangle_1 + \langle x \rangle_1 \langle y \rangle_2 + \langle x \rangle_2 \langle y \rangle_1 + \langle x \rangle_2 \langle y \rangle_2 + \langle x \rangle_2 \langle y \rangle_3 + \langle x \rangle_3 \langle y \rangle_2 + \langle x \rangle_3 \langle y \rangle_3 + \langle x \rangle_3 \langle y \rangle_1 + \langle x \rangle_1 \langle y \rangle_3 \\
&= \langle x \rangle_1 \times (\langle y_1 \rangle + \langle y \rangle_2 + \langle y \rangle_3) + \langle x \rangle_2 \times (\langle y_1 \rangle + \langle y \rangle_2 + \langle y \rangle_3) + \langle x \rangle_3 \times (\langle y_1 \rangle + \langle y \rangle_2 + \langle y \rangle_3) \\
&= \langle x \rangle_1 \times y + \langle x \rangle_2 \times y + \langle x \rangle_3 \times y \\
&= (\langle x \rangle_1 + \langle x \rangle_2 + \langle x \rangle_3) \times y \\
&= x \times y
\end{aligned}
$$

□

C

Mutual Information Estimators

In this Appendix, we describe the two Mutual Information (MI) estimators used in this work: (1) the Kraskov, Stögbauer and Grassberger (KSG) [105, 154] estimator to estimate the MI between two continuous random variables; (2) the MI estimator proposed by B. Ross [275], for mixtures of discrete and continuous random variables. The descriptions contained in this Appendix closely follow the method descriptions presented in [154, 275].

C.1 Mutual information estimator for continuous random variables

We will start by providing a high-level description of the continuous-continuous KSG mutual information estimator [154] and the intuition behind this estimator. Although it is only used for the manipulation of a continuous attribute (i.e., age), understanding this estimator will allow the reader to understand the intuition behind nearest-neighbour MI estimators and consequently understand the continuous-discrete MI estimator proposed by B. Ross [275].

The mutual information $I(Z, Y)$ between two continuous variables Z and Y can be expressed in terms

of the individual differential entropies and the entropy between the two random variables:

$$I(Z,Y) = H(Z) + H(Y) - H(Z,Y), \tag{C.1}$$

having each $H(\cdot)$ defined as:

$$H(S) = E[-\log \mu_s(s)] = -\frac{1}{N}\sum_{i=1}^{N} \log \mu_s(s_i), \tag{C.2}$$

where S is any random variable and μ_s is its corresponding the probability density function.

Given a set of N observations taken from dataset $\mathcal{D}$ of the joint variable $M = (Z,Y)$, $m_i = (z_i, y_i)$, with $i \in 1 \dots N$, the goal of an MI estimator is to use these observations to obtain $I(Z,Y)$.

From eq. (C.1), it is possible to see that the MI can be computed through its entropy terms. However, it is not possible to compute these terms directly because $\mu_z(z)$, $\mu_y(y)$ and $\mu_{z,y}(z,y))$ are unknown. Instead, one needs to leverage the observations and use them to estimate the value of each entropy term.

To do so, KSG applies the Kozachenko-Leonenko (KL) [153] k-nearest neighbour entropy estimator. This estimator works by defining a probability distribution $P_k(\epsilon)$ of the distance $(\epsilon/2)$ between each sample s_i – sampled from a continuous random variable S – and its k^{th} neighbour.

Let us consider that each p_i corresponds to the mass of a d_S-dimensional ϵ-ball around s_i, where d_S is the dimensionality of S. The KL estimator leverages the fact that, by estimating $p_i(\epsilon)$, it is possible to indirectly estimate the density $\mu_s(s_i)$ (assuming it is constant within the entire ϵ-ball), since, by definition:

$$\mu_s(s_i) \approx \frac{p_i(\epsilon)}{v_{d_s}\epsilon^{d_s}} \tag{C.3}$$

where v_{d_s} is the volume of the d_S-dimensional unit ball, and ϵ its radius. $v_{d_S} = 1$ for the maximum norm, and $v_{d_S} = \pi^{\frac{d_S}{2}}/\Gamma(\frac{d_S}{2}+1)$ for the l_2 norm, with $\Gamma(\cdot)$ corresponding to the *gamma* function.

Considering that ϵ_i^d can be computed for each sample s_i – it corresponds to twice the distance between s_i and its k^{th} neighbour – to obtain the density it is only necessary to further compute $p_i(\epsilon)$. However, what is required is the expected value of $\mu_s(s_i)$. For this reason, in KL the expected value of $\log(p_i)$ is computed directly [153, 154]:

$$E[\log(p_i)] = \psi(k) - \psi(N) \tag{C.4}$$

with k being the pre-defined number of neighbours, N the number of observations, and $\psi(\cdot)$ the *digamma* function [2].

Combining eqs. (C.2), (C.3) and (C.4), one obtains the full KL estimator:

$$\hat{H}(S) = \psi(N) - \psi(k) + \log(v_{d_s}) + \frac{d_s}{N}\sum_{i=1}^{N} \log(\epsilon_i) \tag{C.5}$$

This can be extended to the joint random variable $M = (Z, Y)$, as:

$$\hat{H}(X,Y) = \psi(N) - \psi(k) + \log\left(v_{d_Z} v_{d_Y}\right) + \frac{d_Z + d_Y}{N} \sum_{i=1}^{N} \log \epsilon_i, \tag{C.6}$$

where v_{d_Z} and v_{d_Y} correspond to the volume of the d_Z and d_Y-dimensional unit balls and $\epsilon_i/2$ corresponds to the distance between two observations in the joint space Z.

To obtain $I(Z, Y)$ one could simply apply eqs. (C.5) and (C.6). However, the distance scales of the joint space Z, and variables Z and Y may be very different. To circumvent this issue, the KSG estimator (specifically, Algorithm (2) of [154]) first finds the k^{th} neighbour of sample m_i in the joint space M, with distance $\epsilon_i/2$, using the maximum norm $\|m - m'\| = \max\{\|z - z'\|, \|y - y'\|\}$, for any metric space in X or Y. It then considers the number of points n_{s_i} that are within distance $\epsilon_{s_i}/2$ for each of the marginal sub-spaces of Z and Y, as a replacement of the original fixed number of neighbours k. This yields a second estimator $\hat{H}(S)$ for the differential entropies:

$$\hat{H}(S) = \psi(N) - \frac{1}{N} \sum_{i=1}^{N} \psi(n_{s_i} + 1) - \log\left(v_{d_S}\right) - \frac{d_S}{N} \sum_{i=1}^{N} \log \epsilon_{s_i}, \tag{C.7}$$

where S corresponds to either Z or Y. Finally, by combining equations (C.6) and (C.7), results in:

$$\hat{I}(Z,Y) = \psi(k) + \psi(N) - \langle \psi(n_z + 1) + \psi(n_y + 1) \rangle, \tag{C.8}$$

where $\langle ... \rangle = \frac{1}{N} \sum_{i=1}^{N} ...$ is the average operator.

In our preliminary experiments, we found that this estimator was not able to perform well when large differences in the dimensionality of each marginal space occurred, or when very different scales of X and Y were present, a result that is consistent with what is reported in the literature [104]. Instead, we used the adapted estimator of Gao et al. [105], which introduces a bias-correction term that accounts for the volumes in each dimension, and that uses the l_2 distance instead of the maximum norm [105]:

$$\hat{I}(Z,Y) = \log(N) + \psi(k) + \log \frac{v_z v_y}{v_z + v_y} - \langle \log(n_z) + \log(n_y) \rangle, \tag{C.9}$$

C.2 Mutual information estimator for discrete and continuous random variables

The continuous-discrete MI estimator proposed by Ross [275] applies a similar idea to that of Kraskov et al. [154], leveraging the k-nearest neighbour KL entropy estimator [153].

From eq. (C.1), it can be shown that for a discrete random variable Y, and a continuous random

variable Z [275]:

$$I(X,Y) = -\langle \log \mu_z(z) \rangle + \langle \log \mu_{z|y}(z|y) \rangle. \tag{C.10}$$

Using this, the author then applies the KL differential entropy estimator (cf. eq. (C.5)) twice, to estimate each term. This leads to:

$$\hat{I}(z_i, y_i) = \psi(N) + \psi(k) - \psi(N_{y_i}) - \psi(n_{z_i}), \tag{C.11}$$

where $I(z_i, y_i)$ is the mutual information for a single observation (z_i, y_i), and where N_{y_i} corresponds to number of samples in $\mathcal{D}$ with the same discrete value y_i. This is relevant as it shows that the notion of neighbour changes from the previous estimator, and instead a sample is only considered a "neighbour" if it comes from the subset of $\mathcal{D}$ where $Y = y_i$. For this reason, $\epsilon/2$ is set as the distance between z_i and the k^{th} sample that shares the same value y_i, and n_{z_i} is counted as the number of samples, now for the full set of $\mathcal{D}$, that are within this distance.
Finally, to compute the MI for the full set of samples, one computes the average of all $I_i(z_i, y_i)$:

$$\hat{I}(X,Y) = \psi(N) + \psi(k) - \langle \psi(N_y) \rangle - \langle \psi(n_z) \rangle. \tag{C.12}$$