A Survey of Voice and Communication Protection Solutions Against Wiretapping

Christoforos Ntantogian[*], Eleni Veroni[*], Georgios Karopoulos[°], Christos Xenakis[*]
[*]Department of Digital Systems, University of Piraeus,
email: {dadoyan, veroni, xenakis}@unipi.gr
[°] Joint Research Center (JRC), European Commission,
email: georgios.karopoulos@ec.europa.eu
[°] This work was performed while this author was with the department of Informatics and Telecommunications of the University of Athens, Greece.

Abstract

This paper categorizes, presents and evaluates a set of schemes and solutions that provide end-to-end encryption for voice communications. First, we analyze the research works that propose new schemes that enable the transfer of encrypted speech over the voice channel of the 2nd generation mobile network. Next, we analyze a set of popular widespread software applications that use Voice over IP technology to provide secure communications, and finally, we investigate commercial solutions, which are hardware-based and offer voice encryption for both 2nd generation and Voice over IP communications. After the presentation of the existing solutions, we evaluate them based on the following criteria: i) security level provided, ii) possible performance issues and iii) usability. We conclude this work by providing future research directions. To the best of our knowledge, this is the first paper that categorizes and provides a comprehensive evaluation of end-to-end voice encryption schemes for mobile networks.

Keywords: Voice encryption; wiretapping; mobile networks; secure communications; secure voice; end-to-end encryption.

1 Introduction

Nowadays, mobile networks are interconnected systems of various technologies and networks. Older technologies like Global System for Mobile Communications (GSM), interoperate with new generation networks such as Universal Mobile Telecommunications System (UMTS) and Long-Term Evolution (LTE) to combine coverage and high data rates. In parallel to the evolution of radio access technologies, attacks in mobile networks have become also more sophisticated; from impersonation attacks, due to the lack of mutual authentication in GSM, to advanced persistent threats in UMTS and new attack vectors in LTE networks. Thus, security is still a major concern of mobile users, since several security loopholes have been exploited in the past by adversaries. As a matter of fact, confidential voice communications, which is one of the most important privacy requirements of mobile users, has become a prime target of perpetrators in the past [1].

In GSM, call interception can be achieved by breaking the voice encryption. This can be accomplished relatively easily, since the A5/1, a 64-bit encryption stream cipher responsible for confidentiality preservation on GSM air interface, is vulnerable to brute force attacks. As a solution, the GSM specifications have introduced a new stronger algorithm, named A5/3 to replace A5/1. A5/3 is based on the same algorithm that UMTS voice encryption has also adopted called KASUMI. 3GPP advocates that KASUMI provides strong security guarantees for the confidentiality of the transmitted voice, while there are no attacks discovered so far, except for theoretical ones [2]. However, the adoption of the A5/3 by mobile operators seems to be slow in many countries as shown in the GSMmap website[1], leaving their subscribers vulnerable to attacks. Moreover, even if A5/1 is replaced by A5/3, downgrade attacks are possible, where the attacker can enforce the mobile device to use weak encryption algorithms (i.e., A5/1 or A5/2) or even totally disable the encryption. All the above attacks can be easily achieved using IMSI catchers, which are fake base stations under the possession of the attacker that can lure the mobile equipment to connect to them. In this way, the attacker achieves a Man-in-The-Middle (MiTM) position between the user and the legitimate base stations. From this point, the attacker can break the

[1] www.gsmmap.org

A5/1 key or try to downgrade the security capabilities. An alarming fact is that nowadays IMSI catchers can be built with easily accessible hardware such as Universal Software Radio Peripheral (USRP) that costs less than 700€, along with a free open source software named OpenBTS, which implements the three lower layers of the GSM protocol stack. On the other hand, the security architecture of the UMTS and LTE networks has been redesigned and fortified to defeat many of the attacks that can be performed in GSM networks. However, these new generation mobile networks are not impenetrable and share their own set of security flaws [3]. Moreover, downgrade attacks are also possible, where an attacker can enforce the user to use the insecure GSM network, instead of UMTS or LTE networks. Therefore, the privacy of the users in next generation networks is still not guaranteed.

Regardless of the security flaws of GSM, UMTS or LTE networks, the security architecture of mobile networks has an inherent characteristic that undermines the privacy of users: *That is, voice encryption is not provided in an end-to-end manner*. Thus, the user is enforced to place trust to the mobile operators, which he/she is subscribed to. The above important remarks are also pinpointed in a recent report in 2017 [4] by the Department of Homeland Security (DHS) in consultation with the National Institute of Standards and Technology (NIST), clearly mentioning that (pp 55 – Table 4): *"Due to the nature of carrier networks no voice or data should depend solely on the network for confidentiality or integrity protection"*. Moreover, the report points out that a proper defense to mitigate all possible attacks on mobile network is the following: *"Ensure devices use end-to-end encryption for all communications paths"*. Therefore, protecting from call interceptions is a timely topic both for the research community and the industry and new solutions are required to improve the privacy of mobile users.

This survey categorizes, presents and evaluates a set of schemes and solutions that provide end-to-end encryption for voice communications. The categorization we followed relies on the implementation level: i) research works that have been validated through the means of simulation/emulation, ii) commercial software-based solutions, and iii) commercial hardware-based solutions. More specifically, in the first category we analyze research results, as these have been documented in published papers in the field. All the considered papers propose new schemes that enable the transfer of encrypted speech over the voice channel of the GSM mobile network. As GSM is highly insecure and call interceptions are feasible, the solutions of this category try to improve the privacy of GSM mobile subscribers. Every proposed scheme in this category tries to overcome the restrictions imposed by the GSM voice channel pertaining to limited bandwidth. In the second category, we examine commercial solutions that are software-based and do not require additional equipment, or alterations on the already existing hardware in order to operate. To this end, we analyze a set of popular widespread software applications that use VoIP technology to provide secure communications. Applications of this category take advantage of the available bandwidth in next generation mobile network (i.e., UMTS, LTE), and are built with privacy and security by design features. The last category includes commercial solutions, which rely on the use of specialized hardware, such as an external headset, to offer voice encryption for both GSM and VoIP communications.

The rest of the paper is structured as follows. Section 2 provides the background by analyzing the GSM voice processing steps, as well as protocols and technologies used for voice transmission over IP. Next, Section 3 includes the threat model, as well as the security and functionality requirements. Section 4 analyzes schemes for transmission of voice over the GSM voice channel. Section 5 describes applications for VoIP security, while Section 6 presents commercial hardware products for secure communications. Finally, Section 7 evaluates the presented solutions, mainly, in terms of security, performance and usability, while Section 8 discuss possible research directions. Finally, Section 9 concludes the paper.

2 Background

In mobile networks, voice calls have been supported using circuit-switched (CS) technology, where a dedicated *voice channel* is established for the transmission of voice traffic. This is the case for GSM and UMTS networks, where voice is transmitted through CS network (note that GSM has also a dedicated data channel named Circuit Switched Data (CSD), which is not suitable for voice transmission – see Section 4.1). However, UMTS networks also employ packet-switched (PS) technology for *data channels*, which can be used for voice transmission over IP (i.e., VoIP) achieving 2 Mbps data rate. Finally, LTE networks have ditched CS networks and carry voice traffic, exclusively, over *data channels* based on IP networks (i.e., VoIP). Nowadays, a mobile network is a composition of old and new network access

technologies. Therefore, a mobile user has effectively three different ways to establish calls: i) the GSM voice channel, ii) the UMTS voice channel and iii) the UMTS/LTE data channel. In this paper, we will focus on the first and third methods (i.e., GSM voice channel and UMTS/LTE data channel), since there are no solutions for end-to-end encryption for the second method (i.e., UMTS voice channel), except for [5] that proposes a secure scheme for video conference over UMTS voice channel.

All proposed solutions for end-to-end GSM voice encryption focus on techniques to transfer encrypted data (i.e., voice) over the limited-bandwidth voice channel. In the following, we analyze how human voice is processed before transmission over the voice channel of the GSM mobile networks. This information will be useful to comprehend the technical challenges of end-to-end voice encryption over the GSM voice channel.

2.1 Voice calls over GSM

In GSM, the steps to perform voice processing before the transmission of the speech are the following:

1) Analog to Digital (A/D) Conversion: First, the analogue voice waveform is sampled, typically with the frequency of 8 KHz. Due to the fact that each sample is 13 bits, the final bit rate is 104 kbps at this step (13x8000).

2) Speech Coding: In this step, the digitized speech first is divided into 20ms frames, each containing 160 samples. These frames are passed through the GSM voice codecs (also known as vocoders). The latter aim at reducing the bit rate of speech that has been converted from its analogue form into a digital format, to enable it to be carried within the available bandwidth for the channel (i.e., compression). Audio codecs use a technique based on Linear Prediction Coding (LPC) or some variation of it, in order to model speech signals and achieve large levels of compression. The LPC-based compression methods are lossy, and thus, the compressed speech signal is not the same with the original signal (sample-by-sample), but perceptually resembled.

GSM supports several codecs that have different compression rates and all of them use a variation of the LPC method. The first one, GSM-Full Rate (FR) and is based on RPE-LTP (Regular Pulse Excitation - Long Term Prediction) compression algorithm, which uses Code Excited Linear Prediction (CELP) method. The Half Rate (HF) requires half the bandwidth of the FR codec; thus, network capacity for voice traffic is doubled, at the expense of audio quality. The Enhanced Full Rate (EFR) (or GSM 06.60) is a speech coding standard that was developed in order to improve the quite poor quality of GSM-FR codec. A newer coder named Adaptive Multi Rate (AMR) uses (among others) the Algebraic CELP (ACELP) method for efficient compression. Moreover, other voice related technologies such as VoIP, utilize alternative voice codecs (i.e., G729). The GSM voice coders can be selected depending the speech quality and traffic conditions.

3) Channel coding, interleaving & burst formatting: In channel coding, one or more control and user data signals are combined with error protection or error correction information. This typically comprises the addition of error detection and error protection bits, along with the rearrangement of bit order for transmission. Moreover, interleaving reorders data that is to be transmitted so consecutive bytes of data are distributed over a larger sequence of them in order to reduce the effect of burst errors. Finally, the assembling of bursts (i.e., burst formatting) takes place, in order to send a series of bits in succession.

4) Ciphering and Modulation: The final step is the encryption of the digitized voice data (using A5/1, A5/2 or the new A5/3 ciphering algorithms). Regarding modulation, GSM uses Gaussian Minimum-Shift Keying (GMSK) and Time Division Multiple Access (TDMA) as its access scheme.

Apart from voice processing, GSM employs several techniques to improve the overall voice communication. In particular, to provide feedback to the user that a connection is still present, a Comfort Noise Generator (CNG) is used to generate some background noise, even when no speech data is being transmitted. This is executed, locally, at the receiver. In GSM, bandwidth is additionally increased using a technique called Discontinuous Transmission (DTX), which activates the transmitter only during speech activity periods. DTX is related with potential degradation of the modulated signal quality, due to signal cutting (modulation signal detected as noise) and noise contrast effects. Moreover, the Voice Activity Detection (VAD) identifies the presence or absence of speech in the input signal, in order to provide transmission only to speech, discarding silence and noise signals. One of the main features of VAD used for the signal classification (i.e. voice or noise) is the short-term energy of the signal. Thus, VAD plays a crucial role for the transmission of coded signals over the GSM voice channel, since once a signal is classified as nonspeech, it will be rejected and not transmitted.

2.2 Voice calls over data channels (VoIP)

Voice calls over data channels (i.e., VoIP) traverse IP networks such as the PS networks of mobile operators (UMTS or LTE) or residential networks for Internet access. To setup a call, first a signaling protocol is executed to discover and establish a connection between the two peers. Next, a data transfer protocol is performed, which carries the actual voice waveform inside IP data packets.

2.2.1 Signaling Protocol

Voice transferred over IP networks mainly uses the Session Initiation Protocol (SIP). The latter is a centralized protocol in the sense that its network infrastructure is based on client-server model. During call setup, the client communicates with the SIP server to determine where the call should be routed. For client authentication, SIP uses a mechanism based on the HTTP digest authentication. On the other hand, SIP specification does not include any specific security mechanism for confidentiality and integrity. As SIP is an application layer protocol, messages can be protected in the IP layer using IPsec or in the presentation layer using TLS. In the latter case, SIP servers utilize certificates for authentication to the SIP clients.

2.2.2 Data Protocol

Although signaling traffic may pass through a SIP proxy, the media traffic is exchanged using a data protocol in a peer-to-peer fashion, between the parties that have established a communication channel. Real Time Transport Protocol (RTP) is an application layer protocol over UDP to carry media traffic, such as audio and video streams. Since RTP does not offer security mechanisms, a secure version named SRTP was introduced to provide confidentiality, integrity and protection from replay attacks for RTP media stream. SRTP uses AES with 128-bit key size in counter mode (although the key size can be extended to 192 and 256 bits) or F8 mode (i.e., a variation of the output feedback (OFB) mode) as well as HMAC-SHA1 for integrity protection.

SRTP protects media traffic but it does not describe how encryption keys should be distributed between the communication parties. This can be achieved using the ZRTP protocol for key agreement. The latter is based on ephemeral Diffie-Hellman protocol for key negotiation and exchange over an insecure channel between two endpoints. The ephemeral Diffie-Hellman provides also Perfect Forward Secrecy (PFS), a security property which precludes retroactively compromising a call, in case of key material disclosure in the future. One major drawback of Diffie-Hellman is the absence of protection against MiTM attacks. Instead, ZRTP detects MiTM attacks using a Short Authentication String (SAS), which is a cryptographic hash of Diffie-Hellman parameters. The SAS values should be the same in both endpoints, otherwise an active MiTM attack has taken place during the Diffie-Hellman key exchange and modified the Diffie-Hellman parameters. One possible way to compare the SAS values is for the communicating parties to read aloud the displayed SAS at any time during the session. If the values are not the same, it is an indication of a MiTM attack presence.

3 Threat model and requirements

3.1 Threat Model

Threat modeling allows us to identify every possible threat category against the system. One effective method to accomplish this step is by recognizing the capabilities of the adversary who is susceptible/prone to perform the attacks. We assume that the adversary has the following security capabilities:

1) **Man-in-The-Middle (MiTM):** The adversary has the ability to have a MiTM position in the communication path between two mobile users typically by setting a false base station. In this way, the attacker is able to perform passive or active interception.
2) **Access to the core network:** The adversary has physical or remote access to the core network and the skillset to compromise the hardware or software of an entity within the core network (i.e. install a backdoor in a server).
3) **Computational capabilities:** We assume that the computational capabilities of the adversary are unlimited. The adversary has all the required toolset (software and hardware) to exploit well known

vulnerabilities in mobile networks (e.g., break the weak A5/1 encryption algorithm using rainbow tables).

3.2 Security & Functionality Requirements

Here we pinpoint a set of high-level security (*S*) and functionality (*F*) requirements that solutions for voice encryption should fulfill:
S1: The solution should provide strong end-to-end encryption. That is, only the communication parties (sender and receiver) should be able to successfully encrypt and decrypt the voice communications.
S2: Key management procedures including key storage, deletion and revoking should be carefully designed to avoid undermining the security of the system.
S3: The basis of voice encryption is secure key agreement. Otherwise, the solution will be insecure regardless of how strong the encryption algorithm is.
S4: The solution should provide Perfect Forward Secrecy (PFS), a property which guarantees that an attacker cannot decrypt past communications, even if long term encryption keys are compromised.
S5: The selection of encryption algorithms and key sizes should be based on established security guidelines and standards (e.g., NIST).
S6: Implementation-wise, end-users should not be able to modify the security configurations of the solution. Otherwise, there is an imminent danger of downgrading the provided security level.
S7: Trust should not be assumed, if third parties are included in the voice encryption solution.
F1: It should cope with deforming GSM codecs that are designed for voice transmission and not data (see Section 4.1).
F2: Keeping error rate as low as possible is of paramount importance. Otherwise, the encrypted data will be scrambled and the decryption of the data will fail.
F3: It should support the highest bit rate possible so that the quality of the voice communications is not affected.
F4: Time delays at the call setup procedure should be kept minimal.
F5: From the end-user's point of view, the complexity of the proposed solution should be kept low as much as possible. Otherwise, end-users will not adopt it.

4 Secure voice communication through data transmission over GSM voice channel

4.1 Voice encryption limitations in GSM voice channels

As discussed above, the only solution to preserve the confidentiality of user data across the mobile telecommunication networks is the deployment of end-to-end encryption. In order to achieve that, a new encryption/decryption module has to be added in the existing transmission process that encrypts speech signal at user end, before it enters the GSM air interface.

In GSM, in order for a mobile user to make a voice call, the following procedure takes place. First, the originating mobile user will request a channel from its radio access network. After the channel is granted, an authentication and key agreement will be executed between the originating mobile user and its serving network. Next, the originating mobile user will send a setup message, containing the number of the terminating mobile user so that the former (i.e., originating mobile user) discovers the serving network of the latter (i.e., terminating mobile user). On the other hand, the serving network of the terminating mobile user broadcasts a paging request message that contains the IMSI or TMSI identity of the terminating mobile user. The latter compares and verifies its own identity with the received IMSI or TMSI. Subsequently, the terminating mobile user will send a channel request message to the radio access network and finally an authentication and key agreement will be executed. If the subscriber is authenticated successfully, the call will be established through an encrypted channel.

The encrypted speech signal can be treated either as data and pass through the GSM Circuit Switch Data (CSD) channel, or as voice and be transmitted by the dedicated GSM voice channel. In the first scenario, where the encrypted speech is transmitted over the GSM CSD channel, real-time two-way conversations appear to be practically impossible. The GSM CSD channel was initially designed as a secondary service for reliable data transmission and requires approximately 18 seconds required for the

GSM modem handshake. The very high latency and the high number of unnecessary retransmissions render the GSM CSD channel insufficient for voice transmission.

In the second scenario, where the GSM voice channel is used, the transmission of encrypted speech has to face various issues, which are caused from the fact that data sent over the voice channel is processed as digitized speech. In particular, as we mentioned in section 2.1, mobile terminals employ advanced speech compression/decompression techniques to efficiently use the bandwidth of the communication channel. These techniques work on the assumption that the input waveform will be human speech and they use speech production model parameters, such as pitch and vocal tract models, to efficiently compress the input signal. However, adding a new layer of encryption would make the original waveform to be randomized and consequently fail to meet the expected speech characteristics, be treated as noise instead and never make it to transmission. A more technical analysis of the random distortions produced by voice codecs is provided in [6], which mentions that these distortions are related to several factors including the following ones: i) the voice coders include the lossy LPC-based compression technique, which is used for perceptual resemblance between the synthesized and the original speech, but from a signal analysis perspective, the two waveforms (i.e., synthesized and original) differ considerably, ii) compression and decompression of the signals are nonlinear iii) voice coders operate on the assumption that adjacent input samples are correlated. For voice signals, indeed nearby samples are highly correlated, but for data signals the correlation is insignificant. This may cause increased distortions which in turn lead to higher detection error rates reducing the overall performance.

Moreover, other techniques such as Comfort Noise Generation (CNG), which are used to reduce bandwidth usage during voice silence periods, will inadvertently cause distortions in the modulated voice waveform. Apart from distortions, the VAD module, which detects whether the waveform entering the network is speech or not, will suppress the signal and the encrypted data will be lost if it concludes that there is no speech. Therefore, the modulated encrypted voice signal should pass through the voice channel without triggering VAD. Finally, since speech signal has low bandwidth, voice channels are narrowband with 4 KHz maximum bandwidth, which limits the data rates inherently.

4.2 Voice encryption using GSM voice channel

The existing works that utilize the GSM voice channel to encrypt and transfer voice propose a new modem specially designed to overcome channel nonidealities and be compatible with the technical characteristics of GSM. In general, the proposed solutions follow three different approaches: i) parameter mapping, ii) codebook optimization, and iii) modulation optimization. The parameter mapping approach involves mapping of the input bit stream into speech parameters of some speech production model. In this method, the modulator utilizes the input data to synthesize a speech-like signal, based on a speech production model. The codebook optimization approach encodes the data into a finite alphabet of predefined symbols in an optimized way to ensure reliable and efficient transmission over the GSM voice codec. Finally, the modulation optimization method is based on well-known digital modulation methods, and the aim is to optimize the involved modulation parameters for reliable communication over the GSM voice codecs. In general, it is considered that the methods based on the parameter mapping approach have the extra benefit of producing signals that resemble voice. Thus, the VAD system will not interfere and will not deactivate the transmission, as in the case of noise signals.

4.2.1 Parameter Mapping

First, Katugampala et al. [7] in their work mapped the input data on line spectral frequencies, pitch frequency and speech frames energy. These parameters were selected to synthesize a speech-like waveform, because they represent the most important characteristics of a voice signal. The authors implemented a real-time prototype, which produces communication quality speech using the voice channel on GSM-to-GSM connections. A throughput of 3 kbps was achieved with 2.9% Bit Error Rate (BER) on GSM-to-GSM connections. By adding error correcting codes, 1.2 kbps throughput with 0.03% BER was achieved.

A Similar solution that uses also parameter mapping is presented in [8]. Contrary to [7] which utilize EFR, this work [8] applies the FR vocoder, which has the advantages of short time delay, good interoperability and high concealment for encrypted speech. The above work maps data into the speech parameters such as pitch frequencies related to formants in a waveform generated by an autoregressive speech model. A method that also uses the autoregressive speech model is presented in [9], which

achieves data transmission over the EFR voice codec with a throughput of 533 b/s, and a BER equal to 26.47%. Moreover, Boloursaz et al. [10] proposed a scheme for data communication through codecs that utilize ACELP speech coding method. This work performs a modulation on the input data stream using algebraic codebook pulse positions, creating a Pulse Position Modulation (PPM) signal for efficient data transmission through the AMR voice codec.

The above proposed solutions do not actually deploy an encryption algorithm in their scheme for evaluation. Instead, they focus on describing how data can be transmitted over a non-reliable voice channel. One of the few works of this category that actually employ encryption in the proposed scheme can be found in [11]. In this scheme, the speech is converted to digital bit stream, which in turn it is encrypted, and then the encrypted data is converted to speech-like waveform before transmitted using the 13 kbps GSM FR codec. The authors use the pitch, energy and line spectral frequencies parameters, while the LPC model is used to synthesize speech like signals. The authors have performed simulations using MATLAB, while an AES-128 in Electronic Code Book (ECB) mode encryption algorithm was implemented in an FPGA board for voice encryption. Nevertheless, the authors do not provide numerical results of the simulation. Moreover, they mention that due to the fact that digital voice data is sent encrypted, all bits should be conveyed correctly; otherwise, the original data cannot be recovered back successfully. However, the assumption that the transmission of encrypted voice data is free of errors may not hold true for all cases. Although, the employed encryption scheme uses AES in ECB mode, which means that the errors are not propagated in successive encrypted blocks, all the blocks that contain at least one bit error cannot be decrypted correctly.

Finally, Yang et al. [12] proposed an encrypted speech transmission solution for the FR speech encoding mainly based on [7]. The proposed approach was simulated using Visual C++. The authors neither discuss details of the encryption algorithm nor provide the numerical results for the achieved throughput.

4.2.2 Codebook Optimization

A representative work in this category is by LaDue et al. [13], which proposed a modem based on a set of predefined speech like symbols that were produced offline using a cooperative-competitive Genetic Algorithm (GA) search procedure. This modem attained 2 kbps throughput with 10^{-6} Symbol Error Rate (SER) and 4 kbps with 3% SER on the GSM EFR codec. On the other hand, Shahbazi et al. [14], [15] followed a different approach. They searched in a database of human speech waveforms to find the best performing symbols (i.e., this database of human speech is named TIMIT). The proposed modem was evaluated and achieved a 2 kbps data rate with a SER equal to $3.7\text{x}10^{-6}$ kbps using AMR [15] and 2 kbps data rate with SER equal to $1.5\text{x}10^{-5}$ using EFR [14].

Another work presented in [16] was motivated by the fact that all previous solutions focus on one specific voice coding scheme (e.g., GSM FR), while as we mentioned before, the GSM network includes several voice codecs with different data rates (e.g., HR, FR, AMR, EFR) To this end, the authors in [16] proposed a generic data transmission technique capable of transferring voice using different voice codecs. They applied a pattern search method to generate an optimized finite alphabet (i.e., symbols). The proposed scheme was evaluated using FR, AMR, EMR and HR voice codecs. The best performance was achieved by a throughput of 2.1 kbps using EFR (10^{-3} BER).

Both LaDue et al. [13] and Sapozhnykov et al. [16] do not reduce codebook search space; thus, their optimization is complicated and more time is required to reach a solution. On the contrary, Boloursaz et al. [6], [17] showed that by considering search space reduction, better performance (i.e., high data rate) with low search complexity is achieved. More specifically, the works in [6] and [17] propose a technique that use codebooks of speech like symbols for data transmission using AMR. The channel is modeled based on a Discrete Memoryless Channel (DMC) optimizing its capacity, instead of the SER that most previous works attempt to optimize.

4.2.3 Optimized Modulation

In this category, Chmayassani et al. [18] used the well-known Frequency Shift Keying (FSK) and Quadrature Amplitude Modulation (QAM) for modulation of the data input into audio signals, achieving a throughput of 2.5 kbps with $3\text{x}10^{-3}$ BER using EFR. Moreover, Dhananjay et al. [19] proposed an FSK modulation achieving a data rate of 1.2 kbps with $2.6\text{x}10^{-3}$ BER with EFR using simulations. Zdenko et al. [20] describes as modulation method the phase-continuous Orthogonal Frequency Division Multiplexing (OFDM) amplitude shift keying. Chen and Guo [21] analyze the OFDM method of

modulating the input bit stream on orthogonal multi-frequency sinusoidal carriers. Sheikh et al [22] propose the use of QPSK modulation and the use of Gold code sequences to encrypt voice data.

Towards this direction, Biancucci et al. [23] proposed a modulation algorithm based on Frequency Modulation (FM) capable of converting encrypted voice data to a waveform, which conforms to GSM voice channel specifications. In the proposed scheme, the digital data stream is encrypted either by an AES with 256-bit key size or a Self-Synchronizing Stream Cipher (SSSC). The latter (i.e., SSSC ciphers) is not considered secure, because it transmits additional signals, which on the one hand allows synchronization of the stream cipher, but on the other hand, it can be exploited by an attacker to leak information for the encryption key. Evidently, the use of AES is more secure than SSSC, but it has a bigger error propagation and time complexity. The device prototype which is used for evaluation of the proposed scheme is composed of two mobile phones and Desktop computers acting as sender and the other as receiver. The connection between computers and mobile phones is provided by a 3.5 mm jack cable. For encryption the AES-256 is used without however mentioning its mode. As the authors acknowledge, the numerical results showed that the actual modulator bit rate does not allow a real-time communication and the proposed algorithm requires more optimization to achieve the maximum allowed bit rate without increasing BER value.

Finally, a solution proposed by Taleb Ali et al. [24], tuned the parameters of M-ary FSK (M-FSK) modulation to minimize the degradation introduced by vocoders. The authors mention that their goal was to minimize the BER and they didn't require high data rates, since the proposed solution was designed for the WikiWalk system, a system for transferring short GPS frames limited to a few characters to deliver vocal guidance service for pedestrians. Thus, the proposed scheme in [24] is not suitable for speech, which typically requires a high data rate but it can tolerate high BER.

4.2.4 Other solutions

There are also some previous works, which present important contributions in this field of research, but they do not fit into the above categories. One of these works is presented in [25], where the authors proposed a modem for data transmission as well as a protocol for authentication and key agreement. The authors mention that their goal is not to achieve secure voice transmission, which has been already analyzed by previous works, but instead to design a reliable mutual authentication and key agreement protocol for the calling parties over the GSM voice channel. The authors optimize FSK modulation, in order to be able to transmit data, while a novel link layer has been proposed for reliable transmission. More importantly, the authors have designed a TLS inspired protocol for mutual authentication and session key agreement, capable of overcoming the bandwidth limitations of the GSM voice channel. The authors have evaluated their proposed scheme using MATLAB to model the channel characteristics, while they have employed Desktop computers to emulate mobile phones. Numerical results show that they achieved 0.3% BER on average for AMR codec by transmitting 2 kb/s.

Chumchu et al. [26] use an external device (which they have implemented as a prototype that costs less than 40$) and Bluetooth protocol to convey RC4 based encrypted voice to a mobile phone without however, performing thorough experiments. Finally, Ridha et al. [27] propose the use of an external headphone that can be connected to any mobile phone that has an audio jack. This headphone undertakes the task of encrypting speech wavelets using a technique named underdetermined blind source separation. The latter aims at recovering unknown signals or sources from their observed mixture. The authors have performed experiments using MATLAB, but they do not provide results for the achieved throughput and BER.

A theoretical work is presented by Kazemi et al. [28], in which they propose an information-theoretic model for the derivation of analytical bounds on the voice codec capacity. Moreover, they design a decoder to properly detect the transmitted symbols, by conjecturing Weibull and chi-square distributions to model the probability distribution of channel output. The authors have also performed simulations using AMR vocodec to estimate the capacity values based on their proposed model.

5 Software solutions for voice encryption using VoIP

VoIP solutions traverse only data networks and therefore they do not experience the limitations of the previous solutions, which are based on the GSM voice channel as the transmission medium. For this reason, VoIP solutions can be built with advanced security features as they do not face performance issues as their GSM counterparts. As we mentioned in Section 2.2.1, the SIP protocol is a standard for

multimedia applications that use VoIP. It is a signaling protocol to establish a session between two or more VoIP entities. Based on the underlying network architecture, which is used to establish a call, VoIP solutions can be divided into two main categories: i) centralized, and, ii) decentralized topologies. In the following, we will analyze prominent VoIP solutions from each category based on their security and privacy features.

5.1 Centralized topology

In a typical call establishment scenario with SIP, the caller should know the IP address as well as the port number of the callee's device. The SIP architecture relies on centralized authorities (i.e., proxy and registrar servers) to provide this information to the SIP entities and correctly route calls. Therefore, the security of a SIP architecture is based on a third-party relationship which should be trusted (e.g., the VoIP provider is responsible to maintain and distribute encryption keys).

One of the most popular VoIP solutions for mobile devices is Silent Phone[2]. After a call establishment, security is provided by the ZRTP for key exchange. Moreover, voice is encrypted based on SRTP using the AES 256-bit algorithm. It is important to mention that Silent Phone has successfully completed the FIPS 140-2 validation process. This means that the application is certified for use within the US government.

Another popular mobile application that uses centralized topology based on SIP is the Signal Messenger[3], which offers private messages and secure voice calls. The uniqueness of this application stems from the fact that the provided security is based on its own cryptographic protocol named *Signal*. The most important feature of *Signal* is the use of the double Ratchet protocol. The latter is based on a modified Diffie-Hellman key exchange named X3DH, which allows for a key exchange where one party is not available. To make this possible, X3DH brings in a third party, a *"server"*. The server can be a single entity or split across multiple actual computers. X3DH is combined with Key Derivation Function (KDF) chains to form the double Ratchet protocol, which features security properties such as encryption, integrity and authentication as well as plausible deniability and forward secrecy. A unique feature of double Ratchet protocol is called self-healing, because in case an attacker compromises at some point a session key, it automatically prevents obtaining the cleartext of subsequent messages by altering the encryption key. The voice stream and messages are secured with an encrypt-then MAC scheme (encryption is AES with 256 key size in CBC mode and the MAC algorithm is HMAC-SHA256).

At first the Signal protocol was greeted with skepticism, since the practice of introducing new cryptographic protocols is considered a risky process. However, researchers which audited the application concluded that Signal satisfies several standard security protocols and it is cryptographically sound [29]. Nowadays, the Signal protocol is adopted by many popular applications including WhatsApp and Facebook Messenger.

5.2 Decentralized topology

In this section, we analyze applications that avoid using a centralized, server-based infrastructure. Instead, the solutions of this category rely on decentralized topologies to bootstrap the discovery and call setup. The applications are based either on distributed hash tables or blockchains.

5.2.1 Distributed Hash Tables

Ring[4] is a free, multi-platform, mobile and desktop application that uses distributed hash tables (DHT) as the overlay network to establish peer discovery and call setups. DHTs are peer-to-peer networks used for provisioning and support of storing data and retrieving it under a number of nodes which collaborate. DHTs is highly scalable due to their efficient lookup procedure. Another advantageous characteristic of DHT is fault tolerance. To this end, Ring uses a DHT for anonymously establishing communications in an efficient and scalable manner. Moreover, this application offers a digital identity named "RingID", which does not store personal information on the network. Ring stores all secrets (private key for

[2] https://www.silentcircle.com/products- and- solutions/devices/
[3] https://signal.org
[4] https://ring.cx

encryption and identity) only on the machine that runs it, while uses SRTP (thus AES) for encrypting the media traffic.

5.2.2 Blockchain

Crypviser[5] is an innovative solution, which aims to use a blockchain-based authentication model that allows its users to identify and confirm each other's public keys, without relying on a centralized PKI server. More specifically, a blockchain is a distributed database that holds a growing list of records named blocks, which are linked and secured using hash functions. By design, blockchains are inherently resistant to modifications of the data inside the blocks. The use of blockchains eliminates MiTM threats and any kind of manipulation attempts from the server and third parties' sides. Crypviser has also issued a crypto token named CVCoin, which covers charges of blockchain transactions for authentication and authorization purposes, as well as to identify the users' public encryption keys. In the near future, the second version of Crypviser aims to develop the first blockchain in which the participating nodes will be mobile phones.

Another innovative blockchain-based VoIP solution is named ENUMER[6] which is based on the Electronic Number Mapping System (ENUM) protocol [30] on top of the Emercoin blockchain[7]. The ENUM protocol acts as a distributed address book that allows finding the path to a certain IP PBX by a telephone number serviced by it. However, due to its centralized nature, ENUM is prone to denial of service attacks and it has not been widely used. To overcome the above challenges, a decentralized ENUM implementation (i.e., ENUMER) based on the Emercoin blockchain, which acts as a distributed name-value storage (NVS), has been implemented.

6 Hardware-based solutions for voice encryption

In this section, we describe commercial voice encryption solutions, which rely on the use of an extra hardware device. We have identified three types of such devices: i) A headset or a dedicated device that is connected between a headset and mobile phone; ii) A Secure Element (SE) in the form of a microSD implementation or a Trusted Execution Environment (TEE); and iii) A specialized mobile phone. It is important to mention that the list of the solutions that we analyze is not exhaustive. Here we select to present products that are representative and they have relatively good documentation. Moreover, all the solutions of this category are based on VoIP technology. The only notable exception is a device named Jackpair that targets the GSM voice channel (see below).

6.1 Headset implementation

6.1.1 JackPair

JackPair advertises to be a one-click solution that can protect users' privacy with a touch of a button. It is an encryption hardware device that is connected between a phone and a headset through standard 3.5mm audio jacks. JackPair provides an authentication, encryption and key management platform that works over the GSM voice channel without relying on intermediate servers. In order for it to work, both communicating parties should use the JackPair device. Then, either side can push the button to pair up the two devices, triggering Jackpair to encrypt the user's voice using a *"One-Time-Secret Key (OTSK)"* created on the fly. For the key exchange procedure, the Diffie-Hellman protocol is used that manages to keep the key secret even while in an unsecure network. A SAS value (see Section 2.2) is also employed as MiTM protection mechanism for the Diffie-Hellman protocol. The encryption algorithm in JackPair is Salsa20, a 256-bit key stream cipher. Salsa20 uses OTSK as the seed for PRNG (Pseudo Random Number Generator) to create key streams with similar property of One-Time Pad. According to the designers, an AES implementation would not be feasible due to bandwidth and latency constraints of low bit rate voice channels. A Public Key Cryptography implementation was also avoided due to the complexity of PKI (centralized server needed for key management) that would make the JackPair experience less user-friendly.

[5] https://crypviser.network
[6] http://enumer.org
[7] https://emercoin.com/en

Unlike cryptophones or other encryption devices (see below), JackPair comes with a relatively affordable price (i.e., costs less than $100). After its initial release, JackPair plans to be open source for both software and hardware. However, the project has significantly delayed (the initial release date was on December 2014) and at the time of writing this paper it remains unknown when (and if) it will be released to the public.

6.1.2 TopSec Mobile by Rohde & Schwarz

TopSec Mobile is a headset that provides secure end-to-end voice calls in IP-based networks. It can be connected not only to smartphones but also to PCs, fixed-network phones and satellite terminals using the Bluetooth standard 2.0. It operates on all data networks and requires a VoIP server subscription. These devices can be controlled centrally, using the TopSec Administrator software. This software (i.e., TopSec Administrator) makes it possible to create user groups and generate certificates for the TopSec devices, which enable automatic authentication and secure firmware updates.

Regarding the TopSec Mobile security scheme, AES-256 encryption is employed and Elliptic Curve Diffie-Hellman 384-bit is used for key agreement. The authentication procedure is certificate-based and control codes (i.e., which are similar to SAS values of ZRTP) are used as a mechanism against MiTM attacks. Regarding authentication, the devices receive a certificate and generate a public key pair that is used for the authentication and only when this mutual process is successful, the encrypted connection can be established. It's important to highlight the fact that during the encrypted call time, the smartphone's microphone and speaker are disabled, since the Operating System (OS) cannot be trusted. The German Federal Office for Information Security (BSI) has evaluated the encryption device and its security level is classed as *Restricted*.

6.2 Secure Element or Trusted Execution Environment implementations

6.2.1 SecuSUITE for BlackBerry 10

SecuSUITE for BlackBerry 10 is a Secure Element (SE) in the form of a microSD implementation to provide privacy enhancements for mobile users. It contains the NXP SmartMX P5CCTO72 crypto-controller and a PKI coprocessor for authentication, integrated into a microSD. The additional high-speed coprocessor manages to encrypt voice and data communication using AES-128, effectively protecting voice and data (voice, texts, calendar etc.) from 3rd party attacks. The microSD offers up to 4GB of memory capacity and supports all new generation BlackBerry smartphones.

Another SE in the form of a microSD is TrustChip. It employs a 32-bit ARM crypto coprocessor providing AES-256 Galois/Counter Mode (GCM) mode for secure key management & bi-directional authentication, without affecting mobile battery life. Additionally, TrustChip offers accessibility to its developer kit (TrustChip API SDK) for developing custom, secure applications that interact with the microSD implementation.

6.2.2 TrustCall

Trustcall[8] is a voice encryption application for Android and iPhone devices. One interesting feature of this application is that critical - from a security point of view – functionality is executed inside a TEE to provide hardware level security. TEE is a secure area of the processor that guarantees sensitive data and code execution remains in an isolated environment out of reach of the main OS. In this way, TEE provides a level of protection against software attacks. Trustcall supports Trustonic-based TEE implementations that come typically in Samsung mobile devices. On top of TEE, Trustcall application includes several security features, such as anti-debug protection, code obfuscation, jailbreak and root detection.

6.3 Cryptophones

6.3.1 GSMK Cryptophone 500 by GSMK

The CryptoPhone 500 is an Android-based secure mobile phone designed by GSMK. This product advertises secure messaging and VoIP communication on GPRS, 3G & WLAN. It employs two strong encryption algorithms, AES256 and Twofish, as well as 4096 Diffie-Hellman for key exchange and

[8] https://www.trustonic.com/news/company/koolspan-selects-trustonic-for-protecting-trustcall-application-code-integrity/

SHA256 hash function. The employed security scheme is free of centralized or operator-owned key generation, since every encryption key is created locally and destroyed as soon as the call ends. SAS values are also used as a MiTM defense mechanism.

The most important feature, that this particular cryptophone has to offer, is the baseband firewall. Every smartphone, apart from its main OS, is equipped with a second OS (i.e., a real time OS) that runs in the baseband processor of the mobile phone. The baseband modem is responsible for all device activities pertaining to the mobile network (e.g., call setup). A patent-pending software permits the GSMK CryptoPhone 500 to monitor the baseband processor for suspicious activity, detect baseband attacks and initiate the proper countermeasures. The user is alerted whenever the phone is connected to a compromised/fake cell tower that could force the mobile to fall back to 2G and use no encryption during calls.

Moreover, apart from voice encryption and security features, this phone provides enhanced protection from malware attacks in the main OS. Built on top of a Samsung Galaxy SIII device, it runs a hardened version of Android with granular security management and security-optimized components and communication stacks. GSMK CryptoPhone 500 costs $3.500 and its source code is open for security auditing.

6.3.2 Blackphone 2

Blackphone 2 is another cryptophone running an enhanced Android version, named Silent OS. The latter is built specifically to protect the privacy of the device owner and comes preloaded with the Silent Phone application for secure communications and file transfer, as well as 3rd party apps that control all the data shared. Blackphone 2 appears to be more enterprise-oriented, integrating Mobile Device Management (MDM) systems and allowing corporates to create a customized managed space along with user's space for personal purposes, thereby protecting sensitive work data. The device is equipped with secure boot mechanism, in order to protect the system from any unauthorized modification and guarantee its integrity. In case a vulnerability is found in the OS, Silent Circle promises to distribute fast, over-the-air updates that solve the problem within 72 hours.

Concerning the voice communication protection, Blackphone 2 carries out encrypted calls over all data networks via the Silent Phone application that uses ZRTP for key exchange, encryption and MiTM attack protection (i.e., SAS). Regarding the OS related security features of the Blackphone 2, encrypted and self-destructing texts, encrypted file transfer and secure voice calls are offered through the Silent Circle apps. Stored data are by default encrypted with AES128. The enhanced security settings expand those of the Android OS, offering separate encryption, automatic disabling of installation from unknown sources and randomized pin pad display for secure passcode entry. The security settings are all managed under a dedicated application called *"Blackphone Security Center"* to provide easy access to users and let them decide exactly how and when data is shared. The financial cost of this device is $799.

7 Evaluation

7.1 Solutions for voice encryption over GSM voice channel

First, we analyze the proposed solutions for transmission of voice over GSM voice channel (Table 1 summarizes the most representative works of this category). The first observation is that several solutions propose either the use of insecure and outdated encryption algorithms or suggest the adoption of non-standardized methods, which are not compliant with well-established and widely accepted security guidelines (e.g., NIST). To exemplify, insecure algorithms that are utilized in these works are the use of SSSC [23] or RC4 [26], while examples of non-standardized methods are the use of Gold code sequences as a source of randomness [22] and encryption based on the intractability problem of underdetermined blind source separation [27]. The only notable exception is the scheme presented in [16] which propose the use of AES (although in ECB mode which is not considered a secure mode), while the work in [23] propose two different encryption algorithms, which one of them is AES without mentioning however the mode of operation (the other encryption algorithm is SSSC which is not secure).

Another striking observation is that the reviewed solutions neither consider, nor provide information for various encryption parameters. A prime example is the key size, which is neglected from these studies. A larger key can have a drastic effect on the performance of the encryption process. Another parameter which is not considered is the encryption mode. The majority of block ciphers propagate errors in the ciphertext to consecutive blocks (e.g., CBC mode propagates errors while ECB

mode does not). On the other hand, stream ciphers may not propagate errors, since encryption takes place byte by byte independently. However, the downside of stream ciphers is that in case of synchronization loss (i.e., insertion/deletion of bits, bit slips caused by timing errors), the overall performance will be deteriorated, since decryption data will be scrambled, if error correction bits are not adequate. Thus, both block and stream ciphers share their advantages and drawbacks depending on the application environment and the nature of errors (e.g., burst errors). Selecting the appropriate parameters for an end-to-end encryption can be a challenging task and more attention to these details is required [31].

Moreover, there are also some works that do not actually consider an encryption scheme (e.g. [7], [9]- see Table 1). Thus, these works do not prove the feasibility of transmitting encrypted speech over the GSM voice channel. Evidently, encryption entails extra overhead in the voice processing, due to padding and extra computations required, causing delays in the communication. Moreover, errors in the encrypted data during transmission may result in scrambled data that cannot be decrypted, which means that error correcting codes should be properly evaluated and applied in these solutions.

We have also identified that the proposed solutions do not discuss at all how the key agreement will take place. In the absence of a key agreement procedure, all the proposed encryption solutions are incomplete in terms of security. The only notable exception is the work presented in [25] that propose a TLS-based key agreement protocol over GSM voice channel.

Regardless of the employed encryption algorithm, all the proposed schemes are inherently vulnerable to software level attacks. More specifically, modern mobile phones consist of two main hardware components as they have dual CPU architecture. The first one is the application processor and is responsible for running the user's application, while the second one is the baseband processor responsible for radio channel operations and communication with the cellular network. The rich OS (Android or iOS) that runs in the application processor employs several software level defensive mechanisms that protect from buffer overflows (e.g., address space layout randomization, stack cookies, etc.). However, this is not the case for the OS (typically a real time OS) that runs at the baseband processor. Typically, these OS do not have defensive mechanisms to prevent or detect buffer overflows for two main reasons. First, their implementation is rather old, and, second, performance issues may arise from such mechanisms. Also, the software of baseband processor OS is poorly documented and closed source, which means there is a lack of auditing. In the past [32], several buffer overflow attacks in popular mobile devices have been discovered in the OS of the baseband processors, resulting in arbitrary code execution. Thus, we can deduce that a malicious piece of code that injects into the baseband processor can gain access to the encryption keys of the mobile network. Evidently, in case of a key compromise, no solution regardless of the employed encryption algorithm can protect the user’s communication. It is important to mention that from the presented solutions, only the GSMK Cryptophone 500 includes a baseband firewall to protect the baseband processor from attacks.

Moreover, the majority of the studies under examination consider only a single, GSM voice codec, which is mainly the GSM EFR or the GSM-FR. However, as we have mentioned, GSM incorporates different voice codecs with various data rates and characteristics. Thus, a solution for data transmission over the GSM voice channel must cope with different voice codecs, in order to be generic and applicable. Otherwise, its applicability is limited only to the vocoder that it supports. The work in [16] is one of the few works that proposes a generic solution that takes into account multiple vocoders. On a more technical discussion, the authors in [28] identified that the three categories of examined solutions (i.e., parameter mapping, codebook optimization and modulation optimization) share common limitations. First, these methods exhibit suboptimal performance, by making assumptions regarding their decision rules prior transmission. Second, the performance of the above approaches is deteriorated, since they ignore the symbol dependencies due to differential coding. Third, all the presented schemes do not take into account perturbations that may occur (e.g., due to adjustments) by the voice codecs.

From a performance point of view, in general the solutions that use parameter mapping outperform the rest of the schemes that use codebook or modulation optimization. This happens because the parameter mapping method leads to a data signal that resembles a voice waveform and therefore VAD is not triggered, a fact that allows data to pass through the GSM channel. However, due to the fact that the evaluation experiments in the proposed solutions are performed in controlled laboratory conditions [9], we cannot derive safe conclusions for the performance of the proposed solutions. On top of this, the proposed works use different experimental methods for evaluating the proposed modem providing numerical results of the achieved data rate and the obtained BER. That is, some of them use simulation code written in a generic programming language (typically C/C++) or MATLAB to model

the GSM channel, while other use Desktop computers as mobile phones and FPGAs for encryption (see Table 1). Thus, there is no common basis to objectively evaluate and compare the effectiveness of each solution. For example, as shown in Table 1, the work in [18] achieves 3 kbps and 3% BER using simulations, while the work in [9] achieves 0.533 kbps and 26.47% BER using an emulation environment. Thus, performance-wise, we cannot argue for the effectiveness of the proposed solutions. We believe that new works not only should be based on a common testbed for evaluation but also to utilize an environment that provides realistic measurements. Such a realistic testbed can be implemented using a USRP hardware and the OpenBTS open source software that will power the USRP.

Another area that the examined solutions of this category neglect to analyze is battery consumption. Evidently, both encryption and the modifications that the considered solutions introduce in the voice processing steps incur computational overheads that may result in increased battery consumption. Interestingly enough, there is no single scheme that evaluates whether it is energy efficient and can be actually implemented in mobile devices.

Table 1: Comparison of proposed solutions for encryption in GSM voice channel

Solutions	Vocoder	Data Rate (kbps)	BER (%)	Evaluation	Method	Security
Ozkan et al. [11]	FR	0.600	-	Simulations	Parameter mapping	AES-ECB using FPGA
Kotnik et al. [9]	EFR	0.533	26.47	Emulation with PC	Parameter mapping	-
Sapozhnykov et. al. [16]	FR EFR AMR	1.4 2.1 0.76	$4x10^{-3}$ $1x10^{-3}$ 0.00	Emulation using PC	Codebook optimization	-
Chmayssani et al. [18]	EFR	3	3	Simulations	Modulation optimization	-
Katugampala et al. [7]	EFR	3	2.9	Emulation using PC	Parameter mapping	-
Cumchu et al. [26]	AMBE-2000	-	-	Prototype Implementation	-	RC4
Biancucci et al. [23]	RPE-LTP	-	-	Emulation using PC	Modulation Optimization	AES or SSSC
Sheikh et al. [22]	Daubechies wavelets ‘Haar’ and ‘Db10’ (compression only)	-	0.0081 (Haar) 0.0065 (Db10)	-	-	Gold Code Sequences
Ridha et al. [27]	-	-	-	Simulations	-	Blind Source Separation

7.2 Software solutions for voice encryption using VoIP

All the software applications that are based on VoIP technology deploy centralized architectures and require a SIP server to establish the communication channel. However, these applications avoid to clearly mention the use of a centralized server, as they tend to advertise only the encryption and key negotiation algorithms, neglecting other critical operations such as peer discovery which is equally important. Despite the fact that VoIP service providers cannot decrypt and intercept on the communication, since the communication parties perform key exchange in a P2P fashion (e.g., typically through ZRTP protocol), they can still collect and store metadata in the centralized servers used in peer discovery. That is, the VoIP service providers know exactly the identity, the IP address and the nearby location of the communicating parties, as well as the exact time the call took place and how long it lasted. It is also interesting to mention that despite the fact that the privacy leakage can be mitigated using SIP Privacy protocol that allows SIP entities to remain anonymous, no solution has implemented it yet.

To avoid placing trust to third party VoIP service providers, there is a category of mobile applications that use decentralized architectures based on DHT or blockchains. Decentralized

architectures avoid the use of servers for peer discovery and call setup. Thus, the biggest advantage is that users' privacy is enhanced. Apart from security improvements, decentralized architectures are scalable and there are no single points of failure such as in centralized architectures. The public nature of the blockchain guarantees transparency over how applications work and holds a tamper proof record of activities, providing strong incentives for honest behavior. Applications that are based on blockchains can also provide monetary incentives, if they are combined with smart contract technology. As a matter of fact, Crypviser plans to take advantage of smart contracts in a future release. On the other hand, the major drawback of decentralized architectures is the complexity and the fact that, so far, the applications that belong to this category are limited.

Finally, in most of these mobile applications, security relies also on the smartphone OS, where the underlying application runs on top of. In particular, if the OS itself is compromised, then the mobile application will fail to provide secure voice communications, since a malware given enough privileges (i.e., root privileges) can obtain the encryption keys. Due to its popularity, Android has been the prime target of mobile malware in the past years. While more rare, three zero-day vulnerabilities in iOS were exploited in targeted attacks to infect phones (using Pegasus malware[9]). All these cases prove that a defense in depth solution is required and VoIP solutions relying only on the OS security give a false sense of security to their users.

7.3 Hardware-based solutions for voice encryption

Solutions employing external encryption devices appear to be much more secure than the solutions of the other two categories (i.e., encryption over GSM voice channel and voice encryption for VoIP). This is based on the fact that using external devices, the input and output of these devices are encrypted data and even if the mobile phone is compromised, the encryption keys are out of reach of the attacker. Yet again, as mentioned in the website of Jackpair device, it's of a great importance to make sure that when external devices are used, the mobile's actual microphone is disabled. Otherwise, a malware can open the microphone of the infected mobile phone and hear the communication of the user that holds and uses the device. The negative side of this category is the poor or even non-existent documentation of the products. That is, limited information is shared with the potential buyers about the architecture, the communication protocols and the encryption algorithms. Thus, a detailed assessment of the design and implementation of these products is not possible.

The use of extra hardware to perform voice encryption provides another advantageous characteristic compared to software-only solution. As we mentioned previously (see Section 7.2), all mobile applications are vulnerable to software-based attack at the level of the OS. On the contrary, in this category (i.e., hardware-based solutions), Trustcall TEE and SecuSUITE perform security critical functionalities inside TEE and in a MicroSD card respectively. Therefore, they are considered to be immune to software attacks. This happens because SecuSUITE uses a MicroSD as a crypto-processor to perform security critical functionalities and store keys, while Trustcall uses TEE to execute code (i.e., voice encryption/decryption process) in an isolated memory region, which is out of reach from the main smartphone OS. The security properties of MicroSD and TEE guarantee that a malware cannot obtain the encryption keys even if it has root access to OS. However, TEE is not bullet-proof and recently several loopholes have been discovered in the various TEE implementation from different vendors, most notably Qualcomm [33]. In case of a vulnerable TEE implementation, all the security assurances provided by TEE are not valid anymore. Moreover, TEE (and MicroSD-based crypto-processors) do not provide security on the data path between the TEE (or MicroSD) and the microphone. This means, that when voice enters the device through the microphone it is unencrypted and then, it's being handled by the OS to be transferred to the TEE or MicroSD for encryption. Therefore, a malware can have access to the microphone and eavesdrop on the voice data, before enters TEE (or MicroSD) for encryption. In this case, the whole implementation relies again on the security of the OS, similar to the standalone applications.

In case of TEE, a robust solution against this attack would be the use of microphones (and more generally peripherals) that are trusted in the sense that the data path between the microphone and the TEE is also isolated by the main OS. This means that a malware can only obtain access to the trusted microphone only if it breaches the TEE, while compromising the OS is not adequate anymore. As a

[9] https://info.lookout.com/rs/051-esq-475/images/lookout-pegasus-technical-analysis.pdf

matter of fact, Trustonic TEE has implemented trusted keyboard and screen (but not microphone). These two trusted peripherals, which are known as Trusted User Interface (TUI) by the specifications of TEE by GlobalPlatform, protect users from malware that modify the screen (i.e., to trick the user to perform an unwanted action) or capture the touchstrokes of the user (i.e., touchlogger). However, based on our analysis we deduce that microphone is also another important peripheral that needs to be trusted, in order to enable secure applications that utilize TEE.

Regarding cryptophones, all of them run a custom ROM of an Android version. Although the intention of these customizations is to enhance the overall security of the device, a discovered bug[10] in Blackphone proves that the exact opposite can be achieved (i.e., undermining the overall security) if proper auditing and secure implementation practices are not followed. Moreover, cryptophones can be easily identified by service providers via their IMEI number, which may lead to unwanted attention or even denial of service.

From a usability point of view, the commercial products that are based on a headset implementation enforce the users to carry an extra device (e.g., SecuSUITE) or replace their phone with a cryptophone (e.g., Blackphone). On the other hand, solutions that are based on MicroSD or TEE do not share these usability issues. However, there are many mobile devices that do not include a TEE, while some devices (notably iPhones) do not have a MicroSD. Regardless implementation, the majority of the commercial solutions work in pairs. That is, users on both ends need to be equipped with the same product in order for the secure communication to take place. Finally, it is important to mention that the products of this category are not cost effective, except for Jackpair which costs around $100.

8 Future Directions

8.1 Research directions

In this section we present the identified research directions. More specifically, in light of our analysis, we believe that the future work in this area should deal with one or more of the following research issues:

- First, future work should try to optimize data transmission over the GSM voice channel using strong encryption algorithms. The use of outdated, insecure or non-standardized encryption algorithms should be avoided by any means.
- The evaluation of any work should take into account not only performance parameters (e.g., throughput) but also security parameters such as key size, mode of operation (CBC, counter, etc.).
- Research works for GSM voice encryption should also estimate the imposed communication and computation costs. This is of paramount importance in order to assess not only the imposed communication overheads, but also the energy efficiency of the proposed solutions. Power consumption in mobile devices is an important aspect and it has been neglected by the related studies on GSM voice encryption.
- Solutions for GSM voice encryption should evaluate their proposed solutions using a real testbed and avoid simulators/emulators. Open source projects like OpenBTS and easily available hardware such as USRP reduce the barrier to create a testbed for GSM radio communications.
- The design of key agreement protocols appropriate for GSM voice channel should be taken into account by researchers. Without a secure key agreement solution, all encryption schemes are incomplete.
- Solutions in this research field should also take into account the effects of noise cancelation (also known as active noise control). The latter is a method for reducing unwanted sound by the addition of a second sound specifically designed to cancel the unwanted one. Our review of the related works revealed that there is no single solution that considers noise cancelation.
- Future work in this area should provide more open source and freely available tools. To the best of our knowledge, there is no single work that releases to the public the source code of the proposed solution along with the obtained results. Thus, it is not possible to repeat the experiments and assess the research outcomes.
- Moreover, it will be interesting to propose new solutions that take advantage of the UMTS voice channel, where the high available bandwidth can facilitate the design of new modems that provide

[10] https://www.sentinelone.com/blog/vulnerability-in-blackphone-puts-devices-at-risk-for-takeover/

end-to-end security in UMTS voice channel. Currently, there is a significant gap for UMTS voice channel encryption compared to GSM counterparts.

- At the level of mobile devices, a security evaluation of baseband processors both from a software and hardware point of view is of paramount importance. As the research community has focused on the OS security, baseband processors security has been neglected.
- Regarding VoIP technology, we believe that new solutions should take advantage of blockchain as well as smart contract technology to create decentralized applications for secure voice communications, without third party involvement. Mobile-based blockchains for voice communications is also an interesting area of research [34].
- New solutions can take advantage of Trusted Execution Environments (TEE) not only at the level of mobile devices, but also at the server side (using the Intel SGX technology) to come up with innovative secure solutions with verifiable security properties (due to the remote attestation protocols).
- Finally, more research is required for all aspects of the LTE security architecture including the privacy of voice transmission.

8.2 Security considerations for 5G networks

Currently in its final standardization phase, the launch of 5G networks is expected to drastically change the mobile communication landscape, by introducing the concepts of cloudification, softwarization and virtualization in telecommunications. Although it is clear that data services drive the 5G network evolution, voice communication is still considered an important service for the mobile carriers. According to the 3GPP specifications, 5G will use the 4G voice communication architecture and rely on the IP Multimedia Subsystem (IMS) to provide voice communication services. In 5G, New Radio (NR) will succeed the LTE of 4G as radio access technology, and voice over NR (VoNR) will be realized through the extension of certain existing infrastructure, such as IMS. Cloud and virtualization technologies will be highly employed in 5G to reduce costs and offer more flexibility. However, this new service delivery paradigm and network realization will raise new privacy concerns, since an evolved threat landscape will arise, affecting also the voice communication over 5G.

The introduction of 5G will considerably change the trust relationships as new actors will be involved including third-party service providers, cloud providers, infrastructure providers and operators that share the network infrastructure. Additionally, the use of cloud and virtualization is bound to increase the attack surface, resulting in a high dependency on secure software. Decoupling software and hardware means that telecom software can no longer rely on the specific security attributes of a dedicated telecom hardware platform. Therefore, when operators host third-party software in their telecom clouds, executing on the same hardware as native telecom services, there will be increased demands on virtualization with strong isolation properties, in order to ensure absolute security [35].

To address the new advanced security requirements, 5G intends to break out from the SS7 and Diameter and use Internet protocols like HTTP, TLS, and REST API. However, the possible vulnerabilities of these implementations can be exploited more easily compared to the legacy technologies, since they can be included into the existing penetration testing tools. Finally, with the introduction of 5G, carriers will face the challenge of coexistence of 2G, 3G, 4G, and 5G networks, while, at the core network level, the voice communication services will continue using technologies, such as the Circuit Switched Fallback (CSFB) and voice over LTE (VoLTE), as well as the VoNR technology under the NR coverage. This fact will increase the number of interconnections in the network, where each component employs different technologies and protocols, adding in this way new vulnerable points. Based on the above observations, the conjecture is that end-to-end voice encryption solutions specifically designed for 5G networks, should pay attention on the new attack vectors to avoid downgrading the security level that promise to deliver.

9 Conclusions

This paper presented and evaluated a set of end-to-end voice encryption solutions for mobile users. We identified three categories of voice encryption solutions: i) research works that propose schemes that enable the transfer of encrypted speech over the 2nd generation mobile network voice channel; ii) software applications that use Voice over IP technology; iii) hardware-based commercial solutions. Our analysis

showed that several solutions have a false sense of security, as they propose either the use of insecure and outdated encryption algorithms, or suggest the adoption of non-standardized methods, which are not compliant with well-established and widely accepted security guidelines. Specifically, regarding the first category, while some research works advertise the secure communication of voice, they do not actually consider an encryption scheme and thus, in these cases, the feasibility of transmitting encrypted speech over the 2nd generation mobile network voice channel cannot be proved. Moreover, the security of software solutions for Voice over IP is solely based on the assumption that the underlying operating system is not vulnerable to attacks that would allow a malware to infect the device. On the other hand, solutions employing external encryption devices appear to be much more secure than the solutions of the other two categories. Especially for Trusted Execution Environment based mobile applications, our conjecture is that a combination of Trusted Execution Environment and trusted data path between the Trusted Execution Environment and the microphone of the device constitutes a robust solution for mobile devices without the usability and cost issues of external hardware devices. Finally, in order to derive safe conclusions for the performance and security level of each solution, a common testbed and a detailed documentation are absolutely necessary.

Acknowledgments

This work was supported in part by the FutureTPM project of Horizon H2020 Framework Programme of the European Union, under GA number 779391, and by the H2020-MSCA-RISE-2017 SealedGRID project, under GA number 777996.

References

[1] H. B. Wolfe, "The mobile phone as surveillance device: progress, perils, and protective measures.," in IEEE Computer, 2017.

[2] S. Nobuyuki, Y. Igarashi, T. Kaneko and K. Higuchi, "New integral characteristics of KASUMI derived by division property," in International Workshop on Information Security Applications,Springer, 2016.

[3] C.-Y. Li, T. Guan-Hua, P. Chunyi, Y. Zengwen, L. Yuanjie, L. Songwu and W. Xinbing, "Insecurity of voice solution volte in lte mobile networks," in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, 2015.

[4] "Study on mobile device security," Department of Homeland Security (DHS), 2017.

[5] A. Castiglione, G. Cattaneo, G. De Maio and F. Petagna , "SECR3T: Secure End-to-End Communication over 3G Telecommunication Networks," in Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, 2011.

[6] M. Boloursaz, A. Hadavi, R. Kazemi and F. Behnia, "Secure data communication through GSM Adaptive Multi Rate voice channel," in 6th International Symposium on Telecommunications (IST), 2012.

[7] N. Katugampala, K. Al-Naimi, S. Villette and A. Kondoz, "Real-time end-to-end secure voice communications over GSM voice channel," in 2005 13th European Signal Processing Conference, 2005.

[8] M. Rashidi, A. Sayadiyan and P. Mowlaee, "Data Mapping onto Speech-like Signal to Transmission over the GSM Voice Channel," in 2008 40th Southeastern Symposium on System Theory (SSST), 2008.

[9] B. Kotnik, Z. Mezgec, J. Svecko and A. Chowdhury, "Data transmission over GSM voice channel using digital modulation technique based on autoregressive modeling of speech production," Digital Signal Processing, vol. 19, no. 4, pp. 612-627, July 2009.

[10] B. Boloursaz, R. Kazemi, D. Nashtaali, M. Nasiri and F. Behnia, "Secure data over GSM based on algebraic codebooks," in East-West Design & Test Symposium (EWDTS 2013), 2013.

[11] M. A. Ozkan, B. Ors and G. Saldamli, "Secure voice communication via GSM network," in 7th International Conference on Electrical and Electronics Engineering (ELECO), 2011.

[12] Y. Yang, S. Feng, W. Ye and X. Ji, "A Transmission Scheme for Encrypted Speech over GSM Network," in International Symposium on Computer Science and Computational Technology, 2008.

[13] C. K. LaDue, V. V. Sapozhnykov and K. S. Fienberg, "A Data Modem for GSM Voice Channel," IEEE Transactions on Vehicular Technology, vol. 57, no. 4, pp. 2205-2218, 2008.

[14] A. Shahbazi, A. H. Rezaie, A. Sayadiyan and S. Mosayyebpour, "A novel speech-like symbol design for data transmission through GSM voice channel," in IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), 2009.

[15] A. Shahbazi, A. H. Rezaei, A. Sayadiyan and S. Mosayyebpour, "Data Transmission over GSM Adaptive Multi Rate Voice Channel Using Speech-Like Symbols," in International Conference on Signal Acquisition and Processing, 2010.

[16] V. V. Sapozhnykov and K. S. Fienberg, "A Low-rate Data Transfer Technique for Compressed Voice Channels," Journal of Signal Processing Systems, vol. 68, no. 2, pp. 151-170, 2012.

[17] M. Boloursaz, A. H. Hadavi, R. Kazemi and F. Behnia, "A data modem for GSM Adaptive Multi Rate voice channel," in East-West Design & Test Symposium (EWDTS 2013), 2013.

[18] T. Chmayssani and G. Baudoin, "Data transmission over voice dedicated channels using digital modulations," in 18th International Conference Radioelektronika, 2008.

[19] A. Dhananjay, A. Sharma, M. Paik, J. Chen, T. Karthik, J. Li and L. Subramanian, "Hermes: data transmission over unknown voice channels," in MobiCom, 2010.

[20] Z. Mezgec, A. Chowdhury and B. Kotnik, "Implementation of PCCD-OFDM-ASK RobustData Transmission over GSM Speech Channel," INFORMATICA, vol. 20, no. 1, pp. 51-78, 2009.

[21] L. Chen and Q. Guo, "An OFDM-based secure data communicating scheme in GSM voice channel," in International Conference on Electronics, Communications and Control (ICECC), 2011.

[22] J. A. Sheikh, S. Akhtar, S. A. Parah and G. M. Bhat, "A New Method of Haar and Db10 Based Secured Compressed Data Transmission Over GSM Voice Channel," in Intelligent Techniques in Signal Processing for Multimedia Security, Springer International Publishing Switzerland, 2016, pp. 401-426.

[23] G. Biancucci, A. Claudi and A. F. Dragoni, "Secure Data and Voice Transmission over GSM Voice Channel: Applications for Secure Communications," in 4th International Conference on Intelligent Systems, Modelling and Simulation, 2013.

[24] G. B. O. V. Béchir Taleb Ali, "Data transmission over mobile voice channel based on M-FSK modulation," in IEEE Wireless Communications and Networking Conference (WCNC), 2013.

[25] B. Reaves, L. Blue and P. Traynor, "AuthLoop: End-to-End Cryptographic Authentication for Telephony over Voice Channels," in Proceedings of the 25th USENIX Conference on Security Symposium, 2016.

[26] P. Chumchu, A. Phayak and P. Dokpikul, "A simple and cheap end-to-end voice encryption framework over GSM-based networks," in Computing, Communications and Applications Conference, 2012.

[27] O. A. L. A. Ridha, G. N. Jawad and S. F. Kadhim, "Modified Blind Source Separation for Securing End-to-End Mobile Voice Calls," IEEE Communications Letters, vol. 22, no. 10, pp. 2072-2075, 2018.

[28] R. Kazemi, M. Boloursaz, S. M. Etemadi and F. Behnia, "Capacity Bounds and Detection Schemes for Data Over Voice," IEEE Transactions on Vehicular Technology, vol. 65, no. 11, pp. 8964 - 8977, 2016.

[29] K. Cohn-Gordon, C. Cremers, B. Dowling, L. Garratt and D. Stebila, "A Formal Security Analysis of the Signal Messaging Protocol," in IEEE European Symposium on Security and Privacy (EuroS&P), 2017.

[30] S. Bradner, L. Conroy and K. Fujiwara, Writers, The E.164 to Uniform Resource Identifiers (URI), Dynamic Delegation Discovery System (DDDS) Application (ENUM), RFC 6116. [Performance]. Harvard University, 2011.

[31] A. D. Elbayoumy and S. Shepherd, "Stream or block cipher for securing VoIP?," International Journal of Network Security , vol. 5, no. 2, pp. 128-133, 2007.

[32] R.-P. Weinmann, "Baseband Attacks: Remote Exploitation of Memory Corruptions in Cellular Protocol Stacks," in 6th USENIX conference on Offensive Technologies, 2012.

[33] A. Machiry, E. Gustafson, C. Spensky, C. Salls, N. Stephens, R. Wang, A. Bianchi, Y. Ryn Choe, C. Kruegel and G. Vigna, "BOOMERANG: Exploiting the Semantic Gap in Trusted Execution Environments," NDSS Symposium, 2017.

[34] K. Suankaewmanee, D. Thai Hoang, D. Niyato, S. Sawadsitang, P. Wang and Z. Han, "Performance Analysis and Application of Mobile Blockchain," in International Conference on Computing, Networking and Communications (ICNC), 2018.

[35] “5G security - scenarios and solutions,”Ericsson White paper, 2017.