Introducing a Verified Authenticated Key Exchange Protocol over Voice Channels for Secure Voice Communication

Piotr Krasnowski[1,2], Jerome Lebrun[1] and Bruno Martin[1]
[1]*Univ. Côte d'Azur, I3S-CNRS, Sophia Antipolis, France*
[2]*BlackBoxSécu, Sophia Antipolis, France*

Keywords: Authenticated Key Exchange, Secure Voice Communications, Data over Voice, Vocal Verification, Crypto Phone, Tamarin Prover, Formal Protocol Verification.

Abstract: Increasing need for secure voice communication is leading to new ideas for securing voice transmission. This work relates to a relatively new concept of sending encrypted speech as pseudo-speech in audio domain over existing civilian voice communication infrastructure, like 2G-4G networks and VoIP. Such a setting is more universal compared to military "Crypto Phones" and can be opened for public evaluation. Nevertheless, secure communication requires a prior exchange of cryptographic keys over voice channels, without reliance on any Public Key Infrastructure (PKI).
This work presents the first formally verified and authenticated key exchange (AKE) over voice channels for secure military-grade voice communications. It describes the operational principles of the novel communication system and enlists its security requirements. The voice channel characteristics in the context of AKE protocol execution is thoroughly explained, with a strong emphasis on 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). The protocol is detailed and analyzed in terms of fundamental security properties and successfuly verified in a symbolic model using Tamarin Prover.

1 INTRODUCTION

An increasing concern of privacy violation in voice communications has motivated the development of secure voice over IP (VoIP) communicators, with Telegram and Signal being the iconic examples[1,2]. However, these applications are inherently insecure against spying malware installed on the smart-phone (Scott-Railton et al., 2017). Parallely, military-grade applications requiring higher protection rely on dedicated hardware, most commonly in the form of Crypto Phones. These closed and unverifiable solutions suffer from high costs and low flexibility, as typically encrypted phones allow communications exclusively over one 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 on Fig.1, is based on voice encryption in the audio domain. The

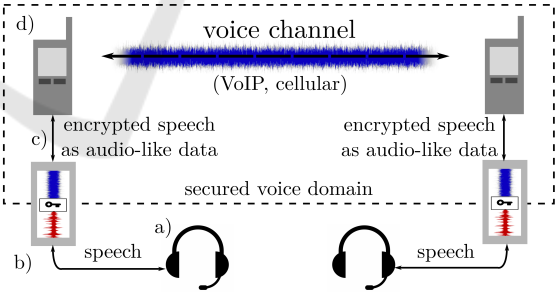

Figure 1: Encrypted voice over voice channel scheme.

speech is recorded 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 data stream shaped into pseudo-speech audio signal 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.

In such a setting, voice encryption is performed

[1]https://signal.org
[2]https://core.telegram.org

Krasnowski, P., Lebrun, J. and Martin, B.
Introducing a Verified Authenticated Key Exchange Protocol over Voice Channels for Secure Voice Communication.
DOI: 10.5220/0009156506830690
In *Proceedings of the 6th International Conference on Information Systems Security and Privacy (ICISSP 2020)*, pages 683-690
ISBN: 978-989-758-399-5; ISSN: 2184-4356

outside of the phone, hence protecting against audio-recording malware. To limit the risk of a system corruption, the Crypto Box has only analog input/output interfaces to the headset and to the phone. However, for security reasons, it is necessary that other analog inputs of the phone (particularly the built-in microphone) are blocked by a special case or removed.

From the system perspective, two Crypto Boxes are the end-points of a secured voice domain. Everything in between, including mobile phones itself, are elements of 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 is done purely in the analog domain, the selection of the specific voice communication technology is therefore a secondary issue. Compatibility with most of the 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 an unreliable environment and without the access to a confidential communication infrastructure.

The major motivation in our approach is to secure voice communication even with untrusted phones, as these should not be actively involved in the setup of a secure connection or store sensitive data. Instead, the trust is given to Crypto Box manufacturers, responsible for software implementation or update policy. Though, the open framework enables various hardware solutions, including combining the phone and the Crypto Box into one device.

Producing encrypted speech in real-time appears to be quite technically challenging. Firstly, the recorded speech is encoded into the vocal parameters in a similar manner as during speech compression. Later, the speech parameters are encrypted and mapped onto the audio waveform. This technique, called Data over Voice (DoV), proved its feasibility in practical scenarios (Katugampala et al., 2004; Shahbazi et al., 2009; Dhananjay et al., 2010; Biancucci et al., 2013). However, since voice channels are designed to carry voice signal without much loss of perceptual quality, which is a different goal than the transmission of data, the achievable bitrate for DoV typically is at most 2 kbps. Even in case of modern digital VoIP applications, the received voice is much distorted compared to the input signal, making the transmission resembling a communication over highly distortive analog channel. Sending encrypted voice with such constraints is possible thanks to strong error correction and voice compression by coders like MELP or Codec2.

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, which gives 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, as an extension of the Real-Time Transport Protocol, 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 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, having three modes of operation and extensive negotiation signaling, even ZRTP seems to be overly complex for communication over voice channels. Moreover, none of the protocols put a sufficient emphasis on resistance to strong message distortion or desynchronization in low-bandwidth environment.

To the best of authors' knowledge, this is the first paper focusing on authenticated key exchange (AKE) protocols over voice channels. The work aims at giving the understanding of the very specific channel constraints, leading to a protocol highly adapted to voice channel characteristics and system requirements. The protocol provides double authentication in a single mode of operation, by signatures and vocal comparison of SAS. In addition, it is flexible enough to support authentication of user who did not yet share the signing public keys between each other, with SAS-only authentication or unilateral signature authentication. Finally, the same protocol can be used to authenticate the exchange of signing public keys.

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 first concern is recording and analyzing the network traffic by omnipresent passive eavesdroppers. Active adversaries controlling the network are more likely to block or distort communication, which is technically very

simple. However, a powerful and knowledgeable adversary 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) is considered secure.

A design process of the protocol is motivated by an anticipated user experience. However, due to severe constraints of the voice channel characteristics, the biggest challenges are related to protocol complexity, synchronization and robustness. A major bottleneck is a large message round-trip time, around 2 seconds long, which causes the whole protocol runtime prohibitively long even in case of simple protocols. Another limitation is a very small bandwidth implying a reduction of the message size. Moreover, the protocol has to be robust against fading and signal distortion, requiring a significant simplification of signalization and strong error correction mechanisms. Finally, in order to decrease battery power consumption, cryptographic operations should be rather lightweight and optimized. When implementing, relying on popular and verified network security libraries, like OpenSSL or NaCL, could be a strong practical advantage.

Adaptation to hardware and channel constraints should not lead to 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 derivation of the session key, putting a special emphasis on preventing Man-In-The-Middle (MITM) attacks and achieving Perfect Forward Secrecy (PFS). The crucial property of the protocol is to enable the authentication of peers, 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 provide a comfortable communication. Each received message can be used to effectively estimate channel characteristics and to improve decoding capabilities.

3 PROTOCOL DESCRIPTION

This section presents the symbolic model of the authenticated key exchange protocol over voice channels and provides a brief explanation.

3.1 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 first need to establish a non-encrypted voice connection with a preferred voice application. Then, they can initiate a secure communication. The system model assumes that identity information used to make a call (phone number, user account, credentials etc.) is independent from the authentic user identity and from 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 adversary tries to send multiple messages to a recipient, are not effectively different than 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 are able to detect random channel errors and differentiate them from intentional malicious manipulations.

3.2 Symbolic Model of the Protocol

The proposed protocol, that is presented on Figure 2 next page, relies on Ephemeral (Elliptic-Curve) Diffie-Hellman (EC)DHE exchange (Hankerson et al., 2005), authenticated by signatures (existentially unforgeable and deterministic) or Short Authentication Strings. 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 an authenticated way before the communication starts and are stored in the Crypto Box address book. However, in many real scenarios it is not possible to properly provide such a verification key. If the signature cannot be verified by the recipient, the protocol offers vocal verification as an alternative, which authenticates 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 acknowledgement and the optional vocal verification. Table 1 contains the glossary of terms used in the protocol specification, along with their bit-lengths.

Table 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	X

Setup: The negotiation stage has been considerably simplified. Participants have to mutually agree on starting the key exchange procedure, therefore the actual key exchange protocol is preceded only by fast and automatic role negotiation in order 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 values that are used to derive the Session Key (K_S) and the *SAS*. Alice sends her public *ID*, the nonce, the ephemeral public key and the hash, with 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 noting 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 a reception of Bob's acknowledgment by Alice. The acknowledgment is a confirmation of error-less message reception, so can be non-encrypted.

Short Authentication String Comparison: Each participant can request for vocally challenging *SAS* equality with the peer. *SAS* comparison is obligatory

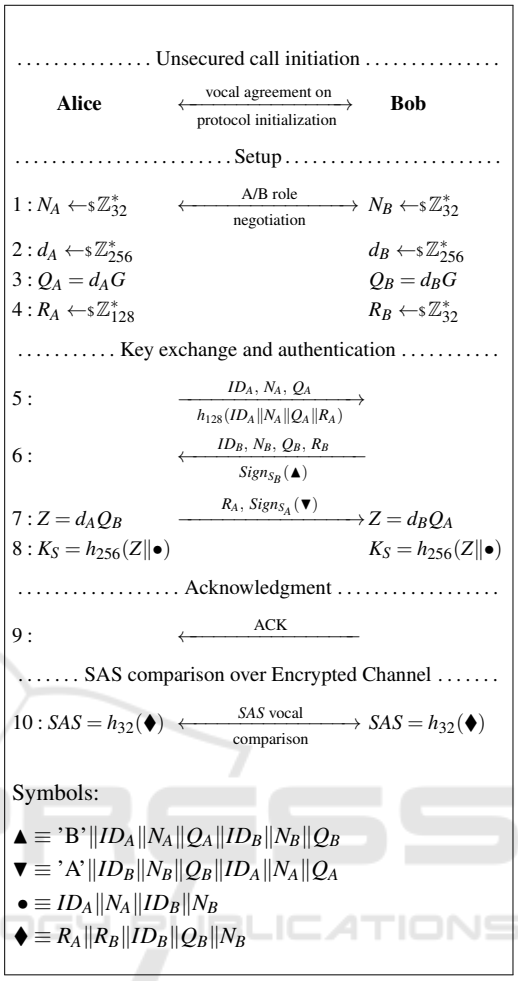

Figure 2: Key exchange protocol over voice channels.

if any of the users was not able to verify the signature. It is assumed that the comparison process is authenticated - users are able to recognize voice characteristics of the peer (timbre, tempo, etc.). The *SAS* is displayed on the Crypto Box as a short string of digits or words to be vocally uttered by the users.

4 FORMAL VERIFICATION

Verification of the protocol is performed in a symbolic model, where all cryptographic primitives are assumed perfect and give the adversary no advantage (Dolev and Yao, 1983). In the analyzed scenario, it means that all parties generate truly random numbers, signatures are unforgeable and ECDH parameters do not reveal any secret information. Formal symbolic verification can be considered as a first step of a protocol analysis, paving the way to computational model verification (Goldwasser and Micali, 1984; Blanchet,

2012), in which the adversary gets the power to attack cryptographic algorithms.

A formal analysis in a symbolic model of the proposed protocol was done with Tamarin Prover (Schmidt, 2012; Meier, 2013; Meier et al., 2013), a powerful and increasingly popular automatic verification tool designed at ETH Zürich. Tamarin is based on multiset rewriting (MSR) language and supports generic Diffie-Hellman group operations. In addition, Tamarin can model many cryptographic primitives like signatures or hashes and offers an impressive database of examples, that makes the tool suitable for the protocol evaluation.

4.1 Protocol Modeling

Verification by Tamarin implies providing an abstract protocol model, which tries to faithfully express relevant information from the security perspective, but still being within the scope of feasibility of the analysis. The protocol model code can be found in (Krasnowski, 2019). Several protocol restrictions were relaxed, 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 protocol rule which is not obligatory, simulating a realistic case when users simply ignore it. Vocal challenging is modeled as communicating over an authenticated (not secret) channel, that is a channel which the adversary can intercept but not modify. Last ACK message is skipped.

4.2 Security Properties and Verification Results

The protocol model was checked against the Dolev-Yao adversary (Dolev and Yao, 1983), having a full control over the network and with the power to reveal the long-term secret key of any user (ephemeral data is considered secure). Evaluation was done in four authentication configurations: mutual signature authentication between two honest users, unilateral signature authentication (when only one user can verify the signature of the peer), 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 for signing key compromise impersonation (adversary can impersonate only corrupted users).

Results of protocol verification can be found in Table 2. Protocol configurations involving signature authentication or authenticated *SAS* comparison are proven to provide perfect forward secrecy and injective agreement. Unilateral signature authentication between two honest users who know each other guarantees the same level of security as mutual signature authentication. Surprisingly, vocal verification does not protect against reflection attack, because the user can trivially compare *SAS* with herself. Table results indicate the importance of authentication - none of the properties were verified if no authentication was performed.

Table 2: Security properties verified by Tamarin in four authentication scenarios: (a) mutual signature authentication, (b) unilateral signature authentication, (c) SAS vocal verification and (d) no authentication.

Authentication scenario:	(a)	(b)	(c)	(d)
Session Key secrecy	✓	✓	✓	✗
forward secrecy	✓	✓	✓	✗
injective agreement	✓	✓	✓	✗
reflection attack	✓	✓	✗	✗
key compromise impersonation	✓	✓	-	-

5 SECURITY CONSIDERATIONS

The following sections explain in more detail the protocol characteristics, providing several justifications and practical recommendations. It starts from the overview of fundamental protocol elements: the choice of public-based cryptography, the role of signatures and of Short Authentication Strings. Next section enlists potential protocol weaknesses and possible fixes.

5.1 Discussion

Public Key Agreement versus Symmetric Cryptography: In exceptionally constrained resource devices, such as IoT sensors or RFID cards, a pursue for ultra-lightweight key exchange protocols led to the shift from the public key encryption towards symmetric encryption techniques (Echevarria et al., 2016; Lee et al., 2014; Baashirah and Abuzneid, 2018). Even the ZRTP protocol offers a possibility of a key exchange in a lightweight preshared mode. In this configuration, two entities share a secret which is used to encrypt or refresh the keying material for the new session. In order 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, in voice channels such a risk cannot be eliminated, 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 the track of all key updates or a costly and potentially unsecure "guessing" 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 not possible, limiting the usability of symmetric cryptography. In the light of above-mentioned reasons and relatively smaller hardware restrictions compared to IoT sensors, public-based key exchange scheme seems more adequate.

Role of Short Authentication Strings: If the key exchange is not interfered 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 authentication of users based on voice identification. Secondly, the inequality of codes may indicate the presence of an active MITM adversary. However, MITM manipulations would be undetected if the adversary is somehow able to influence or precompute the *SAS* value before the users.

Computation of the code 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 the peer - Alice by sending the hash of R_A in the first message and Bob by revealing R_B before R_A. Such construction, inspired by (Pasini and Vaudenay, 2006), prevents adaptive selection of seeds by each party. The same rule applies to the adversary, 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 a 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 noting, that *SAS* value does not have to be confidential, since it plays only the authentication role and cannot be modified without detection.

In practice, the security of vocal verification depends also on how users abide to 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 (Callas et al., 2011; Zimmermann, 1996). The device should encourage the mutual *SAS* comparison by indicating a part of the *SAS* to pronounce and a part to hear from the peer.

Signature-based Authentication: Signatures provide 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 higher computational cost. A 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. In consequence, if the user is corrupted, the adversary should be able to impersonate only that person.

When one user cannot obtain a verification key due to 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: both honest users securely exchange a secret or the adversary 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 adversary cannot interfere undetected in another way than preventing the successful exchange. However, the user who failed to authenticate the peer is still complied to challenge the *SAS*, because from her perspective it is the only formal way to verify the absence of the MITM manipulations.

A signature key management policy, due to a lack of any PKI infrastructure, has a crucial impact on the system security and usability. This work points out two possible schemes, decentralized and fully centralized, which can be chosen depending on the needs. In a centralized system, keys are managed by an offline central authority, keeping the track of all records and being responsible for key distribution and update. In a decentralized case, each user is entitled to generate her own key pair and distribute public keys to specific users in authenticated way. Following the PGP model, sharing the key can be performed remotely based on speaker identification and vocal authentication of the channel. Thus, the proposed protocol with SAS comparison gives the possibility to authenticate the exchange of signature verification keys.

Identity Protection: In many situations protecting the identity of the user is as important as securing the

content of the speech. However, calling anybody with a civilian communication networks 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 at least 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. The *ID* and the signatures of Alice and Bob can be sent encrypted with the key derived from DH secret exchanged during first message round-trip, in a similar manner as in the Initial Exchange of IKEv2 standard (Kaufman et al., 2010). The complexity of a protocol providing anonymity would increase, since it will require additional data encryption. It is also important to carefully evaluate the way the encryption key is derived and how it is related to the the session key, in order to give no foothold for cryptanalysis.

5.2 Possible Attacks and Threats

Many protocol vulnerabilities are focusing on the selection of specific cryptographic algorithms, its implementation and finally on compliance to 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 speaker's voice (Gao et al., 2018) question the level of authentication provided by voice recognition. Instead of breaking the *SAS* security, the adversary may simply simulate or replay the speaker pronouncing the code (Shirvanian et al., 2018). The risk is amplified by the fact that the voice sent is highly compressed and thus significantly differs from its real characteristics. For this reason, it is recommended to extend 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 adversary.

If honest users are capable to verify signatures of each other and of achieving strong authentication, the adversary 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 *ID*s along the *SAS*. However, the real solution would be to force signature verification during each protocol by default.

Finally, the proposed protocol cannot protect against the consequences of a device being stolen or misused, giving the responsibility to the manufacturer to provide strong enough password or biometric protection. To minimize the negative consequences of a theft, the device should be protected against physical tampering, making reverse engineering very difficult.

6 CONCLUSIONS

Our work is the first attempt to resolve the problem of cryptographic key exchange over voice channels for military-grade secure voice communications. It also introduces challenges related to secure communications over voice channels like extremely small 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 the importance of user authentication in absence of a trusted server. All these concerns and limitations justify the need of a dedicated protocol instead of relying fully on standardized solutions.

We proposed a simplified key exchange protocol between two honest parties which is based on the ephemeral elliptic curve Diffie-Hellman (ECDHE) protocol. The protocol offers two ways of authentication - by signatures and by vocally challenging the equality of the Short Authentication Strings. A symbolic model of the protocol was analyzed using Tamarin Prover in order to verify the crucial security properties as Perfect Forward Secrecy and mutual agreement on the Session Key. The process of the verification was explained, pointing out the limitations of a symbolic analysis, particularly model simplifications and perfect cryptography assumption.

Formal verification was followed by the discussion of the protocol properties, like unilateral authentication provided by one-side signature verification or the role of vocal comparison in preventing MITM attacks. The analysis led to the observation that all analyzed techniques in themselves do not provide perfect authentication, thus informal identity authentication methods has 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 serve as a guide for engineers working on the implementations of exchange protocols over voice channels or in similar scenarios.

ACKNOWLEDGEMENTS

This work is supported by grant DGA Cifre-Defense program No 01D17022178 DGA/DS/MRIS.

REFERENCES

Arkko, J., Carrara, E., Lindholm, F., Norrman, K., and Naslund, M. (2004). Mikey: Multimedia Internet KEYing (Proposed standard RFC3830). Network Working Group. Retrieved from https://tools.ietf.org/html/rfc3830.

Baashirah, R. and Abuzneid, A. (2018). Survey on prominent RFID authentication protocols for passive tags. *Sensors*, 18(10):3584.

Biancucci, G., Claudi, A., and Dragoni, A. F. (2013). Secure data and voice transmission over GSM voice channel: Applications for secure communications. In *2013 4th International Conference on Intelligent Systems, Modelling and Simulation*, pages 230–233. IEEE.

Blanchet, B. (2012). Security protocol verification: Symbolic and computational models. In *Proceedings of the First international conference on Principles of Security and Trust*, pages 3–29. Springer-Verlag.

Boyd, C. and Mathuria, A. (2003). *Protocols for authentication and key establishment*, volume 1. Springer.

Callas, J., Johnston, A., and Zimmermann, P. (2011). ZRTP: Media path key agreement for unicast secure RTP (RFC6189). IETF. Retrieved from https://tools.ietf.org/html/rfc6189.

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*, pages 113–124. ACM.

Dodis, Y. and Fiore, D. (2017). Unilaterally-authenticated key exchange. In *International Conference on Financial Cryptography and Data Security*, pages 542–560. Springer.

Dolev, D. and Yao, A. (1983). On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208.

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.

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.

Goldwasser, S. and Micali, S. (1984). Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299.

Hankerson, D., Menezes, A. J., and Vanstone, S. (2005). Guide to elliptic curve cryptography. *Computing Reviews*, 46(1):13.

Katugampala, N., Al-Naimi, K., Villette, S., and Kondoz, A. (2004). Real time data transmission over GSM voice channel for secure voice & data applications. In *2nd IEE Secure Mobile Commmunications Forum: Exploring the Technical Challenges in Secure GSM and WLAN*. IET.

Kaufman, C., Hoffman, P., Nir, Y., and Eronen, P. (2010). Internet Key Exchange Protocol Version 2 (IKEv2) (RFC7296). IETF.

Krasnowski, P. (2019). Tamarin code of the key exchange over voice channels. Available at https://github.com/PiotrKrasnowski/AKE_over_Voice.

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.

Lowe, G. (1997). A hierarchy of authentication specifications. In *Proceedings 10th Computer Security Foundations Workshop*, pages 31–43. IEEE.

Maurer, U., Tackmann, B., and Coretti, S. (2013). Key Exchange with Unilateral Authentication: Composable security definition and modular protocol design. *IACR Cryptology ePrint Archive*, 2013:555.

Meier, S. (2013). *Advancing automated security protocol verification*. PhD thesis, ETH Zurich.

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.

Pasini, S. and Vaudenay, S. (2006). SAS-based authenticated key agreement. In *Public Key Cryptography - PKC 2006*, pages 395–409. Springer Berlin Heidelberg.

Schmidt, B. (2012). *Formal analysis of key exchange protocols and physical protocols*. PhD thesis, ETH Zurich.

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. Technical report.

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)*, pages 478–483. IEEE.

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(3):311–333.

Zimmermann, P. (1996). PGPfone Owner's Manual. Pretty Good Privacy. Retrieved from http://web.mit.edu/network/pgpfone/manual/.