Joint source-cryptographic-channel coding for real-time secure voice communications on voice channels

Piotr Krasnowski

UNIVERSITÉ CÔTE D'AZUR

ÉCOLE DOCTORALE
SCIENCES ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

THÈSE DE DOCTORAT

Codage conjoint source-chiffrement-canal pour les canaux de communication vocaux sécurisés en temps réel

Piotr KRASNOWSKI

Laboratoire d'Informatique, Signaux et Systèmes de Sophia Antipolis (I3S)
UMR7271 Université Côte d'Azur CNRS

Présentée en vue de l'obtention du grade de docteur en Informatique **de** l'Université Côte d'Azur

Dirigée par :
Bruno MARTIN, Professeur
Co-encadrée par :
Jerome LEBRUN, Chercheur CNRS

Soutenue le : 27 Mai 2021

Devant le jury, composé de :
J.-C. RÉGIN, Professeur, Univ. Côte d'Azur
W. PUECH, Professeur, Univ. de Montpellier
T. van WATERSCHOOT, Professeur, KU Leuven
J. LEBRUN, Chargé de Recherche, CNRS
B. MARTIN, Professeur, Univ. Côte d'Azur
M. PELLEAU, Maître de Conférences, Univ. Côte d'Azur
T. PLESSE, Expert, Direction Générale de l'Armement
J.-B. YUNÈS, Maître de Conférences HDR, Univ. de Paris
P. ADENOT, Ingénieur de Recherche, Mozilla

Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE

DGA

MINISTÈRE DES ARMÉES

AGENCE INNOVATION DÉFENSE

JOINT SOURCE-CRYPTOGRAPHIC-CHANNEL CODING FOR REAL-TIME SECURE VOICE COMMUNICATIONS ON VOICE CHANNELS

CODAGE CONJOINT SOURCE-CHIFFREMENT-CANAL POUR LES CANAUX DE COMMUNICATION VOCAUX SÉCURISÉS EN TEMPS RÉEL

Piotr KRASNOWSKI

Jury :

Président du jury
Jean-Charles RÉGIN, Professeur des Universités, Université Côte d'Azur

Rapporteurs
William PUECH, Professeur des Universités, Université de Montpellier
Toon van WATERSCHOOT, Professeur d'Établissement Étranger, KU Leuven, Belgique

Examinateurs
Jerome LEBRUN, Chargé de Recherche, CNRS
Bruno MARTIN, Professeur des Universités, Université Côte d'Azur
Marie PELLEAU, Maître de Conférences, Université Côte d'Azur
Thierry PLESSE, Expert, Direction Générale de l'Armement
Jean-Baptiste YUNÈS, Maître de Conférences HDR, Université de Paris

Membres invités
Paul ADENOT, Ingénieur de Recherche, Mozilla

Université Côte d'Azur

Codage conjoint source-chiffrement-canal pour les canaux de communication vocaux sécurisés en temps réel

Résumé

Les risques croissants de violation de la vie privée et d'espionnage associés à la forte croissance des communications mobiles ont ravivé l'intérêt du concept originel de chiffrement de la parole sous forme de signaux audio transmis sur des canaux vocaux non spécifiques. Les méthodes habituelles utilisées pour la transmission de données cryptées par téléphonie analogique se sont révélées inadaptées pour les communications vocales modernes (réseaux cellulaires, VoIP) avec leurs algorithmes de compression de la voix, de détection d'activité vocale et de suppression adaptative du bruit. La faible bande passante disponible, les distorsions non linéaires des canaux et les phénomènes d'évanouissements du signal motivent l'introduction d'une approche conjointe du codage et du chiffrement de la parole adaptée aux distorsions introduites par les canaux vocaux modernes.

Dans cette thèse sont développés, analysés et validés divers schémas sûrs et efficaces pour le chiffrement et la transmission de la parole en temps réel pour les canaux vocaux modernes. En plus du chiffrement de la parole, cette étude couvre les aspects sécurité et algorithmique de l'ensemble du système de communication vocale - aspects critiques d'un point de vue industriel.

La thèse détaille un système de chiffrement de la parole associé à un codage avec perte, par brouillage aléatoire des paramètres vocaux (volume, hauteur, timbre) de certaines représentations de la parole. En résulte un pseudo-signal vocal chiffré robuste aux erreurs ajoutées par les canaux de transmission modernes. La technique de chiffrement repose sur l'introduction de translations et rotations aléatoires sur des maillages de tores plats associés à des codes sphériques. Face aux erreurs de transmission, le schéma déchiffre approximativement les paramètres vocaux et reconstruit, grâce à un synthétiseur vocal utilisant un réseau de neurones par apprentissage, un signal de parole perceptuellement très proche du signal d'origine. Le dispositif expérimental a été validé par la transmission de signaux de type pseudo-voix chiffrés sur un canal vocal réel. Les signaux de parole déchiffrés ont été favorablement notés lors d'une évaluation subjective de qualité incluant environ 40 participants.

La thèse décrit également une nouvelle technique de transmission de données sur canaux vocaux en utilisant un dictionnaire d'ondes harmoniques courtes représentant les mots d'un code quaternaire. La technique fournit un débit binaire variable allant jusqu'à 6.4 kbps et a été testée avec succès sur différents canaux vocaux réels. Enfin, est présenté aussi un protocole d'échange de clés cryptographiques dédié pour les canaux vocaux authentifiés par signatures et vérification vocale. La sécurité du protocole a été vérifiée sous forme d'un modèle symbolique par l'assistant de preuve formelle Tamarin.

L'étude conclut qu'une communication vocale sécurisée sur des canaux vocaux numériques réels est techniquement et de fait viable lorsque les canaux vocaux utilisés pour la communication sont suffisamment stables et ne présentent que des distorsions prévisibles.

Mots-clés : Communications vocales sécurisées, Données sur les canaux vocaux, Chiffrement de la voix, Codage conjoint de la parole, Vérification formelle, Sécurité sémantique.

Joint source-cryptographic-channel coding for real-time secure voice communications on voice channels

Abstract

The growing risk of privacy violation and espionage associated with the rapid spread of mobile communications renewed interest in the original concept of sending encrypted voice as audio signal over arbitrary voice channels. The usual methods used for encrypted data transmission over analog telephony turned out to be inadequate for modern vocal links (cellular networks, VoIP) equipped with voice compression, voice activity detection, and adaptive noise suppression algorithms. The limited available bandwidth, nonlinear channel distortion, and signal fadings motivate the investigation of a dedicated, joint approach for speech encoding and encryption adapted to the distortion introduced by modern voice channels.

This thesis aims to develop, analyze, and validate secure and efficient schemes for real-time speech encryption and transmission via modern voice channels. In addition to speech encryption, this study covers the security and operational aspects of the whole voice communication system, as this is relevant from an industrial perspective.

The thesis introduces a joint speech encryption scheme with lossy encoding, which randomly scrambles the vocal parameters of some speech representation (loudness, pitch, timbre) and outputs an encrypted pseudo-voice signal robust against channel distortion. The enciphering technique is based on random translations and random rotations using lattices and spherical codes on flat tori. Against transmission errors, the scheme decrypts the vocal parameters approximately and reconstructs a perceptually analogous speech signal with the help of a trained neural-based voice synthesizer. The experimental setup was validated by sending encrypted pseudo-voice over a real voice channel, and the decrypted speech was tested using subjective quality assessment by a group of about 40 participants.

Furthermore, the thesis describes a new technique for sending data over voice channels that relies on short harmonic waveforms representing quaternary codewords. This technique achieves a variable bitrate up to 6.4 kbps and has been successfully tested over various real voice channels. Finally, the work considers a dedicated cryptographic key exchange protocol over voice channels authenticated by signatures and a vocal verification. The protocol security has been verified in a symbolic model using Tamarin Prover.

The study concludes that secure voice communication over real digital voice channels is technically viable when the voice channels used for communication are stationary and introduce distortion in a predictable manner.

Keywords: Secure Voice Communications, Data over Voice Channels, Voice Encryption, Joint Speech Coding, Formal Verification, Semantic Security.

Acknowledgements

Undertaking this PhD has been a truly life-forming experience that broadened my horizons and profoundly changed my view of the world. However, it would not have been possible without many people who provided me the guidance and support.

First of all, I am deeply grateful to the Agence de l'Innovation de Défense and the Direction Générale de l'Armement for the funding needed to pursue my research and the financial and administrative support when the future of my PhD project was at risk.

I would like to sincerely thank my joint supervisors, Professor Bruno Martin and Dr.-Sc. Jerome Lebrun, for being my real academic and life mentors. I cannot fully express my gratitude for your wise scientific guidance and your help in solving all the hurdles I encountered during my studies. Thank you very much for every inspiring discussion, your encouraging words in challenging situations, and all your hard work until late night hours.

I gratefully appreciate the work of my tutor Thierry Plesse from the Direction Générale de l'Armement, who supervised the progress of my research, and whose unwavering support enabled me to continue this PhD.

I thank my former company advisor Arnaub Graube for his in-depth technical advice, his patience with explaining to me all the subtleties of engineering work, and for showing that solving any scientific problem does not end just by publishing a research article.

I express my gratefulness to Professor Cheon Jung-Hee and Professor Park Hyung-Ju for inviting me to Seoul National University. It was a great honor to work with you and a wonderful experience I will always keep in my memory.

I would like to thank all the professors and researchers I had a pleasure to collaborate with during my work at I3S: Professor Enrico Formenti, Dr. MCF Cinzia Di Giusto, Dr. MCF Sandrine Julia, Dr. MCF Marie Pelleau, Professor Jean-Charles Régin, and the PhD students: Marie Baillet, Samvel Balassanian Dersarkissian, François Doré, Rémy Garcia, Loïc Germerie, Laetitia Gibart, Assia Kamal-Idrissi, Nicolas Isoart, Laetitia Laversa, Ninad Manerikar, Sara Riva, and Giulia Rocco. I appreciate all the time we spent together in a friendly atmosphere. Without your help and advice, my PhD would have been far incomplete.

Many thanks to all my colleagues from I3S-CNRS and INRIA whom I met during my studies: Dmitry Anisimov, Eva Gil San Antionio, Jean-Philippe Bauchet, Denys Bulavka, Fernando Ireta, Franco Fusco, Nicolas Girard, Jean-Marie Kai, Lyes Khacef, Ivana Kojcic, Muxingzi Li, Pedro Marinho, Timothée O'Donnel, Cédric Portaneri, Siddharth Pritam, Flora Quilichini, Miguel Romero, Melissa Sanabria, Méliné Simsir, Onur Tasar, Carlos Jorge Zubiaga Peña. I was truly impressed by the vibrant, diverse, and inclusive community you have built. It has made my PhD such an exciting experience.

I offer my special thanks to Arnab Dey, Sardor Israilov, and Giulia Rocco for bringing hope during the sombre and solitary lockdown periods and helping me to re-discover the meaning of friendship.

Finally, I would like to thank my dear Polish friends: Małgorzata Drozd, Nikodem Dymski, Paweł Młynarski, and Katarzyna Tomasiak, with whom I spent many joyful moments on discussing the specificities of the French life and on exploring the beautiful French Riviera.

Publication List

Journal papers

— Krasnowski, P., Lebrun, J., and Martin, B. (2020). Introducing a Novel Data over Voice Technique for Secure Voice Communication. Submitted to Wireless Personal Communications. Springer. Preprint: https://arxiv.org/abs/2102.10869.

— Krasnowski, P., Lebrun, J., and Martin, B. (2021). Introducing an Experimental Distortion-Tolerant Speech Encryption for Secure Voice Communication. Submitted to Speech Communication, Elsevier. Preprint: https://arxiv.org/abs/2102.09809.

Conference paper

— Krasnowski, P., Lebrun, J., and Martin, B. (2020). Introducing a Verified Authenticated Key Exchange Protocol over Voice Channels for Secure Voice Communication. In 6th International Conference on Information Systems Security and Privacy (pp. 683-690). SCITEPRESS-Science and Technology Publications. DOI: 10.5220/0009156506830690.

Posters

— Piotr Krasnowski, Bruno Martin, Jerome Lebrun. Enciphered data/voice over real-time voice channels. European School of Information Theory (ESIT 2019), Apr. 2019, Sophia-Antipolis, France. <hal-02337668>

— Piotr Krasnowski. Introducing a Verified Authenticated Key Exchange Protocol over Voice Channels for Secure Voice Communications. 6th International Conference on Information Systems Security and Privacy, Feb. 2020, Valetta, Malta. <hal-03059639v2>

Presentations

— Piotr Krasnowski. Joint source-cryptographic-channel coding real-time secured voice communications on voice channels. Summer School on Real-World Crypto and Privacy, Jun. 2019, Sibenik, Croatia. <hal-02337657>

— Piotr Krasnowski. Secure voice communications over voice channels. Seoul National University, Dec. 2019, Seoul, South Korea. <hal-02561994>

List of contents

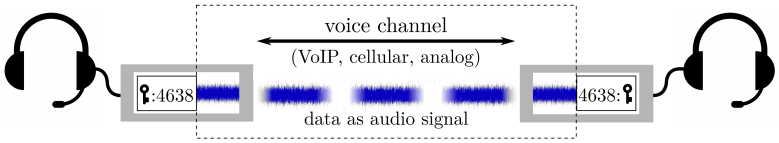
CHAPTER 1

Introduction

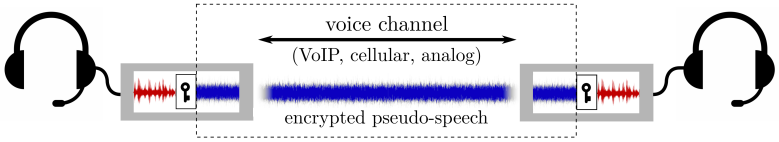
1.1 Problem outline

The mainstreaming of mobile networks opens new possibilities for personal communication. However, the rising numbers of reported privacy violations and cyber-espionage cases undermine confidence in the communication infrastructure. Another issue is inadequate security of many voice communication systems, such as GSM which encrypted voice traffic using the insecure A5/1 stream cipher with a 64-bit key [Biham and Dunkelman, 2000]. Low trust results in a growing need for alternative methods of securing vocal communication.

This work addresses the issue of secure voice communications over untrusted voice channels. The procedure for establishing a secure vocal link is illustrated in Figure 1.1. In the first step, two users carrying dedicated devices initiate an insecure call using a preferred communication technique, like cellular telephony, Voice over Internet Protocol, or fixed-line telephone circuits. Then, the two devices securely acknowledge their cryptographic keys by sending binary messages over the voice channel, the same way as ordinary voice. Once the cryptographic key is computed and authenticated, the speakers can start a secure conversation. Each device encrypts speech in real-time into an incomprehensible noise and sends the encrypted signal over the channel. Upon reception, the paired device decrypts the signal and restores the initial voice.



(a) Key exchange and authentication with short codes displayed on the devices.



(b) Encrypted conversation.

Figure 1.1 – Establishing a secure vocal link over a voice channel.

The outlined communication scheme involved into sending the encrypted audio is more complicated than exchanging encrypted bits over digital packet networks. Modern voice channels aim at preserving speech intelligibility at an acceptable speech quality degradation. This goal is accomplished by applying perceptual speech processing, such as voice compression, voice activity detection, and adaptive noise suppression. All these operations considerably modify the synthetic signal, hindering secure communication.

On the other hand, secure communication over voice channels is supposed to be more versatile because the encrypted audio signal can be made compatible with arbitrary communication infrastructure. Furthermore, the encrypted audio signal is more likely to pass through firewalls without being blocked [Lee et al., 2017]. Finally, the system can protect against spying malware installed on the portable device if speech encryption is done by an external unit [Krasnowski et al., 2020]. The mentioned advantages suggest that the proposed setting could be especially useful for diplomatic and military services, journalists, lawyers, and traders who require secure communications in an unreliable environment and without confidential communication infrastructure. Consequently, the system should reflect high security requirements by elevating the level of secrecy, privacy, and authentication.

1.2 Research rationale

Transforming speech into an encrypted signal is a multi-step process that consists of speech encoding, enciphering, appending redundancy for error protection, and signal synthesis. The classical and also the most straightforward approach is to treat each of these steps separately, as depicted in Figure 1.2. However, this approach is not suitable in secure voice communications because the information conveyed in speech is non-uniformly distributed in the time-frequency plane. Consequently, the system ignores the fact that some speech parameters are more relevant in terms of privacy and intelligibility, resulting in suboptimal bandwidth allocation and data protection. It is especially true in communication over a very low-bitrate vocal channel with distortion.

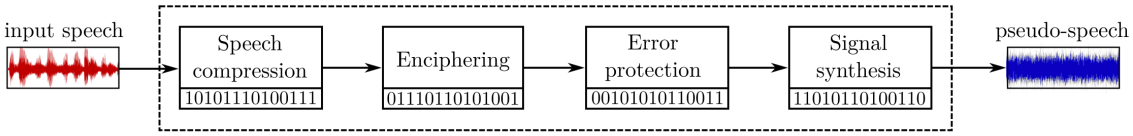


Figure 1.2 – Separated processing chain in digital speech encryption.

Firstly, the error-protection block does not prioritize the most relevant voice parameters needed to maintain the speech quality. In parallel, the speech encoder cannot be sensibly tuned to mitigate typical transmission errors introduced by the voice channel.

Secondly, classical cryptographic algorithms are unsuitable for enciphering speech as they usually require error-less data for decryption. In the discussed setting, however, transmission errors are unavoidable. On the contrary, one may observe that perfect data decryption is not mandatory, and small imperfections are likely to be perceptually irrelevant. Consequently, it would be advantageous to design a cryptographic scheme aware of the channel constraints and the unequal importance of vocal parameters.

The aforementioned interdependencies suggest that speech encoding, encryption, error-correction, and signal synthesis must be considered jointly. As a result, robust real-time communication over voice channels requires combining the processing blocks into a single unit, as shown in Figure 1.3.

This thesis aims to develop, analyze, and validate secure and efficient schemes for real-time speech encryption and transmission via real voice channels in the form of pseudo-voice in the audio domain. This multi-domain study combines elements from the fields of audio signal processing, cryptography, cybersecurity, and error-correcting codes, encouraging the joint approach for speech encryption schemes with lossy encoding and resistant to the distortion introduced by digital voice channels.

In addition to developing algorithms for speech encryption, this study covers the relevant aspects of the whole voice communication system from an industrial perspective. Cryptographic key management, key exchange protocols, and usage scenarios are important parts of this work.

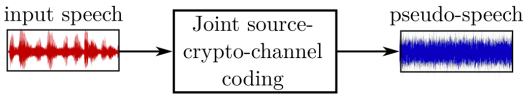


Figure 1.3 – Joint source-cryptographic-channel coding of speech.

1.3 Origins of secure voice communications

The history of secure voice communications is marked by various inventions that progressively shaped our modern understanding of secure communications. The first 'secure' speech scrambler patented in 1881 alternated speech signal between multiple telephone circuits at a high rate [Rogers, 1881]. The system did not offer high protection, and until the First World War was replaced by analog frequency inverters making the speech signal incomprehensible for an ordinary listener (Figure 1.4). Again, it turned out that trained operators could understand inverted speech, effectively breaking the secrecy of communication [Kak, 1983]. In the third attempt, engineers added analog band splitters for permuting speech subbands, as shown in Figure 1.5. The A-3 splitter installed by AT&T in 1937 for radiotelephone service used five subbands with $5! \cdot 2^5 = 3840$ combinations of permutations and inversions. However, only 11 codes were considered suitable for privacy, which was far too low to resist cryptanalysis [Kahn, 1996].

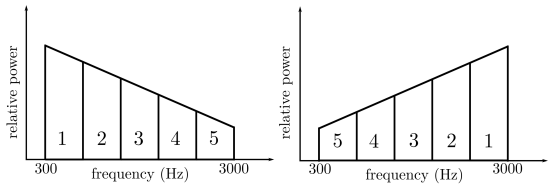


Figure 1.4 – Frequency inverter in which speech is band limited.

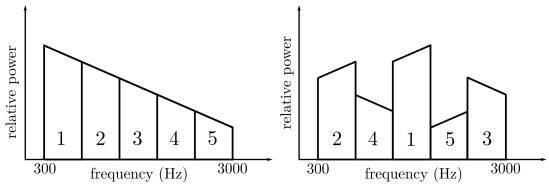


Figure 1.5 – Bandsplitting, and band permutation with band inversions.

The progress in analog speech scrambling moved towards higher permutation complexity and reached maturity by introducing sequentially updated time-frequency segment permutations (TFSP) [Jayant et al., 1983]. In these systems, speech was scrambled by combined time and frequency permutations using permutation matrices. Moreover, the matrices were frequently updated according to a random sequence to improve robustness to cryptanalysis. Despite these efforts, novel analog speech scramblers were consistently broken during the wartime period and later [Goldburg et al., 1993, Kahn, 1996, Zhao et al., 2007, Ghasemzadeh et al., 2014].

Another family of analog speech scramblers altered speech with additive or multiplicative masking noise that could be filtered out at the unscrambler (Figure 1.6) [Sugar, 1974, MacKinnon, 1980]. First systems superimposed masking tones or white noise [Sivian, 1928], whereas more advanced systems used chaotic maps [Kocarev et al., 1992]. Unfortunately, transmission errors and imperfect filters significantly degraded the speech quality at the receiving end. In consequence, designers faced a negative tradeoff between speech quality and secrecy.

The limitations of analog speech scrambling encouraged engineers to experiment with digital speech encryption. The inspiration was the famous Vernam cipher patented in 1919 [Vernam, 1919, Vernam, 1926], the electronic realization of an unbreakable 'one-time pad' used during the First World War for telegram communication. The initial idea was to encipher a digital representation of the speech samples with a random digital keystream (Figure 1.7) [MacKinnon, 1980]. While highly secure, the technique required wide bandwidth (12-18 kbps), error-less digital channels, which were prohibitively expensive at that time. The technical challenge remained unresolved until the late 30s when first voice coders (vocoders) were proposed [Dudley, 1939]. Vocoders enabled speech compression and reduction of encrypted data to be send.

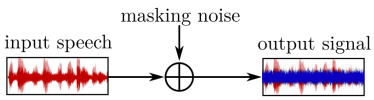


Figure 1.6 – Speech masking.

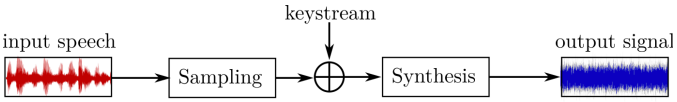


Figure 1.7 – Digital speech enciphering.

Probably the first secret telephony system using a vocoder was SIGSALY [1] ('X-System'), constructed in 1943 for secure communication during the Second World War [Bennett, 1983]. Figure 1.8 illustrates the block diagram of the transmitting part. The input speech was filtered by ten band splitters distributed nearly uniformly over the 150-3000 Hz range. The amplitudes of filtered streams were sampled by a six-level non-uniform quantizer, resulting ten trains of numbers between 0 and 5. Parallelly, a separate unit performed a two-step pitch prediction. The output was two numbers between 0 and 5 representing the coarse pitch value and the pitch refinement, or a single voicing bit.

The twelve trains of encoded values were independently enciphered with twelve trains of random six-valued numbers by a modulo-6 addition. The random numbers were obtained by continuously sampling white noise generated by a hot gas tube, based on the one-time-pad principle. The result of enciphering were twelve trains of randomly-looking numbers.

The last step of speech encryption involved signal synthesis. Each of the twelve trains had an assigned subchannel in the telephone band. Data transmission was done by modulating twelve frequency carriers, each carrier occupying one subchannel. The encoding and enciphering operations were repeated every 20 milliseconds, resulting in the 3.6 kbps transmission bitrate.

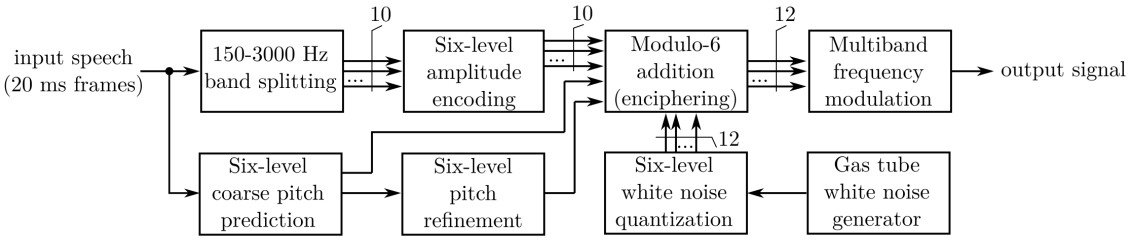


Figure 1.8 – Block diagram of a SIGSALY transmitter.

1. https://www.nsa.gov/about/cryptologic-heritage/historical-figures-publications/publications/wwii/sigsaly-story/

A successful deployment of SIGSALY, sometimes viewed as one of the most outstanding technical achievement, marked the beginning of the vocoder era in secure voice communications. Over the years, as more data bandwidth became available, the quality of speech synthesis was improved, and the price of computing units went down, vocoder-based speech communication became the predominant technique in the military and civilian applications. Around 1972, there were first attempts to replace secure voice communication over fading telephone lines with packet networks [Forgie, 1975], paving the way to modern secure VoIP communication.

1.4 Contributions

1.4.1 Data over Voice (DoV) technique

In Chapter 3, we propose a novel technique for data transmission over digital voice channels with Linear Predictive Coding (LPC), such as cellular networks and VoIP. The technique relies on codebooks of short waveforms with 7-10 phase-modulated harmonics over the 0 to 4 kHz audio band. The solution's novelty comes from the simplified codebook building process, which uses short quaternary error correction codes to determine appropriate phases of the harmonics. Moreover, the structure of the codebook of symbols enables many computational optimizations at the receiving side and partial compensation of channel distortion. The method is more versatile than other data transmission techniques over digital voice channels, often trained to a specific channel model. The possibility of variable transmission rate up to 6.4 kbps gives control over the robustness against channel distortion.

The DoV technique has been successfully validated in experiments with mobile phones and real-world voice channels (3G networks, WhatsApp, Skype, FaceTime, Signal Messenger). We achieved robust data transfer up to 2.4 kbps over 3G networks and 6.4 kbps over VoIP. The high throughput and the robustness enabled transmission of encrypted speech in a setting simulating real-time communication. As a result, the proposed method can be used in cryptographic key exchange and secure voice communications over voice channels.

The work has been presented in the article 'Introducing a Novel Data over Voice Technique for Secure Voice Communication' and submitted to the journal of Wireless Personal Communications, Springer.

1.4.2 Distortion-tolerant encryption of vectors on N-spheres

In Chapter 4, we describe a new method of enciphering unit vectors on a hypersphere that is robust against transmission error. The technique exploits spherical group codes and rotations from a commutative group of orthogonal matrices introduced by Slepian [Slepian, 1968] in the new context of securing data. The scrambled data are indistinguishable from uniformly distributed noise in the presence of an eavesdropper when the matrices are selected according to a secure pseudo-random sequence with fresh secret seed.

The presented encryption scheme decrypts data approximately despite channel distortion. It makes the scheme suitable for protecting voice or images in real-time applications that prioritize robustness over representation's fidelity. To describe the scheme's ability to decrypt mildly distorted ciphertexts, we defined a new notion of distortion-tolerant encryption.

The enciphering technique is an essential building block in the experimental speech encryption scheme described later in Chapter 5. The scheme is used to securely scramble multi-dimensional spectral envelopes, which are responsible for speech timbre perception.

1.4.3 Distortion-tolerant speech encryption

In Chapter 5, we present a novel distortion-tolerant speech encryption scheme for secure voice communications over voice channels that combines the robustness of analog speech scrambling and a higher security level offered by digital ciphers. The system scrambles vocal parameters of a speech signal (loudness, pitch, timbre) using random translations and the previously mentioned random rotations on a hypersphere of parameters. In the next step, randomized parameters are encoded to a pseudo-speech signal adapted to transmission over digital voice channels equipped with voice activity detection.

The use of translations and rotations in enciphering makes the speech decryption algorithm tolerant against moderate channel distortion. Upon reception of some pseudo-speech signal, the receiver restores distorted copies of the initial vocal parameters. Despite some deciphering errors, an integrated neural-based vocoder based on the LPCNet architecture [Valin and Skoglund, 2019] reconstructs an intelligible speech.

The experimental implementation of this speech encryption scheme has been tested by simulations and sending an encrypted signal over FaceTime between two iPhones 6 connected to the same WiFi network. Moreover, speech excerpts restored from encrypted signals were evaluated by a speech quality assessment on a group of about 40 participants. The experiments demonstrated that the proposed scheme produces intelligible speech with a gracefully progressive quality degradation depending on the channel distortion. Finally, the preliminary computational analysis suggested that the presented setting may operate on high-end portable devices in nearly real-time.

1.4.4 Authenticated Key Exchange over voice channels

In Chapter 6, we designed a two-party authenticated key exchange (AKE) protocol for secure voice communication over fading voice channels. The protocol does not require any reliable data-driven side-channel nor a remote trusted party like a Certificate Authority (CA). The protocol is based on the Ephemeral Elliptic Curve Diffie-Hellman key exchange and provides a flexible double authentication mechanism with cryptographic signatures, and Short Authentication Strings (SAS) pronounced aloud by the users. Considerable protocol simplifications lead to robustness against signal fading and message dropouts.

The protocol's security properties were successfully verified in a symbolic model using Tamarin Prover [Meier et al., 2013], a cryptographic protocol verification tool. Formal symbolic verification can be considered as the first important step in evaluating the protocol security. Parallelly, the work emphasizes some practical aspects of the cryptographic key exchange such as user-friendliness, easiness of implementing, and resilience against adversarial attacks.

The protocol and the symbolic verification in Tamarin Prover were published in 'Introducing a Verified Authenticated Key Exchange Protocol over Voice Channels for Secure Voice Communication' at the 6th International Conference on Information Security and Privacy [Krasnowski et al., 2020].

1.5 Organization

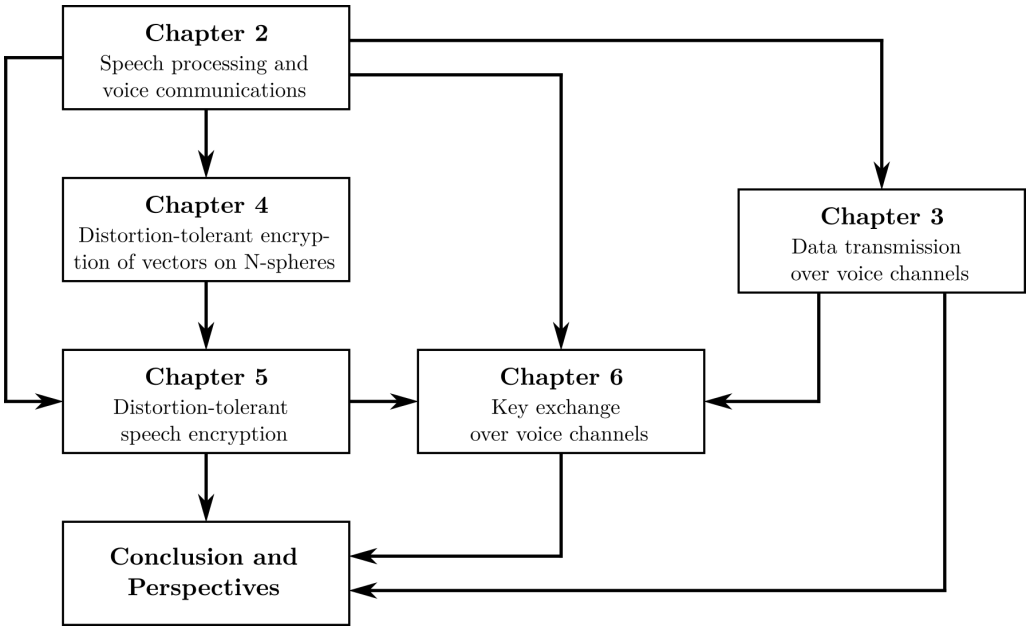


Figure 1.9 – Flow diagram of the manuscript.

Chapter 2 reviews the principles of speech parametrization, speech synthesis, and digital voice communication, which help to understand some challenges related to secure communications over voice channels. The investigation revealed that many real voice channels (cellular networks, VoIP) rely on Linear Predictive Coding (LPC), inspiring the robust DoV technique presented in **Chapter 3**. Furthermore, the analysis of various speech coding techniques concluded that achieving error-less data transmission over general voice channels is very unlikely. This result motivated the work on distortion-tolerant encryption techniques in **Chapter 4** and **Chapter 5**. Finally, the characterization of cellular networks and VoIP highlighted their susceptibility to signal dropouts, which encouraged designing a robust AKE protocol detailed in **Chapter 6**.

Chapter 3 describes the DoV technique based on codebooks of harmonic waveforms. The technique description is preceded by an examination of the typical signal distortion introduced by a selection of LPC coders. This chapter details an efficient codebook design method using quaternary codes and derives a demodulation rule with distortion compensation. Transmission performance over cellular networks and VoIP is presented and discussed. Finally, the chapter outlines an experimental secure voice system using DoV and proposes a technique countering voice activity detection by repetitive silence insertion.

Chapter 4 details the distortion-tolerant encryption scheme for scrambling spherical data using commutative spherical group codes. The chapter starts by recalling the fundamental theory of lattices, spherical codes, and secure pseudo-random generators. The chapter then introduces the notion of distortion-tolerant encryption that describes the system's capability to decipher distorted ciphertexts approximately.

The encryption scheme using spherical codes is thoroughly detailed. The chapter explains the computational indistinguishability of encryptions in the presence of an eavesdropper when the source of randomness is a secure pseudo-random number generator (PRNG). For illustration, the chapter finishes with a toy example of distortion-tolerant color scrambling of an RGB image.

Chapter 5 introduces the distortion-tolerant speech encryption scheme. The chapter details all processing steps, including speech encoding, enciphering, pseudo-speech synthesis, and final speech reconstruction using so-called LPCNet voice synthesizer based on Machine Learning. Further, the chapter investigates the security, operational, and computational aspects of the system. Finally, it presents the results of some simulations and a speech quality assessment.

Chapter 6 presents the authenticated key exchange protocol over fading voice channels. The chapter enlists the security requirements posed by secure voice communication systems and discusses some typical use cases. Moreover, the chapter briefly describes a symbolic protocol verification in Tamarin Prover from the user perspective. The protocol properties and verification results are thoroughly discussed.

Chapter 7 concludes the manuscript and gives prospects for future work.

1.6 Highlights

My PhD started in December 2017 as a DGA [2] CIFRE [3]-Defense fellowship at the cybersecurity startup BlackBoxSécu in Sophia Antipolis, France, and in cooperation with I3S [4]-CNRS [5] and the Université Côte d'Azur, France. The project was co-funded by BlackBoxSécu and the DGA grant No 01D17022178.

During my PhD at BlackBoxSécu, I helped with developing a stand-alone device for speech encryption that could be connected to a mobile phone in tandem, and would enable a secure communication over voice channels. This application-oriented study resulted in a new Data over Voice technique and an optimized Authenticated Key Exchange protocol over voice channels, both described in this manuscript.

In December 2019, I was a visiting student at the group of Prof. Cheon Jung-hee at Seoul National University, South Korea, where I had a great pleasure to observe the cutting-edge research on homomorphic encryption, and to exchange valuable ideas.

In February 2020 my DGA CIFRE-Defense grant was converted to a fully academic PhD at I3S and the Université Côte d'Azur due to some financial difficulties faced by BlackBoxSécu. The studies were funded by AID [6] from the grant No SED0456JE75. Starting from that moment, I moved my attention towards more theoretical aspects of my project, which was a new distortion-tolerant speech encryption scheme using spherical commutative group codes.

2. Direction Générale de l'Armement
3. Conventions Industrielles de Formation par la Recherche
4. https://www.i3s.unice.fr/
5. http://www.cnrs.fr/
6. Agence de l'Innovation de Défense

CHAPTER 2

Speech processing and voice communications

Secure voice communication over voice channels encompasses three technical problems associated with speech processing: ensuring conversation secrecy, maintaining high speech quality at the reception side, and enabling robust and low-latency signal transmission via vocal channels. These seemingly different topics should be considered jointly due to severe constraints posed by voice channels, which operate differently than traditional data-driven communication channels.
This chapter provides some essential background for understanding challenges related to secure voice communication over voice channels. The chapter reviews the principles of speech parametrization, speech synthesis, and digital voice communication. Furthermore, it gives a general overview of popular voice coding techniques, briefly characterizes vocal channels, and points out some relevant state-of-the-art solutions. The chapter concludes with some guiding rules for designing a robust communication system over voice channels.

2.1 Motivation

The secure voice communication setting considered in this study involves sending a synthetic signal with data using some voice-oriented application. This seemingly simple problem, however, turns out to be rather challenging. For example, Figure 2.1 illustrates a received and recorded synthetic signal sent between two mobile phones through WhatsApp. The signal consisted of eight phase-modulated harmonics in the 0.4-3.2 kHz range with short repeated synchronization tones at 3.6 kHz. Harmonics were modulated using traditional Phase-Shift-Keying (PSK) modulation, and carried some binary information. After a few seconds of transmission, WhatsApp suppressed these modulated harmonics and significantly amplified the synchronization tones instead. The binary data encoded into harmonics were irreversibly lost.

This experiment revealed that real-world digital voice channels cannot be considered as classical communication channels. Voice channels aim at preserving speech intelligibility, which is quite a different goal than of data-driven channels. Consequently, voice-oriented applications perform complex processing that may completely disarrange the fine time-structure of the input signal.

Voice channels are the primary limiting factor for a secure communication system. The pseudo-speech signal must conform with the speech model implemented in a particular voice channel, otherwise risking signal distortion or suppression. Moreover, some channel characteristics (e.g., audio bandwidth, signal compression ratio, latency) limit the available data throughput for sending encrypted speech. Consequently, robust communication is possible only if the available channel throughput (i.e., determined by compression bitrate) is sufficiently high.

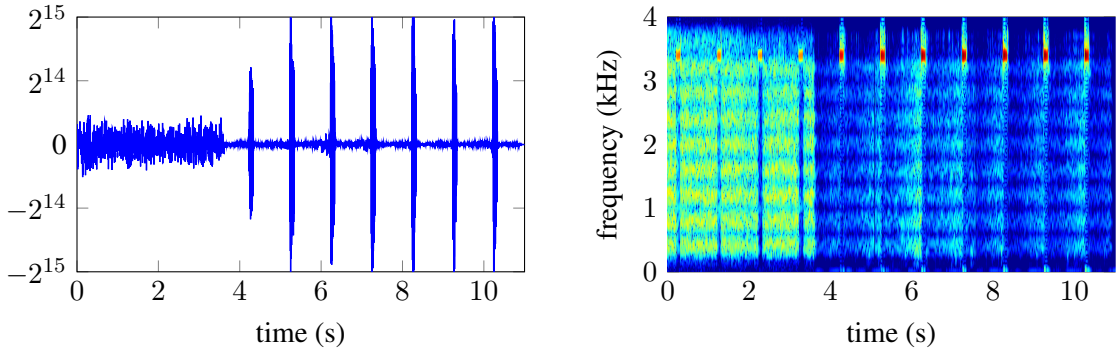


Figure 2.1 – Received and recorded signal sent over the WhatsApp application presented in the time domain (left) and the time-frequency domain (right). The signal consisted of eight phase-modulated harmonics in the 0.4-3.2 kHz range and short repeated synchronization tones at 3.6 kHz. After a few seconds of the transmission, WhatsApp suppressed the phase-modulated part of the signal and amplified synchronization tones.

This chapter reviews the principles of speech coding and communication and gives some essential background needed to answer the three questions: how modern voice channels process and transmit the input signal, how to produce pseudo-speech that conforms with digital voice channels, and what speech data to protect. The chapter outlines speech production and perception mechanisms that underlie speech parameterization, speech compression, and speech synthesis. Furthermore, it focuses on the core elements and parameters of 2G/3G cellular networks and Voice over Internet Protocol (VoIP), which are anticipated to cover most usage scenarios for the secure voice system.

This chapter is organized as follows. Section 2.2 recalls the *speech chain* concept, helpful in understanding speech processing paradigms. Section 2.3 describes the source-filter speech production model and the speech encoding method based on Linear Predictive Coding. Section 2.4 focuses on the auditory system and perceptual speech parametrization. Section 2.5 overviews four prominent voice coding families: waveform coders, LPC coders, perceptual coders, and parametric coders, which are widely adopted in digital voice communications and can be used in speech encryption with lossy encoding. Moreover, the section enlists possible speech synthesis methods, including autoregressive-moving-average (source-filter) synthesis, sinusoidal speech synthesis, and synthesis using trained neural networks. Finally, Section 2.6 briefly characterizes voice channels in cellular networks and VoIP with an emphasis on operational constraints (delay, bandwidth, reliability) and algorithms such as Voice Activity Detection (VAD), Adaptive Gain Control (AGC), and Noise Suppression (NS). Section 2.7 summarizes the chapter.

2.2 The speech chain

The speech signal is an acoustic, analog waveform that conveys some encoded message and is used in natural human communication [Benesty et al., 2008, Chap. 1, Rabiner and Schafer, 2011, Chap. 1]. Apart from the linguistic content, speech provides much paralinguistic information such as the speaker's identity [Furui, 1996, Campbell, 1997], age and gender [Ptacek and Sander, 1966, Childers and Wu, 1991, Metze et al., 2007, Li et al., 2013], emotions [Nwe et al., 2003, Vogt and André, 2006], and the linguistic origin [Hanani et al., 2013, Kolly and Dellwo, 2014]. From a security standpoint, all this information is considered sensitive and must be protected by the speech encryption system. If protection of some paralinguistic information becomes too burdensome for the system (for example, due to insufficient bandwidth), this information should be suppressed rather than leaked.

The linguistic and paralinguistic information conveyed in the speech signal may be seen as a multi-step speech production process, shown in Figure 2.2. Speech production involves an abstract message formulation, language-level representation, and physiological articulation. These encoding steps add natural redundancy that robustifies communication in a noisy environment. The speech signal captured at the ear is decoded in the reversed order. Firstly, the auditory system performs a spectral analysis of the received speech and extracts the spectral features. In the next step, the auditory nerves forward the spectral information to the brain, which recognizes particular phonemes, and forms words and sentences. Finally, the brain translates the linguistic representation into the initial message [Rabiner and Schafer, 2011, Chap. 1]. These combined speech production and perception processes are called the *speech chain* [Denes and Pinson, 1993].

The speech chain intuitively describes the information flow in vocal communication. The initial message formulation, described as a product of the syllable-level information density (in bits/syllable) and the average utterance speed (in syllables/second), is done approximately at the rate of 39 bps [Coupé et al., 2019]. This estimated value is believed to reflect the neurocognitive capabilities of the human brain and be independent of the speaker's language. Another approach is to represent the message as a sequence of phonemes. For example, the English phonological system consists of 42 phonemes [Cohen, 1971] where each phoneme could be encoded with 6 bits. Assuming an average utterance speed of 10 phonemes/seconds, we obtain 60 bps of the information rate [Ramasubramanian and Doddala, 2015]. Appending the minimum prosodic information (e.g., duration, intensity, intonation) also adds some redundancy [Rabiner and Schafer, 2011, Chap. 1].

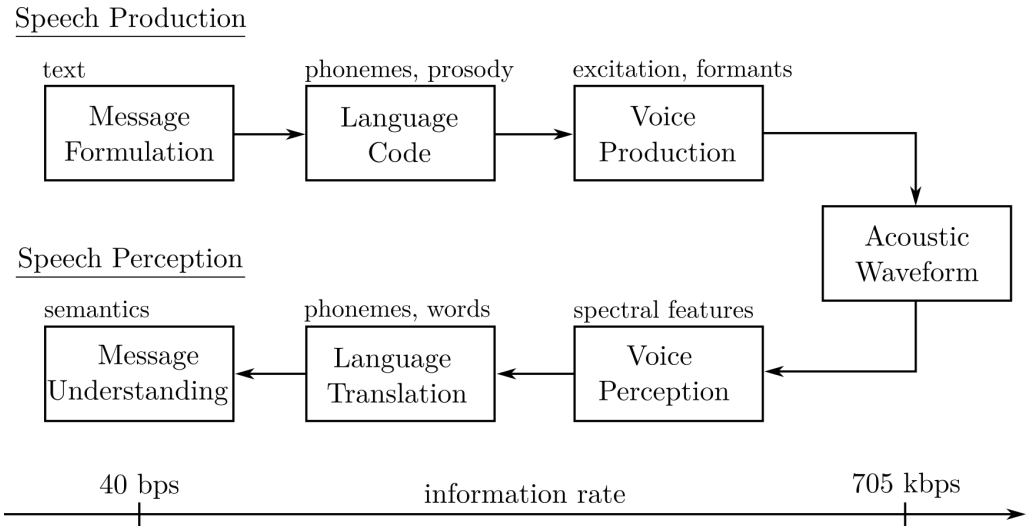


Figure 2.2 – Diagram of speech production and perception.

Speech articulation transforms the discrete message into the analog, continuous domain. The slowly moving articulatory system needs about 100 ms to significantly change the produced sound's characteristic [Gay et al., 1974, Tasko and Westbury, 2004, Flanagan, 1972]. The result of speech articulation is a speech signal that could be represented as a sequence of speech samples. Since the audible spectrum ranges between 20 Hz to 20 kHz (the upper bound drops somewhat in adulthood) [Purves et al., 2018, Chap. 13], one may assume the 'CD-quality' sampling rate of 44.1 kHz and 16-bit quantization. The data rate of such sampled signal goes up to 705 kbps.

Interestingly, there is growing evidence for some physical correlation between the production and perceptual systems [Miller et al., 1986, Wilson et al., 2004, Ackermann et al., 2007]. Thus, it is believed that the information rates associated with respective speech production and perception steps are comparable to each other [Rabiner and Schafer, 2011, Chap. 1].

The last important element of the speech chain is the transmission channel. In natural conversation, the channel is a noisy acoustic connection between the speaker and the listener. In long-distance communication, an acoustic speech signal is transduced to electric intensity and sent in the analog or the digital domain over the network.

Modern digital voice channels encode and compress speech signals to reduce the data to send. Again, the speech chain provides a good outline of strategies for extracting relevant information from the speech signal. For instance, increasingly popular speech-to-text applications mimic the whole speech perception process, firstly converting the speech signal into a sequence of features, and then mapping these features into phonemes, syllables, and finally into full words [Huang et al., 2001, Chap. 9, Gold et al., 2011, Chap. 22-24, Rabiner and Schafer, 2011, Chap. 14]. However, building highly abstract representations of the speech signal requires some computational effort and introduces a significant processing delay. Thus, coding algorithms intended for real-time voice communications are mostly inspired by physiological voice production and perception mechanisms. The same processing constraint applies to securing speech and to producing an encrypted pseudo-speech signal in real-time.

2.3 Source-filter model and Linear Predictive Coding

One of the most efficient techniques used in real-time voice compression (and hence popular in vocal communications) takes inspiration from the mechanism of speech production by phonatory and articulatory systems, illustrated in Figure 2.3. The phonatory system, consisting of lungs, a larynx, and vocal cords, is responsible for producing excitation sound. The role of articulatory organs - a lower jaw, a velum, a tongue, and lips - is to change the shape of a *vocal tract*, which begins at the opening of the glottis, and then continues through a pharynx, oral and nasal cavities, and finishes at lips and nostrils [Benesty et al., 2008, Chap. 2, Rabiner and Schafer, 2011, Chap. 3, Bäckström, 2017].

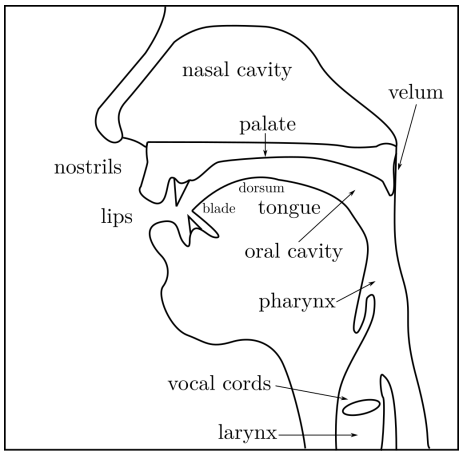


Figure 2.3 – Phonatory and articulatory systems. Krasnowski.

According to the source-filter model of speech [Müller, 1840, Fant, 1960, Kelly and Lochbaum, 1962], voice production starts with the air pushed out by the lungs and flowing through the vocal cords located in the larynx. When the vocal cords are tense, the air pressure causes the organ to open and close in a quasi-periodic manner. As a result, the vibrating vocal cords generate pulse-shaped buzzing sound. The cords' tension and size, combined with the airflow speed, control the sound frequency. Then, the buzzing excitation propagates through the vocal tract, where it is spectrally and temporarily shaped. A continuous movement of the tongue, the teeth, and the lips change the tract's transfer function, resulting in different sounds emitted by the speaker. The size and the shape of the tract may significantly differ between each person, resulting in the unique prosodic voice characteristics that enable speaker identification. Table 2.1 lists ranges of the sound frequency produced in glottis for male, female and child speakers [Rabiner and Schafer, 2011, Chap. 3], and Table 2.2 describes the average size of the vocal tract [Fitch and Giedd, 1999].

Table 2.1 – The range of frequencies (pitch periods) produced in glottis by male, female and child speakers [Rabiner and Schafer, 2011, Chap. 3].

	Minimum		Average		Maximum	
Male	80 Hz	(12.5 ms)	125 Hz	(8.0 ms)	200 Hz	(5.0 ms)
Female	149 Hz	(6.7 ms)	227 Hz	(4.4 ms)	345 Hz	(2.9 ms)
Child	200 Hz	(5.0 ms)	303 Hz	(3.3 ms)	500 Hz	(2.0 ms)

Table 2.2 – The average 'curved' lengths (in millimeters) of lip, tongue blade, tongue dorsum, velum and pharynx in male, female and child speakers [Fitch and Giedd, 1999].

	Lip	Blade	Dorsum	Velum	Pharynx	Total
Male	14.5	25.5	26.8	34.0	60.4	161.2
Female	12.8	24.8	24.6	37.2	46.9	146.4
Child (5-10)	13.5	20.0	22.6	27.5	35.7	119.5

The sound characteristics produced by the vocal cords and the moving vocal tract can be well observed on a spectrogram (Figure 2.4). The voiced parts of the signal have a clear harmonic structure, with the harmonics located at the multiples of the fundamental frequency of excitation. On the contrary, the unvoiced signal resembles noise of intensity concentrating at higher frequencies. The darker lines in the spectrogram correspond to formants, the tract's resonant frequencies. When the articulatory organs are moving, formants are continuously changing their position and amplitude. In practice, variation and relative position of formants are crucial in phonemic classification [Lindblom and Studdert-Kennedy, 1967, Syrdal and Gopal, 1986].

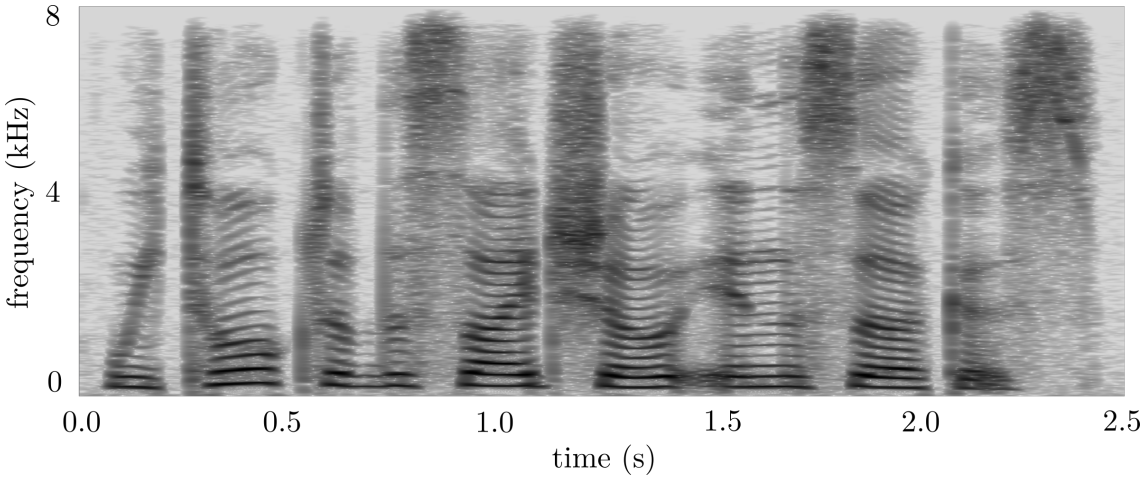


Figure 2.4 – Spectrogram of the recorded speech signal.

In speech coding, the outlined speech production system is often approximated by a simplified, discrete-time model illustrated in Figure 2.5. According to the model, speech production consists of two distinct elements: a discrete-time excitation (source signal) generator and a filtering transfer function. The latter element represents the combined effect of the glottal pulse shaping, the vocal tract, and radiation at the lips [Rabiner and Schafer, 2011, Chap. 3].

An excitation signal can take two different forms, depending on whether a generated sound is classified as voiced or unvoiced. When the sound is voiced, the excitation generator simulates vibrating vocal cords and produces an impulse train (Dirac comb) with a specified period. In another case, the generator outputs white noise. The resulted excitation signal is amplified by some gain G.

Derivation of the correct tract transfer function is challenging because it requires the exact knowledge of the transient tract shape and the soft tissue covering. Instead, the vocal tract is approximated by N concatenated lossless tubes, as illustrated in Figure 2.6. Then, the transfer

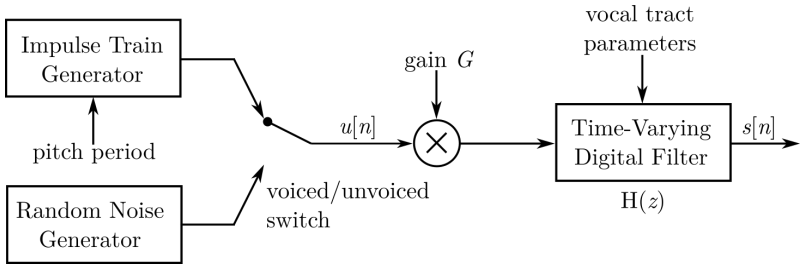


Figure 2.5 – Simplified discrete-time, source-filter model for speech synthesis.

function $\mathrm{V}(z)$ representing the vocal tract (without the glottal shaping and the lips radiation) takes the form of the N-pole filter with coefficients $a_1,\ ...,\ a_N$:

$$\mathrm{V}(z) = \frac{1}{1 - \sum_{k=1}^{N} a_k z^{-k}}. \tag{2.1}$$

The function $\mathrm{V}(z)$ is experimentally extended to an all-pole transfer function $\mathrm{H}(z)$ with 12-16 coefficients to include effects introduced by the glottal shaping, the nasal cavity, and the lips radiation:

$$\mathrm{H}(z) = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}}. \tag{2.2}$$

In real-world applications, the transfer function $\mathrm{H}(z)$ is usually not known a priori. It is possible to estimate $\mathrm{H}(z)$ and the excitation signal from the sampled speech signal $s[n]$ in the process called *linear prediction analysis*. The big success of linear prediction popularized the term 'linear predictive coding' (LPC) when referring to all speech processing techniques based on the linear source-filter model shown in Figure 2.5 [Rabiner and Schafer, 2011, Chap. 9].

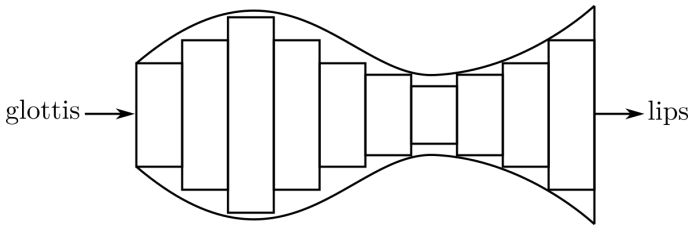


Figure 2.6 – Lossless acoustic tube model of a vocal tract. Krasnowski.

Assuming that the recorded speech follows the source-filter model shown in Figure 2.5 with the excitation $u[n]$ and the all-pole filter $\mathrm{H}(z)$ described by coefficients $a_1,\ ...,\ a_p$, discrete-time speech samples $s[n]$ can be expressed using the difference equation:

$$s[n] = \sum_{k=1}^{p} a_k s[n-k] + Gu[n]. \tag{2.3}$$

Thus, $\mathrm{H}(z)$ acts like a linear prediction filter. Since the true values of the coefficients a_k and the excitation samples $Gu[n]$ are not known beforehand, the prediction algorithm searches coefficients α_k such that the prediction error $e[n]$ (the residual) has the minimum energy $\sum_n e^2[n]$:

$$s[n] = \sum_{k=1}^{p} \alpha_k s[n-k] + e[n]. \tag{2.4}$$

The resulting coefficients α_k and the residual $e[n]$ are assumed to be the true coefficients a_k and the excitation $Gu[n]$, respectively.

In practical realizations, the algorithm processes only short portions of the speech signal, assumed to be stationary [Elliott and Theunissen, 2009]. For the speech interval of length L and starting at $\tilde{n}$, samples $s_{\tilde{n}}[\ell]$ used in the prediction analysis have the following form:

$$s_{\tilde{n}}[\ell] = s[\ell + \tilde{n}]w[\ell], \quad \ell = 0, ..., L-1, \tag{2.5}$$

where $w[\ell]$ is the Hamming window centered at $\lfloor L/2 \rfloor$ and equal to 0 outside $0, ..., L-1$. In such a case, the prediction algorithm aims at minimizing the residual energy over the interval:

$$\mathcal{E}_{\tilde{n}} = \sum_{\ell=0}^{L-1+p} \left(s_{\tilde{n}}[\ell] - \sum_{k=1}^{p} \alpha_k s_{\tilde{n}}[\ell-k] \right)^2. \tag{2.6}$$

The minimization of $\mathcal{E}_{\tilde{n}}$ has a unique solution. Taking the partial derivatives $\partial \mathcal{E}_{\tilde{n}} / \partial \alpha_i$ for $i = 1, ..., p$, and equating them to 0 gives p linear equations with p unknowns:

$$\sum_{\ell=0}^{L-1+p} s_{\tilde{n}}[\ell-i] s_{\tilde{n}}[\ell] = \sum_{k=1}^{p} \alpha_k \sum_{\ell=0}^{L-1+p} s_{\tilde{n}}[\ell-i] s_{\tilde{n}}[\ell-k], \quad 1 \leq i \leq p. \tag{2.7}$$

Equations 2.7 can be efficiently solved by the Levinson-Durbin algorithm [Makhoul, 1975].

With the $\mathrm{H}(z)$ finally obtained, the signal residual $e[n]$ is further processed to compute the gain G and estimate the harmonicity ratio (voicing). When the signal is sufficiently harmonic, the speech encoder estimates the pitch period. Otherwise, the excitation is replaced with white noise.

The two-level excitation model is sufficient for typical vocal sounds. For instance, Figures 2.7 and 2.8 present residual signals obtained from LPC analysis of the real vowel $/e/$ and the fricative $/s/$ recordings. It can be noticed that the residual of the voiced vowel is similar to a pulse train and the residual of the unvoiced fricative resembles noise.

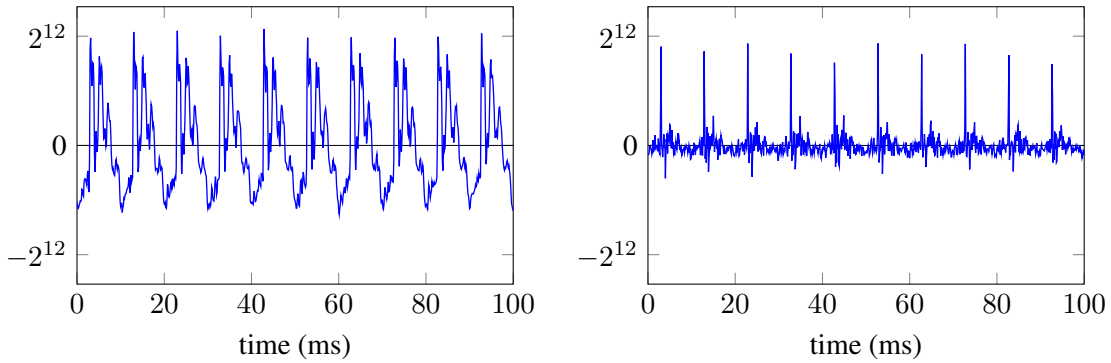


Figure 2.7 – LPC analysis of vowel $/e/$: (left) time-domain waveform, (right) filtering residual obtained with the 12th order LPC filter.

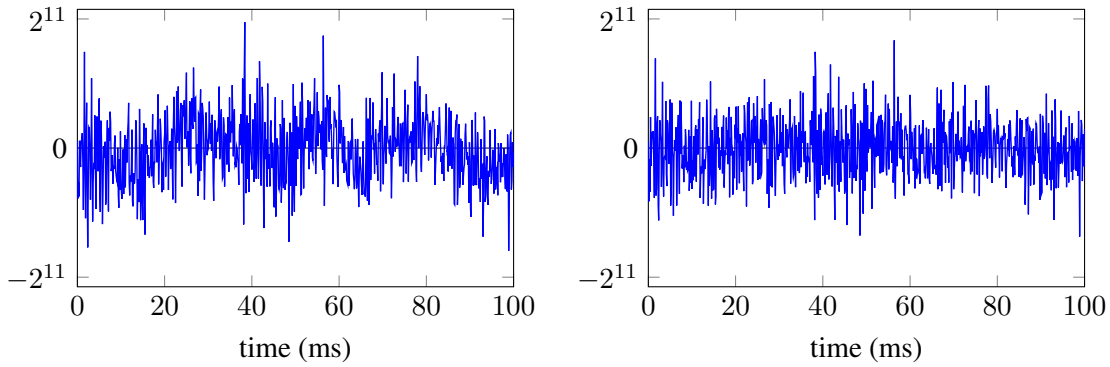


Figure 2.8 – LPC analysis of fricative $/s/$: (left) time-domain waveform, (right) filtering residual obtained with the 12th order LPC filter.

Figure 2.9 displays the LPC filter frequency response obtained from a linear predictive analysis of vowel /e/. The filter phase response is highly nonlinear, whereas magnitude has distinguishable resonances corresponding to the vowel's formants.

Despite its simplicity, linear predictive coding achieves satisfactory results in representing speech signals at medium bitrates (4-16 kbps). In consequence, LPC became a popular technique in voice communication systems, including cellular networks and VoIP. For this reason, it seems natural that the encrypted pseudo-speech used in secure communication must fit into the source-filter speech model and be robust against LPC processing. For example, linear filtering is likely to change the complex spectral profile of the transmitted signal. The crucial question is how a particular voice channel corrupts unvoiced residuals. When the coder replaces unvoiced sounds with synthetic white noise, information encoded into an encrypted signal can be irreversibly lost.

Fortunately, a majority of voice coders adopted in digital communications do not replace noisy residuals entirely. Instead, coders often apply sophisticated encoding techniques in order to represent the residual faithfully. In practice, the encoding method of the residual determines the distortion characteristics introduced by the voice channel.

A more detailed characterization of LPC-based voice channels and some strategies for producing conforming pseudo-speech signal is covered in Chapter 3.

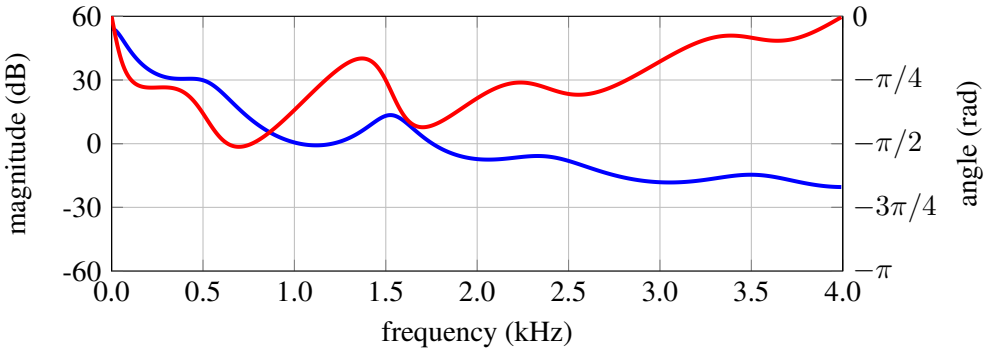


Figure 2.9 – Magnitude (blue line, left y-axis) and phase (red line, right y-axis) response of the LPC filter of order 12, obtained by the LPC analysis of vowel $/e/$.

2.4 Psychoacoustics and perceptual speech coding

An alternative method to encode speech relies on the psychoacoustic properties of the human auditory system. Unlike source-filter modeling, which simulates the voice production, perceptual speech coding aims at preserving signal information relevant for the listener. This approach is the one underlying general audio coding as done in MPEG-2 Audio Layer III (MP3) [ISO/IEC, 1998], and Advanced Audio Coding (AAC) [ISO/IEC, 2006].

Speech perception is determined primarily by the physiological processing of sound captured by the pinna. The processing starts at the eardrum, which transforms acoustic pressure waves into mechanical vibrations. Next, the vibrations are transduced by the adjoint auditory ossicles (malleus, incus, and stapes) and transmitted through the oval window to the cochlea. Inside this snail-shaped organ filled with fluid, the mechanical vibrations produce multiple resonances at different longitudinal locations corresponding to specific frequency components (Figure 2.10). Finally, the frequency-sensitive hair-cell sensors distributed along the basilar membrane detect the resonances and trigger the generation of electric pulses transmitted to the brain by the auditory nerve.

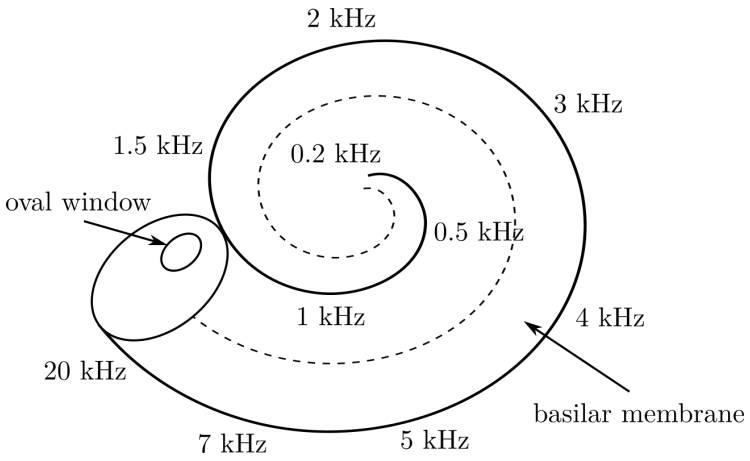


Figure 2.10 – Simplified model of the cochlea and the cochlea longitudinal frequency distribution.

The basilar membrane may be seen like a bank of non-linear filters transforming the time-domain signal into its short-time spectral representation [Rabiner and Schafer, 2011, Chap. 4, Gold et al., 2011, Chap. 14, Fastl and Zwicker, 2006]. The frequency response of the membrane is often compared to a set of intensity-sensitive, overlapping bands. The number and widths of the filters have been estimated experimentally [Kiang and Moxon, 1974], leading to the idealized concept of 24 critical bands [Fastl and Zwicker, 2006] that covers the audible band in average adults. According to the model, the bands are approximately constant below 500 Hz and are logarithmically widening at higher frequencies. The non-linear band distribution is usually represented in the Bark scale, which numbers center frequencies of bands by integers from 1 to 24 (Table 2.3). The Bark scale approximates the octave-based perception of notes [Kallman, 1982].

The auditory filter-bank model is extensively used in perceptual speech coding for representing the short-time speech spectrum. To improve encoding efficiency, speech perception models consider non-linear sound perception and temporal/frequency masking [Lyon, 1982, Kubin and Kleijn, 1999].

Table 2.3 – Bark scale critical bands [Fastl and Zwicker, 2006].

Bark	Center Freq. (Hz)	Bandwidth (Hz)	Bark	Center Freq. (Hz)	Bandwidth (Hz)
1	50	100	13	1850	280
2	150	100	14	2150	320
3	250	100	15	2500	380
4	350	100	16	2900	450
5	450	110	17	3400	550
6	570	120	18	4000	700
7	700	140	19	4800	900
8	840	150	20	5800	1100
9	1000	160	21	7000	1300
10	1170	190	22	8500	1800
11	1370	210	23	10500	2500
12	1600	240	24	13500	3500

Perceptual speech coding proved to be very useful in speech recognition [Huang et al., 2001, Chap. 9, Rabiner and Schafer, 2011, Chap. 14], voice transformation [Benesty et al., 2008, Chap. 24] and speech synthesis [Kondoz, 2004, Milner and Shao, 2006, Juvela et al., 2018]. However, speech coders relying solely on auditory models provide lower voice quality than source-filter coders at the same compression rate. It is because source-filter coding makes stronger assumptions about the encoded speech signal, enabling sparser representation [Bäckström, 2017, Benesty et al., 2008, Chap. 18].

The physiological properties of the auditory system enable the listener to distinguish three components of a vocal (or a musical) sound: loudness, pitch, and timbre [Huang et al., 2001, Chap. 2]. Although sound perception varies among listeners, a considerable effort has been undertaken to relate sound perception with objective signal properties [Stevens, 1956, Von Békésy and Wever, 1960, Goldstein, 1973]. The studies revealed that these perceptual qualities can be roughly related to three signal characteristics: signal intensity, fundamental frequency, and spectral envelope, as listed in Table 2.4. Moreover, it can be assumed that these components are roughly independent of each other, i.e., a modification of one physical parameter changes only the corresponding perceptual component.

Table 2.4 – Simplified relation between the perceptual and the physical qualities of speech signal.

Perceptual Quality	Physical Quality
loudness	signal intensity
pitch	fundamental frequency
timbre	spectral envelope

Speech loudness is perceptually related to sound intensity, which is defined as the ratio between the average flow of the energy through a unit area I and the accepted referential threshold of hearing $I_0 = 10^{-12}\ \mathrm{W/m^2}$. Sound intensity is usually represented in the logarithmic scale:

$$\mathrm{IL} = 10 \log_{10} \left(\frac{I}{I_0} \right) \ \mathrm{dB}. \tag{2.8}$$

Sound intensity of a natural speech varies usually between 20 dB - 80 dB [Rabiner and Schafer, 2011, Chap. 4].

The sensitivity of the auditory system to sound intensity varies in frequency. Figure 2.11 presents the equal-loudness curves [Fletcher and Munson, 1933], indicating the sound intensity levels perceived by an average listener as equally loud. The highest sensitivity to sound intensity is located between 500 Hz and 5 kHz, and largely overlaps with the speech spectrum between 150 Hz and 7 kHz [Rabiner and Schafer, 2011, Chap. 4]. Subband speech and audio coders often exploit the nonlinear sound perception by optimizing the quantization of frequency bands [Benesty et al., 2008, Chap. 18].

Pitch, expressed in mel units, is a perceptual quantity describing the frequency of a tone. The nonlinear relation between the pitch and the true frequency is fitted by the following formula:

$$\text{pitch} = 1127 \log_{10}\left(1 + \frac{f}{700}\right) \text{ mel}, \tag{2.9}$$

where f is the tone frequency in Hz [Stevens et al., 1937, Rabiner and Schafer, 2011, Chap. 4].

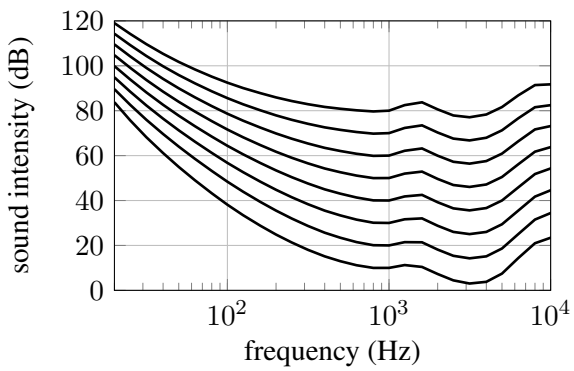


Figure 2.11 – Equal-loudness curves as a function of frequency and for different values of the sound intensity [ISO, 2003]

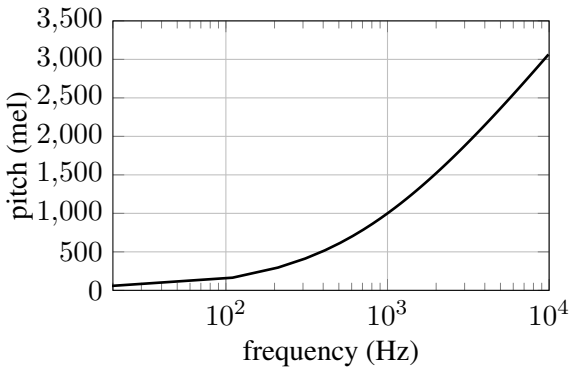


Figure 2.12 – Subjective pitch in mels as a function of the actual frequency of a pure tone.

As suggested by Figure 2.12, the listener's ability to discriminate two similar tones decreases when the frequency of the lower tone goes up. This observation agrees with the critical band theory because two close tones are more likely to fall into the same spectral band at higher frequencies than at lower frequencies [Fastl and Zwicker, 2006].

The third perceptual component, timbre (the color of sound), is sometimes described as a quality distinctive from pitch and loudness [ANSI, 1994]. The research on describing timbre perception is still ongoing (for a relevant discussion, see [Sethares, 2004, Aucouturier and Bigand, 2012, Siedenburg et al., 2016]). Nevertheless, it is generally agreed that timbre is a multidimensional property describing spectro-temporal characteristic of a sound. In the case of voice, timbre is usually associated with formants or spectral envelope [Huang et al., 2001, Chap. 2]. Such an interpretation links timbre with the vocal tract's shape, contrarily to pitch and loudness related to excitation.

Compared to pitch and loudness, an efficient parametrization of timbre appears troublesome. There are many propositions for approximate spectral envelope representations, such as Perceptual Linear Prediction (PLP) model or LPC filter frequency response [Rabiner and Schafer,

2011, Chap. 8]. One efficient representation of timbre in the speech recognition domain are Mel-Frequency Cepstral Coefficients (MFCC) [Davis and Mermelstein, 1980]. MFCC are defined as the discrete-cosine transform (DCT) of the logarithm of the mel-spectrum of a speech frame:

$$\mathrm{MFCC}[n] = \frac{1}{R}\sum_{r=1}^{R} \log_{10}(\mathrm{MF}[r]) \cos\left[\frac{2\pi}{R}\left(r+\frac{1}{2}\right)n\right],$$

where MF is the mel-spectrum of the speech frame computed based upon a filter bank of $R \geq n$ weighting functions (Figure 2.13) that approximate the critical band spacing:

$$\mathrm{MF}[r] = \frac{\sum_{k=L_r}^{U_r} |\mathrm{W}_r[k]\mathrm{X}[k]|^2}{\sum_{k=L_r}^{U_r} |\mathrm{W}_r[k]|^2},\ r = 1, 2, ..., R,$$

where X is the Discrete Fourier Transform (DFT) of the speech frame, and $\mathrm{W}_r[k]$ is the weighting function of the r-th filter ranging over DFT indices L_r to U_r. The number R and the distribution of filters may slightly differ among various implementations.

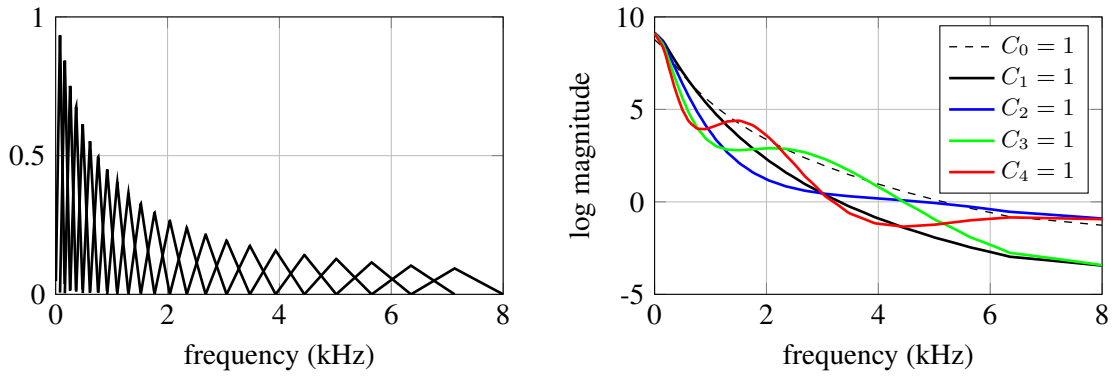


Figure 2.13 – (Left) Weighting functions for mel-scale filtering and (right) smooth interpolation of the DFT spectra X obtained when looking for all MFCC forced to zero except a single one in $\{C_0, ..., C_4\}$ equal to 1. The computation of these MFCC is done using the bank of weighting functions in the left figure.

This somehow unobvious MFCC definition reflects the nonlinearity of timbre perception. The filter distribution roughly corresponds to critical bands, whereas the logarithm of windowed energies approximates loudness perception. Finally, DCT decomposes the filtered spectrum into orthogonal elements. Since the speech energy tends to concentrate in the first 13-19 coefficients, computed MFCC are often truncated [Benesty et al., 2008, Chap. 9, Rabiner and Schafer, 2011, Chap. 8].

The popularity of MFCC in the speech recognition domain comes from their ability to extract relevant features from spectral envelopes. There had been even an interesting attempt to introduce MFCC as an objective timbre measure [Terasawa et al., 2012, Terasawa et al., 2005]. Nevertheless, provided that many MFCC-based speech synthesizers perform rather poorly [Chazan et al., 2000, Milner and Shao, 2006, Yoon et al., 2007, Boucheron et al., 2011, Juvela et al., 2018], it seems that the transformation removes some salient information from a signal.

Speech parametrization using the signal energy, the fundamental frequency, and the spectral envelope works especially well for voiced sounds. As an example, Figure 2.14 presents the spectral magnitude of the same vowel /e/ as in Figure 2.7. It can be noticed that the mentioned parameters

accurately describe the magnitude of the harmonic signal. However, the representation almost entirely ignores the phases of harmonics. Discarding the phase information is motivated by partial agnosticism of the auditory system to signal phase, especially above 1 kHz [Benesty et al., 2008, Chap. 4]. This effect could be caused by a roll-off of firing synchrony in auditory nerves above 2 kHz [Alves-Pinto et al., 2014].

Several speech encoding techniques used in vocal communication originate from major studies on sound perception, i.e., signal pre-emphasis, frequency masking of weak tones, nonlinear quantization, and noise shaping [Bäckström, 2017, Chap. 4]. Consequently, the pseudo-speech signal used in secure communication should be robust against perceptually-oriented processing in the spectral domain, not only against source-filter coding.

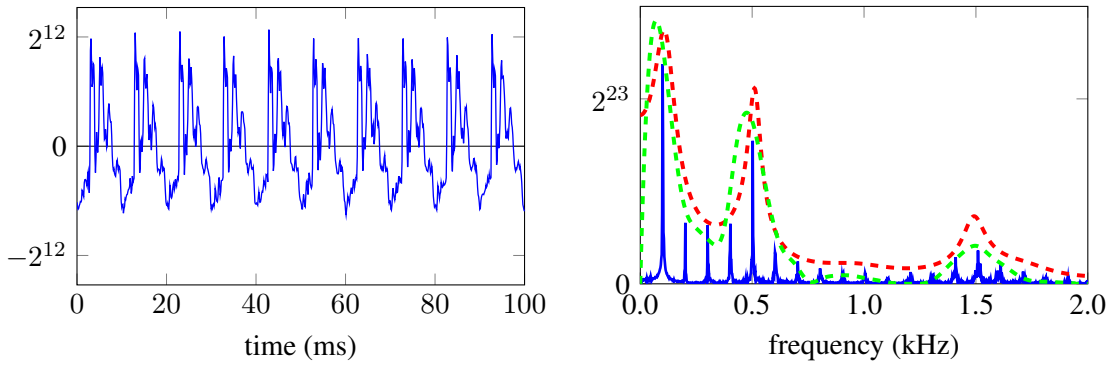


Figure 2.14 – LPC analysis of vowel /e/: (left) time-domain waveform and (right) spectrum of the waveform (blue solid line), frequency response of the 18th order LPC filter (red dashed line), and the mel cepstrum smoothing obtained using the MFCC analysis and truncating MFCC above $n = 13$ (green dashed line).

2.5 Voice coders

Digital voice communication systems send encoded speech as binary data and re-synthesize signal at the reception side. The paramount role of voice codecs, or *vocoders*, is to reduce the transmission bitrate without a significant perceptual quality degradation of decoded voice. Concurrently, voice coding algorithms have to meet several operational goals, such as a small computational delay, low computational and memory requirements, and finally, robustness to channel imperfections (e.g., lost frames, bit errors) [Benesty et al., 2008, Chap. 14].

Figure 2.15 illustrates a simplified diagram of a generic speech codec. The encoder's input is a low-passed speech signal sampled typically at 8 kHz (narrowband speech) or 16 kHz (wideband speech). Low-pass filtering is done in accordance with the Nyquist-Shannon sampling theorem which states that the highest signal's frequency component should be at most half the sampling rate. The reduction of the signal bandwidth from 20 kHz to 4 kHz or 8 kHz significantly lowers the data rate for sending. On the other hand, reduced bandwidth degrades speech quality.

The speech samples are modeled as representing stationary signal segments (frames), usually of some fixed or variable duration between 5 ms - 20 ms. Voice encoders map these input speech frames into a set of parameters linked to the selected speech model and a distortion measure.

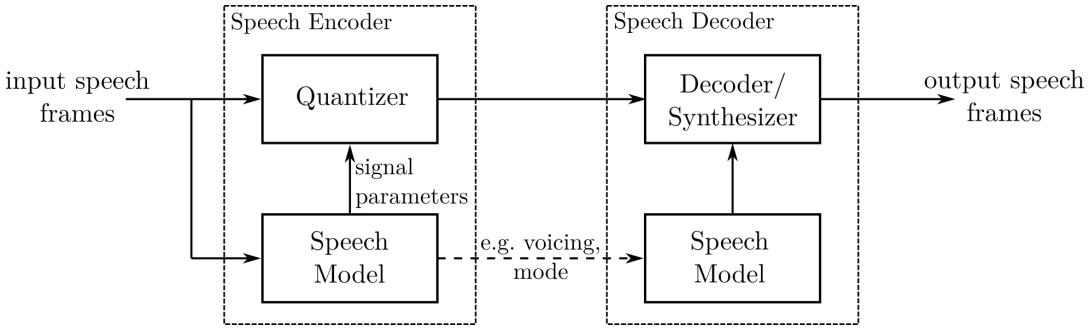


Figure 2.15 – Block diagram of a generic speech encoder and decoder.

Besides, considerable compression gains (from 64 kbps to around 2 kbps and less [Ramasubramanian and Doddala, 2015]) are achieved through the exploitation of quantization and temporal redundancy in speech. The result of the speech compression is a block-based binary stream of a fixed or variable rate. Upon reception of a compressed data, the encoded and quantized parameters are fed into a speech synthesizer, which produces a smoothly varying synthetic speech signal.

The speech model implemented in a voice coder determines the characteristics of the distortion caused by compression. As an example, Figure 2.16 illustrates the same speech waveform compressed by three different vocoders. Despite the perceptual similarity, the obtained waveforms significantly differ. This result seems to question the possibility of designing a genuinely universal and efficient method for sending data over arbitrary voice channels.

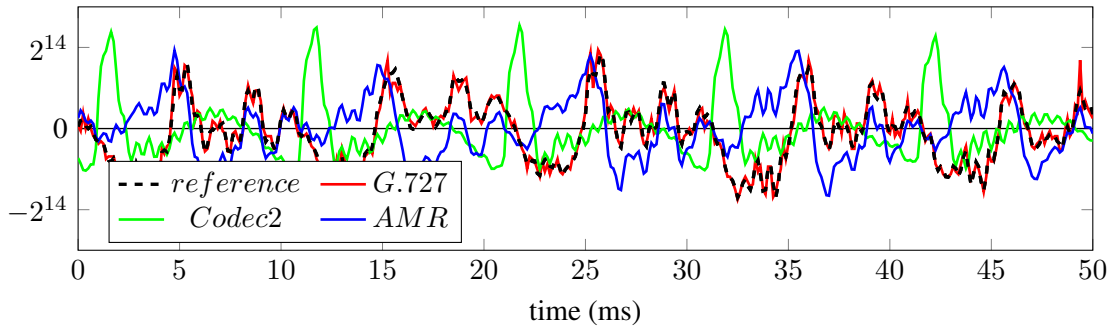


Figure 2.16 – Recorded speech signal compressed by the selection of different voice coders. G.726 is an ADPCM coder, which directly compresses the waveform. Codec2 preserves the spectral envelope and energy of the signal but discards phase information. AMR, an LPC coder, tries to fit the signal waveform into the speech production model. Despite significant differences in their representation, the presented signals are perceptually similar.

Voice coders can be roughly grouped into four major families: waveform coders, source-filter coders, perceptual coders, and parametric coders [Benesty et al., 2008, Chap. 14]. As will be briefly described later in the section, some speech models attract attention as candidates for speech encryption and pseudo-speech synthesis. These models are implemented in low-bitrate parametric coders with a simple structure that is easy to encode, manipulate, and randomize. On the other hand, some coders seem to be suitable for data transmission due to the small or compensable distortion they introduce.

2.5.1 Waveform coders

Waveform coders directly encode the time-domain speech signal without strong assumptions regarding signal characteristics. The most straightforward waveform coding is adopted in the standard G.711, which describes the Pulse Code Modulation (PCM) with fixed μ-law or A-law pseudo-logarithmic quantization [ITU-T, 1988f]. The μ-law quantization reflects the nonlinear intensity perception and improves the perceptual speech quality compared to linear quantization. The encoded signal is sampled at 8 kHz with 8-bit resolution, giving a constant bitrate of 64 kbps.

A modified version of PCM, referred Adaptive Differential PCM (ADPCM), is standardized by G.726 [ITU-T, 1990]. Unlike the generic PCM coding, ADPCM encodes adaptively differences between speech samples and reduces the bitrate of the 64 kbps PCM down to 40, 32, 24, or 16 kbps. Adaptive encoding does not degrade much the perceptual speech quality because consecutive speech samples tend to be highly correlated. A similar approach to G.726 is adopted in wideband G.722 [ITU-T, 2012a]. Additionally, the G.722 coder splits the signal bandwidth into lower (0-4 kHz) and higher (4-8 kHz) parts, encoded separately using ADPCM. The subband separation enables more efficient bit allocation in the lower frequency band, which is perceptually more important than the higher band.

Waveform coders are suitable for encoding signals which are not necessarily speech-like and are closer to traditional modulated signals. This characteristic may underlie efficient data transmission. For example, a series of International Telecommunication Union (ITU) standards describes modem data transmission over digital telephony network [ITU-T, 1988c, ITU-T, 1988a, ITU-T, 1988b, ITU-T, 1988d, ITU-T, 1988e, ITU-T, 1998]. Some of the standardized modems are listed in Table 2.5. The dominant technique used in these modems is amplitude and phase modulation (QAM) of a single carrier. The modems take advantage of frequency division and trellis-coded modulation at higher bitrates to efficiently use the available voice bandwidth.

While voice channels equipped with waveform coders seem to be the most universal for transmission of encrypted pseudo-speech, their high bitrates exceeding 16 kbps are not appropriate for use in speech encryption.

Table 2.5 – Selected modems used in data transmission over telephony network.

standard	V.21	V.22	V.22bis	V.27	V.29	V.34
bitrate (kbps)	0.3	1.2	2.4	4.8	9.6	33.6
modulation	BFSK	QPSK	16QAM	D8PSK	16QAM	QAM+TCM

2.5.2 LPC coders

Linear source-filter coders form a large family of algorithms that rely on the simplified Kelly-Lochbaum (KL) speech production model [Kelly and Lochbaum, 1962]. The KL model approximates vocal tract by lossless acoustic tubes represented as a digital ladder filter. Vocal tract approximation by LPC coders is done similarly in every coder, using linear prediction analysis. The fundamental difference characterizing source-filter coders is the way they encode the excitation.

The most well-known technique used in LPC coders is Code Excited Linear Prediction (CELP) [Schroeder and Atal, 1985]. Instead of a simple switch between the voiced pulse train and the white noise generator, as earlier described in Figure 2.5, CELP coders utilize a codebook of predefined excitation signals. Given the residual of the prediction analysis, the coder searches the closest

codebook vector (or the sum of vectors) by minimizing a Mean Squared (Weighted) Error (MSE) [Benesty et al., 2008, Chap. 17]. Popular examples of CELP coders are Speex [Herlein et al., 2009], Opus-Silk [Valin et al., 2012] and iLBC [Duric et al., 2004], all of them used in VoIP.

There are many CELP variants, depending on how the excitation codebook is structured and exploited. Examples such as Algebraic CELP (ACELP) [Adoul et al., 1987, 3GPP, 2018a], Vector Sum CELP (VSELP) [Gerson and Jasiuk, 1990, ETSI, 2000] or Conjugate Structure CELP (CS-CELP) [Salami et al., 1998, ITU-T, 2012b] have been standardized and adopted in the industry.

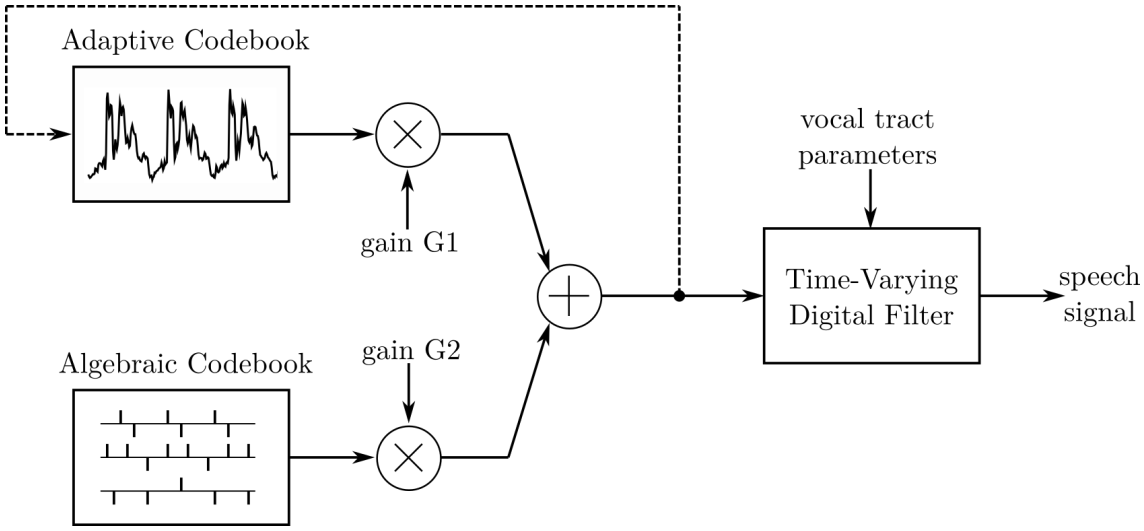


Figure 2.17 – Simplified diagram of an algebraic CELP synthesizer with an adaptive codebook.

Figure 2.17 depicts a simplified model of an adaptive ACELP speech synthesizer. The excitation signal is formed from two sources: an adaptive dictionary built upon the past excitation signal and a sparse algebraic codebook that encodes a difference between the previous and the current frame. The codebook gains and the sparse algebraic excitation are estimated during the encoding stage by minimizing the MSE.

The best-known example of an ACELP coder is Adaptive Multi-Rate (AMR) codec implemented in cellular networks. The codec is available in narrowband (AMR-NB) [3GPP, 2018a] and wideband (AMR-WB) [ITU-T, 2012c] versions and compresses sampled speech signals respectively to 4.75 - 12.2 kbps and 6.6 - 23.85 kbps. AMR performs linear prediction analysis on 20 ms frames and computes LPC filters with 10 (AMR-NB) or 16 (AMR-WB) taps encoded as line spectral pairs (LSP) [Soong and Juang, 1984].

The specificity of AMR is its sparse algebraic excitation encoder. The codebook search is performed on 5 ms subframes, which in the narrowband scenario results in processing vectors of 40 samples. In the 12.2 kbps mode of AMR-NB, all positions in a subframe are grouped into five tracks, where each track contains two pulses with amplitude -1 or $+1$ (Table 2.6). The encoder chooses positions for ten non-zero pulses so that the combined output from the algebraic and adaptive dictionaries gives the closest representation of excitation. As more subframes are encoded, the adaptive dictionary builds a refined model of excitation.

The construction of adaptive ACELP coders is motivated by the observation that excitation and the vocal tract change slowly in time [Gay et al., 1974, Edwards and Chang, 2013]. Adaptive ACELP coders can represent complex excitation signals with high perceptual quality, given a

sufficiently large portion of a stationary speech signal. This valuable property in speech encoding becomes an obstacle in pseudo-speech transmission over voice channels with ACELP compression. A highly variable encrypted signal is very likely to be smoothed during compression, and many information-carrying signal details could be discarded. Therefore, a robust pseudo-speech signal should fit into the general source-filter speech model and be adapted to the ACELP excitation encoder.

Table 2.6 – Possible positions of individual pulses in the algebraic codebook in the 12.2 kbps mode of the narrowband AMR.

Track	Pulse	Positions
1	i_0, i_5	0, 5, 10, 15, 20, 25, 30, 35
2	i_1, i_6	1, 6, 11, 16, 21, 26, 31, 36
3	i_2, i_7	2, 7, 12, 17, 22, 27, 32, 37
4	i_3, i_8	3, 8, 13, 18, 23, 28, 33, 38
5	i_4, i_9	4, 9, 14, 19, 24, 29, 34, 39

2.5.3 Perceptual coders

In contrast to LPC coders, perceptual coders aim to encode general audio signals with speech and background music. This flexibility of audio coders encourages their use in speech-related applications in place of LPC coders. In the past years, a big limitation of general audio coders was their large algorithmic delay. For example, AAC [ISO/IEC, 2006] at a 24 kHz sampling rate and a 24 kbps bitrate introduced delay of about 100 ms, too high for use in natural speech conversation that requires end-to-end transmission delay no larger than 150 ms [Benesty et al., 2008, Chap. 18]. Gradual algorithmic optimizations introduced in recent coders opened new possibilities in many real-time audio applications.

Perceptual coders rely on filterbank-based audio coding. The first processing step involves transforming the signal to a spectral representation using analysis filterbanks, such as the Modified Discrete Cosine Transform (MDCT) [Princen et al., 1987], polyphase filterbanks [Rothweiler, 1983], or hybrid structures [Brandenburg et al., 1992]. In the next step, the transformed signal is analyzed following a perceptual model, e.g., hearing thresholds, temporal and frequency masking, and masking between tonal and noise signals [Hellman, 1972]. The perceptual analysis gives information for optimum quantization and bit allocation.

The examples of general audio coders for real-time applications are Low-Delay Advanced Audio Coder (AAC-LD) [ISO/IEC, 2006] used in Apple's VoIP application FaceTime, and Opus' Constrained Energy Lapped Transform (Opus-CELT) [Valin et al., 2012] used for real-time audio streaming and in-chat gaming. Both coders are based on MDCT with partially overlapping windows and perceptually-weighted spectrum quantization. Low algorithmic latency (20 ms and below) compared to classical audio coders has been achieved by shortened frames, reduced look-ahead in the filterbank analysis, and introducing several computational optimizations. However, the data rates characterizing these coders, 32-64 kbps for AAC-LD and 48-128 kbps for CELT, are very high. Consequently, they cannot be considered as good speech models for speech encryption.

2.5.4 Low-bitrate parametric coders

At bitrates below 4 kbps, high quality speech encoding is very hard to obtain. Instead, parametric coders capture only the key perceptual features of speech, preserving speech intelligibility at an acceptable perceptual quality. A common characteristic of parametric coders is the use of simplistic speech models that enable efficient compression at expense of lower speech fidelity. Due to significant quality degradation, parametric speech coders are used in critically constrained environments, such as satellite and military communications [Benesty et al., 2008, Chap. 16]. Examples of parametric coding techniques are Mixed Excitation Linear Prediction (MELP) [McCree et al., 1996], Multiband Excited coding (MBE) [Griffin and Lim, 1988], Sinusoidal Transform Coding (STC) [McAulay and Quatieri, 1986] and Waveform Interpolation (WI) [Kleijn, 1991].

Sinusoidal coders model a continuous-time speech signal locally as a sum of N equally-spaced sine waves:

$$s(t) = \sum_{k=1}^{N-1} A_k \sin(k\omega_0 \cdot t + \phi_k), \tag{2.10}$$

where ω_0 is the fundamental angular frequency, and A_k and ϕ_k are the amplitudes and the phases of harmonics. While the harmonic model is accurate for voiced sounds, it can also satisfactorily represent unvoiced speech [McAulay and Quatieri, 1986]. The principle of sinusoidal coding is thus to estimate the harmonic parameters for each speech frame. During synthesis, the mismatch between consecutive frames is mitigated using sine interpolation or the overlap-add technique.

Probably the most popular sinusoidal coder is the open-source Codec2 [1] designed by Rowe, which offers speech compression rates 0.7 - 3.2 kbps and full speech intelligibility. In Codec2, each speech frame of 20 ms or 40 ms is parametrized by its fundamental frequency, energy, spectral envelope (amplitudes of harmonics), and voicing. The phase information in the compressed signal is discarded and regenerated by the decoder.

Another model is used in MBE coders, where a speech frame in the spectral domain $\mathrm{S}(\omega)$ is represented as the product of the spectral envelope $\mathrm{H}(\omega)$ and the excitation spectrum $|\mathrm{E}(\omega)|$:

$$\mathrm{S}(\omega) = \mathrm{H}(\omega)|\mathrm{E}(\omega)|. \tag{2.11}$$

The MBE coder subdivides the excitation spectrum into several bands (20 or more) and classifies each band as voiced/unvoiced. Consequently, the coder parametrizes speech signals by the fundamental frequency, the spectral envelope, and a sequence of voicing decisions. The decoder synthesizes all voiced bands using the sinusoidal model and reconstructs unvoiced bands with white noise shaping. The improved version of MBE (IMBE) operating at 6.4 kbps is implemented in satellite communications [Hardwick and Lim, 1991].

Although the most significant progress in source-filter and parametric coding took place in the 80s and 90s, it still inspires new speech synthesis techniques. Recently, Jean-Marc Valin (Mozilla) and Jan Skoglund (Google LLC) proposed a new architecture of a speech coder, named LPCNet [Valin and Skoglund, 2019, Valin and Skoglund, 2019]. The mathematical model of speech synthesis implemented in LPCNet is similar to all other LPC coders representing discrete-time

1. https://www.rowetel.com/wordpress/?page_id=452

speech signal $s[n]$ as the linear combination of past samples $p[n]$ and a residual (an excitation) $e[n]$:

$$s[n] = p[n] + e[n] \tag{2.12}$$

$$p[n] = \sum_{k=1}^{16} \alpha_k s[n-k] \; . \tag{2.13}$$

Unlike in traditional coders, however, the excitation signal $e[n]$ is modeled by using a trained neural network.

The block diagram of the synthesis part of LPCNet is illustrated in Figure 2.18. The coder operates on 10 ms speech frames sampled at 16 kHz. The frames are encoded using 20 (quantized) features: a pitch period, a soft voicing, and 18 Bark-scale cepstral coefficients similar to previously described MFCC. During speech synthesis, these 18 cepstral coefficients are transformed into linear prediction coefficients α_1, ..., α_{16}.

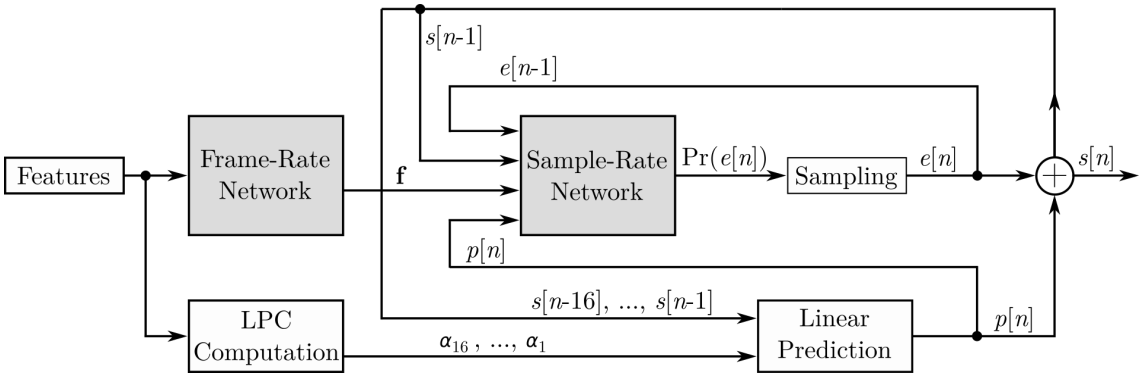


Figure 2.18 – LPCNet architecture [Valin and Skoglund, 2019].

The speech synthesizer consists of two neural networks. The first network takes as an input the 20-dimensional feature vector and once per 10 ms outputs a conditioning vector $\mathbf{f}$ used by the second so-called sample-rate neural network. The role of the sample-rate network is to predict a probability distribution of the excitation $\Pr(e[n])$ at the time n, given four inputs: the previous excitation sample $e[n-1]$, the past synthesized speech sample $s[n-1]$, the current prediction $p[n]$ and the computed conditioning vector $\mathbf{f}$. The excitation $e[n]$ is obtained by sampling from the predicted distribution. This last feature distinguishes LPCNet from deterministic speech coders.

LPCNet achieves remarkable results in producing near-natural wideband speech and compressing speech signals down to 1600 bps. Additionally, the optimized architecture of the coder makes its implementation on low-power devices an achievable goal. However, LPCNet requires a considerable amount of data and time to train the networks. Nevertheless, after extensive training, the coder's performance shows how little information is needed to produce intelligible speech in real-time and that speech quality can be enhanced using synthetic-only excitation.

Parametric coders rely heavily on Vector Quantization (VQ) [Gray, 1984] and other codebook-based techniques, providing significant compression gains [Ramasubramanian and Doddala, 2015]. In combination with a simplified structure of speech models, low-bitrate parametric coders are desirable candidates for use in speech encryption. On the other hand, data transmission over a voice channel with parametric speech coding is significantly constrained by the implemented speech model. Unless the pseudo-speech signal structure is adapted to a particular channel, the achievable data rate would be prohibitively low.

2.6 Digital voice channels

Voice channels cannot be analyzed apart from the infrastructure that enables long-distance communication. The system parameters (e.g., transmission delay, drop-out probability, available bandwidth, handheld devices used) significantly impact the voice channel properties. Furthermore, a recorded speech signal is usually subject to quality-enhancing processing, such as background noise suppression or echo cancellation. While these algorithms considerably improve the user's experience of speech perception, they also add more complexity upon standard voice compression.

This section outlines two popular mobile voice communication systems: wireless cellular networks (GSM/UMTS) and VoIP. The description focuses on a few selected elements of the analyzed systems, which detail important voice channel properties and may impact the encrypted pseudo-speech transmission.

Cellular networks of the second and the third generation (GSM/UMTS) are circuit-switched systems for mobile voice communication [Holma and Toskala, 2005, Chap. 2, Schwartz, 2005, Chap. 8, Eberspächer et al., 2008]. Figure 2.19 depicts a simplified schematic of the established vocal connection between two users equipped with proper mobile phones and registered to different cellular base stations (BS). In the presented situation, one-way voice transmission over cellular networks consists of three stages. The speech is firstly encoded on the mobile phone by the AMR coder and sent as data signal from the mobile device up to the BS. Afterward, the station decompresses the encoded bits into 8 kHz PCM speech samples at the 64 kbps bitrate [ITU-T, 1988f] and forwards the signal to the destined BS through the public switched telephone network (PSTN), or the integrated services digital network (ISDN) [ITU-T, 1993]. Upon reception of the PCM samples, the receiving BS encodes the speech again with AMR and sends the data to the recipient.

The operation of cellular networks reveals that the sent voice can be processed twice by two different vocoders. Provided that AMR coder strongly compresses speech signals, the mean opinion score (MOS) [ITU-T, 1996a, ITU-T, 2016a] of the perceptual speech quality in cellular communication is unsurprisingly mediocre [ETSI, 2018c]. On the other hand, circuit-switched vocal links in cellular networks have to meet very stringent requirements for the Quality of Service (QoS). More specifically, the transmission delay between two mobile phones should not exceed 100 ms for the 95th percentile of the delay distribution, and mobile phones are granted the minimum bandwidth enabling a smooth conversation [3GPP, 2018c].

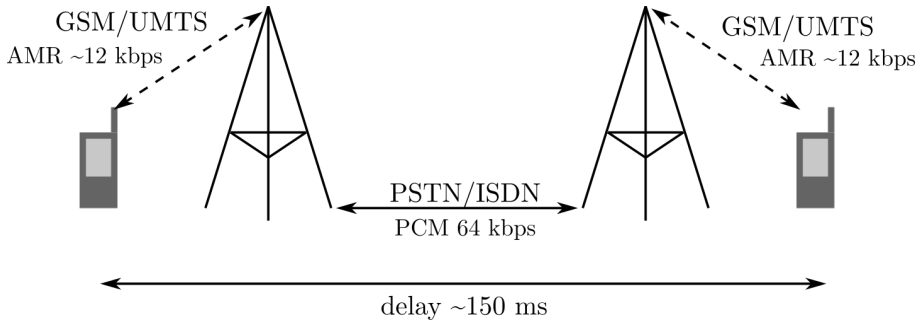


Figure 2.19 – Simplified model of a 2G/3G cellular voice connection.

Unlike cellular networks, VoIP does not guarantee any QoS. According to the International Standardization Organization (ISO) recommendation G.114, the maximum transmission delay should stay below 150 ms [ITU-T, 2003a]. These proposed guidelines often cannot be met, especially in large-distance communication with the communication latency as large as 300 ms and above [ITU-T, 2003a]. Moreover, the quality of transmitted speech depends on the available network throughput and reliability. Excessive average latency, jitter, and lost packet probability of a poorly performing network may significantly lower the user's experience [Benesty et al., 2008, Chap. 15]. In order to counteract the quality degradation, several techniques have been adopted in VoIP technology, such as traffic prioritization, dynamic bandwidth allocation, half-duplexing, lost packet concealment, and discontinuous transmission (DTX). However, given excellent network conditions, VoIP offers a high voice quality with up to 4.5 MOS score [Goudarzi et al., 2011].

It is worth noticing that cellular networks and VoIP pose different challenges to the secure voice communication system under study. Cellular networks offer reliability and stability, which is an advantage for maintaining pseudo-speech synchronization. Nonetheless, strong voice compression significantly limits the available throughput for encrypted speech signals and prevents decent voice quality after signal decryption.

On the contrary, mild signal compression and the high throughput available in VoIP applications enable sending more encrypted data in the audio signal. However, the high probability of lost and delayed packets may deteriorate communication. From the operational standpoint, the encrypted voice data included in delayed packets is irreversibly lost. Moreover, lost-packet concealment in VoIP that fills missing fragments of the signal using some synthetic signal generator may hinder systems' synchronization.

Apart from network traffic management, modern voice communications systems apply a variety of speech-enhancing algorithms on the recorded voice. The most popular among these techniques are Voice Activity Detection (VAD) [Bäckström, 2017, Chap. 13, ITU-T, 2012b], Echo Cancellation (EC) [ITU-T, 2015], Adaptive Gain Control (AGC) [Heitkamper and Walker, 1993], Noise Suppression (NS) [ETSI, 2018b] and Comfort Noise Generation (CNG) [ITU-T, 2012b, ETSI, 2018a].

Voice Activity Detection performs real-time speech analysis to classify voice frames as speech-like or containing either silence or background noise. The classification is made by a specialized unit based on the range of temporal and spectral features (e.g., tonality, spectral envelope, signal intensity, zero-crossing rate) [Bäckström, 2017, Chap. 13]. Depending on the classifier's decision, the analyzed speech frames may be passed through the channel or rejected. Since in natural conversation speakers talk for less than 40% of the time [Freeman et al., 1989], silent frame suppression improves network bandwidth allocation and saves some phone energy.

Some detectors are paired with the speech encoder. For example, Figure 2.20 illustrates a diagram of the VAD classifier cooperating with AMR [3GPP, 2018b]. The classifier performs filterbank analysis of speech frames and re-uses intermediate parameters from the AMR pitch analysis (pitch period, autocorrelation values) and the long-term linear prediction (complex signal autocorrelation vector). Given the inputs, the algorithm compares the estimated background noise level with the filterbank analysis result and calculates the noise-to-signal energy ratio. The classifier raises the VAD flag when the computed ratio stays above an adaptive threshold for some time.

Another negative phenomenon is an echo that causes an annoying sensation of hearing one's own voice with a delay larger than 40 ms [ITU-T, 2003b]. The speech echo often occurs due to a cross-talk between transmission cables or when a loudspeaker's output is again captured by the device's microphone [Lin et al., 2008]. Canceling the echo is usually done using adaptive filtering

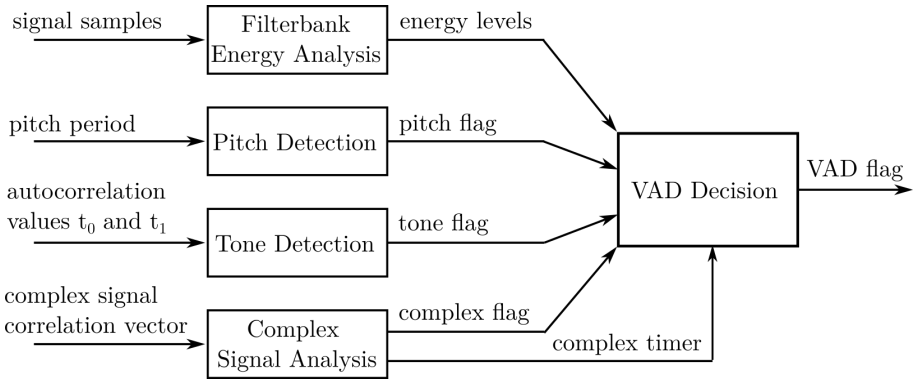


Figure 2.20 – Diagram of the Option 1 VAD included in the AMR encoder [3GPP, 2018b].

[ITU-T, 2015, Benesty et al., 2008, Chap. 45]. The echo cancellation unit models the echo in real-time and subtracts the modeled signal from the recorded speech.

Although some VAD and speech enhancing algorithms can be found in various recommendations and specifications, their exact implementations are often proprietary solutions unavailable to a broad public. Since phone manufacturers, VoIP application developers, and network providers may incorporate dedicated speech enhancing techniques, accurate voice channel modeling is often intractable. Instead of reverse-engineering these algorithms, it is usually more practical to search for techniques that could reset VAD's counters and consequently block VAD's activation. For instance, Figure 2.21 illustrates the signal sent over the WhatsApp application between the same phones as in the first example presented in Figure 2.1. A sensible manipulation of frequency bands and silence insertions successfully prevented signal suppression. Nevertheless, it is still worth considering the most fundamental properties of speech-enhancing algorithms to understand the principal properties of a good synthetic signal suitable for encrypted data transmission.

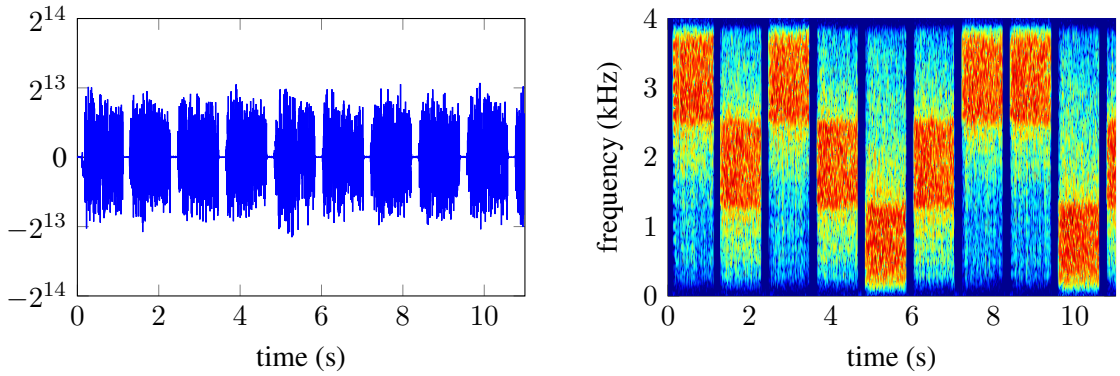


Figure 2.21 – Phase-modulated harmonic signal sent over the WhatsApp application. The frequent alternation of audio bands in the signal and silence insertions may counter adaptive noise suppression and voice activity detection algorithms in a voice channel.

2.7 Summary

This chapter described the principles underlying the operation of voice channels in cellular networks and VoIP. The chapter explained the speech production and auditory mechanism, and gave some hints on the idea behind source-filter coding and perceptual coding. Finally, the chapter presented some selected speech coders and speech processing techniques.

A diversity of speech coding techniques adopted in modern communications suggests that designing a universal technique for sending encrypted speech over arbitrary voice channels in real-time would be extremely difficult. Moreover, secure voice communication over voice channels is possible only when the channel capacity is sufficiently large for sending speech data in real-time. Instead, it may be worthwhile to target specific types of voice channels, depending on a usage scenario. Since cellular networks and many prominent VoIP applications rely on source-filter and waveform coding, it may seem reasonable to make encrypted signals compatible with generic LPC coders. Nevertheless, the pseudo-speech signal must be tuned to a particular voice channel due to implementation differences that modify channel's characteristics. That includes a dedicated strategy for countering the VAD algorithm.

Furthermore, data encoding onto pseudo-speech must be robust against perceptual processing in the spectral domain. Thus, the attractive idea is to relate encoded data with parameters of vowel-like sounds that are compatible with source-filter, perceptual, and waveform coders. The principles of voiced sounds have been exploited in Chapter 3 describing a novel technique for data transmission over voice channels. However, despite the high robustness of the produced signal, we discovered that error-less data transfer over real voice channels cannot be guaranteed. This communication characteristic appeared troublesome because traditional cryptographic algorithms require perfect data reception for decryption. In order to overcome this limitation, we designed a dedicated distortion-tolerant cryptographic scheme that can decrypt data approximately. The distortion-tolerant speech encryption scheme is introduced and described in Chapters 4 and 5.

Due to a limited bandwidth available for communication over digital voice channels, it is impossible to encrypt and send a complete speech signal. Instead, it is necessary to compress speech before encryption, preferably using one of the low-bitrate parametric speech coders. The most suitable coders are those using a simple speech model with a clear relation between the model parameters and the perceptual speech qualities (pitch, loudness, timbre). For example, the sinusoidal speech model and the Multiband Excited (MBE) model are good candidates. However, a common disadvantage of parametric coders is their low quality of decoded speech. The new generation of synthesizers using trained neural networks, such as LPCNet, may significantly improve perceptual speech quality at a very low bitrate.

Chapter 3 introduces a technique for sending data over channels with LPC coders. The technique relies on short harmonic waveforms that mimic the structure of harmonic vowel signals. The method is compatible with cellular networks and many VoIP applications, and is robust against some VAD algorithms.

CHAPTER 3

Data transmission over voice channels

This chapter presents a novel Data over Voice (DoV) technique based on codebooks of short harmonic waveforms. The technique provides a sufficiently fast and reliable data transfer over cellular networks and many VoIP applications. The new method relies on general principles of Linear Predictive Coding for voice compression (LPC voice coding) and is more versatile compared to solutions trained on exact channel models. The technique gives a high control over the desired transmission rate and robustness to channel distortion. Furthermore, an efficient codebook design approach inspired by quaternary error-correcting codes is proposed.
The proposed technique was thoroughly tested with real voice calls. These experiments have successfully validated the usability of the proposed DoV technique for secure voice communication over cellular networks and VoIP. The chapter details the system parameters, emphasizing the system's security and technical challenges.

3.1 Motivation

Secure data transmission over voice channels (telephone lines) dates back to the early 40s, and the development of the probably first secret telephony system for use during World War II, nicknamed SIGSALY or 'Green Hornet' [Bennett, 1983]. However, the term *modem* ('*mod*ulator-*dem*odulator') in context of a device converting data into the format suitable for transmission originated in the late 50s when referring to *Bell 101 modem* developed by Bell Labs. The modems were part of SAGE (Semi-Automatic Ground Environment), the largest computer-aided information system of that time [Hellige, 1994]. Their role was to transmit digital radar pictures over telephone wires using frequency-shift keying (FSK) modulation, initially at 750 bps and later at 1.3 kbps and 2.1 kbps. In the following years, the rapid progress in modem technology contributed to creating the digital subscriber line (DSL) with data rates over 10 kbps [Golden et al., 2006].

A bit surprisingly, the possibility of exploiting cellular vocal networks for data transmission went unnoticed until the seminal work of Katugampala, Vilette, and Kondoz in 2003 [Katugampala et al., 2003], in which the authors suggested sending encrypted bits of compressed voice between two cellular phones. Since then, cellular vocal networks have attracted attention as a potential high-priority, low-bandwidth data communication channel with errors. The work on Data over Voice (DoV) technology in cellular networks enabled new applications, such as emergency call system eCall [Werner et al., 2009], messaging over voice [Dhananjay et al., 2010], point of sell (POS) financial transactions [Mezgec et al., 2009], automatic network address translator (NAT) traversal [Patro et al., 2011], and secure data and voice communications [Katugampala et al., 2005, Chen and Guo, 2011].

With the quickly expanding data-driven 4G networks and the deployment of 5G networks, the use of voice channels for sending data diminishes. Nevertheless, DoV techniques are still crucial in secure voice communications, for example, provided by Crypto Phones or other specialized devices [Krasnowski et al., 2020]. On the other hand, voice channels can be maliciously used for extruding private data or in Advanced Persistent Threat (APT) attacks [Lee et al., 2017].

The crucial challenges related to DoV are a consequence of principles underlying digital voice channels. Namely, voice channels aim at preserving speech intelligibility and quality while reducing the perceptually redundant information. In contrast to classical data channels, voice channels significantly distort the sent signal due to transcodings and audio processing. Moreover, modern digital voice channels are selective to signal parameters conforming to the speech model adopted in a particular system. To mitigate signal degradation caused by voice channels, several authors proposed DoV techniques based on encoding the data signal into speech-like parameters, codebook training, or optimized modulation techniques.

Katugampala et al. [Katugampala et al., 2003] proposed a system that uses predefined codebooks to map bits into vocal parameters: energy, pitch, and spectral envelope (encoded as line spectral pairs, LSP [Soong and Juang, 1984]). The encoded parameters are transformed into a pseudo-speech signal adapted to transmission over a cellular network. Data extraction is done by a paired speech analyzer, which restores vocal parameters from the signal and decodes codebook indices. The system enabled transmission over a real GSM voice channel at the rate of 3000 bps with 2.9% BER [Katugampala et al., 2005]. Similar techniques were presented by Ozkan et al. [Özkan and Örs, 2015], and Rashidi et al. [Rashidi et al., 2008], who achieved respectively transmission rates of 1600 bps and 2000 bps in a simulation environment.

LaDue et al. [LaDue et al., 2008], and Sapozhnykov and Fienberg [Sapozhnykov and Fienberg, 2012] investigated genetic and pattern matching algorithms to construct codebooks of short

speech-like waveforms. Instead of synthesizing pseudo-speech, the authors proposed encoding bitstream directly into a sequence of symbols selected from a trained wavetable. Upon reception, received symbols were decoded with a bank of matched filters. The technique achieved the remarkable 4000 bps with 2.3 % BER over enhanced full rate (EFR) voice channel. Unfortunately, the training process was time-consuming and required considerable computational resources. Moreover, the obtained wavetable was compatible with a unique channel model and hence impractical in real communication.

The problem of long and heavy computations has been tackled by Shahbazi et al. [Shahbazi et al., 2009], and Boloursaz et al. [Boloursaz et al., 2013], who simplified the codebook construction by limiting the search to signals from the TIMIT speech database [Zue et al., 1990]. Parallelly, Kazemi et al. [Kazemi et al., 2015] proposed an exciting idea to exploit sphere packing techniques to construct waveforms with a large minimum distance and an improved detection rate.

Finally, there exists a range of DoV techniques based on well-established, classical signal modulation. Zhan Xu [Xu, 2017], Chmayssani and Baudoin [Chmayssani and Baudoin, 2008] tested by simulations phase shift keying modulation (PSK) and quadrature amplitude modulation (QAM), and achieved bitrates within the range 1 - 3 kbps. Ali et al. [Taleb Ali et al., 2013] exploited M-ary frequency shift keying (M-FSK), whereas Dhananjay et al. [Dhananjay et al., 2010] introduced a modified binary FSK (BFSK) tolerant to a small frequency deviation. Chen and Guo [Chen and Guo, 2011] reported a solution using orthogonal frequency division multiplexing (OFDM) modulation combined with PSK.

An inspiring technique based on Amplitude Shift Keying (ASK), named PCCD-OFDM-ASK, has been presented by Mezgec et al. [Mezgec et al., 2009]. Phase-Continuity and Context Dependency (PCCD) refers to techniques providing phase continuity of the modulated signal. In PCCD-OFDM-ASK, blocks of 8-bit sequences are encoded onto eight orthogonal harmonics, numbered from 1 to 8. In contrast to classical OFDM, each bit in the 8-bit block is represented by the presence or absence of an orthogonal carrier. For instance, the binary 8-bit sequence '10001010' is mapped to a symbol with harmonics present only at positions 1, 5, and 7. The scheme offers robust transmission up to 500 bps over real cellular voice channels.

This chapter introduces a new DoV codebook-based modulation over cellular networks and VoIP for the needs of secure voice communication. The novelty comes from our simplified and universal codebook design process compared with the usual extensive codebook training on a selected voice model. Nevertheless, the method can be adapted to a particular channel, avoiding codebook over-tuning in the presence of fluctuating channel characteristics. Modulation parameters are easily adjustable in order to balance the transmission bitrate and the robustness to errors.

The proposed technique was thoroughly tested with real voice calls. The scheme achieves up to 6.4 kbps over VoIP voice channels using 4G wireless network and 2.4 kbps over 3G cellular calls (see Section 3.5.3). It also enables safe voice transmission with an effective binary error rate significantly below 1%.

This chapter is organized as follows. Section 3.2 outlines challenges related to sending data over voice channels with LPC-based speech compression. Section 3.3 investigates signal distortion introduced by three selected LPC coders: AMR, Speex, and Opus-Silk. Section 3.4 describes the novel DoV technique, including codebook construction, signal generation, and demodulation. Section 3.5 presents performance results obtained by simulations and real-world experiments, and Section 3.6 proposes a secure voice communication scheme using DoV. Finally, Section 3.7 concludes the chapter.

3.2 Digital voice channels

This section introduces crucial challenges related to data transmission over voice channels. It outlines the specific behavior of voice channels, very different compared to classical communication channels, and highlights the desired properties of DoV signals.

3.2.1 Voice channel characteristics

In real-world implementations, a complete voice channel is typically the concatenation of algorithms that transform a speech signal into binary data suitable for transmission over the network. Despite the lossy nature of speech processing, the received binary information is sufficient to re-synthesize a speech perceptually similar to the initial. However, from a DoV perspective, it is more convenient to consider voice channels as communication channels with particular constraints and signal distortion characteristics.

The core elements of any digital voice channel are *voice codecs*, which compress and encode sampled speech waveform exploiting principles of speech production and perception [Rabiner and Schafer, 2011]. Real-time voice coders usually process speech on a frame basis by mapping portions of a speech waveform into sets of vocal parameters. These algorithms may perform high-pass filtering, differential encoding, and adaptive quantization to improve the compression ratio depending on the available network throughput. Unfortunately, such operations add memory and latency to a voice channel, and make it non-linear and non-stationary.

In addition to voice compression, modern voice communication systems apply techniques such as *Voice Activity Detection* (VAD) [Bäckström, 2017], *Adaptive Gain Control* (AGC) [Heitkamper, 1995] or *Noise Suppression* (NS) [Tsoukalas et al., 1997]. In opposition to voice coders, the implementation of these algorithms is rarely public and their impact on the DoV cannot be fully predicted.

Combining all the mentioned elements of real voice channels, achieving an analytic model of signal distortion is usually intractable. Nevertheless, it is still worthwhile to consider the most fundamental properties of voice channels and construct the DoV scheme agnostic to small variations of the voice channel characteristics.

3.2.2 LPC coders

Most of the voice coders operating in the upper-middle bitrate range (10 kbps –16 kbps) listed in ITU, IETF and 3GPP standards, and which are widely adopted in cellular and VoIP systems, rely on *Linear Predictive Coding* (LPC). LPC coders take their inspiration from the simplified speech production model, often referred to as a source-filter model [Fant, 1960, Lochbaum and Kelly, 1962]. According to the model, voice sound originates from a single source $e(t)$ and is filtered by a vocal tract with an impulse response $v(t)$. Such a simplification is justified for voiced and stationary sounds, which can be approximately represented by the buzzing excitation produced in the glottis and shaped when passing through the pharynx and between tongue, teeth, and lips. The resulting signal has the form $s(t) = e(t) * v(t)$, where $*$ denotes the convolution product.

However, considering voice as the convolution of excitation and vocal tract shaping would be of little practical value without effective methods for separating these components. The excitation and vocal tract characteristics can be well approximated during LPC analysis (hence LPC coders). The outputs of LPC analysis consist of a linear prediction filter describing the vocal tract's filtering effect and a residual that can be viewed as an excitation signal.

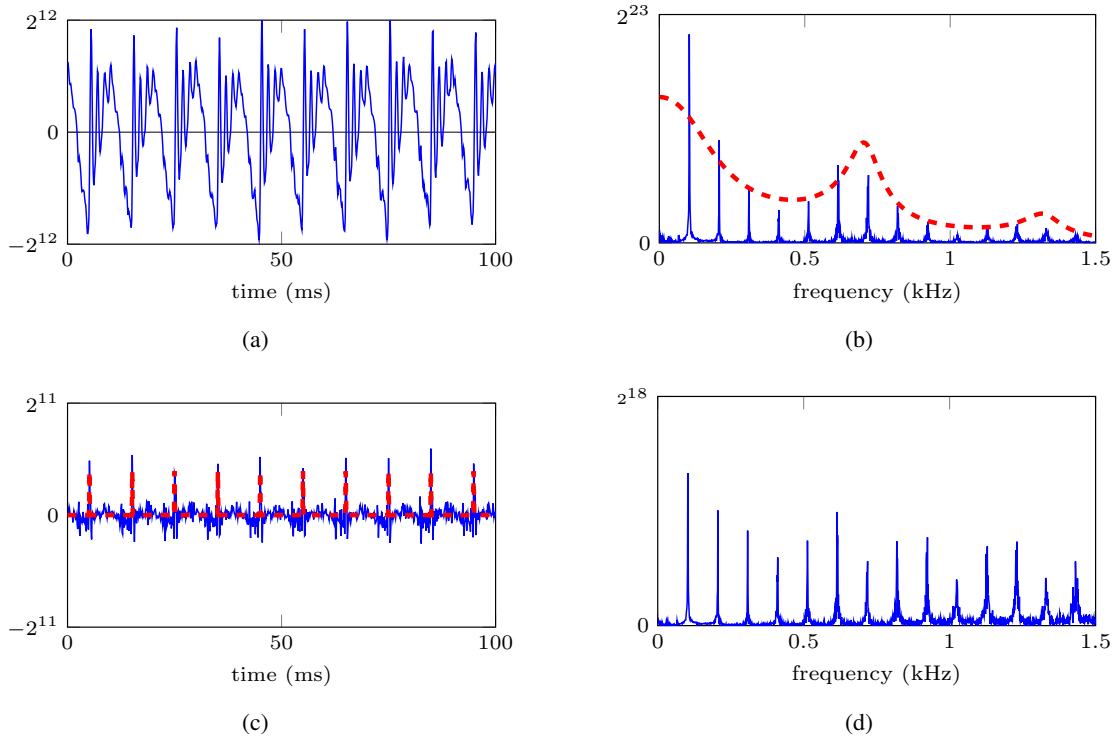


Figure 3.1 – LPC analysis of vowel $/a/$: (a) time domain waveform, (b) spectrum of the waveform (blue solid line) and frequency response of the 12th order LPC filter (red dashed line), (c) residual of LPC analysis (solid blue line) and excitation peaks (red dashed line), (d) frequency spectrum of a residual.

As an example, Figures 3.1(a) and 3.1(b) present 100 ms of a real recording of vowel $/a/$ in the time and the frequency domain. It can be noticed that this spectrum has a harmonic structure and could be accurately parameterized by its energy, spectral envelope, and fundamental frequency. The dashed line in Figure 3.1(b), which coincides with the spectral envelope of a vowel, represents the frequency response of the estimated LPC filter. On the other hand, the peaks of the residual signal in Figure 3.1(c) correspond to a buzzing excitation from the glottis. Finally, the frequency spectrum of a residual in Figure 3.1(d) has less different formants (acoustic harmonic resonances), compared to the initial spectrum in Figure 3.1(a). Thus, we can reach the intuitive conclusion that LPC analysis separates the spectral envelope from the harmonic content of the signal.

Source-filter separation emphasizes the relevant vocal information, which is advantageous in signal compression. Figure 3.2 depicts a simplified diagram of speech analysis and synthesis by a generic LPC coder. The encoder estimates LPC coefficients and calculates the residual of a small portion of speech (typically 5ms –20ms). Lossy encoding of the residual puts stress on preserving the harmonic content of the speech, whereas LPC filters are often weighted to boost formants, taking advantage of the human auditory system's specificities and information redundancy. From this point, it is understandable that vocal parameters in a waveform are usually well preserved during compression, while the less speech-like are removed. The output waveform is also smoothed in the time and spectral domains to remove ringing effects caused by frame-based processing.

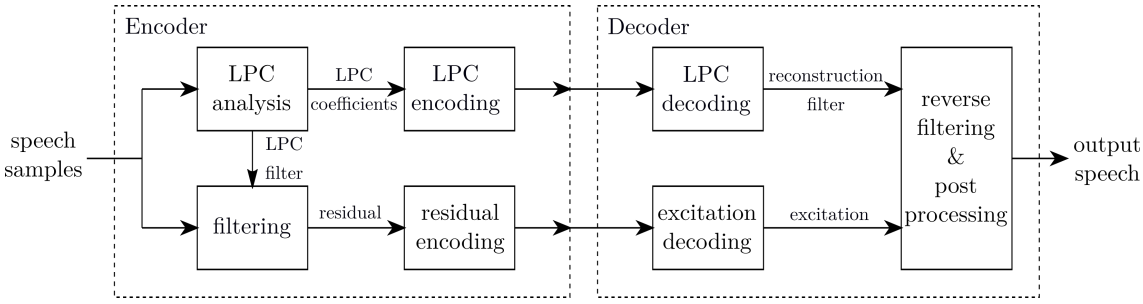


Figure 3.2 – Simplified diagram of LPC encoder and decoder.

Linear Predictive Coding achieves remarkable results in representing and compressing smoothly varying voiced sounds but often struggles with encoding short and noisy plosives (like /p/ or /t/), which do not fit into the source-filter speech model. To improve the robustness for noisy sounds, LPC coders incorporate more flexibility into the excitation encoder. This observation suggests that the potential performance of the DoV technique would mostly depend on the accuracy and reaction time of residual encoding.

Despite preserving core speech intelligibility, time-domain LPC coding destroys the fine time-structure of compressed signals. Thus, it is not obvious how voice channels equipped with LPC coders modify the sent signal. In Section 3.3, we describe a simplified framework that will allow us to evaluate the typical distortion introduced by LPC voice channels.

3.3 Data over LPC voice coders

This section presents a novel DoV technique based on codebooks of phase-modulated harmonic waveforms. The proposed solution is the result of extensive simulation experiments with three representative LPC narrow-band coders: AMR [3GPP, 2018a], Speex v1.2 [Herlein et al., 2009] and Opus-Silk v1.3.1 [Valin et al., 2012].

The section begins with a thorough analysis of signal distortion characteristics caused by selected voice compression algorithms. The investigation leads to a significant improvement in harmonic signal demodulation. Finally, the section proposes a simplified codebook design approach.

3.3.1 Multi-tone modulation over LPC voice coders

By their construction optimized to vowel sounds, LPC coders are suitable for synthesizing multi-tone signals. On the other hand, the versatility of residual encoding allows easy manipulation of phase information, which above 2 kHz typically plays a lesser role in speech intelligibility [Rabiner and Schafer, 2011, Alves-Pinto et al., 2014]. Combining phase modulation with multiple subcarriers is particularly interesting, as it opens the possibility of applying spectrally-efficient orthogonal frequency-division multiplexing (OFDM) modulation [Nee and Prasad, 2000]. The OFDM approach has been already analyzed in the context of DoV in [Chen and Guo, 2011]. Their solution is based on 27 independently modulated carriers and achieved a high bitrate of 2.4 kbps over the (now obsolete) RPE-LTP GSM voice coder at an acceptably low error rate.

Figure 3.3 presents the signal-to-noise ratios (SNR) of a multi-tone signal compressed by AMR, Speex, and Opus-Silk at different compression rates. It may be noticed that the distortions

introduced by the different coders are roughly similar. However, the large amount of distortion poses a big challenge for reliable data transmission, especially at compression bitrates below 10 kbps. Thus, a better understanding of the characteristics of signal distortion would help designing a more robust communication scheme. For the sake of consistency, the following experiments were made only for fixed compression bitrates: AMR 12.2 kbs, Speex 11 kbps, and Opus-Silk 12 kbps.

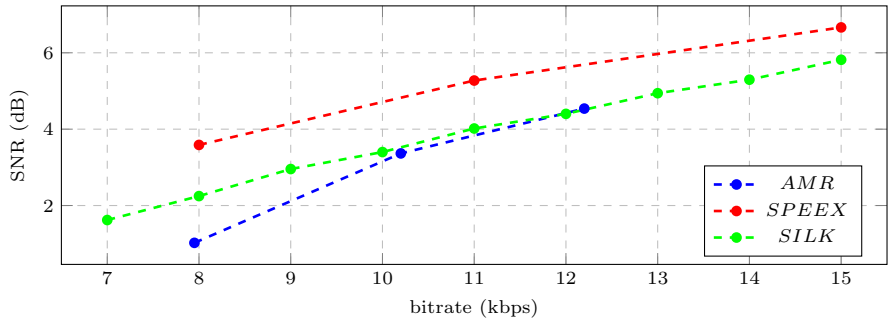
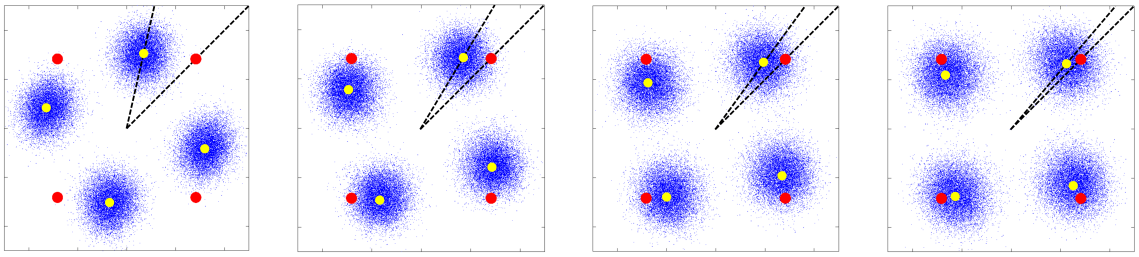


Figure 3.3 – SNR of a multi-tone signal compressed by a selection of LPC coders. The multi-tone signal consisted of eight harmonics at frequencies 400 Hz, 800 Hz, ..., 3200 Hz with a 400 Hz step. The harmonics were independently phase-modulated with a modulation of order 4 and a modulation rate of 200 baud.

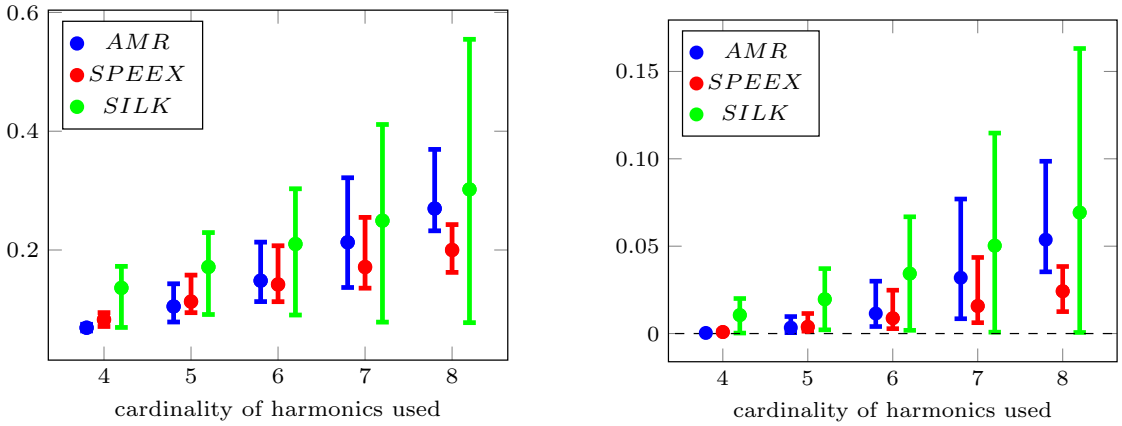


(a) 400 Hz. (b) 800 Hz. (c) 1200 Hz. (d) 1600 Hz.

Figure 3.4 – Scatter plots of a four-harmonic signal compressed by AMR. Each plot represents the distortion of one phase-modulated harmonic at 400 Hz, 800 Hz, 1200 Hz, and 1600 Hz, with a modulation rate of 200 baud. Blue points correspond to compressed symbols, red dots denote the initial phase constellation, whereas yellow dots denote the sample means of compressed four symbols. The angle of the phase shift (restricted by black rays) varies in frequency.

Since LPC coders process the signal jointly, it is not clear how the presence of other harmonics affects the distortion of each component. The distortion introduced by each studied coder has a similar nature, as presented in Figure 3.4. Apart from random noise-like distortion, all samples are subject to constant phase shift (this effect was also observed in [Lee et al., 2017, Xu, 2017]). The phase shift depends on the frequency and the specific LPC coder, but not on symbol duration. The phase shift is probably introduced during speech synthesis by the LPC reconstruction filter with a non-uniform phase response.

Figure 3.5 presents the energy-normalized variance of spectral distortion and related error rates of phase detection in multi-tone signals compressed by a selection of LPC coders. It can be noticed, that there is a direct relation between the variance of distortion and the error rate. In addition, as



(a) Variance of distortion.

(b) Error rate of phase detection.

Figure 3.5 – The average energy-normalized variance of distortion and related error rates of phase detection in multi-tone signals compressed by a selection of LPC coders. The averaged variance was measured over the in-phase/quadrature (I/Q) representation for the four symbols constellation. The four variances are computed along radial lines from the origin to the sample mean of the received decoded symbols (see Figure 3.4). The initial multi-tone signal consisted of four independently phase-modulated harmonics at frequencies 400 Hz, 800 Hz, 1200 Hz, and 1600 Hz, with a modulation order 4 and a modulation rate of 200 baud. Then, the set of carriers was expanded by adding harmonics at 2000 Hz, 2400 Hz, ..., 3200 Hz with a 400 Hz step. Colored bars denote the lowest and highest values among harmonics, and bullets indicate the average.

the cardinality of harmonics in the multi-tone signal goes up, the variability of error rates rises. Nevertheless, harmonics are not distorted uniformly, which is especially noticeable for Silk. It is because the codec puts a more significant emphasis on preserving lower frequencies [Valin et al., 2012], especially important for the auditory perception of voice [Gold et al., 2011].

The sample density distributions of this noise-like distortion are approximately Gaussian, like those presented in Figure 3.6. As the frequency goes up, the width (i.e., variance) is getting larger. This observation supports the intuition that the harmonics at lower frequencies are generally less distorted by compression.

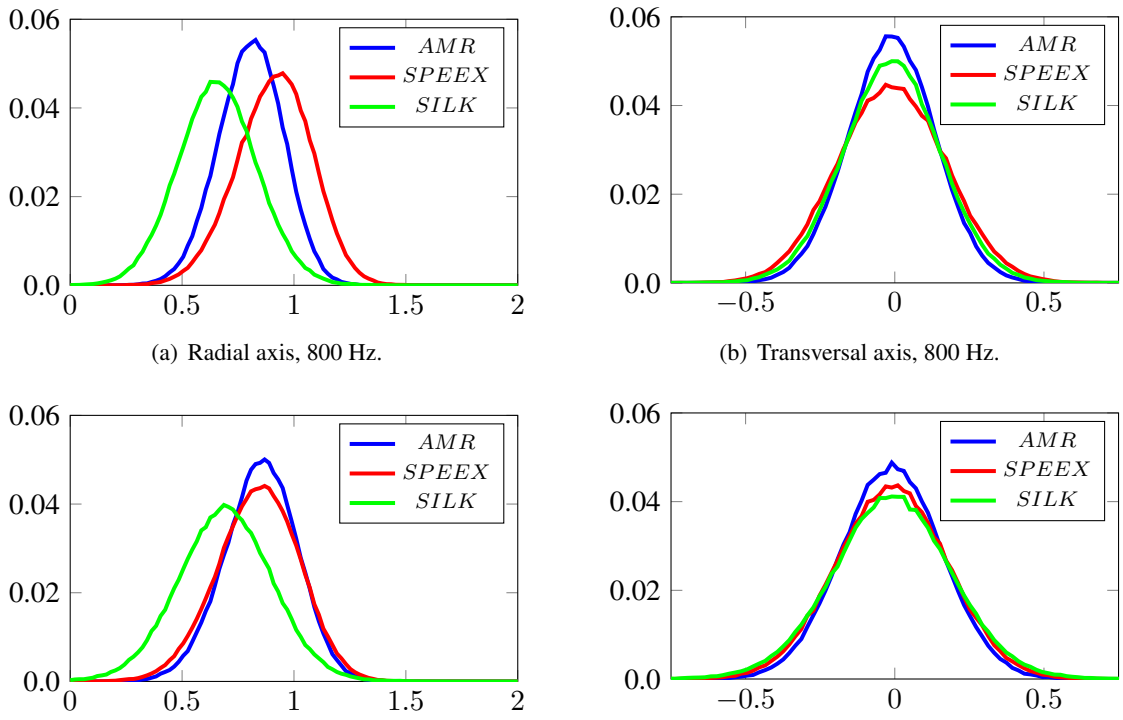
Figures 3.7(a) and 3.7(b) present Mardia's bivariate skewness and kurtosis of a noise-like distortion. Mardia's skewness and kurtosis of a p-variate random sample $\mathrm{x}_1, \ldots, \mathrm{x}_n$ whose sample mean vector $\bar{\mathbf{x}}$ and sample covariance S are defined as [Mardia, 1970]:

$$\text{skewness} = \frac{1}{n^2} \sum_{k=1}^{n} \sum_{\ell=1}^{n} \Big[(\mathbf{x}_k - \bar{\mathbf{x}}) S^{-1} (\mathbf{x}_\ell - \bar{\mathbf{x}}) \Big]^3 , \tag{3.1}$$

$$\text{kurtosis} = \frac{1}{n} \sum_{k=1}^{n} \Big[(\mathbf{x}_k - \bar{\mathbf{x}}) S^{-1} (\mathbf{x}_k - \bar{\mathbf{x}}) \Big]^2 . \tag{3.2}$$

For a sample taken from a p-variate normal distribution, the statistics simplify to:

$$\text{skewness} = 0 \qquad \text{and} \qquad \text{kurtosis} = p(p+2) . \tag{3.3}$$



(a) Radial axis, 800 Hz.

(b) Transversal axis, 800 Hz.

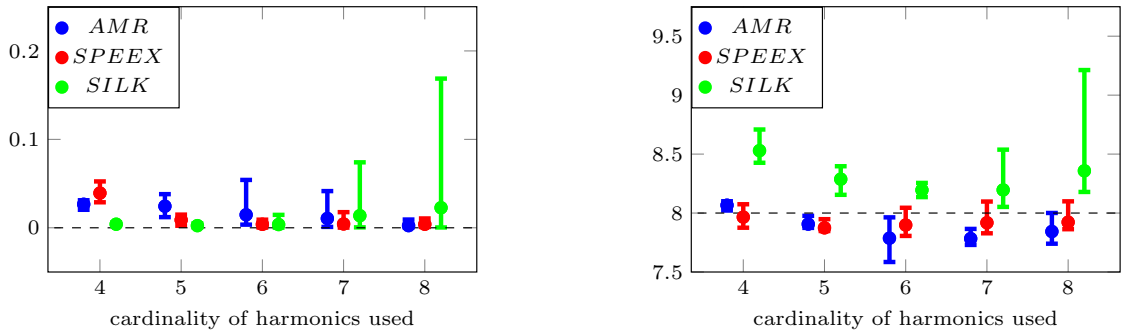
(c) Radial axis, 1600 Hz.

(d) Transversal axis, 1600 Hz.

Figure 3.6 – The average sample probability density function of a distortion of two harmonics at frequencies 800 Hz and 1600 Hz, compressed by a selection of LPC coders. The compressed signal consisted of four independently phase-modulated carriers at frequencies 400 Hz, 800 Hz, 1200 Hz, and 1600 Hz, with a modulation order 4 and a modulation rate of 200 baud. The averaged sample probability density function was measured over the I/Q representation for the four symbols constellation. The four density functions along radial axes are computed along the lines from the origin to the sample mean of the received decoded symbols (see Figure 3.4). The four transversal axes are perpendicular to the radial axes and intersect the sample mean of the received decoded symbols. The x-axes are normalized to the initial amplitude value of each harmonic.

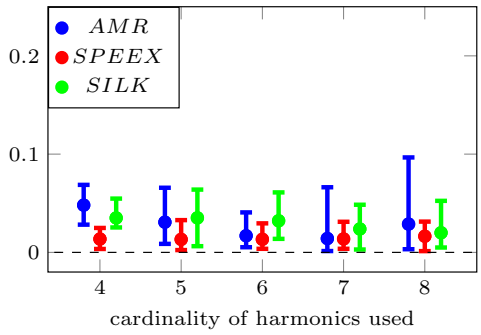
It can be noticed that in the case of AMR and Speex (and to some extent Silk), the computed Mardia's skewness and kurtosis are close respectively to 0 and 8, which are the values characterizing symmetric bivariate normal distribution [Mardia, 1974]. Crucially, distortion is not significantly correlated both in time and between harmonics (Figures 3.7(c) and 3.7(d)). As a result, there is some evidence to treat the noise-like distortion as independent and memoryless. It can be seen as an advantage for demodulation but is also quite surprising because the analyzed coders are deterministic and non-linear. It suggests that distortion characteristics depend not only on LPC coders but also on statistical properties of the modulated signal.

An open question remains, though, for other LPC coders at similar compression rates. Precisely, LPC coding's basic principles do not imply the independence of distortion in the time and the frequency domain. On the other hand, it is arguable that such properties of the proposed modulation, like harmonicity and constant spectral amplitude, are compatible with LPC coding's fundamental properties. Therefore, it should be suitable for the vast majority of LPC coders.

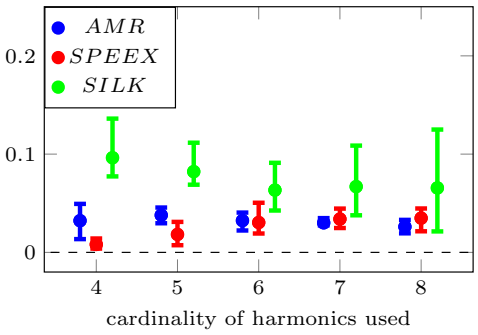


(a) Skewness.

(b) Kurtosis.



(c) Inter-harmonic correlation.



(d) Time correlation.

Figure 3.7 – Statistical parameters of spectral distortion in multi-tone signals compressed by a selection of LPC coders. The initial multi-tone signal consisted of four independently phase-modulated harmonics at frequencies 400 Hz, 800 Hz, 1200 Hz, and 1600 Hz, with a modulation order 4 and a modulation rate of 200 baud. Then, the set of carriers was expanded by adding harmonics at 2000 Hz, 2400 Hz, ..., 3200 Hz with a 400 Hz step. Colored bars denote the lowest and highest values among harmonics, and bullets indicate the average.

3.4 Proposed DoV technique

Figure 3.8 depicts the typical diagram of a data transmission system over voice channel, which uses a codebook of M pre-defined discrete-time audio waveforms. Signal generation is a two-step procedure that firstly encodes the binary input into a sequence of indices $(m_0, m_1, ...)$ and then maps these indices into a concatenation of codebook symbols $s = (\mathbf{s}_{m_0}, \mathbf{s}_{m_1}, ...)$. Finally, the resulting discrete-time audio signal s is played to the (digital) audio input of a voice channel.

On the reception side, the demodulator splits the received sampled audio signal $r = (\mathbf{r}_{m_0}, \mathbf{r}_{m_1}, ...)$ into short chunks of fixed length corresponding to the symbol duration, and then performs symbol-by-symbol matched-filtering with all codebook entries. In the last steps, the demodulator extracts the indices of the codebook symbols giving the highest correlation value and decodes the binary information.

In the proposed DoV technique, a codebook symbol is a vector of waveform samples $\mathbf{s}_m = [s_m[0], ..., s_m[N-1]]$ sampled at 8 kHz and of duration between 2.5-10 ms. Each symbol consists of some small number K (between 7-10) of orthogonal harmonics modulated by quadrature phase-shift keying (4-PSK):

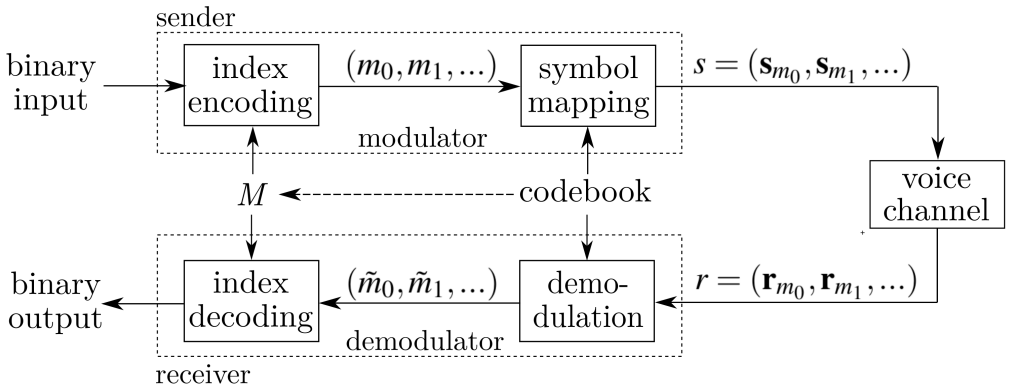


Figure 3.8 – Modulation and demodulation of a discrete DoV signal using a codebook of M predefined discrete audio waveforms.

$$s_m[n] = \Re\left(\sum_{k=0}^{K-1} C_{m,k} \exp\left(j(k+k_0)\omega_0 \frac{n}{N}\right)\right), \quad n = 0, 1, ..., N-1, \tag{3.4}$$

where $0 \leq m < M$ is the symbol index, ω_0 denotes the fundamental angular frequency and k_0 is the subband of the lowest harmonic. Finally, $\mathbf{C}_m = \{C_{m,k} \mid 0 \leq k < K\}$ denotes a sequence of K complex PSK symbols over the phase-amplitude plane:

$$C_{m,k} = A \cdot \exp(j2\pi\varphi_{m,k}/4), \quad k = 0, ..., K-1, \tag{3.5}$$

where A is the amplitude and $\Phi_m = \{2\pi\varphi_{m,k}/4 \mid 0 \leq k < K,\ \varphi_{m,k} \in \mathbb{Z}_4\}$ denotes a sequence of PSK phases (the selection of phase sequences will be detailed in Section 3.4.1). Examples of such waveforms are presented in Figure 3.9.

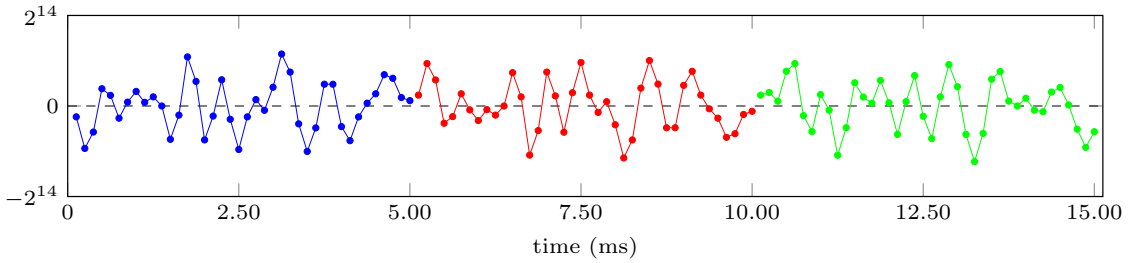


Figure 3.9 – Three discrete-time codebook waveforms (respectively blue, red and green dots) of duration 5 ms and consisting of 10 harmonics at frequencies 600 Hz, 800 Hz, ..., 2400 Hz, with a 200 Hz step.

The symbol structure is equivalent to the discrete-time base-band representation of 4PSK-OFDM modulation [Nee and Prasad, 2000]. Therefore, the received symbols can be processed in a similar manner using subband de-multiplexing. Let $\tilde{\mathbf{C}} = \{\tilde{C}_k \mid 0 \leq k < K\}$ be the sequence of PSK symbols obtained from some received codebook symbol. Assuming a typical AWGN (Additive White Gaussian Noise) channel, the maximum likelihood OFDM symbol detection can be expressed by the L2 norm minimization in the complex plane [Schulze and Lüders, 2005]:

$$\tilde{m} = \arg\min_{m} \sum_{k=0}^{K-1} \left| \tilde{C}_k - A \exp\left(j2\pi\varphi_{m,k}/4\right) \right|^2 . \tag{3.6}$$

However, the experiments in Section 3.3.1 indicated that compression by the selected LPC coders causes group delay in the processed signal and alters each harmonic with a distortion of different variance. The estimated phase shift $\hat{\phi}_k$ and the variance of distortion $\hat{\sigma}_k^2$ respective to each harmonic can be computed using a training sequence and the following estimators for sample mean and sample variance [Witte and Witte, 2017]:

$$\hat{\mu}_k = |\hat{\mu}_k| \exp\left(j\hat{\phi}_k\right) = \frac{1}{L} \sum_{\ell=0}^{L-1} \tilde{C}_{m_\ell,k} \exp(-j2\pi\varphi_{m_\ell,k}/4), \tag{3.7}$$

$$\hat{\sigma}_k^2 = \frac{1}{L-1} \sum_{\ell=0}^{L-1} \left| \tilde{C}_{m_\ell,k} \exp(-j2\pi\varphi_{m_\ell,k}/4) - \hat{\mu}_k \right|^2 , \tag{3.8}$$

where $\tilde{\mathbf{C}}_{m_\ell} = \{\tilde{C}_{m_\ell,k} \mid 0 \leq k < K\}$ denotes the $\ell - th$ sequence of PSK symbols measured at the reception side and $\Phi_{m_\ell} = \{2\pi\varphi_{m_\ell,k}/4 \mid 0 \leq k < K,\ \varphi_{m_\ell,k} \in \mathbb{Z}_4\}$ denotes the initial phases of the corresponding codebook symbols in the training sequence.

With the estimated $\hat{\phi}_k$ and $\hat{\sigma}_k^2$, one may apply the phase shift compensation and spectral weighting of distortion in the demodulation rule (3.6):

$$\tilde{m} = \arg\min_{m} \sum_{k=0}^{K-1} \left| \tilde{C}_k \exp(-j\hat{\phi}_k) - A \exp\left(j2\pi\varphi_{m,k}/4\right) \right|^2 / \hat{\sigma}_k^2. \tag{3.9}$$

Finally, rewriting (3.9) and removing the constant terms gives a more convenient demodulation rule, which is maximizing the real part of a complex dot product [Schulze and Lüders, 2005]:

$$\tilde{m} = \arg\max_{m} \Re\left(\sum_{k=0}^{K-1} \tilde{C}_k \cdot \frac{A}{\hat{\sigma}_k^2} \exp\left(-j2\pi\varphi_{m,k}/4 - j\hat{\phi}_k\right) \right). \tag{3.10}$$

In contrast to time-domain matched-filtering, the proposed demodulation rule enables phase and variance correction in the channel distortion. Secondly, it becomes more efficient when the codebook size grows. Instead of performing M matched-filtering operations on a symbol of length N, this demodulator needs to compute the in-phase/quadrature (I/Q) representations of $K < N$ PSK symbols and to correlate them with M different phase sequences. As an example, given the triple $(K, M, N) = (8, 256, 40)$, matched filtering in the time domain requires at least $256 \cdot 40 = 10240$ real-value multiplications. On the other hand, demodulation using Equation 3.10 involves computing the complex PSK symbols ($2 \cdot 8 \cdot 40 = 640$ real-value multiplications) and comparing the obtained sequence with all phase combinations in the codebook ($8 \cdot 256 = 2048$ complex multiplications, or at least 4096 real-value multiplications).

Despite the computational improvement, the codebook's preferable size ranges between 64 and 256 elements and should not overreach 4096 elements. These values would make the real-time demodulation computationally practical on portable devices, especially if the codebook has a symmetric structure that enables further computational optimizations.

Another factor in the process of selecting the codebook size is the transmission bitrate. Full 4PSK-OFDM modulation offers transmission up to $2K = \log_2(4^K)$ information bits per symbol. However, the modulation is susceptible to excessive distortion or attenuation of some harmonics in

spectrally selective voice channels. Instead, it is advisable to choose only a subset of all possible OFDM phase combinations to enlarge the minimum distance between symbols. This approach makes a transmission over voice channels more robust to spectrally selective distortion, as a large distortion of some harmonics would be compensated by a moderate distortion of the others. On the other hand, smaller modulation order $M < 4^K$ decreases the bitrate.

3.4.1 Codebook design

Construction of a suitable DoV codebook relies on finding (or training) a subset of harmonic symbols with a large minimum distance. However, this task becomes challenging as the number of symbol combinations increases. This subsection gives a proposition of a suboptimal codebook design method, which produces a set of harmonic waveforms sufficiently different from each other.

For $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{C}^K$, let $\mathrm{d}_E(\mathbf{x}_1, \mathbf{x}_2)$ be the Euclidean metric over the complex space and for $\mathbf{y}_1, \mathbf{y}_2 \in \mathbb{Z}_4^K$, let $\mathrm{d}_L(\mathbf{y}_1, \mathbf{y}_2)$ be the Lee metric over $\mathbb{Z}_4^K$:

$$\mathrm{d}_L(\mathbf{y}_1, \mathbf{y}_2) = \sum_{k=0}^{K-1} \min(|y_{1,k} - y_{2,k}|,\ 4 - |y_{1,k} - y_{2,k}|).$$

In addition, let us define the bijective function $f : \mathbb{C}^K \to \mathbb{Z}_4^K$ which takes the phase indices $\varphi_{m,k}$ of every 4-PSK sequence $\mathbf{C}_m = \{A \cdot \exp(j2\pi\varphi_{m,k}/4) \mid 0 \leq k < K,\ \varphi_{m,k} \in \mathbb{Z}_4\}$, and maps to a quaternary codeword $f(\mathbf{C}_m) = \{\varphi_{m,k} \mid 0 \leq k < K\}$ over $\mathbb{Z}_4^K$. For any two 4-PSK sequences $\mathbf{C}_{m_1}$ and $\mathbf{C}_{m_2}$, we get an isometric property:

$$2A^2 \mathrm{d}_L(f(\mathbf{C}_{m_1}), f(\mathbf{C}_{m_2})) = \mathrm{d}_E^2(\mathbf{C}_{m_1}, \mathbf{C}_{m_2}).$$

It can be noticed that the same relation holds for the minimum distance between all PSK sequences in the OFDM codebook and elements of the associated quaternary codewords. The selection of the most distinct OFDM symbols could be thus replaced by the construction of a quaternary code $\mathscr{C} \subset \mathbb{Z}_4^K$ (not necessarily a subgroup), that maximizes the minimum Lee distance.

In the perspective of non-binary codes with a defined minimum distance, these OFDM symbols can be seen as error correcting codes encoded in the spectral domain [Wilkinson and Jones, 1995]. In consequence, quaternary codes provide a new degree of freedom in the DoV codebook design. By some sensible manipulation of the number of harmonics K, the symbol duration N, and the minimum distance between codebook symbols d, it is possible to find a codebook providing the required bitrate and maintaining sufficient robustness to distortion. Moreover, the codebook generation is computationally constrained mostly by finding quaternary codes, which is a much faster process compared to training a full codebook of waveforms. Finally, quaternary codes can be reused to produce waveforms of different duration and harmonic frequencies. It is also worth noticing that the above motivation for exploiting non-binary codes is slightly different from other works focusing mainly on reducing the peak-to-mean energy ratio of the OFDM signal [Davis and Jedwab, 1999, Chen and Liang, 2007, Ginige et al., 2001, Hisojo et al., 2014].

Due to some rotational symmetries of quaternary codes, there is no unique codebook with a largest minimum distance. It gives more flexibility in the fine-tuning of the codes to make them more suitable in real operation. It is advisable to select a codebook with a possibly uniform distribution of phase values and remove symbols with the highest maximum amplitude. Table 3.1 presents the minimum distance of several quaternary codes found by a greedy algorithm coined

CodebookSearch. The subroutine ChooseInitial inserts a random or some pre-defined initial codeword into the codebook, while the subroutine SelectCodeword iteratively selects a codeword to remain within the uniform distribution of phase values in the expanded set.

To improve the computational demodulation efficiency, one may exploit the reflection symmetry of the codebook produced by the algorithm. Since for any $0 \leq 2m < M$ we have $\mathbf{s}_{2m} = -\mathbf{s}_{2m+1}$, it is sufficient to correlate the received PSK sequence only with codebook symbols having the even indices and then to check the sign of computation.

Table 3.1 – Minimum Lee distance of additive quaternary codes of length $n = 7, 8, 9$ and 10, found by Algorithm 1. Parameter k denotes the number of (quaternary) information bits of the code. From the perspective of OFDM symbols, value n is related to the cardinality of harmonics, while k describes the codebook size equal to 4^k.

$n \setminus k$	2.0	2.5	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5	7.0	7.5	8.0
7	6	6	4	4	3	3	2	2	2	1	1	-	-
8	8	8	6	6	4	4	4	4	2	2	2	2	1
9	8	8	6	6	5	4	4	4	3	2	2	2	1
10	10	9	7	6	6	5	5	4	4	3	3	2	2

Algorithm 1: CodebookSearch(C, M)

Data: the set of quaternary codewords C, an even size of codebook M;
Result: a set Cb of M quaternary codes;
Cb $\longleftarrow \varnothing$;
`// select the first codeword (random or pre-defined)`
$c_0 \longleftarrow$ ChooseInitial(C);
Cb $\longleftarrow$ Cb $\cup \{c_0, -c_0\}$;
for $i \leftarrow 1$ **to** $\lfloor M/2 \rfloor - 1$ **do**
  `// select codewords in C with`
  `// a maximum Lee distance from Cb`
  S $\longleftarrow$ MaxLeeDistance(C, Cb);
  `// select a codeword from S respective`
  `// to uniform distribution`
  $c_{2i} \longleftarrow$ ChooseCodeword(S, Cb);
  Cb $\longleftarrow$ Cb $\cup \{c_{2i}, -c_{2i}\}$;
end

3.5 Experiments

This section presents the performance results of the DoV scheme described in Section 3.4. Simulations are followed by experimental tests over 3G calls (based on AMR) and selected VoIP applications (Skype, WhatsApp, and Signal exploiting Opus-Silk and FaceTime using AAC-LD). Examples of some DoV signals recorded during tests are available online. [1]

1. https://github.com/PiotrKrasnowski/Data_over_Voice

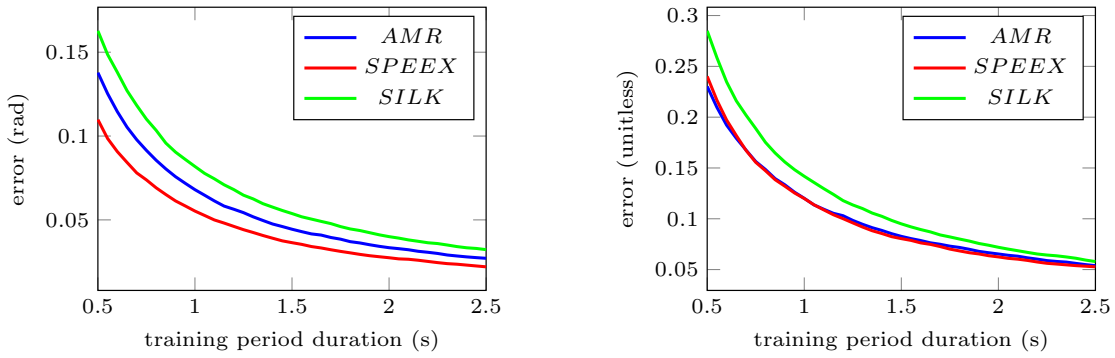
3.5.1 Channel estimation

Efficient detection of received DoV symbols, described by Equation 3.10 in Section 3.3, requires voice channel characterization using the training sequence. Intuitively, the larger number of symbols in the sequence, the more accurate is the estimation. We estimated the standard error SE of the phase shift $\hat{\phi}_k(t)$ and the variance of distortion $\hat{\sigma}_k^2(t)$ as a function of training duration t, using Monte Carlo simulations and the following formulas:

$$\widehat{\mathrm{SE}}^2_{\hat{\phi}_k(t)} = \frac{1}{L}\sum_{\ell=1}^{L}\left(\hat{\phi}_{k,\ell}(t) - \bar{\phi}_k\right)^2, \tag{3.11}$$

$$\widehat{\mathrm{SE}}^2_{\hat{\sigma}_k^2(t)/\bar{\sigma}_k^2} = \frac{1}{L}\sum_{\ell=1}^{L}\left(\hat{\sigma}_{k,\ell}^2(t) - \bar{\sigma}_k^2\right)^2 / \bar{\sigma}_k^2, \tag{3.12}$$

where L is the number of Monte Carlo runs, $\hat{\phi}_{k,\ell}(t)$ and $\hat{\sigma}^2_{k,\ell}(t)$ denote respectively the estimated phase shifts and the variances of distortion in the $\ell - th$ Monte Carlo run, and the reference values $\bar{\phi}_k$ and $\bar{\sigma}_k^2$ were obtained from a sequence of 50000 DoV symbols (250 seconds of a signal). Figure 3.10 depicts the maximum standard error of $\hat{\phi}_k(t)$ and $\hat{\sigma}_k^2(t)/\bar{\sigma}_k^2$ taken over all harmonics k and for every t between 0.5 and 2.5 seconds with a 0.05 second step. It can be observed that 2 seconds of training period should give a sufficiently accurate channel characterization.



(a) Standard error of phase-shift estimation.

(b) Standard error of variance estimation.

Figure 3.10 – Estimated standard error of the phase-shift $\hat{\phi}_k$ and the normalized variance $\hat{\sigma}_k^2/\bar{\sigma}_k^2$ estimators of distortion introduced by a selection of coders. The graphs present the maximum standard error over all harmonics k, and for every t between 0.5 and 2.5 seconds with a 0.05 second step. Results obtained based on 1000 Monte Carlo runs. The reference values $\bar{\phi}_k$ and $\bar{\sigma}_k^2$ were computed from a sample of 50000 symbols. The DoV signal consisted of 8 harmonics at frequencies 400 Hz, 800 Hz, ..., 3200 Hz with a modulation rate of 200 baud.

3.5.2 Simulations

The symbol error rate primarily depends on the distortion variance and the minimum distance between codebook symbols. For example, it can be noticed in Figure 3.11(a) that compressing by AMR leads to significantly lower error rates when compared to compression using the Silk

codec. This result agrees with the experimental outcomes shown in Figure 3.5 in Section 3.3. Nevertheless, when the voice channel's capacity goes up, the amount of distortion, and thus the error rate gradually decreases, as indicated by Figure 3.11(b).

The characteristic staircase shape of the graphs in Figures 3.11(a) and 3.11(b) corresponds to the codebook minimum distance d in function of the codebook size (ref. Table 3.1). Thus, the symbol error rates obtained can be viewed as the approximated probability of the signal distortion exceeding the distance $d/2$. Consequently, it is generally advantageous to design the codebook with a larger number of orthogonal harmonics, leading to increased minimum distance and improved robustness.

Despite its simplicity, the presented scheme suffers from the large size of the codebooks used, especially at higher bitrates. The exponentially growing number of correlations becomes a major practical limitation for real-time signal demodulation. The problem can be tackled by scaling down the symbol duration at the expense of higher relative distortion and a smaller number of orthogonal frequency slots. As shown by Figure 3.11(c), a modulation based on smaller codebooks of shorter symbols provides similar performance at a much lower computational cost.

3.5.3 Real-world tests

The DoV technique has been tested over a real voice channel between mobile phones, using pre-computed DoV signals. The selected phones for experiments were two iPhones 6 running iOS 12 and a Huawei P8 Lite running Android 8, each registered to a different major French mobile network operator. The DoV performance over 3G calls is displayed in Table 3.2, and the performance over VoIP calls using 4G wireless network is shown in Table 3.3. The duration of the training period was extended to 4 seconds to ensure the reliability of the experiments.

Table 3.2 – Symbol error rate of DoV signal over 3G call with and without channel estimation.

10 harmonics, symbol duration 5 ms		
bitrate	4 s training period	no training
1.0 kbps	$< 1.0 \cdot 10^{-4}$	$< 1.0 \cdot 10^{-4}$
1.2 kbps	$< 1.0 \cdot 10^{-4}$	$< 1.0 \cdot 10^{-4}$
1.4 kbps	$1.2 \cdot 10^{-4}$	$2.9 \cdot 10^{-4}$
1.6 kbps	$2.6 \cdot 10^{-4}$	$4.8 \cdot 10^{-4}$
1.8 kbps	$6.0 \cdot 10^{-4}$	$1.5 \cdot 10^{-3}$
2.0 kbps	$1.2 \cdot 10^{-3}$	$2.6 \cdot 10^{-3}$
2.2 kbps	$9.4 \cdot 10^{-3}$	$1.4 \cdot 10^{-2}$
2.4 kbps	$1.6 \cdot 10^{-2}$	$2.2 \cdot 10^{-2}$

8 harmonics, symbol duration 2.5 ms		
bitrate	4 s training period	no training
1.2 kbps	$< 1.0 \cdot 10^{-3}$	$< 1.0 \cdot 10^{-3}$
1.6 kbps	$< 1.0 \cdot 10^{-3}$	$1.3 \cdot 10^{-3}$
2.0 kbps	$1.2 \cdot 10^{-3}$	$3.5 \cdot 10^{-3}$
2.4 kbps	$1.3 \cdot 10^{-2}$	$2.6 \cdot 10^{-2}$
2.8 kbps	$3.4 \cdot 10^{-2}$	$6.6 \cdot 10^{-2}$
3.2 kbps	$1.0 \cdot 10^{-1}$	$1.3 \cdot 10^{-1}$
3.6 kbps	$1.2 \cdot 10^{-1}$	$1.9 \cdot 10^{-1}$
4.0 kbps	$2.0 \cdot 10^{-1}$	$2.6 \cdot 10^{-1}$

Table 3.3 – Symbol error rate of DoV signal over VoIP.

8 harmonics, symbol duration 2.5 ms, 4 s training period				
bitrate	Face Time	Skype	Signal Messenger	WhatsApp
4.0 kbps	$< 1.0 \cdot 10^{-4}$	$< 1.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-4}$	$9.6 \cdot 10^{-4}$
4.8 kbps	$< 1.0 \cdot 10^{-4}$	$1.0 \cdot 10^{-4}$	$9.3 \cdot 10^{-4}$	$5.1 \cdot 10^{-3}$
5.6 kbps	$< 1.0 \cdot 10^{-4}$	$1.2 \cdot 10^{-4}$	$4.4 \cdot 10^{-3}$	$1.7 \cdot 10^{-2}$
6.4 kbps	$6.7 \cdot 10^{-4}$	$3.0 \cdot 10^{-3}$	$6.2 \cdot 10^{-2}$	$8.6 \cdot 10^{-2}$

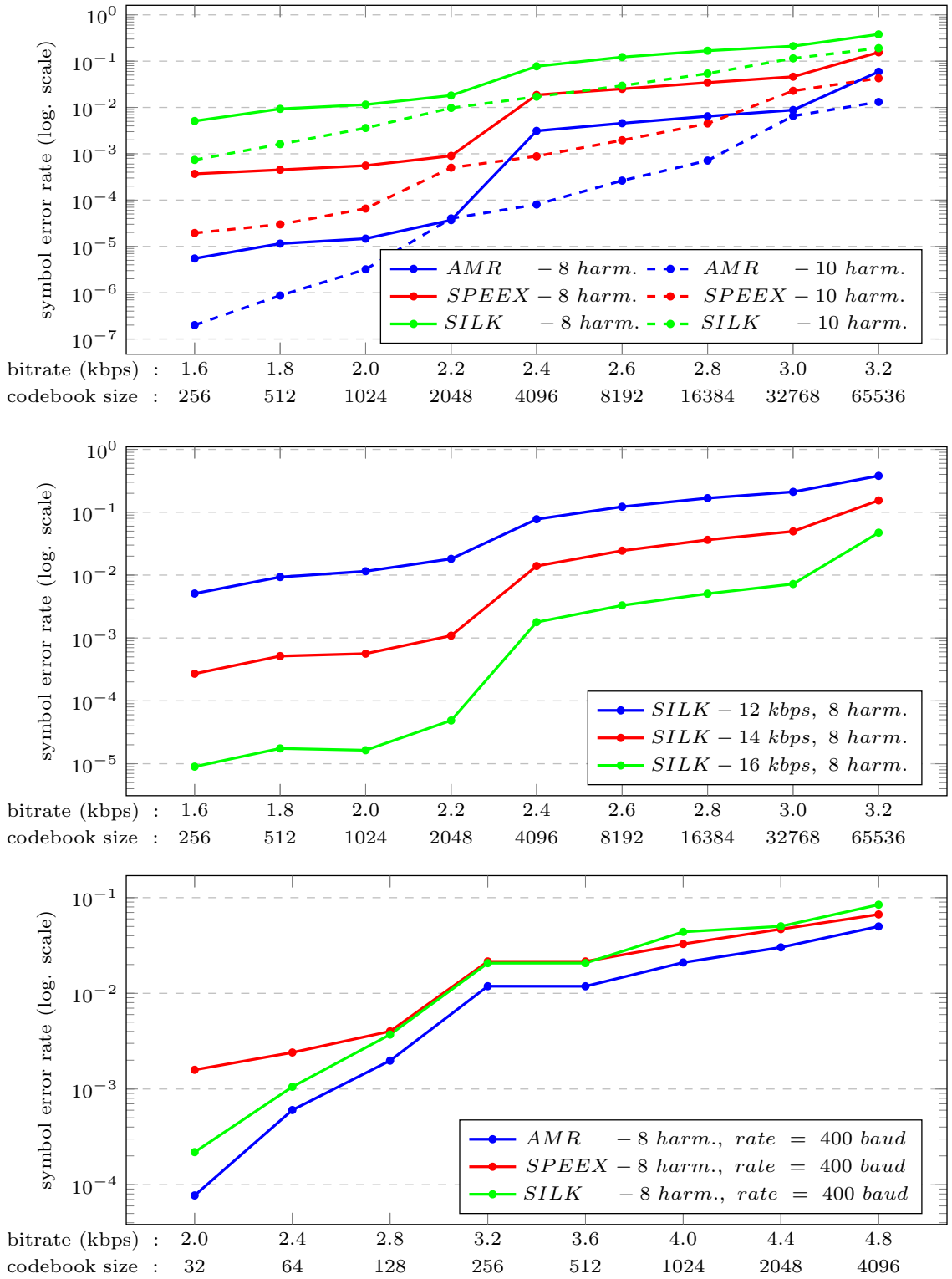


Figure 3.11 – Symbol error rate of a DoV signal compressed by AMR, Speex and Opus-Silk. To ensure reliability of the simulations, duration of the training period was extended to 4 seconds. If not indicated otherwise, symbol rate equals 200 baud. DoV signals consisted of 10^7 symbols produced according to an output of a built-in pseudo-random generator with a pre-defined seed.

In the case of the 3G connection, the overall symbol error rates given in Table 3.2 are higher compared to the simulation results presented in Figure 3.11. Additional signal distortion is possibly caused by several signal processing stages in the phones and also by multiple voice compression in the network [Katugampala et al., 2003]. Nevertheless, the DoV signal based on faster modulation and smaller codebook sizes again demonstrated lower error rates. Finally, the results emphasize the importance of voice channel estimation, which significantly improves the symbol error rate. Figure 3.12 displays the small fragment of the DoV signal sent over the 3G channel.

Contrary to 3G, VoIP enables very high DoV bitrates, up to full OFDM narrowband transmission at 6.4 kbps. The improved results provided in Table 3.3 are achieved due to mild signal distortion given by high throughput and network stability. However, since VoIP is a packet-based system without any guarantee of Quality of Service (QoS), short interruptions in the network connection may cause many packet dropouts. The negative impact of dropouts is typically mitigated by the re-synthesis of lost frames by VoIP application, leading to non-recoverable damages to the DoV signal and hindering the system's re-synchronization.

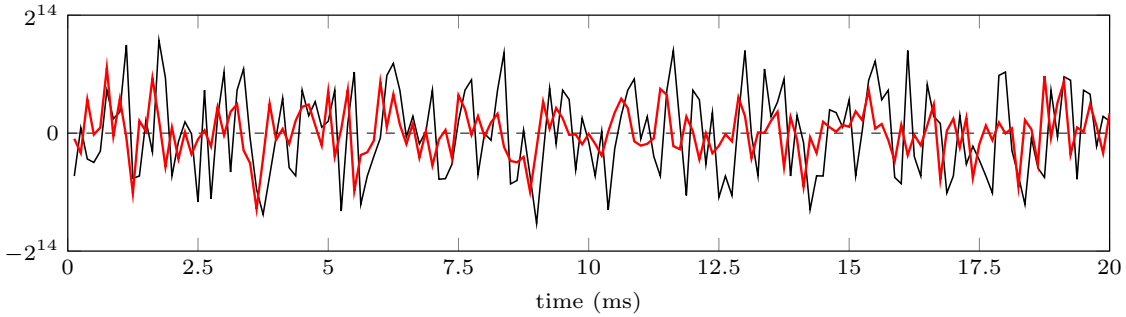


Figure 3.12 – DoV signal at the bitrate 2.8 kbps, before (black line) and after (red line) transmission over the 3G network. The fragment displays eight consecutive DoV symbols of duration 2.5 ms consisting of 8 harmonics at frequencies 400 Hz, 800 Hz, ..., 3200 Hz, with a 400 Hz step.

3.6 Secure voice communication

This section provides a detailed proposition of a scheme for secure voice communication over 3G and VoIP, using small portable devices with limited battery capacity. The system has been successfully tested in a controlled, real-world environment and with pre-computed DoV signals. The performance results are followed by a short discussion on security and computational complexity.

3.6.1 Communication system

Figure 3.13 presents a simplified diagram of a system for secure voice communication over a voice channel, which transforms consecutive portions of speech into DoV frames of the same duration. The scheme substantially resembles a classical digital communication system: it consists of speech encoding, followed by encryption, error correction, and data modulation blocks. Although the input and output signals of the processing chain are analog, all internal processing is performed digitally.

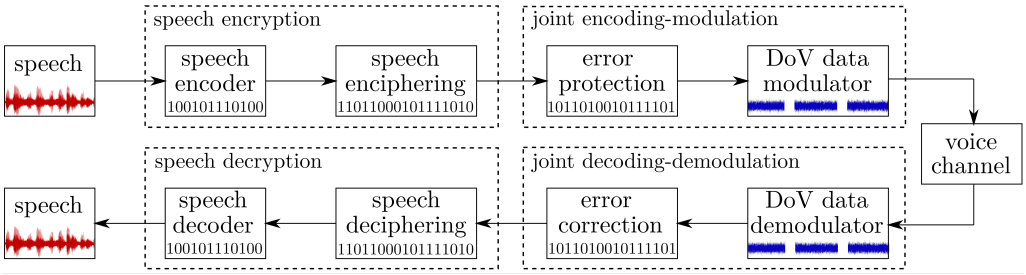


Figure 3.13 – Encrypted speech over voice channel scheme.

The system settings should be a trade-off between operational constraints (restricted bandwidth, real-time processing, synchronization) and the desired security level against eavesdroppers and active attackers from within the network. Depending on the voice channel type, two modes of operation may be considered: a low mode designed for 3G cellular calls and a high mode for VoIP. The system parameters selected in the following experiments are presented in Table 3.4 and are used only for illustration.

Table 3.4 – Selected parameters of the secure voice communication system.

version:	low mode (3G)	high mode (VoIP)
DoV frame		
codebook size:	64	4096
DoV symbol order:	6 bits	12 bits
modulation rate:	400 baud	400 baud
bitrate:	2400 bps	4800 bps
frame duration:	80 ms	60 ms
frame length:	32 symbols / 192 bits	24 symbols / 288 bits
Reed-Solomon coding		
RS symbol order:	6 bits	6 bits
message length:	20 symbols / 120 bits	28 symbols / 168 bits
· encrypted speech:	96 bits	144 bits
· frame counter:	16 bits	16 bits
· control checksum:	8 bits	8 bits
code length:	28 symbols / 168 bits	40 symbols / 240 bits
redundancy:	8 symbols / 48 bits	12 symbols / 72 bits
Voice enciphering	AES 256 (CTR mode)	AES 256 (CTR mode)
Voice compression	Codec2 1200 bps	Codec2 2400 bps

The processing chain starts with low-bitrate speech compression. In this work, voice is encoded by *Codec2*, an open-source algorithm developed by Rowe [2] and J.-M. Valin, which offers speech compression down to 450 bps [Erhardt et al., 2019]. In the next step, the encoded voice frames are enciphered by AES in the counter mode of operation and with a secret key of 256 bits with a random initial value (IV).

2. https://rowetel.com

The encrypted binary stream is protected against channel errors by shortened Reed-Solomon (RS) codes with erasures [Lin and Costello, 2001, Neubauer et al., 2007] and 6-bit symbols. The error correction capabilities of RS codes depend only on the redundancy length, which is not the case for Turbo and LDPC codes [Tahir et al., 2017]. Moreover, non-binary symbol processing of RS codewords seems suitable for symbol-to-symbol demodulation of the DoV signal. In particular, one or more RS symbols can be represented by a single DoV symbol.

Erasure decoding improves correction capabilities of RS codes, provided that the localization of errors are known. The demodulator may try to guess the erroneous symbols, using a straightforward metric that considers symbol energy and its distance to the closest codebook symbol. Thus, when the first decoding attempt fails, the decoder may reiterate decoding with new estimated erasure positions until the 8-bit control checksum (8-CRC) matches.

Table 3.5 – Performance of encrypted voice transmission over cellular voice channels and VoIP.

	3G	Face Time	Skype	Signal Messenger	WhatsApp
effective BER:	$3.7 \cdot 10^{-3}$	$< 1.0 \cdot 10^{-4}$	$< 1.0 \cdot 10^{-4}$	$< 1.0 \cdot 10^{-4}$	$7.8 \cdot 10^{-4}$
effective FER:	$1.9 \cdot 10^{-2}$	$< 1.0 \cdot 10^{-3}$	$< 1.0 \cdot 10^{-3}$	$< 1.0 \cdot 10^{-3}$	$2.1 \cdot 10^{-3}$

In the proposed scheme, each RS codeword is directly encoded into one DoV frame, as described in Figure 3.14. A constant header and a counter (CTR) enable decoding and decryption of DoV frames independently from each other, simplifying the re-synchronization in the presence of signal dropouts. Extensive experiments have shown that a 10-ms header is usually sufficiently long to keep signal synchronization or detect a DoV frame after signal restoration. In addition, the 16-bit counter permits re-synchronization after more than one hour of lost connection.

The duration of a DoV frame is equal to the portion of speech encoded by this frame, which is a valid requirement for real-time communication. Selected voice compression rates, 1.2 kbps, and 2.4 kbps depending on the mode, are low enough to append error correction redundancy at the end of each DoV frame.

The system was tested over cellular and VoIP calls. Table 3.5 presents the decoding results of several minutes of speech recording sent through using the 4G mobile data connectivity between two iPhones 6 registered to different network operators. The effective bit error rates (BER) and frame error rates (FER) take into account errors due to system de-synchronizations and dropouts.

Figure 3.15 shows the consecutive waveforms of a signal processed by a 3G network. The initial speech waveform presented in Figure 3.15(a) is compressed, encrypted, and encoded into the DoV signal of equal duration in Figure 3.15(b). The received signal displayed in Figure 3.15(c) is strongly attenuated after less than two seconds of transmission, classified by the Voice Activity Detector (VAD) as non-speech-like. However, correct decoding is still possible as long as the harmonic structure of the signal is preserved, as shown in Figure 3.15(d).

The distortion of the received signal varies depending on the network type and the phones used for communication. To counteract the blockage of stationary signals by VAD and Noise Suppression, several authors suggest to alternate two DoV codebooks defined over two non-overlapping bandwidths [Shahbazi et al., 2010, Sapozhnykov and Fienberg, 2012]. This work proposes another complementary technique: periodic silence insertion in place of some DoV frames, as depicted in Figure 3.16. It was observed that depending on the chosen rate of silence insertion and the type of connection, these silences significantly postpone or even prevent signal suppression. On the reception side, these inserted silences can be classified as lost frames and re-synthesized.

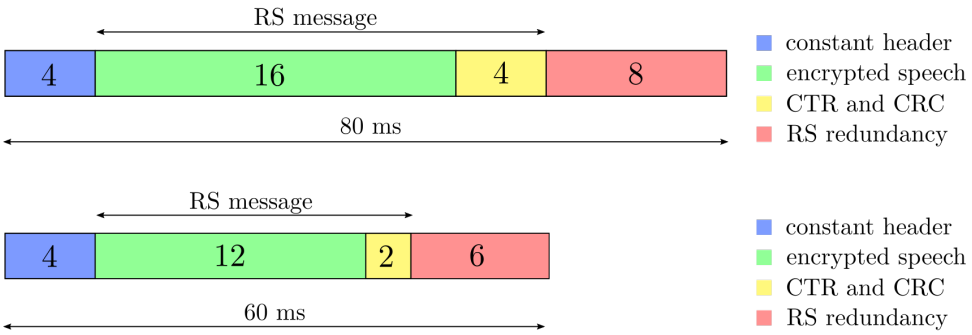


Figure 3.14 – DoV frame structure in a low (up) and a high (bottom) mode of operation. The numbers indicate the lengths of frame sections, given as a cardinality of DoV symbols. In the high mode, one DoV symbol represents two RS symbols.

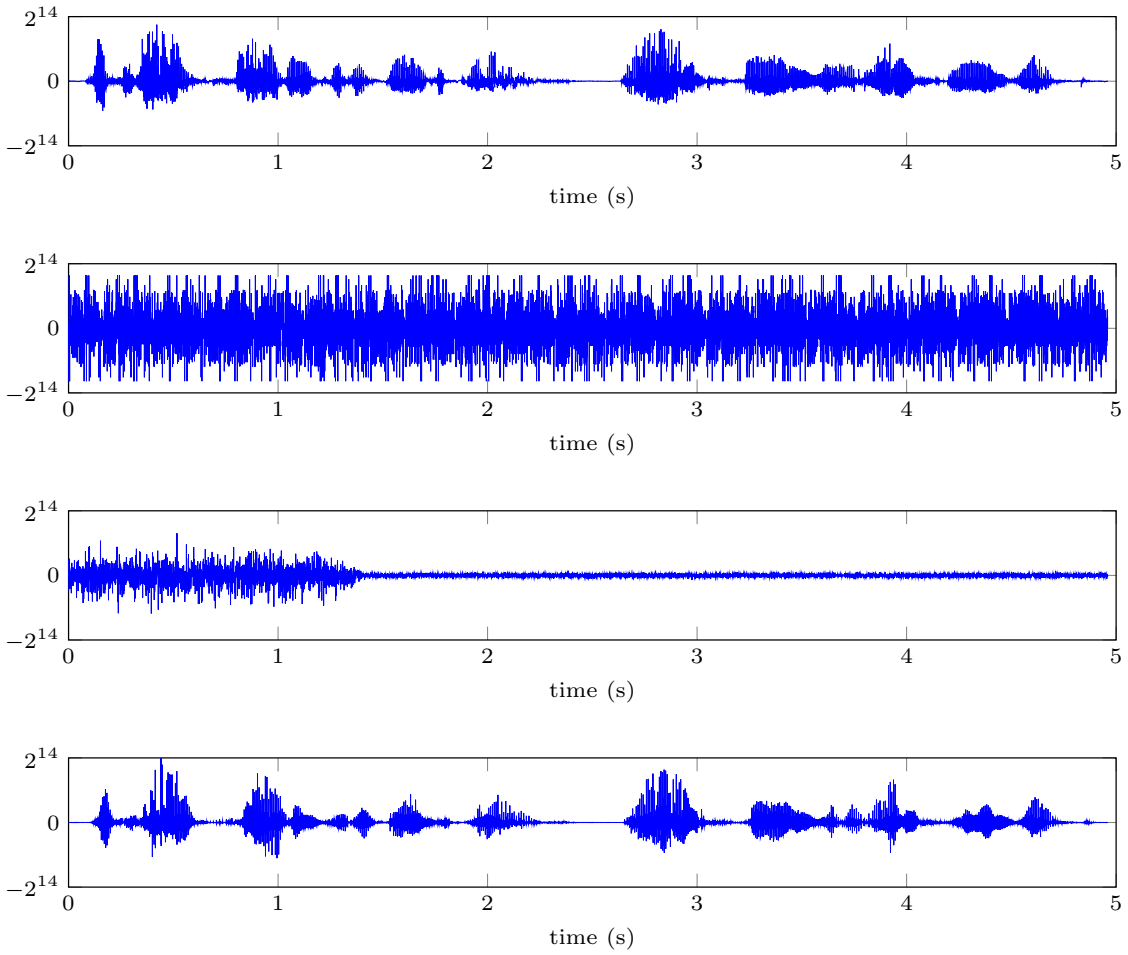


Figure 3.15 – Consecutive stages of the signal in secure voice communication over a 3G call. From top to bottom: the initial speech, the sent DoV signal, the received DoV signal and the re-synthesized speech. The received signal was fully decodable despite strong signal attenuation.

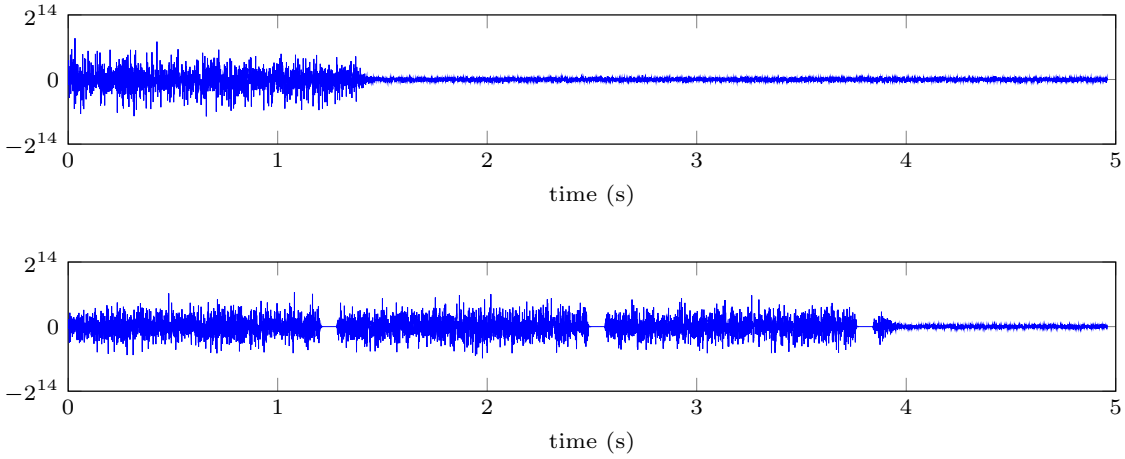


Figure 3.16 – Comparison of the received DoV signal in a 3G call (top) without and (bottom) with silence insertion every 16th frame. Depending on the connection type and the silence insertion rate, this technique may postpone or prevent signal suppression.

3.6.2 Security discussion

Introducing a dedicated system for voice communication is a response to an increased risk of being intercepted. Thus, a cryptographic scheme should reflect higher requirements for secrecy and authentication. A major risk is the recording and off-line cryptanalysis of the network traffic by passive eavesdroppers. Securing the communication against eavesdroppers is especially important because the encrypted and non-speech signal can be easily detected by some advanced Data Leakage Prevention (DLP) and Content Monitoring and Filtering (CMF) systems protecting against unauthorized data extrusion [Chae et al., 2015, Hauer, 2015, Lee et al., 2017]. Active attackers controlling the network are more likely to block or distort the fragile DoV signal, which is technically very simple. However, a malicious attacker who can synthesize a compatible DoV signal in real-time may modify the signal or insert its own.

The chosen AES cipher in the counter mode of operation, if implemented correctly, is believed to provide security against passive eavesdroppers [Lipmaa et al., 2000, Jonsson, 2003]. On the other hand, enciphering in counter mode does not guarantee data integrity [Katz and Lindell, 2015], giving some space for adversarial manipulations. Therefore, the common practice is to combine the AES in counter mode with a cryptographic message authentication function [Housley, 2004]. Unfortunately, due to severe bandwidth limitations appending the authentication check is not viable. Instead, it would be possible to randomly shuffle the positions of encrypted bits within one DoV frame [Morris et al., 2009, Stefanov and Shi, 2012]. The motivation for this is to prevent malicious attackers from intentional modifications of the transmitted content. While still capable of replacing several DoV symbols, the attacker should not benefit from distorting the transmitted signals.

Finally, it is assumed that both users share a common secret cryptographic key used for encryption. Secure key exchange can become challenging when the voice channel is the only available communication channel. With decentralized implementations of the proposed system, there would be no practical possibility to add the Trusted Third Party for user's authentication. A few protocols overcome this limitation by using vocal verification [Pasini and Vaudenay, 2006, Callas et al.,

2011, Krasnowski et al., 2020]. In such a scenario, users compare freshly generated random strings vocally while challenging another speaker's voice profile.

3.6.3 Computational complexity

The goal of real-time operation on small portable devices puts a big emphasis on computational optimization of the proposed system. It can be noticed that PSK-OFDM modulation [3GPP, 2020], AES-CTR encryption [Housley, 2004, Park et al., 2011], Reed-Solomon error correction [Biard and Noguet, 2008, 3GPP, 2017] and the speech encoding [Wisayataksin, 2019] algorithms mentioned in this work have been already widely adopted in wireless communication with mobile phones or in the computationally constrained environment, including real-time applications. However, the presented system was implemented in GNU Octave environment, [3] serving as a proof-of-concept only. There is still considerable work to be done to efficiently integrate all these elements into a single system operating on a device with limited resources, like mid-range smartphones.

3.7 Summary

This chapter has detailed a new and versatile Data over Voice technique for secure voice communications over LPC-based voice channels, like cellular networks and VoIP. Based on codebooks with harmonic symbols, the proposed solution is well-grounded on the fundamental principles of LPC coding.

A thorough analysis of OFDM signals compressed by some prominent voice coders revealed that the distortion is statistically close to a symmetric bivariate Gaussian distribution over the complex phase-amplitude plane. However, this distortion is not uniformly distributed in the spectral domain. Thus, we proposed an optimized demodulation metric based on spectrally weighted Euclidean distance with phase shift correction.

The tedious design process of DoV codebooks has been considerably simplified by using quaternary error correction codes. With OFDM symbols being treated as codes over a quaternary ring, codebook construction reduces to finding a set of quaternary codes that maximizes the minimum Lee distance.

The performance of our DoV technique has been evaluated through simulations and real-world tests over real voice connections between two mobile phones. A bitrate of 2.4 kbps over 3G call and 6.4 kbps over VoIP have been achieved with acceptably low symbol error rates. These tests highlight the need to properly characterize the channel distortion before transmission properly.

Finally, the work described a scheme for secure voice communications over voice channels in high and low bitrate modes of operation. The system has been practically validated for real-time voice transmission over cellular networks and VoIP with small effective bit error rates. To mitigate the negative impact of VAD, we also proposed a new method based on the insertion of repetitive silences.

The promising results presented in this work suggest some further investigation of the proposed DoV technique. A big emphasis has to be put on signal synchronization on the reception side and reducing the computational cost of signal demodulation. Additionally, sensible codebook structuring, combined with the exploitation of phase symmetries, may significantly lower the number of correlations in a demodulator.

3. https://www.gnu.org/software/octave/

Chapter 4 introduces a new enciphering scheme adapted to communication over channels with errors. In contrast to the DoV technique described in this section, the scheme is tolerant to transmission error. The method enciphers spherical data using random rotations from a commutative group of orthogonal matrices and is particularly suitable for transmitting perceptually-oriented data, such as multidimensional speech timbre.

CHAPTER 4

Distortion-tolerant encryption of vectors on N-spheres

This chapter presents a distortion-tolerant encryption scheme for scrambling unit vectors on hyperspheres, using rotations from a group of orthogonal matrices, and a non-binary pseudo-random number generator (PRNG) with a fresh secret seed. The method gives indistinguishable encryptions in the presence of an eavesdropper and is robust against channel errors. This makes the encryption scheme suitable in secure voice communications over voice channels, for example by scrambling a multi-dimensional vocal timbre representation.

The technique has been successfully tested on a toy example. In the test, we scrambled the colors in an image by performing pseudo-random rotations, and later corrupted the enciphered data with Gaussian noise. The descrambled image exhibited a gracefully progressive quality degradation depending on the distortion intensity, while the informative content of the image remained well preserved.

Glossary

List of abbreviations

CVP	Closest Vector Problem
PPT	Probabilistic Polynomial Time
PRNG	Pseudo-Random Number Generator
RMSE	Root Mean Squared Error
SNR	Signal-to-Noise Ratio

Notation - lattices and spherical codes

$\Lambda, \Lambda_\alpha, \Lambda_\beta$	lattices
S^n	unit sphere in $\mathbb{R}^{n+1}$ centered at the origin
$\mathscr{C}$	spherical commutative group code
$\boldsymbol{\sigma}$	initial codeword of the code $\mathscr{C}$
$\mathcal{G}$	commutative group of orthogonal matrices
G_i	orthogonal matrix i from $\mathcal{G}$
$\mathrm{T}_{\boldsymbol{\xi}}$	flat torus in $\mathbb{R}^{2n}$ associated with a positive unit vector $\boldsymbol{\xi} \in \mathbb{R}^n$
$\Phi_{\boldsymbol{\xi}}(\bullet)$	torus mapping $\mathbb{R}^n \to \mathbb{R}^{2n}$ associated with $\boldsymbol{\xi} \in \mathbb{R}^n$ (see Section 4.3.2)
$\gamma_n(\mathbf{x})$	function $S^n \to \mathbb{R}^n$ which outputs angular coordinates of $\mathbf{x}$

Notation - encryption scheme

λ	security parameter
s	secret seed
$\mathbf{r}$	pseudo-random sequence
$\mathbf{x}, \mathbf{y}$	initial and decoded vectors on S^n
X, Y	sequences of initial and decoded vectors on S^n
$\mathbf{p}, \mathbf{u}$	encoded and enciphered codewords in $\mathscr{C}$
P, U	sequences of encoded and enciphered codewords in $\mathscr{C}$
$\mathbf{v}, \mathbf{q}$	received and deciphered vectors on the flat torus $\mathrm{T}_{\boldsymbol{\xi}}$
V, Q	sequences of received and deciphered vectors on $\mathrm{T}_{\boldsymbol{\xi}}$

Packing and covering radii of some selected lattices

lattice	packing density	covering density
cubic lattice Z^n	0.5	$\sqrt{n}/2$
checkerboard lattice D_n	$1/\sqrt{2}$	1 ($n = 3$) or $\sqrt{n}/2$ ($n > 3$)
Gosset lattice Γ_8	$1/\sqrt{2}$	1

4.1 Motivation

The investigation on speech coding and voice channels carried in Chapter 2 and Chapter 3 revealed that achieving error-less data transmission over real voice channels was very unlikely. This pessimistic outcome undermines the usefulness of many prominent cryptographic algorithms in speech encryption. The reason is their non-compliance to transmission error that prevents adversarial data manipulation and guarantees exact message decryption. In contrast, successful operation of voice channels equipped with speech coders proves that a good enough approximation of vocal parameters is sufficient to reconstruct intelligible speech. Consequently, some imperfect data decryption in secure voice communication is acceptable if the lower decryption accuracy is somehow compensated by the higher robustness against errors.

Designing a secure cryptographic scheme that operates correctly despite encrypted data distortion could be achieved using unconventional enciphering techniques. The desired property of such a scheme would be some kind of resilience of ciphertexts that would increase the robustness to distortion without compromising the secrecy of encrypted data. A similar problem confronted providers of cloud-based solutions who want to extract useful statistical information from the stored data without violating data privacy. The proposed remedies were new encrypting techniques, such as homomorphic [Gentry, 2009], order-preserving [Agrawal et al., 2004] and distance-preserving [Tex et al., 2018] encryptions.

Intuitively speaking, an encryption scheme is distance-preserving when the distance between any two pieces of encrypted data is the same after decryption. The technique has been applied in distance-based clustering on encrypted data [Yin et al., 2018] and distance calculation on spherical geophysical coordinates [Šeděnka and Gasti, 2014, Zhou et al., 2018]. However, the distance-preserving property is also useful for protecting audio media streams in real-time applications. For example, when the enciphered vocal parameters are degraded by some small channel error, the original signal could still be approximately decrypted without disrupting the communication.

Applying distance-preserving encryption technique directly on speech parameters is far from being straightforward. Naively, we could scramble three perceptual speech components: pitch, loudness, and timbre. Pitch and loudness are associated with fundamental frequency and signal energy, both of them scalars. However, timbre is a multi-dimensional signal loosely related to spectral envelope, and with many possible representations. For instance, in the speech recognition domain a spectral envelope is usually encoded by 13-19 Mel-Frequency Cepstral Coefficients (MFCC) [Davis and Mermelstein, 1980, Benesty et al., 2008, Chap. 9, Rabiner and Schafer, 2011, Chap. 8].

In this chapter, we propose an encryption scheme that is robust against channel errors, decrypts approximately, and is suitable for enciphering the spectral envelopes of speech signals (how it is done will be detailed in Chapter 5). The technique scrambles unit vectors on a hypersphere using a secure Pseudo-Random Number Generator (PRNG), a commutative (abelian) group of orthogonal matrices, and a spherical group code of equal footing. Despite the fact that spherical group codes where known at least since the late 60s from the seminal work of Slepian [Slepian, 1968], useful literature on using these codes for enciphering remains scarce. The encryption scheme is thoroughly detailed, with an emphasis on the security aspects and robustness to channel error. The scheme is proven to produce indistinguishable encryptions in the presence of an eavesdropper when the enciphering algorithm takes as input a secure pseudo-random sequence. Furthermore, the chapter introduces an extension of the notion of distance-preserving encryption, named distortion-tolerant

encryption, that is better suited for describing the robustness of encryption schemes against channel errors.

This chapter is organized as follows. Section 4.2 reviews some basic lattice theory and details some important lattices. Section 4.3 describes spherical commutative group codes and shows the explicit construction of spherical codes from pairs of nested lattices, as described in [Costa et al., 2017]. Section 4.4 recalls the fundamental definitions of indistinguishability of encryptions in the presence of an eavesdropper and of non-binary pseudo-random number generators. Section 4.5 details the notion of distortion-tolerant encryption and briefly describes its fundamental properties. Section 4.6 presents the distortion-tolerant encryption scheme. Section 4.7 provides for illustration a toy example of distortion-tolerant color scrambling in an image. Section 4.8 summarizes this work.

4.2 Lattices and lattice packings

This section recalls some basic theory related to lattices and describes some special lattices of high density and with efficient decoding algorithms. The interested reader may get more details from [Conway and Sloane, 1999].

4.2.1 Lattices and lattice packings

A lattice in $\mathbb{R}^n$ is a subgroup of the additive group $\mathbb{R}^n$ that is defined by all integer linear combinations of some independent vectors.

Definition 4.2.1. Let $\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_m$ be linearly independent vectors in $\mathbb{R}^n$. A lattice Λ with basis $\{\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_m\}$ is defined as the set of all linear combinations of the basis vectors:

$$\Lambda = \{u_1\mathbf{b}_1 + u_2\mathbf{b}_2 + ... + u_m\mathbf{b}_m : u_1, ..., u_m \in \mathbb{Z}\}. \tag{4.1}$$

A lattice may equivalently be viewed as the set of vector endpoints determined by linear combinations of the basis vectors. The number m of independent vectors in the basis is the rank of Λ. If $m = n$, the lattice Λ is called full rank. The basis vectors form a *generator matrix* $B = [\mathbf{b}_1, \mathbf{b}_2, ..., \mathbf{b}_m]$.

For any lattice Λ, there is an infinite family of bases and their associated generator matrices. Theorem 4.2.1 describes the relation between all generator matrices spanning Λ.

Theorem 4.2.1. *[Costa et al., 2017, Chap. 2] Two matrices B and C generate the same lattice Λ if and only if there exists a unimodular matrix U (a matrix with integer entries and whose determinant is either 1 or -1) such that $C = BU$.*

A subset $\Lambda' \subset \Lambda$ is said to be a *sublattice* of the lattice Λ if and only if Λ' is an additive subgroup of Λ. Let B and B' be the respective generator matrices of Λ and Λ'. If Λ' is full-rank, there exists a square integer matrix H with a non-zero determinant such that $B' = BH$. Moreover, the matrix H can be decomposed into its Smith normal form $H = PDQ$ where P and Q are unimodular matrices, and D is a diagonal matrix such that $\mathrm{diag}(D) = [d_1, d_2, ..., d_n]^T$, $d_i \in \mathbb{N}$ and d_i divides d_{i+1} for $i = 1, ..., n-1$ [Cohen, 1993, Sec. 2.4].

Since P and Q^{-1} are unimodular as in Theorem 4.2.1, the matrix $C := BP = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n]$ is a generator of Λ and $C' := B'Q^{-1} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n]$ is a generator of Λ'. We obtain $C' = CD$,

meaning that the new basis vectors of Λ' are integer multiples of the new basis vectors of the lattice Λ: $\mathbf{w}_i = d_i \mathbf{v}_i$ for $i = 1, ..., n$. These new basis representations show that the quotient group Λ/Λ' is isomorphic to $\mathbb{Z}_{d_{k_0}} \oplus ... \oplus \mathbb{Z}_{d_n}$, where d_{k_0} is the first element of $\mathrm{diag}(D)$ larger than 1 [Lavor and Gomes, 2018, p. 108-109].

Every full-rank lattice Λ can be characterized by its volume $\mathrm{V}(\Lambda)$, which is the volume of the fundamental parallelotope determined by the set of neighbor lattice points. The lattice volume is independent of the selected generator matrix B and is equal to:

$$\mathrm{V}(\Lambda) = \sqrt{\det(B^T B)} = |\det(B)|. \tag{4.2}$$

The space containing the lattice points can be decomposed into *Voronoi cells* [Okabe et al., 2000]. Let $\| \bullet \|$ denote some metric in $\mathbb{R}^n$. The Voronoi region associated with a lattice point $\mathbf{x} \in \Lambda$ is the set of points in $\mathbb{R}^n$ closer to $\mathbf{x}$ than to any other point of lattice Λ with respect to the chosen metric:

$$\mathcal{V}_\Lambda(\mathbf{x}) = \{\mathbf{y} \in \mathbb{R}^n : \|\mathbf{x} - \mathbf{y}\| \leq \|\mathbf{z} - \mathbf{y}\|, \text{ for all } \mathbf{z} \in \Lambda\}. \tag{4.3}$$

If $\| \bullet \|$ is the Euclidean distance measure, we have the volume of the Voronoi cell associated with $\mathbf{0}$ equal to $\mathrm{V}(\Lambda)$: $\mathrm{vol}(\mathcal{V}_\Lambda(\mathbf{0})) = \mathrm{V}(\Lambda)$ [Costa et al., 2017, Chap. 2].

Many practical problems in digital coding, computational geometry, and optimization often summarize to finding a dense sphere packing in a multidimensional Euclidean space, which can be intuitively understood as distributing some balls of a fixed radius. Some arrangements of balls can be called 'densest' in the space $\mathbb{R}^n$ when there exists no other (non-equivalent) arrangement of balls that occupies a larger portion of that space.

If an arrangement of balls is regular, the center points of the balls form a lattice. Furthermore, the density of the arrangement is independent of the chosen lattice basis, making it a useful measure for comparing the densities of different lattices.

Definition 4.2.2. Let $\mathbf{x}_{min}$ denote the shortest non-zero vector in Λ and let $\mathcal{B}^n(r)$ denote the n-dimensional ball of radius r around the origin: $\mathcal{B}^n(r) = \{\mathbf{x} \in \mathbb{R}^n : \|\mathbf{x}\| \leq r\}$. The packing density of Λ is defined as:

$$\Delta(\Lambda) = \frac{\mathrm{vol}(\mathcal{B}^n(\rho))}{\mathrm{V}(\Lambda)}, \tag{4.4}$$

where $\rho = \|\mathbf{x}_{min}\|/2$ is called the packing radius.

The dual problem to sphere packing is the sphere covering problem, linked to the covering density of Λ. As opposed to the previous measure, the covering density describes the smallest volume of balls covering the whole space $\mathbb{R}^n$, in relation to $\mathrm{V}(\Lambda)$. Because the balls may overlap, the covering density is never smaller than 1.

Definition 4.2.3. The covering density of Λ is defined as:

$$\Theta(\Lambda) = \frac{\mathrm{vol}(\mathcal{B}^n(\mu))}{\mathrm{V}(\Lambda)}, \tag{4.5}$$

where the covering radius μ is the smallest positive value such that the union of all the translates of the ball $\mathcal{B}^n(\mu)$ by vectors in Λ cover $\mathbb{R}^n$:

$$\bigcup_{\mathbf{x} \in \Lambda} (\mathcal{B}^n(\mu) + \mathbf{x}) = \mathbb{R}^n. \tag{4.6}$$

In the rest of this chapter, if not stated otherwise, $\| \bullet \|$ will denote the Euclidean norm.

4.2.2 Special lattices and finding the closest lattice points

Some lattices are characterized by some special structures and high densities which are of interest in this work. In particular, one of the desired properties is the simplicity of finding the closest lattice point $\mathbf{v}$ to any $\mathbf{z} \in \mathbb{R}^n$:

$$\mathbf{v} = \min_{\mathbf{w} \in \Lambda} \|\mathbf{z} - \mathbf{w}\| \,. \tag{4.7}$$

Given a random lattice and a target vector, the closest vector problem (CVP) is proved to be NP-hard [Micciancio and Goldwasser, 2012]. For certain classes of lattices, however, there exist very efficient ways of finding the closest or approximately closest vectors.

The cubic lattice Z^n is the lattice with all points being n-tuples of integers. It has the packing radius $\rho = 0.5$ and the covering radius $\mu = \sqrt{n}/2$. The regular structure of Z^n enables a very efficient procedure for finding closest lattice points. Let $\mathrm{g}(z)$ be the function which rounds $z \in \mathbb{R}$ to one closest integer. If the result is not unique, $\mathrm{g}(z)$ outputs the integer with the smallest absolute value. Then, for any vector $\mathbf{z} = [z_1, ..., z_n]^T$, the closest vector in Z^n is $\mathrm{g}(\mathbf{z}) = [\mathrm{g}(z_1), ..., \mathrm{g}(z_n)]^T$ [Conway and Sloane, 1999, Chap. 4].

The checkerboard lattice D_n is the lattice with elements having integer coordinates summing up to an even number:

$$\mathrm{D}_n = \{[z_1, ..., z_n]^T \in \mathbb{Z}^n \;:\; z_1 + ... + z_n \equiv 0 \mod 2\},\; n \geq 3 \,. \tag{4.8}$$

The packing radius of D_n is $\rho = 1/\sqrt{2}$, and the covering radius is $\mu = \rho\sqrt{2}$ $(n = 3)$ or $\mu = \rho\sqrt{n/2}$ $(n > 3)$. The lattice D_n is the densest lattice in dimensions $n = 3$, 4 and 5.

Due to the explicit relation between point coordinates in D_n, CVP in the Euclidean metric is nearly as simple as in the cubic lattice Z^n [Conway and Sloane, 1982]. Let $\mathrm{h}(\mathbf{z})$ be defined similarly to the function $\mathrm{g}(\mathbf{z})$ with the only difference that the component of $\mathbf{z}$ farthest from the integer is rounded the other way (for example, $\mathrm{h}([0.2, 1.7]) = [0, 1]$). Clearly, one of the points $\mathrm{g}(\mathbf{z})$ or $\mathrm{h}(\mathbf{z})$ has an even sum of coordinates. Thus, a point belonging to D_n is also the closest lattice point to $\mathbf{z}$.

The Gosset lattice Γ_8 was recently proven to be (up to an isomorphism) the densest lattice in dimension 8 [Viazovska, 2017]. It is defined as:

$$\Gamma_8 = \left\{ \mathbf{z} = [z_1, ..., z_8]^T \in \mathbb{Z}^8 \cup (\mathbb{Z} + 1/2)^8 : \sum_{i=1}^{8} z_i \equiv 0 \mod 2 \right\} , \tag{4.9}$$

and its generator matrix is given by:

$$B = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 \end{bmatrix} .$$

The Gosset lattice has packing density $\rho = 1/\sqrt{2}$ and covering density $\mu = 1$. It can be noticed that $\Gamma_8 = \mathrm{D}_8 \cup (\mathrm{D}_8 + \frac{1}{2})$, where $(\mathrm{D}_8 + \frac{1}{2})$ denotes the lattice D_n translated by $[\frac{1}{2}, \frac{1}{2}]^T$. This observation is useful in CVP: for any $\mathbf{z} \in \mathbb{R}^8$, the closest vector of Γ_8 is one of the four vectors $\mathrm{g}(\mathbf{z})$, $\mathrm{h}(\mathbf{z})$, $\mathrm{g}(\mathbf{z} - \frac{1}{2}) + \frac{1}{2}$, $\mathrm{h}(\mathbf{z} - \frac{1}{2}) + \frac{1}{2}$ [Conway and Sloane, 1982].

4.3 Spherical commutative group codes from lattices

In the digital communications domain, spherical codes are used in source and channel coding [Adoul et al., 1984, Hamkins and Zeger, 2002], coded signal modulation [Burr, 1989] and MIMO space-time coding [Utkovski and Lindner, 2006]. However, the available literature on exploiting spherical codes for data scrambling is very scarce if existing. This section describes some *spherical commutative group codes* associated with dense lattices that will be the building block of the presented enciphering scheme. The section starts by recalling the basic definitions and some theorems related to spherical group codes. The theory linking spherical group codes and lattices owes a lot to the work of Costa and his research group [Costa et al., 2017].

4.3.1 Spherical commutative group codes

A spherical commutative (abelian) group code on the unit hypersphere $S^{n-1} \subset \mathbb{R}^n$ is a set of unit vectors closed under matrix multiplication from some orthogonal matrix group $\mathcal{G}$. These spherical commutative group codes were introduced in [Slepian, 1968] as a new encoding method for sending information over the Gaussian channel.

Definition 4.3.1. [Slepian, 1968] A spherical commutative group code $\mathscr{C}$ of order M is a set of M unit vectors $\mathscr{C} = \{G\boldsymbol{\sigma} : \ G \in \mathcal{G}\}$, where $\boldsymbol{\sigma}$ lies on the unit hypersphere $S^{n-1} \subset \mathbb{R}^n$ and $\mathcal{G}$ is a finite group of order M of $n \times n$ orthogonal matrices.

Commutative spherical group codes are geometrically uniform, i.e., for any $\mathbf{u}, \mathbf{v} \in \mathscr{C}$, there exists an isometry $\mathrm{f}_{\mathbf{u},\mathbf{v}}$ such that $\mathrm{f}_{\mathbf{u},\mathbf{v}}(\mathbf{u}) = \mathbf{v}$ and $\mathrm{f}_{\mathbf{u},\mathbf{v}}(\mathscr{C}) = \mathscr{C}$ [Forney, 1991]. Moreover, they have congruent Voronoi regions, the same detection probability in the presence of transmission noise, and a distribution of codewords invariant to multiplication by matrices from $\mathcal{G}$. Although spherical commutative group codes do not offer packing densities as high as general spherical codes, this shortcoming is compensated by their simple structure and the easiness of encoding and decoding [Costa et al., 2017, p. 81]. Examples 4.3.1 and 4.3.2 in Section 4.3.3 show two instances of spherical codes.

Every element in $\mathcal{G}$ can be uniquely represented as a product of powers of generator matrices $\{G_1, ..., G_k\}$, such that $G_i \in \mathcal{G}$ for $i = 1, ..., k$, and G_i generate $\mathcal{G}$:

$$\mathcal{G} = \{G_1^{w_1} \cdot G_2^{w_2} \cdot ... \cdot G_k^{w_k} : 0 \leq w_i \leq d_i - 1, \ i = 1, ..., k\} .$$

Furthermore, $\mathcal{G}$ is isomorphic to $\mathbb{Z}_{d_1} \oplus ... \oplus \mathbb{Z}_{d_k}$ where $d_1 \cdot d_2 \cdot ... \cdot d_k = M$ and d_i divides d_{i+1} for $i = 1, ..., k-1$ [Cohen, 1993, Sec. 2.4]. We can thus conveniently index $G \in \mathcal{G}$ (and $G\boldsymbol{\sigma} \in \mathscr{C}$) by a vector $[w_1, ..., w_k]^T \in \mathbb{Z}_{d_1} \oplus ... \oplus \mathbb{Z}_{d_k}$.

The elements of $\mathcal{G}$ can also be viewed as a composition of rotations and reflections mapping points on the hypersphere into other codewords in $\mathscr{C}$. It is the consequence of Theorem 4.3.1 which states that we can orthogonally transform $\mathcal{G}$ into a new group with rotation and reflection parts separated.

Theorem 4.3.1. *[Gantmakher, 1959, p. 292] In every finite group $\mathcal{G}$ of $n \times n$ orthogonal matrices, every element $G_i \in \mathcal{G}$ for $i = 1, ..., M$ can be mapped by one and only one real orthogonal transformation Q into a block-diagonal form:*

$$QG_iQ^T = \left[\begin{array}{ccc|ccc} \mathrm{Rot}\left(\frac{2\pi a_{i,1}}{M}\right) & 0 & ... & & ... & 0 \\ 0 & \ddots & & & & \vdots \\ \vdots & & \mathrm{Rot}\left(\frac{2\pi a_{i,q}}{M}\right) & & & \\ \hline & & & b_{i,2q+1} & & \vdots \\ \vdots & & & & \ddots & 0 \\ 0 & ... & & ... & 0 & b_{i,n} \end{array}\right]$$

where $a_{i,j} \in \{0, ..., M\}$ for $j = 1, ..., q$ and $b_{i,j} = \pm 1$ for $j = (2q+1), ..., n$ and $\mathrm{Rot}(x)$ are 2×2 rotation matrices:

$$\mathrm{Rot}(x) = \begin{bmatrix} \cos(x) & -\sin(x) \\ \sin(x) & \cos(x) \end{bmatrix}.$$

We say, that $\mathcal{G}$ is free from reflection blocks, when $2q = n$ [Costa et al., 2017, p. 79]. Such a group of matrices can be transformed into a group that contains only spherical rotations.

4.3.2 Torus mapping

A flat torus mapping makes the link between spherical group codes on the unit sphere in $\mathbb{R}^{2n}$ and lattices in $\mathbb{R}^n$. This connection is used to construct spherical codes from dense lattices with a simple structure.

For each unit vector $[\xi_1, ..., \xi_n]^T = \boldsymbol{\xi} \in S^{n-1}$ with positive coordinates $\xi_i > 0$, and for every $[u_1, ..., u_n]^T = \mathbf{u} \in \mathbb{R}^n$, let the mapping $\Phi_{\boldsymbol{\xi}} : \mathbb{R}^n \to \mathbb{R}^{2n}$ be defined as:

$$\Phi_{\boldsymbol{\xi}}(\mathbf{u}) = [\xi_1 \cos(u_1/\xi_1), \xi_1 \sin(u_1/\xi_1), ..., \xi_n \cos(u_n/\xi_n), \xi_n \sin(u_n/\xi_n)]^T. \tag{4.10}$$

The image of $\Phi_{\boldsymbol{\xi}}$ describes a flat torus $\mathrm{T}_{\boldsymbol{\xi}}$ contained on the surface of a unit hypersphere $S^{2n-1} \subset \mathbb{R}^{2n}$. Moreover, the same flat torus can be injectively mapped to a bounded box in $\mathbb{R}^n$, as depicted in Figure 4.1:

$$\mathcal{P}_{\boldsymbol{\xi}} = \{\mathbf{u} \in \mathbb{R}^n : 0 \leq u_i \leq 2\pi\xi_i,\ 1 \leq i \leq n\}. \tag{4.11}$$

The whole family of flat tori with $\|\boldsymbol{\xi}\| = 1$ and $\xi_i \geq 0$ foliates the hypersphere S^{2n-1}, meaning that every point on the sphere belongs to one and only one flat torus [Lawson Jr, 1974, Candel and Conlon, 2000].

The Euclidean distance between two points on the flat torus associated with the vector $\boldsymbol{\xi} = [\xi_1, ..., \xi_n]^T$ is given by [Torezzan et al., 2013]:

$$\|\Phi_{\boldsymbol{\xi}}(\mathbf{u}) - \Phi_{\boldsymbol{\xi}}(\mathbf{v})\| = 2\sqrt{\sum_{i=1}^{n} \xi_i^2 \sin^2\left(\frac{u_i - v_i}{2\xi_i}\right)}, \tag{4.12}$$

and is bounded by [Torezzan et al., 2015]:

$$\frac{2}{\pi}\|\mathbf{u}-\mathbf{v}\| \leq 2\xi_{min} \sin\left(\frac{\|\mathbf{u}-\mathbf{v}\|}{2\xi_{min}}\right) \leq \|\Phi_{\boldsymbol{\xi}}(\mathbf{u})-\Phi_{\boldsymbol{\xi}}(\mathbf{v})\| \leq 2\sin\left(\frac{\|\mathbf{u}-\mathbf{v}\|}{2}\right) \leq \|\mathbf{u}-\mathbf{v}\|, \tag{4.13}$$

where $\xi_{min} = \min_{1 \leq i \leq n} \xi_i \neq 0$ and assuming $\|\mathbf{u} - \mathbf{v}\| \leq \xi_{min}$. It may be noticed that the distance $\|\Phi_{\boldsymbol{\xi}}(\mathbf{u}) - \Phi_{\boldsymbol{\xi}}(\mathbf{v})\|$ approaches $\|\mathbf{u} - \mathbf{v}\|$ when $\|\mathbf{u} - \mathbf{v}\|$ tends to 0.

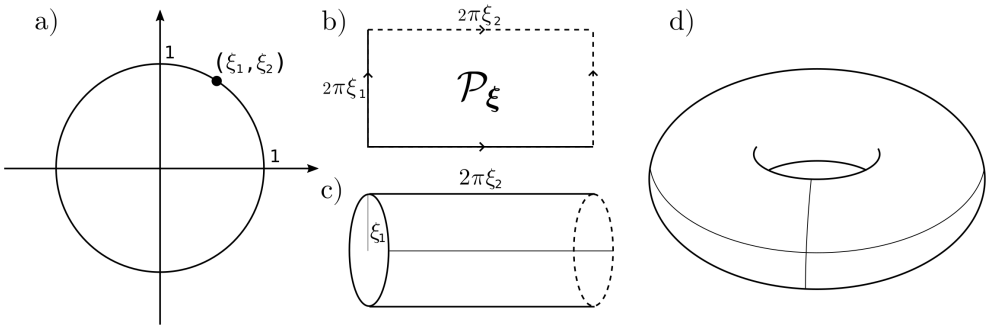


Figure 4.1 – Illustration of a 2-dimensional flat torus $\mathrm{T}_{\boldsymbol{\xi}}$ in $\mathbb{R}^4$ associated with $\Phi_{\boldsymbol{\xi}}$: a) the point $\boldsymbol{\xi}$ on S^1, b) the flat surface $\mathcal{P}_{\boldsymbol{\xi}}$ of the torus, c) the first folding of $\mathcal{P}_{\boldsymbol{\xi}}$, d) the second folding of $\mathcal{P}_{\boldsymbol{\xi}}$ realizable only in $\mathbb{R}^4$. From [Costa et al., 2017].

4.3.3 Spherical commutative group codes from lattices

Figure 4.2 presents a simple example of two nested lattices in $\mathbb{R}^2$ associated with a spherical group code in $\mathbb{R}^4$ through a torus mapping. The red dots in the middle of the picture belong to an orthogonal lattice $\Lambda_\beta = 2\pi\xi_1\mathbb{Z} \times 2\pi\xi_2\mathbb{Z}$, where $(\xi_1, \xi) = (0.8, 0.6)$. Since $\boldsymbol{\xi} = [\xi_1, \xi_2]^T$ is nonnegative with a unit norm, the points of Λ_β can be viewed as vertices of frames that are the pre-images of the flat torus $\mathrm{T}_{\boldsymbol{\xi}}$ through the map $\Phi_{\boldsymbol{\xi}}$.

Besides, the red and black dots combined form another lattice Λ_α such that $\Lambda_\beta \subset \Lambda_\alpha$. It can be noticed, that the quotient $\Lambda_\alpha/\Lambda_\beta$ of order 4 can be mapped by $\Phi_{\boldsymbol{\xi}}$ to some spherical code $\mathscr{C}$ on $S^3 \in \mathbb{R}^4$. Moreover, since $\Lambda_\alpha/\Lambda_\beta$ is closed under translation by a single basis vector (the blue arrow), the code should be closed under an associated rotation on S^3. Consequently, the code $\mathscr{C}$ is a commutative group code of order 4 with a single generator matrix.

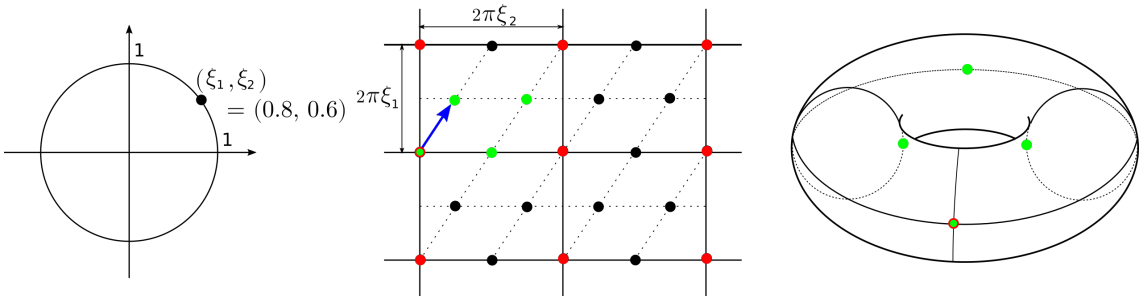


Figure 4.2 – Division of the 2-dimensional plane into frames associated with the flat torus mapping $\Phi_{\boldsymbol{\xi}}$, $\boldsymbol{\xi} = [0.8, 0.6]^T$, and the pair of nested lattices Λ_β (red dots) and Λ_α (black, red, and green dots) defined over the same plane. The image $\Phi_{\boldsymbol{\xi}}(\Lambda_\alpha)$ is a spherical code of order 4 (green dots on the torus). The blue arrow is a basis vector of the quotient $\Lambda_\alpha/\Lambda_\beta$, which can be associated with a basic rotation generating the spherical code on S^3.

As will be detailed in Corollary 4.3.2, there is an isomorphism between some spherical group codes in even dimensions larger than 2 (hence, $\mathscr{C} \subset \mathbb{R}^{2n}$, $n > 1$) and quotients of associated lattices. Conversely, Corollary 4.3.3 presents an explicit construction of a spherical code from a pair of nested lattices.

Corollary 4.3.2. *[Siqueira and Costa, 2008, p. 113] Let $\mathscr{C} = \mathcal{G}\boldsymbol{\sigma} \subset \mathbb{R}^{2n}$ be a spherical commutative group code of order M, where $\mathcal{G}$ is a group of orthogonal matrices free from reflection blocks, and $\boldsymbol{\sigma} = [\xi_1, 0, \xi_2, 0, ..., \xi_n, 0]^T$, $\xi_i \geq 0$, is the initial vector with unit norm. Then, the inverse image of the code $\Phi_{\boldsymbol{\xi}}^{-1}(\mathscr{C})$ is the full rank lattice Λ_α generated by the set:*

$$\{\boldsymbol{\alpha}_i : \boldsymbol{\alpha}_i = [2\pi a_{i,1}/M, ..., 2\pi a_{i,n}/M]^T, 0 \leq a_{i,j} < M,\ a_{i,j} \in \mathbb{Z},\ j = 1, ..., n,\ i = 1, ..., M\},$$

where $2\pi a_{i,j}/M$ come from block-diagonalization of elements in $\mathcal{G}$. The lattice Λ_α has the sublattice $\Lambda_\beta = \prod_{j=1}^{n} 2\pi\xi_j\mathbb{Z}$ with an orthogonal basis, and $\mathcal{G}$ is isomorphic to the quotient $\Lambda_\alpha/\Lambda_\beta$.

Corollary 4.3.3. *[Costa et al., 2017, p. 82] Let $\Lambda_\beta \subset \Lambda_\alpha$ be a pair of full rank lattices with generator matrices $A_\beta = [\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, ..., \boldsymbol{\beta}_n]$ and A_α, respectively. Moreover, let the basis of Λ_β be orthogonal. There exists an integer matrix H such that $A_\beta = A_\alpha H$. Matrix H has a Smith normal form $H = PDQ$ where P and Q are unimodular matrices and D is a diagonal matrix with $\mathrm{diag}(D) = [d_1, d_2, ..., d_n]^T$, $d_i \in \mathbb{N}$ and d_i divides d_{i+1} for $i = 1, ..., n-1$.*

Let us define $b_i = \|\boldsymbol{\beta}_i\|$, $b = \sqrt{\sum_{j=1}^{n} \|\boldsymbol{\beta}_j\|^2}$, $\xi_i = b_i/b$, $\boldsymbol{\xi} = [\xi_1, \xi_2, ..., \xi_n]^T$ and the torus mapping $\Phi_{\boldsymbol{\xi}}$. Then, the quotient of the normalized nested lattices $(2\pi b^{-1}\Lambda_\alpha)/(2\pi b^{-1}\Lambda_\beta)$ is associated with a spherical code $\mathscr{C} \subset S^{2n-1}$ with the initial vector $\boldsymbol{\sigma} = [\xi_1, 0, \xi_2, 0, ..., \xi_n, 0]^T$, and a generator group of matrices determined by the Smith normal decomposition of H.

Discussion Corollary 4.3.3 states that for a given pair of nested lattices $\Lambda_\beta \subset \Lambda_\alpha$, Λ_β being orthogonal, and an integer matrix H such that $A_\beta = A_\alpha H$, with the Smith normal form $H = PDQ$, one can easily get generator matrices of $\mathcal{G}$. Indeed, since P and Q are unimodular, $B_\alpha = A_\alpha P = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n]$ and $B_\beta = A_\beta Q^{-1} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_n]$ are also respective generator matrices of lattices Λ_α and Λ_β. We get $B_\beta = B_\alpha D$ ($\mathbf{w}_i = d_i\mathbf{v}_i$) and $\Lambda_\alpha/\Lambda_\beta \cong \mathbb{Z}_{d_{k_0}} \oplus ... \oplus \mathbb{Z}_{d_n}$ where d_{k_0} is the first element of $\mathrm{Diag}(D)$ larger than 1. Thus, we can associate the quotient $\Lambda_\alpha/\Lambda_\beta$ with some orthogonal $2n \times 2n$ matrix group $\mathcal{G} \cong \mathbb{Z}_{d_{k_0}} \oplus ... \oplus \mathbb{Z}_{d_n}$ of order $\det(H) = d_1 \cdot d_2 \cdot ... \cdot d_n$.

Furthermore, the basis vectors $\mathbf{v}_{k_0}, ..., \mathbf{v}_n$ can be associated with a direction of rotations. However, these vectors should be expressed in terms of the old orthogonal basis A_β which determines the rectangular frame defined as pre-image of the flat torus. As a result, we may express the generator matrices $\{G_1, G_2, ..., G_k\}$ in block-diagonal form:

$$G_j = \begin{bmatrix} \mathrm{Rot}(2\pi r_{j+k_0,1}) & 0 & ... & 0 \\ 0 & \mathrm{Rot}(2\pi r_{j+k_0,2}) & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & \mathrm{Rot}(2\pi r_{j+k_0,n}) \end{bmatrix}_{2n \times 2n}$$

where k is the number of elements in $\mathrm{Diag}(D)$ larger than 1, $k_0 = n - k$, $j = 1, ..., k$ and $r_{j+k_0,i}$ are elements of the matrix $R = A_\beta^{-1} A_\alpha P$.

It can be noticed that performing basic rotations on the codewords of $\mathscr{C}$ is equivalent to translating points in the pre-image of the flat torus. Thus, we can impose some design rules that improve the properties of the constructed spherical code.

Firstly, from the bounds given by Equation 4.13 we get that for any $\mathbf{u}, \mathbf{v} \in (2\pi b^{-1}\Lambda_\alpha)/(2\pi b^{-1}\Lambda_\beta)$ the distance $\|\Phi_{\boldsymbol{\xi}}(\mathbf{u}) - \Phi_{\boldsymbol{\xi}}(\mathbf{v})\|$ is larger than $2\|\mathbf{u} - \mathbf{v}\|/\pi$. Consequently, the maximization of the minimum distance between vectors of $(2\pi b^{-1}\Lambda_\alpha)$ results in the improvement of distance distribution between codewords on the hypersphere S^{2n-1}. In particular, selecting dense lattices like the checkerboard lattice D_n or the Gosset lattice Γ_8 in the construction will lead to a larger minimum distance between codewords in $\mathscr{C}$.

The distribution of codewords on the hypersphere S^{2n-1} can be further improved by selecting a proper vector $\boldsymbol{\xi} = [\xi_1, ..., \xi_n]^T$. Provided that $2\xi_{min}\sin(\|\mathbf{u}-\mathbf{v}\|/(2\xi_{min})) \leq \|\Phi_{\boldsymbol{\xi}}(\mathbf{u}) - \Phi_{\boldsymbol{\xi}}(\mathbf{v})\|$, we should maximize the minimum nonnegative component ξ_{min}. This is achieved by taking $\boldsymbol{\xi}$ such that $\xi_1 = \xi_2 = ... = \xi_n$.

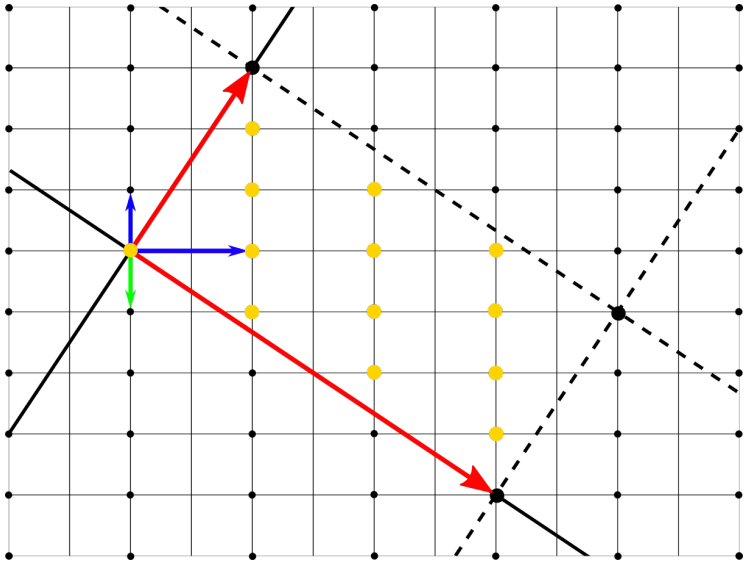


Figure 4.3 – Example of a construction of a spherical code on S^3. The red arrows are the basis vectors of the orthogonal lattice Λ_β, whereas the blue arrows are the basis vectors of the lattice Λ_α such that $\Lambda_\beta \subset \Lambda_\alpha$. The points in $2\pi/\sqrt{65}\Lambda_\beta$ determine the frame associated with the pre-image of the flat torus mapping $\Phi_{\boldsymbol{\xi}}$, where $\boldsymbol{\xi} = [\sqrt{13/65}, \sqrt{52/65}]^T$. The points of Λ_α can be mapped to a spherical code of order 13. The code has only one generator matrix, associated with the basis vector of the quotient $\Lambda_\alpha/\Lambda_\beta$ (the green arrow). Yellow points are the lattice points associated with the spherical code.

Example 4.3.1 – Let the lattices Λ_α and Λ_β have the following generator matrices:

$$A_\alpha = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \qquad A_\beta = \begin{bmatrix} 2 & 6 \\ 3 & -4 \end{bmatrix}.$$

We have $\Lambda_\beta \subset \Lambda_\alpha$ and Λ_β is orthogonal, as seen in Figure 4.3. The matrix H with the Smith normal form:

$$H = \begin{bmatrix} 1 & 3 \\ 3 & -4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 3 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 13 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 0 & 1 \end{bmatrix}.$$

Thus, $\mathcal{G} \cong \Lambda_\alpha/\Lambda_\beta \cong \mathbb{Z}_{13}$ is of order 13 and has one generator matrix. The new basis of Λ_α is $B_\alpha = A_\alpha P$ with two vectors $[2,3]^T$ and $[0,-1]^T$, where only the second contributes to spherical rotations in $\mathcal{G}$ (it can be verified on the picture, that indeed this vector generates a code of order 13). Finally, the vector $[0,-1]^T$ should be expressed in the basis A_β, giving a vector $[-3/13, 1/13]^T$. The 4×4 generator matrix of the spherical commutative group $\mathcal{G}$ is given by:

$$\begin{bmatrix} \mathrm{Rot}(2\pi(-3/13)) & 0 \\ 0 & \mathrm{Rot}(2\pi(1/13)) \end{bmatrix}.$$

Finally, the initial vector of the spherical code is $\boldsymbol{\sigma} = [\sqrt{13/65}, 0, \sqrt{52/65}, 0]^T$.

Example 4.3.2 – Let the lattices $\Lambda_\alpha = \Gamma_8$ and Λ_β have the following generator matrices:

$$A_\alpha = \begin{bmatrix} 2 & -1 & 0 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 \end{bmatrix} \quad A_\beta = \begin{bmatrix} 2k & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2k & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2k & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2k & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2k & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2k & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2k & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2k \end{bmatrix},$$

where $k \in \mathbb{N}$. Matrix H, such that $A_\beta = A_\alpha H$, is of the form:

$$H = \begin{bmatrix} k & k & k & k & k & k & k & -7k \\ 0 & 2k & 2k & 2k & 2k & 2k & 2k & -12k \\ 0 & 0 & 2k & 2k & 2k & 2k & 2k & -10k \\ 0 & 0 & 0 & 2k & 2k & 2k & 2k & -8k \\ 0 & 0 & 0 & 0 & 2k & 2k & 2k & -6k \\ 0 & 0 & 0 & 0 & 0 & 2k & 2k & -4k \\ 0 & 0 & 0 & 0 & 0 & 0 & 2k & -2k \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4k \end{bmatrix},$$

and can be decomposed into its Smith normal form $H = PDQ$:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} k & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2k & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2k & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2k & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2k & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2k & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2k & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4k \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & -7 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & -6 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & -5 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & -4 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & -3 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & -2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

The form of the matrix D indicates that the spherical code is of order $M = k(2k)^6(4k)$, and is isomorphic to $\mathbb{Z}_2^6 \oplus \mathbb{Z}_4$ when $k = 1$ or isomorphic to $\mathbb{Z}_k \oplus \mathbb{Z}_{2k}^6 \oplus \mathbb{Z}_{4k}$ when $k > 1$. From the Smith normal decomposition we obtain 8 generator matrices (7 when $k = 1$) generating the

spherical code $\mathscr{C} = \mathcal{G}\boldsymbol{\sigma} \subset S^{15}$. The columns of the following matrix define angles of rotations in the generator matrices $\{G_1, G_2, ..., G_8\}$ of the group $\mathcal{G}$:

$$2\pi A_{\beta}^{-1} A_{\alpha} P = \begin{bmatrix} 2\pi/k & -2\pi/2k & 0 & 0 & 0 & 0 & 0 & 2\pi/4k \\ 0 & 2\pi/2k & -2\pi/2k & 0 & 0 & 0 & 0 & 2\pi/4k \\ 0 & 0 & 2\pi/2k & -2\pi/2k & 0 & 0 & 0 & 2\pi/4k \\ 0 & 0 & 0 & 2\pi/2k & -2\pi/2k & 0 & 0 & 2\pi/4k \\ 0 & 0 & 0 & 0 & 2\pi/2k & -2\pi/2k & 0 & 2\pi/4k \\ 0 & 0 & 0 & 0 & 0 & 2\pi/2k & -2\pi/2k & 2\pi/4k \\ 0 & 0 & 0 & 0 & 0 & 0 & 2\pi/2k & 2\pi/4k \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\pi/4k \end{bmatrix}.$$

The initial vector of the spherical code is $\boldsymbol{\sigma} = [1, 0, ..., 1, 0]^T/\sqrt{8} \in S^{15}$. Finally, the pre-image of the spherical code $\Phi_{\boldsymbol{\xi}}^{-1}(\mathscr{C})$ is the scaled Gosset lattice:

$$\frac{2\pi}{2k\sqrt{8}}\Gamma_8.$$

4.4 Asymptotic secrecy of pseudo-random generators

The following definitions recall the fundamental secrecy concepts related to symmetric-key encryption schemes and pseudo-random generators. The theory starts with defining a secret-key encryption scheme and the notion of perfect secrecy, and is followed by a more realistic approach when the eavesdropper running in probabilistic polynomial time (PPT) is given some small advantage in breaking the security of the scheme.

Definition 4.4.1. [Katz and Lindell, 2015, §3.2 in Chap. 3] A symmetric encryption scheme is a tuple of probabilistic polynomial-time algorithms $\Pi = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ such that:

1. The key-generation algorithm KeyGen takes as input 1^{λ}, and outputs a key k from the finite key space $\mathcal{K}$. The parameter λ is called a security parameter.
2. The encryption algorithm Enc takes as input a key $k \in \mathcal{K}$ and a plaintext message m from the message space $\mathcal{M}$, and outputs a ciphertext c from the ciphertext space $\mathcal{C}$.
3. The decryption algorithm Dec takes as input a key $k \in \mathcal{K}$ and a ciphertext $c \in \mathcal{C}$, and outputs a message $m \in \mathcal{M}$ or an error.

For every security parameter λ, every key $k \in \mathcal{K}$ and every message $m \in \mathcal{M}$ it holds that $\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m$. If messages cannot be longer than a fixed and predefined value, we say that Π is a fixed-length private-key encryption scheme.

In cryptography and security analysis, we often work with probabilities rather than with particular instantiations. Thus, all the keys k, plaintext messages m, and ciphertexts c are chosen or computed with some probabilistic distribution. Usually, these distributions are being related to random variables K,M and C, so that for example $\Pr(K = k)$ denotes the probability of selecting $k \in \mathcal{K}$. Additionally, random variables K and M are assumed to be independent.

Definition 4.4.2. [Katz and Lindell, 2015, Chap. 2] An encryption scheme $\Pi = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ with a message space $\mathcal{M}$ is perfectly secret if for every probability distribution over $\mathcal{M}$, every message $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$ such that $\Pr(C = c) > 0$, we have:

$$\Pr(M = m \mid C = c) = \Pr(M = m).$$

Definition 4.4.2 states that the encryption scheme Π is perfectly secret if no algorithm with unbounded computational power can get any information about the message m out of the ciphertext c. In practice, it is more convenient to use the notion of indistinguishability in experiment called the adversarial indistinguishability challenge. In the experiment, a stateful algorithm $\mathcal{A}$ (an adversary) specifies two arbitrary messages $m_0, m_1 \in \mathcal{M}$, and then is given uniformly at random one of the ciphertexts $\mathrm{Enc}_k(m_0), \mathrm{Enc}_k(m_1)$ encrypted with a random key. The adversary's goal is to correctly guess if he received the encryption of m_0 or m_1.

Definition 4.4.3. [Katz and Lindell, 2015, Chap. 3] The adversarial indistinguishability challenge $\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi}(\lambda)$ is defined as:

1. The adversary $\mathcal{A}$ is given input 1^λ, and he chooses a pair of distinct messages m_0, m_1 of equal length.
2. A key k is generated by running $\mathrm{KeyGen}(1^\lambda)$, and a bit $b \in \{0,1\}$ is chosen uniformly at random. Then, the computed challenge ciphertext $c = \mathrm{Enc}_k(m_b)$ is given to $\mathcal{A}$.
3. $\mathcal{A}$ outputs bit b'.
4. The output of the challenge is 1 if $b = b'$ and 0 otherwise. If $\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi}(\lambda) = 1$, we say that $\mathcal{A}$ succeeds.

Lemma 4.4.1. *[Katz and Lindell, 2015, Chap. 2] Encryption scheme $\Pi = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ with a message space $\mathcal{M}$ is perfectly secret if and only if it is perfectly indistinguishable, i.e., for every adversary $\mathcal{A}$ it holds:*

$$\Pr(\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi}(\lambda) = 1) = \frac{1}{2}.$$

Perfectly secret encryption schemes are non-practical because the size of the key space $\mathcal{K}$ should be at least as large as the message space $\mathcal{M}$, $|\mathcal{K}| \geq |\mathcal{M}|$ [Shannon, 1949]. For this reason, we often relax secrecy requirements and grant the adversary $\mathcal{A}$ with limited computational power with a small probabilistic chance of breaking the scheme, i.e., to obtain an advantage in the indistinguishability challenge:

$$\Pr(\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi}(\lambda) = 1) > \frac{1}{2}.$$

In return, the size of the key space may become far smaller than the size of the message space.

Definition 4.4.4. [Katz and Lindell, 2015, Chap. 3] A function $\mathbf{f} : \mathbb{N} \to \mathbb{R}_+ \cup \{0\}$ is negligible if for every positive polynomial p there is an integer N such that for all integers $n > N$ it holds that $\mathbf{f}(n) < \frac{1}{p(n)}$.

Proposition 4.4.2. *[Katz and Lindell, 2015, Chap. 3] Let $\mathbf{negl}_1$ and $\mathbf{negl}_2$ be negligible functions. We have:*

1. *The function $\mathbf{negl}_3(n) = \mathbf{negl}_1(n) + \mathbf{negl}_2(n)$ is negligible.*
2. *For any positive polynomial p, the function $\mathbf{negl}_4(n) = p(n) \cdot \mathbf{negl}_1(n)$ is negligible.*

In the asymptotic approach, we let all PPT adversaries to get at most negligible advantage over the scheme, given some integer-valued security parameter λ. In the indistinguishability challenge, the advantage $\mathrm{Adv}(\mathcal{A}(1^\lambda))$ is defined as the absolute difference between the success probability achieved by $\mathcal{A}$ compared to a random guess.

Definition 4.4.5. [Katz and Lindell, 2015, Chap. 3] A symmetric encryption scheme $\Pi = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ has indistinguishable encryptions in the presence of an eavesdropper, if for all PPT adversaries $\mathcal{A}$ there is a negligible function $\mathbf{negl}$ such that for all λ it holds:

$$\mathrm{Adv}(\mathcal{A}(1^\lambda)) = \left|\Pr(\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi}(\lambda) = 1) - \frac{1}{2}\right| \leq \mathbf{negl}(\lambda),$$

where the probability is taken over the randomness used by $\mathcal{A}$ and the randomness used in the challenge.

Finally, we recall the definition of a non-binary *pseudorandom generator*, as a function $\{0,1\}^{l(\lambda)} \rightarrow \{0,1,...,q-1\}^{L(\lambda)}$, $l(\lambda) < L(\lambda)$ polynomials, which given a perfectly binary random seed $s \in \{0,1\}^{l(\lambda)}$, outputs a string of $L(\lambda)$ integers modulo q with a distribution indistinguishable from the uniform by any PPT statistical test (distinguisher). The test returns 1 when it distinguishes an input string's distribution from the uniform distribution and returns 0 otherwise. Pseudo-random generators can be viewed as instantiations of *stream cipher keys*, widely adopted in modern cryptography. The generators can output discrete values different from only 0 and 1.

Definition 4.4.6. Based on [Katz and Lindell, 2015, Chap. 3]. Let L and l be polynomials and let $\mathcal{G}_q$ be a polynomial-time algorithm such that $q \in \mathbb{N}$ and for any λ and input $s \in \{0,1\}^{l(\lambda)}$ the output of $\mathcal{G}_q(s)$ is $w \in \{0,1,...,q-1\}^{L(\lambda)}$. We say that $\mathcal{G}_q$ is a pseudo-random generator, if the following conditions hold:

1. For every λ it holds that $L(\lambda) > l(\lambda)$.
2. For any PPT distinguisher $\mathcal{D}$, there is a negligible function $\mathbf{negl}$ such that

$$\mathrm{Adv}(\mathcal{D}(1^\lambda)) = |\Pr(\mathcal{D}(\mathcal{G}_q(s)) = 1) - \Pr(\mathcal{D}(r) = 1)| \leq \mathbf{negl}(\lambda),$$

where the first probability is taken over uniform choices of $s \in \{0,1\}^{l(\lambda)}$ and randomness of $\mathcal{D}$, and the second probability is taken over uniform choices of $r \in \{0,1,...,q-1\}^{L(\lambda)}$ and randomness of $\mathcal{D}$.

4.5 Distortion-tolerant encryption

In its primary sense, a distortion-tolerant property denotes the capability to decipher ciphertexts distorted by a transmission channel. Traditionally, encrypted data are protected by error correction coding with a predefined correction capability. In this work, we consider encryption schemes that output an approximation of the initial message, given a distorted ciphertext.

The distortion-tolerant property takes inspiration from the notion of distance-preserving encryption described in Definition 4.5.1.

Definition 4.5.1. [Tex et al., 2018] Let $\mathcal{M}$ be a data set, $\mathcal{K}$ be a key space, d be a distance measure and Enc be an encryption algorithm for data items in $\mathcal{M}$. Then, Enc is d-distance preserving if:

$$\forall m_0, m_1 \in \mathcal{M} \text{ and } \forall k \in \mathcal{K} \ : \ \mathrm{d}(\mathrm{Enc}_k(m_0), \mathrm{Enc}_k(m_1)) = \mathrm{d}(m_1, m_2).$$

When considering encryption of databases, Definition 4.5.1 cannot be relaxed without introducing inaccuracies in the result of queries on encrypted data [Tex et al., 2018]. However, in the case of perceptual audio-visual data, small inaccuracies are usually acceptable or perceptually irrelevant. It is especially true in real-time applications that prioritize robustness and efficiency over the quality of representation.

Definition 4.5.2. Let $\mathrm{d}_{\mathcal{M}}$ and $\mathrm{d}_{\mathcal{C}}$ denote distance measures over the plaintext space $\mathcal{M}$ and the ciphertext space $\mathcal{C}$, respectively. We say that the encryption scheme $\Pi = (\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ is distortion-tolerant with respect to these measures, if for every key $k \in \mathcal{K}$, any two ciphertexts $c_1, c_2 \in \mathcal{C}$, and $\delta > 0$ not too large, there is $\tau > 0$ such that:

1. $\mathrm{d}_{\mathcal{C}}(c_1, c_2) < \delta \implies \mathrm{d}_{\mathcal{M}}(\mathrm{Dec}_k(c_1), \mathrm{Dec}_k(c_2)) < \tau\delta$.
2. $\tau\delta \ll \max_{c_i, c_j \in \mathcal{C}} \mathrm{d}_{\mathcal{M}}(\mathrm{Dec}_k(c_i), \mathrm{Dec}_k(c_j))$.

The notion of distortion-tolerant encryption introduced here is considerably relaxed compared to the notion of distance-preserving encryption. Firstly, we use two different metrics over the message and the ciphertext spaces. Furthermore, we allow some distance expansion τ between the decrypted plaintexts $\mathrm{Dec}_k(c_1)$ and $\mathrm{Dec}_k(c_2)$, which is still small compared to the maximum distance between plaintexts. Finally, the distortion-tolerant property applies locally in a ciphertext neighborhood.

Every encryption scheme with a distortion-tolerant property is malleable by design. [1] Without additional data-integrity mechanisms, an active attacker can modify the decrypted plaintext by carefully changing the encrypted data. Moreover, given a pair $(m, \mathrm{Enc}_k(m))$, the attacker may easily guess or approximate the decryption result of all ciphertexts close to $\mathrm{Enc}_k(m)$. On the other hand, the malleability does not necessarily compromise the secrecy of encryptions if the enciphering algorithm uses a fresh cryptographic key $k \in \mathcal{K}$ for every encryption.

4.6 Enciphering using spherical codes

This section introduces a lossy enciphering technique for scrambling a sequence of points on the unit hypersphere S^n, $n > 1$. The encryption scheme encodes the spherical points to codewords of a spherical commutative group code $\mathscr{C} \subset S^{2n-1}$ and performs pseudo-random rotations from a group of orthogonal matrices $\mathcal{G}$. The result of this enciphering is a randomly-looking sequence of codewords from $\mathscr{C}$ with a distribution indistinguishable from the uniform distribution.

This enciphering of codewords using rotations from $\mathcal{G}$ satisfies the distance-preserving property because the spherical group code is geometrically uniform and also closed under rotations from $\mathcal{G}$. It turns out to be very useful if we consider the transmission of scrambled sequences of codewords over a noisy channel. Provided that the channel noise level is not too high, this noise would map into the decrypted plaintext without introducing much error.

4.6.1 Overview of the encryption scheme

The enciphering procedure scrambles a fixed-length sequence of spherical points $X = (\mathbf{x}_1, ..., \mathbf{x}_{L(\lambda)})$, $\mathbf{x}_\ell \in S^n$, into a sequence of spherical codewords $U = (\mathbf{u}_1, ..., \mathbf{u}_{L(\lambda)})$, $\mathbf{u}_\ell \in \mathscr{C}$, where λ is an integer-valued security parameter and L is a polynomial. The spherical code $\mathscr{C}$ lies on the flat torus $\mathrm{T}_{\boldsymbol{\xi}}$ and is associated with the quotient of two full rank lattices $\Lambda_\beta \subset \Lambda_\alpha \subset \mathbb{R}^n$ through a flat torus mapping $\Phi_{\boldsymbol{\xi}}$, $\boldsymbol{\xi} = [\xi_1, ..., \xi_n]^T$. Furthermore, the code has initial vector $\boldsymbol{\sigma} = [\xi_1, 0, ..., \xi_n, 0]^T$ and is generated by a group $\mathcal{G}$ of $2n \times 2n$ orthogonal matrices with k generator matrices $\{G_1, ..., G_k\}$ isomorphic to $\mathbb{Z}_{d_1} \oplus ... \oplus \mathbb{Z}_{d_k}$, where $d_1 \cdot .... \cdot d_k = M$ is the order of the group.

1. An encryption algorithm is 'malleable' if it is possible to transform a ciphertext into another ciphertext which decrypts to a bona fide plaintext.

The diagram illustrating the enciphering scheme is shown in Figure 4.4. The scheme consists of six procedures (KeyGen, RandGen, Encode, Decode, Encrypt, Decrypt):

1. $\mathrm{KeyGen}(1^\lambda)$: for a given integer-valued security parameter λ (preferably $\lambda = 256$, and no less than 128), $\mathrm{KeyGen}(1^\lambda)$ outputs a secret seed s chosen uniformly from $\{0,1\}^{l(\lambda)}$, where l is a polynomial such that $l(\lambda) < L(\lambda)$.
2. $\mathrm{RandGen}(s)$: for a given secret seed s, $\mathrm{RandGen}(s)$ generates a sequence of $L(\lambda)$ randomly-looking, positive modulo integers $\mathbf{r} = (r_1, ..., r_{L(\lambda)})$, $1 \le r_\ell \le M$.
3. $\mathrm{Encode}(X)$: for a given sequence of spherical points $X = (\mathbf{x}_1, ..., \mathbf{x}_{L(\lambda)})$, $\mathbf{x}_\ell \in S^n$, the algorithm encodes every $\mathbf{x}_\ell$ into a codeword $\mathbf{p}_\ell \in \mathscr{C}$ and outputs a sequence of spherical codewords $P = (\mathbf{p}_1, ..., \mathbf{p}_{L(\lambda)})$.
4. $\mathrm{Decode}(Q)$: for a given sequence $Q = (\mathbf{q}_1, ..., \mathbf{q}_{L(\lambda)})$ of points on the flat torus $\mathrm{T}_{\boldsymbol{\xi}}$, the algorithm maps every $\mathbf{q}_\ell$ into a point $\mathbf{y}_\ell \in S^n$ and outputs a sequence $Y = (\mathbf{y}_1, ..., \mathbf{y}_{L(\lambda)})$ of points on S^n.
5. $\mathrm{Encrypt}_{\mathbf{r}}(P)$: for a given sequence of positive integers $\mathbf{r} = (r_1, ..., r_{L(\lambda)})$, $r_\ell \le M$, and a sequence of codewords $P = (\mathbf{p}_1, ..., \mathbf{p}_{L(\lambda)})$, $\mathbf{p}_\ell \in \mathscr{C}$, the enciphering procedure transforms every codeword $\mathbf{p}_\ell$ into another codeword $\mathbf{u}_\ell$ and outputs a sequence $U = (\mathbf{u}_1, ..., \mathbf{u}_{L(\lambda)})$ depending upon $\mathbf{r}$.
6. $\mathrm{Decrypt}_{\mathbf{r}}(V)$: for a given sequence of positive integers $\mathbf{r} = (r_1, ..., r_{L(\lambda)})$, $r_\ell \le M$, and a sequence $V = (\mathbf{v}_1, ..., \mathbf{v}_{L(\lambda)})$ of points on the flat torus $\mathrm{T}_{\boldsymbol{\xi}}$, the deciphering procedure outputs a sequence $Q = (\mathbf{q}_1, ..., \mathbf{q}_{L(\lambda)})$ of points on $\mathrm{T}_{\boldsymbol{\xi}}$ depending upon $\mathbf{r}$.

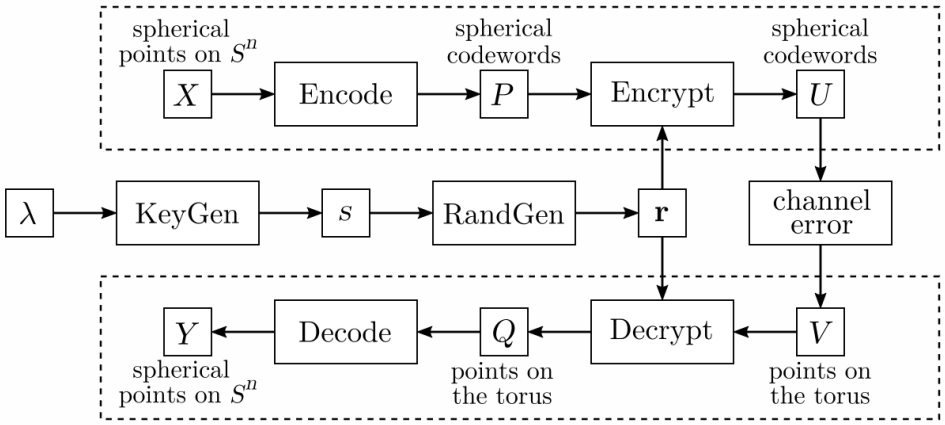


Figure 4.4 – Simplified diagram of the encryption scheme for scrambling spherical data.

4.6.2 Encoding

The encoding procedure $\mathrm{Encode}(X)$, $X = (\mathbf{x}_1, ..., \mathbf{x}_{L(\lambda)})$ is performed by blocks, i.e., $\mathrm{Encode}(X) = (\mathrm{Encode}(\mathbf{x}_1), ..., \mathrm{Encode}(\mathbf{x}_{L(\lambda)}))$. Encoding of every $\mathbf{x}_\ell$ consists of two consecutive actions: mapping $\mathbf{x}_\ell$ on the flat torus $\mathrm{T}_{\boldsymbol{\xi}}$ and then searching a close codeword $\mathbf{p}_\ell \in \mathscr{C}$ to that mapped point.

Let γ_n be a function which maps any point $\mathbf{x} \in S^n$ to a vector $\boldsymbol{\varphi} = [\varphi_1, ..., \varphi_n]^T$, where $\varphi_1, ..., \varphi_{n-1} \in [0, \pi)$ and $\varphi_n \in [0, 2\pi)$ are the spherical coordinates of $\mathbf{x} = [x_1, ..., x_{n+1}]^T$ such that:

$$\begin{cases} x_1 & = \cos(\varphi_1), \\ x_2 & = \sin(\varphi_1)\cos(\varphi_2), \\ & \vdots \\ x_n & = \sin(\varphi_1) \cdot ... \cdot \sin(\varphi_{n-1})\cos(\varphi_n), \\ x_{n+1} & = \sin(\varphi_1) \cdot ... \cdot \sin(\varphi_{n-1})\sin(\varphi_n). \end{cases}$$

The point $\boldsymbol{\xi} \odot \boldsymbol{\varphi}$, where $\odot$ denotes the Hadamard product, [2] lies inside the hyperbox $\prod_{i=1}^{n}[0, 2\pi\xi_i)$ and can be mapped to a new point $\Phi_{\boldsymbol{\xi}}(\boldsymbol{\xi} \odot \boldsymbol{\varphi})$ on the flat torus $\mathrm{T}_{\boldsymbol{\xi}}$.

The second step is to find a codeword $\mathbf{p} \in \mathscr{C}$ close to $\Phi_{\boldsymbol{\xi}}(\boldsymbol{\xi} \odot \boldsymbol{\varphi})$ in terms of Euclidean distance in a process which can be viewed as quantization. The higher is the order of the code, the smaller the quantization error gets. Other factors contributing to the quantization error are the density of the lattice Λ_α, and the selection process of a vector $\boldsymbol{\xi}$ which determines the geometrical distribution of codewords on the sphere S^{2n-1}.

The search can be considerably simplified by taking advantage of the isomorphism between the group $\mathcal{G}$ and the quotient $\Lambda_\alpha/\Lambda_\beta$. Instead of comparing distances between $\Phi_{\boldsymbol{\xi}}(\boldsymbol{\xi} \odot \boldsymbol{\varphi})$ and the codewords in $\mathscr{C}$, one may find the closest lattice point $\mathbf{z} \in \Lambda_\alpha$ to $\boldsymbol{\xi} \odot \boldsymbol{\varphi}$ and map this point to $\mathbf{p} = \Phi_{\boldsymbol{\xi}}(\mathbf{z})$. Unfortunately, the closest lattice vector problem can become quite challenging unless the lattice Λ_α has a special structure. For this reason, it is advisable to construct a spherical code $\mathscr{C}$ associated with a lattice Λ_α isomorphic to one of the special lattices with efficient decoding procedures (i.e., Z^n, D_n or Γ_8).

Although $\mathbf{z}$ is the closest lattice point to $\boldsymbol{\xi} \odot \boldsymbol{\varphi}$, the point $\Phi_{\boldsymbol{\xi}}(\mathbf{z})$ may be not the closest to $\Phi_{\boldsymbol{\xi}}(\boldsymbol{\xi} \odot \boldsymbol{\varphi})$. Lemma 4.6.1 demonstrates that the maximum distance of $\Phi_{\boldsymbol{\xi}}(\mathbf{z})$ from the true closest point $\mathbf{c} \in \mathscr{C}$ does not exceed 2μ, where μ is the covering radius of Λ_α. Thus, provided a sufficiently dense lattice Λ_α with a small covering radius, the overall quantization error is reduced.

Lemma 4.6.1. *Let $\mathbf{x} \in \mathbb{R}^n$ and let $\mathscr{C}$ be a commutative group code in $\mathbb{R}^{2n}$ associated with a lattice Λ in $\mathbb{R}^n$ through an inverse image $\Phi_{\boldsymbol{\xi}}^{-1}(\mathscr{C})$. Let $\mathbf{z}$ be the closest point of Λ to $\mathbf{x}$ in terms of Euclidean metric. Then, the distance between $\Phi_{\boldsymbol{\xi}}(\mathbf{z})$ and the closest code of $\mathscr{C}$ to $\Phi_{\boldsymbol{\xi}}(\mathbf{x})$ gets no larger than 2μ, where μ is the covering radius of Λ.*

Proof.

Let $\mathcal{Q}_{\mathbf{z}} = \{\mathbf{q} \in \Lambda : \mathcal{V}(\mathbf{q}) \cap \mathcal{V}(\mathbf{z}) \neq \varnothing\}$ be the set containing $\mathbf{z}$ and its closest lattice neighbors in terms of Euclidean metric, and let $\mathcal{F}(\Phi_{\boldsymbol{\xi}}(\mathcal{Q}_{\mathbf{z}}))$ be the smallest convex polytope on the surface of S^{2n-1} that contains $\Phi_{\boldsymbol{\xi}}(\mathcal{Q}_{\mathbf{z}})$ (without loss of generality, we can assume that such a polytope exists). We have $\Phi_{\boldsymbol{\xi}}(\mathbf{x}) \in \mathcal{F}(\Phi_{\boldsymbol{\xi}}(\mathcal{Q}_{\mathbf{z}}))$. In addition, $\mathcal{F}(\Phi_{\boldsymbol{\xi}}(\mathcal{Q}_{\mathbf{z}}))$ lies on the union of all the Voronoi regions of $\Phi_{\boldsymbol{\xi}}(\mathbf{q}) \in \Phi_{\boldsymbol{\xi}}(\mathcal{Q}_{\mathbf{z}})$. Thus, the closest codeword $\mathbf{c} \in \mathscr{C}$ to $\Phi_{\boldsymbol{\xi}}(\mathbf{x})$ also belongs to $\mathcal{F}(\Phi_{\boldsymbol{\xi}}(\mathcal{Q}_{\mathbf{z}}))$. Finally:

$$\|\Phi_{\boldsymbol{\xi}}(\mathbf{c}) - \Phi_{\boldsymbol{\xi}}(\mathbf{x})\| \leq \|\Phi_{\boldsymbol{\xi}}(\mathbf{c}) - \Phi_{\boldsymbol{\xi}}(\mathbf{z})\| \leq \max_{\mathbf{q} \in \mathcal{Q}_{\mathbf{z}}} \|\Phi_{\boldsymbol{\xi}}(\mathbf{q}) - \Phi_{\boldsymbol{\xi}}(\mathbf{z})\| \leq \max_{\mathbf{q} \in \mathcal{Q}_{\mathbf{z}}} \|\mathbf{q} - \mathbf{z}\| \leq 2\mu \,.$$

□

2. For two matrices A and B of size $n \times m$, $A \odot B$ is the $n \times m$ matrix C such that $C_{ij} = A_{ij}B_{ij}$.

We can notice, that the quantization on S^n is non-uniform. Since the encoding process involves quantizing spherical coordinates $\varphi_1, ..., \varphi_{n-1}$ using a dense lattice, the finest quantization resolution concentrates near the poles of S^n and the coarsest close to the equator. This characteristic must be taken into account when constructing a sufficiently dense encoding that minimizes the maximum quantization error on S^n.

4.6.3 Decoding

In essence, the procedure Decode reverses the operations of the procedure Encode, i.e., it maps a sequence $Q = (\mathbf{q}_1, ..., \mathbf{q}_{L(\lambda)})$ of points on $\mathrm{T}_{\boldsymbol{\xi}}$ back to a sequence $Y = (\mathbf{y}_1, ..., \mathbf{y}_{L(\lambda)})$ of points on the hypersphere S^n.

For every $\mathbf{q} \in \mathrm{T}_{\boldsymbol{\xi}}$, let $\mathbf{z} = \Phi_{\boldsymbol{\xi}}^{-1}(\mathbf{q})$ be restricted to the hyperbox $\prod_{i=1}^{n}[0, 2\pi\xi_j)$. Then, the rescaled version of $\mathbf{z}$ defined as $\mathbf{z}' = [z_1/\xi_1, ..., z_n/\xi_n]^T$ belongs to $\prod_{j=1}^{n}[0, 2\pi)$. For $i = 1, ..., n-1$, if the $i-th$ coordinate of $\mathbf{z}' = [z'_1, ..., z'_n]^T$ is larger or equal to π, we substitute z'_i by $2\pi - z'_i$. Finally, we apply the inverse $\mathbf{y} = \gamma_n^{-1}(\mathbf{z}') \in S^n$.

4.6.4 Encryption

The enciphering procedure $\mathrm{Encrypt}$ scrambles the input sequence of spherical codewords $P = (\mathbf{p}_1, ..., \mathbf{p}_{L(\lambda)})$ by performing orthogonal matrix multiplications. The rotation matrices are selected from the group $\mathcal{G}$ with regard to a pseudo-random sequence of positive integers $\mathbf{r} = (r_1, ..., r_{L(\lambda)})$, $r_\ell \leq M$ from the output of $\mathrm{RandGen}$. The procedure can be expressed as:

$$\mathrm{Encrypt}_{\mathbf{r}}(P) = (\mathrm{Select}_{\mathcal{G}}(r_1) \cdot \mathbf{p}_1,\ \mathrm{Select}_{\mathcal{G}}(r_2) \cdot \mathbf{p}_2, ...,\ \mathrm{Select}_{\mathcal{G}}(r_{L(\lambda)}) \cdot \mathbf{p}_{L(\lambda)}), \tag{4.14}$$

where $\mathrm{Select}_{\mathcal{G}}(r_\ell)$ is some deterministic and injective function which chooses one of the M matrices from $\mathcal{G}$. The selection of matrices from the group $\mathcal{G}$ can be realized efficiently by taking advantage of the isomorphism between $\mathcal{G}$ and the cubic lattice $\mathbb{Z}_{d_1} \oplus ... \oplus \mathbb{Z}_{d_k}$, where $d_1 \cdot .... \cdot d_k = M$.

The principle of the procedure $\mathrm{Encrypt}$ is analogous to enciphering using a binary stream cipher. The only substantial difference is the cardinality of possible values in the pseudo-random sequence. Consequently, the secrecy of the procedure in the presence of an eavesdropper should depend only on the quality of the pseudo-random generator. The well-known secrecy properties of enciphering by binary stream ciphers [Katz and Lindell, 2015] are extended to non-binary ciphers in Lemmas 4.6.2 and 4.6.3. Furthermore, Lemma 4.6.4 demonstrates that an eavesdropper cannot get any partial information from the ciphertext (i.e., to approximate the plaintext value).

Lemma 4.6.2. *Let $\mathscr{C} = \mathcal{G}\boldsymbol{\sigma}$ be a spherical commutative group code of order M with the initial vector $\boldsymbol{\sigma} \in S^{2n-1}$, and associated with a commutative group of orthogonal matrices $\mathcal{G}$. Moreover, let $\mathrm{TrueRandGen}(\lambda)$ be a function which outputs true random entropy $\mathbf{r} \xleftarrow{\$} \{1, ..., M\}^{L(\lambda)}$ obtained by an entropy collector, where $\xleftarrow{\$}$ denotes a uniformly distributed probabilistic process assignment. Then, the encryption scheme $\tilde{\Pi} = (\mathrm{TrueRandGen}, \mathrm{Encrypt}, \mathrm{Decrypt})$ with messages of fixed length $L(\lambda)$ is perfectly secret.*

Proof.

Let $\mathcal{P}$ be a random variable such that $\Pr(\mathcal{P} = P)$ is the probability of selecting the vector of plaintext codewords P, and let $\mathcal{U}$ be a random variable such that $\Pr(\mathcal{U} = U)$ denotes the

probability of obtaining the encrypted vector of codewords $U = \mathrm{Encrypt}_{\mathbf{r}}(P)$. Recalling Definition 4.4.2, the scheme $\tilde{\Pi}$ would be perfectly secure if for every vector of codewords P chosen with any probability distribution, and every vector of encrypted codewords U, it holds that:

$$\Pr(\mathcal{P} = P \mid \mathcal{U} = U) = \Pr(\mathcal{P} = P).$$

We will firstly prove that given a uniformly random vector $\mathbf{r} \xleftarrow{\$} \mathrm{TrueRandGen}(\lambda)$, each $\mathbf{u}_\ell \in U$ can be any codeword of $\mathscr{C}$ with uniform probability. Then, we will show the mutual independence of $\mathbf{u}_1, ..., \mathbf{u}_{L(\lambda)}$, resulting in U being a vector of uniformly random codewords from $\mathscr{C}$ for any distribution of $\mathcal{P}$. Finally, we will prove the perfect secrecy of $\tilde{\Pi}$.

Let $\mathrm{Select}_{\mathcal{G}}(r_\ell) = G_{r_\ell}$. Since $\mathbf{p}_\ell, \mathbf{u}_\ell \in \mathscr{C}$, there exists $G_{w_\ell} \in \mathcal{G}$ such that $\mathbf{p}_\ell = G_{w_\ell}\boldsymbol{\sigma}$ and $\mathbf{u}_\ell = G_{r_\ell}G_{w_\ell}\boldsymbol{\sigma}$. It is enough to show, that $G_{enc,\ell} = G_{r_\ell}G_{w_\ell}$ can be any element of $\mathcal{G}$ with uniform probability.

The group of orthogonal matrices $\mathcal{G}$ with multiplication operation is isomorphic to $\mathcal{Z} = \mathbb{Z}_{d_1} \oplus ... \oplus \mathbb{Z}_{d_j}$ with addition operation, where $d_i \in \mathbb{N}$ and $d_1 \cdot .... \cdot d_j = M$. Therefore, $G_{w_\ell}, G_{r_\ell}, G_{enc,\ell}$ can be isomorphically mapped to $s_{w_\ell}, s_{r_\ell}, s_{enc,\ell} \in \mathcal{Z}$, where $s_{r_\ell} + s_{w_\ell} = s_{enc,\ell}$.

Let $X_\ell : \Omega_{X_\ell} \to \mathcal{Z}$ and $Y_\ell : \Omega_{Y_\ell} \to \mathcal{Z}$ be independent random variables such that X_ℓ has a uniform distribution over $\mathcal{Z}$ and Y_ℓ has the same probability distribution as $\mathcal{P}$. We have:

$$\begin{aligned}
\Pr(\mathrm{Encrypt}_{r_\ell}(\mathbf{p}_\ell) = G_{enc,\ell}\boldsymbol{\sigma}) &= \Pr(X_\ell + Y_\ell = s_{enc,\ell}) \\
&= \sum_{m=1}^{M} \sum_{s_n \in A_n - s_m} \Pr(X_\ell = s_m) \Pr(Y_\ell = s_n) \\
&= \sum_{m=1}^{M} \sum_{s_n \in A_n - s_m} \frac{1}{M} \Pr(Y_\ell = s_n) \\
&= \frac{1}{M} \sum_{m=1}^{M} \sum_{s_n \in A_n - s_m} \Pr(Y_\ell = s_n) = \frac{1}{M},
\end{aligned}$$

where $A_n - s_m = \{s_n \in \mathcal{Z} : s_n + s_m = s_{enc,\ell}\}$.

We have yet to show the statistical independence. For X_ℓ, Y_ℓ defined as before, let $Z_\ell = X_\ell + Y_\ell$. Z_ℓ are mutually independent if and only if:

$$\Pr[Z_\ell = s_{k_\ell} \mid (Z_1 = s_{k_1}) \wedge ... \wedge (Z_{\ell-1} = s_{k_{\ell-1}}) \wedge (Z_{\ell+1} = s_{k_{\ell+1}}) \wedge ... \wedge (Z_{L(\lambda)} = s_{k_{L(\lambda)}})] = \Pr(Z_\ell = s_{k_\ell}).$$

Let $[(Z_1 = s_{k_1}) \wedge ... \wedge (Z_{\ell-1} = s_{k_{\ell-1}}) \wedge (Z_{\ell+1} = s_{k_{\ell+1}}) \wedge ... \wedge (Z_{L(\lambda)} = s_{k_{L(\lambda)}})]$ be denoted by $B_{\ell,k}$. Then, for any $1 \leq \ell \leq L(\lambda)$ we have:

$$\begin{aligned}
\Pr(Z_\ell = s_{k_\ell} \mid B_{\ell,k}) &= \Pr(X_\ell + Y_\ell = s_{k_\ell} \mid B_{\ell,k}) \\
&= \sum_{m=1}^{M} \sum_{s_n \in A_n - s_m} \Pr(X_\ell = s_m \mid B_{\ell,k}) \Pr(Y_\ell = s_n \mid B_{\ell,k}) \\
&= \sum_{m=1}^{M} \sum_{s_n \in A_n - s_m} \Pr(X_\ell = s_m) \Pr(Y_\ell = s_n \mid B_{\ell,k}) \\
&= \sum_{m=1}^{M} \sum_{s_n \in A_n - s_m} \frac{1}{M} \Pr(Y_\ell = s_n \mid B_{\ell,k}) = \frac{1}{M} \\
&= \Pr(Z_\ell = s_{k_\ell}),
\end{aligned}$$

where $A_n - s_m$ is defined as before. Finally, we get:

$$\begin{aligned}\Pr(\mathcal{P} = P \mid \mathcal{U} = U) &= \frac{\Pr(\mathcal{U} = U \mid \mathcal{P} = P) \cdot \Pr(\mathcal{P} = P)}{\Pr(\mathcal{U} = U)} \\ &= \frac{\Pr(\text{Encrypt}_{\mathbf{r}}(P) = U) \cdot \Pr(\mathcal{P} = P)}{\Pr(\mathcal{U} = U)} \\ &= \frac{M^{-L(\lambda)} \cdot \Pr(\mathcal{P} = P)}{M^{-L(\lambda)}} = \Pr(\mathcal{P} = P).\end{aligned}$$

□

The consequence of Lemma 4.6.2 is that every PPT adversary $\mathcal{A}$ has no advantage in the indistinguishability challenge if the matrices from $\mathcal{G}$ are selected uniformly at random:

$$\Pr(\text{PrivK}^{eav}_{\mathcal{A},\tilde{\Pi}}(\lambda) = 1) = \frac{1}{2}.$$

Lemma 4.6.3. *Let* $\mathbf{r} = \text{RandGen}(s)$*, where* RandGen *is a pseudo-random generator,* s *is chosen uniformly at random from* $\{0,1\}^{l(\lambda)}$ *and* $\mathbf{r} \in \{1, ..., M\}^{L(\lambda)}$*. In addition, let* $\mathscr{C}$ *be a spherical commutative group code of order* M *with the initial vector* $\boldsymbol{\sigma} \in S^{2n-1}$ *and associated with a commutative group of orthogonal matrices* $\mathcal{G}$*. Then, for any sequence* $P = (\mathbf{p}_1, ..., \mathbf{p}_{L(\lambda)})$ *of codewords* $\mathbf{p}_\ell \in \mathscr{C}$ *chosen independently from* s*, the procedure* $\text{Encrypt}_{\mathbf{r}}(P)$ *gives indistinguishable encryptions in the presence of an eavesdropper.*

Proof.

Let $\Pi = (\text{RandGen}, \text{Encrypt}, \text{Decrypt})$ and $\tilde{\Pi} = (\text{TrueRandGen}, \text{Encrypt}, \text{Decrypt})$, where TrueRandGen is a function which outputs true random entropy over $\{1, ..., M\}^{L(\lambda)}$ obtained by an entropy collector. We will show that the existence of any PPT adversary gaining a non-negligible advantage in the indistinguishability challenge over the encryption scheme Π implies the existence of a PPT distinguisher that differentiates RandGen from TrueRandGen.

Let $\mathcal{A}$ be a PPT adversary who gets a non-negligible advantage in the indistinguishability challenge $\text{PrivK}^{eav}_{\mathcal{A},\Pi}$ and let $\mathbf{f}$ be a non-negligible function such that:

$$\Pr(\text{PrivK}^{eav}_{\mathcal{A},\Pi}(\lambda) = 1) > \frac{1}{2} + \mathbf{f}(\lambda).$$

Moreover, let $\mathcal{D}$ be an algorithm which tries to distinguish the outputs of RandGen from TrueRandGen by emulating the indistinguishability challenge with $\mathcal{A}$. The behavior of $\mathcal{D}$ is described as follows:

1. $\mathcal{D}$ is given a vector $\mathbf{r}$ of length $L(\lambda)$ generated by RandGen or by TrueRandGen.
2. The algorithm asks $\mathcal{A}(1^\lambda)$ to produce two vectors of codewords $U_0 = [\mathbf{u}_{0,1}, ..., \mathbf{u}_{0,L(\lambda)}]$ and $U_1 = [\mathbf{u}_{1,1}, ..., \mathbf{u}_{1,L(\lambda)}]$, where $\mathbf{u}_{0,\ell}, \mathbf{u}_{1,\ell} \in \mathscr{C}$ for $\ell = 1, ..., L(\lambda)$.
3. $\mathcal{D}$ randomly chooses $b \in \{0,1\}$ and produces $V = \text{Encrypt}_{\mathbf{r}}(U_b)$.
4. $\mathcal{D}$ provides V to $\mathcal{A}$ and then obtains b'. Finally, $\mathcal{D}$ outputs 1 if $b' = b$ and 0 otherwise.

$\mathcal{D}$ is PPT, because $\mathcal{A}$ is PPT. Moreover, $\mathcal{D}$ outputs 1 with exactly the same probability as $\mathcal{A}$ in the indistinguishability challenge:

$$\begin{aligned}\Pr(\mathcal{D}(\mathbf{r} \leftarrow \text{TrueRandGen}) = 1) &= \Pr(\text{PrivK}^{eav}_{\mathcal{A},\tilde{\Pi}}(\lambda) = 1) = \frac{1}{2} \\ \Pr(\mathcal{D}(\mathbf{r} \leftarrow \text{RandGen}(s)) = 1) &= \Pr(\text{PrivK}^{eav}_{\mathcal{A},\Pi}(\lambda) = 1) > \frac{1}{2} + \mathbf{f}(\lambda).\end{aligned}$$

Finally, we get:

$$\mathrm{Adv}(\mathcal{D}(1^\lambda)) = |\Pr(\mathcal{D}(\mathbf{r} \leftarrow \mathrm{RandGen}(s)) = 1) - \Pr(\mathcal{D}(\mathbf{r} \leftarrow \mathrm{TrueRandGen}) = 1)| =$$
$$= \left|\Pr(\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi}(\lambda) = 1) - \Pr(\mathrm{PrivK}^{eav}_{\mathcal{A},\tilde{\Pi}}(\lambda) = 1)\right| < \left|\frac{1}{2} + \mathbf{f}(\lambda) - \frac{1}{2}\right| = \mathbf{f}(\lambda),$$

what means that $\mathcal{D}$ efficiently distinguishes output of $\mathrm{RandGen}$ from $\mathrm{TrueRandGen}$, contradicting the pseudo-randomness of $\mathrm{RandGen}$.

□

Lemma 4.6.3 states that the eavesdropper is unable to determine the exact value of any $\mathbf{p}_\ell \in P$ from the sequence $\mathrm{Encrypt}_{\mathbf{r}}(P)$ with a probability significantly higher than a random guess. However, in the case of voice encryption, the attacker is usually satisfied with obtaining the approximation of $\mathbf{p}_\ell$. Lemma 4.6.4 extends the result of Lemma 4.6.3 to a situation where the attacker tries to determine if $\mathbf{p}_\ell$ belongs to some subset $\mathscr{B} \subset \mathscr{C}$ of codewords.

Lemma 4.6.4. *Let* $\mathbf{r} = \mathrm{RandGen}(s)$*, where* $\mathrm{RandGen}$ *is a pseudo-random generator,* s *is chosen uniformly at random from* $\{0,1\}^{l(\lambda)}$*, and* $\mathbf{r} \in \{1,...,M\}^{L(\lambda)}$*. In addition, let* $\mathscr{C}$ *be a spherical commutative group code of order* M *with the initial vector* $\boldsymbol{\sigma} \in S^{2n-1}$ *and associated with a commutative group of orthogonal matrices* $\mathcal{G}$*. Then, for any sequence* $P = (\mathbf{p}_1,...,\mathbf{p}_{L(\lambda)})$ *of codewords* $\mathbf{p}_\ell \in \mathscr{C}$ *chosen independently from* s*, for any* $0 < \ell \leq L(\lambda)$*, and for any PPT adversary* $\mathcal{A}$*, there is a negligible function* $\mathbf{negl}$ *such that:*

$$\Pr(\mathcal{A}(1^\lambda, \mathbf{p}_\ell \in \mathscr{B} \mid \mathrm{Encrypt}_{\mathbf{r}}(P)) = 1) \leq \frac{|\mathscr{B}|}{M} + \mathbf{negl}(\lambda),$$

where $\mathscr{B}$ *is any subset of* $\mathscr{C}$ *with cardinality* $|\mathscr{B}|$*, and the probability is taken over uniform choice of* $s \in \{0,1\}^{l(\lambda)}$ *and the randomness of* $\mathcal{A}$*.*

Proof.

Let us assume a PPT adversary $\mathcal{A}$ who can efficiently estimate $\mathbf{p}_\ell$ for some fixed ℓ, i.e., there exists a non-negligible function $\mathbf{f}$ such that:

$$\Pr(\mathcal{A}(1^\lambda, \mathbf{p}_\ell \in \mathscr{B} \mid \mathrm{Encrypt}_{\mathbf{r}}(P)) = 1) > \frac{|\mathscr{B}|}{M} + \mathbf{f}(\lambda),$$

We will show that we can construct an adversary $\mathcal{A}^*$ who violates the indistiguishability of encryptions of the scheme $\Pi = (\mathrm{RandGen}, \mathrm{Encrypt}, \mathrm{Decrypt})$.

Let $\mathcal{I}_{\mathscr{B}}$, $\mathcal{I}_{\mathscr{C}/\mathscr{B}}$ be the sets of all vectors of codewords in $\mathscr{C}$ of length $L(\lambda)$ whose $\ell - th$ codeword belongs respectively to $\mathscr{B}$ or to $\mathscr{C}/\mathscr{B}$. Then, let us define the behavior of $\mathcal{A}^*$ playing the indistiguishability challenge:

1. $\mathcal{A}^*$ chooses at random P_0 from $\mathcal{I}_{\mathscr{B}}$ and P_1 from $\mathcal{I}_{\mathscr{C}/\mathscr{B}}$.
2. Upon reception of the encrypted vector U, $\mathcal{A}^*$ invokes $\mathcal{A}(1^\lambda, \mathbf{p}_\ell \in \mathscr{B} \mid U)$
3. $\mathcal{A}^*$ forwards the output of $\mathcal{A}$ as b'.

The adversary $\mathcal{A}$ is PPT, hence $\mathcal{A}^*$ is also PPT. In addition, $\mathcal{A}^*$ succeeds if and only if $\mathcal{A}$ outputs b. Therefore, we obtain:

$$\Pr[\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi}(\lambda)=1] = \Pr[\mathcal{A}(1^\lambda, \mathbf{p}_\ell \in \mathscr{B} \mid \mathrm{Encrypt}_{\mathbf{r}}(P_b)) = b] =$$
$$= \frac{1}{2}\Pr_{P_0 \leftarrow \mathcal{I}_{\mathscr{B}}}[\mathcal{A}(1^\lambda, \mathbf{p}_\ell \in \mathscr{B} \mid \mathrm{Encrypt}_{\mathbf{r}}(P_0)) = 1] + \frac{1}{2}\Pr_{P_1 \leftarrow \mathcal{I}_{\mathscr{C}/\mathscr{B}}}[\mathcal{A}(1^\lambda, \mathbf{p}_\ell \in \mathscr{B} \mid \mathrm{Encrypt}_{\mathbf{r}}(P_1)) = 0] >$$
$$> \frac{1}{2}\left(\frac{|\mathscr{B}|}{M} + \mathbf{f}(\lambda)\right) + \frac{1}{2}\left(\frac{M-|\mathscr{B}|}{M} + \mathbf{f}(\lambda)\right) = \frac{1}{2} + \mathbf{f}(\lambda).$$

It follows that $\mathcal{A}^*$ efficiently breaks the scheme Π, contradicting indistinguishability of Π.

□

4.6.5 Decryption

The procedure Decrypt is similar to enciphering with the only difference that we take the transpose of the selected matrices before performing the rotation:

$$\mathrm{Decrypt}_{\mathbf{r}}(V) = (\mathrm{Select}_{\mathcal{G}}(r_1)^T \cdot \mathbf{v}_1,\ \mathrm{Select}_{\mathcal{G}}(r_2)^T \cdot \mathbf{v}_2, ...,\ \mathrm{Select}_{\mathcal{G}}(r_{L(\lambda)})^T \cdot \mathbf{v}_{L(\lambda)}), \quad (4.15)$$

where $V = (\mathbf{v}_1, ..., \mathbf{v}_{L(\lambda)})$ is a sequence of points on the flat torus $\mathrm{T}_{\boldsymbol{\xi}}$ (not necessarily codewords).

Lemma 4.6.5. *Let $\mathbf{v}_1, \mathbf{v}_2 \in \mathrm{T}_{\boldsymbol{\xi}}$. We have $\|\mathbf{v}_1 - \mathbf{v}_2\| = \|\mathrm{Decrypt}_r(\mathbf{v}_1) - \mathrm{Decrypt}_r(\mathbf{v}_2)\|$, where $\|\bullet\|$ denotes the Euclidean metric.*

Proof.

Let $\mathrm{Decrypt}_r(\mathbf{v}_1) = \mathbf{q}_1$ and $\mathrm{Decrypt}_r(\mathbf{v}_2) = \mathbf{q}_2$. There exist $G_d \in \mathcal{G}$ such that $\mathbf{v}_2 = G_d\mathbf{v}_1$ and $\mathbf{q}_2 = G_d\mathbf{q}_1$, and two matrices $G_v, G_q \in \mathcal{G}$ such that $\mathbf{v}_1 = G_v\boldsymbol{\sigma}$ and $\mathbf{q}_1 = G_q\boldsymbol{\sigma}$. We have:

$$\|\mathbf{v}_1 - \mathbf{v}_2\| = \|G_v\boldsymbol{\sigma} - G_dG_v\boldsymbol{\sigma}\| = \|\boldsymbol{\sigma} - G_d\boldsymbol{\sigma}\| = \|G_q\boldsymbol{\sigma} - G_dG_q\boldsymbol{\sigma}\| = \|\mathbf{q}_1 - \mathbf{q}_2\|.$$

□

The deciphering procedure is distance-preserving in the Euclidean metric, as demonstrated by Lemma 4.6.5. However, we are often more concerned about the distance relations resulting from the composition of the procedures Decrypt and Decode.

Let $\mathbf{p}, \mathbf{q} \in \Pi_{i=1}^{n-1}[0, \pi\xi_i) \times [0, 2\pi\xi_n)$ and let $\gamma_{\boldsymbol{\xi},n}(\mathbf{z}) = \gamma_n(\mathbf{z} \odot \boldsymbol{\xi})$. The Euclidean distance $\|\gamma^{-1}_{\boldsymbol{\xi},n}(\mathbf{p}) - \gamma^{-1}_{\boldsymbol{\xi},n}(\mathbf{q})\|$ is bounded by:

$$0 \le \|\gamma^{-1}_{\boldsymbol{\xi},n}(\mathbf{p}) - \gamma^{-1}_{\boldsymbol{\xi},n}(\mathbf{q})\| = \quad (4.16)$$
$$= 2 - 2\sum_{i=1}^{n}\cos(u_i)\cos(v_i)\prod_{j=1}^{i-1}\sin(u_j)\sin(v_j) - 2\prod_{i=1}^{n}\sin(u_i)\sin(v_i) \le \quad (4.17)$$
$$\le \frac{\|\mathbf{p}-\mathbf{q}\|}{\xi_{min}}. \quad (4.18)$$

The above bounds relate the metric in the hyperbox $\Pi_{i=1}^{n-1}[0, \pi\xi_i) \times [0, 2\pi\xi_n)$ with the metric on the hypersphere S^n. We already know from Equation 4.13, that:

$$\frac{2}{\pi}\|\mathbf{p}-\mathbf{q}\| \le \|\Phi_{\boldsymbol{\xi}}(\mathbf{p}) - \Phi_{\boldsymbol{\xi}}(\mathbf{q})\| \le \|\mathbf{p}-\mathbf{q}\|,$$

where $\xi_{min} = \min_{1 \leq i \leq n} \xi_i \neq 0$ and assuming $\|\mathbf{p} - \mathbf{q}\| \leq \xi_{min}$. Thus, the procedure Decode may lead to a distance expansion by a factor up to $\tau = \pi/(2\xi_{min})$:

$$\|\mathrm{Encrypt}_{\mathbf{r}}(\Phi_{\boldsymbol{\xi}}(\mathbf{p})) - \mathrm{Encrypt}_{\mathbf{r}}(\Phi_{\boldsymbol{\xi}}(\mathbf{q}))\| < \delta \implies \|\mathrm{Decode}(\mathbf{p}) - \mathrm{Decode}(\mathbf{q})\| < \frac{\pi}{2\xi_{min}}\delta$$

for $\delta > 0$ not too large. On the other hand, from (4.17) we get that the distance $\|\gamma^{-1}_{\boldsymbol{\xi},n}(\mathbf{u}) - \gamma^{-1}_{\boldsymbol{\xi},n}(\mathbf{v})\|$ goes down to 0 when approaching the poles of the hypersphere S^n. Consequently, we may notice that the vectors near the poles of S^n should be relatively less distorted by a transmission channel than the vectors close to the equator.

4.6.6 Transmission over the Gaussian channel

Spherical commutative group codes described in this chapter can be viewed as equal-energy block codes with constant average power, and used as error-correction codes adapted for transmission over noisy channels [Slepian, 1968]. In this work, we want to investigate the impact of channel noise on reception error and symbol decoding. In contrast to classical digital communications, some detection error is acceptable. In the optimal scenario, a growing noise level should gradually increase the level of detection errors. However, a significant noise may break this desired continuity between transmission and detection errors, severely disrupting the communication.

Let $\mathbf{u} \in \mathscr{C}$ be an encrypted codeword sent over the Gaussian channel. Upon reception, the recipient observes $\hat{\mathbf{v}} = \mathbf{u} + \mathbf{n}$, where $\mathbf{n}$ represents channel noise sampled from Gaussian distribution with zero mean. Provided that the enciphered vector $\mathbf{u}$ can be any codeword of $\mathscr{C}$ with equal probability, the maximum likelihood detector selects the closest codeword in $\mathscr{C}$ in terms of the Euclidean metric. Moreover, due to the uniform geometrical distribution of codewords on the sphere, the error probability of the optimal detector is the same for every sent codeword. This property turns out to be very useful in the context of this work because the deciphering error caused by Gaussian noise is statistically independent of the ciphertext.

Before decryption, the received codeword $\hat{\mathbf{v}} = [\hat{v}_1, ..., \hat{v}_{2n}]$ should be projected to $\mathbf{v} = G\boldsymbol{\sigma}$ on the flat torus $\mathrm{T}_{\boldsymbol{\xi}}$. The rotation matrix G is of the form:

$$\begin{bmatrix} \mathrm{Rot}(\alpha_1) & 0 & ... & 0 \\ 0 & \mathrm{Rot}(\alpha_2) & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & \mathrm{Rot}(\alpha_n) \end{bmatrix}_{2n \times 2n}$$

where $\alpha_1, \alpha_2, ..., \alpha_n$ are some unknown rotation angles. Assuming a Gaussian noise, the vector $\mathbf{v} = [v_1, ..., v_{2n}]^T$ can be found by projecting the coordinates of $\hat{\mathbf{v}}$ onto the respective circles of radius $\sqrt{\sigma^2_{2i-1} + \sigma^2_{2i}}$:

$$[v_{2i-1}, v_{2i}]^T = [\hat{v}_{2i-1}, \hat{v}_{2i}]^T \frac{\sqrt{\sigma^2_{2i-1} + \sigma^2_{2i}}}{\sqrt{\hat{v}^2_{2i-1} + \hat{v}^2_{2i}}}, \; i = 1, ..., n. \tag{4.19}$$

The projection of $\hat{\mathbf{v}}$ onto $\mathbf{v}$ is an additional source of error which is difficult to tackle. When the noise level is small, however, the overall distortion caused by the operation is limited. Another situation may occur, when the distance $\|\hat{\mathbf{v}} - \mathbf{u}\|$ overreaches ξ_{min}. In such a case, $\mathbf{v}$ may be projected on the opposite side of the torus $\mathrm{T}_{\boldsymbol{\xi}}$, leading to a high error as illustrated in Figure 4.5. A possible solution to this problem is to increase the energy of the transmitted vectors.

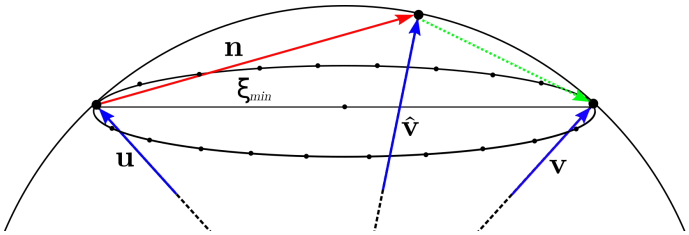


Figure 4.5 – Projection of vectors to the flat torus $\mathrm{T}_{\boldsymbol{\xi}}$ in presence of excessive noise. The initial vector $\mathbf{u} \in \mathscr{C}$ on the orbit of radius ξ_{min} is transmitted over a transmission channel and received as $\hat{\mathbf{v}} = \mathbf{u} + \mathbf{n}$. The vector $\hat{\mathbf{v}}$ is projected to the vector $\mathbf{v}$ on the opposite side of the orbit, far from $\mathbf{u}$.

4.7 Scrambling of image colors

The enciphering scheme described in this chapter is designed for audio-visual data represented using spherical coordinates. This section presents a simple distortion-tolerant encryption model for scrambling colors in an image to illustrate the encoding scheme detailed in the previous sections.

Color of an image pixel is usually represented in the RGB model as a sum of additive primary colors: red, green, and blue. In computer graphics and related domains, alternative representations of the RGB model may be more suitable for encoding and color-related transformations. For example, one may consider a modification of the popular HCL (hue, chroma, lightness) representation [Zeileis et al., 2009]:

$$\mathrm{L} = \max(\mathrm{R},\mathrm{G},\mathrm{B}) + \min(\mathrm{R},\mathrm{G},\mathrm{B}) - 1, \tag{4.20}$$

$$\mathrm{C} = \sqrt{\max(\mathrm{R},\mathrm{G},\mathrm{B})^2 - \min(\mathrm{R},\mathrm{G},\mathrm{B})^2}, \tag{4.21}$$

$$\mathrm{H}' = \begin{cases} 0 & \text{if } \mathrm{C} = 0 \\ \frac{\mathrm{G}-\mathrm{B}}{\mathrm{C}} \mod 6 & \text{if } \max(\mathrm{R},\mathrm{G},\mathrm{B}) = \mathrm{R}, \\ \frac{\mathrm{B}-\mathrm{R}}{\mathrm{C}} + 2 & \text{if } \max(\mathrm{R},\mathrm{G},\mathrm{B}) = \mathrm{G}, \\ \frac{\mathrm{R}-\mathrm{G}}{\mathrm{C}} + 4 & \text{if } \max(\mathrm{R},\mathrm{G},\mathrm{B}) = \mathrm{B}, \end{cases} \tag{4.22}$$

where $\mathrm{R},\mathrm{G},\mathrm{B} \in [0,1]$. The colorspace in the $(\mathrm{L},\mathrm{C},\pi/3\mathrm{H}')$ representation can be illustrated as a 3D ball of radius one centered at the origin (Fig. 4.6). The north and the south poles of the sphere represent respectively white and black, whereas the equator contains all the colors with maximum chroma C. Moreover, as we approach the center of the sphere, the intensity of all the colors fade to grey.

Since the proposed enciphering scheme requires spherical data, the $(\mathrm{L},\mathrm{C},\mathrm{H}')$ representation may be expressed in spherical coordinates:

$$\mathrm{I} = \sqrt{\mathrm{L}^2 + \mathrm{C}^2}, \tag{4.23}$$

$$\Upsilon = \begin{cases} 0 & \text{if } \mathrm{C} = 0, \\ \arctan\left(\frac{\mathrm{L}}{\mathrm{C}}\right) + \frac{\pi}{2} & \text{otherwise}, \end{cases} \tag{4.24}$$

$$\mathrm{H} = \frac{\pi}{3}\mathrm{H}'. \tag{4.25}$$

Parameter I denotes the modulus and $\Upsilon \in [0,\pi]$, $\mathrm{H} \in [0,2\pi]$ are the spherical angles of a color.

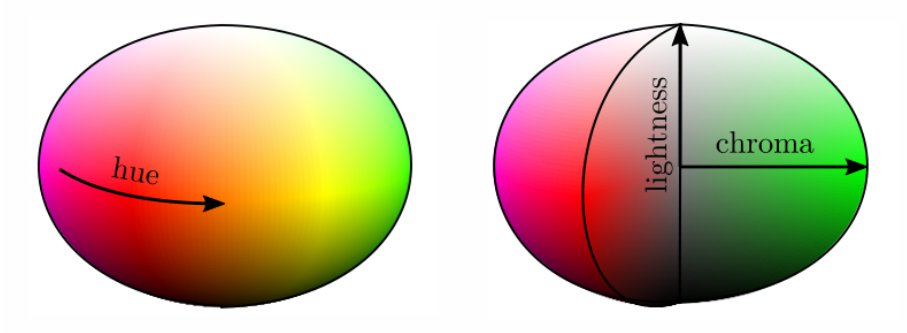


Figure 4.6 – RGB colors in the modified lightness-chroma-hue (LCH) representation.

In the following toy example, we will scramble the angular coordinates (Υ, H) of pixels in an image of size $L = 384 \times 512$ pixels presented in Figure 4.7a, and transmit these enciphered codewords over a Gaussian channel. We will assume the parameter I is sent unencrypted over a secret channel without errors for the simplicity of the example.

Let $X = (\mathbf{x}_1, ..., \mathbf{x}_L)$ be a sequence obtained by reading the image pixels by rows and discarding the modulus I. Pixels are represented in the Cartesian coordinate system by $\mathbf{x}_\ell = [\cos(\Upsilon_\ell), \sin(\Upsilon_\ell)\cos(\mathrm{H}_\ell), \sin(\Upsilon_\ell)\sin(\mathrm{H}_\ell)]^T \in S^2$. The procedure $\mathrm{Encode}(X)$ is done in three steps. Firstly, we extract the angular coordinates of every pixel $\gamma_2(\mathbf{x}_\ell) = [\Upsilon_\ell, \mathrm{H}_\ell]^T$. Then, we quantize the vector $[\Upsilon_\ell, \mathrm{H}_\ell]^T/\sqrt{2}$ with a budget of 31 bits by searching the closest vector in the scaled checkerboard lattice:

$$\Lambda = \frac{2\pi}{2^{15}}\mathrm{D}_2. \tag{4.26}$$

Finally, we map the quantized vectors using a torus map $\Phi_{\boldsymbol{\xi}}$, where $\boldsymbol{\xi} = [1, 1]^T/\sqrt{2}$. We obtain a sequence $P = \mathrm{Encode}(X)$ of codewords from a spherical group code $\mathscr{C} = \mathcal{G}\boldsymbol{\sigma}$ of order 2^{31}, where $\boldsymbol{\sigma} = [1, 0, 1, 0]^T/\sqrt{2}$ is the initial codeword, and $\mathcal{G}$ is a group of rotations associated with the quotient $\Lambda/(2\pi\mathrm{Z}^2)$. The group $\mathcal{G}$ is isomorphic to $\mathbb{Z}_{2^{15}} \oplus \mathbb{Z}_{2^{16}}$ and has two 4×4 generator matrices:

$$G_1 = \begin{bmatrix} \mathrm{Rot}(2\pi/2^{15}) & 0 \\ 0 & \mathrm{Rot}(0) \end{bmatrix} \quad \text{and} \quad G_2 = \begin{bmatrix} \mathrm{Rot}(2\pi/2^{16}) & 0 \\ 0 & \mathrm{Rot}(2\pi/2^{16}) \end{bmatrix}.$$

The sequence $P = [\mathbf{p}_1, ..., \mathbf{p}_L]^T$ is scrambled using two independent generators PRNG_A and PRNG_B with different seeds s_A and s_B, which were instantiated in our toy example by a built-in NumPy [3] random integer sequence generator. The first generator outputs numbers in the range $\{1, ..., 2^{15}\}$ and the second in the range $\{1, ..., 2^{16}\}$, resulting in 31 bits of random data. Given some pairs of random numbers $(r_{A,\ell}, r_{B,\ell})$ produced by the generators, the scrambling procedure is described by $\mathbf{u}_\ell = G_1^{r_{A,\ell}} G_2^{r_{B,\ell}} \mathbf{p}_\ell$. All scrambling operations are summarized in Algorithm 2.

Figure 4.7b depicts the scrambled image obtained by performing the procedure Decode on the enciphered sequence U and restoring the initial modulus I. Despite the seemingly random colors of the pixels, a careful observer may notice some intensity leakage from I, which apparently keeps some shape information in the encrypted image.

3. https://numpy.org/

Figure 4.8 illustrates several images restored from scrambled codewords U distorted by Gaussian noise with signal-to-noise ratio (SNR) equal to 20 dB and 10 dB. The inserted noise introduces granularity into the deciphered image, similar to salt-and-pepper noise or speckle-noise in synthetic aperture radar (SAR) images [Lee, 1981]. Nonetheless, even at high noise intensity, a traditional Korean building, a tree, and three persons shown in the picture are easily recognizable.

(a) Initial image. (b) Scrambled and decoded image.

Figure 4.7 – Scrambling of colors in the image of size 384×512. Pixel colors of the initial image were converted to $(\mathrm{I}, \Upsilon, \mathrm{H})$ representation. In the next step, the angular coordinates (Υ, H) were scrambled using a spherical group code in $\mathbb{R}^4$, and decoded back to the initial domain. The parameter I remained unchanged.

(a) SNR = 20 dB. (b) SNR = 10 dB.

Figure 4.8 – Images restored from scrambled spherical codewords distorted by Gaussian noise.

The quality of a descrambled image is directly related to the error introduced by noise. Let $U = (\mathbf{u}_1, ..., \mathbf{u}_L)$ be a sequence of scrambled spherical codewords representing the angular coordinates (Υ, H) of image pixels, $V = (\mathbf{v}_1, ..., \mathbf{v}_L)$ be a sequence of received spherical points projected onto the flat torus $\mathrm{T}_{\boldsymbol{\xi}}$, and $Y = (\mathbf{y}_1, ..., \mathbf{y}_L)$ be a sequence of descrambled spherical points in $\mathbb{R}^3$. Figure 4.9 displays the errors $\|\mathbf{x}_\ell - \mathbf{y}_\ell\|$ and $\|\mathbf{u}_\ell - \mathbf{v}_\ell\|$ of first 100 codewords, caused by Gaussian noise at SNR = 15 dB. It can be noticed that $\|\mathbf{x}_\ell - \mathbf{y}_\ell\|$ is usually slightly larger than $\|\mathbf{u}_\ell - \mathbf{v}_\ell\|$.

Figure 4.10 displays the sample root mean square error (RMSE) of X and U caused by Gaussian noise at SNR between 5 dB and 25 dB with a 0.5 dB step. The statistics are defined as:

$$\mathrm{RMSE}_X = \sqrt{\frac{\sum_{\ell=1}^{L} \|\mathbf{x}_\ell - \mathbf{y}_\ell\|^2}{L}}, \tag{4.27}$$

$$\mathrm{RMSE}_U = \sqrt{\frac{\sum_{\ell=1}^{L} \|\mathbf{u}_\ell - \mathbf{v}_\ell\|^2}{L}}. \tag{4.28}$$

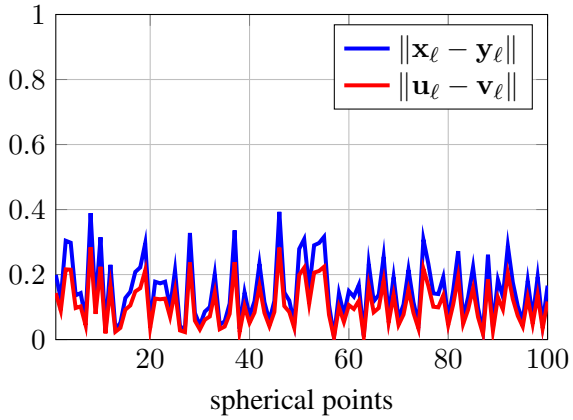


Figure 4.9 – Error between (blue) initial/deciphered and (red) sent/received spherical points. Error caused by Gaussian noise at SNR = 15 dB inserted into enciphered data.

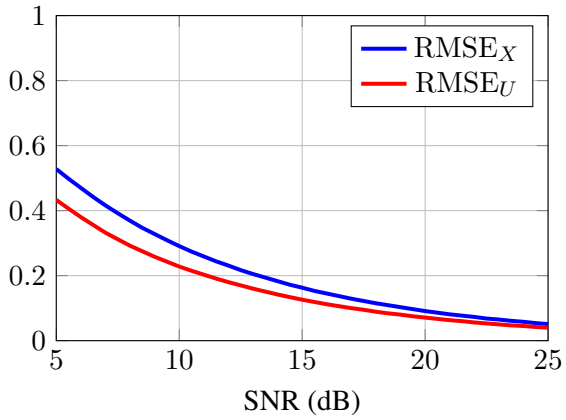


Figure 4.10 – RMSE of received and deciphered spherical data in function of Gaussian noise intensity.

Algorithm 2:

Data: Image with L pixels, random seeds s_A and s_B;
Result: a sequence U of L scrambled codewords;
$U \longleftarrow \emptyset$;
for $\ell \leftarrow 1$ **to** L **do**
  $\mathrm{pixel}_\ell \longleftarrow \mathrm{ReadPixel}(\mathrm{Image})$; `// read a pixel`
  `// discard the modulus` I, `keep` Υ `and` H
  $\mathbf{x}_\ell \longleftarrow \mathrm{ProjectToSphere}(\mathrm{pixel}_\ell)$;
  `// encode to a spherical codeword`
  $\mathbf{p}_\ell \longleftarrow \mathrm{Encode}(\mathbf{x}_\ell)$;
  `// encipher the codeword`
  $(\mathbf{r}_{A,\ell}, \mathbf{r}_{B,\ell}) \longleftarrow (\mathrm{PRNG}_A(s_A), \mathrm{PRNG}_B(s_B))$;
  $\mathbf{u}_\ell \longleftarrow G_1^{r_{A,\ell}} G_2^{r_{B,\ell}} \mathbf{p}_\ell$;
  `// append the enciphered codeword to` U
  $U \longleftarrow \mathrm{append}(U, \mathbf{u}_\ell)$;
end

4.8 Summary

This chapter presented a novel technique for scrambling spherical points on S^n, using a spherical code on a flat torus. Enciphering is done by performing random rotations from a commutative group of $2n \times 2n$ orthogonal matrices isomorphic to quotients of nested lattices with efficient decoding algorithms.

The use of geometrically uniform spherical codes and commutative rotations makes encrypted data particularly suitable for transmission over a communication channel. Moreover, robust deciphering of ciphertexts distorted by the channel may be advantageous when a small inconsistency between the initial and decrypted information is acceptable.

The encryption scheme gives indistinguishable encryptions in the presence of an eavesdropper when rotation matrices are chosen based on the output of a non-binary pseudo-random generator with a fresh secret seed. However, the ciphertexts are malleable by design, and the scheme does not offer any data integrity.

The technique has been illustrated by scrambling colors in an image. The introduction of Gaussian noise with varying intensity to the ciphertext caused a gracefully progressive degradation of the decrypted image quality. Nevertheless, the main content of the image remained well preserved even in the presence of significant noise. These promising results suggest that the presented scheme would be suitable in other audio-visual applications that prioritize information content over quality (i.e., speech intelligibility versus speech quality).

Some further investigation of the encryption scheme would be beneficial. Firstly, the quantization of spherical points on S^n is non-uniform, with the finest resolution near the hypersphere's poles and the coarsest resolution near the equator. This non-uniformity leads to a suboptimal allocation of bits used for the representation of vectors on the hypersphere. Furthermore, a more balanced codeword distribution could reduce the amount of randomness required by the enciphering block and help to predict the impact of channel noise on decryption error.

Another goal to pursue would be the construction of spherical codes with high density and a simple structure in any selected dimensions. The use of dense lattices in the construction does not necessarily imply high density for the associated spherical codes.

Finally, it would be worthwhile to propose some data-integrity mechanism that allows the recipient to detect unacceptable ciphertext modifications from a malicious attacker in the presence of channel distortion. Despite the lack of strong security guarantees, such a mechanism could be useful in real-time or near real-time applications when the attacker has little time to carry any sophisticated attack.

Chapter 5 presents an experimental speech encryption scheme for secure voice communications, which enciphers data using spherical commutative group codes. The scheme scrambles independently three parameters of speech signals: energy, pitch and spectral envelope, and produces a speech-like signal adapted for transmission over voice channels. The system is capable of decrypting vocal parameters distorted by voice channels. The reconstruction of the initial speech is achieved by a vocoder with trained neural networks.

CHAPTER 5

Distortion-tolerant speech encryption

This chapter presents an experimental distortion-tolerant speech encryption scheme for secure voice communications over voice channels that combines the robustness of speech scramblers and a higher security level offered by digital ciphers. The system scrambles vocal parameters of a speech signal (energy, pitch, spectral envelope) using the spherical commutative group codes described in Chapter 4, and outputs a pseudo-speech signal robust against channel distortion or signal compression. Initial speech is reconstructed using a synthesizer based on the LPCNet architecture. The decrypted speech quality depends on channel distortion without experiencing cliff effects with a sudden loss of digital signal reception, which is typical in the digital domain.

The encryption scheme is thoroughly detailed, emphasizing design constraints, operational characteristics, security, robustness against distortion, and computational complexity. The simulations are supported by real-world experiments. The encrypted signal was successfully transmitted over FaceTime between two mobile phones, and a group of about 40 listeners evaluated the perceptual quality and intelligibility of decrypted speech. The results are thoroughly described and analyzed.

Glossary

List of abbreviations

AWGN	Additive White Gaussian Noise
LPC	Linear Prediction Coding
MAC	Message Authentication Code
MFCC	Mel-Frequency Cepstral Coefficients
MNRU	Modulated Noise Reference Unit
MOS	Mean Opinion Score
MUSHRA	MUltiple Stimuli with Hidden Reference and Anchor
PCM	Pulse Coded Modulation
PPT	Probabilistic Polynomial Time
PRNG	Pseudo-Random Number Generator
PSD	Power Spectral Density
RMSE	Root Mean Squared Error
SNR	Signal-to-Noise Ratio
VAD	Voice Activity Detection
VoIP	Voice over Internet Protocol

Notation - spherical codes and lattices

S^n	unit sphere in $\mathbb{R}^{n+1}$ centered at the origin
$\mathscr{C}$	spherical commutative group code (see Section 4.3)
$\boldsymbol{\sigma}$	initial codeword of the code $\mathscr{C}$
$\mathcal{G}$	commutative group of orthogonal matrices
G_i	orthogonal matrix i from $\mathcal{G}$
$\mathrm{T}_{\boldsymbol{\xi}}$	flat torus in $\mathbb{R}^{2n}$ associated with a positive unit vector $\boldsymbol{\xi} \in \mathbb{R}^n$
$\Phi_{\boldsymbol{\xi}}(\bullet)$	torus mapping $\mathbb{R}^n \to \mathbb{R}^{2n}$ associated with $\boldsymbol{\xi} \in \mathbb{R}^n$ (see Section 4.3.2)
$\mathcal{P}_{\boldsymbol{\xi}}$	pre-image of the torus mapping $\Phi_{\boldsymbol{\xi}}$
$\gamma_n(\mathbf{x})$	function $S^n \to \mathbb{R}^n$ which outputs angular coordinates of $\mathbf{x}$
Γ_8	Gosset lattice in $\mathbb{R}^8$

Notation - speech and pseudo-speech

$\varepsilon_{(init)}, \varepsilon_{(dec)}$	initial and deciphered frame energies
$\tilde{\varepsilon}_{(enc)}, \tilde{\varepsilon}_{(rec)}$	enciphered and received frame energies (mark the tilde)
$p_{(init)}, p_{(dec)}$	initial and deciphered pitch periods
$\tilde{p}_{(enc)}, \tilde{p}_{(rec)}$	enciphered and received pitch periods (mark the tilde)
$\mathbf{D}_{(init)}, \mathbf{D}_{(dec)}$	initial and deciphered vectors on S^8 representing spectral envelopes
$\tilde{\mathbf{D}}_{(enc)}, \tilde{\mathbf{D}}_{(rec)}$	enciphered and received vectors on S^{15} representing spectral envelopes
$p_{min},\ p_{max}$	minimum and maximum value of $p_{(init)}$ and $p_{(dec)}$
$\tilde{p}_{min},\ \tilde{p}_{max}$	minimum and maximum value of $\tilde{p}_{(enc)}$ and $\tilde{p}_{(rec)}$
$\varepsilon_{min},\ \varepsilon_{max}$	minimum and maximum value of $\varepsilon_{(init)}$ and $\varepsilon_{(dec)}$
$\tilde{\varepsilon}_{min},\ \tilde{\varepsilon}_{max}$	minimum and maximum value of $\tilde{\varepsilon}_{(enc)}$ and $\tilde{\varepsilon}_{(rec)}$

Notation - enciphering and deciphering

$\mathbf{v}$	scrambling vector of length 10
$\rho_{(init)}, \rho_{(dec)}$	integers representing initial and deciphered frame energies
$\rho_{(enc)}, \rho_{(rec)}$	integers representing enciphered and received frame energies
$\kappa_{(init)}, \kappa_{(dec)}$	integers representing initial and deciphered pitch periods
$\kappa_{(enc)}, \kappa_{(rec)}$	integers representing enciphered and received pitch periods
$\chi_{(init)}, \chi_{(dec)}$	vectors in $\mathbb{R}^8$ representing initial and deciphered spectral envelopes
$\chi_{(enc)}, \chi_{(rec)}$	vectors in $\mathbb{R}^8$ representing enciphered and received spectral envelopes
$\rho_{low},\ \rho_{high}$	bounds of $\rho_{(init)}$ and $\rho_{(dec)}$
$\kappa_{low},\ \kappa_{high}$	bounds of $\kappa_{(init)}$ and $\kappa_{(dec)}$

5.1 Motivation

Early systems for secure voice communication relied on analog signal scrambling in time and frequency domains. Their role was to obscure a conversation by making the speech signal unintelligible for interceptors [Kak, 1983, MacKinnon, 1980]. Although unsecure [Goldburg et al., 1993, Zhao et al., 2007], analog-domain techniques had two crucial advantages over emerging digital systems based on enciphering compressed speech. Firstly, they offered high speech quality at the receiving end, compared with suboptimal low-bitrate digital voice compression and synthesis. Secondly, the scrambled speech signal was exceptionally robust against distortion introduced by telephone lines because transmission noise was linearly added to the reconstructed speech.

In the late 70s and the early 80s, researchers attempted to combine the robustness of analog scrambling and the security offered by rigorous digital enciphering. The result was a new class of transform-domain scramblers that performed digital scrambling of linearly transformed speech samples [Kak, 1983]. Since all transformations done on speech samples were norm-preserving, the noise energy introduced into the ciphertext did not expand after decryption. The first transform-domain scrambler that was exploiting approximately band-limited prolate spheroidal sequences [Slepian and Pollak, 1961] was presented by Wyner [Wyner, 1979, Kaliski, 1984], and inspired new speech scrambling techniques [Lin-Shan Lee et al., 1984, Goldburg et al., 1993].

This chapter presents an experimental joint source-cryptographic enciphering scheme for secure voice communications over voice channels, which to some extent enjoys the similar distortion-tolerant property of speech scramblers. The lossy enciphering unit scrambles the perceptual speech parameters (loudness, pitch, timbre) of a recorded speech signal using the distance-preserving techniques described in Chapter 4, and produces a synthetic signal adapted for transmission over a voice channel. Upon reception, a recipient who owns a valid cryptographic key restores distorted copies of the original speech parameters and decodes the speech signal with the help of a trained neural vocoder.

The system architecture and its operation is thoroughly detailed, emphasizing security aspects, computational complexity, and robustness to distortion. The scheme operates on speech frames and produces an enciphered signal of equal duration, what can be seen as a strong advantage for making the system working real-time. Moreover, it is justified that encrypted speech is computationally indistinguishable from random when enciphering is done using a secure pseudo-random number generator (PRNG) with a secret seed of a sufficient length.

Simulations and real-world experiments follow the system description. Simulations confirmed the scheme's capability to decode mildly distorted signals. Furthermore, the encrypted speech signal was transmitted over FaceTime between two mobile phones and successfully decrypted. A speech quality assessment with about 40 listeners showed that the proposed encryption scheme produces intelligible speech and is robust against Gaussian noise at SNR = 15 dB and voice compression at bitrate 48 kbps with the Opus-Silk speech coder. Finally, the preliminary computational analysis suggests that the optimized system implementation may run on high-end portable devices. The experimental code used in simulations and speech samples evaluated in the speech quality assessment are available online. [1]

This chapter is organized as follows. Section 5.2 introduces and details the speech encryption algorithm. Section 5.3 discusses the system's operation and security, and Section 5.4 presents the evaluation results. Finally, Section 5.5 concludes the work and gives future prospects.

1. https://github.com/PiotrKrasnowski/Speech_Encryption

5.2 Speech encryption scheme

Figure 5.1 illustrates a simplified model of distortion-tolerant speech encryption scheme, consisting of a speech enciphering unit and a complementary deciphering unit. The enciphering unit takes as input a binary key-stream produced by a pseudo-random number generator (PRNG) with a secret seed s of length at least 128 bits, and samples of a narrowband speech signal. In the first processing step, the speech encoder maps 20 ms speech frames indexed by $\ell = 0, 1, 2, ...$ into a sequence of vocal parameters $(\varepsilon_{(init),\ell}, p_{(init),\ell}, \mathbf{D}_{(init),\ell})$, where $\varepsilon_{(init),\ell}$ corresponds to the frame's energy, $p_{(init),\ell}$ is a pitch period, and $\mathbf{D}_{(init),\ell}$ is a vector representing the shape of a spectral envelope.

The encoding process is followed by enciphering using randomness produced by the PRNG. Vocal parameters of every frame are independently scrambled into a new set of parameters $(\tilde{\varepsilon}_{(enc),\ell}, \tilde{p}_{(enc),\ell}, \tilde{\mathbf{D}}_{(enc),\ell})$ defined over a new space of pseudo-speech parameters (tagged by a tilde). Finally, the scrambled sequence is forwarded to the pseudo-speech synthesizer, which produces a harmonic, wideband signal resembling pseudo-speech. The synthetic signal is a concatenation of 25 ms frames with a 5 ms overlap, where every frame carries one set of enciphered parameters. Consequently, the encrypted signal duration is the same as the duration of the initial speech, which is an essential requirement in real-time operation.

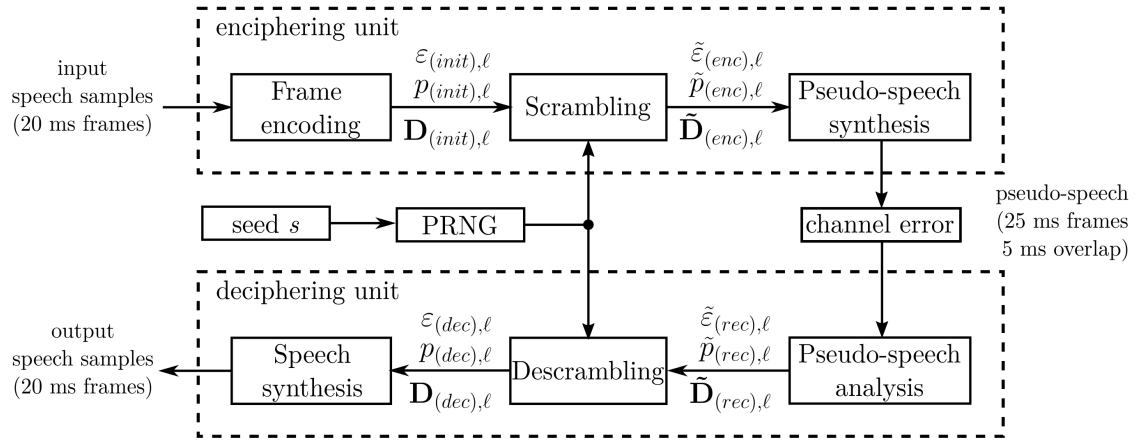


Figure 5.1 – Simplified diagram of the distortion-tolerant speech encryption scheme.

Due to the speech-like properties of the synthetic signal, it can be transmitted over a wideband digital voice channel without much risk of suppression by a Voice Activity Detector (VAD). Upon reception of the signal samples, the paired deciphering unit extracts distorted copies of the sent parameters $(\tilde{\varepsilon}_{(rec),\ell}, \tilde{p}_{(rec),\ell}, \tilde{\mathbf{D}}_{(rec),\ell})$ and performs descrambling using the same binary key-stream produced by its PRNG. In the last step, restored parameters $(\varepsilon_{(dec),\ell}, p_{(dec),\ell}, \mathbf{D}_{(dec),\ell})$ are decoded into narrowband speech, perceptually similar to the input speech signal.

The crucial property of the presented speech encryption system is its ability to descramble enciphered parameters $(\tilde{\varepsilon}_{(rec),\ell}, \tilde{p}_{(rec),\ell}, \tilde{\mathbf{D}}_{(rec),\ell})$ distorted by a voice channel. As the amount of channel distortion goes up, so does the distortion of the resynthesized speech. As a result, we obtain a progressive and controlled speech quality degradation without significant loss of intelligibility. Preservation of intelligibility comes from the remarkable human tolerance to distorted speech signals.

5.2.1 Speech encoding

The speech encoder in the presented encryption scheme is essentially a harmonic speech encoder that models speech signals as a combination of amplitude-modulated harmonics. The perceived fundamental frequency of a harmonic speech signal is usually referred to as *pitch*, perceived signal energy as *loudness*, whereas the spectral envelope is related to *speech timbre*.

The encoder operates sequentially on 20 ms speech frames of 160 samples with a 10 ms look-ahead. Every frame is processed in the same manner, so we skip the frame indexation ℓ for simplicity. A speech frame is firstly pre-emphasized with a first-order filter $I(z) = 1 - 0.85z^{-1}$ to boost high-frequency signal components. It is then encoded into a set of 10 basic parameters: a pitch period and an approximation of the spectral envelope expressed by 9 coefficients. The pitch period expressed in samples per cycle is defined as:

$$p_{(init)} \coloneqq \frac{f_s}{f_0}, \tag{5.1}$$

where f_0 is the estimated fundamental frequency of the harmonic structure of the speech signal with $f_s = 8000$ Hz being the sampling frequency. The spectral envelope is obtained from the Power Spectral Density (PSD) on a moving window of 40 ms of speech signal with 20-ms offset and 50% overlap. The PSD is windowed using 9 mel-scaled triangular filters shown in Figure 5.2, resulting in 9 band-limited energies E_1, ..., E_9 such that their sum is close enough to the frame energy:

$$\varepsilon_{(init)} \coloneqq \sum_{i=1}^{9} \mathrm{E}_i. \tag{5.2}$$

It may be noticed that the vector of square roots of energy coefficients $[\sqrt{\mathrm{E}_1}, ..., \sqrt{\mathrm{E}_9}]^T$ can be seen as a point on the non-negative part of the 9-dimensional hypersphere centered at 0. The radius $\sqrt{\varepsilon_{(init)}}$ of the 8-sphere is related to the frame energy, whereas the normalized vector:

$$\mathbf{D}_{(init)} \coloneqq \left[\sqrt{\mathrm{E}_1/\varepsilon_{(init)}}, ..., \sqrt{\mathrm{E}_9/\varepsilon_{(init)}}\right]^T \tag{5.3}$$

corresponds to the shape of the spectral envelope, i.e., speech timbre. Since a typical spectral envelope consists of about 4 formants [Rabiner and Schafer, 2011], it is a reasonable assumption that $\mathbf{D}_{(init)}$ should capture the most relevant features in the speech spectrum.

The enciphering procedure requires the encoded pitch period and the signal energy to be bounded by some predefined intervals $[p_{min},\ p_{max}]$ and $[\varepsilon_{min},\ \varepsilon_{max}]$. Thus, if $p_{(init)}$ or $\varepsilon_{(init)}$ exceed these intervals, they are thresholded to the closest bound. A selection of bounds is a compromise between the dynamic range required for proper speech representation and its sensitivity to distortion. Moreover, the lower energy bound ε_{min} is slightly larger than 0, meaning that the scheme could be unable to register some very low-amplitude sounds.

5.2.2 Enciphering

A blockwise scrambling is applied on the input parameters $(p_{(init)}, \varepsilon_{(init)}, \mathbf{D}_{(init)})$ defined over the space of speech parameters into a new set $(\tilde{p}_{(enc)}, \tilde{\varepsilon}_{(enc)}, \tilde{\mathbf{D}}_{(enc)})$ defined over the space of pseudo-speech parameters. Each of these parameters is critical for maintaining speech intelligibility [Huang et al., 2001, Chap. 2], and hence contains information that could be exploited by a

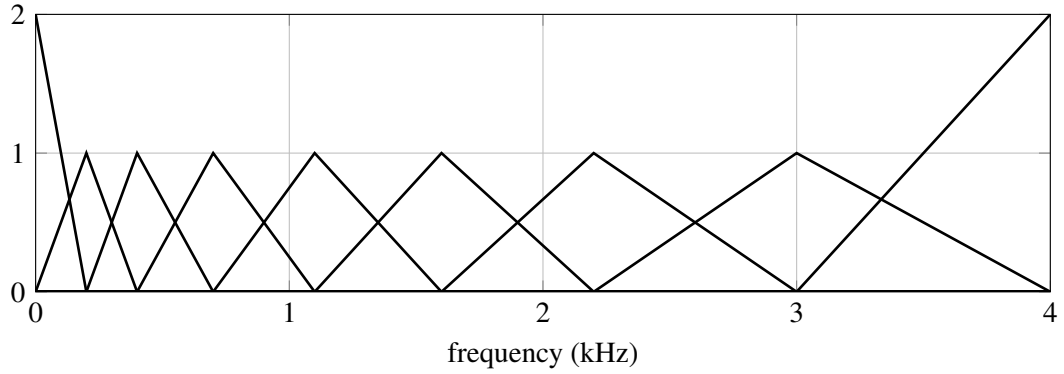


Figure 5.2 – Nine mel-scaled triangular spectral windows used in speech encoding. The amplitude of two side filters is doubled to compensate their missing halves.

cryptanalyst to reconstruct the vocal message. For this reason, we consider them as equally salient. Consequently, $(p_{(init)}, \varepsilon_{(init)}, \mathbf{D}_{(init)})$ are enciphered using a single, shared PRNG.

The enciphering of each frame requires a vector of 10 freshly-generated random integers $\boldsymbol{\nu} = [\nu_1, \ \ldots \ \nu_{10}]^T$, where ν_3 belongs to the additive ring $\mathbb{Z}_{2^{15}}$ with 2^{15} elements, ν_{10} belongs to $\mathbb{Z}_{2^{17}}$, and the remaining coefficients belong to $\mathbb{Z}_{2^{16}}$. These non-uniform ranges of values determine the quantization resolution of the input parameters: $p_{(init)}$ and $\varepsilon_{(init)}$ are quantized using 2^{16} levels, and the vector $\mathbf{D}_{(init)}$ is encoded by one of the 2^{128} possible values. Consequently, we obtain a 16-bit quantization per encoded coefficient, which is a reasonable resolution for encoding vocal parameters. The vector $\boldsymbol{\nu}$ can be efficiently computed from a sequence of 160 bits produced by the PRNG. Given the random bits, the scrambling block splits the binary sequence into chunks of length 15, 16 and 17 bits, and reads them as unsigned integers.

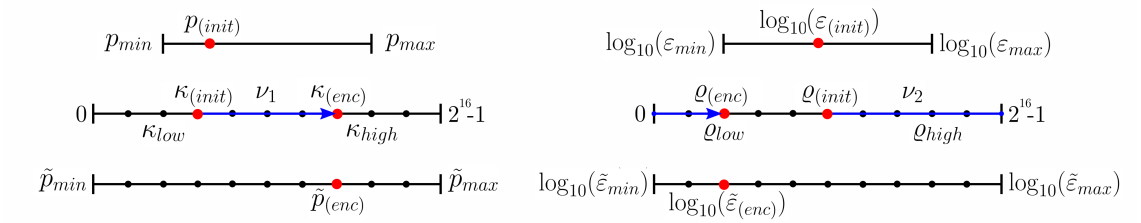


Figure 5.3 – Enciphering of the frame pitch period $p_{(init)}$ and the frame energy $\varepsilon_{(init)}$. From top to bottom: Step 1: $p_{(init)}$ and $\log_{10}(\varepsilon_{(init)})$ are linearly scaled and rounded to $\kappa_{(init)}$ and $\varrho_{(init)}$ in $\mathbb{Z}_{2^{16}}$. Step 2: $\kappa_{(init)}$ and $\varrho_{(init)}$ are translated by random ν_1 and ν_2 over $\mathbb{Z}_{2^{16}}$ to $\kappa_{(enc)}$ and $\varrho_{(enc)}$. Step 3: $\kappa_{(enc)}$ and $\varrho_{(enc)}$ are linearly scaled to $\tilde{\varepsilon}_{(enc)}$ and $\tilde{p}_{(enc)}$ defined over the space of pseudo-speech parameters.

Enciphering of pitch and energy is illustrated in Figure 5.3. The input pitch period $p_{(init)}$ is linearly scaled into an interval $[\kappa_{low}, \kappa_{high}]$ such that $0 < \kappa_{low} < \kappa_{high} < 2^{16} - 1$, and rounded to the closest integer $\kappa_{(init)} \in \mathbb{Z}_{2^{16}}$. Similarly, the frame energy in logarithmic scale $\log_{10}(\varepsilon_{(init)})$ is transformed to $\varrho_{(init)} \in [\varrho_{low}, \varrho_{high}]$. Then, the obtained integers $\kappa_{(init)}$ and $\varrho_{(init)}$ are translated respectively by ν_1 and ν_2 over the additive ring $\mathbb{Z}_{2^{16}}$:

$$\kappa_{(enc)} = (\kappa_{(init)} + \nu_1) \mod 2^{16} \tag{5.4}$$

$$\varrho_{(enc)} = (\varrho_{(init)} + \nu_2) \mod 2^{16}. \tag{5.5}$$

Finally, the enciphered integers $\kappa_{(enc)}$ and $\varrho_{(enc)}$ are linearly scaled to $\tilde{p}_{(enc)} \in [\tilde{p}_{min},\ \tilde{p}_{max}]$ and $\tilde{\varepsilon}_{(enc)} \in [\tilde{\varepsilon}_{min},\ \tilde{\varepsilon}_{max}]$.

Enciphering of $\mathbf{D}_{(init)} \in S^8$ is based on the framework for scrambling spherical data described in Chapter 4. The only difference is that enciphering is performed over the pre-image $\mathcal{P}_{\boldsymbol{\xi}} \subset \mathbb{R}^8$ of the torus mapping $\Phi_{\boldsymbol{\xi}}$, where $\boldsymbol{\xi} = 1/\sqrt{8}[1, ..., 1]^T \in \mathbb{R}^8$. The result of enciphering is mapped to S^{15}, as shown in Figure 5.4.

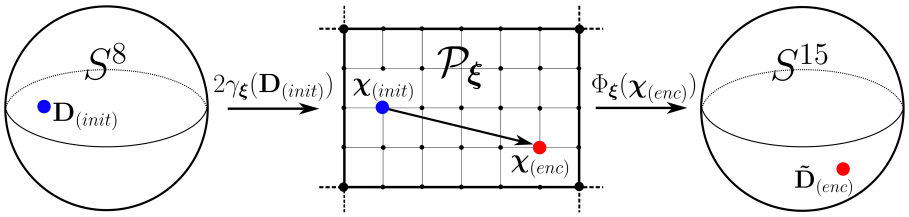


Figure 5.4 – Enciphering of $\mathbf{D}_{(init)}$. The scaled spherical coordinates $2\gamma_{\boldsymbol{\xi}}(\mathbf{D}_{(init)})$ are quantized by searching the closest lattice vector from Λ. The obtained $\boldsymbol{\chi}_{(init)} \in \Lambda$ is randomly translated by a vector from the quotient $\Lambda/(2\pi\mathrm{Z}^8/\sqrt{8})$ over $\mathcal{P}_{\boldsymbol{\xi}}$ and mapped to $\tilde{\mathbf{D}}_{(enc)} = \Phi_{\boldsymbol{\xi}}(\boldsymbol{\chi}_{(enc)})$ on S^{15}.

Let $\gamma_{\boldsymbol{\xi}}(\mathbf{D}_{(init)}) = \boldsymbol{\xi} \odot \gamma_8(\mathbf{D}_{(init)})$ be a vector representing scaled spherical coordinates of the vector $\mathbf{D}_{(init)}$. It may be noticed that $2\gamma_{\boldsymbol{\xi}}(\mathbf{D}_{(init)})$ (note the factor of 2) lies in the pre-image $\mathcal{P}_{\boldsymbol{\xi}}$ of a flat torus $\mathrm{T}_{\boldsymbol{\xi}}$. In the first step, the vector $2\gamma_{\boldsymbol{\xi}}(\mathbf{D}_{(init)})$ is quantized by searching the closest lattice point of a scaled Gosset lattice:

$$\Lambda = \frac{2\pi}{2^{k+1}\sqrt{8}}\Gamma_8, \tag{5.6}$$

where $k = 15$ is the scaling factor. The quotient of nested lattices $\Lambda/(2\pi\mathrm{Z}^8/\sqrt{8})$ is associated through the torus mapping $\Phi_{\boldsymbol{\xi}}$ to a spherical code $\mathscr{C} = \mathcal{G}\boldsymbol{\sigma} \subset S^{15}$, where $\mathcal{G}$ is isomorphic to $\mathbb{Z}_{2^{15}} \oplus \mathbb{Z}^6_{2^{16}} \oplus \mathbb{Z}_{2^{17}}$ and $\boldsymbol{\sigma} = 1/\sqrt{8}[1, 0,\ ...\ 1, 0]^T \in \mathbb{R}^{16}$ is the initial codeword in $\mathscr{C}$.

Let $\boldsymbol{\chi}_{(init)} \in \Lambda$ be the closest lattice vector to $2\gamma_{\boldsymbol{\xi}}(\mathbf{D}_{(init)})$. The vector $\boldsymbol{\chi}_{(init)}$ is translated by a random vector:

$$\boldsymbol{\chi}_{(enc)} = \boldsymbol{\chi}_{(init)} + \nu_3\boldsymbol{\beta}_1 + ... + \nu_{10}\boldsymbol{\beta}_8, \tag{5.7}$$

where $\boldsymbol{\beta}_1$, ..., $\boldsymbol{\beta}_8$ are the column vectors of the generator matrix B_Λ:

$$B_\Lambda = \begin{bmatrix} 2\pi/2^{15} & -2\pi/2^{16} & 0 & 0 & 0 & 0 & 0 & 2\pi/2^{17} \\ 0 & 2\pi/2^{16} & -2\pi/2^{16} & 0 & 0 & 0 & 0 & 2\pi/2^{17} \\ 0 & 0 & 2\pi/2^{16} & -2\pi/2^{16} & 0 & 0 & 0 & 2\pi/2^{17} \\ 0 & 0 & 0 & 2\pi/2^{16} & -2\pi/2^{16} & 0 & 0 & 2\pi/2^{17} \\ 0 & 0 & 0 & 0 & 2\pi/2^{16} & -2\pi/2^{16} & 0 & 2\pi/2^{17} \\ 0 & 0 & 0 & 0 & 0 & 2\pi/2^{16} & -2\pi/2^{16} & 2\pi/2^{17} \\ 0 & 0 & 0 & 0 & 0 & 0 & 2\pi/2^{16} & 2\pi/2^{17} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\pi/2^{17} \end{bmatrix}$$

and $[\nu_3, ..., \nu_{10}]^T \in \mathbb{Z}_{2^{15}} \oplus \mathbb{Z}_{2^{16}}^6 \oplus \mathbb{Z}_{2^{17}}$ is derived from the randomness of the PRNG. Finally, the enciphered vector is mapped to the flat torus $\tilde{\mathbf{D}}_{(enc)} = \Phi_{\boldsymbol{\xi}}(\boldsymbol{\chi}_{(enc)})$.

The secrecy properties of $\mathbf{D}_{(init)}$ enciphering presented in Chapter 4 remain unchanged due to the equivalence between performing rotations on the flat torus $\mathrm{T}_{\boldsymbol{\xi}}$ and translations in the pre-image $\mathcal{P}_{\boldsymbol{\xi}}$, and the isomorphism between the group $\mathcal{G}$ and the quotient of lattices $\Lambda/(2\pi\mathrm{Z}^8/\sqrt{8})$.

5.2.3 Pseudo-speech synthesis

The last stage of speech encryption involves the synthesis of a 'well-formed' audio signal. The role of audio synthesis is to enable an efficient transmission of the enciphered values $(\tilde{p}_{(enc)}, \tilde{\varepsilon}_{(enc)}, \tilde{\mathbf{D}}_{(enc)})$ over a digital voice channel, and to prevent signal blockage by a VAD. Robust operation requires finding a tradeoff between producing a signal sufficiently speech-like yet simple to encode and decode. Furthermore, the encoding procedure should comply with a typical signal distortion characteristic introduced by a particular channel to benefit from distortion-tolerant enciphering.

Since $\tilde{p}_{(enc)}$, $\tilde{\varepsilon}_{(enc)}$ and $\tilde{\mathbf{D}}_{(enc)}$ represent the enciphered pitch period, the energy and the spectral envelope of a speech frame, the natural approach is to relate these values with some homologous parameters of an encrypted signal. Then, a perceptual distortion of the signal would be proportionally mapped to the deciphered speech, to some extent reflecting the quality of the voice channel used for transmission.

Every 25 ms frame of a pseudo-speech signal consists of three segments. The first and the last 5 ms of a frame play the role of guard periods. The remaining 15 ms is where the enciphered parameters are encoded. Once a frame is synthesized, it is windowed by a trapezoidal window and concatenated in an overlap-then-add manner, as illustrated in Figure 5.5.

A 25 ms signal frame $\mathbf{y}_t$ sampled at $f_s = 16$ kHz contains the samples of a harmonic waveform:

$$\mathbf{y}[n] = \sum_{k=1}^{K_{(\omega_0)}} \eta \mathrm{A}_k \cos(k\omega_0/f_s \cdot n + \phi_k - k\omega_0/f_s \cdot 80),\ n = 0, 1, ..., 399, \tag{5.8}$$

where ω_0 is the fundamental frequency, A_k are the amplitudes of harmonics, η is a real-valued energy scaling factor, $(\phi_k - k\omega_0/f_s \cdot 80)$ are the initial phases and $K_{(\omega_0)}$ is the number of harmonics depending on ω_0. Given the harmonicity of $\mathbf{y}_t$, the encoding of $\tilde{p}_{(enc)}$, $\tilde{\varepsilon}_{(enc)}$ and $\tilde{\mathbf{D}}_{(enc)}$ essentially reduces to a careful manipulation of ω_0, A_k and ϕ_k. In addition, only the middle samples $\mathbf{y}[80], ..., \mathbf{y}[319]$ are involved in the encoding process. Once the harmonic parameters of the frame are determined, the remaining part of $\mathbf{y}_t$ is reproduced.

The encoding of enciphered parameters into $\mathbf{y}_t$ is performed sequentially, starting from the pitch, then the shape of the spectral envelope, and finally the energy. The most natural approach for encoding the pitch is to assign $\omega_0 := 2\pi f_s/\tilde{p}_{(enc)}$. Then, one may apply a classical open-loop cross-correlation method [Rabiner and Schafer, 2011, Chap. 10, Gold et al., 2011, Chap. 31] to extract $\tilde{p}_{(enc)}$ back from the received signal. In order to benefit from some off-the-shelf pitch detectors, the value $\tilde{p}_{(enc)}$ should lie within the natural range of pitch values (50 Hz - 300 Hz). Spectral shaping involves finding a proper relation between the amplitudes A_k and the initial phases ϕ_k. Finally, the spectrally shaped frame is scaled to match the desired frame energy $\tilde{\varepsilon}_{(enc)} = \sum_{n=80}^{319} \mathbf{y}^2[n]$.

Compared to encoding the pitch and the energy, mapping the vector $\tilde{\mathbf{D}}_{(enc)}$ of length 16 into the spectrum of $\mathbf{y}_t$ seems to be less straightforward. It may be noticed that the fixed pitch and

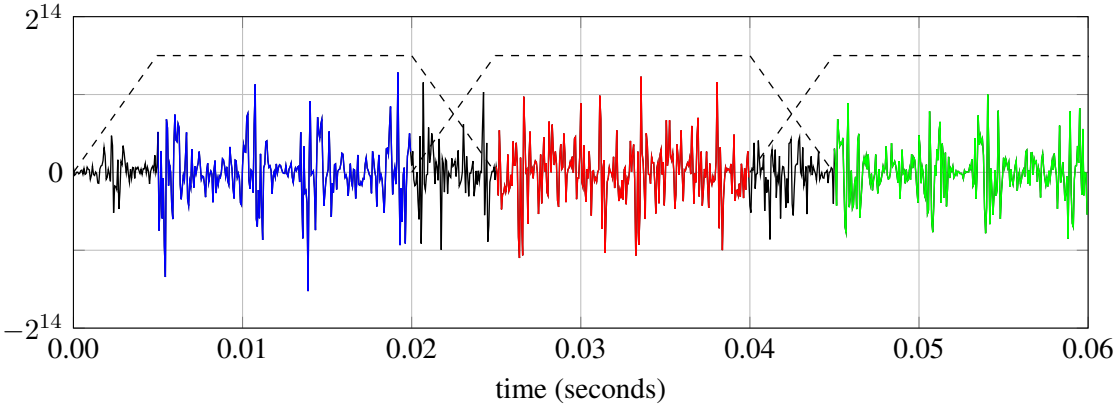


Figure 5.5 – Three 25 ms frames of a pseudo-speech harmonic signal. Every colored portion of the waveform encodes a different set of enciphered parameters $\tilde{p}_{(enc)}$, $\tilde{\varepsilon}_{(enc)}$ and $\tilde{\mathbf{D}}_{(enc)}$. The frames are windowed using a trapezoidal window and overlapped, forming 5 ms guard periods.

energy put many constraints on the spectral envelope of the frame and the values ω_0, A_k, and ϕ_k. Despite this fact, it is desirable to keep the encoding and the decoding procedures of $\tilde{\mathbf{D}}_{(enc)}$ possibly independent from processing the pitch and the energy. Furthermore, it is required that a proper data extraction from $\mathbf{y}_t$ is possible for any combination of $\tilde{p}_{(enc)}$, $\tilde{\varepsilon}_{(enc)}$ and $\tilde{\mathbf{D}}_{(enc)}$.

In order to overcome these limitations, we propose a slightly modified framework for encoding $\tilde{\mathbf{D}}_{(enc)}$. The encoding process relies on a bank of 16 adjacent spectral windows, illustrated in Figure 5.6. The main idea of using these spectral windows is to encode each coordinate of $\tilde{\mathbf{D}}_{(enc)}$ into a frequency band associated to its respective spectral window. Unlike in speech encoding, the windows are square-shaped, and linearly distributed between the 300-6700 Hz range. The proposed selection of spectral windows aims to improve transmission robustness over a voice channel rather than to capture the perceptually relevant spectral features. As a result, the proposed framework is similar to using Frequency Division Multiplexing (FDM) [Weinstein and Ebert, 1971] for mitigating frequency fading.

Another difference is related to how the spectral windows are applied. Instead of windowing the signal PSD as is the case in speech analysis, the windows are directly applied on the Discrete Fourier Transform (DFT) of sampled $\mathbf{y}_t$. As will be explained later in the section, this change significantly simplifies the encoding process. Besides, it seems better suited for data transmission over channels with an additive, independent noise such as AWGN because distortion would linearly map to $\tilde{\mathbf{D}}_{(enc)}$.

Shaping the spectrum of the harmonic frame $\mathbf{y}_t$ can be achieved by a simultaneous manipulation of the amplitudes and the initial phases of the harmonics. Thus, it is advantageous to consider a complex-domain rewriting of $\mathbf{y}_t$, in which the amplitude A_k and the initial phase ϕ_k are merged into a single complex term $\check{\mathrm{A}}_k = \mathrm{A}_k \exp(j\phi_k)$:

$$\mathbf{z}[n] = \sum_{k=1}^{K_{(\omega_0)}} \check{\mathrm{A}}_k \exp(n \cdot jk\omega_0/f_s), \; n = 0, 1, ..., 239. \tag{5.9}$$

The complex samples $\mathbf{z}[0], ..., \mathbf{z}[239]$ correspond to respective samples $\mathbf{y}[80], ..., \mathbf{y}[319]$ that encode the enciphered parameters.

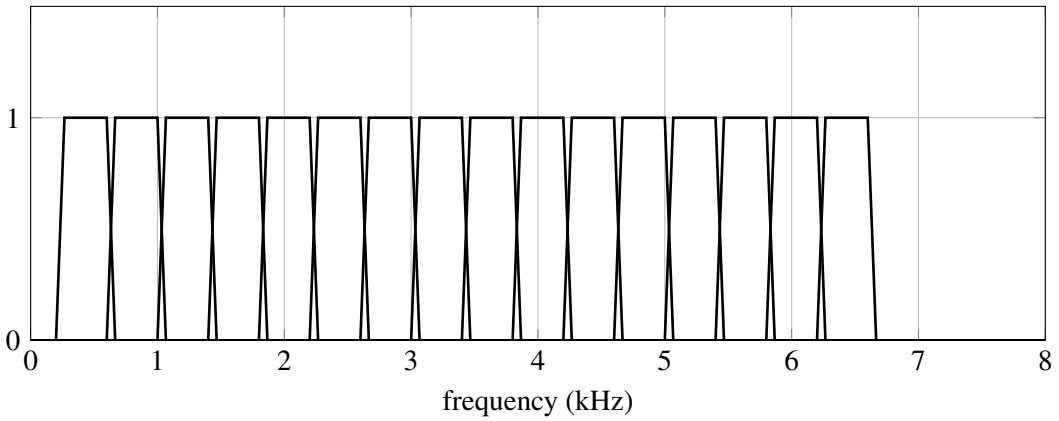


Figure 5.6 – Sixteen square-shaped spectral windows distributed uniformly over 300-6700 Hz.

Let $\mathbf{Z}$ be the column vector representing the DFT of samples $\mathbf{z}[0], ..., \mathbf{z}[239]$. The vector $\mathbf{Z}$ is a sum of harmonic components:

$$\mathbf{Z} = \sum_{k=1}^{K_{(\omega_0)}} \check{\mathrm{A}}_k \mathbf{B}^T_{(\omega_0),k}, \tag{5.10}$$

where $\mathbf{B}_{(\omega_0),k}$ is a row vector representing the DFT of complex sinusoidal samples $\exp(j2\pi n\omega_0/f_s)$ for $n = 0, 1, ..., 239$. The goal of the encoding process is to find $\mathbf{Z} = [\mathrm{Z}_1, ..., \mathrm{Z}_{240}]^T$ such that:

$$\tilde{\mathrm{D}}_{(enc),k} \cdot \mathrm{e}^{j\frac{2\pi}{16}k} = \sum_{n=1}^{240} \mathrm{Z}_n \mathrm{H}_{k,n}, \ k = 1, ..., 16, \tag{5.11}$$

where $\mathbf{H}_k = [\mathrm{H}_{k,1}, ..., \mathrm{H}_{k,240}]$ is a row vector representing the k-th spectral window sampled at frequencies $\frac{n}{240} \cdot f_s$ for $n = 0, 1, ..., 239$. As a result, each element of $\tilde{\mathbf{D}}_{(enc)}$ is represented by a complex sum of windowed DFT samples. The predefined component $\exp(j2k\pi/16)$ is inserted to prevent the result of summation from being purely real and improve the time-domain waveform shape of the synthesized pseudo-speech frame.

The summations in 5.10 and 5.11 can be expressed conveniently in a matrix form:

$$\tilde{\mathbf{D}}_{(enc)} \odot \mathbf{W}_{16} = \mathrm{H}\mathbf{Z} \qquad \text{and} \qquad \mathbf{Z} = \mathrm{B}_{(\omega_0)}\check{\mathbf{A}}, \tag{5.12}$$

where $\mathbf{W}_{16} = [\mathrm{e}^{j\frac{2\pi}{16}1}, \mathrm{e}^{j\frac{2\pi}{16}2}, ..., \mathrm{e}^{j\frac{2\pi}{16}16}]^T$ is the vector of the 16 roots of unity, H is a 16×240 matrix representing 16 spectral windows sampled over the frequency domain of $\mathbf{Z}$, $\mathrm{B}_{(\omega_0)}$ is a $240 \times K_{(\omega_0)}$ matrix with columns $\mathbf{B}_{(\omega_0),k}$, $\check{\mathbf{A}}$ is the column vector of $K_{(\omega_0)}$ complex amplitudes $\check{\mathrm{A}}_k$ as defined by (5.9), and $\odot$ denotes the Hadamard product. As a result, we obtain a simple, linear relation between $\tilde{\mathbf{D}}_{(enc)}$ and the amplitudes of harmonics:

$$\tilde{\mathbf{D}}_{(enc)} \odot \mathbf{W}_{16} = \mathrm{H}\mathrm{B}_{(\omega_0)}\check{\mathbf{A}} \tag{5.13}$$

The problem of finding $\check{\mathbf{A}}$ such that Equation 5.13 holds is under-determined, because $K_{(\omega_0)}$ is larger than 16 by design. Instead, we can compute the least-square solution using the Moore-Penrose pseudo-inverse:

$$\check{\mathbf{A}} = (\mathrm{H}\mathrm{B}_{\omega_0})^{\dagger}(\tilde{\mathbf{D}}_{(enc)} \odot \mathbf{W}_{16}) \tag{5.14}$$

where $(\bullet)^\dagger$ denotes the pseudo-inverse operation. In order to improve computational efficiency, the pseudo-inverse matrix $(\mathrm{HB}_{\omega_0})^\dagger$ can be pre-computed and kept in a look-up table.

The least-square solutions obtained by the Moore-Penrose pseudo-inverse imply that the computed magnitudes $|\breve{\mathrm{A}}_k|$ are small. It has a positive impact on the time-domain waveform shape, minimizing the risk of producing high-amplitude peaks that are likely to be clipped during transmission. Another advantage of the pseudo-inverse is its fast computation, suitable for real-time processing.

Finally, we assign $\mathrm{A}_k \coloneqq |\breve{\mathrm{A}}_k|$ and $\phi_k \coloneqq \mathrm{Arg}(\breve{\mathrm{A}}_k)$, and set the energy scaling factor η in Equation 5.8 to match the desired energy $\tilde{\varepsilon}_{(enc)}$.

The remaining issue is extracting $\tilde{\mathbf{D}}_{(enc)}$ from $\mathbf{y}_t$. The previously described encoding process involved complex samples $\mathbf{z}[0], ..., \mathbf{z}[239]$, where $\mathbf{y}[80+n] = \eta\Re(\mathbf{z}[n])$ and η is the previously computed scaling factor. Let $\mathbf{Y}$ be a column vector representing the DFT of samples $\mathbf{y}[80], ..., \mathbf{y}[319]$ computed over 240 points. From the general properties of Discrete Fourier Transform we have:

$$\mathbf{Y}_n = \frac{\eta}{2}(\mathbf{Z}_n + \bar{\mathbf{Z}}_{241-n}),\ n = 1, ..., 240, \tag{5.15}$$

where $\bar{\mathbf{Z}}_{241-n}$ denotes the complex conjugate of $\mathbf{Z}_{241-n}$. Provided that $\mathbf{Z}$ is a sum of complex sinusoids of frequency no larger than 6700 Hz, the values $\mathbf{Z}_n$ for $n > 120$ are close to zero. As a result, we can approximate the vector $\mathbf{Y}$ as:

$$\mathbf{Y} \approx \begin{cases} \frac{\eta}{2} \cdot \mathbf{Z}_n, & \text{for } n = 1, ..., 120, \\ \frac{\eta}{2} \cdot \bar{\mathbf{Z}}_{241-n}, & \text{for } n = 121, ..., 240. \end{cases} \tag{5.16}$$

Finally, the enciphered vector can be approximately retrieved by taking:

$$\tilde{\mathbf{D}}_{(enc)} \odot \mathbf{W}_{16} \approx \frac{2}{\eta}\mathrm{H}\mathbf{Y}. \tag{5.17}$$

We estimated the root mean squared error (RMSE) of $\tilde{\mathbf{D}}_{(enc)} \odot \mathbf{W}_{16}$ approximations by simulating a sequence of $L = 10000$ pseudo-speech frames from parameters $(\tilde{p}_{(enc)}, \tilde{\varepsilon}_{(enc)}, \tilde{\mathbf{D}}_{(enc)})$ selected randomly in every frame. We used the following formula:

$$\mathrm{RMSE}_{\tilde{\mathbf{D}}} = \sqrt{\frac{\sum_{\ell=1}^{L} \|\tilde{\mathbf{D}}_{(enc),\ell} \odot \mathbf{W}_{16} - 2\eta_\ell^{-1}\mathrm{H}\mathbf{Y}_\ell\|^2}{L}},$$

where $\tilde{\mathbf{D}}_{(enc),\ell}$ is the ℓ-th encoded vector, $\mathbf{Y}_\ell$ is a vector representing the DFT of the ℓ-th produced pseudo-speech frame, and η_ℓ is the respective scaling factor. The obtained error was 0.011, far lower than an anticipated distortion introduced by the voice channel.

5.2.4 Signal transmission and analysis

Successful decoding of a synthetic signal produced by the pseudo-speech synthesizer requires a high-precision, nearly sample-wise synchronization. Consequently, the presented speech encryption scheme is foremost suited for digital data storage and transmission over fully digital voice communication systems like VoIP, in which a high level of synchronization can be maintained. Upon reception, the signal analyzer processes sequentially the received signal frames and retrieves enciphered parameters.

Let $\hat{\mathbf{y}}[0], ..., \hat{\mathbf{y}}[399]$ be the samples of some received pseudo-speech frame $\hat{\mathbf{y}}_t$ and let $\hat{\mathbf{Y}}$ be a column vector of length 240 with the DFT of the sequence $\hat{\mathbf{y}}[80], ..., \hat{\mathbf{y}}[319]$. The received parameters $\tilde{p}_{(rec)}$ and $\tilde{\varepsilon}_{(rec)}$ are defined as:

$$\tilde{p}_{(rec)} \coloneqq \frac{2\pi f_s}{\hat{\omega}_0}, \tag{5.18}$$

$$\tilde{\varepsilon}_{(rec)} \coloneqq \sum_{n=80}^{319} \hat{\mathbf{y}}^2[n], \tag{5.19}$$

where $\hat{\omega}_0$ is the estimated fundamental frequency of the signal frame and $f_s = 16000$ Hz is the sampling frequency. If any of the values $\tilde{p}_{(rec)}$ and $\tilde{\varepsilon}_{(rec)}$ exceed the intervals $[\tilde{p}_{min},\ \tilde{p}_{max}]$ and $[\tilde{\varepsilon}_{min},\ \tilde{\varepsilon}_{max}]$, they are tresholded to the closest bound.

The vector $\tilde{\mathbf{D}}_{(rec)}$ is retrieved from $\hat{\mathbf{Y}}$ in two steps. Firstly, we compute the normalized real-valued sum of the windowed DFT:

$$\hat{\mathbf{D}}_{(rec)} \coloneqq \frac{\Re\left(2H\hat{\mathbf{Y}} \odot \bar{\mathbf{W}}_{16}\right)}{\left\|\Re\left(2H\hat{\mathbf{Y}} \odot \bar{\mathbf{W}}_{16}\right)\right\|}, \tag{5.20}$$

where $\bar{\mathbf{W}}_{16} = [\mathrm{e}^{-j\frac{2\pi}{16}1}, \mathrm{e}^{-j\frac{2\pi}{16}2}, ..., \mathrm{e}^{-j\frac{2\pi}{16}16}]^T$ and $\Re(2H\hat{\mathbf{Y}} \odot \bar{\mathbf{W}}_{16})$ is the real component of $2H\hat{\mathbf{Y}} \odot \bar{\mathbf{W}}_{16}$. Then, the vector $\hat{\mathbf{D}}_{(rec)}$ is projected to $\tilde{\mathbf{D}}_{(rec)}$ on the flat torus $\mathrm{T}_{\boldsymbol{\xi}}$ using Formula 4.19 in Section 4.6.6.

5.2.5 Deciphering

Given the set of received parameters $(\tilde{p}_{(rec)}, \tilde{\varepsilon}_{(rec)}, \tilde{\mathbf{D}}_{(rec)})$, the descrambling algorithm reverses enciphering operations using the same vector $\boldsymbol{\nu} = [\nu_1, ..., \nu_{10}]^T$ of random integers produced by the PRNG. The values $\tilde{p}_{(rec)}$ and $\log_{10}(\tilde{\varepsilon}_{(rec)})$ are firstly linearly scaled to $\kappa_{(rec)}$ and $\varrho_{(rec)}$ over the interval $[0, 2^{16} - 1]$. Unlike in the enciphering stage, these values are not quantized. In the next step, $\kappa_{(rec)}$ and $\varrho_{(rec)}$ are deciphered by respective translations $-\nu_1$ and $-\nu_2$ modulo 2^{16}:

$$\kappa_{(dec)} = (\kappa_{(rec)} - \nu_1) \mod 2^{16} \tag{5.21}$$

$$\varrho_{(dec)} = (\varrho_{(rec)} - \nu_2) \mod 2^{16}, \tag{5.22}$$

where $\nu_1, \nu_2 \in \mathbb{Z}_{2^{16}}$ are obtained from the PRNG. If the values $\kappa_{(dec)}$ and $\varrho_{(dec)}$ exceed the respective intervals $[\kappa_{low}, \kappa_{high}]$ and $[\varrho_{low}, \varrho_{high}]$, they are tresholded to the closest bound. In the last step, the values are transformed back into the intervals $[p_{min},\ p_{max}]$ and $[\log_{10}(\varepsilon_{min}),\ \log_{10}(\varepsilon_{max})]$ representing the domain of speech parameters.

The deciphering of the unit vector $\tilde{\mathbf{D}}_{(rec)}$ is done by translating $\boldsymbol{\chi}_{(rec)} = \Phi_{\boldsymbol{\xi}}^{-1}(\tilde{\mathbf{D}}_{(rec)})$:

$$\boldsymbol{\chi}_{(dec)} = \boldsymbol{\chi}_{(rec)} - \nu_3\boldsymbol{\beta}_1 - ... - \nu_{10}\boldsymbol{\beta}_8 \mod \frac{2\pi}{\sqrt{8}}, \tag{5.23}$$

where $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_8$ are the columns of B_Λ, $[\nu_3, ..., \nu_{10}]^T$ is a random vector obtained from the PRNG, and the modulo operation is done element-wise. Let $\boldsymbol{\varphi}_{(dec)} = \sqrt{8}\boldsymbol{\chi}_{(dec)}/2$. If any of the angles is $\boldsymbol{\varphi}_{(dec)} = [\varphi_{(dec),1}, ..., \varphi_{(dec),7}]$ is larger than $\pi/2$, it is replaced by $\varphi_{(dec),i} \coloneqq \pi/2 - \varphi_{(dec),i}$. The deciphered spectral envelope vector is $\mathbf{D}_{(dec)} = \gamma_8^{-1}(\boldsymbol{\varphi}_{(dec)})$.

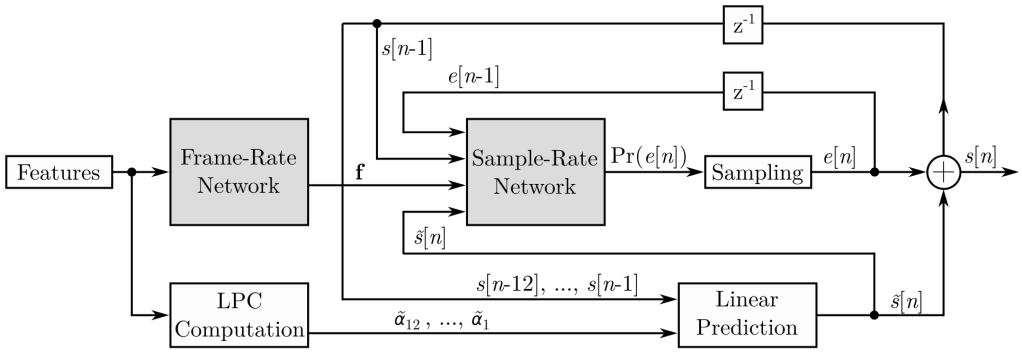


Figure 5.7 – Overview of the narrowband LPCNet architecture. The symbol z^{-1} denotes a one-sample delay.

5.2.6 Speech resynthesis

The output of the descrambling process is a sequence $(p_{(dec),\ell}, \varepsilon_{(dec),\ell}, \mathbf{D}_{(dec),\ell})$ representing harmonic parameters of 20 ms speech frames. The final speech resynthesis is possible with any adapted speech synthesizer supporting harmonic speech parametrization. An example of a suitable narrowband harmonic speech synthesizer is Codec2. [2]

Unfortunately, although parametric sinusoidal speech coders succeed in producing intelligible speech, they often struggle to maintain satisfactory speech quality. In this work, we improve upon harmonic speech synthesis by using our own modification of the LPCNet, a Machine Learning (ML) based synthesizer introduced by Jean-Marc Valin (Mozilla) and Jan Skoglund (Google LLC) [Valin and Skoglund, 2019]. Unlike the original LPCNet which operates on wideband speech signals sampled at 16 kHz, our modified synthesizer produces narrowband speech signal sampled at 8 kHz. We also modified the input of the LPCNet to make it compatible with the vocal parameters that we use in our speech encryption algorithm. Finally, we extended the training procedure of neural networks in order to improve the robustness of the LPCNet against distortion introduced by a voice channel.

The narrowband LPCNet recreates the samples of a speech signal $s[n]$ from a sum of the linear prediction $\tilde{s}[n]$ and the excitation $e[n]$:

$$s[n] = \tilde{s}[n] + e[n] \tag{5.24}$$

$$\tilde{s}[n] = \sum_{k=1}^{12} \alpha_k s[n-k], \tag{5.25}$$

where α_1, ..., α_{12} are the 12-th order linear prediction coefficients (LPC) for the current frame. The excitation samples $e[n]$ are produced by two concatenated neural networks that model an excitation signal from the input vocal parameters.

Figure 5.7 depicts a simplified diagram of the modified narrowband LPCNet algorithm. The speech synthesizer combines two recurrent neural networks: a frame-rate network processing 20 ms speech frames (160 samples) and a sample-rate network operating at 8 kHz. Network architectures are presented in Figure 5.8. The frame-rate network takes as input the sequence of feature

2. https://rowetel.com

vectors computed from $(p_{(dec),\ell}, \varepsilon_{(dec),\ell}, \mathbf{D}_{(dec),\ell})$ and produces a sequence of frame-rate conditioning vectors $\mathbf{f}_\ell$ of length 128. Vectors $\mathbf{f}_\ell$ are sequentially forwarded to the sample-rate network and padded with last value to get a frame with 160 samples.

The role of the sample-rate network is to predict the multinomial probability distribution of the current excitation sample $\Pr(e[n])$, given the current conditioning vector $\mathbf{f}_\ell$, the previous signal sample $s[n-1]$, the previous excitation sample $e[n-1]$ and the current prediction $\tilde{s}[n]$. The current excitation sample $e[n]$ is obtained by randomly generating a single sample from $\Pr(e[n])$. The synthesis output of the narrowband LPCNet are pre-emphasized speech samples $s[n] = \tilde{s}[n]+e[n]$, filtered with a de-emphasis filter $J(z) = \frac{1}{1-0.85z^{-1}}$. The operation of the narrowband LPCNet algorithm stops when the last feature vector is processed, and the sample-rate network synthesizes the last speech frame.

Computing a feature vector from a set of vocal parameters $(p_{(dec),\ell}, \varepsilon_{(dec),\ell}, \mathbf{D}_{(dec),\ell})$ requires few steps. Let $\mathbf{E}_\ell = \varepsilon_{(dec),\ell} \cdot \mathbf{D}_{(dec),\ell} \odot \mathbf{D}_{(dec),\ell}$ be a vector representing 9 band-limited energies of the ℓ-th encoded speech frame. Then, the ℓ-th feature vector has the form $[\mathrm{C}_{\ell,0}, \mathrm{C}_{\ell,1}, ..., \mathrm{C}_{\ell,8}, \rho_\ell]^T$, where $\mathrm{C}_{\ell,0}$, $\mathrm{C}_{\ell,1}$, ..., $\mathrm{C}_{\ell,8}$ is the discrete cosine transform (DCT-II) of the sequence $\log_{10}(\mathrm{E}_{\ell,1})$, ..., $\log_{10}(\mathrm{E}_{\ell,9})$, and where $\rho_\ell = (p_{(dec),\ell} - 100)/50$ is the scaled pitch period. Taking into consideration the mel-scaled distribution of spectral windows used in the speech encoder, the coefficients $\mathrm{C}_{\ell,0}$, $\mathrm{C}_{\ell,1}$, ..., $\mathrm{C}_{\ell,8}$ can be viewed as 9-band Mel-Frequency Cepstral Coefficients (MFCC).

The prediction samples $\tilde{s}[n]$ are computed from the predictor coefficients $\tilde{\alpha}_1$, ..., $\tilde{\alpha}_{12}$ obtained from $\mathbf{E}_\ell$ and updated for every frame. The mel-scaled windowed energies in $\mathbf{E}_\ell$ are firstly interpolated into a linear-frequency PSD and then converted to an autocorrelation using an inverse FFT. The LPC coefficients are obtained from the autocorrelation using the Levinson-Durbin algorithm [Makhoul, 1975].

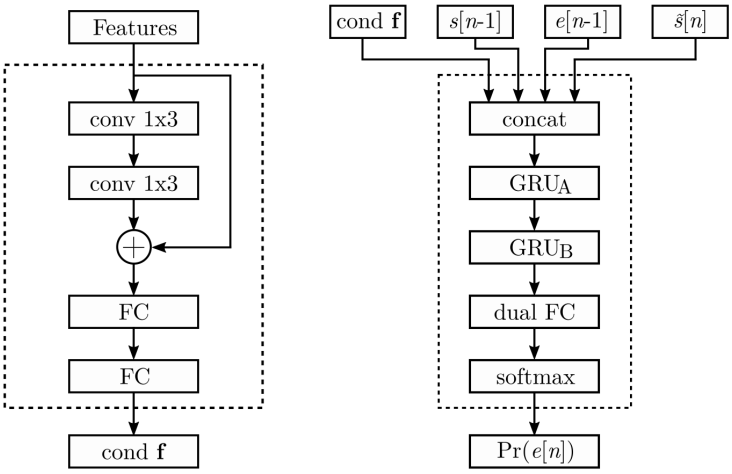


Figure 5.8 – Architectures of the frame-rate (left) and the sample-rate (right) networks. The frame network consists of two convolutional layers with a filter of size 3, followed by two fully-connected layers. The output of the convolutional layers is added to an input connection. The sample-rate network firstly concatenates four inputs and passes the resulted combination to two gated recurrent units ($\mathrm{GRU_A}$ of size 384 and $\mathrm{GRU_B}$ of size 16), followed by a dual fully connected layer [Valin and Skoglund, 2019]. The output of the last layer is used with a soft-max activation.

Obtaining $\tilde{\alpha}_k$ from the low-resolution bands is different than in the classical approach, in which the autocorrelation and the predictor α_k are computed directly from speech samples [Rabiner and Schafer, 2011, Chap. 9]. Figure 5.9 displays a frequency response of two predictors obtained from the same pre-emphasized speech frame using both estimation methods. Despite a substantial difference between the responses, the sample-rate network in the narrowband LPCNet learns to compensate, as pointed out in [Valin and Skoglund, 2019].

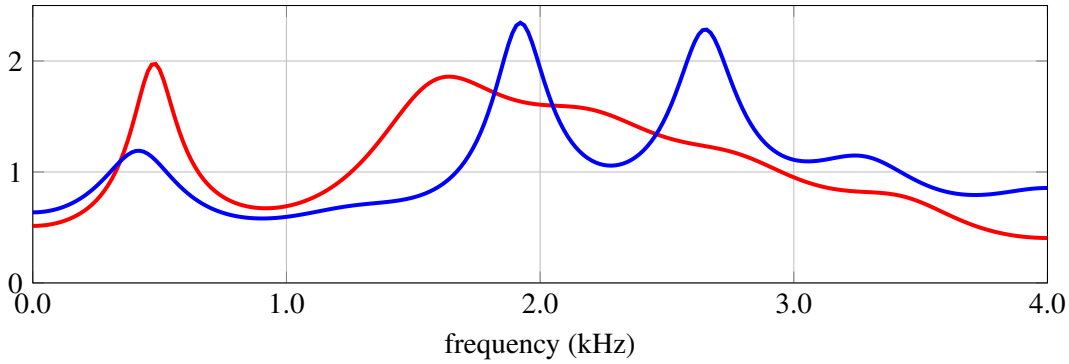


Figure 5.9 – Frequency response of a 12-th order predictor α_k derived from the autocorrelation of 160 pre-emphasized speech samples (blue line) and a frequency response of a 12-th order predictor $\tilde{\alpha}_k$ derived from the mel-scaled energy bands of the same speech frame by the narrowband LPCNet (red line). Despite the significant difference between the two spectra, the sample-rate network of the narrowband LPCNet is trained to compensate the difference without degrading the speech quality.

5.3 Discussion

This section discusses several aspects associated with system security, tolerance to channel distortion, selection of system parameters, and the training the neural networks in the speech synthesizer.

5.3.1 Security considerations

The security of the proposed speech encryption scheme cannot be rigorously proved without an in-depth specification, which may significantly differ in particular implementations. Instead, we provide an informal justification for the asymptotic indistinguishability of encryptions in an experiment comparable with the classical adversarial indistinguishability challenge presented in Definition 4.4.3 in Section 4.4. The encryptions are indistinguishable when a secure binary pseudo-random number generator with a fresh seed taken uniformly at random is adequately implemented.

Let L and l be polynomials, λ be an integer-valued security parameter, $\mathbf{x}_t$ be an arbitrary speech signal of finite duration $t \in [0, 320L(\lambda))$, $\mathrm{RandGen}$ be a binary pseudo-random number generator and $s \in \{0,1\}^{l(\lambda)}$ be a random seed. In addition, let $\Pi = (\mathrm{RandGen}, \mathrm{Enc}, \mathrm{Dec})$ be the speech encryption scheme described in Section 5.2, where Enc takes as an input the speech signal $\mathbf{x}_t$ and a vector $\mathbf{r} \leftarrow \mathrm{RandGen}(s)$ such that $\mathbf{r} \in \{0,1\}^{160L(\lambda)}$, and outputs a synthetic signal $\mathbf{y}_t = \mathrm{Enc}_{\mathbf{r}}(\mathbf{x}_t)$ of the same duration as $\mathbf{x}_t$. Furthermore, let us define an adversarial indistinguishability challenge $\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi}(\lambda)$:

Definition 5.3.1. The adversarial indistinguishability challenge $\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi}(\lambda)$ is defined as:

1. The adversary $\mathcal{A}$ is given input 1^λ, and he chooses a pair of distinct signals $\mathbf{x}_{0,t}$, $\mathbf{x}_{1,t}$ of finite duration $t \in [0, 320L(\lambda))$.
2. A random seed s is chosen and a sequence $\mathbf{r}$ is generated by running $\mathrm{RandGen}(s)$. The challenge $\mathbf{y}_t = \mathrm{Enc}_{\mathbf{r}}(\mathbf{x}_{b,t})$ is given to $\mathcal{A}$, where $b \in \{0, 1\}$ is chosen uniformly at random.
3. $\mathcal{A}$ outputs bit b'.
4. The output of the challenge is 1 if $b = b'$ and 0 otherwise.

Below we present a proposition for the indistinguishability of encryptions in the presence of an eavesdropper and provide the sketch of the proof.

Proposition 5.3.1. *Let* $\Pi = (\mathrm{RandGen}, \mathrm{Enc}, \mathrm{Dec})$ *be the speech encryption scheme described in Section 5.2. There is a negligible function* $\mathbf{negl}$ *such that for any PPT adversary* $\mathcal{A}$ *it holds:*

$$\mathrm{Adv}(\mathcal{A}(1^\lambda)) = \left|\Pr(\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi}(\lambda) = 1) - \frac{1}{2}\right| \leq \mathbf{negl}(\lambda).$$

Sketch of Proof.

Firstly, we can observe that the security of the speech encryption scheme does depend neither on the speech analysis nor the pseudo-speech synthesis algorithms. Indeed, the single output of the speech encoder is a sequence of parameters $\{(\varepsilon_{(init),\ell}, p_{(init),\ell}, \mathbf{D}_{(init),\ell})\}_{\ell=1}^{L(\lambda)}$, which is forwarded to the scrambling block. The result of enciphering is a new sequence $\{(\tilde{\varepsilon}_{(enc),\ell}, \tilde{p}_{(enc),\ell}, \tilde{\mathbf{D}}_{(enc),\ell})\}_{\ell=1}^{L(\lambda)}$, being the single input of the pseudo-speech synthesizer. In consequence, the indistinguishability of the synthetic pseudo-speech signal $\mathbf{y}_t$ reduces to the indistinguishability of the enciphered sequence from any sequence taken uniformly at random.

The enciphering of the initial speech parameters is done using a sequence of scrambling vectors $\{\boldsymbol{\nu}_\ell\}_{\ell=1}^{L(\lambda)}$, where $\boldsymbol{\nu}_\ell \in \mathbb{Z}_{16}^2 \oplus \mathbb{Z}_{15} \oplus \mathbb{Z}_{16}^6 \oplus \mathbb{Z}_{17}$ and $\{\mathbf{v}_\ell\}_{\ell=1}^{L(\lambda)}$ produced from $\mathbf{r}$ by sequentially reading short bitstrings as unsigned integers. We can easily show that if the binary pseudo-random generator $\mathrm{RandGen}$ is secure, then the resulting sequence $\{\mathbf{v}_\ell\}_{\ell=1}^{L(\lambda)}$ is indistinguishable from any sequence $\{\mathbf{v}^*_\ell\}_{\ell=1}^{L(\lambda)}$ produced from a random binary sequence $\mathbf{r}^* \in \{0, 1\}^{160L(\lambda)}$ output by the true entropy collector $\mathrm{TrueRandGen}$.

The rest of the proof essentially repeats the reasoning from Lemma 4.6.2 and Lemma 4.6.3. Let $\tilde{\Pi} = (\mathrm{TrueRandGen}, \mathrm{Enc}, \mathrm{Dec})$ be a new encryption scheme. In a first step, we may prove that the result of enciphering $\{(\tilde{\varepsilon}^*_{(enc),\ell}, \tilde{p}^*_{(enc),\ell}, \tilde{\mathbf{D}}^*_{(enc),\ell})\}_{\ell=1}^{L(\lambda)}$ with a random sequence $\{\boldsymbol{\nu}^*_\ell\}_{\ell=1}^{L(\lambda)}$ obtained from $\mathrm{TrueRandGen}$ is perfectly secure, i.e., the enciphered values $\{\tilde{\varepsilon}^*_{(enc),1}, \tilde{p}^*_{(enc),1}, \tilde{\mathbf{D}}^*_{(enc),1}, \tilde{\varepsilon}^*_{(enc),2}, ..., \tilde{\mathbf{D}}^*_{(enc),L(\lambda)}\}$ are generated independently from uniform distributions over their respective discrete domains. Consequently, we get:

$$\Pr(\mathrm{PrivK}^{eav}_{\mathcal{A},\tilde{\Pi}}(\lambda) = 1) = \frac{1}{2}.$$

Then, we can show the indistinguishability of the sequence $\{(\tilde{\varepsilon}_{(enc),\ell}, \tilde{p}_{(enc),\ell}, \tilde{\mathbf{D}}_{(enc),\ell})\}_{\ell=1}^{L(\lambda)}$ by contradiction: the existence of a PPT adversary $\mathcal{A}$ who distinguishes the sequence from purely random with a non-negligible advantage implies the existence of a PPT distinguisher $\mathcal{D}$ breaking the security of $\mathrm{RandGen}$. From this, we conclude that there is a negligible function $\mathbf{negl}$ such that the advantage of any PPT adversary $\mathcal{A}$ participating in the experiment $\mathrm{PrivK}^{eav}_{\mathcal{A},\Pi}$ is at most $\mathrm{Adv}(\mathcal{A}(1^\lambda)) < 0.5 + \mathbf{negl}(\lambda)$. □

Proposition 5.3.1 states that the encryption scheme produces indistinguishable encryptions in the presence of an eavesdropper, provided that every speech signal is enciphered using a secure pseudo-random bit generator with a fresh and uniformly distributed random seed. However, selecting a proper binary pseudo-random generator is far from being trivial and should be done very carefully. In particular, it is not evident if a 'good' bit generator can always be adequately transformed into a non-binary generator and vice-versa. For instance, in [Baigneres et al., 2007] it is shown that a poorly designed non-binary sequence expanded into a bitstream could pass a randomness test by some bit-oriented distinguishers. Some statistical test suitable for checking the randomness of non-binary ciphers can be found in [Baigneres et al., 2007, Epishkina, 2018]

The selection of a suitable pseudo-random number generator is out of the scope of this work. Nevertheless, some promising candidates of binary pseudo-random number generators can be found in the NIST Special Publication 800-90A [Barker et al., 2015]. Crucially, the presented generators are evaluated for their potential use as non-binary number generators over integer rings $\mathbb{Z}_{2^n}$, $n \in \mathbb{N}$. An example of such a generator uses Advanced Encryption Standard (AES) in the CTR mode of operation and a secret 256-bit seed. The generator is claimed to securely produce up to 2^{48} bitstrings of length 2^{19} if the input seed is taken uniformly at random. Furthermore, the input seed is updated after every request for backtracking resistance. The maximum bitstring length 2^{19} in a unique request is sufficient to encipher more than one minute of one-way voice communication. Finally, a parallelization of bitstring generation provided by the CTR mode is an advantage in real-time operation.

An obvious weakness of the presented scheme is the lack of mechanisms providing data integrity. Since the enciphered speech signal does not include any side information, the recipient cannot verify the source and received data correctness. Moreover, it is not clear whether a reliable data integrity mechanism even exists in this lossy framework, given that the received signal is likely to differ from the initial signal and that malleability-by-design is one of the basic features of the presented speech encryption scheme. Instead, it is important to ensure the proper authentication of the users and secure exchange of cryptographic keys (or secret seeds) before the session starts [Canetti and Krawczyk, 2001, Katz and Lindell, 2015, Chap. 10]. Some solutions include mutual authentication using public certificates or symmetric pre-shared keys [Rescorla, 2018, Barker, 2020]. This problem is covered in more detail in Chapter 6.

Despite the absence of data integrity in real-time communication, an adversarial manipulation on encrypted speech giving a meaningful deciphered speech is technically challenging. Synthetic signal fragility and high synchronization requirements between the legitimate users suggest that the attacker is more likely to interrupt the communication. However, such an interruption is effectively not much different from signal blockage by a VAD.

If the enciphered speech signal is stored, a binary representation of the signal in PCM or a compressed form should be accompanied by a message authentication code (MAC) [Katz and Lindell, 2015, Chap. 5] computed with a dedicated authentication key.

5.3.2 Tolerance to signal distortion and large deciphering errors

Let $\mathbf{y}_t$ be an encrypted speech signal sent over a voice channel, and $\hat{\mathbf{y}}_t$ be the signal received by the recipient. Due to channel distortion, parameters $(\tilde{\varepsilon}_{(rec),\ell}, \tilde{p}_{(rec),\ell}, \tilde{\mathbf{D}}_{(rec),\ell})$ extracted from $\hat{\mathbf{y}}_t$ usually diverge from the enciphered sequence $(\tilde{\varepsilon}_{(enc),\ell}, \tilde{p}_{(enc),\ell}, \tilde{\mathbf{D}}_{(enc),\ell})$. The transmission error propagates during descrambling, causing a deciphering error between the initial $(\varepsilon_{(init),\ell}, p_{(init),\ell}, \mathbf{D}_{(init),\ell})$ and deciphered $(\varepsilon_{(dec),\ell}, p_{(dec),\ell}, \mathbf{D}_{(dec),\ell})$ values.

When the distortion is low, transmission and deciphering errors are related by inequalities:

$$\left|\log_{10}\left(\frac{\varepsilon_{(init),\ell}}{\varepsilon_{(dec),\ell}}\right)\right| \leq \left(\frac{2^{16}}{\varrho_{high}-\varrho_{low}}\frac{\log_{10}(\varepsilon_{max}/\varepsilon_{min})}{\log_{10}(\tilde{\varepsilon}_{max}/\tilde{\varepsilon}_{min})}\right)\cdot\left|\log_{10}\left(\frac{\tilde{\varepsilon}_{(enc),\ell}}{\tilde{\varepsilon}_{(rec),\ell}}\right)\right| \tag{5.26}$$

$$\left|p_{(init),\ell}-p_{(dec),\ell}\right| \leq \left(\frac{2^{16}}{\kappa_{high}-\kappa_{low}}\frac{p_{max}-p_{min}}{\tilde{p}_{max}-\tilde{p}_{min}}\right)\cdot\left|\tilde{p}_{(enc),\ell}-\tilde{p}_{(rec),\ell}\right| \tag{5.27}$$

$$\left\|\mathbf{D}_{(init),\ell}-\mathbf{D}_{(dec),\ell}\right\| \leq \frac{\pi}{\sqrt{2}}\cdot\left\|\tilde{\mathbf{D}}_{(enc),\ell}-\tilde{\mathbf{D}}_{(rec),\ell}\right\|, \tag{5.28}$$

where $|\bullet|$ is the modulus and $\|\bullet\|$ denotes the Euclidean norm. In consequence, the deciphering procedure is distortion-tolerant with respect to parameters $\log_{10}(\tilde{\varepsilon}_{(enc),\ell})$, $\tilde{p}_{(enc),\ell}$ and $\tilde{\mathbf{D}}_{(enc),\ell}$, with three independent expansion factors.

The distortion-tolerant property with respect to pseudo-speech parameters holds unless the amount of distortion in the received signal $\hat{\mathbf{y}}_t$ becomes too large. When the values $|\log_{10}(\tilde{\varepsilon}_{(enc),\ell}/\tilde{\varepsilon}_{(rec),\ell})|$, $|\tilde{p}_{(enc),\ell}-\tilde{p}_{(rec),\ell}|$ and $\|\tilde{\mathbf{D}}_{(enc),\ell}-\tilde{\mathbf{D}}_{(rec),\ell}\|$ exceed some specific thresholds, there is a risk of a deciphering error much larger than indicated by the bounds. These large deciphering errors are perceived by the listener as unpleasant flutter degrading the overall perceived speech quality, and should be avoided.

A strong perceptual speech degradation is usually related to large deciphering errors of energy or pitch. In the example depicted in Figure 5.10, a silent speech frame with the energy $\varepsilon_{(init)}=\varepsilon_{min}$ is enciphered to $\tilde{\varepsilon}_{(enc)}$ and sent over a noisy channel in a form of a pseudo-speech frame. Upon reception, the recipient observes $\tilde{\varepsilon}_{(rec)}$ such that $|\log_{10}(\tilde{\varepsilon}_{(rec)}/\tilde{\varepsilon}_{(enc)})|>\varrho_{low}/2^{16}$. However, the result of deciphering is $\varepsilon_{(dec)}=\varepsilon_{max}$, the exact opposite of the initial value.

It may be noticed that deciphering error making a silent frame maximally loud is more damaging for perceptual quality than suppressing a loud frame into silence. A varying perceptual impact of deciphering errors is the main justification for fine-tuning the guard bounds for pitch and energy. Nevertheless, in order to maintain a robust operation of the enciphering scheme, it is important to ensure experimentally that the values $|\log_{10}(\tilde{\varepsilon}_{(rec),\ell}/\tilde{\varepsilon}_{(enc),\ell})|$ and $|\tilde{p}_{(rec),\ell}-\tilde{p}_{(enc),\ell}|$ stay within the guard limits with the high probability.

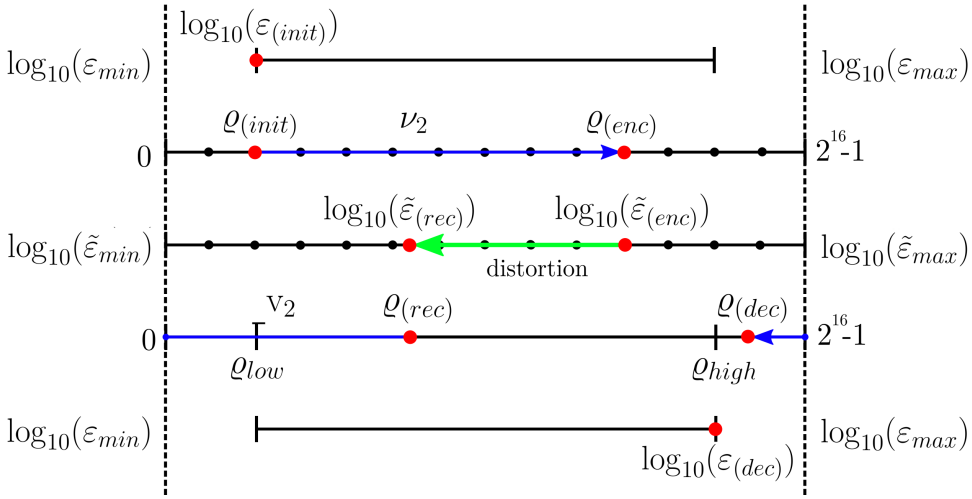


Figure 5.10 – Large deciphering error of energy due to excessive distortion.

Another kind of a deciphering error occurs while processing the spectral envelope of a pseudo-speech frame $\tilde{\mathbf{D}}_{(enc),\ell}$. As mentioned in Chapter 4, channel distortion may cause the received $\hat{\mathbf{D}}_{(enc),\ell} \in S^{15}$ to move away from the flat torus $\mathrm{T}_{\boldsymbol{\xi}}$. When the distance from the flat torus overreaches

$$2\sin\left(\frac{\sin^{-1}\left(1/\sqrt{8}\right)}{2}\right), \tag{5.29}$$

the vector $\hat{\mathbf{D}}_{(enc),\ell}$ could be projected to $\tilde{\mathbf{D}}_{(rec),\ell}$ on the opposite side of the torus. Figure 5.11 illustrates a simplified scenario of a wrong projection in $\mathbb{R}^4$. The projection of the vector $\tilde{\mathbf{D}}_{(enc)}$ to $\tilde{\mathbf{D}}_{(rec)}$ along one dimension of the torus can be viewed as translation of the corresponding coordinate of $\boldsymbol{\chi}_{(enc)} = \Phi_{\boldsymbol{\xi}}^{-1}(\tilde{\mathbf{D}}_{(enc)})$ in the pre-image of the torus. A wrong projection causes an unpredictable change in the spectral envelope of the deciphered frame.

When the channel distortion is sub-proportional to the logarithm of signal energy, the risk of a projection going on the wrong side of the torus can be mitigated by increasing the minimum pseudo-speech frame energy $\tilde{\varepsilon}_{min}$. It is because the norm in the denominator of Equation 5.20 goes up when $\tilde{\varepsilon}_{min}$ is increased, making the error $\|\tilde{\mathbf{D}}_{(enc),\ell} - \tilde{\mathbf{D}}_{(rec),\ell}\|$ relatively smaller.

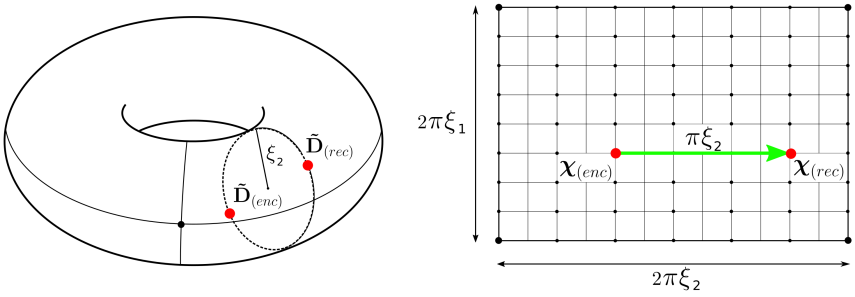


Figure 5.11 – Projection of $\tilde{\mathbf{D}}_{(enc)}$ to the opposite side of the flat torus $\mathrm{T}_{[\xi_1,\xi_2]}$ along single dimension, seen as translation by $\pi\xi_2$ in the pre-image of the torus.

5.3.3 Selection of bounds for the signal parameters

A selection of good bounds for the speech parameters $[p_{min},\ p_{max}]$, $[\varepsilon_{min},\ \varepsilon_{max}]$, pseudo-speech parameters $[\tilde{p}_{min},\ \tilde{p}_{max}]$, $[\tilde{\varepsilon}_{min},\ \tilde{\varepsilon}_{max}]$, and guards $[\kappa_{low},\ \kappa_{high}]$, $[\varrho_{low},\ \varrho_{high}]$ is a tradeoff between a sufficient dynamic range for the encoding and a good robustness to channel distortion. When channel distortion is insignificant, the risk of large deciphering errors is nearly negligible. Then, in compliance with Inequalities 5.26-5.28, it is advantageous to enlarge $[\tilde{p}_{min},\ \tilde{p}_{max}]$ and $[\tilde{\varepsilon}_{min},\ \tilde{\varepsilon}_{max}]$, and reduce the guard regions. As the distortion goes up, the guard regions and $\tilde{\varepsilon}_{min}$ should be adequately increased, and the intervals $[p_{min},\ p_{max}]$, $[\varepsilon_{min},\ \varepsilon_{max}]$ slightly limited. This can be done adaptively depending on fluctuating channel characteristics.

Another factor is pitch detection accuracy in the pseudo-speech analyzer. Voice-oriented pitch estimators analyze the signal assuming small pitch variation over time [Szczerba and Czyzewski, 2005, Rabiner and Schafer, 2011, Chapter 10]. However, the assumption is not valid in an encrypted signal for which the pitch period changes randomly every 20 ms.

The two most common types of pitch estimation errors in noisy signals are transient errors, and octave errors [Beauchamp et al., 1993]. A transient error occurs when an abrupt change of

fundamental frequency within a speech frame violates the stationarity assumption. An octave error describes a situation when the predictor incorrectly outputs a multiple $k\omega_0$ or a fraction $1/k\omega_0$ ($k \in \mathbb{N}$) of the correct fundamental frequency ω_0. These errors are mitigated by pitch tracking [Benesty et al., 2008, Chap. 10]. However, since the pitch period in the encrypted signal is uncorrelated in time, pitch tracking seems redundant if not harmful in our case as it would smooth the received values. Instead, it is essential to maintain frame synchronization and ensure that neither the adjacent frames nor the guard periods damage the pitch estimation.

Octave errors usually exceed guard intervals, leading to performance loss. Thus, it may be worth selecting the limits $[\tilde{p}_{min},\ \tilde{p}_{max}]$ such that $\tilde{p}_{max} \leq 2\tilde{p}_{min}$. For example, $[\tilde{p}_{min},\ \tilde{p}_{max}] = [80,\ 160]$ ($[\tilde{\omega}_{min},\ \tilde{\omega}_{max}] = [100\text{ Hz},\ 200\text{ Hz}]$) seems to be a reasonable tradeoff between the range of possible pitch values and the robustness against octave errors.

5.3.4 The narrowband LPCNet training data

The quality of synthesized speech strongly depends on the capability of the narrowband LPCNet algorithm to operate in more imperfect conditions than during the training [Valin and Skoglund, 2019]. As suggested in [Oord et al., 2016], it is possible to improve the robustness of the network by adding noise during the training stage.

In our speech encryption scheme, there exist two independent sources of imperfections. The first source is the lower quality of real-world speech recordings taken for encryption, and the second source is channel distortion. Motivated by this fact, the training process of the narrowband LPCNet was divided into two stages. During the two-step training, the ML networks consecutively learn to cope with the non-idealities of speech signals and the transmission channel. Splitting the training overcomes several typical problems with learning convergence as when the network cannot effectively compensate for both kinds of distortion at the same time. Moreover, it seems to be more practical if one considers re-training the network to different channel conditions.

The first training stage is identical to the training process described in [Valin and Skoglund, 2019]. During the training, the network learns to predict the excitation sequence $e[n]$, using as input the previous excitation samples $e[n-1]$, the previous signal samples $s[n-1]$, the current prediction samples $\tilde{s}[n]$ and the frame-rate speech features (9-band Bark-scale cepstral coefficients, pitch). A diagram for producing the training data is shown in Figure 5.12. Except for the frame features, the input data is μ-law quantized by the Q block [ITU-T, 1988f]. The input noise is injected into the speech signal in the μ-law domain to make it proportional to signal amplitude. Noise distribution varies across the training data from no noise to a uniform distribution in the [-3,3] range. This injected noise results in a -10 dB to 30 dB SNR in the speech signals used for training. It can be noticed that the injected noise propagates to all sample-rate input data and thus prevents the undesirable situation when the LPCNet models noise with the same shape as the LPC synthesis filter. Therefore, the training process with injected noise helps the network learn to insert a proper dither noise into a synthesized signal [Jayant and Rabiner, 1972, Zorila et al., 2012].

After the first training stage, the network can produce intelligible speech signals from noiseless feature vectors. However, the output of the speech encryption scheme is likely to be distorted. The second stage of the training simulates a scenario when the frame-rate features are transmitted in encrypted form over a voice channel. The input speech signal is fully encrypted with a given random sequence to $\mathbf{y}_t$, as illustrated in Figure 5.13. The injected distortion simulates a typical error introduced by a particular voice channel (i.e., Gaussian noise or speech compression). Finally, a distorted signal $\hat{\mathbf{y}}_t$ is decrypted, and distorted parameters are fed into the network.

The major issue with the distortion injection model in Figure 5.13 is its computational cost associated with processing the training data. Since channel distortion is independent of the input signal, and the reception error linearly propagates to deciphered values, it is possible to inject distortion directly into the speech parameters, as in Figure 5.14. The distribution of injected distortion simulates channel distortion statistics obtained by simulations or measurements.

The two-stage training of the networks on a GPU card Nvidia Quadro RTX 4000 and using one hour of training speech takes approximately five days. Furthermore, we experimented with the speech quality produced by the synthesizer trained to English or Japanese [3] language. The results obtained suggest that the synthesizer should be trained to the language used later for secure communication.

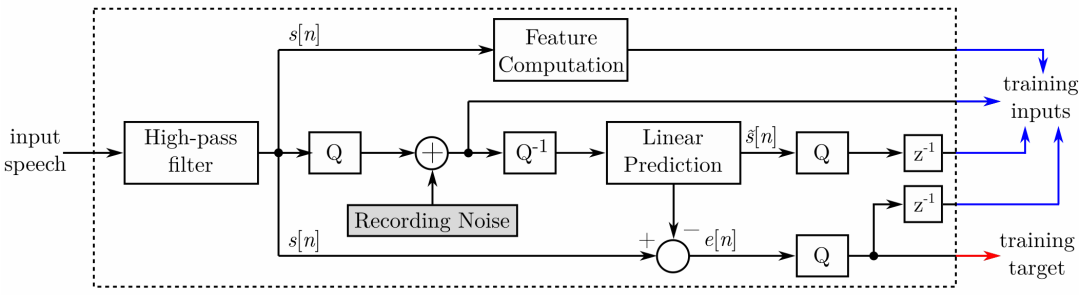


Figure 5.12 – Computing the training data with noise injection in the first training stage to simulate a noisy recording. The Q block denotes μ-law quantization and Q^{-1} denotes conversion from μ-law to the linear domain. The prediction filter $\tilde{s}[n] = \sum_{k=1}^{12} \tilde{a}_k z^{-k}$ is applied to the noisy and quantized input. The excitation samples $e[n]$ are the difference between the clear speech samples $s[n]$ and the predicted ones. The symbol z^{-1} denotes a one-sample delay.

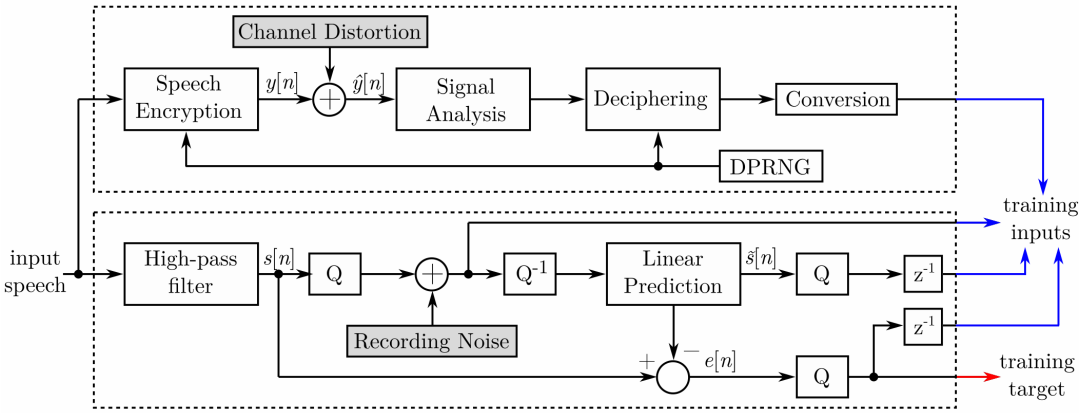


Figure 5.13 – Computing the training data with noise injection in the second training stage to simulate a voice channel with distortion (on the top) and a noisy reception (at the bottom). The diagram simulates encryption of speech parameters with the given random sequence and transmission over a voice channel with predefined distortion. The symbol z^{-1} denotes a one-sample delay.

3. Japanese has a very simple phonology, that makes it particularly useful for experimenting with ML techniques.

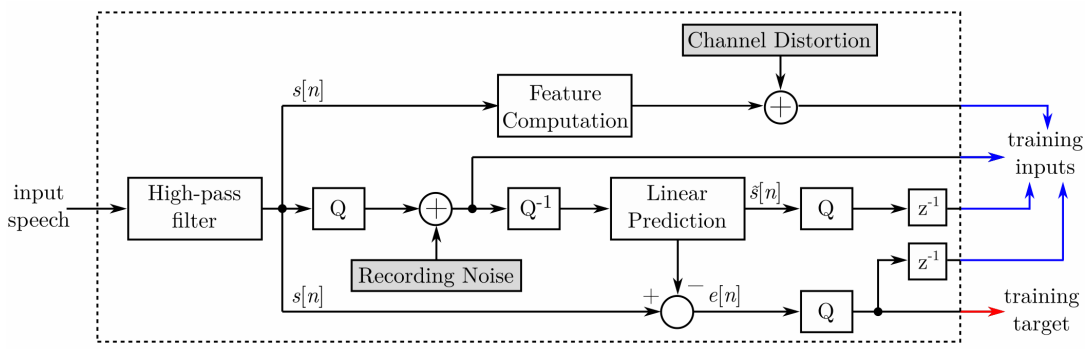


Figure 5.14 – Equivalent diagram for computing the training data with noise injection in the second training stage. The second distortion simulating a voice channel is injected directly into speech parameters. The distribution of distortion simulates distortion introduced by a transmission channel and can be obtained by simulations or experiments. The symbol z^{-1} denotes a one-sample delay.

5.4 Evaluation

This section presents the test results from some experiments with our speech enciphering algorithm. The tests verify the scheme's capability to decrypt distorted pseudo-speech signals. Furthermore, the section investigates a scenario when the receiver is not fully synchronized in time and amplitude with the sender. The simulations are validated by real-world experiments.

Based on some measurements of a signal distortion introduced by FaceTime and Skype in the audio mode, we estimated the SNR in a typical VoIP-based voice channel to be between 10 dB and 15 dB. On the other hand, similar experiments with 3G networks revealed that signal distortion in cellular networks is much higher, and gives SNR values closer to 3-5 dB. Due to excessive distortion in cellular networks and erratic speech quality, we decided to evaluate our encryption scheme for its compatibility with VoIP-based applications. The robustness of deciphering was evaluated by inserting additive white Gaussian noise (AWGN) into an encrypted signal or compressing the encrypted signal with Opus-Silk 1.3.1 [Valin et al., 2012]. Opus-Silk was chosen for experimentation because, unlike AMR or Speex, its compression rate can be easily adjusted.

The precomputed encrypted signal was successfully sent over FaceTime between two iPhones 6 running iOS 12 connected to the same domestic WiFi network and decrypted offline. The use of FaceTime on WiFi is justified by high connection stability (limited drop-outs, constant delay) which greatly simplifies signal synchronization at the receiving end. Additionally, the selected speech excerpts reconstructed from encrypted signals were evaluated in a speech quality/intelligibility assessment on a large group of about 40 participants.

The section concludes with computational analysis in Section 5.4.4. The system's computational complexity was estimated by measuring all floating-point operations performed during running our experimental software. The measurements suggest that the computationally optimized encryption algorithm may operate in real-time on high-end portable devices.

Selected initial, encrypted, distorted, and decrypted speech samples are available online. [4]

4. https://github.com/PiotrKrasnowski/Speech_Encryption

5.4.1 Experimental setup

Tables 5.1 and 5.2 present the encoding parameters of speech and pseudo-speech signals. The intervals $[\varepsilon_{min},\ \varepsilon_{max}]$ and $[p_{min},\ p_{max}]$ were obtained from the TSP English speech corpus. [5] The selection of the intervals $[\tilde{\varepsilon}_{min},\ \tilde{\varepsilon}_{max}]$, $[\tilde{p}_{min}, \tilde{p}_{max}]$ and the bounds $[\kappa_{low},\ \kappa_{high}]$, $[\varrho_{low},\ \varrho_{high}]$ was done based on simulations.

The speech encryption and decryption algorithms were implemented mainly in Python. The speech encoder and the speech synthesizer were obtained from the LPCNet repository [6] and adapted to the scheme. The pitch prediction with tracking for speech was based on open-loop cross-correlation search [Rabiner and Schafer, 2011, Chapter 10], whereas prediction for pseudo-speech relies on a more accurate maximum-likelihood estimator [7] without tracking [Nielsen et al., 2017, Nielsen et al., 2014]. In the simulations, the enciphering stage takes as input a given pseudo-random bitstring produced by a built-in NumPy [8] PCG-64 generator, a 128-bit implementation of Melissa O'Neill's permutation congruential algorithm [O'Neill, 2014].

Table 5.1 – Parameters used for speech encoding and synthesis.

Parameter	Value
frame length	20 ms
sampling frequency	8 kHz
sample representation	int16
energy bounds	$(\varepsilon_{min},\ \varepsilon_{max}) = (10,\ 10^8)$
pitch period bounds	$(p_{min},\ p_{max}) = (16,\ 128)$
energy guard bounds	$(\varrho_{low},\ \varrho_{high}) = (2^{13},\ 2^{16} - 2^{13} - 1)$
pitch guard bounds	$(\kappa_{low},\ \kappa_{high}) = (2^{13},\ 2^{16} - 2^{13} - 1)$

Table 5.2 – Parameters used for pseudo-speech encoding and synthesis.

Parameter	Value
frame length	25 ms
guard period	5 ms
sampling frequency	16 kHz
sample representation	int16
energy bounds	$(\tilde{\varepsilon}_{min},\ \tilde{\varepsilon}_{max}) = (10^9,\ 10^{10})$
pitch period bounds	$(\tilde{p}_{min},\ \tilde{p}_{max}) = (80,\ 160)$

The narrowband LPCNet was trained in two steps on one hour of speech from the multi-speaker TSP English corpus (12 male and 12 female speakers). In the second step of the training, inserted distortion simulated a white Gaussian noise at SNR = 20 dB. Each network was trained for 100 epochs per training step, with a batch consisting of 64 speech sequences of 300 ms. The training was performed on a GPU card Nvidia Quadro RTX 4000 with Keras [9] and Tensorflow [10] using the CuDNN GRU implementation. The selected optimization method was AMSGrad [Reddi et al., 2018] with a step size $\alpha = \frac{\alpha_0}{1+\delta \cdot b}$, where $\alpha_0 = 0.001$, $\delta = 5 \times 10^{-5}$ and b is the batch number.

5. https://www-mmsp.ece.mcgill.ca/Documents/Data/
6. https://github.com/mozilla/LPCNet/
7. https://github.com/jkjaer/fastF0Nls/
8. https://numpy.org/
9. https://keras.io/
10. https://www.tensorflow.org/

5.4.2 Simulations

The first experiment tested the encryption and decryption operations, assuming noise-less transmission. In the example in Figure 5.15, the time-domain envelopes of the initial and the reconstructed speech sentence are very similar. A high degree of similarity can also be observed in the spectrograms presented in Figure 5.16. It may be noticed that the trained speech synthesizer faithfully reconstructs the fundamental frequency and the formants of the initial speech. On the other hand, the encrypted signal in the time and the frequency domains resembles band-limited noise.

Adding distortion into the encrypted signal degrades the decrypted speech. The time-domain envelope of the decrypted speech sentence in Figure 5.17 is still similar to the initial speech but not identical anymore. It may be observed that pseudo-speech decryption has a denoising effect on low-amplitude speech and silence.

The reception and deciphering errors of the same speech sentence are depicted in Figure 5.18. As can be seen, the errors on energy and timbre are non-negligible. However, in contrast to the error $|\varrho_{(init),\ell} - \varrho_{(dec),\ell}|$, the impact of the error $\|\mathbf{D}_{(init),\ell} - \mathbf{D}_{(dec),\ell}\|$ on decrypted speech perception is more unpredictable. Unlike energy and timbre, pitch is very well preserved.

The scheme's robustness has been tested against AWGN at SNR between 5-25 dB and Opus-Silk v1.3.1 compression at bitrates between 28-64 kbps. In each case, the error of received and deciphered parameters were expressed in terms of the RMSE defined as:

$$\mathrm{RMSE}_{\tilde{\varepsilon},(rec)} = \sqrt{\frac{\sum_{\ell=1}^{L} |\varrho_{(enc),\ell} - \varrho_{(rec),\ell}|^2}{L}}, \qquad \mathrm{RMSE}_{\varepsilon,(dec)} = \sqrt{\frac{\sum_{\ell=1}^{L} |\varrho_{(init),\ell} - \varrho_{(dec),\ell}|^2}{L}},$$

$$\mathrm{RMSE}_{\tilde{p},(rec)} = \sqrt{\frac{\sum_{\ell=1}^{L} |\kappa_{(enc),\ell} - \kappa_{(rec),\ell}|^2}{L}}, \qquad \mathrm{RMSE}_{p,(dec)} = \sqrt{\frac{\sum_{\ell=1}^{L} |\kappa_{(init),\ell} - \kappa_{(dec),\ell}|^2}{L}},$$

$$\mathrm{RMSE}_{\tilde{\mathbf{D}},(rec)} = \sqrt{\frac{\sum_{\ell=1}^{L} \|\tilde{\mathbf{D}}_{(enc),\ell} - \tilde{\mathbf{D}}_{(rec),\ell}\|^2}{L}}, \quad \mathrm{RMSE}_{\mathbf{D},(dec)} = \sqrt{\frac{\sum_{\ell=1}^{L} \|\mathbf{D}_{(init),\ell} - \mathbf{D}_{(dec),\ell}\|^2}{L}}.$$

As shown in Figure 5.19, $\mathrm{RMSE}_{\varepsilon,(dec)}$ and $\mathrm{RMSE}_{\mathbf{D},(dec)}$ gradually rise when the signal distortion goes up. However, the nearly perfect alignment of $\mathrm{RMSE}_{\varepsilon,(rec)}$ and $\mathrm{RMSE}_{\varepsilon,(dec)}$ suggests that the impact of large deciphering errors on energy is statistically negligible. In consequence, the guard bounds $(\varrho_{low},\ \varrho_{high})$ could be relaxed. Additionally, it can be noticed that the error $\mathrm{RMSE}_{\tilde{\mathbf{D}},(rec)}$ is smaller than $\mathrm{RMSE}_{\mathbf{D},(dec)}$. It is because the spherical angles $\mathbf{D}_{(dec)} = \gamma_8^{-1}(\sqrt{8}\boldsymbol{\chi}_{(dec)}/2)$ are divided by 2 in the decoding stage.

The error $\mathrm{RMSE}_{p,(dec)}$ remains small for every analyzed distortion. The rarely occurring errors on pitch are usually significant and easy to detect. The observation suggests that a simple pitch tracker added at the output of the descrambling block would overperform guard bounds $(\kappa_{low},\ \kappa_{high})$ as an error correction mechanism.

In a realistic scenario, the receiver is not always perfectly synchronized in time with the sender. Moreover, some voice channels equipped with adaptive gain control (AGC) may modify the signal amplitude. As suggested by Figure 5.20, the deciphering unit is, to some extent, tolerant of energy mismatch in the encrypted signal caused by AGC. Provided that the mismatch is no larger than the energy guard intervals, a modified signal is decrypted into an energy-scaled speech. On the other hand, the deciphering unit is very vulnerable to synchronization error. As shown in Figure 5.21, the error in deciphered timbre rises dramatically when the mismatch exceeds 0.3 ms.

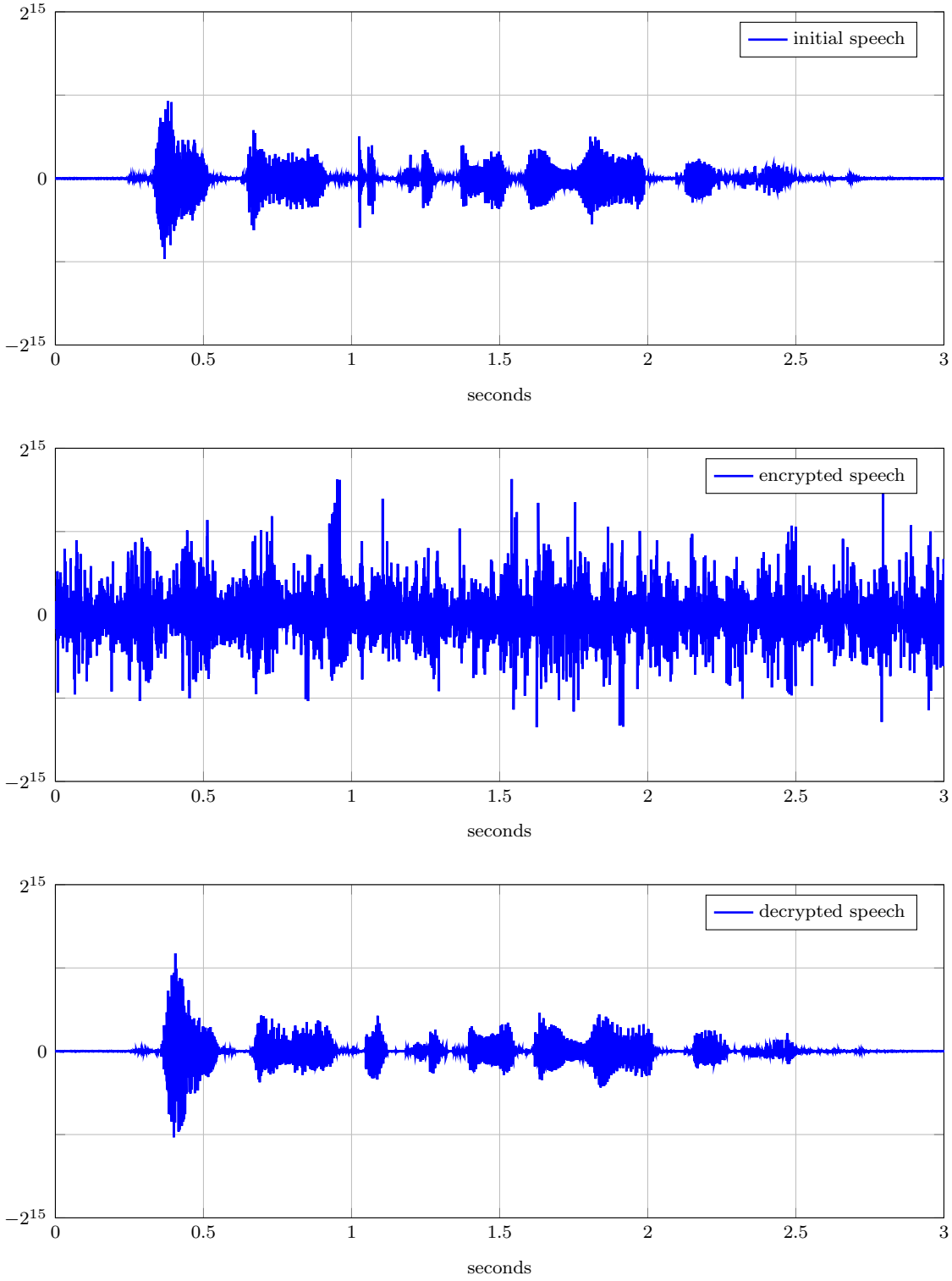


Figure 5.15 – Waveforms at different stages of signal encryption.

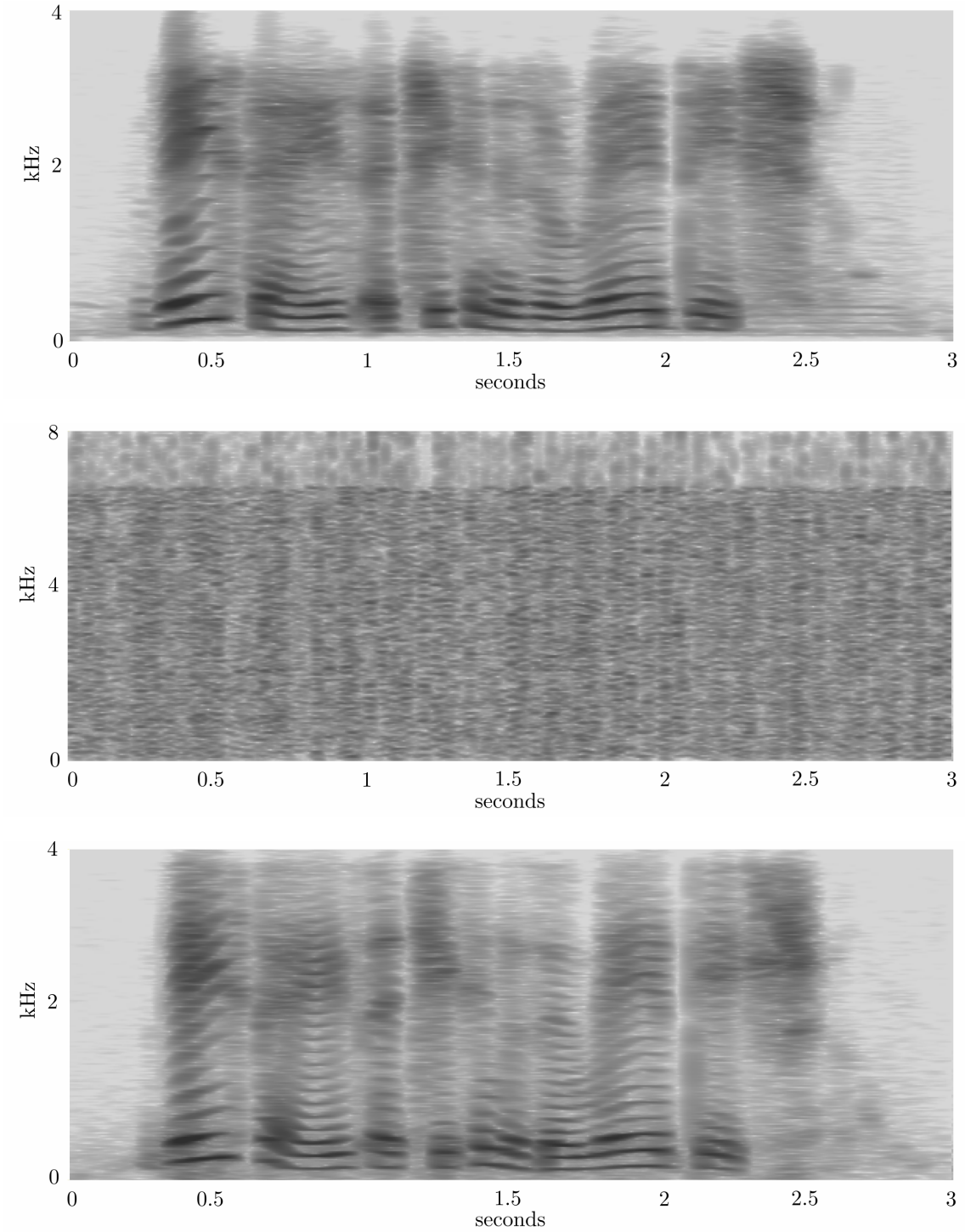


Figure 5.16 – Consecutive stages of signal encryption, presented in the time-frequency domain. From top to bottom: initial speech, encrypted signal and resynthesized speech. The spectrograms were obtained using a 1024-point Hann window.

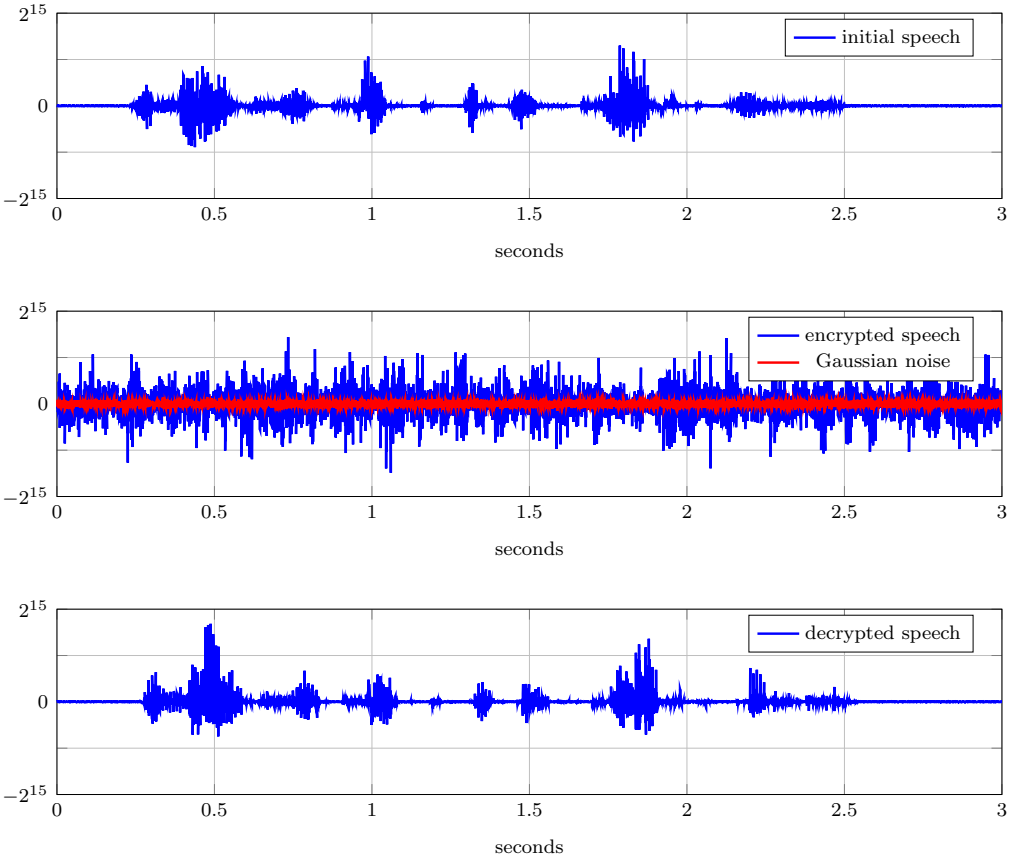


Figure 5.17 – Waveforms at different stages of signal encryption with Gaussian noise at SNR = 15 dB added to encrypted pseudo-speech.

Finally, the scheme's robustness was checked over FaceTime on WiFi between two iPhones 6 running iOS 12. The precomputed pseudo-speech excerpts of total duration 120 seconds and enciphered with a predefined pseudo-random sequence were uploaded on one of the phones, sent over FaceTime in chunks about 10-20 second long, and recorded on the second device. Figure 5.22 illustrates an example of the recorded signal and the decrypted speech. Table 5.3 lists the RMSE of received and deciphered parameters retrieved from 120 seconds of a recorded signal.

Table 5.3 – RMSE of received and deciphered values in communication over FaceTime between two iPhones 6. Results retrieved from 120 seconds of a recorded signal.

$\text{RMSE}_{\varepsilon,(dec)}$	1493.30	$\text{RMSE}_{\tilde{\varepsilon},(rec)}$	1644.73
$\text{RMSE}_{p,(dec)}$	525.70	$\text{RMSE}_{\tilde{p},(rec)}$	867.30
$\text{RMSE}_{\mathbf{D},(dec)}$	0.12	$\text{RMSE}_{\tilde{\mathbf{D}},(rec)}$	0.16

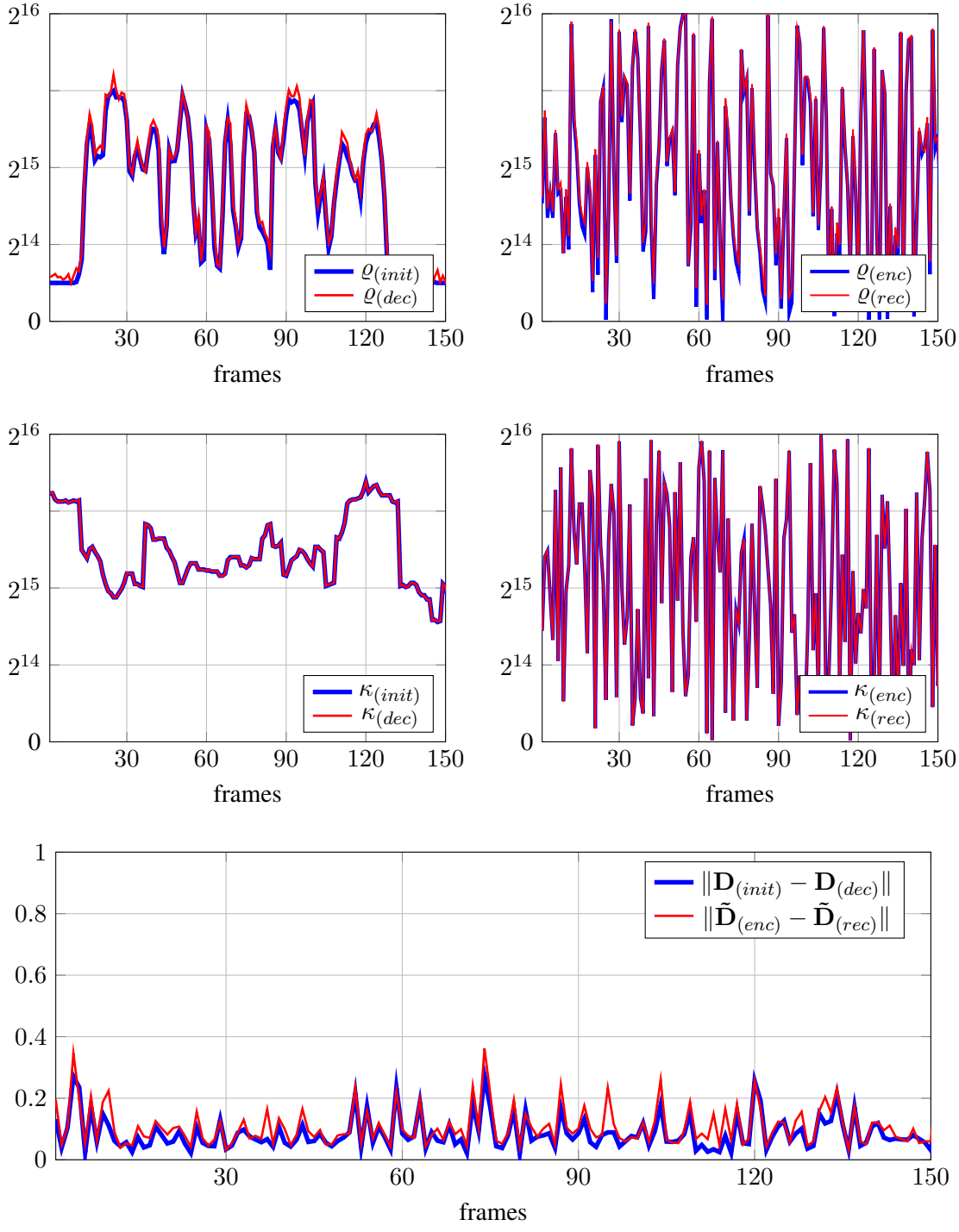


Figure 5.18 – Distortion of received and deciphered parameters caused by adding Gaussian noise at SNR = 15 dB to encrypted speech.

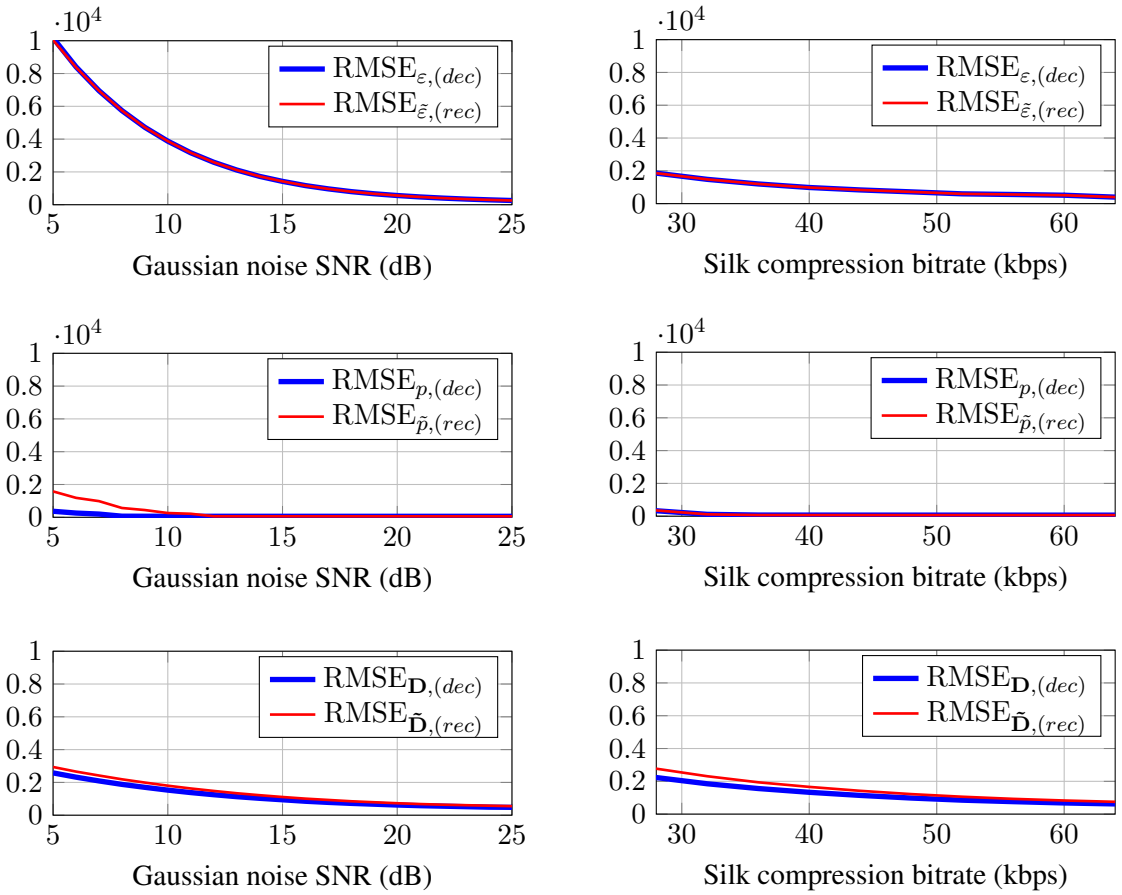


Figure 5.19 – Root mean squared error (RMSE) of deciphered speech values and received pseudo-speech values caused by adding Gaussian noise to encrypted speech (left column) or by compressing the encrypted speech with Opus-Silk (right column). Simulation based on 100000 frames.

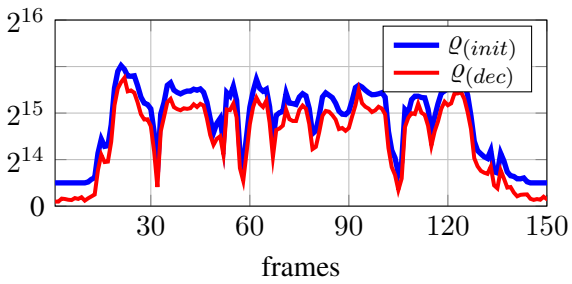


Figure 5.20 – Deciphered speech energy from encrypted signal scaled by the factor 0.85 and distorted by Gaussian noise at SNR = 20 dB.

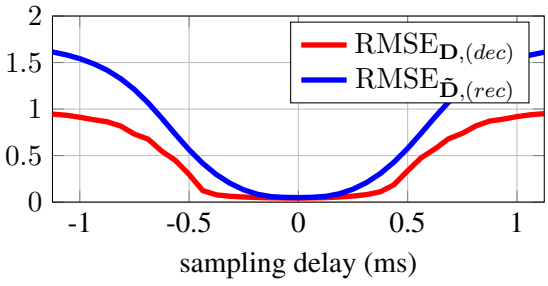


Figure 5.21 – RMSE of received and deciphered vectors representing the shape of spectral envelope; caused by imperfect sampling synchronization on the receiving side. Simulation based on 100000 frames.

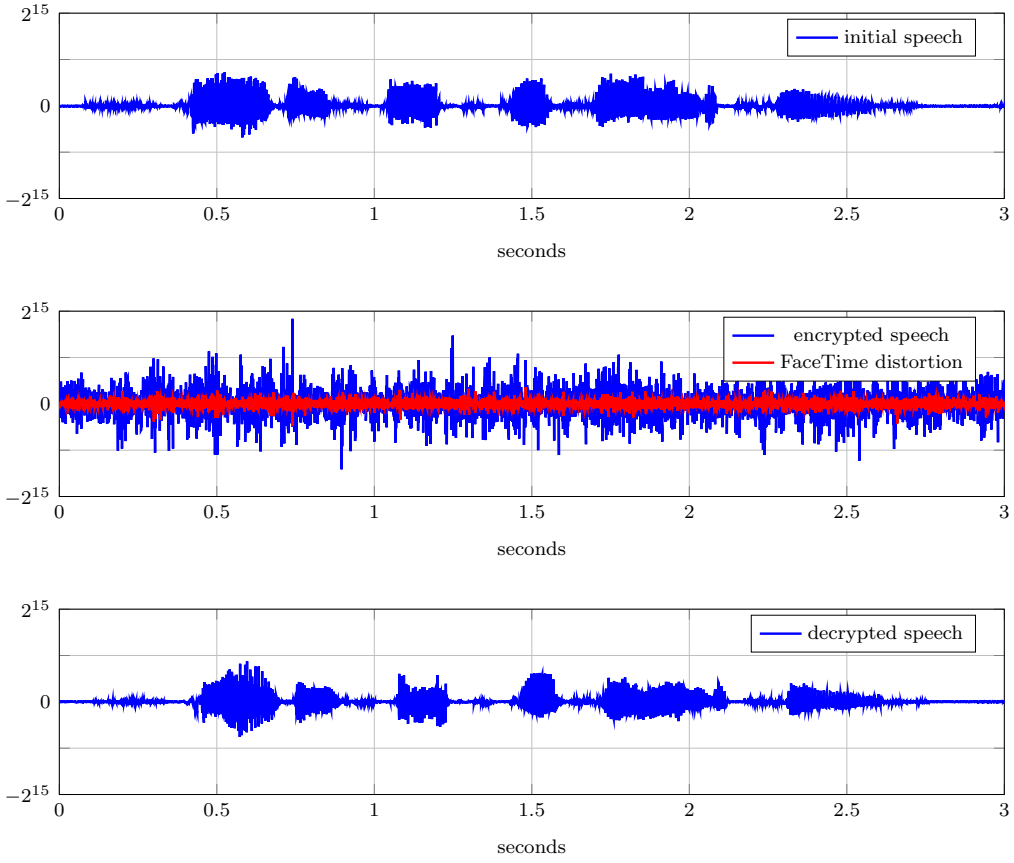


Figure 5.22 – Consecutive stages of signal encryption in communication over FaceTime between two iPhones 6. The recordings are available online at https://github.com/PiotrKrasnowski/Speech_Encryption.

5.4.3 Speech quality evaluation

As reported in [Kleijn et al., 2018], objective measures of speech quality (i.e., PESQ [ITU-T, 2001] and POLQA [ITU-T, 2018]) are suboptimal for evaluating Machine Learning based, non-waveform vocoders. Consequently, we conducted a subjective listening test on a large number of anonymous volunteers. The tests consisted of two parts. The first part checked the subjective intelligibility of decrypted speech in perfect transmission conditions. The second part assessed the subjective quality of speech restored from an encrypted signal with different distortion types. The subset of speech samples used in the listening test has been selected from the LibriSpeech corpus [Panayotov et al., 2015] and is available online. [11]

The intelligibility experiment was inspired by the speech intelligibility rating (SIR) [Cox and McDaniel, 1989]. During the test, the participants listened to 10 English sentences (4 female and 4 male speakers) of about 10 seconds each. In the first round, the speech utterances were consecutively encrypted and decrypted, without distorting. In the second round, listeners were given the initial sentences sampled at 8 kHz, which served as the reference. After listening to each speech sample, the participants were asked to estimate the percentage of recognized words in the sen-

11. https://github.com/PiotrKrasnowski/Speech_Encryption

tence. The ratings were defined as numbers between 0 and 100, where 0 denoted no recognized word and 100 denoted that all words were recognized (Fig. 5.23). As opposed to rigorous, one-word or vowel/consonant intelligibility tests [ITU-T, 2016b], testing the word intelligibility of a sentence allows listeners to take advantage of the context. Because the participants were anticipated to be mostly non-native English speakers, they were allowed to listen to the sentences multiple times.

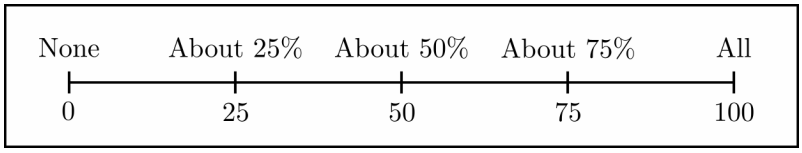


Figure 5.23 – Rating scale used in the perceptual speech intelligibility test.

The quality assessment followed a MUSHRA methodology [ITU-R, 2015] adapted for perceptual evaluation of medium quality speech signals. The method is believed to provide more reliable and reproducible results than the Mean Opinion Score (MOS) measure [ITU-T, 2016a], although it is not immune to biases either [Zielinski et al., 2007]. In the MUSHRA test, a participant is given several test audio files (called excerpts) which represent the same speech utterance processed by different algorithms. To allow the participant a thorough and unbiased evaluation, these excerpts are given simultaneously and in randomized order. Among these randomized excerpts, some represent the actual speech samples under test, whereas the remaining excerpts are a hidden reference, a low-quality anchor, and a mid-quality anchor. During the quality test, the listeners were asked to rate the subjective speech quality (i.e., naturalness, fidelity) against the reference, as a number between 0 and 100 (Figure 5.24). The value 100 denoted 'Excellent' quality, meaning that the perceived quality of the test excerpt was identical to the reference.

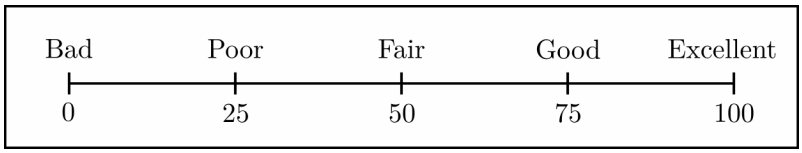


Figure 5.24 – Rating scale used in the perceptual speech quality test.

The MUSHRA tests were conducted in two rounds. The first round aimed at evaluating the quality of sentences that were consecutively encrypted by our algorithm, distorted by AWGN of varying intensity, and decrypted. In the second round, encrypted signals were compressed by Opus-Silk at varying compression rate. In addition, the test excerpts in the second round included speech utterances decrypted from the signal sent over FaceTime and recorded on iPhone 6. In both rounds, the participants had to rate 6 different sentences (3 female and 3 male) of about 10 seconds each. The reference was a wideband signal sampled at 16 kHz, the mid-anchor was a narrowband signal sampled at 8 kHz, and the low-anchor was a narrowband speech signal sampled at 8kHz and with the MNRU distortion at SNR = 15 dB [ITU-T, 1996b]. In contrast to the reference signal, the mid-anchor may serve as a good benchmark to our tested signals due to the same speech bandwidth. The systems tested in the speech quality assessment are summarized in Table 5.4.

The assessment was carried out entirely online using webMUSHRA, the framework for Web-based listening tests [Schoeffler et al., 2015, Schoeffler et al., 2018]. The URL of the assessment

was publicly available and widely distributed using social media. The participants were anonymous volunteers and their speech quality evaluation was not supervised by us. However, we assume that the listeners were mostly non-native English speakers with unreported hearing impairments. Table 5.5 lists the number of participants taking part in each test. The participants were asked to wear headphones or earphones and were allowed to adjust the sound volume. Few participants were excluded from aggregated responses because of rating the hidden reference below 90 more than once in a single test (mostly accidentally).

Table 5.4 – Hidden anchors and tested systems in the MUSHRA-based speech quality assessment.

Reference and anchors	
Label	Description
reference	wideband speech sampled at 16 kHz
mid-anchor	narrowband speech sampled at 8 kHz
low-anchor	8 kHz narrowband speech with MNRU at SNR = 15 dB
Systems under test in the assessment 1	
Label	Description
no distortion	decrypted speech from signal with no distortion
20 dB SNR	decrypted speech from signal with AWGN at SNR = 20 dB
15 dB SNR	decrypted speech from signal with AWGN at SNR = 15 dB
10 dB SNR	decrypted speech from signal with AWGN at SNR = 10 dB
Systems under test in the assessment 2	
Label	Description
Silk 64 kbps	decrypted speech from signal compressed with Silk at 64 kbps
Silk 48 kbps	decrypted speech from signal compressed with Silk at 48 kbps
Silk 32 kbps	decrypted speech from signal compressed with Silk at 32 kbps
FaceTime	decrypted speech from signal sent over FaceTime

Table 5.5 – Number of participants in the listening test.

Test	Participants
Intelligibility test	44
Quality test 1	40*
Quality test 2	37**
* 18 listeners rated 5 utterances instead of 6 4 listeners excluded for reference underrating	
** 3 listeners excluded for reference underrating	

Table 5.6 presents sample mean and sample standard deviation of the intelligibility test. On average, the participants recognized about 12% fewer words in synthesized speech samples than in the reference. The average rating of particular sentences varied slightly from 82% to 89%. On the other hand, a speaker-level average ranged from 58% to 99%. This high variability of average ratings given by the listeners explains a considerable standard deviation of aggregated responses.

The results of the MUSHRA-based quality assessment are depicted in Figures 5.25 and 5.26. In both test rounds, the hidden reference was rated correctly as 'Excellent.' The average rating of the mid-anchors given by the participants was about 75% ('Good'), and the average rating of the low-anchors was about 30% ('Poor').

The average rating of test excerpts labeled ‘no distortion’ was 64% (‘Good’/‘Fair’). Compared with the average rating of mid-anchors, our algorithm reduced the speech quality by about 10%. It may be noticed that this difference in speech quality between the mid-anchors and the excerpts labeled ‘no distortion’ is similar to the intelligibility loss in the SIR-based intelligibility assessment.

The introduction of distortion into encrypted signals resulted in degraded speech quality. Gaussian noise at SNR equal to 20 dB, 15 dB, and 10 dB lowered the average ratings of speech quality respectively to 59% (‘Fair’), 46% (‘Fair’), and 19% (‘Poor’/‘Bad’). It can be noticed that a small channel distortion, like the one introduced by AWGN at SNR = 20 dB, has a relatively minor impact on perceived speech quality. On the contrary, the quality becomes bad when SNR reaches 10 dB. A similar observation can be made in the case of signal compression by Opus-Silk. The compression of encrypted signals at 64 kbps, 48 kbps, and 32 kbps reduces the rated speech quality respectively to 59% (‘Fair’), 52% (‘Fair’), and 28% (‘Poor’). The excerpts decrypted from signals sent over FaceTime were rated at 49% (‘Fair’).

Table 5.6 – Intelligibility test results.

System	Sample mean	Sample standard deviation
Reference	97.5	6.6
Decrypted	86.0	14.7

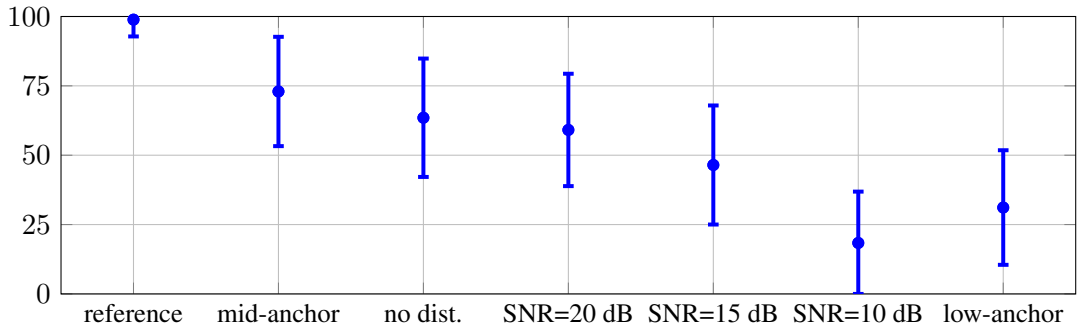


Figure 5.25 – Results of the MUSHRA-based subjective quality assessment of speech decrypted from signals with added Gaussian noise of different intensity. Bars mark standard deviation.

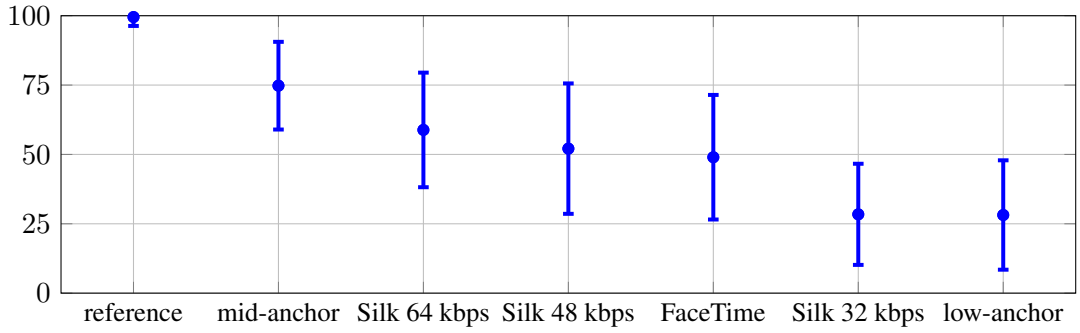


Figure 5.26 – Results of the MUSHRA-based subjective quality assessment of speech decrypted from signals compressed by Opus-Silk vocoder at different compression rates and from signals sent over FaceTime between two iPhones 6. Bars mark standard deviation.

The statistical similarity of given ratings was evaluated by the non-parametric Kruskall-Wallis test [Kruskal and Wallis, 1952], which is more suitable for ordinal scales [Mendonça and Delikaris-Manias, 2018]. The ratings of speech signals labeled 'no noise' come from the same statistical distribution as speech signals labeled 'SNR = 20 dB' with the 0.09 confidence. Additionally, the ratings of speech labeled 'Silk 48 kbps' are similar to speech labeled 'FaceTime' with 0.25 confidence, and the ratings of speech labeled 'Silk 32 kbps' come from the same distribution as the low-anchor with 0.59 confidence. The ratings of the remaining systems were similar, with the confidence much lower than 0.05.

The obtained results suggest that the speech encryption scheme described in this study can produce intelligible speech. Moreover, the average speech quality of excerpts labeled 'FaceTime' hints about the possibility of making our system compatible with VoIP. However, high variability in listeners' responses indicates that the quality of decrypted speech is insufficient for having a casual conversation. Thus, some progress has to be made to improve the system's robustness to distortion and the quality of speech synthesis.

5.4.4 Algorithmic latency and computational complexity

The minimum algorithmic latency in our encryption scheme is the sum of delays introduced respectively by the enciphering and deciphering algorithms. The speech encoder introduces 30 ms of delay (20 ms frame and 10 ms look-ahead), and the pseudo-speech analyzer introduces an additional 20 ms delay. Finally, two 1x3 convolutional layers in the speech synthesizer use a 40 ms look-ahead (2 frames). The combined 90 ms of the minimum algorithmic latency is significant and may reduce the perceived quality of conversation. A possible solution is to reduce the analysis look-ahead to 5 ms, and the synthesis look-ahead to 20 ms, like in the wideband LPCNet [Valin and Skoglund, 2019].

The speech encoder implemented in the encryption scheme is reported to have a complexity of 16 MFLOPS, where about 8 MFLOPS are used for pitch prediction [Valin and Skoglund, 2019]. Moreover, the authors hint at the possibility of significant optimizations. These given values relate to the scenario when a speech signal is sampled at 16 kHz. Thus, we roughly estimate our 8 kHz speech encoder's complexity to about 8 MFLOPS, including torus mapping transformations.

Enciphering (and deciphering) is relatively lightweight, as it requires only ten additions modulo per 20 ms frame. However, a higher computational load is associated with producing secure bitstrings by the pseudo-random generator at a rate 8 kbps. For this reason, it is especially important to select a PRNG based on well-established ciphers adapted for real-time applications, such as AES-CTR [Käsper and Schwabe, 2009, Park and Lee, 2018].

Pseudo-speech synthesis consists of two steps: computing the harmonic parameters of a frame and producing the signal samples. The complexity of the first step is dominated by deriving the complex amplitudes of harmonics $\breve{\mathbf{A}}$ of length K using Equation 5.14, where K is the number of harmonics in a particular frame. Provided that all complex matrices $(\mathrm{H}\tilde{\mathrm{B}}_{\omega_0})^{\dagger}$ are precomputed and stored in the memory, the vector $\breve{\mathbf{A}}$ can be obtained by searching the appropriate matrix from the look-up table, by element-wise complex vector multiplication, and finally by one complex matrix $16 \times K$ multiplication. On the other hand, frame synthesis requires $\mathcal{O}(400K)$ floating-point operations, where 400 is the number of samples within a frame with two guard periods.

The pseudo-speech analyzer is mostly occupied by estimating the received fundamental frequency. The maximum-likelihood estimator implemented in the scheme has complexity $\mathcal{O}(N \log N) + \mathcal{O}(NK)$, where $N = 2^{16}$ is the number of possible pitch values [Nielsen et al., 2017]. Consequently, lowering the resolution of estimations or replacing the pitch predictor with a more efficient version will give considerable computational gains.

The most computationally involving element in the encryption scheme is the final speech reconstruction. The LPCNet model implemented in the scheme has complexity:

$$C = (3dN_A^2 + 3N_B(N_A + N_B) + 2N_BQ) \cdot 2f_s, \tag{5.30}$$

where $N_A = 384$, $N_B = 16$, $d = 10\%$, $Q = 256$ is the number of μ-law levels, and f_s is the sampling frequency. For $f_s = 16$ kHz, the estimated complexity of the synthesizer is 3 GFLOPS [Valin and Skoglund, 2019]. Additionally, it is reported that a C implementation of the synthesizer requires 20% computing power of a 2.4 GHz Intel Broadwell core, 68% of a 2.5 GHz Snapdragon 845 core (Google Pixel 3), and 31% of a 2.84 GHz Snapdragon 855 core. From this, we estimate that the complexity of the lightweight, narrowband implementation of LPCNet is about 2 GFLOPS, and it could operate in real-time on portable devices.

Table 5.7 lists the computational complexity of various parts of our algorithm estimated using PyPaPi [12] library [Terpstra et al., 2010] when processing 60 minutes of a recorded speech in Python. The measurements were done under Ubuntu kernel 5.8.0-25 and using Intel Core i7 2.9 GHz without multi-threading. The pseudo-random bitstring used for enciphering and deciphering was precomputed and stored in the memory.

The listed results suggest that every tested part has a complexity low enough to be carried by a portable device, especially if one considers migrating the experimental Python code to a compiled code. Moreover, a replacement of the pitch predictor in the pseudo-speech analyzer would lead to significant optimization gains. On the other hand, the computational analysis does not include other essential elements of the system, such as keeping signal synchronization or adaptive energy equalization.

Table 5.7 – Estimated complexity using PyPaPi library.

Process	MFLOPS
speech encoding	8*
enciphering	2
pseudo-speech synthesis	1032
pseudo-speech analysis	2756
• pitch prediction	2123
• remaining	634
deciphering	2
speech synthesis	2000*
* from [Valin and Skoglund, 2019]	

12. https://flozz.github.io/pypapi/

5.5 Summary

In this chapter, we proposed a new speech encryption scheme for secure voice communications over voice channels. The lossy speech encoding technique implemented in the system preserves and protects only basic vocal parameters: fundamental frequency (pitch), energy (loudness), and spectral envelope (timbre). The vocal parameters are enciphered using spherical group codes and then encoded to a synthetic audio signal adapted for transmission over wideband voice channels. Speech is reconstructed by the narrowband vocoder based on the LPCNet architecture.

Enciphering of vocal parameters is done using norm-preserving techniques: pitch and fundamental frequency are enciphered by translations, whereas spectral envelope by rotations on the hypersphere in 16 dimensions. These techniques enable successful decryption of signals distorted by moderate transmission noise, like AWGN, or processed by some wideband VoIP applications such as FaceTime. However, the enciphering mechanism does not provide any data integrity. Instead, it is critical to ensure strong identity authentication in the initial cryptographic key exchange. Authenticated key exchange over voice channels is covered in Chapter 6.

The robustness of the speech encryption scheme against channel distortion was verified experimentally. Simulations showed that the system could correctly decrypt pseudo-speech with additive Gaussian noise at SNR = 15 dB or compressed by the Opus-Silk codec at the 48 kbps rate. On the other hand, an encrypted signal is sensitive to synchronization error larger than 0.3 milliseconds. Furthermore, the results of the speech quality assessment indicated that the proposed encryption scheme could produce intelligible speech with the quality depending on channel distortion.

The preliminary complexity evaluation and the successful transmission of encrypted signals between two mobile phones hint that the proposed encryption scheme may work in real-time on high-end portable devices. However, secure communication is susceptible to short signal dropouts or de-synchronization. Consequently, robust communication is possible only over a stable vocal link between the users. Additionally, adaptive voice-enhancing algorithms implemented in commercial mobile phones (such as voice detection and noise suppression) usually lead to considerable degradation of the speech quality. This problem can be tackled using dedicated CryptoPhones or stand-alone devices connected with mobile phones in tandem, as described in the next chapter.

The presented experimental scheme requires further investigation. Firstly, speech quality could be improved by replacing our narrowband speech synthesizer with the 4 kHz bandwidth with a synthesizer with the 8 kHz bandwidth. The biggest challenge is to find a new representation for the spectral envelope, which is compatible with the enciphering technique. The presented solution uses 9 mel-scaled frequency windows that are insufficient for encoding the wideband spectrum. A possible solution is to increase the number of mel-scaled windows to 18 and apply a dimensionality reduction technique, such as Principal Component Analysis (PCA) [Wold et al., 1987] or autoencoding [Kramer, 1991]. Dimensionality reduction may increase encoding efficiency because the coefficients within a single speech frame tend to be highly correlated.

Other improvements can be obtained in the pseudo-speech synthesis. The proposed synthesis technique, while computationally efficient, is very phase-sensitive and not enough speech-like. Instead of encoding the enciphered vector $\tilde{\mathbf{D}}_{(enc)}$ into the real part of the complex frame spectrum, it would be advantageous to encode $\tilde{\mathbf{D}}_{(enc)}$ into the power spectral density (PSD). The main limitation is that the vector $\tilde{\mathbf{D}}_{(enc)}$ contains both positive and negative values, whereas PSD is always non-negative. For this reason, envelope encoding could be performed in the cepstral domain.

Furthermore, it may be worth adding a correction unit at the deciphering output for detecting and smoothing deciphering errors. Since the vocal parameters in natural speech do not change quickly over time, the detection of large errors should be relatively straightforward. For example, the correction unit could use machine learning techniques to correct errors on a particular channel. A clear separation between the correction unit and the speech synthesizer could improve the quality of synthesized speech and simplify the two-step network training.

Communication performance strongly depends on the stability of a vocal link. The problem with fading channels could be mitigated by combining distortion-tolerant speech encryption and multiple description coding (MDC) [Goyal, 2001, Venkataramani et al., 2003, Wah and Dong Lin, 2005]. Multiple description coding is a technique that fragments one media stream into several substreams. Each substream is decodable into the initial stream, and decoding more substreams improves the quality. The MDC could be used to split encrypted speech into multiple audio streams and increase communication reliability.

In Chapter 6, we investigate authenticated key exchange over fading voice channels between two speakers. Secure voice communication is possible only with a secret cryptographic key shared by both parties. However, key authentication becomes very challenging without Public Key Infrastructure (PKI) or reliable data-driven side channels. The next chapter proposes a robust Diffie-Hellman key exchange authenticated by digital signatures and vocal verification. The exchange requires data transmission over a voice channel, for example, using the DoV technique described in Chapter 3.

CHAPTER 6

Key exchange over voice channels

Secure communication over voice channels requires a prior exchange of cryptographic keys over voice channels, without reliance on any Public Key Infrastructure (PKI). This chapter describes our formally verified and authenticated key exchange (AKE) over voice channels for secure voice communications, firstly introduced in [Krasnowski et al., 2020] and presented at ICISSP 2020. It outlines the operational principles of the novel communication system and enlists its security requirements. The voice channel characteristics in the context of AKE protocol execution are thoroughly explained, emphasizing differences to classical store-and-forward data channels. Namely, a robust protocol has been designed specifically for voice channels with double authentication based on signatures, and Short Authentication Strings (SAS) compared vocally by the users. The protocol is detailed and analyzed in terms of fundamental security properties and successfully verified in a symbolic model using Tamarin Prover.

Glossary

List of abbreviations

AKE	Authenticated Key Exchange
CA	Certificate Authority
DoS	Denial-of-Service
DoV	Data over Voice
ECDHE	Ephemeral Elliptic Curve Diffie-Hellman
MAC	Message Authentication Code
MITM	Man-In-The-Middle
MSR	Multiset Rewritting
PFS	Perfect Forward Secrecy
PKI	Public Key Infrastructure
SAS	Short Authentication String
TTP	Trusted Third Party
VAD	Voice Activity Detection
VoIP	Voice over Internet Protocol

Notation - protocols

ID_U	fixed user identifier
N_U	random and unique nonce
K_S	Session Key
SAS	Short Authentication String displayed on the device
$(R_A,\ R_B)$	Short Authentication String seeds
$(d_U,\ Q_U)$	secret/public ECDHE key pair
$(S_U,\ V_U)$	signing/verification key pair
$Sign_{S_U}(\cdot)$	signature (signed with S_U)
$Enc_{K_U}(\cdot)$	ciphertext (enciphered with a symmetric key K_U)
$h_X(\cdot)$	hash function with truncation to X bits

6.1 Motivation

An increasing concern of privacy violation in voice communications has motivated the development of secure voice over IP (VoIP) applications, with Telegram and Signal being the iconic examples. [1] However, these applications are inherently insecure against spying malware installed on the smart-phone [Scott-Railton et al., 2017]. Parallelly, cryptographically secure applications requiring higher protection rely on dedicated hardware, most commonly Crypto Phones. These closed and unverifiable solutions suffer from high costs and low flexibility, as typically encrypted phones allow communications exclusively over a single kind of a voice channel, like GSM.

The mentioned limitations encourage the search for open solutions complementary to Crypto Phones, combining flexibility and high protection provided by specialized hardware. A new idea, depicted in Figure 6.1, is based on voice encryption in the audio domain. The speech is acquired by (a) the headset's microphone and then forwarded to (b) the encryption device (here called the Crypto Box). The Crypto Box processes the speech and enciphers vocal parameters of the signal. The encrypted speech in the form of a data stream shaped into a pseudo-speech audio signal (e.g., as proposed in Chapters 3 and 5) is transmitted by (c) the audio link to the audio input of (d) the phone and sent through 2G-4G networks or VoIP. Finally, the received pseudo-speech is deciphered by the paired Crypto Box on the other side of the channel.

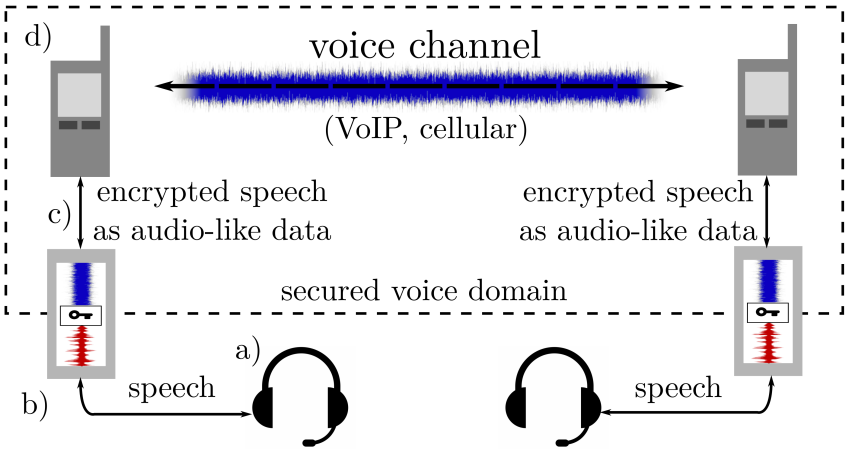


Figure 6.1 – Encrypted voice over voice channel scheme.

In such a setting, voice encryption is performed outside of the phone, protecting against audio-recording malware. To limit the system corruption risk, the Crypto Box has only analog input/output interfaces to the headset and the phone. However, it is necessary for security reasons that other analog inputs of the phone (particularly the built-in microphones) should be blocked by a special casing or removed.

From the system perspective, two Crypto Boxes are the end-points of a secured voice domain. Everything in between, including mobile phones themselves, is a communication infrastructure that enables voice transmission. The framework adds a new layer of security, protecting against spying malware installed on the phone. Since all communications between encrypting devices are done purely in the analog domain, selecting the specific voice communication technology is,

1. https://signal.org, https://core.telegram.org

therefore, a secondary issue. Compatibility with most vocal communication methods, like VoIP applications or 2G-4G networks, significantly widens the range of usability scenarios. The described setting, which is not intended for a daily-usage, is of great interest for business, diplomatic and military services, who require secure communications in unreliable environments and without access to confidential communication infrastructure.

The major motivation in our approach is to secure voice communications even with untrusted phones. Consequently, the phones should not be actively involved in the setup of a secure connection or sensitive data storage. The open framework enables various hardware solutions, including combining the phone and the Crypto Box into a single device.

Secure speech enciphering requires a prior exchange of session keys between the Crypto Boxes. Due to system requirements, the key exchange can only be made through the same point-to-point voice channel using Data over Voice (DoV) technique like the one described in Chapter 3. Consequently, there is no practical possibility of adding an online trusted third party (TTP) or a certificate authority (CA). Such a limitation is a big concern for users' authentication.

Research on secure key exchange between two honest parties without any TTP led to the creation of standards suitable for VoIP applications, like the extension of the Real-Time Transport Protocol (RTTP), called ZRTP [Callas et al., 2011], and Multimedia Internet KEYing (MIKEY) protocol [Arkko et al., 2004]. Especially ZRTP is interesting in the context of this work because it provides an authentication mechanism in the absence of any Public Key Infrastructure (PKI) or a pre-shared secret. In these situations, authentication is based on vocally comparing Short Authentication Strings (SAS). Unfortunately, with three modes of operation and extensive negotiation signaling, even ZRTP seems overly complicated for communication over voice channels. Moreover, none of the protocols put a sufficient emphasis on resistance to strong message distortion or desynchronization in a low-bandwidth environment.

6.2 System requirements

The need for hardware-based voice encryption is a response to an increased risk of being intercepted. Thus, a cryptographic scheme should reflect higher requirements for secrecy and authentication. The primary threat is recording and analyzing the network traffic by omnipresent passive eavesdroppers. Active attackers controlling the network are more likely to block or distort communications, which is technically very simple. However, a powerful and knowledgeable attacker who is able to analyze and synthesize a compatible pseudo-speech may try to modify a message or insert his own. Finally, in critical situations, the encrypting device could be hijacked in order to extract long-term keys. On the other hand, in our work, we assume that the encryption device does not allow any intrusion into its internal memory during the operation, so all ephemeral data stored on the device (and deleted after each protocol run) should be considered secure.

The design process of the protocol is motivated by an anticipated user experience. However, due to the severe constraints of the voice channel characteristics, the most significant challenges are protocol complexity, synchronization, and robustness. A major bottleneck is a large message round-trip time, around 2 seconds long, making the whole protocol run-time prohibitively long even in simple protocols. Another limitation is the limited bandwidth implying a reduction of the message size. Moreover, the protocol must be robust against fading and signal distortion, requiring signalization simplification and strong error correction mechanisms. Finally, in order to reduce battery power consumption, the cryptographic operations should be relatively lightweight

and optimized. When implementing, relying on popular and verified network security libraries, like OpenSSL, NaCL, or Mbed TLS, could be a practical advantage.

Adaptation to hardware and channel constraints should not lead to any significant relaxation of the security level. It will be detailed that the key exchange protocol provides strong mutual agreement on the parameters used for the computation of the session key. Moreover, it aims at preventing Man-In-The-Middle (MITM) attacks and achieving Perfect Forward Secrecy (PFS). The protocol enables users' authentication, no matter if they share a common secret or not.

A successful and fast key exchange is an indicator of sufficiently good channel conditions that offers comfortable communication. Every received message can be used to estimate the channel characteristics effectively and to improve the decoding efficiency.

6.3 Key exchange protocols and symbolic security verification

Designing security protocols is prone to errors, which resulted in the publication of several flawed protocols [Just and Vaudenay, 1996, Lowe, 1996, Lauter and Mityagin, 2006, Farrell, 2009, Tsay and Mjølsnes, 2012]. Therefore, every protocol design ought to be thoroughly scrutinized by some formal security verification. The verification should focus on the security of the ciphers used in the protocol by eliminating logical flaws in the protocol's messages.

At the high level, the cryptographic algorithms can be validated in the computational model by showing the equivalence between the protocol's security and some computationally hard problems, e.g., integer factorization. The formalization often uses the indistinguishability approach [Canetti and Krawczyk, 2001, Katz and Lindell, 2015], in which the adversary with finite power tries to distinguish the generated key from an independent random string with fixed length.

Proving the protocol's computational security provides clear advantages such as concrete bounds on the probability of successful attack and the required keying material. However, it is often a laborious and manual task. Moreover, it is difficult to show that a particular protocol is free from logical flaws, or in other words, if its security cannot be compromised using only intended protocol interactions.

The presence of logical flaws in the protocol is particularly important in communication over channels controlled by the attacker. For instance, apart from interception, the attacker may block and replay messages or insert his own versions. Furthermore, it is often assumed that the attacker may concurrently communicate with many legitimate participants and replay messages.

The risk of logical flaws in the protocol can be mitigated or eliminated in the symbolic model, using some formalization of the protocol execution. In contrast to the computational model, the protocol messages in a symbolic model are sets of symbolic terms representing the algorithms used to construct a particular message. Additionally, the cryptographic primitives and algorithms are assumed perfect, meaning that they can be considered secure if checked in the computational model. Formal symbolic verification is the first step of a protocol analysis, paving the way to computational model verification [Goldwasser and Micali, 1984, Blanchet, 2012].

The protocol simplification offered by a symbolic formalism enables the automatic construction of symbolic security proofs. This task can be undertaken by many tools, such as AVISPA [Armando et al., 2005], Scyther [Cremers, 2008], ProVerif [Blanchet, 2001] or Tamarin Prover [Meier et al., 2013]. Though, proving the security of a protocol is in general NP-complete for a bounded number of sessions [Rusinowitch and Turuani, 2003] and becomes undecidable for an unbounded number of sessions [Durgin et al., 2004]. As a result, the automatic verification tools rely on

heuristics and provide solutions for some classes of protocols. Some especially advanced proving techniques are implemented in Tamarin and ProVerif, capable of constructing efficiently security proofs even for an unbounded number of threads. Crucially, the heuristics applied during verification does not affect the results' soundness (the properties that are proved are always true).

ProVerif represents cryptographic protocols by a set of Horn rules (clauses) [Gupta, 1999]. Thus, security proving boils down to deriving facts in a Horn logic theory [Blanchet, 2009]. Horn formalism enables the modeling of many cryptographic protocol verification problems, including Diffie-Hellman exponentiation and signing. However, the approximations taken during the proving process may give some false alarms, which should be eliminated manually. To reduce the chance of non-termination resulting from undecidability, ProVerif uses a specialized resolution prover and several reduction techniques.

Unfortunately, the reduction to Horn theory prevents modeling protocols with non-monotonic state, i.e., protocols that intentionally reveal some secret during their execution. The possible workarounds, such as process replication with new names, may raise more false-alarms.

The problem mentioned above can be partially avoided in Tamarin, [2] a powerful and increasingly popular automatic verification tool designed at ETH Zürich. Tamarin models protocols as multiset rewriting (MSR) systems that specify security properties in a guarded fragment of first-order logic. The tool supports generic Diffie-Hellman group operations and many cryptographic primitives like signatures or hashes [Schmidt et al., 2012, Schmidt et al., 2014].

In Tamarin, the protocol's execution can be viewed as a labeled transition system representing the evolving adversarial knowledge, the messages sent over the network, freshly generated values, and the protocol state. Thus, a single protocol realization takes the form of a time-stamped trace, and the security properties are modeled as trace properties. Consequently, the verification (or falsification) of the protocol's security reduces to exploring the possible traces that may violate the specified security property.

From the user's perspective, the protocol verification can be done in two ways: using a heuristic-based fully automated mode and an interactive mode. The verification result is bringing some counterexample or the proof of correctness in the unbounded number of sessions and fresh values. Nonetheless, because of undecidability, Tamarin may also not terminate.

Tamarin's clear advantage is a user-friendly browser-based interface, which simplifies the analysis of the verification output. When a counterexample is found, a diagram of the protocol execution is displayed. Moreover, Tamarin offers an impressive database of examples that makes the tool suitable for protocol evaluation.

6.3.1 Simple example of a protocol verification using Tamarin

Tamarin's operation can be better understood using a simple example of KE protocol, which involves sending Diffie-Hellman public keys authenticated by signatures. The protocol, presented in Figure 6.2, consists of three messages exchanged between Alice and Bob, who share the signature verification keys. The protocol execution starts with Alice sending her identification number ID_A and the public key g^x. Bob responds by sending his identifier ID_B, public key g^y, and the signature signed with key S_B. The last message is the signature of Alice signed with S_A. The protocol finishes by computing the session key K_S. The set $\mathbb{Z}_q^*$ denotes the integer ring $\mathbb{Z}_q$ without the zero element, $\|$ denotes concatenation, and $\leftarrow\!\!\$$ denotes a uniformly distributed probabilistic process assignment [Katz and Lindell, 2015].

2. https://tamarin-prover.github.io/

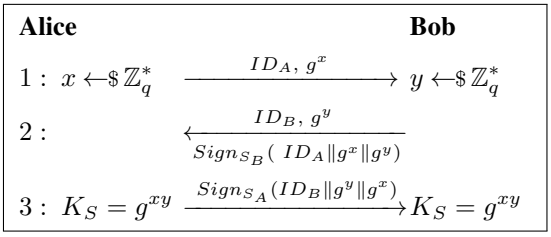


Figure 6.2 – Key exchange protocol with signatures.

Protocol modeling: The Tamarin code describing the protocol model is listed in Figure 6.3. The model consists of a key pair generation rule Generate_pk and four multiset rules which specify the protocol interactions. Each of the rules can be invoked an unlimited number of times and in any order.

```
theory Protocol_example
begin

builtins: diffie-hellman, signing

// Generation of the signature key pair
rule Generate_pk :
    [ Fr( ~ltk ) ] // fresh long-term key
    --[ Generate( $ID_U ) ]-> // generate keys for some user U
    [ !Sign_Key( $ID_U, ~ltk ), !Verif_Key( $ID_U, pk( ~ltk ) ) ]

// Protocol rules
rule Initiator_1 :
    [ Fr( ~Aprivkey ) ] // fresh ephemeral private key
    -->
    [ Initiator_State( $ID_A, ~Aprivkey ), Out(< $ID_A, 'g'^~Aprivkey >) ]

rule Responder_1 :
    [ Fr( ~Bprivkey ), !Sign_Key( $ID_B, ltkB ), In(< $ID_A, Apubkey >) ]
    -->
    [ Responder_State( $ID_B, $ID_A, ~Bprivkey, Apubkey ),
      Out(< $ID_B, 'g'^~Bprivkey, sign{ $ID_A, Apubkey, 'g'^~Bprivkey }ltkB >) ]

rule Initiator_2 :
    let
      session_key = Bpubkey^~Aprivkey // shared DH secret
    in
    [ Initiator_State( $ID_A, ~Aprivkey ), !Sign_Key( $ID_A, ltkA ),
      In(< $ID_B, Bpubkey, signB >),  !Verif_Key( $ID_B, pk( ltkB ) ) ]
    --[ Session_Initiator( $ID_A, $ID_B, session_key ),
        Verify_Signature_Initiator( signB, sign{ $ID_A, 'g'^~Aprivkey, Bpubkey }ltkB ) ]->
    [ Out( sign{ $B, Bpubkey, 'g'^~Aprivkey }ltkA ) ]

rule Responder_2 :
    let
        session_key = Apubkey^~Bprivkey // shared DH secret
    in
    [ Responder_State( $ID_B, $ID_A, ~Bprivkey, Apubkey ),
      In( signA ), !Verif_Key( $ID_A, pk( ~ltkA ) ) ]
    --[ Session_Responder( $ID_B, $ID_A, session_key ),
        Verify_Signature_Responder( signA, sign{ $B, 'g'^~Bprivkey, Apubkey }~ltkA ) ]->
    [ ]
```

Figure 6.3 – Tamarin code of the protocol model.

Rules are defined by four elements: their name, left-hand side facts (inputs), transition facts (actions), and right-hand side facts (outputs). As an example, the rule Initiator_2 takes as input the state Initiator_State from the rule Initiator_1, the fact Sign_Key representing the Initiator's signing key, the fact Verif_Key representing the Responder's verification key and the In-fact denoting the message assumed to be sent by the Responder. Furthermore, the expected message is the concatenation of the Responder's ID_B, her public key 'g'^Bprivkey, and her signature signB. The transition fact Session_Initiator binds the constants representing the users' identifiers with the generated session key symbol, whereas Verify_Signature_Initiator links the received signature with the signature expected by the Initiator. Finally, the rule Initiator_2 generates the Out-fact with the Initiator's signature.

The adversary would need to be granted some appropriate capabilities and access to specific knowledge to make the protocol model more realistic. The fact types used in the rules put specific limitations to adversarial manipulations. The In-fact representing the message received from the network can accept messages from a legitimate user and the adversary. The Out-facts, once generated, can be replayed and manipulated. On the other hand, the constants generated by Fr-facts (from Fr-esh) are unique and unguessable by definition.

Similar properties can also characterize constants. The dollar sign $ next to the symbol means that the symbol is known globally (like the user's identifier). The tilde ~ denotes a random value, unguessable before being revealed.

The cryptographic primitives used in the model are assumed to be secure in the computational model. For instance, the adversary cannot extract the secret key from the group exponents or forge the signatures. However, the adversary is capable of generating his public keys and signatures. Therefore the protocol rules must permit the reception of invalid input and provide some method for its verification.

```
44  lemma Session_Key_Secrecy_1 :
45    "
46    // for any participants A and B, session key and timepoint i
47    All ID_A ID_B session_key #i.
48       // if the responder B completed a run presumably with the initiator A
49       Session_Responder( ID_B, ID_A, session_key ) @ i
50       ==> // then
51       // the adversary never knows the session_key
52       not(Ex #k. KU( session_key ) @ k)
53     "
```

Figure 6.4 – Tamarin code of the security lemma Session_Key_Secrecy_1.

Security properties: In the next step, one may specify the protocol's desired properties to be verified by Tamarin. These properties are defined as trace properties, called lemmas. They may relate to the protocol's secrecy, authentication, or resistance to some malicious manipulations (i.e., replay attacks, desynchronization).

To verify the secrecy of the session key, we may specify the lemma Session_Key_Secrecy_1, listed in Figure 6.4. The lemma states that if any responder accepts the key secret_key and establishes the session with any legitimate initiator at some arbitrary time-point i, it implies that there is no time-point (in the past nor in the future) when the adversary may learn that key.

However, Tamarin disproves the lemma and presents a counterexample shown in Figure 6.5. It can be observed that the adversary simulates the initiator and sends his signature. Interestingly, the adversary also injects an insecure public key 'g' (instead of some 'g'^Aprivkey). As a result, the Responder accepts the session key of the form 'g'^Bprivkey, which is her public key.

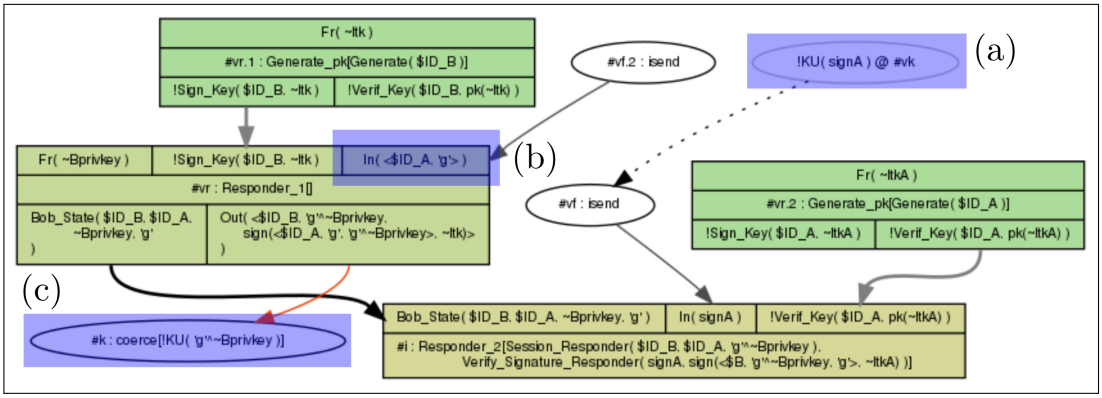


Figure 6.5 – Counterexample falsifying the lemma Session_Key_Secrecy_1. The green boxes represent the protocol rules. The specific inputs of the rules are connected with the outputs of former rules by arrows. The ellipses denote adversarial actions. In this counterexample, the adversary (a) creates invalid signature, (b) impersonates legitimate user $ID_A and inserts the invalid public key 'g', and (c) obtains the session key 'g'^Bprivkey.

The attack on the protocol security was possible because the model does not enforce the Responder's signature verification. This shortcoming, while realistic, should be taken into account in the specification of the lemma. The new proposition for the lemma can be found in Figure 6.6. In addition to previous requirements, the Responder accepts only the expected signature signed by the Initiator. Tamarin validates this new security property.

```
55  lemma Session_Key_Secrecy_2 :
56    "
57    // for any participants A and B, session key and timepoint i
58    All ID_A ID_B session_key signA #i.
59       // if the responder B completed a run presumably with the initiator A
60       Session_Responder( ID_B, ID_A, session_key ) @ i &
61       // and the signature of A is verified by the responder
62       Verify_Signature_Responder( signA, signA ) @ i
63       ==> // then
64       // the adversary never knows the session_key
65       not(Ex #k. KU( session_key ) @ k)
66     "
```

Figure 6.6 – Tamarin code of the improved security lemma Session_Key_Secrecy_2.

The presented example underscores the need for a careful evaluation of the protocol model constructed in relation to the anticipated adversarial power and protocol handling by legitimate users. For instance, one may notice that the example is still not realistic because secret keys cannot be revealed to the adversary (accidentally or intentionally). Thus, to mitigate the risk of missing out on some dangerous forms of attacks, there is a common practice to grant the adversary more considerable capabilities and knowledge than in reality. If the protocol withstands powerful adversarial manipulations, it could be suitable for implementation.

6.4 Protocol description

This section presents the symbolic model of the authenticated key exchange protocol over voice channels and provides a brief discussion.

6.4.1 Preliminaries

Let us describe the key exchange between honest users Alice and Bob, who know each other, without any legitimate trusted third-party participating. The operational framework requires that Alice and Bob must establish a non-encrypted voice connection with a preferred voice application before initiating secure communication. The system model assumes that identity information used to make a call (i.e., phone number, user account, credentials) is independent of the authentic user identity and the identification number of the voice encryption hardware. Only one running session at a time is possible since each device cannot process more than one message simultaneously. Therefore, several kinds of Denial-of-Service (DoS) attacks, when the attacker tries to send multiple messages to a recipient, are not effectively different from distorting or blocking the channel.

In highly unreliable channels like voice channels, Alice and Bob are never sure of message delivery. Thus, several synchronization techniques are needed, i.e., repeat requests, retransmissions, and time-outs. For simplicity and space limitations, most details on synchronization will be omitted here. Additionally, thanks to strong error-detection coding, users can detect random channel errors and discriminate them from intentional malicious manipulations.

6.4.2 Symbolic model of the protocol

The proposed protocol presented in Figure 6.7 relies on Ephemeral (Elliptic-Curve) Diffie-Hellman (EC)DHE exchange [Hankerson et al., 2004], authenticated by signatures (existentially unforgeable and deterministic) or Short Authentication Strings. An example of a signature algorithm suitable for use is ECDSA with the SHA256 function [ANSI, 2005]. The output of the hash function can be truncated depending on the needs [Quynh, 2012]. Before the protocol starts, Alice and Bob agree on the elliptic curve, and the lengths of keys and nonces. Public verification keys should be provided to the recipients in some authenticated way before the communication starts and stored in the Crypto Box address book. However, in many real scenarios, it is not possible to adequately provide such a verification key. If the recipient cannot verify the signature, the protocol offers vocal verification as an alternative, authenticating the speakers and the parameters used to derive the current session key.

The protocol interaction consists of several steps: the setup, the key exchange and authentication, the protocol acknowledgment, and the optional vocal verification. Table 1 contains the glossary of terms used in the protocol specification, along with their bit-lengths.

Setup: The negotiation stage has been considerably simplified. Participants have to agree on starting the key exchange procedure mutually. Therefore the actual key exchange protocol is preceded only by fast and automatic role negotiation to prevent mutual interference or logjams. Then, both Alice and Bob choose a random private integer d, a random and unique nonce N, a random value R and compute a public key Q. Unique nonce guarantees the uniqueness of the triple (ID, Q, N).

Key exchange and authentication: In this stage, Alice and Bob exchange the values that are used to obtain the Session Key (K_S) and the SAS. Alice sends her public ID, the nonce, the ephemeral public key, and the hash, with her R_A included. Bob responds with his values, appends R_B, and additionally sends his signature over all sent parameters required for K_S calculation. Alice answers with her signature over the same data and finally reveals R_A. It is worth noticing that the protocol permits a situation when the signature cannot be verified. If any of the recipients did not obtain a verification key corresponding to the sender's ID, the signature is checked against channel errors but not processed further.

Protocol acknowledgment: When all cryptographic parameters are exchanged, voice encryption can be started. Encryption is initiated after the reception of Bob's acknowledgment by Alice. The acknowledgment is a confirmation of error-less message reception so that it can be non-encrypted.

Short Authentication String comparison: Each participant can request a check of vocally challenging SAS equality with the peer. SAS comparison is obligatory if any of the users were not able to verify the signature. It is assumed that the comparison process is authenticated meaning that the users are able to recognize the voice characteristics of the peer (e.g., personal info,timbre, tempo). The SAS is displayed on the Crypto Box as a short string of digits or words to be vocally uttered by the users.

Table 6.1 – Glossary.

Acronyms	Definitions	Bits
ID_U	fixed user identifier	32
N_U	random and unique nonce	32
K_S	Session Key	256
SAS	Short Authentication String	32
$(R_A,\ R_B)$	Short Authentication String seeds	(128, 32)
$(d_U,\ Q_U)$	secret/public ECDHE key pair	(256, 256)
$(S_U,\ V_U)$	signing/verification key pair	(256, 256)
$Sign_{S_U}(\cdot)$	signature (signed with S_U)	256
$h_X(\cdot)$	hash function with truncation	X

6.5 Formal verification

This section presents the results of the protocol verification done with Tamarin Prover.

6.5.1 Protocol modeling

Verification by Tamarin implies providing an abstract protocol model that tries to express relevant information from a security perspective faithfully, but still within the analysis's feasibility. The protocol model code can be found in Annex A and online. [3] Several protocol restrictions were relaxed to make them compatible with the channel's characteristics, allowing users to run multiple protocol instantiations at the same time and to 'forget' the verification key of the peer. SAS verification is performed by a separate non-obligatory protocol rule simulating a realistic case when users simply ignore it. Vocal challenging is modeled as communicating over an authenticated (not secret) channel, which the adversary can intercept but not modify. The last ACK is skipped.

3. https://github.com/PiotrKrasnowski/AKE_over_Voice

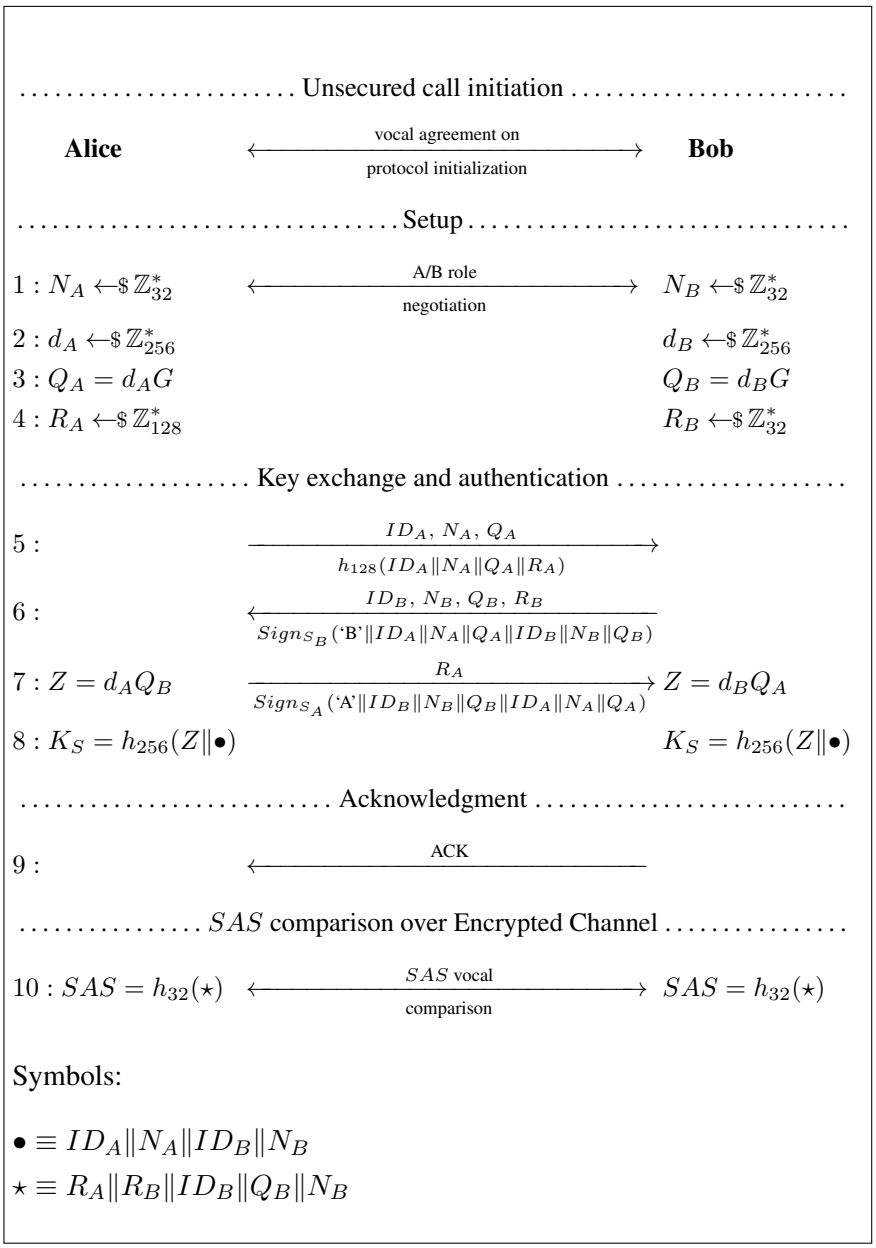


Figure 6.7 – Key exchange protocol over voice channels.

6.5.2 Security properties and verification results

The protocol model was checked against the Dolev-Yao adversary [Dolev and Yao, 1983], having full control over the network and the power to reveal the long-term secret key of any user (ephemeral data is considered secure). The evaluation was done in four authentication configurations: mutual signature authentication between two honest users, unilateral signature authentication (when only one user can verify the peer's signature), vocal verification, or no authentication.

Verification focused on most critical security properties: (perfect forward) secrecy and a mutual injective agreement [Lowe, 1997] on the Session Key. The protocol was also verified for

resilience to reflection attacks (a user cannot accept her own identity as a peer) and signing key compromise impersonation (adversary can impersonate only corrupted users). The descriptions of the properties used in specifying the security lemmas (some informally) are listed below.

Session Key Secrecy: whenever A sets up a session key, apparently with B, then the adversary cannot learn the session key unless the long-term key of A or B has been revealed.

Perfect Forward Secrecy (PFS): whenever A sets up a session key, apparently with B, then the adversary cannot learn the session key unless the long-term key of A or B has been revealed *before* the setup.

Non-injective Agreement: whenever A (acting as an initiator) completes a run of the protocol, apparently with responder B, then B has previously been running the protocol, apparently with A, and B was acting as responder in his run, and the two agents agreed on the data values corresponding to all the variables in some set of data items S.

Injective Agreement: same as non-injective agreement, plus that each run of A corresponds to a unique run of B.

Reflection Attack Resilience: whenever A completes a run of the protocol, apparently with B, then B is different than A.

Key Compromise Impersonation (KCI) Resilience: whenever the adversary corrupts a party A and reveals her long-term key, the adversary cannot impersonate another uncorrupted party.

The protocol verification results can be found in Table 6.2. The protocol configurations involving signature authentication or authenticated SAS comparison were proven to provide perfect forward secrecy and injective agreement. Unilateral signature authentication between two honest users who know each other guarantees the same security as mutual signature authentication. Surprisingly, vocal verification does not protect against reflection attack because the user can trivially compare SAS with herself. The results in Table 6.2 indicate the importance of authentication: none of the properties were verified if no authentication was performed. Crucially, the security of SAS-based authentication and unilateral signature authentication is valid only under the assumption that the peers can truly identify each other by voice.

Table 6.2 – Security properties verified by Tamarin in four authentication scenarios.

Authentication scenario:	mutual signature authentication	unilateral signature authentication	SAS vocal verification	no authentication
Session Key secrecy	✓	✓	✓	✗
forward secrecy	✓	✓	✓	✗
non-injective agreement	✓	✓	✓	✗
injective agreement	✓	✓	✓	✗
reflection attack	✓	✓	✗	✗
key compromise impersonation	✓	✓	-	-

6.6 Security considerations

This section explains in more detail some protocol characteristics, providing several justifications and practical recommendations. It starts from an overview of fundamental protocol elements: the choice of public-based cryptography, the role of signatures, and Short Authentication Strings. Later, the section enlists potential protocol weaknesses and some possible fixes.

6.6.1 Discussion

Public key agreement versus symmetric cryptography: In exceptionally constrained resource devices, such as IoT sensors or RFID cards, the pursue for ultra-lightweight key exchange protocols led to some shift from the public key encryption towards symmetric encryption techniques [Lee et al., 2014, Echevarria et al., 2016, Baashirah and Abuzneid, 2018]. Even the ZRTP protocol offers a possibility of key exchange in a lightweight preshared mode. In this configuration, two entities share a secret used to encrypt or refresh the keying material for a new session. To achieve Perfect Forward Secrecy, the long-term secret should be regularly updated, desirably after each successful key exchange run. The update decision has to be mutual, otherwise risking one-side update and user desynchronization. Unfortunately, such a risk cannot be eliminated in voice channels because the last update confirmation message may not be delivered. Decreasing the chance of desynchronization by sending more confirmation messages would negatively affect the protocol run-time. Another solution, based on on-the-fly resynchronization mechanisms, requires an online server keeping track of all key updates or a costly and potentially insecure 'guessing' of the long-term parameters until decryption is successful [Baashirah and Abuzneid, 2018]. Finally, as was emphasized before, in some scenarios, the exchange of long-term secret is impossible, limiting the usability of symmetric cryptography. In light of the reasons mentioned above and relatively smaller hardware restrictions compared to IoT sensors, a public-based key exchange scheme seems adequate.

Role of Short Authentication Strings: If the key exchange is not interfered with by a third party, both participants obtain the same Short Authentication String. Challenging SAS vocally between honest users has a twofold role. Firstly, it enables the authentication of users based on voice identification. Secondly, the inequality of codes may indicate the presence of an active MITM attacker. However, MITM manipulations would remain undetected if the attacker can somehow influence or precompute the SAS value before the users.

The code computation depends on seed values R_A and R_B chosen randomly by honest users. Importantly, Alice and Bob are forced to select seeds before knowing the value of their respective peer: Alice by sending the hash of R_A in the first message and Bob by revealing his R_B before R_A. Such a construction, inspired by [Pasini and Vaudenay, 2006], prevents adaptive selection of seeds by each party. The same rule applies to the attacker who cannot predict the SAS value until it is too late. The only hope for him is a random guess with a low probability of success or an extraction of R_A from the hash sent in the first message by brute force search. For this last reason, the length of R_A should be considerably larger than R_B. On the other hand, the difference of lengths is partially compensated by taking $Q_B \| N_B$ as an additional input of the hash function. It is worth noticing that the SAS value does not have to be confidential since it plays only an authentication role and cannot be modified without detection.

In practice, the security of vocal verification also depends on how users abide by it. The SAS could be represented by a smaller number of simple pictographs or easily pronounceable words, the same way as in the ZRTP, which has the PGP Word List incorporated into its framework [Zimmermann, 1996, Callas et al., 2011]. The device should encourage the mutual SAS comparison by indicating a part of the SAS value to pronounce and a part to hear from the peer.

Signature-based authentication: Signatures enable device authentication and message integrity, similarly to message authentication codes (MAC) which are simpler and easier to compute. Indeed, in some scenarios choosing hash-based MAC instead of signatures would be sufficient. However, signatures give wider flexibility, justifying their higher computational cost. The natural advantage of signatures is that they do not require mutual agreement and secure exchange of a long-term secret between two parties. Moreover, each user keeps in memory only one private signing key, used regardless of the receiver's identity. Consequently, if the user is corrupted, the attacker should be able to impersonate only that person.

When one user cannot obtain a verification key due to an insecure environment, it is still possible to achieve unilateral authentication [Boyd and Mathuria, 2003, Maurer et al., 2013, Dodis and Fiore, 2017]. One-side authentication prevents MITM attacks, leaving only two possibilities: honest users securely exchange a secret, or the attacker is an authenticator [Maurer et al., 2013]. It naturally implies that if the users want to communicate and they know they can perform unilateral authentication, the attacker cannot interfere undetected in another way than preventing the successful exchange. However, the user who failed to authenticate the peer is still compelled to challenge the SAS because, from her perspective, it is the only formal way to verify the absence of the MITM manipulations.

Due to a lack of any PKI infrastructure, a signature key management policy has a crucial impact on system security and usability. We will point out two possible schemes, decentralized and fully centralized, which can be chosen depending on the needs. In a centralized system, the keys are managed by an offline central authority, keeping track of all records and being responsible for key distribution and update. In a decentralized case, each user is entitled to generate her key pair and distribute public keys to specific users authentically. As in the PGP model, sharing the key can be performed remotely based on speaker identification and vocal authentication. Thus, the proposed protocol with SAS comparison gives the possibility to authenticate the exchange of signature verification keys.

6.6.2 Possible attacks and threats

Many protocol vulnerabilities focus on selecting specific cryptographic algorithms, their implementation, and compliance with the protocol rules. The biggest threat is posed by not respecting the obligation of SAS comparison by real users, opening a space for MITM attacks.

The capabilities of modern speech synthesizers which exploit AI techniques to impersonate a speaker's voice [Gao et al., 2018] question the level of authentication provided by voice recognition. Instead of breaking the SAS security, the attacker may simulate or replay the speaker pronouncing the code [Shirvanian et al., 2018]. The risk is amplified because the voice sent is highly compressed and thus significantly differs from its real characteristics. For this reason, it is recommended to extend the sequence comparison by contextual questions (like describing the last watched movie) or to share personal information known only by the peer but not by the attacker.

If honest users can verify signatures of each other and achieve strong authentication, the attacker may try a downgrade-attack. It can be done simply by modifying users' ID and imposing vocal verification. The problem may be partially solved by displaying the IDs along the SAS. However, the real solution would be to force signature verification by default.

Finally, the proposed protocol cannot protect against the consequences of a device being stolen or misused, giving the manufacturer the responsibility to provide strong enough password or biometric protection. The device should be protected against physical tampering, making reverse engineering very difficult, and minimizing the negative consequences of theft.

Alice		**Bob**
..... Unsecured call initiation and Setup		
$1 : N_{A1} \leftarrow\$\, \mathbb{Z}^*_{32}$	$\xleftarrow[\text{negotiation}]{\text{A/B role}}$	$N_{B1} \leftarrow\$\, \mathbb{Z}^*_{32}$
$2 : N_{A2} \leftarrow\$\, \mathbb{Z}^*_{128}$		$N_{B2} \leftarrow\$\, \mathbb{Z}^*_{128}$
$3 : d_A \leftarrow\$\, \mathbb{Z}^*_{256}$		$d_B \leftarrow\$\, \mathbb{Z}^*_{256}$
$4 : Q_A = d_A G$		$Q_B = d_B G$
$5 : R_A \leftarrow\$\, \mathbb{Z}^*_{128}$		$R_B \leftarrow\$\, \mathbb{Z}^*_{32}$
..... Key exchange and authentication		
$6 :$	$\xrightarrow[h_{128}(N_{A1} \Vert Q_A \Vert R_A)]{N_{A1},\, Q_A}$	$Z = d_B Q_A$
$7 : Z = d_A Q_B$	$\xleftarrow[Enc_{K_{B1}}(ID_B \Vert N_{B2})]{N_{B1},\, Q_B,\, R_B}$	$K_{B1} = h_{256}(Z \Vert \text{'B1'} \Vert N_{B1} \Vert N_{A1})$
$8 : K_A = h_{256}(Z \Vert \text{'A'} \Vert N_{A1} \Vert N_{B1})$	$\xrightarrow[Enc_{K_A}(ID_A \Vert N_{A2} \Vert Sign_{S_A}(\blacktriangle))]{R_A}$	$K_S = h_{256}(Z \Vert \bullet)$
$9 : K_S = h_{256}(Z \Vert \bullet)$	$\xleftarrow{Enc_{K_{B2}}(Sign_{S_B}(\blacktriangledown))}$	$K_{B2} = h_{256}(Z \Vert \text{'B2'} \Vert N_{B1} \Vert N_{A1})$
..... Acknowledgment		
$10 :$	$\xrightarrow{\text{ACK}}$	
..... SAS comparison over Encrypted Channel		
$11 : SAS = h_{32}(R_A \Vert R_B \Vert Q_B \Vert N_{B1})$	$\xleftrightarrow[\text{comparison}]{SAS \text{ vocal}}$	$SAS = h_{32}(R_A \Vert R_B \Vert Q_B \Vert N_{B1})$

Symbols:

$\blacktriangle \equiv ID_B \Vert N_{B2} \Vert Q_B \Vert ID_A \Vert N_{A2} \Vert Q_A$
$\blacktriangledown \equiv ID_A \Vert N_{A2} \Vert Q_A \Vert ID_B \Vert N_{B2} \Vert Q_B$
$\bullet \equiv ID_A \Vert N_{A2} \Vert ID_B \Vert N_{B2}$

Figure 6.8 – Key exchange protocol over voice channels with identity protection.

6.6.3 Protocol with identity protection

In many situations, protecting the user's identity is as crucial as securing the speech content. However, calling anybody with a civilian communication network is always associated with revealing user metadata (i.e., phone number, user credentials, location). Even if the metadata is publicly known, it may be advantageous to hide the identity of the encrypting device from passive eavesdroppers.

It is possible to redesign the proposed protocol to attain identity anonymity without the change of any other substantial protocol property mentioned in Section 6.5.2. The modified protocol is presented in Figure 6.8 and verified in Tamarin. The IDs and the signatures of Alice and Bob are sent encrypted with the key derived from a DH secret exchanged during first message round-trip, similarly to the Initial Exchange of IKEv2 standard [Kaufman et al., 2010]. The protocol includes one more message, which is required for confirming the reception of Alice's ID by Bob. On the other hand, there is no verbal agreement on the protocol initialization over an unsecured call to prevent the leakage of the speaker's identity by his or her voice profile. Instead, protocol initialization is done automatically when the call starts. Unfortunately, the proposed protocol does not protect the identity against the curious attacker who may initiate the protocol execution and break the connection once the Responder's identity is revealed in the second message.

The complexity of a protocol providing anonymity would increase since it will require additional data encryption, i.e., using AES. It is also important to carefully evaluate how the encryption key is derived and how it is related to the session key, giving no foothold for cryptanalysis. In the proposed protocol, each encrypted message is encrypted with a different key (hence K_{B1} and K_{B2}). In addition, the session key derivation uses two additional nonces N_{A2} and N_{B2}, sent encrypted. The Tamarin code modeling the protocol and security lemmas is listed in Annex B and is available online. [4]

6.7 Summary

This work attempts to bring a solution to the problem of a cryptographic key exchange over voice channels for cryptographically secure voice communications. It also introduces challenges related to secure communications over voice channels like limited available bandwidth, no guarantee of message delivery, and the issue of battery consumption. The paper lists the security requirements posed to the system, like protecting against interception and MITM attacks, emphasizing user authentication in the absence of a trusted server. All these concerns and limitations justify the need for a dedicated protocol instead of relying entirely on standardized solutions.

We proposed a simplified key exchange protocol between two honest parties based on the ephemeral elliptic curve Diffie-Hellman (ECDHE) protocol. The protocol offers two ways of authentication: signatures and Short Authentication Strings. A symbolic model of the protocol was analyzed using Tamarin Prover to verify the crucial security properties as Perfect Forward Secrecy and mutual agreement on the Session Key. The verification process was explained, pointing out the limitations of the symbolic analysis, such as model simplifications and perfect cryptography assumption.

Formal verification was followed by discussing the protocol properties, like unilateral authentication provided by one-side signature verification or the role of vocal comparison in preventing

4. https://github.com/PiotrKrasnowski/AKE_over_Voice

MITM attacks. The analysis led to the observation that all analyzed techniques do not provide perfect authentication per se. Thus, some informal identity authentication methods had to be introduced.

Potential vulnerabilities and attacks on the system were also covered in this work. Several propositions and practical solutions regarding key management, proper SAS comparison, or identity protection can guide engineers working on exchange protocols over voice channels or in similar scenarios.

The presented verification is just the first step in designing a secure protocol implementation. In a future work, it is necessary to specify the algorithms involved in the exchange and prove the protocol's security in the computational model. In the next step, the protocol should be implemented on some portable devices and tested over real-voice channels.

CHAPTER 7

Conclusion and Perspectives

This thesis addressed the problem of secure voice communications via digital voice channels. The goal of the study was to propose, investigate, and validate some new real-time speech encryption schemes adapted to error-prone transmission in the audio domain. The thesis also targeted many practical aspects of secure voice communications that would make the investigated schemes compatible with real-world voice channels.

Modern voice communication systems, such as cellular networks and VoIP, process speech signal with several adaptive algorithms that remove background noise and compress the signal before sending it. These operations are non-linear, implementation-dependent, and often unavailable to scrutiny. Consequently, sending encrypted voice as pseudo-speech over real-world voice channels remains a big challenge. The investigation of voice channels that we carried in this study helped us to identify four research problems: (1) data transmission over real voice channels, (2) distortion-tolerant enciphering of multi-dimensional data, (3) joint speech compression and encryption robust against transmission error, and (4) authenticated cryptographic key exchange over fading channels.

Secure systems rely on provably secure ciphers, verifiable cryptographic protocol models, and correct implementations. This study addressed the first two challenges by proposing a speech encryption scheme with provable security (albeit with strong assumptions on the security of pseudo-random generators and seeds) and describing a key exchange protocol verified by Tamarin Prover. Nevertheless, we tried not to miss the primary objective that is a successful deployment on real devices. Limitations of real-world voice communication systems and anticipated user experience considerably influenced the solutions proposed.

Contributions

Data over Voice (DoV) technique

The review of existing voice communication systems revealed that cellular networks and many VoIP applications rely on the classical source-filter speech model and Linear Predictive Coding (LPC). For this reason, we proposed a robust and versatile DoV technique inspired by harmonic vowels and thus compatible with LPC-based voice channels. The technique is based on codebooks of short (2.5 - 5 ms) waveforms consisting of 7-10 phase-modulated harmonics. The codebook design process is significantly simplified and boils down to constructing quaternary codes to maximize the minimum Lee distance.

The technique's robustness was initially verified by compressing produced DoV signals by three prominent LPC-based voice coders: Speex and Opus-Silk used in VoIP, and AMR adopted in cellular networks. The tests revealed that compression causes group delay in the phase-modulated harmonics, which could be efficiently compensated at the reception side.

A simple structure of DoV symbols inherited from quaternary codes gives control over the transmission rate and robustness against channel distortion. This property is crucial when considering data transmission over a real voice channel with fluctuating characteristics. An available bitrate ranging from 1 kbps to 6.4 kbps is sufficient for sending the encrypted bits of compressed speech or side information needed to establish a secure connection (e.g., cryptographic key exchange).

The technique has been successfully validated by sending synthetic DoV signals between mobile phones over 3G networks, FaceTime, Skype, WhatsApp, and Signal Messenger. The achieved transmission bitrates varied from 2.4 kbps in 3G networks up to 6.4 kbps in VoIP, with a bit error rate lower than 1%. However, correct signal synchronization is mandatory for correct signal demodulation. Furthermore, it is crucial to prevent the triggering of Voice Activity Detection (VAD). Frequency band alternation and repetitive silence insertion are possible remedies.

We also presented an experimental system for secure voice communication based on the DoV technique in high and low operation modes. The system uses Codec2 for speech compression, AES-256 encryption in the CTR mode, and shortened Reed-Solomon codes for encrypted data protection. The encrypted data are transmitted using the DoV technique at 2.4 kbps or 4.8 kbps rate, depending on the operation mode. Some tests with mobile phones and real-voice channels confirmed that the proposed system could secure voice communications over voice channels. The recordings used in the experiments are available online. [1]

Distortion-tolerant enciphering of vectors on spheres

The investigation of robust data transmission over real voice channels concluded by an observation that transmission errors could not be fully eliminated. This characteristic of voice channels undermines the usefulness of many well-established cryptographic schemes that are intolerant to error. This study introduced a notion of distortion-tolerant encryption describing the cryptographic scheme's capability to decrypt enciphered data approximately despite data distortion. The new notion is a relaxation of distance-preserving encryption designed for protecting remote databases. Due to their robustness against channel distortion, distortion-tolerant encryption schemes are suitable for sending encrypted data over voice channels. However, they are also susceptible to adversarial manipulations because it is hard to differentiate intentional data modification from random noise.

We presented a distortion-tolerant scheme for enciphering vectors on spheres, which can be used to scramble high-dimensional vectors representing vocal timbre. The proposed scheme encodes unit vectors to codewords of a dense spherical commutative group code constructed from a pair of nested lattices. Data enciphering is done by performing rotations selected from an associated group of orthogonal matrices, and relying on the output of a Pseudo-Random Number Generator (PRNG) with a secret seed of length at least 128 bits. All rotations in the group are commutative and reversible. Consequently, small transmission errors are still mapped to plaintext during decryption, which makes the system distortion-tolerant.

1. https://github.com/PiotrKrasnowski/Data_over_Voice

The encryption scheme gives indistinguishable encryptions in the presence of an eavesdropper when the source of randomness is a secure non-binary PRNG with a fresh seed. In real implementations, it is essential to ensure high-quality entropy to obtain the seed. Furthermore, the selected deterministic non-binary sequence generator must give unpredictable output up to the instantiated security strength determined by the seed. A possible solution is to construct a dedicated generator that outputs non-binary numbers. Such a dedicated generator must be carefully evaluated by well-designed statistical tests, which seems to be more challenging for non-binary sequences than binary sequences [Baigneres et al., 2007, Epishkina, 2018]. For instance, we attempted to measure the randomness of scrambled data using the Statistical Tests Suites (STS) known as diehard [2] tests developed by Prof. G. Marsaglia from FSU, USA, and dieharder [3] tests developed by Prof. R. Brown from Duke, USA. These statistical evaluations were inconclusive due to the inadequacy of binary statistical tests for non-binary sequences. Another idea is to use a trusted binary sequence generator and sequentially read its output as unsigned integers.

The toy example of color scrambling developed in that part suggests that our encryption scheme may be useful in applications different from voice encryption. However, it is mandatory to find a suitable data representation as unit vectors on spheres. The idea is to link distances on the hypersphere with perceptual similarities of encoded data. With such an approach, quantization and channel errors would have a limited impact on decrypted data perception. Consequently, distortion-tolerant encryption is applicable whenever robustness against distortion is prioritized over fidelity.

Distortion-tolerant speech encryption

We proposed an experimental speech encryption scheme for secure voice communication over voice channels. This scheme is distortion-tolerant and could operate close to real-time. The encryption algorithm encodes speech frames into energy, fundamental frequency, and the spectral envelope shape. These three parameters correspond to perceptual speech qualities: loudness, pitch, and timbre, and are crucial for preserving speech intelligibility. The sequence of energy and pitch values are scrambled using a pseudo-random sequence of translations. In contrast, the spectral envelope shapes are firstly represented as unit vectors on the hypersphere S^8, encoded to codewords of a spherical group code on S^{15} using a flat torus mapping, and finally, rotated by elements from a commutative group of 16×16 orthogonal matrices.

The encryption unit encodes scrambled parameters into a wideband speech-like signal robust to voice channel distortion. Upon reception of the signal at the receiving side, the decryption unit extracts distorted copies of the vocal parameters and tries to reconstruct the original narrowband speech. The speech synthesizer is a narrowband modification of the LPCNet algorithm with frame-rate and sample-rate neural networks, and is trained to compensate for imperfections in speech recordings and voice channel distortions.

The encryption scheme gives indistinguishable encryptions in the presence of an eavesdropper when translations and rotations are selected according to the output of a secure binary PRNG with a fresh seed. The relation between the scheme's security and the PRNG underscores the need for a trusted random bitstream generator convertible to a non-binary generator. This study points out a bitstream generator published by NIST SP 800-90A based on AES-256 in the counter mode

2. https://web.archive.org/web/20160125103112/http://stat.fsu.edu/pub/diehard/
3. https://webhome.phy.duke.edu/~rgb/General/dieharder.php

of operation and a secret seed of 256 bits. Regardless of the selected generator, however, the encryption scheme does not provide data integrity. Instead, it is mandatory to ensure a strong user authentication level, for example, during the cryptographic key exchange.

The speech encryption scheme was experimentally implemented by using the Python programming language and tested by simulations. These tests confirmed that encrypted signal is robust against additive Gaussian noise at SNR = 15 dB and compression by Opus-Silk down to 48 kbps bitrate. On the other hand, synthetic pseudo-speech is sensitive to phase-shift and synchronization error larger than 0.3 milliseconds. As a result, precise synchronization in communication over real voice channels becomes a mandatory requirement.

The preliminary computational complexity evaluation conducted on the experimental Python implementation hints that speech encoding, enciphering, and speech synthesis may be carried on high-end mobile phones nearly in real-time. However, our investigation did not cover many elements of the whole communication system, such as maintaining synchronization and generating secure random sequences. Thus, a careful complexity evaluation and optimization must precede implementation of the scheme on portable devices.

The feasibility of secure voice communication using the proposed technique was validated by real-world experiments. The precomputed pseudo-speech signal was sent over FaceTime between two iPhones 6 connected to the same WiFi network. The received signal could be decrypted into intelligible speech. Finally, we conducted an online perceptual speech quality assessment with a group of about 40 non-native English speakers. The experiment showed that the quality of decrypted speech gradually decays with growing channel distortion. The recordings used in the speech quality assessment and the pseudo-speech signals recorded on the mobile phones are available online. [4]

Authenticated Key Exchange (AKE) protocol over voice channels

Secret voice communication over a voice channel must be preceded by computing and sharing session keys in a secure and authenticated way. In some real-world scenarios, the exchange could be performed only over the same voice channel and without access to a Public Key Infrastructure. For this reason, we complemented the Ephemeral Elliptic Curve Diffie-Hellman (ECDHE) key exchange with a flexible authentication robust against channel fading. Our customized ECDHE protocol could be used together with a key derivation algorithm (e.g., PBKDF2 [Kaliski, 2000, Moriarty et al., 2017]) to produce all the necessary secret key material.

The protocol offers two ways of authentication, by cryptographic signatures (e.g., ECDSA with the SHA256 function) and by Short Authentication Strings (SAS) compared vocally (e.g., eight hexadecimal numbers displayed on the device). Both authentication mechanisms are combined into a single mode of operation, as it simplifies signaling and allows users flexible authentication such as unilateral signature authentication. The message exchange between both parties over the voice channel can be done using the introduced DoV technique.

Protocol properties such as Perfect Forward Secrecy (PFS) and injective agreement on the session key were verified in a symbolic model by Tamarin Prover. The code used for verification is available online. [5] Formal verification in the symbolic model gives higher confidence in the protocol's security. Nevertheless, the symbolic analysis has some limitations and must be followed by verification using the computational model and then by implementation auditing.

4. https://github.com/PiotrKrasnowski/Speech_Encryption
5. https://github.com/PiotrKrasnowski/AKE_over_Voice

The protocol does not prevent the issue of the speakers skipping SAS verification. Moreover, it is unclear whether a highly compressed speech signal, as in the case studied for our speech encryption scheme, contains enough paralinguistic information to enable speaker identification. These problems may be partially solved by introducing visual or acoustic effects encouraging a proper comparison and adding informal contextual questions.

System limitations and conclusion of the study

This study did not cover several implementation aspects of the whole communication system, such as signal synchronization, automatic adaptation to fluctuating channel characteristics, computational complexity, and implementation security. Nonetheless, the investigation of real voice channels and experimental results presented in this thesis strongly suggest that secure voice communication over digital voice channels is technically viable. The requirement is a reliable connection between the speakers to enable encrypted data transmission without the risk of a signal blockage and a synchronization loss.

On the other hand, the variety of speech coding techniques and implicit algorithms implemented in voice channels cast doubt on whether a universal technique compatible with any arbitrary voice channel can be found. Instead, it may be beneficial to dedicate the security system to some selected types of channel, e.g., cellular networks, VoIP, and fixed IP-phones. Despite the narrower range of targeted voice channels, the secure system should remain flexible to adapt to a particular channel and fluctuating transmission conditions. The system's flexibility could be enhanced by robust resynchronization mechanisms, adaptive bitrate, energy equalization, and countering voice detectors.

In some scenarios, requirements for connection stability, high throughput, and low channel distortion cannot be met. Thus, initiating a secure voice connection should be preceded by channel estimation. An interesting idea is to combine channel estimation with cryptographic key exchange to reduce call setup duration. A successful key exchange followed by vocal verification could become a good indicator of favorable channel conditions. Another solution is using dedicated devices that do not alter recorded speech before forwarding the signal to the voice channel input.

The encountered limitations prevent the studied voice communication system from becoming compatible with every arbitrary communication infrastructure. Nevertheless, the system can still offer some autonomy relative to particular communication infrastructure and a stronger security against spying malware. In our opinion, these advantages make the system a valuable complementary technique and justify further investigation.

Perspectives

Future work should focus on improving and extending the proposed techniques to make the next steps towards a fully operational system implementation.

The flexibility of the proposed Data over Voice technique is not yet fully exploited. It is worth experimenting with the automatic adjustment of the signal parameters depending on the fluctuating channel characteristic. For example, the transmitter may adaptively modify the harmonic frequencies and the size of the codebook used to minimize channel distortion at the receiving end. Moreover, the transmitter may determine the best strategy for countering voice detectors by a sensible combination of bandwidth alternation, amplitude variation, or silence insertion. Signal adaptation methods should significantly improve communication robustness.

An interesting direction to pursue is finding a better representation of speech parameters in speech encryption, especially of timbre. The primary objective is to find a perceptually linear speech representation that approximates distance relations in the perceptual domain. New speech representations should be investigated together with pseudo-speech synthesis techniques, so that signal distortion introduced by a voice channel would change speech perception in a quasi-linear way. One of the propositions would be to represent timbre as Mel-Frequency Cepstral Coefficients (MFCC) throughout all the processing steps.

Another research area is the use of machine learning techniques in secure speech communications. The experiments conducted so far indicate that neural-based voice synthesis improves decrypted speech quality, helps to reduce the transmission data rate, and enables compensation of distortion introduced by a particular voice channel. The multiple roles of neural networks suggest that machine learning techniques may become an essential tool in future secure voice applications. Some progress could be made by investigating new adapted network architectures, or by reducing their computational complexity for use on portable devices. Besides, an exciting idea would be to allow users to modify their voice characteristics without altering the linguistic content.

The symbolic protocol model described in this study should be extended by adding suitable ciphers and detailing the protocol's parameters to meet computational security requirements. The complete protocol model could be combined with the introduced DoV technique and implemented on prototype devices. In the final step, it may be worth combining key exchange with channel estimation in the form of training sequences.

Bibliography

[3GPP, 2017] 3GPP (2017). Universal Mobile Telecommunications System (UMTS); Multiplexing and Channel Coding (FDD). Technical Report TS 25.212, Release 15, 3GPP, https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=468.

[3GPP, 2018a] 3GPP (2018a). Adaptive Multi-Rate (AMR) speech codec ; Transcoding functions. Technical Report TS 26.090, Release 15, 3GPP, http://www.3gpp.org/ftp//Specs/archive/26_series/26.090/26090-f00.zip.

[3GPP, 2018b] 3GPP (2018b). Mandatory speech codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec; Voice Activity Detector (VAD). Technical Report TS 26.094, Release 15, 3GPP, https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=1396.

[3GPP, 2018c] 3GPP (2018c). Quality of Service (QoS) concept and architecture. Technical Report TS 26.092, Release 15, 3GPP, https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=783.

[3GPP, 2020] 3GPP (2020). LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation. Technical Report TS 36.211, Release 16, 3GPP, https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=2425.

[Ackermann et al., 2007] Ackermann, H., Mathiak, K., and Riecker, A. (2007). The contribution of the cerebellum to speech production and speech perception: clinical and functional imaging data. *The cerebellum*, 6(3):202–213, DOI: https://doi.org/10.1080/14734220701266742.

[Adoul et al., 1984] Adoul, J., Lamblin, C., and Leguyader, A. (1984). Baseband speech coding at 2400 bps using "Spherical vector quantization". In *ICASSP'84. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 9, pages 45–48. IEEE, DOI: https://doi.org/10.1109/ICASSP.1984.1172450.

[Adoul et al., 1987] Adoul, J., Mabilleau, P., Delprat, M., and Morissette, S. (1987). Fast CELP coding based on algebraic codes. In *ICASSP '87. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 12, pages 1957–1960. IEEE, DOI: https://doi.org/10.1109/ICASSP.1987.1169413.

[Agrawal et al., 2004] Agrawal, R., Kiernan, J., Srikant, R., and Xu, Y. (2004). Order Preserving Encryption for Numeric Data. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, SIGMOD '04, page 563–574, New York, NY. Association for Computing Machinery, ISBN: 1581138598, DOI: https://doi.org/10.1145/1007568.1007632.

[Alves-Pinto et al., 2014] Alves-Pinto, A., Palmer, A. R., and Lopez-Poveda, E. A. (2014). Perception and coding of high-frequency spectral notches: potential implications for sound localization. *Frontiers in neuroscience*, 8:112, DOI: https://doi.org/10.3389/fnins.2014.00112.

[ANSI, 1994] ANSI (1994). ANSI/ASA S1.1-1994, American National Standard Acoustical Terminology.

[ANSI, 2005] ANSI (2005). Public Key Cryptography for the Financial Services Industry: the Elliptic Curve Digital Signature Algorithm (ECDSA). Technical report, American National Standards Institute.

[Arkko et al., 2004] Arkko, J., Carrara, E., Lindholm, F., Norrman, K., and Naslund, M. (2004). Mikey: Multimedia Internet KEYing (RFC3830). Technical report, IETF, https://tools.ietf.org/html/rfc3830.

[Armando et al., 2005] Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuéllar, J., Drielsma, P. H., Héam, P.-C., Kouchnarenko, O., Mantovani, J., et al. (2005). The AVISPA Tool for the Automated Validation of Internet Security Protocols and Applications. In *International Conference on Computer Aided Verification*, pages 281–285. Springer, DOI: https://doi.org/10.1007/11513988_27.

[Aucouturier and Bigand, 2012] Aucouturier, J.-J. and Bigand, E. (2012). Mel Cepstrum & Ann Ova: The Difficult Dialog Between MIR and Music Cognition. *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012*, pages 397–402.

[Baashirah and Abuzneid, 2018] Baashirah, R. and Abuzneid, A. (2018). Survey on prominent RFID authentication protocols for passive tags. *Sensors*, 18(10):3584, DOI: https://doi.org/10.3390/s18103584.

[Bäckström, 2017] Bäckström, T. (2017). *Speech Coding with Code-Excited Linear Prediction*. Springer, Cham, Switzerland.

[Baigneres et al., 2007] Baigneres, T., Stern, J., and Vaudenay, S. (2007). Linear cryptanalysis of non binary ciphers. In *International Workshop on Selected Areas in Cryptography*, pages 184–211. Springer, DOI: https://doi.org/10.1007/978-3-540-77360-3_13.

[Barker, 2020] Barker, E. (2020). Recommendation for Key Management: Part 1 - General. Technical report, National Institute of Standards and Technology, DOI: https://doi.org/10.6028/NIST.SP.800-57pt1r5.

[Barker et al., 2015] Barker, E., Feldman, L., and Witte, G. (2015). Recommendation for Random Number Generation Using Deterministic Random Bit Generators. Technical report, National Institute of Standards and Technology, https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf.

[Beauchamp et al., 1993] Beauchamp, J. W., Maher, R. C., and Brown, R. (1993). Detection of Musical Pitch from Recorded Solo Performances. In *Audio Engineering Society Convention 94*. Audio Engineering Society, http://www.aes.org/e-lib/browse.cfm?elib=6616.

[Benesty et al., 2008] Benesty, J., Sondhi, M. M., and Huang, Y. (2008). *Handbook of Speech Processing*. Springer, Berlin Heidelberg, Germany.

[Bennett, 1983] Bennett, W. (1983). Secret Telephony as a Historical Example of Spread-Spectrum Communication. *IEEE Transactions on Communications*, 31(1):98–104, DOI: https://doi.org/10.1109/TCOM.1983.1095724.

[Biard and Noguet, 2008] Biard, L. and Noguet, D. (2008). Reed-Solomon Codes for Low Power Communications. *Journal of Communications*, 3, DOI: https://doi.org/10.4304/jcm.3.2.13-21.

[Biham and Dunkelman, 2000] Biham, E. and Dunkelman, O. (2000). Cryptanalysis of the A5/1 GSM stream cipher. In *International Conference on Cryptology in India*, pages 43–51. Springer, DOI: https://doi.org/10.1007/3-540-44495-5_5.

[Blanchet, 2001] Blanchet, B. (2001). An efficient cryptographic protocol verifier based on prolog rules. In *Proceedings. 14th IEEE Computer Security Foundations Workshop, 2001.*, pages 82–96. IEEE, DOI: https://doi.org/10.1109/CSFW.2001.930138.

[Blanchet, 2009] Blanchet, B. (2009). Automatic Verification of Correspondences for Security Protocols. *Journal of Computer Security*, 17(4):363–434, DOI: https://doi.org/10.3233/JCS-2009-0339.

[Blanchet, 2012] Blanchet, B. (2012). Security protocol verification: Symbolic and computational models. In *International Conference on Principles of Security and Trust*, pages 3–29. Springer, DOI: https://doi.org/10.1007/978-3-642-28641-4_2.

[Boloursaz et al., 2013] Boloursaz, M., Hadavi, A. H., Kazemi, R., and Behnia, F. (2013). A data modem for GSM Adaptive Multi Rate voice channel. In *East-West Design Test Symposium (EWDTS 2013)*, pages 1–4. IEEE, DOI: https://doi.org/10.1109/EWDTS.2013.6673152.

[Boucheron et al., 2011] Boucheron, L. E., De Leon, P. L., and Sandoval, S. (2011). Low Bit-Rate Speech Coding Through Quantization of Mel-Frequency Cepstral Coefficients. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(2):610–619, DOI: https://doi.org/10.1109/TASL.2011.2162407.

[Boyd and Mathuria, 2003] Boyd, C. and Mathuria, A. (2003). *Protocols for authentication and key establishment*, volume 1. Springer, Berlin, Germany.

[Brandenburg et al., 1992] Brandenburg, K., Eberlein, E., Herre, J., and Edler, B. (1992). Comparison of filterbanks for high quality audio coding. In *[Proceedings] 1992 IEEE International Symposium on Circuits and Systems*, volume 3, pages 1336–1339. IEEE, DOI: https://doi.org/10.1109/ISCAS.1992.230257.

[Burr, 1989] Burr, A. G. (1989). Spherical codes for M-ary code shift keying. In *Second IEE National Conference on Telecommunications 1989*, pages 67–72. IET, https://ieeexplore.ieee.org/document/20683.

[Callas et al., 2011] Callas, J., Johnston, A., and Zimmermann, P. (2011). ZRTP: Media path key agreement for unicast secure RTP. Technical Specification RFC 6189, IETF, https://tools.ietf.org/html/rfc6189.

[Campbell, 1997] Campbell, J. P. (1997). Speaker recognition: a tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, DOI: https://doi.org/10.1109/5.628714.

[Candel and Conlon, 2000] Candel, A. and Conlon, L. (2000). Foliations. I, volume 23 of Graduate Studies in Mathematics. *American Mathematical Society, Providence, RI*, 5.

[Canetti and Krawczyk, 2001] Canetti, R. and Krawczyk, H. (2001). Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 453–474. Springer, DOI: https://doi.org/10.1007/3-540-44987-6_28.

[Chae et al., 2015] Chae, C.-J., Shin, Y., Choi, K., Kim, K.-B., and Choi, K.-N. (2015). A privacy data leakage prevention method in P2P networks. *Peer-to-Peer Networking and Applications*, 9, DOI: https://doi.org/10.1007/s12083-015-0371-x.

[Chazan et al., 2000] Chazan, D., Hoory, R., Cohen, G., and Zibulski, M. (2000). Speech reconstruction from mel frequency cepstral coefficients and pitch frequency. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 3, pages 1299–1302. IEEE, DOI: https://doi.org/10.1109/ICASSP.2000.861816.

[Chen and Liang, 2007] Chen, H. and Liang, H. (2007). Combined selective mapping and binary cyclic codes for PAPR reduction in OFDM systems. *IEEE Transactions on Wireless Communications*, 6, DOI: https://doi.org/10.1109/TWC.2007.060145.

[Chen and Guo, 2011] Chen, L. and Guo, Q. (2011). An OFDM-based secure data communicating scheme in GSM voice channel. In *2011 International Conference on Electronics, Communications and Control (ICECC)*. IEEE, DOI: https://doi.org/10.1109/ICECC.2011.6066715.

[Childers and Wu, 1991] Childers, D. G. and Wu, K. (1991). Gender recognition from speech. Part II: Fine analysis. *The Journal of the Acoustical society of America*, 90(4):1841–1856, DOI: https://doi.org/10.1121/1.401664.

[Chmayssani and Baudoin, 2008] Chmayssani, T. and Baudoin, G. (2008). Data transmission over voice dedicated channels using digital modulations. In *2008 18th International Conference Radioelektronika*. IEEE, DOI: https://doi.org/10.1109/RADIOELEK.2008.4542682.

[Cohen, 1971] Cohen, A. (1971). *The phonemes of English*. Springer, Dodrecht, The Netherlands, ISBN: 978-94-010-2969-8.

[Cohen, 1993] Cohen, H. (1993). *A course in computational algebraic number theory*. Springer-Verlag, Berlin, Heidelberg, Germany.

[Conway and Sloane, 1982] Conway, J. and Sloane, N. (1982). Fast quantizing and decoding and algorithms for lattice quantizers and codes. *IEEE Transactions on Information Theory*, 28(2):227–232, DOI: https://doi.org/10.1109/TIT.1982.1056484.

[Conway and Sloane, 1999] Conway, J. H. and Sloane, N. J. A. (1999). *Sphere packings, lattices and groups - 3rd Edition*. Springer Science & Business Media, New York, NY.

[Costa et al., 2017] Costa, S. I., Oggier, F., Campello, A., Belfiore, J.-C., and Viterbo, E. (2017). *Lattices Applied to Coding for Reliable and Secure Communications*. Springer Nature, Cham, Switzerland.

[Coupé et al., 2019] Coupé, C., Oh, Y. M., Dediu, D., and Pellegrino, F. (2019). Different languages, similar encoding efficiency: Comparable information rates across the human communicative niche. *Science advances*, 5(9):eaaw2594, DOI: https://doi.org/10.1126/sciadv.aaw2594.

[Cox and McDaniel, 1989] Cox, R. M. and McDaniel, D. M. (1989). Development of the Speech Intelligibility Rating (SIR) test for hearing aid comparisons. *Journal of Speech, Language, and Hearing Research*, 32(2):347–352, DOI: https://doi.org/10.1044/jshr.3202.347.

[Cremers, 2008] Cremers, C. J. (2008). The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols. In *International Conference on Computer Aided Verification*, pages 414–418. Springer, DOI: https://doi.org/10.1007/978-3-540-70545-1_38.

[Davis and Jedwab, 1999] Davis, J. A. and Jedwab, J. (1999). Peak-to-mean power control in OFDM, Golay complementary sequences, and Reed-Muller codes. *IEEE Transactions on Information Theory*, 45, DOI: https://doi.org/10.1109/18.796380.

[Davis and Mermelstein, 1980] Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366, DOI: https://doi.org/10.1109/TASSP.1980.1163420.

[Denes and Pinson, 1993] Denes, P. and Pinson, E. (1993). *The speech chain - the physics and biology of spoken language*. W.H. Freeman and Co., New York, NY.

[Dhananjay et al., 2010] Dhananjay, A., Sharma, A., Paik, M., Chen, J., Kuppusamy, T. K., Li, J., and Subramanian, L. (2010). Hermes: data transmission over unknown voice channels. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. Association for Computing Machinery, DOI: https://doi.org/10.1145/1859995.1860010.

[Dodis and Fiore, 2017] Dodis, Y. and Fiore, D. (2017). Unilaterally-Authenticated Key Exchange. In *International Conference on Financial Cryptography and Data Security*, pages 542–560. Springer, DOI: https://doi.org/10.1007/978-3-319-70972-7_31.

[Dolev and Yao, 1983] Dolev, D. and Yao, A. (1983). On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208, DOI: https://doi.org/10.1109/TIT.1983.1056650.

[Dudley, 1939] Dudley, H. (1939). Remaking speech. *The Journal of the Acoustical Society of America*, 11(2):169–177, DOI: https://doi.org/10.1121/1.1916020.

[Durgin et al., 2004] Durgin, N., Lincoln, P., Mitchell, J., and Scedrov, A. (2004). Multiset Rewriting and the Complexity of Bounded Security Protocols. *Journal of Computer Security*, 12(2):247–311, DOI: https://doi.org/10.3233/JCS-2004-12203.

[Duric et al., 2004] Duric, A., Telio, and Andersen, S. (2004). Real-time Transport Protocol (RTP) Payload Format for internet Low Bit Rate Codec (iLBC) Speech. Technical Specification RFC 3952, IETF, https://tools.ietf.org/html/rfc3952.

[Eberspächer et al., 2008] Eberspächer, J., Vögel, H.-J., Bettstetter, C., and Hartmann, C. (2008). *GSM-architecture, protocols and services*. John Wiley & Sons, Chichester, UK.

[Echevarria et al., 2016] Echevarria, J. J., Legarda, J., Larrañaga, J., and Ruiz-de Garibay, J. (2016). lwAKE: A lightweight Authenticated Key Exchange for Class 0 Devices. *International Journal of Distributed Sensor Networks*, 12(5):6236494, DOI: https://doi.org/10.1155/2016/6236494.

[Edwards and Chang, 2013] Edwards, E. and Chang, E. F. (2013). Syllabic (2–5 hz) and fluctuation (1–10 hz) ranges in speech and auditory processing. *Hearing research*, 305:113–134, DOI: https://doi.org/10.1016/j.heares.2013.08.017.

[Elliott and Theunissen, 2009] Elliott, T. M. and Theunissen, F. E. (2009). The modulation transfer function for speech intelligibility. *PLoS comput biol*, 5(3):e1000302, DOI: https://doi.org/10.1371/journal.pcbi.1000302.

[Epishkina, 2018] Epishkina, A. (2018). A technique to test non-binary random number generator. *Procedia computer science*, 145:193–198, DOI: https://doi.org/10.1016/j.procs.2018.11.039.

[Erhardt et al., 2019] Erhardt, S., Kurin, T., Lurz, F., Weigel, R., and Koelpin, A. (2019). An Open-Source Speech Codec at 450 bit/s with Pseudo-Wideband Mode. In *2019 49th European Microwave Conference (EuMC)*, pages 1048–1051. IEEE.

[ETSI, 2000] ETSI (2000). Digital cellular telecommunications system (Phase 2+); Half rate speech; Half rate speech transcoding. Technical Report EN 300 969, V8.01, ETSI, https://www.etsi.org/deliver/etsi_en/300900_300999/300969/08.00.01_60/en_300969v080001p.pdf.

[ETSI, 2018a] ETSI (2018a). Mandatory speech codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec; Comfort noise aspects. Technical Report TS 126 092 V15.0.0, ETSI, https://portal.etsi.org/webapp/WorkProgram/Report_WorkItem.asp?WKI_ID=61153.

[ETSI, 2018b] ETSI (2018b). Minimum performance requirements for Noise Suppresser; Application to the Adaptive Multi-Rate (AMR) speech encoder. Technical Report TS 126 077 V15.0.0, ETSI, https://portal.etsi.org/webapp/workprogram/Report_WorkItem.asp?WKI_ID=55656.

[ETSI, 2018c] ETSI (2018c). Performance characterization of the Adaptive Multi-Rate (AMR) speech codec. Technical Report TR 126 975 V15.0.0, ETSI, https://www.etsi.org/deliver/etsi_tr/126900_126999/126975/15.00.00_60/tr_126975v150000p.pdf.

[Fant, 1960] Fant, G. (1960). *Acoustic Theory of Speech Production: With Calculations based on X-Ray Studies of Russian Articulations*. De Gruyter Mouton, Berlin, Boston, DOI: https://doi.org/10.1515/9783110873429.

[Farrell, 2009] Farrell, S. (2009). Why didn't we spot that? [Practical Security]. *IEEE Internet Computing*, 14(1):84–87, DOI: https://doi.org/10.1109/MIC.2010.21.

[Fastl and Zwicker, 2006] Fastl, H. and Zwicker, E. (2006). *Psychoacoustics: facts and models*, volume 22. Springer Science & Business Media, Berlin Heidelberg, Germany.

[Fitch and Giedd, 1999] Fitch, W. and Giedd, J. (1999). Morphology and development of the human vocal tract: A study using magnetic resonance imaging. *The Journal of the Acoustical Society of America*, 106:1511–22, DOI: https://doi.org/10.1121/1.427148.

[Flanagan, 1972] Flanagan, J. (1972). *Speech Analysis Synthesis and Perception*. Springer-Verlag, Berlin Heidelberg, Germany, DOI: https://doi.org/10.1007/978-3-662-01562-9.

[Fletcher and Munson, 1933] Fletcher, H. and Munson, W. A. (1933). Loudness, Its Definition, Measurement and Calculation. *The Journal of the Acoustical Society of America*, 5(2):82–108, DOI: https://doi.org/10.1121/1.1915637.

[Forgie, 1975] Forgie, J. W. (1975). Speech transmission in packet-switched store-and-forward networks. In *Proceedings of the May 19-22, 1975, national computer conference and exposition*, pages 137–142. Association for Computing Machinery, DOI: https://doi.org/10.1145/1499949.1499978.

[Forney, 1991] Forney, G. D. (1991). Geometrically uniform codes. *IEEE Transactions on Information Theory*, 37(5):1241–1260, DOI: https://doi.org/10.1109/18.133243.

[Freeman et al., 1989] Freeman, D., Cosier, G., Southcott, C., and Boyd, I. (1989). The voice activity detector for the Pan-European digital cellular mobile telephone service. In *International Conference on Acoustics, Speech, and Signal Processing,*, pages 369–372. IEEE, DOI: https://doi.org/10.1109/ICASSP.1989.266442.

[Furui, 1996] Furui, S. (1996). An Overview of Speaker Recognition Technology. *Automatic speech and speaker recognition*, 355:31–56, DOI: https://doi.org/10.1007/978-1-4613-1367-0_2.

[Gantmakher, 1959] Gantmakher, F. R. (1959). *The theory of matrices, vol 1*. Chelsea Publishing Company, New York, NY.

[Gao et al., 2018] Gao, Y., Singh, R., and Raj, B. (2018). Voice impersonation using generative adversarial networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2506–2510. IEEE, DOI: https://doi.org/10.1007/11745853_26.

[Gay et al., 1974] Gay, T., Ushijima, T., Hiroset, H., and Cooper, F. S. (1974). Effect of speaking rate on labial consonant-vowel articulation. *Journal of Phonetics*, 2(1):47–63, DOI: https://doi.org/10.1016/S0095-4470(19)31176-3.

[Gentry, 2009] Gentry, C. (2009). Fully Homomorphic Encryption Using Ideal Lattices. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, page 169–178, New York, NY. Association for Computing Machinery, ISBN: 9781605585062, DOI: https://doi.org/10.1145/1536414.1536440.

[Gerson and Jasiuk, 1990] Gerson, I. A. and Jasiuk, M. A. (1990). Vector sum excited linear prediction (VSELP) speech coding at 8 kbps. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 461–464. IEEE, DOI: https://doi.org/10.1109/ICASSP.1990.115749.

[Ghasemzadeh et al., 2014] Ghasemzadeh, H., Mehrara, H., and Khas, M. T. (2014). Ciphertext only attack on hopping window time domain scramblers. In *2014 4th International Conference on Computer and Knowledge Engineering (ICCKE)*, pages 194–199. IEEE, DOI: https://doi.org/10.1109/ICCKE.2014.6993428.

[Ginige et al., 2001] Ginige, T., Rajatheva, N., and Ahmed, K. M. (2001). Dynamic spreading code selection method for PAPR reduction in OFDM-CDMA systems with 4-QAM modulation. *IEEE Communications Letters*, 5, DOI: https://doi.org/10.1109/4234.957377.

[Gold et al., 2011] Gold, B., Morgan, N., and Ellis, D. (2011). *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*. John Wiley & Sons, Hoboken, NJ.

[Goldburg et al., 1993] Goldburg, B., Sridharan, S., and Dawson, E. (1993). Cryptanalysis of frequency domain analogue speech scramblers. *IEE Proceedings I (Communications, Speech and Vision)*, 140(4):235–239, DOI: https://doi.org/10.1049/ip-i-2.1993.0035.

[Goldburg et al., 1993] Goldburg, B., Sridharan, S., and Dawson, E. (1993). Design and cryptanalysis of transform-based analog speech scramblers. *IEEE Journal on Selected Areas in Communications*, 11(5):735–744, DOI: https://doi.org/10.1109/49.223875.

[Golden et al., 2006] Golden, P., Dedieu, H., and Jacobsen, K. (2006). *Fundamentals of DSL technology*. Auerbach Publications, Boca Raton, FL.

[Goldstein, 1973] Goldstein, J. L. (1973). An optimum processor theory for the central formation of the pitch of complex tones. *The Journal of the Acoustical Society of America*, 54(6):1496–1516, DOI: https://doi.org/10.1121/1.1914448.

[Goldwasser and Micali, 1984] Goldwasser, S. and Micali, S. (1984). Probabilistic Encryption. *Journal of computer and system sciences*, 28(2):270–299, DOI: https://doi.org/10.1016/0022-0000(84)90070-9.

[Goudarzi et al., 2011] Goudarzi, M., Sun, L., and Ifeachor, E. (2011). Modelling speech quality for NB and WB SILK codec for VoIP applications. In *2011 Fifth International Conference on Next Generation Mobile Applications, Services and Technologies*, pages 42–47. IEEE, DOI: https://doi.org/10.1109/NGMAST.2011.18.

[Goyal, 2001] Goyal, V. K. (2001). Multiple description coding: Compression meets the network. *IEEE Signal processing magazine*, 18(5):74–93, DOI: https://doi.org/10.1109/79.952806.

[Gray, 1984] Gray, R. (1984). Vector quantization. *IEEE ASSP Magazine*, 1(2):4–29, DOI: 10.1109/MASSP.1984.1162229.

[Griffin and Lim, 1988] Griffin, D. W. and Lim, J. S. (1988). Multiband excitation vocoder. *IEEE Transactions on acoustics, speech, and signal processing*, 36(8):1223–1235, DOI: https://doi.org/10.1109/29.1651.

[Gupta, 1999] Gupta, G. (1999). *Horn Logic Denotations and Their Applications*, pages 127–159. Springer, Berlin, Heidelberg, DOI: https://doi.org/10.1007/978-3-642-60085-2_6.

[Hamkins and Zeger, 2002] Hamkins, J. and Zeger, K. (2002). Gaussian source coding with spherical codes. *IEEE Transactions on Information Theory*, 48(11):2980–2989, DOI: https://doi.org/10.1109/TIT.2002.804056.

[Hanani et al., 2013] Hanani, A., Russell, M. J., and Carey, M. J. (2013). Human and computer recognition of regional accents and ethnic groups from British English speech. *Computer Speech & Language*, 27(1):59–74, DOI: https://doi.org/10.1016/j.csl.2012.01.003.

[Hankerson et al., 2004] Hankerson, D., Menezes, A., and Vanstone, S. (2004). *Guide to elliptic curve cryptography*. Springer, New York, NY.

[Hardwick and Lim, 1991] Hardwick, J. C. and Lim, J. S. (1991). The application of the IMBE speech coder to mobile communications. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, pages 249–252. IEEE Computer Society, DOI: https://doi.org/10.1109/ICASSP.1991.150324.

[Hauer, 2015] Hauer, B. (2015). Data and Information Leakage Prevention Within the Scope of Information Security. *IEEE Access*, 3:2554–2565, DOI: https://doi.org/10.1109/ACCESS.2015.2506185.

[Heitkamper, 1995] Heitkamper, P. (1995). Optimization of an Acoustic Echo Canceller Combined with Adaptive Gain Control. In *1995 International Conference on Acoustics, Speech, and Signal Processing*. DOI: 10.1109/ICASSP.1995.479488.

[Heitkamper and Walker, 1993] Heitkamper, P. and Walker, M. (1993). Adaptive gain control for speech quality improvement and echo suppression. In *1993 IEEE International Symposium on Circuits and Systems*, pages 455–458. IEEE, DOI: https://doi.org/10.1109/ISCAS.1993.393756.

[Hellige, 1994] Hellige, H. D. (1994). From SAGE via Arpanet to Ethernet: Stages in computer communications concepts between 1950 and 1980. *History and Technology*, 11(1):49–75, DOI: https://doi.org/10.1080/07341519408581854.

[Hellman, 1972] Hellman, R. P. (1972). Asymmetry of masking between noise and tone. *Perception & Psychophysics*, 11(3):241–246, DOI: https://doi.org/10.3758/BF03206257.

[Herlein et al., 2009] Herlein, G., Valin, J.-M., Heggestad, A., and Moizard, A. (2009). RTP Payload Format for the Speex Codec. Technical Specification RFC 5574, IETF, https://tools.ietf.org/html/rfc5574.

[Hisojo et al., 2014] Hisojo, M. A., Lebrun, J., and Deneire, L. (2014). Low PAPR and spatial diversity for OFDM schemes by using L2-orthogonal CPM ST-codes with fast decoding. In *2014 IEEE Latin-America Conference on Communications (LATINCOM)*, pages 1–6. IEEE, DOI: 10.1109/LATINCOM.2014.7041850.

[Holma and Toskala, 2005] Holma, H. and Toskala, A. (2005). *WCDMA for UMTS: Radio access for Third Generation Mobile Communications*. John Wiley & Sons, Chichester, UK.

[Housley, 2004] Housley, R. (2004). Using Advanced Encryption Standard (AES) counter mode with IPsec encapsulating security payload (ESP). Technical Specification RFC 3686, IETF, https://tools.ietf.org/html/rfc3686.

[Huang et al., 2001] Huang, X., Acero, A., Hon, H.-W., and Reddy, R. (2001). *Spoken language processing: A guide to Theory, Algorithm, and System Development*. Prentice Hall PTR, Upper Saddle River, NJ.

[ISO, 2003] ISO (2003). *ISO 226:2003(en) Acoustics — Normal equal-loudness-level contours*. International Organization for Standardization, Geneva, Switzerland.

[ISO/IEC, 1998] ISO/IEC (1998). *ISO/IEC 13818-3, Information technology – Generic coding of moving pictures and associated audio information – Part 3: Audio*. International Organization for Standardization, Geneva, Switzerland.

[ISO/IEC, 2006] ISO/IEC (2006). *ISO/IEC 13818-7, Information technology – Generic coding of moving pictures and associated audio information – Part 7: Advanced Audio Coding (AAC)*. International Organization for Standardization, Geneva, Switzerland.

[ITU-R, 2015] ITU-R (2015). Method for the subjective assessment of intermediate quality level of audio systems. Technical Report Recommendation BS.1534-3, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/R-REC-BS.1534.

[ITU-T, 1988a] ITU-T (1988a). 1200 Bits Per Second Duplex Modem Standardized for Use in the General Switched Telephone Network and on Point-to-Point 2-Wire Leased Telephone-Type Circuits. Technical Report Recommendation V.22, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-V.22/en.

[ITU-T, 1988b] ITU-T (1988b). 2400 Bits Per Second Duplex Modem Using the Frequency Division Technique Standardized for Use on the General Switched Telephone Network and on Point-to-Point 2-Wire Leased Telephone-Type Circuits. Technical Report Recommendation V.22bis, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-V.22bis/en.

[ITU-T, 1988c] ITU-T (1988c). 300 Bits Per Second Duplex Modem Standardized for Use in General Switched Telephone Network. Technical Report Recommendation V.21, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-V.21/en.

[ITU-T, 1988d] ITU-T (1988d). 4800 Bits Per Second Modem with Manual Equalizer Standardized for Use on Leased Telephone-Type Circuits. Technical Report Recommendation V.27, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-V.27-198811-I/en.

[ITU-T, 1988e] ITU-T (1988e). 9600 Bits Per Second Modem Standardized for Use on Point-to-Point 4-Wire Leased Telephone-Type Circuits. Technical Report Recommendation V.29, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-V.29/en.

[ITU-T, 1988f] ITU-T (1988f). Pulse Code Modulation (PCM) of Voice Frequencies. Technical Report Recommendation G.711, International Telecommunication Union, Geneva, Switzerland.

[ITU-T, 1990] ITU-T (1990). Adaptive Differential Pulse Code Modulation (ADPCM). Technical Report Recommendation G.726, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-G.726/en.

[ITU-T, 1993] ITU-T (1993). Integrated Services Digital Networks (ISDNs). Technical Report Recommendation I.120, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-I.120/en.

[ITU-T, 1996a] ITU-T (1996a). Methods for subjective determination of transmission quality. Technical Report Recommendation P.800, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-P.800-199608-I.

[ITU-T, 1996b] ITU-T (1996b). Modulated Noise Reference Unit (MNRU). Technical Report Recommendation P.810, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-P.810/en.

[ITU-T, 1998] ITU-T (1998). A Modem Operating at Data Signalling Rates of up to 33600 bit/s for Use on the General Switched Telephone Network and on Point-to-Point 2-Wire Leased Telephone-Type Circuits. Technical Report Recommendation V.34, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-V.34/en.

[ITU-T, 2001] ITU-T (2001). Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs. Technical Report Recommendation P.862, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-P.862.

[ITU-T, 2003a] ITU-T (2003a). International telephone connections and circuits - General Recommendations on the transmission quality for an entire international telephone connection; One-way transmission time. Technical Report Recommendation G.114, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-G.114-200305-I/en.

[ITU-T, 2003b] ITU-T (2003b). Talker echo and its control. Technical Report Recommendation G.131, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-G.131/en.

[ITU-T, 2012a] ITU-T (2012a). 7 kHz audio-coding within 64 kbit/s. Technical Report Recommendation G.722, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-G.722.

[ITU-T, 2012b] ITU-T (2012b). Coding of Speech at 8 kbit/s Using Conjugate-Structure Algebraic-Code-Excited Linear Prediction (CS-ACELP). Technical Report Recommendation G.729, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-G.729.

[ITU-T, 2012c] ITU-T (2012c). Wideband coding of speech at around 16 kbit/s using Adaptive Multi-Rate Wideband (AMR-WB). Technical Report Recommendation G.722.2, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-G.722.2/en.

[ITU-T, 2015] ITU-T (2015). Digital echo cancellers. Technical Report Recommendation G.168, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-G.168/en.

[ITU-T, 2016a] ITU-T (2016a). Mean opinion score (MOS) terminology. Technical Report Recommendation P.801, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-P.800.1.

[ITU-T, 2016b] ITU-T (2016b). Subjective test methodology for assessing speech intelligibility. Technical Report Recommendation P.807, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-P.807/en.

[ITU-T, 2018] ITU-T (2018). Perceptual objective listening quality prediction. Technical Report Recommendation P.863, International Telecommunication Union, Geneva, Switzerland, https://www.itu.int/rec/T-REC-P.863.

[Jayant and Rabiner, 1972] Jayant, N. and Rabiner, L. (1972). The application of dither to the quantization of speech signals. *Bell System Technical Journal*, 51(6):1293–1304, DOI: https://doi.org/10.1002/j.1538-7305.1972.tb02653.x.

[Jayant et al., 1983] Jayant, N. S., Cox, R. V., McDermott, B. J., and Quinn, A. (1983). Analog scramblers for speech based on sequential permutations in time and frequency. *Bell System Technical Journal*, 62(1):25–46, DOI: https://doi.org/10.1002/j.1538-7305.1983.tb04377.x.

[Jonsson, 2003] Jonsson, J. (2003). On the security of CTR+ CBC-MAC. In *Selected Areas in Cryptography*. Springer, DOI: https://doi.org/10.1007/3-540-36492-7_7.

[Just and Vaudenay, 1996] Just, M. and Vaudenay, S. (1996). Authenticated multi-party key agreement. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 36–49. Springer, DOI: https://doi.org/10.1007/BFb0034833.

[Juvela et al., 2018] Juvela, L., Bollepalli, B., Wang, X., Kameoka, H., Airaksinen, M., Yamagishi, J., and Alku, P. (2018). Speech Waveform Synthesis from MFCC Sequences with Generative Adversarial Networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5679–5683. IEEE, DOI: https://doi.org/10.1109/ICASSP.2018.8461852.

[Kahn, 1996] Kahn, D. (1996). *The Codebreakers: The comprehensive history of secret communication from ancient times to the internet*. The Macmillan Company, New York, NY.

[Kak, 1983] Kak, S. C. (1983). Overview of analogue signal encryption. *IEE Proceedings F - Communications, Radar and Signal Processing*, 130(5):399–404, DOI: https://doi.org/10.1049/ip-f-1.1983.0066.

[Kaliski, 2000] Kaliski, B. (2000). PKCS# 5: Password-based cryptography specification version 2.0. Technical Specification RFC 2898, IETF, https://tools.ietf.org/html/rfc2898.

[Kaliski, 1984] Kaliski, B. S. (1984). Wyner's Analog Encryption Scheme: Results of a Simulation. In *Workshop on the Theory and Application of Cryptographic Techniques*, pages 83–94. Springer, DOI: https://doi.org/10.1007/3-540-39568-7_9.

[Kallman, 1982] Kallman, H. J. (1982). Octave equivalence as measured by similarity ratings. *Perception & Psychophysics*, 32(1):37–49, DOI: https://doi.org/10.3758/BF03204867.

[Käsper and Schwabe, 2009] Käsper, E. and Schwabe, P. (2009). Faster and timing-attack resistant AES-GCM. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 1–17. Springer, DOI: https://doi.org/10.1007/978-3-642-04138-9_1.

[Katugampala et al., 2003] Katugampala, N., Villette, S., and Kondoz, A. (2003). Secure voice over GSM and other low bit rate systems. In *IEE Seminar on Secure GSM and Beyond (Digest No. 2003/10059)*. IET.

[Katugampala et al., 2005] Katugampala, N. N., Al-Naimi, K. T., Villette, S., and Kondoz, A. M. (2005). Real-time end-to-end secure voice communications over GSM voice channel. In *2005 13th European Signal Processing Conference*, pages 1–4. IEEE, ISBN: 978-160-4238-21-1.

[Katz and Lindell, 2015] Katz, J. and Lindell, Y. (2015). *Introduction to modern cryptography*. CRC press, Boca raton, FL.

[Kaufman et al., 2010] Kaufman, C., Hoffman, P., Nir, Y., and Eronen, P. (2010). Internet Key Exchange Protocol Version 2 (IKEv2) (RFC7296). Technical report, IETF, fromhttps://tools.ietf.org/html/rfc7296.

[Kazemi et al., 2015] Kazemi, R., Mashhadi, M. B., Khoozani, M. H., and Behnia, F. (2015). Modem based on sphere packing techniques in high-dimensional Euclidian sub-space for efficient data over voice communication through mobile voice channels. *IET Communications*, 9, DOI: https://doi.org/10.1049/iet-com.2014.0610.

[Kelly and Lochbaum, 1962] Kelly, J. and Lochbaum, C. (1962). Speech synthesis. In *Proceedings of the Fourth International Congress on Acoustics*, pages 1–4, Copenhagen, DK.

[Kiang and Moxon, 1974] Kiang, N. and Moxon, E. (1974). Tails of tuning curves of auditory-nerve fibers. *The Journal of the Acoustical Society of America*, 55(3):620–630, DOI: https://doi.org/10.1121/1.1914572.

[Kleijn, 1991] Kleijn, W. (1991). Speech coding below 4 kb/s using waveform interpolation. In *IEEE Global Telecommunications Conference GLOBECOM'91: Countdown to the New Millennium. Conference Record*, pages 1879–1883. IEEE, DOI: https://doi.org/10.1109/GLOCOM.1991.188688.

[Kleijn et al., 2018] Kleijn, W. B., Lim, F. S. C., Luebs, A., Skoglund, J., Stimberg, F., Wang, Q., and Walters, T. C. (2018). Wavenet Based Low Rate Speech Coding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 676–680. DOI: https://doi.org/10.1109/ICASSP.2018.8462529.

[Kocarev et al., 1992] Kocarev, L., Halle, K. S., Eckert, K., Chua, L. O., and Parlitz, U. (1992). Experimental demonstration of secure communications via chaotic synchronization. *International Journal of Bifurcation and Chaos*, 2(03):709–713, DOI: https://doi.org/10.1142/S0218127492000823.

[Kolly and Dellwo, 2014] Kolly, M.-J. and Dellwo, V. (2014). Cues to linguistic origin: The contribution of speech temporal information to foreign accent recognition. *Journal of Phonetics*, 42:12–23, DOI: https://doi.org/10.1016/j.wocn.2013.11.004.

[Kondoz, 2004] Kondoz, A. M. (2004). *Digital Speech: Coding for Low Bit Rate Communication Systems*. John Wiley & Sons, Chichester, UK.

[Kramer, 1991] Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. *AIChE journal*, 37(2):233–243, DOI: https://doi.org/10.1002/aic.690370209.

[Krasnowski et al., 2020] Krasnowski, P., Lebrun, J., and Martin, B. (2020). Introducing a Verified Authenticated Key Exchange Protocol over Voice Channels for Secure Voice Communication. In *6th International Conference on Information Systems Security and Privacy*. Scitepress Digital Library, DOI: https://doi.org/10.5220/0009156506830690.

[Kruskal and Wallis, 1952] Kruskal, W. H. and Wallis, W. A. (1952). Use of Ranks in One-Criterion Variance Analysis. *Journal of the American Statistical Association*, 47(260):583–621, DOI: https://doi.org/10.1080/01621459.1952.10483441.

[Kubin and Kleijn, 1999] Kubin, G. and Kleijn, W. B. (1999). On speech coding in a perceptual domain. In *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*, volume 1, pages 205–208. IEEE, DOI: https://doi.org/10.1109/ICASSP.1999.758098.

[LaDue et al., 2008] LaDue, C. K., Sapozhnykov, V. V., and Fienberg, K. S. (2008). A Data Modem for GSM Voice Channel. *IEEE Transactions on Vehicular Technology*, 57(4):2205–2218, DOI: https://doi.org/10.1109/TVT.2007.912322.

[Lauter and Mityagin, 2006] Lauter, K. and Mityagin, A. (2006). Security analysis of KEA authenticated key exchange protocol. In *International Workshop on Public Key Cryptography*, pages 378–394. Springer, DOI: https://doi.org/10.1007/11745853_25.

[Lavor and Gomes, 2018] Lavor, C. and Gomes, F. A. (2018). *Advances in Mathematics and Applications: Celebrating 50 Years of the Institute of Mathematics, Statistics and Scientific Computing, University of Campinas*. Springer, Cham, Switzerland.

[Lawson Jr, 1974] Lawson Jr, H. B. (1974). Foliations. *Bulletin of the American Mathematical Society*, 80(3):369–418, DOI: https://doi.org/10.1090/S0002-9904-1974-13432-4.

[Lee, 1981] Lee, J.-S. (1981). Speckle analysis and smoothing of synthetic aperture radar images. *Computer Graphics and Image Processing*, 17(1):24 – 32, ISSN: 0146-664X, DOI: https://doi.org/10.1016/S0146-664X(81)80005-6.

[Lee et al., 2014] Lee, J.-Y., Lin, W.-C., and Huang, Y.-H. (2014). A lightweight authentication protocol for Internet of Things. In *2014 International Symposium on Next-Generation Electronics (ISNE)*, pages 1–2. IEEE, DOI: https://doi.org/10.1109/ISNE.2014.6839375.

[Lee et al., 2017] Lee, S., Ha, Y., Yoon, S., Jo, H., Jang, S., Lee, J., Kim, Y., and Yoon, J. (2017). The Vulnerability Exploitation Conveying Digital Data Over Mobile Voice Call Channels. *Wireless Personal Communications*, 96:1–28, DOI: https://doi.org/10.1007/s11277-017-4229-9.

[Li et al., 2013] Li, M., Han, K. J., and Narayanan, S. (2013). Automatic speaker age and gender recognition using acoustic and prosodic level information fusion. *Computer Speech & Language*, 27(1):151–167, DOI: https://doi.org/10.1016/j.csl.2012.01.008.

[Lin and Costello, 2001] Lin, S. and Costello, D. J. (2001). *Error control coding, Second Edition*. Prentice Hall, Lebanon, IN.

[Lin et al., 2008] Lin, X. S., Khong, A. W., Doroslovăcki, M., and Naylor, P. A. (2008). Frequency-domain adaptive algorithm for network echo cancellation in VoIP. *EURASIP Journal on Audio, Speech, and Music Processing*, 2008(1):156960, DOI: https://doi.org/10.1155/2008/156960.

[Lin-Shan Lee et al., 1984] Lin-Shan Lee, Ger-Chih Chou, and Ching-Sung Chang (1984). A New Frequency Domain Speech Scrambling System Which Does Not Require Frame Synchronization. *IEEE Transactions on Communications*, 32(4):444–456, DOI: https://doi.org/10.1109/TCOM.1984.1096078.

[Lindblom and Studdert-Kennedy, 1967] Lindblom, B. and Studdert-Kennedy, M. (1967). On the R ole of Formant Transitions in Vowel Recognition. *The Journal of the Acoustical Society of America*, 42(4):830–43, DOI: https://doi.org/10.1121/1.1910655.

[Lipmaa et al., 2000] Lipmaa, H., Rogaway, P., and Wagner, D. (2000). CTR-mode encryption. In *1st NIST Workshop on Modes of Operation*, volume 39.

[Lochbaum and Kelly, 1962] Lochbaum, C. and Kelly, J. (1962). Speech synthesis. In *Proceedings of the Speech Communication Seminar*, pages 583–596. Speech Transmission Laboratory.

[Lowe, 1996] Lowe, G. (1996). Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *International Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, pages 147–166. Springer, DOI: https://doi.org/10.1007/3-540-61042-1_43.

[Lowe, 1997] Lowe, G. (1997). A hierarchy of authentication specifications. In *Proceedings 10th Computer Security Foundations Workshop*, pages 31–43. IEEE, DOI: https://doi.org/10.1109/CSFW.1997.596782.

[Lyon, 1982] Lyon, R. (1982). A computational model of filtering, detection, and compression in the cochlea. In *ICASSP'82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 7, pages 1282–1285. IEEE, DOI: https://doi.org/10.1109/ICASSP.1982.1171644.

[MacKinnon, 1980] MacKinnon, N. (1980). The development of speech encipherment. *Radio and Electronic Engineer*, 50:147–155(8), DOI: https://doi.org/10.1049/ree.1980.0022.

[Makhoul, 1975] Makhoul, J. (1975). Linear prediction: A tutorial review. *Proceedings of the IEEE*, 63(4):561–580, DOI: https://doi.org/10.1109/PROC.1975.9792.

[Mardia, 1970] Mardia, K. V. (1970). Measures of multivariate skewness and kurtosis with applications. *Biometrika*, 57, DOI: https://doi.org/10.2307/2334770.

[Mardia, 1974] Mardia, K. V. (1974). Applications of Some Measures of Multivariate Skewness and Kurtosis in Testing Normality and Robustness Studies. *Sankhyā: The Indian Journal of Statistics, Series B*, 32(2):115–128, https://www.jstor.org/stable/pdf/25051892.pdf.

[Maurer et al., 2013] Maurer, U., Tackmann, B., and Coretti, S. (2013). Key Exchange with Unilateral Authentication: Composable Security Definition and Modular Protocol Design. *IACR Cryptology, ePrint Archive*, https://eprint.iacr.org/2013/555.pdf.

[McAulay and Quatieri, 1986] McAulay, R. and Quatieri, T. (1986). Speech analysis/synthesis based on a sinusoidal representation. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 34(4):744–754, DOI: https://doi.org/10.1109/TASSP.1986.1164910.

[McCree et al., 1996] McCree, A., Truong, K., George, E. B., Barnwell, T. P., and Viswanathan, V. (1996). A 2.4 kbit/s MELP coder candidate for the new US Federal Standard. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 200–203. IEEE, DOI: https://doi.org/10.1109/ICASSP.1996.540325.

[Meier et al., 2013] Meier, S., Schmidt, B., Cremers, C., and Basin, D. (2013). The TAMARIN prover for the symbolic analysis of security protocols. In *International Conference on Computer Aided Verification*, pages 696–701. Springer, DOI: https://doi.org/10.1007/978-3-642-39799-8_48.

[Mendonça and Delikaris-Manias, 2018] Mendonça, C. and Delikaris-Manias, S. (2018). Statistical Tests with MUSHRA Data. In *Audio Engineering Society Convention 144*. Audio Engineering Society, http://www.aes.org/e-lib/browse.cfm?elib=19402.

[Metze et al., 2007] Metze, F., Ajmera, J., Englert, R., Bub, U., Burkhardt, F., Stegmann, J., Muller, C., Huber, R., Andrassy, B., Bauer, J. G., et al. (2007). Comparison of Four Approaches to Age and Gender Recognition for Telephone Applications. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, volume 4, pages IV–1089. IEEE, DOI: https://doi.org/10.1109/ICASSP.2007.367263.

[Mezgec et al., 2009] Mezgec, Z., Chowdhury, A., Kotnik, B., and Svečko, R. (2009). Implementation of PCCD-OFDM-ASK robust data transmission over GSM speech channel. *Informatica*, 20, DOI: https://doi.org/10.15388/Informatica.2009.237.

[Micciancio and Goldwasser, 2012] Micciancio, D. and Goldwasser, S. (2012). *Complexity of lattice problems: a cryptographic perspective*, volume 671. Springer Science & Business Media, Ney York, NY.

[Miller et al., 1986] Miller, J. L., Green, K. P., and Reeves, A. (1986). Speaking rate and segments: A look at the relation between speech production and speech perception for the voicing contrast. *Phonetica*, 43(1-3):106–115, DOI: https://doi.org/10.1159/000261764.

[Milner and Shao, 2006] Milner, B. and Shao, X. (2006). Clean speech reconstruction from MFCC vectors and fundamental frequency using an integrated front-end. *Speech Communication*, 48(6):697–715, DOI: https://doi.org/10.1016/j.specom.2005.10.004.

[Moriarty et al., 2017] Moriarty, K., Kaliski, B., and Rusch, A. (2017). PKCS#5: Password-Based Cryptography Specification Version 2.1. Technical Specification RFC 8018, IETF, https://tools.ietf.org/html/rfc8018.

[Morris et al., 2009] Morris, B., Rogaway, P., and Stegers, T. (2009). How to Encipher Messages on a Small Domain. In *Advances in Cryptology - CRYPTO 2009*. Springer, DOI: https://doi.org/10.1007/978-3-642-03356-8_17.

[Müller, 1840] Müller, J. P. (1840). *Handbuch der Physiologie des Menschen: für Vorlesungen. Bd. 2*, volume 2. J. Hölscher.

[Nee and Prasad, 2000] Nee, R. v. and Prasad, R. (2000). *OFDM for Wireless Multimedia Communications*. Artech House, Boston, MT.

[Neubauer et al., 2007] Neubauer, A., Freudenberger, J., and Kuhn, V. (2007). *Coding Theory: Algorithms, Architectures and Applications*. John Wiley & Sons, Chichester, UK.

[Nielsen et al., 2014] Nielsen, J. K., Christensen, M. G., Cemgil, A. T., and Jensen, S. H. (2014). Bayesian Model Comparison With the g-Prior. *IEEE Transactions on Signal Processing*, 62(1):225–238, DOI: https://doi.org/10.1109/TSP.2013.2286776.

[Nielsen et al., 2017] Nielsen, J. K., Jensen, T. L., Jensen, J. R., Christensen, M. G., and Jensen, S. H. (2017). Fast fundamental frequency estimation: Making a statistically efficient estimator computationally efficient. *Signal Processing*, 135:188 – 197, ISSN: 0165-1684, DOI: https://doi.org/10.1016/j.sigpro.2017.01.011.

[Nwe et al., 2003] Nwe, T. L., Foo, S. W., and De Silva, L. C. (2003). Speech emotion recognition using hidden Markov models. *Speech communication*, 41(4):603–623, DOI: https://doi.org/10.1016/S0167-6393(03)00099-2.

[Okabe et al., 2000] Okabe, A., Boots, B., Sugihara, K., and Nok Chiu, S. (2000). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley, Hoboken, NJ, ISBN: 978-0-471-98635-5.

[O'Neill, 2014] O'Neill, M. E. (2014). PCG: A family of simple fast space-efficient statistically good algorithms for random number generation. *ACM Transactions on Mathematical Software*.

[Oord et al., 2016] Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, https://arxiv.org/abs/1609.03499.

[Özkan and Örs, 2015] Özkan, M. A. and Örs, S. B. (2015). Data transmission via GSM voice channel for end to end security. In *2015 IEEE 5th International Conference on Consumer Electronics-Berlin (ICCE-Berlin)*. IEEE, DOI: https://doi.org/10.1109/ICCE-Berlin.2015.7391285.

[Panayotov et al., 2015] Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210. IEEE, DOI: https://doi.org/10.1109/ICASSP.2015.7178964.

[Park and Lee, 2018] Park, J. H. and Lee, D. H. (2018). FACE: fast AES CTR mode encryption techniques based on the reuse of repetitive data. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 469–499, DOI: https://doi.org/10.13154/tches.v2018.i3.469-499.

[Park et al., 2011] Park, J.-H., Paik, J.-H., and Lee, D.-H. (2011). Efficient implementation of AES CTR Mode for a Mobile Environment. *Journal of the KIISC*, 21(5):47–58, https://www.koreascience.or.kr/article/JAKO201109649106054.pdf.

[Pasini and Vaudenay, 2006] Pasini, S. and Vaudenay, S. (2006). SAS-Based Authenticated Key Agreement. In *Public Key Cryptography - PKC 2006*, pages 395–409. Springer, DOI: https://doi.org/10.1007/11745853_26.

[Patro et al., 2011] Patro, A., Ma, Y., Panahi, F., Walker, J., and Banerjee, S. (2011). A system for audio signalling based NAT Traversal. In *2011 Third International Conference on Communication Systems and Networks (COMSNETS 2011)*, pages 1–10. IEEE, DOI: https://doi.org/10.1109/COMSNETS.2011.5716432.

[Princen et al., 1987] Princen, J., Johnson, A., and Bradley, A. (1987). Subband/transform coding using filter bank designs based on time domain aliasing cancellation. In *ICASSP'87. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 12, pages 2161–2164. IEEE, DOI: https://doi.org/10.1109/ICASSP.1987.1169405.

[Ptacek and Sander, 1966] Ptacek, P. H. and Sander, E. K. (1966). Age recognition from voice. *Journal of speech and hearing Research*, 9(2):273–277, DOI: https://doi.org/10.1044/jshr.0902.273.

[Purves et al., 2018] Purves, D., Augustine, G. J., Flitzpatrick, D. A., Hall, W. C., LaMantia, A.-S., Mooney, R. D., Platt, M. L., and White, L. E. (2018). *Neuroscience, 6th Edition*. Sunderland: Sinauer Associates, Inc, Sunderland, MA, ISBN: 978-16-053-5380-7.

[Quynh, 2012] Quynh, D. (2012). Recommendation for Applications Using Approved Hash Algorithms. Technical report, National Institute of Standards and Technology, https://www.nist.gov/publications/recommendation-applications-using-approved-hash-algorithms.

[Rabiner and Schafer, 2011] Rabiner, L. R. and Schafer, R. W. (2011). *Theory and applications of digital speech processing*. Pearson, Upper Saddle River, NJ.

[Ramasubramanian and Doddala, 2015] Ramasubramanian, V. and Doddala, H. (2015). *Ultra Low Bit-Rate Speech Coding*. Springer, New York, NY, DOI: https://doi.org/10.1007/978-1-4939-1341-1.

[Rashidi et al., 2008] Rashidi, M., Sayadiyan, A., and Mowlaee, P. (2008). A Harmonic Approach to Data Transmission over GSM Voice Channel. In *2008 3rd International Conference on Information and Communication Technologies: From Theory to Applications*, pages 1–4. IEEE, DOI: https://doi.org/10.1109/ICTTA.2008.4530052.

[Reddi et al., 2018] Reddi, S. J., Kale, S., and Kumar, S. (2018). On the Convergence of Adam and Beyond. In *International Conference on Learning Representations*. OpenReview.net, https://openreview.net/forum?id=ryQu7f-RZ.

[Rescorla, 2018] Rescorla, E. (2018). The Transport Layer Security (TLS) Protocol Version 1.3. Technical report, IETF, https://tools.ietf.org/html/rfc8446.

[Rogers, 1881] Rogers, J. B. (1881). Telephony. 251,292.

[Rothweiler, 1983] Rothweiler, J. (1983). Polyphase quadrature filters–a new subband coding technique. In *ICASSP'83. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 8, pages 1280–1283. IEEE, DOI: https://doi.org/10.1109/ICASSP.1983.1172005.

[Rusinowitch and Turuani, 2003] Rusinowitch, M. and Turuani, M. (2003). Protocol insecurity with a finite number of sessions and composed keys is NP-complete.

Theoretical Computer Science, 299(1):451 – 475, ISSN: 0304-3975, DOI: https://doi.org/10.1016/S0304-3975(02)00490-5.

[Salami et al., 1998] Salami, R., Laflamme, C., Adoul, J., Kataoka, A., Hayashi, S., Moriya, T., Lamblin, C., Massaloux, D., Proust, S., Kroon, P., and Shoham, Y. (1998). Design and description of CS-ACELP: a toll quality 8 kb/s speech coder. *IEEE Transactions on Speech and Audio Processing*, 6(2):116–130, DOI: https://doi.org/10.1109/89.661471.

[Sapozhnykov and Fienberg, 2012] Sapozhnykov, V. V. and Fienberg, K. S. (2012). A low-rate data transfer technique for compressed voice channels. *Journal of Signal Processing Systems*, DOI: https://doi.org/10.1007/s11265-011-0594-x.

[Schmidt et al., 2012] Schmidt, B., Meier, S., Cremers, C., and Basin, D. (2012). Automated Analysis of Diffie-Hellman Protocols and Advanced Security Properties. In *2012 IEEE 25th Computer Security Foundations Symposium*, pages 78–94. IEEE, DOI: https://doi.org/10.1109/CSF.2012.25.

[Schmidt et al., 2014] Schmidt, B., Sasse, R., Cremers, C., and Basin, D. (2014). Automated Verification of Group Key Agreement Protocols. In *2014 IEEE Symposium on Security and Privacy*, pages 179–194. IEEE, DOI: https://doi.org/10.1109/SP.2014.19.

[Schoeffler et al., 2018] Schoeffler, M., Bartoschek, S., Stöter, F.-R., Roess, M., Westphal, S., Edler, B., and Herre, J. (2018). webMUSHRA—A comprehensive framework for web-based listening tests. *Journal of Open Research Software*, 6(1), DOI: https://doi.org/10.5334/jors.187.

[Schoeffler et al., 2015] Schoeffler, M., Stöter, F.-R., Edler, B., and Herre, J. (2015). Towards the next generation of web-based experiments: A case study assessing basic audio quality following the ITU-R recommendation BS. 1534 (MUSHRA). In *1st Web Audio Conference*, pages 1–6. https://wac.ircam.fr/pdf/wac15_submission_8.pdf.

[Schroeder and Atal, 1985] Schroeder, M. and Atal, B. (1985). Code-excited linear prediction (CELP): High-quality speech at very low bit rates. In *ICASSP '85. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 10, pages 937–940. IEEE, DOI: https://doi.org/10.1109/ICASSP.1985.1168147.

[Schulze and Lüders, 2005] Schulze, H. and Lüders, C. (2005). *Theory and Applications of OFDM and CDMA: Wideband Wireless Communications*. John Wiley & Sons, Chichester, GB.

[Schwartz, 2005] Schwartz, M. (2005). *Mobile Wireless Communications*. Cambridge University Press, Cambridge, UK.

[Scott-Railton et al., 2017] Scott-Railton, J., Marczak, B., Razzak, B. A., Crete-Nishihata, M., and Deibert, R. (2017). Reckless Exploit: Mexican Journalists, Lawyers, and a Child Targeted with NSO Spyware. Report, The Citizen Lab, https://tspace.library.utoronto.ca/bitstream/1807/96731/1/Report%2393--recklessexploit.pdf. Accessed 13 July 2020.

[Sethares, 2004] Sethares, W. A. (2004). *Tuning, Timbre, Spectrum, Scale*. Springer, Berlin Heidelberg, Germany.

[Shahbazi et al., 2010] Shahbazi, A., Rezaei, A. H., Sayadiyan, A., and Mosayyebpour, S. (2010). Data transmission over GSM adaptive multi rate voice channel using speech-like symbols. In *2010 International Conference on Signal Acquisition and Processing*. IEEE, DOI: https://doi.org/10.1109/ICSAP.2010.72.

[Shahbazi et al., 2009] Shahbazi, A., Rezaie, A. H., Sayadiyan, A., and Mosayyebpour, S. (2009). A novel speech-like symbol design for data transmission through GSM voice channel. In *2009 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*. IEEE, DOI: https://doi.org/10.1109/ISSPIT.2009.5407541.

[Shannon, 1949] Shannon, C. E. (1949). Communication theory of secrecy systems. *The Bell system technical journal*, 28(4):656–715, DOI: https://doi.org/10.1002/j.1538-7305.1949.tb00928.x.

[Shirvanian et al., 2018] Shirvanian, M., Saxena, N., and Mukhopadhyay, D. (2018). Short voice imitation man-in-the-middle attacks on Crypto Phones: Defeating humans and machines. *Journal of Computer Security*, 26:311 – 333, DOI: https://doi.org/10.3233/JCS-17970.

[Siedenburg et al., 2016] Siedenburg, K., Fujinaga, I., and McAdams, S. (2016). A Comparison of Approaches to Timbre Descriptors in Music Information Retrieval and Music Psychology. *Journal of New Music Research*, 45(1):27–41, DOI: https://doi.org/10.1080/09298215.2015.1132737.

[Siqueira and Costa, 2008] Siqueira, R. M. and Costa, S. I. (2008). Flat tori, lattices and bounds for commutative group codes. *Designs, Codes and Cryptography*, 49(1-3):307–321, DOI: https://doi.org/10.1007/s10623-008-9183-9.

[Sivian, 1928] Sivian, L. J. (1928). System for Secret Signaling. 1,654,900.

[Slepian, 1968] Slepian, D. (1968). Group codes for the Gaussian channel. *Bell System Technical Journal*, 47(4):575–602, DOI: https://doi.org/10.1002/j.1538-7305.1968.tb02486.x.

[Slepian and Pollak, 1961] Slepian, D. and Pollak, H. O. (1961). Prolate spheroidal wave functions, Fourier analysis and uncertainty—I. *Bell System Technical Journal*, 40(1):43–63, DOI: https://doi.org/10.1002/j.1538-7305.1961.tb03976.x.

[Soong and Juang, 1984] Soong, F. and Juang, B. (1984). Line spectrum pair (LSP) and speech data compression. In *ICASSP '84. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 9, pages 37–40. IEEE, DOI: https://doi.org/10.1109/ICASSP.1984.1172448.

[Stefanov and Shi, 2012] Stefanov, E. and Shi, E. (2012). FastPRP: Fast Pseudo-Random Permutations for Small Domains. *IACR Cryptology ePrint Report 2012/254*, https://eprint.iacr.org/2012/254.pdf.

[Stevens, 1956] Stevens, S. S. (1956). The direct estimation of sensory magnitudes: Loudness. *The American journal of psychology*, 69(1):1–25, DOI: https://doi.org/10.2307/1418112.

[Stevens et al., 1937] Stevens, S. S., Volkmann, J., and Newman, E. B. (1937). A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, DOI: https://doi.org/10.1121/1.1915893.

[Sugar, 1974] Sugar, G. R. (1974). *Voice Privacy Equipment for Law Enforcement Communication Systems*. National Institute of Law Enforcement and Criminal Justice.

[Syrdal and Gopal, 1986] Syrdal, A. and Gopal, H. (1986). A perceptual model of vowel recognition based on the auditory representation of American English vowels. *The Journal of the Acoustical Society of America*, 79(4):1086–1100, DOI: https://doi.org/10.1121/1.393381.

[Szczerba and Czyzewski, 2005] Szczerba, M. and Czyzewski, A. (2005). Pitch detection enhancement employing music prediction. *Journal of Intelligent Information Systems*, 24(2-3):223–251, DOI: https://doi.org/10.1007/s10844-005-0324-6.

[Tahir et al., 2017] Tahir, B., Schwarz, S., and Rupp, M. (2017). BER comparison between convolutional, Turbo, LDPC, and Polar codes. In *2017 24th international conference on telecommunications (ICT)*. IEEE, DOI: https://doi.org/10.1109/ICT.2017.7998249.

[Taleb Ali et al., 2013] Taleb Ali, B., Baudoin, G., and Venard, O. (2013). Data transmission over mobile voice channel based on M-FSK modulation. In *2013 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 4416–4421. IEEE, DOI: https://doi.org/10.1109/WCNC.2013.6555289.

[Tasko and Westbury, 2004] Tasko, S. M. and Westbury, J. R. (2004). Speed–curvature relations for speech-related articulatory movement. *Journal of Phonetics*, 32(1):65–80, DOI: https://doi.org/10.1016/S0095-4470(03)00006-8.

[Terasawa et al., 2012] Terasawa, H., Berger, J., and Makino, S. (2012). In Search of a Perceptual Metric for Timbre: Dissimilarity Judgments among Synthetic Sounds with MFCC-Derived Spectral Envelopes. *J. Audio Eng. Soc*, 60(9):674–685, http://www.aes.org/e-lib/browse.cfm?elib=16372.

[Terasawa et al., 2005] Terasawa, H., Slaney, M., and Berger, J. (2005). A timbre space for speech. In *INTERSPEECH 2005 - Eurospeech, 9th European Conference on Speech Communication and Technology*, pages 1729–1732. ISCA, https://www.isca-speech.org/archive/archive_papers/interspeech_2005/i05_1729.pdf.

[Terpstra et al., 2010] Terpstra, D., Jagode, H., You, H., and Dongarra, J. (2010). Collecting Performance Data with PAPI-C. In Müller, M. S., Resch, M. M., Schulz, A., and Nagel, W. E., editors, *Tools for High Performance Computing 2009*, pages 157–173, Berlin, Heidelberg. Springer Berlin Heidelberg, DOI: https://doi.org/10.1007/978-3-642-11261-4_11.

[Tex et al., 2018] Tex, C., Schäler, M., and Böhm, K. (2018). Towards meaningful distance-preserving encryption. In *Proceedings of the 30th International Conference on Scientific and Statistical Database Management*, pages 1–12. Association for Computing Machinery, DOI: https://doi.org/10.1145/3221269.3223029.

[Torezzan et al., 2013] Torezzan, C., Costa, S. I. R., and Vaishampayan, V. A. (2013). Constructive Spherical Codes on Layers of Flat Tori. *IEEE Transactions on Information Theory*, 59(10):6655–6663, DOI: https://doi.org/10.1109/TIT.2013.2272931.

[Torezzan et al., 2015] Torezzan, C., Strapasson, J. E., Costa, S. I., and Siqueira, R. M. (2015). Optimum commutative group codes. *Designs, Codes and Cryptography*, 74(2):379–394, DOI: https://doi.org/10.1007/s10623-013-9867-7.

[Tsay and Mjølsnes, 2012] Tsay, J.-K. and Mjølsnes, S. F. (2012). A Vulnerability in the UMTS and LTE Authentication and Key Agreement Protocols. In *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security*, pages 65–76. Springer, DOI: https://doi.org/10.1007/978-3-642-33704-8_6.

[Tsoukalas et al., 1997] Tsoukalas, D. E., Mourjopoulos, J. N., and Kokkinakis, G. (1997). Speech Enhancement Based on Audible Noise Suppression. *IEEE Transactions on Speech and Audio Processing*, 5, DOI: https://doi.org/10.1109/89.641296.

[Utkovski and Lindner, 2006] Utkovski, Z. and Lindner, J. (2006). On The Construction of Non-coherent Space Time Codes from High-dimensional Spherical Codes. In *2006 IEEE Ninth International Symposium on Spread Spectrum Techniques and Applications*, pages 327–331. IEEE, DOI: https://doi.org/10.1109/ISSSTA.2006.311788.

[Valin and Skoglund, 2019] Valin, J. and Skoglund, J. (2019). LPCNET: Improving Neural Speech Synthesis through Linear Prediction. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5891–5895. IEEE, DOI: https://doi.org/10.1109/ICASSP.2019.8682804.

[Valin and Skoglund, 2019] Valin, J.-M. and Skoglund, J. (2019). A Real-Time Wideband Neural Vocoder at 1.6kb/s Using LPCNet. In *Proceedings of INTERSPEECH*, pages 3406–3410. International Speech Communication Association, DOI: https://doi.org/10.21437/Interspeech.2019-1255.

[Valin et al., 2012] Valin, J.-M., Vos, K., and Terriberry, T. (2012). Definition of the Opus Audio Codec. Technical Specification RFC 6176, IETF, https://tools.ietf.org/html/rfc6716.

[Venkataramani et al., 2003] Venkataramani, R., Kramer, G., and Goyal, V. K. (2003). Multiple description coding with many channels. *IEEE Transactions on Information Theory*, 49(9):2106–2114, DOI: https://doi.org/10.1109/TIT.2003.815767.

[Vernam, 1919] Vernam, G. S. (1919). Secret Signaling System. 1,310,719.

[Vernam, 1926] Vernam, G. S. (1926). Cipher printing telegraph systems: For secret wire and radio telegraphic communications. *Journal of the AIEE*, 45(2):109–115, DOI: https://doi.org/10.1109/T-AIEE.1926.5061224.

[Viazovska, 2017] Viazovska, M. S. (2017). The sphere packing problem in dimension 8. *Annals of Mathematics*, pages 991–1015, DOI: https://doi.org/10.4007/annals.2017.185.3.7.

[Vogt and André, 2006] Vogt, T. and André, E. (2006). Improving Automatic Emotion Recognition from Speech via Gender Differentiaion. In *Proc. Language Resources and Evaluation Conference (LREC 2006)*, pages 1123–1126. Multimodale Mensch-Technik Interaktion, http://www.lrec-conf.org/proceedings/lrec2006/pdf/392_pdf.pdf.

[Von Békésy and Wever, 1960] Von Békésy, G. and Wever, E. G. (1960). *Experiments in hearing*, volume 8. McGraw-Hill, New York, NJ.

[Šeděnka and Gasti, 2014] Šeděnka, J. and Gasti, P. (2014). Privacy-Preserving Distance Computation and Proximity Testing on Earth, Done Right. In *Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security*, ASIA CCS '14, page 99–110, New York, NY. Association for Computing Machinery, DOI: https://doi.org/10.1145/2590296.2590307.

[Wah and Dong Lin, 2005] Wah, B. W. and Dong Lin (2005). LSP-based multiple-description coding for real-time low bit-rate voice over IP. *IEEE Transactions on Multimedia*, 7(1):167–178, DOI: https://doi.org/10.1109/TMM.2004.840593.

[Weinstein and Ebert, 1971] Weinstein, S. and Ebert, P. (1971). Data transmission by frequency-division multiplexing using the discrete fourier transform. *IEEE Transactions on Communication Technology*, 19(5):628–634, DOI: https://doi.org/10.1109/TCOM.1971.1090705.

[Werner et al., 2009] Werner, M., Pietsch, C., Joetten, C., Sgraja, C., Frank, G., Granzow, W., and Huang, J. (2009). Cellular In-Band Modem Solution for eCall Emergency Data Transmission. In *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*. IEEE, DOI: https://doi.org/10.1109/VETECS.2009.5073434.

[Wilkinson and Jones, 1995] Wilkinson, T. A. and Jones, A. E. (1995). Minimisation of the Peak to Mean Envelope Power Ratio of Multicarrier Transmission Schemes by Block Coding. In *1995 IEEE 45th Vehicular Technology Conference. Countdown to the Wireless Twenty-First Century*, volume 2. IEEE, DOI: https://doi.org/10.1109/VETEC.1995.504983.

[Wilson et al., 2004] Wilson, S. M., Saygin, A. P., Sereno, M. I., and Iacoboni, M. (2004). Listening to speech activates motor areas involved in speech production. *Nature neuroscience*, 7:701–702, DOI: https://doi.org/10.1038/nn1263.

[Wisayataksin, 2019] Wisayataksin, S. (2019). An Efficient Hardware Architecture of Codec2 Low Bit-rate Speech Decoder. In *2019 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*. IEEE, DOI: https://doi.org/10.1109/ICEAST.2019.8802570.

[Witte and Witte, 2017] Witte, R. and Witte, J. (2017). *Statisticss*. Wiley, Hoboken, NJ, ISBN: 978-1-119-25451-5.

[Wold et al., 1987] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1):37 – 52, ISSN: 0169-7439, DOI: https://doi.org/10.1016/0169-7439(87)80084-9. Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists.

[Wyner, 1979] Wyner, A. (1979). An analog scrambling scheme which does not expand bandwidth, Part I: Discrete time. *IEEE Transactions on Information Theory*, 25(3):261–274, DOI: https://doi.org/10.1109/TIT.1979.1056050.

[Xu, 2017] Xu, Z. (2017). Data transmission method based on single carrier over GSM voice channel. *Revista de la Facultad de Ingeniera*, 32(9):23–29.

[Yin et al., 2018] Yin, H., Zhang, J., Xiong, Y., Huang, X., and Deng, T. (2018). PPK-Means: Achieving Privacy-Preserving Clustering Over Encrypted Multi-Dimensional Cloud Data. *Electronics*, 7(11):310–328, ISSN: 2079-9292, DOI: http://dx.doi.org/10.3390/electronics7110310.

[Yoon et al., 2007] Yoon, J. S., Lee, G. H., and Kim, H. K. (2007). A MFCC-based CELP Speech Coder for Server-Based Speech Recognition in Network Environments. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 90(3):626–632, DOI: https://doi.org/10.1093/ietfec/e90-a.3.626.

[Zeileis et al., 2009] Zeileis, A., Hornik, K., and Murrell, P. (2009). Escaping RGBland: Selecting colors for statistical graphics. *Computational Statistics & Data Analysis*, 53(9):3259 – 3270, ISSN: 0167-9473, DOI: https://doi.org/10.1016/j.csda.2008.11.033.

[Zhao et al., 2007] Zhao, Y.-X., Su, M.-C., Chou, Z.-L., and Lee, J. (2007). A Puzzle Solver and Its Application in Speech Descrambling. In *WSEAS International Conference on Computer Engineering and Applications*, pages 171–176. World Scientific and Engineering Academy and Society (WSEAS), ISBN: 9789608457584.

[Zhou et al., 2018] Zhou, Y., Xiang, T., and Li, X. (2018). Efficient and Privacy-Preserving Query on Outsourced Spherical Data. In Vaidya, J. and Li, J., editors, *Algorithms and Architectures*

for Parallel Processing, pages 138–152, Cham, Switzerland. Springer International Publishing, DOI: https://doi.org/10.1007/978-3-030-05063-4_12.

[Zielinski et al., 2007] Zielinski, S., Hardisty, P., Hummersone, C., and Rumsey, F. (2007). Potential Biases in MUSHRA Listening Tests. In *Audio Engineering Society Convention 123*. Audio Engineering Society, http://www.aes.org/e-lib/browse.cfm?elib=14237.

[Zimmermann, 1996] Zimmermann, P. (1996). PGPfone Owner's Manual. Pretty Good Privacy. http://web.mit.edu/network/pgpfone/manual/.

[Zorila et al., 2012] Zorila, T.-C., Kandia, V., and Stylianou, Y. (2012). Speech-in-noise intelligibility improvement based on spectral shaping and dynamic range compression. In *Thirteenth Annual Conference of the International Speech Communication Association*, pages 635–638. https://www.isca-speech.org/archive/interspeech_2012/i12_0635.html.

[Zue et al., 1990] Zue, V., Seneff, S., and Glass, J. (1990). Speech database development at MIT: Timit and beyond. *Speech Communication*, 9(4):351 – 356, DOI: https://doi.org/10.1016/0167-6393(90)90010-7.

Annexes

A Authenticated Key Exchange Protocol

The code below is the Tamarin file 'AKE_over_Voice.spthy' describing the authenticated key exchange (AKE) protocol illustrated in Figure 6.7 in Chapter 6. The code is also available online. [6]

The first file section details special events. The rule Generate_pk generates a signature private/public key pair $(!\mathrm{Ltk}(\$U, \sim \mathrm{ltk}), !\mathrm{Pk}(\$U, \mathrm{pk}(\sim \mathrm{ltk})))$ for arbitrary user U. The rule Reveal_ltk compromises a user U and reveals her signing key ltk to the adversary. Finally, the rules VoiceChannelOut and VoiceChannelIn symbolize two ends of a channel used in vocal verification. In contrast to other rules, the output of VoiceChannelOut cannot be modified by the adversary without being revealed at the rule VoiceChannelIn.

The second file section describes the interaction between the Initiator (Alice) and the Responder (Bob). In the exchange, users share their identifiers, public keys, nonces, signatures, and hashes. The participants commit to all values and verify received hashes. A scenario when a user cannot verify the digital signature is solved by introducing alternative rules Alice_2_nosign and Bob_2_nosign. Mutual vocal verification is symbolized by the rule Compare_SAS. The correct protocol interaction can be checked by running lemmas executable1 and executable2.

In the third part, the code specifies security lemmas. The lemmas Non_Injective_Agreement and Injective_Agreement verify respectively non-injective and injective agreements between participants assuming signature or vocal authentication. The secrecy of the Session Key is checked by lemmas Session_Key_Secrecy and Perfect_Forward_Secrecy. The last lemmas Impersonation_with_signature, Reflection_with_signature, and Reflection_no_signature, test protocol's resilience against user impersonation and reflections.

The presented code can be directly loaded by Tamarin [7] and verified. The verification process takes about 10 minutes under Ubuntu kernel 5.8.0-25 and using Intel Core i7 2.9 GHz without multi-threading.

```
theory AKE_over_voice_channels
begin

builtins: diffie-hellman, signing, hashing

// Generation of the key pair

rule Generate_pk:
  [ Fr(~ltk) ]
  --[ Generate($U) ]->
  [ !Ltk( $U, ~ltk ), !Pk( $U, pk(~ltk) ) ]
```

6. https://github.com/PiotrKrasnowski/AKE_over_Voice
7. https://tamarin-prover.github.io/

```
// Adversary reveals the secret key

rule Reveal_ltk:
    [ !Ltk( A, ltk ) ]
  --[ LtkReveal( A ) ]->
    [ Out(ltk) ]

// Authenticated channel

rule VoiceChannelOut:
    [ Out_A( $A, $B, x ) ]  --[ ChanOut_A( $A, $B, x ) ]->
       [ !Auth( $A, x ), Out(< $A, $B, x >) ]

rule VoiceChannelIn:
    [ !Auth( $A, x ), In($B) ] --[ ChanIn_A( $A, $B, x ) ]->
      [ In_A( $A, $B, x ) ]

// Protocol rules

rule Alice_1:
    [ Fr(~ekA)
    , Fr(~NA)
    , Fr(~RA) ]
    -->
    [ Alice_1( $A, ~NA, ~ekA, ~RA )
    , Out(< $A, ~NA, 'g' ^ ~ekA, h(< $A, ~NA, 'g' ^ ~ekA, ~RA >) >) ]

rule Alice_2:
    [ Alice_1( $A, ~NA, ~ekA, ~RA)
    , !Ltk( $A, ltkA )
    , !Pk( $B, pk(ltkB) )
    , In(< $B, NB, Y, RB,
           sign{'B', $B, $A, NB, ~NA, Y, 'g' ^ ~ekA }ltkB >)
    ]
  --[ SessionKey( $A, $B, < $A, $B, ~NA, NB, Y ^ ~ekA >,
                  h(< ~RA, RB, $A, $B, ~NA, NB, 'g' ^ ~ekA, Y >) )
    , Running( 'B', $B, $A, < $A, $B, ~NA, NB, Y ^ ~ekA >)
    , Compare_RA( h(< $A, ~NA, 'g' ^ ~ekA, ~RA >),
                  h(< $A, ~NA, 'g' ^ ~ekA, ~RA >) )
    , Commit_S( 'A', $A, $B, < $A, $B, ~NA, NB, Y ^ ~ekA >,
                h(< ~RA, RB, $A, $B, ~NA, NB, 'g' ^ ~ekA, Y >) ) ]->
    [ Out(< ~RA, sign{ 'A', $A, $B, ~NA, NB, 'g' ^ ~ekA, Y }ltkA >)
    , Out_A( $A, $B, h(< ~RA, RB, $A, $B, ~NA, NB, 'g' ^ ~ekA, Y >) ) ]
```

```
rule Alice_2_nosign:
    [ Alice_1( $A, ~NA, ~ekA, ~RA)
    , !Ltk( $A, ltkA )
    , In(< $B, NB, Y, RB, S >)
    ]
  --[ SessionKey( $A, $B, < $A, $B, ~NA, NB, Y ^ ~ekA >,
     h(< ~RA, RB, $A, $B, ~NA, NB, 'g' ^ ~ekA, Y >) )
    , Running( 'B', $B, $A, < $A, $B, ~NA, NB, Y ^ ~ekA >)
    , Compare_RA( h(< $A, ~NA, 'g' ^ ~ekA, ~RA >),
                  h(< $A, ~NA, 'g' ^ ~ekA, ~RA >) )
    , Commit( 'A', $A, $B, < $A, $B, ~NA, NB, Y ^ ~ekA >,
              h(< ~RA, RB, $A, $B, ~NA, NB, 'g' ^ ~ekA, Y >) ) ]->
    [ Out(< ~RA, sign{ 'A', $A, $B, ~NA, NB, 'g' ^ ~ekA, Y }ltkA >)
    , Out_A( $A, $B, h(< ~RA, RB, $A, $B, ~NA, NB, 'g' ^ ~ekA, Y >) ) ]

rule Bob_1:
    [ Fr(~ekB)
    , Fr(~NB)
    , Fr(~RB)
    , !Ltk( $B, ltkB )
    , In(< $A, NA, X, hRA >)
    ]
  --[ Running( 'A', $A, $B, <$A, $B, NA, ~NB, X ^ ~ekB >) ]->
    [
      Bob_1( $B, $A, ~NB, NA, ~ekB, X, ~RB, hRA)
    , Out(< $B, ~NB, 'g' ^ ~ekB, ~RB,
           sign{ 'B', $B, $A, ~NB, NA, 'g' ^ ~ekB, X }ltkB >) ]

rule Bob_2:
    [
      Bob_1( $B, $A, ~NB, NA, ~ekB, X, ~RB, hRA )
    , !Pk( $A, pk(ltkA) )
    , In(< RA, sign{ 'A', $A, $B, NA, ~NB, X, 'g' ^ ~ekB }ltkA >)
    ]
  --[ Compare_RA( hRA, h(< $A, NA, X, RA >) )
    , SessionKey( $B, $A, < $A, $B, NA, ~NB, X ^ ~ekB >,
                  h(< RA, ~RB, $A, $B, NA, ~NB, X, 'g' ^ ~ekB >))
    , Commit_S( 'B', $B, $A, < $A, $B, NA, ~NB, X ^ ~ekB >,
                h(< RA, ~RB, $A, $B, NA, ~NB, X, 'g' ^ ~ekB >) ) ]->
    [
     Out_A( $B, $A, h(< RA, ~RB, $A, $B, NA, ~NB, X, 'g' ^ ~ekB >) )
    ]
```

```
rule Bob_2_nosign:
    [ Bob_1( $B, $A, ~NB, NA, ~ekB, X, ~RB, hRA )
    , In(< RA, S >)
    ]
  --[ Compare_RA( hRA, h(< $A, NA, X, RA >) )
   , SessionKey( $B, $A, < $A, $B, NA, ~NB, X ^ ~ekB >,
                 h(< RA, ~RB, $A, $B, NA, ~NB, X, 'g' ^ ~ekB >))
    , Commit( 'B', $B, $A, < $A, $B, NA, ~NB, X ^ ~ekB >,
              h(< RA, ~RB, $A, $B, NA, ~NB, X, 'g' ^ ~ekB >) ) ]->
    [
     Out_A( $B, $A, h(< RA, ~RB, $A, $B, NA, ~NB, X, 'g' ^ ~ekB >) )
    ]

rule Compare_SAS:
    [
      In_A( $A, $B, sequenceA ), In_A( $B, $A, sequenceB )
    ]
    --[ CompareSAS( 'A', $A, $B, sequenceA, sequenceB )
      , CompareSAS( 'B', $B, $A, sequenceB, sequenceA ) ]-> []

// lemmas

lemma executable1 :
exists-trace
  "Ex A B sessKey SAS hRA #i #j #k.
    Commit( 'A', A, B, sessKey, SAS ) @ i &
    Compare_RA( hRA, hRA ) @ i &

    Commit( 'B', B, A, sessKey, SAS ) @ j &
    Compare_RA( hRA, hRA ) @ j &

    CompareSAS( 'A', A, B, SAS, SAS ) @ k &
    CompareSAS( 'B', B, A, SAS, SAS ) @ k &

    not( A = B ) &
    not( Ex U #m . LtkReveal(U) @ m )"

lemma executable2 :
exists-trace
  "Ex A B sessKey SAS #i #j.
    Commit_S( 'A', A, B, sessKey, SAS ) @ i &
    Commit_S( 'B', B, A, sessKey, SAS ) @ j &
    not( A = B ) &
    not( Ex U #k . LtkReveal(U) @ k )"
```

```
// authentication lemmas

lemma Non_Injective_Agreement :
  " //signed
   (
    All role U1 U2 sessKey SAS #i.
      // user U1 playing a role 'role'
      // completed a run presumably with U2
      Commit_S( role, U1, U2, sessKey, SAS ) @ i
      // then U2 was running the protocol with U1
      // and both agreed on sessKey
      ==> ( Ex #j. Running( role, U1, U2, sessKey ) @ j )
        // or the adversary revealed a secret key of the user U2
        | ( Ex #r. LtkReveal(U2) @ r & r < i )
   )
   & // unsigned & SAS compared
   (
    All role U1 U2 sessKey SAS #i #j.
      // user U1 playing a role 'role'
      // completed a run presumably with U2
      Commit( role, U1, U2, sessKey, SAS ) @ i &
      //Compare_RA( hRA, hRA ) @ i &
      CompareSAS( role, U1, U2, SAS, SAS ) @ j & not( U1 = U2 )
      // then U2 was running the protocol with U1
      // and both agreed on sessKey
      ==> ( Ex #k. Running( role, U1, U2, sessKey) @ k)
        // or the adversary revealed a secret key of the user U2
        | ( Ex #r. LtkReveal(U2) @ r & r < i )
   )"

lemma Non_Injective_Agreement_no_sign_nor_SAS :
  exists-trace
  "not All role U1 U2 sessKey SAS #i.
      // user U1 playing a role 'role'
      // completed a run presumably with U2
      Commit( role, U1, U2, sessKey, SAS ) @ i
      // but U2 was NOT running the protocol with U1
      ==> ( Ex #k. Running( role, U1, U2, sessKey ) @ k )
        // or the adversary revealed a secret key of the user U2
        | ( Ex #r. LtkReveal(U2) @ r & r < i )"
```

```
lemma Injective_Agreement :
  " //signed
   (
    All role U1 U2 sessKey SAS #i.
    // user U1 playing a role 'role'
    // completed a run presumably with U2
    Commit_S( role, U1, U2, sessKey, SAS ) @ i
    // then U2 was running the protocol with U1
    // and both agreed on sessKey
    ==> (Ex #j. Running( role, U1, U2, sessKey ) @ j & j < i
      // and there is a unique matching instance
      & not ( Ex U3 U4 #i2. Commit_S( role, U3, U4, sessKey, SAS )
              @i2 & not ( #i2 = #i ) )
      & not ( Ex U3 U4 #i2. Commit( role, U3, U4, sessKey, SAS )
              @i2 & not ( #i2 = #i ) ) )
      // or the adversary revealed a secret key of the user U2
      | ( Ex #r. LtkReveal(U2) @ r & r < i )
   )
   & // unsigned & SAS compared
   (
    All role U1 U2 sessKey SAS #i #j.
    // user U1 playing a role 'role'
    // completed a run presumably with U2
    Commit( role, U1, U2, sessKey, SAS ) @ i & not( U1 = U2 ) &
    CompareSAS( role, U1, U2, SAS, SAS ) @ j
    // then U2 was running the protocol with U1
    // and both agreed on sessKey
    ==> ( Ex #k. Running( role, U1, U2, sessKey ) @ k & k < j
      // and there is a unique matching instance
      & not ( Ex U3 U4 #i2. Commit_S( role, U3, U4, sessKey, SAS )
              @ i2 & not ( #i2 = #i ) )
      & not ( Ex U3 U4 #i2. Commit( role, U3, U4, sessKey, SAS )
              @ i2 & not ( #i2 = #i ) ) )
      // or the adversary revealed a secret key of the user U2
      | ( Ex #r. LtkReveal(U2) @ r & r < i )
    )"
```

```
lemma Injective_Agreement_no_sign_nor_SAS :
  exists-trace
  "not All role U1 U2 sessKey SAS #i.
    // user U1 playing a role 'role'
    // completed a run presumably with U2
    Commit( role, U1, U2, sessKey, SAS ) @ i & not( U1 = U2 )
    // but U2 was NOT running the protocol with U1
    // or there is another matching instance
    ==> (Ex #k. Running( role, U1, U2, sessKey ) @ k & k < i
      & not (Ex U3 U4 #i2. Commit_S( role, U3, U4, sessKey, SAS )
             @ i2 & not (#i2 = #i ) )
      & not (Ex U3 U4 #i2. Commit( role, U3, U4, sessKey, SAS )
             @ i2 & not (#i2 = #i ) ) )
      // or the adversary revealed a secret key of the user U2
      | ( Ex #r. LtkReveal(U2) @ r & r < i )"

// secrecy lemmas

lemma Session_Key_Secrecy :
  " //signed
  (
  All role U1 U2 sessKey SAS #i #l.
     // user U1 playing a role 'role' completed
     // a run presumably with U2
     SessionKey( U1, U2, sessKey, SAS ) @ i &
     Commit_S( role, U1, U2, sessKey, SAS ) @ i &
     // but the adversary knows the key anyway
     K(sessKey) @ l
     // then the adversary revealed a secret key of the user U2
     ==>
     Ex #r. LtkReveal(U2) @ r
  )
  & // unsigned & SAS compared
  (
  All role U1 U2 sessKey SAS #i #k #l.
     // user U1 playing a role 'role'
     // completed a run presumably with U2
     SessionKey( U1, U2, sessKey, SAS ) @ i &
     CompareSAS( role, U1, U2, SAS, SAS ) @ k & not( U1 = U2 ) &
     // but the adversary knows the key anyway
     K(sessKey) @ l
     // then the adversary revealed a secret key of the user U2
     ==>
     Ex #r. LtkReveal(U2) @ r
  ) "
```

```
lemma Session_Key_Secrecy_no_sign_nor_SAS :
  exists-trace
  "not All U1 U2 sessKey SAS #i #j.
     // user U1 playing a role 'role'
     // completed a run presumably with U2
     SessionKey( U1, U2, sessKey, SAS ) @ i &
     // but the adversary knows the key anyway
     K(sessKey) @ j
     ==>
     // then the adversary revealed a secret key of the user U2
     Ex #r. LtkReveal(U2) @ r"

lemma Perfect_Forward_Secrecy :
  " //signed
  (
  All role U1 U2 sessKey SAS #i #l.
     // user U1 playing a role 'role' completed a run presumably with U2
     SessionKey( U1, U2, sessKey, SAS ) @ i &
     Commit_S( role, U1, U2, sessKey, SAS ) @ i &
     // but the adversary knows the key anyway
     K(sessKey) @ l
     ==>
     // then the adversary revealed a secret key of the user U2 earlier
     Ex #r. LtkReveal(U2) @ r & r < i
  )
  & // unsigned & SAS compared
  (
  All role U1 U2 sessKey SAS #i #k #l.
     // user U1 playing a role 'role' completed a run presumably with U2
     SessionKey( U1, U2, sessKey, SAS ) @ i &
     CompareSAS( role, U1, U2, SAS, SAS ) @ k & not( U1 = U2 ) &
     // but the adversary knows the key anyway
     K(sessKey) @ l
     ==>
     // then the adversary revealed a secret key of the user U2 earlier
     Ex #r. LtkReveal(U2) @ r & r < i
  )"
```

```
// other lemmas

lemma Impersonation_with_signature :
  "All role U1 U2 sessKey SAS #i.
    // user U1 playing a role 'role'
    // completed a run presumably with U2
    Commit_S( role, U1, U2, sessKey, SAS ) @ i & not( U1 = U2 ) &
    // but U2 was NOT running the protocol
    // with U1 (adversary impersonated U2)
    not( Ex #j. Running( role, U1, U2, sessKey ) @ j & j < i )
    // then the adversary revealed
    // a secret key of the user U2 earlier
    ==>
    Ex #r. LtkReveal(U2) @ r & r < i"

lemma Reflection_with_signature :
  "All role U1 U2 sessKey SAS #i.
    // user U1 playing a role 'role'
    // completed a run presumably with U2
    Commit_S( role, U1, U2, sessKey, SAS ) @ i &
    // and was not running the protocol as both roles at the same time
    // (this configuration is not possible in the analyzed case)
    // (moreover, U1 would be aware of the double role)
    not( Ex role2 #j. Running( role2, U1, U2, sessKey )
         @ j & not( role = role2 ) )
    ==>
    // then U1 is different than U2
    not( U1 = U2 )"

lemma Reflection_no_signature :
  exists-trace
  "not All role U1 U2 sessKey SAS #i #j.
    // user U1 playing a role 'role'
    // completed a run presumably with U2
    SessionKey( U1, U2, sessKey, SAS ) @ i &
    CompareSAS( role, U1, U2, SAS, SAS ) @ j &
    // and was not running the protocol as both roles at the same time
    // (this configuration is not possible in the analyzed case)
    // (moreover, U1 would be aware of the double role)
    not( Ex role2 #k. Running( role2, U1, U2, sessKey)
         @ k & not( role = role2 ) )
    ==>
    // but U1 is U2
    not( U1 = U2 )"
end
```

B Authenticated Key Exchange Protocol with Identity Protection

The code below is the Tamarin file 'AKE_over_Voice_with_Identity_Protection.spthy' describing the authenticated key exchange (AKE) protocol with identity protection illustrated in Figure 6.8 in Chapter 6. The code is also available online. [8]

Compared to the protocol model presented in Annex A, the Responder and the Initiator encipher their identifiers using a function $\mathrm{senc}(data, symmetric\_key)$ that denotes encryption with a symmetric key. Furthermore, the participants generate two sets of random nonces $\mathrm{NA1}, \mathrm{NA2}$ and $\mathrm{NB1}, \mathrm{NB2}$ for deriving a temporary symmetric key and the Session Key. Finally, a new restriction $\mathrm{Verify\_hRA}$ was added, which obliges the Responder to verify the received hash. The security lemmas remained unchanged.

The presented code can be directly loaded by Tamarin [9] and verified. The verification process takes about 10 minutes under Ubuntu kernel 5.8.0-25 and using Intel Core i7 2.9 GHz without multi-threading.

```
theory AKE_over_voice_with_identity_protecction
begin

builtins: diffie-hellman, signing, hashing, symmetric-encryption

// Generation of the key pair

rule Generate_pk:
  [ Fr(~ltk) ]
  --[ Generate($U) ]->
  [ !Ltk( $U, ~ltk ), !Pk( $U, pk(~ltk) ) ]

// Adversary reveals the secret key

rule Reveal_ltk:
    [ !Ltk( A, ltk ) ]
  --[ LtkReveal( A ) ]->
    [ Out(ltk) ]

// Authenticated channel

rule VoiceChannelOut_Authenticated:
    [ Out_A( $A, $B, x ) ]  --[ ChanOut_A( $A, $B, x ) ]->
          [ !Auth( $A, x ), Out(< $A, $B, x >) ]

rule VoiceChannelIn_Authenticated:
    [ !Auth( $A, x ), In($B) ] --[ ChanIn_A( $A, $B, x ) ]->
          [ In_A( $A, $B, x ) ]
```

8. https://github.com/PiotrKrasnowski/AKE_over_Voice
9. https://tamarin-prover.github.io/

```
// Protocol rules

rule Alice_1:
    [ Fr(~ekA)
    , Fr(~NA1)
    , Fr(~NA2)
    , Fr(~RA) ]
    -->
    [ Alice_1( $A, ~NA1, ~NA2, ~ekA, ~RA )
    , Out(< ~NA1, 'g' ^ ~ekA, h(< ~NA1, 'g' ^ ~ekA, ~RA >) >) ]

rule Alice_2:
    [ Alice_1( $A, ~NA1, ~NA2, ~ekA, ~RA)
    , !Ltk( $A, ltkA )
    , In(< NB1, Y, RB, senc(< $B, NB2 >,
          h(< Y ^ ~ekA, 'B1', NB1, ~NA1 >)) >) ]
  --[ Running( 'B', $B, $A, < $A, $B, ~NA2, NB2, Y ^ ~ekA >) ]->
    [ Out(< ~RA, senc(< $A, ~NA2, sign{ $B, NB2, Y, $A, ~NA2,
                        'g' ^ ~ekA }ltkA >,
           h(< Y ^ ~ekA, 'A', ~NA1, NB1 >)) >)
    , Alice_2( $A, $B, ~NA1, ~NA2, NB1, NB2, ~RA, RB, ~ekA, Y ) ]

rule Alice_3:
    [ Alice_2( $A, $B, ~NA1, ~NA2, NB1, NB2, ~RA, RB, ~ekA, Y )
    , !Pk( $B, pk(ltkB) )
    , In( senc( sign{ $A, ~NA2, 'g' ^ ~ekA, $B, NB2, Y }ltkB,
         h(< Y ^ ~ekA, 'B2', NB1, ~NA1 >))) ]
  --[ SessionKey( $A, $B, < $A, $B, ~NA2, NB2, Y ^ ~ekA >,
                 h(< ~RA, RB, NB1, Y >) )
    , Commit_S( 'A', $A, $B, < $A, $B, ~NA2, NB2, Y ^ ~ekA >,
               h(< ~RA, RB, NB1, Y >) ) ]->
    [ Out_A( $A, $B, h(< ~RA, RB, NB1, Y >) ) ]

rule Alice_3_nosign:
    [ Alice_2( $A, $B, ~NA1, ~NA2, NB1, NB2, ~RA, RB, ~ekA, Y )
    , In( senc( S, h(< Y ^ ~ekA, 'B2', NB1, ~NA1 >))) ]
  --[ SessionKey( $A, $B, < $A, $B, ~NA2, NB2, Y ^ ~ekA >,
                 h(< ~RA, RB, NB1, Y >) )
    , Commit( 'A', $A, $B, < $A, $B, ~NA2, NB2, Y ^ ~ekA >,
              h(< ~RA, RB, NB1, Y >) ) ]->
    [ Out_A( $A, $B, h(< ~RA, RB, NB1, Y >) ) ]
```

```
rule Bob_1:
    [ Fr(~ekB)
    , Fr(~NB1)
    , Fr(~NB2)
    , Fr(~RB)
    , In(< NA1, X, hRA >)
    ]
  -->
    [ Bob_1( $B, ~NB1, ~NB2, NA1, ~ekB, X, ~RB, hRA)
    , Out(< ~NB1, 'g' ^ ~ekB, ~RB, senc(< $B, ~NB2 >,
           h(< X ^ ~ekB, 'B1', ~NB1, NA1 >)) >) ]

rule Bob_2:
    [
      Bob_1( $B, ~NB1, ~NB2, NA1, ~ekB, X, ~RB, hRA )
    , !Ltk( $B, ltkB )
    , !Pk( $A, pk(ltkA) )
    , In(< RA, senc(< $A, NA2, sign{ $B, ~NB2, 'g' ^ ~ekB, $A, NA2, X }ltkA >,
                      h(< X ^ ~ekB, 'A', NA1, ~NB1 >)) >) ]
  --[ ComparehRA( $B, $A, hRA, h(< NA1, X, RA >) )
    , Running( 'A', $A, $B, < $A, $B, NA2, ~NB2, X ^ ~ekB >)
    , Commit_S( 'B', $B, $A, < $A, $B, NA2, ~NB2, X ^ ~ekB >,
                h(< RA, ~RB, ~NB1, 'g' ^ ~ekB >) )
    , SessionKey( $B, $A, < $A, $B, NA2, ~NB2, X ^ ~ekB >,
                  h(< RA, ~RB, ~NB1, 'g' ^ ~ekB >) )  ]->
    [ Out( senc(< sign{ $A, NA2, X, $B, ~NB2, 'g' ^ ~ekB }ltkB >,
                  h(< X ^ ~ekB, 'B2', ~NB1, NA1 >)))
    , Out_A( $B, $A, h(< RA, ~RB, ~NB1, 'g' ^ ~ekB >) ) ]

rule Bob_2_nosign:
    [ Bob_1( $B, ~NB1, ~NB2, NA1, ~ekB, X, ~RB, hRA )
    , !Ltk( $B, ltkB )
    , In(< RA, senc(< $A, NA2, sign{ $B, ~NB2, 'g' ^ ~ekB, $A, NA2, X }ltkA >,
          h(< X ^ ~ekB, 'A', NA1, ~NB1 >)) >) ]
  --[ ComparehRA( $B, $A, hRA, h(< NA1, X, RA >) )
    , Running( 'A', $A, $B, < $A, $B, NA2, ~NB2, X ^ ~ekB >)
    , Commit( 'B', $B, $A, < $A, $B, NA2, ~NB2, X ^ ~ekB >,
              h(< RA, ~RB, ~NB1, 'g' ^ ~ekB >) )
    , SessionKey( $B, $A, < $A, $B, NA2, ~NB2, X ^ ~ekB >,
                  h(< RA, ~RB, ~NB1, 'g' ^ ~ekB >) ) ]->
    [ Out( senc(< sign{ $A, NA2, X, $B, ~NB2, 'g' ^ ~ekB }ltkB >,
                  h(< X ^ ~ekB, 'B2', ~NB1, NA1 >)))
    , Out_A( $B, $A, h(< RA, ~RB, ~NB1, 'g' ^ ~ekB >) ) ]
```

```
// this restriction obliges Bob to verify the received hash
restriction Verify_hRA:
    "All u1 u2 x y #i. ComparehRA( u1, u2, x, y ) @i ==> x = y"

rule Compare_SAS:
    [ In_A( $A, $B, sequenceA ), In_A( $B, $A, sequenceB ) ]
    --[ CompareSAS( 'A', $A, $B, sequenceA, sequenceB )
      , CompareSAS( 'B', $B, $A, sequenceB, sequenceA ) ]-> []

// lemmas

lemma executable_SAS :
exists-trace
  "Ex A B sessKey SAS #i #j #k.
    Commit( 'A', A, B, sessKey, SAS ) @ i &
    Commit( 'B', B, A, sessKey, SAS ) @ j &
    CompareSAS( 'A', A, B, SAS, SAS ) @ k &
    CompareSAS( 'B', B, A, SAS, SAS ) @ k &
    not( A = B ) &
    not( Ex U #m . LtkReveal(U) @ m )"

lemma executable_signature :
exists-trace
  "Ex A B sessKey SAS #i #j.
    Commit_S( 'A', A, B, sessKey, SAS ) @ i &
    Commit_S( 'B', B, A, sessKey, SAS ) @ j &
    not( A = B ) &
    not( Ex U #k . LtkReveal(U) @ k )"

// authentication lemmas

lemma Non_Injective_Agreement :
  " //signed
   (
    All role U1 U2 sessKey SAS #i.
      // user U1 playing a role 'role'
      // completed a run presumably with U2
      Commit_S( role, U1, U2, sessKey, SAS ) @ i
      // then U2 was running the protocol with U1
      // and both agreed on sessKey
      ==> ( Ex #j. Running( role, U1, U2, sessKey ) @ j )
        // or the adversary revealed a secret key of the user U2
        | ( Ex #r. LtkReveal(U2) @ r & r < i )
   )
```

```
   & // unsigned & SAS compared
   (
    All role U1 U2 sessKey SAS #i #j.
      // user U1 playing a role 'role'
      // completed a run presumably with U2
      Commit( role, U1, U2, sessKey, SAS ) @ i &
      CompareSAS( role, U1, U2, SAS, SAS ) @ j & not( U1 = U2 )
      // then U2 was running the protocol with U1
      // and both agreed on sessKey
      ==> ( Ex #k. Running( role, U1, U2, sessKey) @ k)
        // or the adversary revealed a secret key of the user U2
        | ( Ex #r. LtkReveal(U2) @ r & r < i )
   )"

lemma Non_Injective_Agreement_no_sign_nor_SAS :
  exists-trace
  "not All role U1 U2 sessKey SAS #i.
      // user U1 playing a role 'role'
      // completed a run presumably with U2
      Commit( role, U1, U2, sessKey, SAS ) @ i
      // but U2 was NOT running the protocol with U1
      ==> ( Ex #k. Running( role, U1, U2, sessKey ) @ k )
        // or the adversary revealed a secret key of the user U2
        | ( Ex #r. LtkReveal(U2) @ r & r < i )"

lemma Injective_Agreement :
  " //signed
   (
    All role U1 U2 sessKey SAS #i.
    // user U1 playing a role 'role'
    // completed a run presumably with U2
    Commit_S( role, U1, U2, sessKey, SAS ) @ i
    // then U2 was running the protocol with U1
    // and both agreed on sessKey
    ==> (Ex #j. Running( role, U1, U2, sessKey ) @ j & j < i
      // and there is a unique matching instance
      & not ( Ex U3 U4 #i2. Commit_S( role, U3, U4, sessKey, SAS )
              @i2 & not ( #i2 = #i ) )
      & not ( Ex U3 U4 #i2. Commit( role, U3, U4, sessKey, SAS )
              @i2 & not ( #i2 = #i ) ) )
      // or the adversary revealed a secret key of the user U2
      | ( Ex #r. LtkReveal(U2) @ r & r < i )
   )
```

```
   & // unsigned & SAS compared
   (
    All role U1 U2 sessKey SAS #i #j.
    // user U1 playing a role 'role'
    // completed a run presumably with U2
    Commit( role, U1, U2, sessKey, SAS ) @ i & not( U1 = U2 ) &
    CompareSAS( role, U1, U2, SAS, SAS ) @ j
    // then U2 was running the protocol with U1
    // and both agreed on sessKey
    ==> ( Ex #k. Running( role, U1, U2, sessKey ) @ k & k < j
      // and there is a unique matching instance
      & not ( Ex U3 U4 #i2. Commit_S( role, U3, U4, sessKey, SAS )
              @ i2 & not ( #i2 = #i ) )
      & not ( Ex U3 U4 #i2. Commit( role, U3, U4, sessKey, SAS )
              @ i2 & not ( #i2 = #i ) ) )
      // or the adversary revealed a secret key of the user U2
      | ( Ex #r. LtkReveal(U2) @ r & r < i )
    )"

lemma Injective_Agreement_no_sign_nor_SAS :
  exists-trace
   "not All role U1 U2 sessKey SAS #i.
    // user U1 playing a role 'role'
    // completed a run presumably with U2
    Commit( role, U1, U2, sessKey, SAS ) @ i & not( U1 = U2 )
    // but U2 was NOT running the protocol with U1
    // or there is another matching instance
    ==> (Ex #k. Running( role, U1, U2, sessKey ) @ k & k < i
      & not (Ex U3 U4 #i2. Commit_S( role, U3, U4, sessKey, SAS )
             @ i2 & not (#i2 = #i ) )
      & not (Ex U3 U4 #i2. Commit( role, U3, U4, sessKey, SAS )
             @ i2 & not (#i2 = #i ) ) )
      // or the adversary revealed a secret key of the user U2
      | ( Ex #r. LtkReveal(U2) @ r & r < i )"
```

```
// secrecy lemmas

lemma Session_Key_Secrecy :
  " //signed
  (
  All role U1 U2 sessKey SAS #i #l.
     // user U1 playing a role 'role'
     // completed a run presumably with U2
     SessionKey( U1, U2, sessKey, SAS ) @ i &
     Commit_S( role, U1, U2, sessKey, SAS ) @ i &
     // but the adversary knows the key anyway
     K(sessKey) @ l
     // then the adversary revealed a secret key of the user U2
     ==>
     Ex #r. LtkReveal(U2) @ r
  )
  & // unsigned & SAS compared
  (
  All role U1 U2 sessKey SAS #i #k #l.
     // user U1 playing a role 'role'
     // completed a run presumably with U2
     SessionKey( U1, U2, sessKey, SAS ) @ i &
     CompareSAS( role, U1, U2, SAS, SAS ) @ k & not( U1 = U2 ) &
     // but the adversary knows the key anyway
     K(sessKey) @ l
     // then the adversary revealed a secret key of the user U2
     ==>
     (Ex #r. LtkReveal(U2) @ r)
  )"

lemma Session_Key_Secrecy_no_sign_nor_SAS :
  exists-trace
  "not All U1 U2 sessKey SAS #i #j.
     // user U1 playing a role 'role'
     // completed a run presumably with U2
     SessionKey( U1, U2, sessKey, SAS ) @ i &
     // but the adversary knows the key anyway
     K(sessKey) @ j
     ==>
     // then the adversary revealed a secret key of the user U2
     Ex #r. LtkReveal(U2) @ r"
```

```
lemma Perfect_Forward_Secrecy :
  " //signed
  (
  All role U1 U2 sessKey SAS #i #l.
     // user U1 playing a role 'role'
     // completed a run presumably with U2
     SessionKey( U1, U2, sessKey, SAS ) @ i &
     Commit_S( role, U1, U2, sessKey, SAS ) @ i &
     // but the adversary knows the key anyway
     K(sessKey) @ l
     ==>
     // then the adversary revealed
     // a secret key of the user U2 earlier
     Ex #r. LtkReveal(U2) @ r & r < i
  )
  & // unsigned & SAS compared
  (
  All role U1 U2 sessKey SAS #i #k #l.
     // user U1 playing a role 'role'
     // completed a run presumably with U2
     SessionKey( U1, U2, sessKey, SAS ) @ i &
     CompareSAS( role, U1, U2, SAS, SAS ) @ k & not( U1 = U2 ) &
     // but the adversary knows the key anyway
     K(sessKey) @ l
     ==>
     // then the adversary revealed
     // a secret key of the user U2 earlier
     (Ex #r. LtkReveal(U2) @ r & r < i)
  )"

// other lemmas

lemma Impersonation_with_signature :
   "All role U1 U2 sessKey SAS #i.
    // user U1 playing a role 'role'
    // completed a run presumably with U2
    Commit_S( role, U1, U2, sessKey, SAS )
              @ i & not( U1 = U2 ) &
    // but U2 was NOT running the protocol
    // with U1 (adversary impersonated U2)
    not( Ex #j. Running( role, U1, U2, sessKey )
         @ j & j < i )
    // then the adversary revealed
    // a secret key of the user U2 earlier
    ==>
    Ex #r. LtkReveal(U2) @ r & r < i "
```

```
lemma Reflection_with_signature :
   "All role U1 U2 sessKey SAS #i.
    // user U1 playing a role 'role'
    // completed a run presumably with U2
    Commit_S( role, U1, U2, sessKey, SAS ) @ i &
    // and was not running the protocol as both roles at the same time
    // (this configuration is not possible in the analyzed case)
    // (moreover, U1 would be aware of the double role)
    not( Ex role2 #j. Running( role2, U1, U2, sessKey )
         @ j & not( role = role2 ) )
    ==>
    // then U1 is different than U2
    not( U1 = U2 )"

lemma Reflection_no_signature :
  exists-trace
   "not All role U1 U2 sessKey SAS #i #j.
    // user U1 playing a role 'role'
       completed a run presumably with U2
    SessionKey( U1, U2, sessKey, SAS ) @ i &
    CompareSAS( role, U1, U2, SAS, SAS ) @ j &
    // and was not running the protocol as both roles at the same time
    // (this configuration is not possible in the analyzed case)
    // (moreover, U1 would be aware of the double role)
    not( Ex role2 #k. Running( role2, U1, U2, sessKey)
         @ k & not( role = role2 ) )
    ==>
    // but U1 is U2
    not( U1 = U2 )"
end
```