CCCP: Closed Caption Crypto Phones to Resist MITM Attacks, Human Errors and Click-Through

Maliheh Shirvanian
University of Alabama at Birmingham
Birmingham, Alabama
maliheh@uab.edu

Nitesh Saxena
University of Alabama at Birmingham
Birmingham, Alabama
saxena@uab.edu

ABSTRACT

Crypto Phones aim to establish *end-to-end* secure voice (and text) communications based on human-centric (usually) short checksum validation. They require end users to perform: (1) *checksum comparison* to detect traditional data-based man-in-the-middle (data MITM) attacks, and, optionally, (2) *speaker verification* to detect sophisticated voice-based man-in-the-middle (voice MITM) attacks. However, research shows that both tasks are prone to human errors making Crypto Phones highly vulnerable to MITM attacks, especially to data MITM given the prominence of these attacks. Further, human errors under benign settings undermine usability since legitimate calls would often need to be rejected.

We introduce *Closed Captioning Crypto Phones* (CCCP), that remove the human user from the loop of checksum comparison by utilizing *speech transcription*. CCCP simply requires the user to announce the checksum to the other party—the system automatically transcribes the spoken checksum and performs the comparison. Automating checksum comparisons offers many key advantages over traditional designs: (1) the chances of data MITM due to human errors and *"click-through"* could be highly reduced (even eliminated); (2) longer checksums can be utilized, which increases the protocol security against data MITM; (3) users' cognitive burden is reduced due to the need to perform only a *single task*, thereby lowering the potential of human errors.

As a main component of CCCP, we first design and implement an automated checksum comparison tool based on standard *Speech to Text* engines. To evaluate the security and usability benefits of CCCP, we then design and conduct an online user study that mimics a realistic VoIP scenario, and collect and transcribe a comprehensive data set spoken by a wide variety of speakers in real-life conditions. Our study results demonstrate that, by using our automated checksum comparison, CCCP can *completely resist* data MITM, while significantly reducing human errors in the benign case compared to the traditional approach. They also show that CCCP may help reduce the likelihood of voice MITM. Finally, we discuss how CCCP can be improved by designing specialized transcribers and carefully selected checksum dictionaries, and how it can be integrated with existing Crypto Phones to bolster their security and usability.

CCS '17, October 30-November 3, 2017, Dallas, TX, USA
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4946-8/17/10…$15.00
https://doi.org/10.1145/3133956.3134013

CCS CONCEPTS

• **Security and privacy → Distributed systems security**;

KEYWORDS

VoIP security; end-to-end encryption; SAS validation; key exchange validation; mobile app security

1 INTRODUCTION

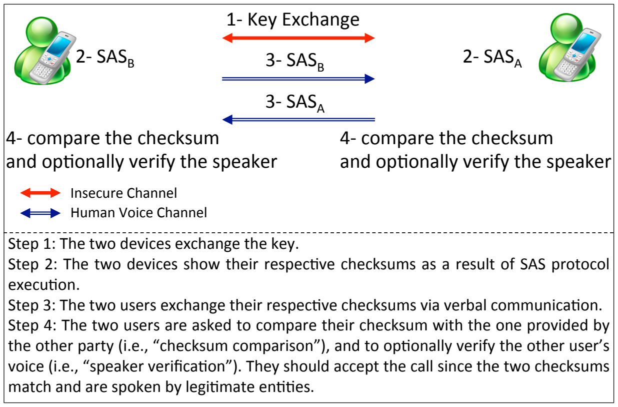
Online voice, video and text communications (VoIP) is one of the most dominant means of real-time communication deployed today. This popularity is exemplified by a plethora of VoIP applications, such as Skype, Viber, WhatsApp or FaceTime, enjoying a burgeoning user base. In contrast to traditional telephony networks, VoIP communication may be more easily susceptible to various forms of attacks, including eavesdropping [3, 4] and man-in-the-middle (MITM) attacks [2, 63]. Governments, intelligence agencies, private organizations, and cyber criminals, often monitor VoIP calls [15], for criminal investigation, political or military endeavors [1], and theft of sensitive information [20]. Considering these vulnerabilities, a fundamental security task is to protect, that is, encrypt as well as authenticate all VoIP sessions. Ideally, this objective should be achieved without relying on third-parties (e.g., an online server) or a centralized infrastructure (e.g., PKI) because such centralized services may themselves get compromised, be malicious or under coercion of law enforcement authorities.

Crypto Phones, such as Zfone [33], Silent Circle [28], and Signal [24] are mobile, PC or web-based VoIP applications that aim to offer end-to-end VoIP security guarantees based on a *decentralized, human-centric* mechanism. Crypto Phones seem to be in high demand in both commercial and personal domains [25]. Prominent mobile apps, WhatsApp and Viber, have also started to offer a similar end-to-end security feature [36, 37].

In order to secure the voice, video or even text communications, Crypto Phones require a cryptographic key, which is agreed upon by the end parties using a specialized key exchange protocol (e.g., [40, 62]). This protocol produces a usually *short* (e.g., 16-bit or 2-word) checksum, called a *Short Authenticated String (SAS)*, per each communicating party, with the characteristic that if an MITM attacker attempts to interfere with the protocol, the *checksums will not match.*

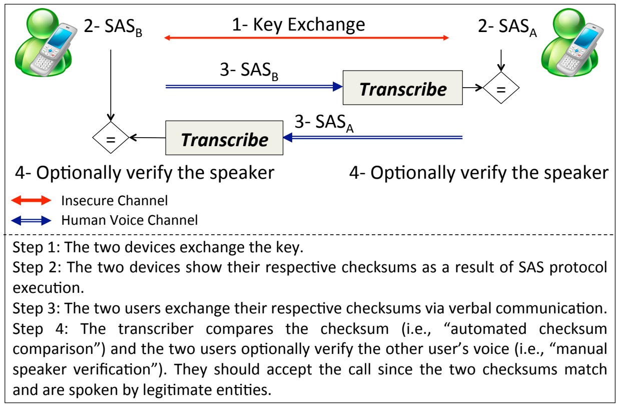
To ensure that the MITM attacker does not interfere with the protocol messages and compromise the protocol security (over the data/voice channel), Crypto Phones rely upon the end users to perform the following tasks (Figure 1 visualizes the benign setting):

- **Checksum Comparison (required):** Verbally communicating and matching checksums displayed on each user's device. This



Step 1: The two devices exchange the key.
Step 2: The two devices show their respective checksums as a result of SAS protocol execution.
Step 3: The two users exchange their respective checksums via verbal communication.
Step 4: The two users are asked to compare their checksum with the one provided by the other party (i.e., “checksum comparison”), and to optionally verify the other user’s voice (i.e., “speaker verification”). They should accept the call since the two checksums match and are spoken by legitimate entities.

Figure 1: Traditional Crypto Phones Checksum Validation



Step 1: The two devices exchange the key.
Step 2: The two devices show their respective checksums as a result of SAS protocol execution.
Step 3: The two users exchange their respective checksums via verbal communication.
Step 4: The transcriber compares the checksum (i.e., “automated checksum comparison”) and the two users optionally verify the other user’s voice (i.e., “manual speaker verification”). They should accept the call since the two checksums match and are spoken by legitimate entities.

Figure 2: Proposed CCCP Checksum Validation

task is needed to defeat *data MITM*, an MITM attack over the data/IP channel.

- **Speaker Verification (optional):** Ascertaining that the voice announcing the checksum is the voice of the legitimate user at the other end of the call. This task may be needed to defeat *voice MITM*, an MITM attack over the human voice communications.

The checksum comparison task is crucial and *mandatory* given that data MITM is a prominent and commonly occurring attack in real-world [19, 30]. The speaker verification task may be optional (like in many traditional designs of Crypto Phones) since voice MITM is considered a more sophisticated attack (Crypto Phones attack hierarchy is depicted in Figure 5). Unfortunately, in practice, the *human errors* in executing the checksum verification and/or speaker verification tasks may adversely affect the security of Crypto Phones. Specifically, failure to detect *mismatching checksums* or *imitated voices* (different speakers’ or synthesized voices) would result in a compromise of Crypto Phones session communications (eavesdropping over voice communications and MITM over text communications).

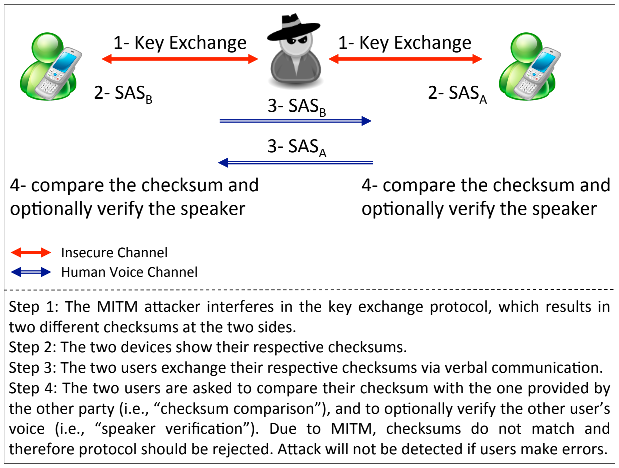
Indeed, recent research [58, 59] emphasizes such human errors demonstrating that current designs of Crypto Phones are highly vulnerable to both data and voice MITM attacks. Moreover, due to these *dual-task* human errors (in case both tasks are required), the security level provided by Crypto Phones protocol actually degrades with the use of longer checksums, contrary to the theoretical guarantees provided by the protocol (which limits the MITM attack success probability to 2^{-k} for a k-bit SAS checksum). These above tasks may also be susceptible to a *“click-through”* (or skip-through), i.e., the user just accepting without paying attention or duly performing the task, as observed in prior device pairing [49] and security warnings research [41]. Furthermore, the human errors in the benign case, i.e., rejection of matching checksums or legitimate users’ voices, adversely affect the usability of the systems since legitimate calls may often be rejected (and then need to be re-established).

In this paper, we set out to address some of these fundamental problems facing traditional Crypto Phones designs, especially focusing on threat model involving data MITM attacker. We introduce *Closed Captioning Crypto Phones* (CCCP), a novel Crypto Phones design that removes the human user from the loop of the checksum comparison task by utilizing speech *transcription*. CCCP requires the user to simply announce the checksum to the other party, and the system automatically transcribes the announced checksum and performs the comparison on behalf of the user (Figure 2). Automating the checksum comparison task in CCCP provides several key advantages over the traditional approach:

(1) The success probability of data MITM, due to human errors or click-through behavior in checksum comparison, could be highly reduced (or even eliminated).
(2) Longer checksums can be optionally utilized, which increase the underlying protocol security against data MITM.
(3) The overall checksum validation task becomes more reliable since the user only needs to perform a *single task* which reduces the cognitive burden [46, 47].

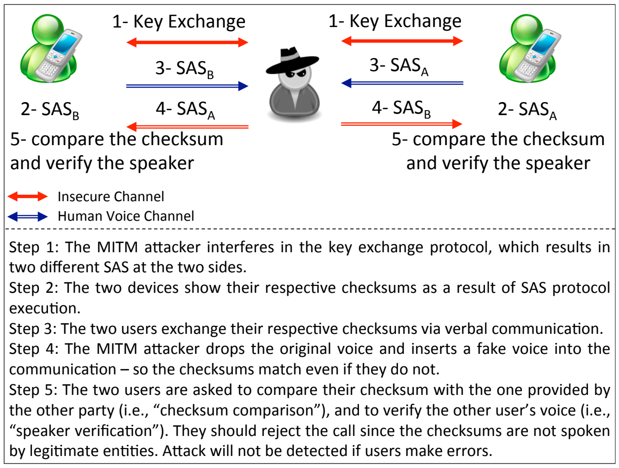
Our Contributions: We believe that our work provides the following contributions to the domain of end-to-end secure communications:

(1) ***A Novel Crypto Phones Checksum Validation Design***: We propose CCCP, a novel Crypto Phones checksum validation methodology based on a simple yet effective idea of automated speech transcription, which can be seamlessly applied to any Crypto Phones protocol and reduce the chances of MITM attacks (especially data MITM) arising from human errors or click-through in the mandatory checksum comparison task, thereby considerably improving the security and usability of the current Crypto Phones design.
Transcription is now considered a mature technology [22, 31], used reliably in many real-life domains, and is, therefore, an excellent candidate to automate the checksum comparison task in Crypto Phones without much added cost. Although transcription may not by itself be fully error-free [18, 27, 38, 55], we show how it can be carefully used to yield a robust *automated checksum comparison tool* as part of our CCCP system. We design and implement this tool based on standard transcription engines, including Google Speech API [17], Apple Mac Dictation [35] and IBM Watson Speech to Text Service.


1- Key Exchange
1- Key Exchange
2- SAS$_B$
3- SAS$_B$
2- SAS$_A$
3- SAS$_A$
4- compare the checksum and optionally verify the speaker
4- compare the checksum and optionally verify the speaker
Insecure Channel
Human Voice Channel

Step 1: The MITM attacker interferes in the key exchange protocol, which results in two different checksums at the two sides.
Step 2: The two devices show their respective checksums.
Step 3: The two users exchange their respective checksums via verbal communication.
Step 4: The two users are asked to compare their checksum with the one provided by the other party (i.e., "checksum comparison"), and to optionally verify the other user's voice (i.e., "speaker verification"). Due to MITM, checksums do not match and therefore protocol should be rejected. Attack will not be detected if users make errors.

Figure 3: Data MITM



Step 1: The MITM attacker interferes in the key exchange protocol, which results in two different SAS at the two sides.
Step 2: The two devices show their respective checksums as a result of SAS protocol execution.
Step 3: The two users exchange their respective checksums via verbal communication.
Step 4: The MITM attacker drops the original voice and inserts a fake voice into the communication – so the checksums match even if they do not.
Step 5: The two users are asked to compare their checksum with the one provided by the other party (i.e., "checksum comparison"), and to verify the other user's voice (i.e., "speaker verification"). They should reject the call since the checksums are not spoken by legitimate entities. Attack will not be detected if users make errors.

Figure 4: Voice MITM

(2) ***Comprehensive Security-Usability Evaluation via an Online User Study***: To evaluate the security and usability benefits provided by CCCP, we design a human factors online study (with $N = 66$ Amazon Mechanical Turk participants), that mimics a realistic VoIP scenario and feeds several challenges to the participants having matching and mismatching 4-word and 8-word spoken in the legitimate user's voices, different speaker's voices and automatically synthesized voices. We transcribe this comprehensive data set consisting of 1188 audio files spoken by a wide variety of speakers in real-life conditions. Our study results demonstrate that, by using our automated checksum comparison, CCCP can: (1) *drastically reduce the chances of false positives under data MITM* to 0% (leading to a security level equivalent to what is promised by the underlying cryptographic protocol), and (2) reduce the false negatives down to about 5%, much lower than traditional designs (Figure 9).
When further considering the optional, more powerful security model involving voice MITM, we find that CCCP can reduce the false positives under different speaker voice MITM attack down to around 12% and under synthesized voice MITM attack down to around 20%, which may be significantly lower than the traditional approach as shown in Figure 9.

2 BACKGROUND

2.1 Protocol and Threat Model

Many checksum-based key exchange protocol exist [44, 50, 52, 62] that Crypto Phones implementations may adopt. A checksum-based key exchange protocol is an authenticated key exchange protocol (over IP channel) which allows Alice and Bob to agree upon a key, based on checksum validation over an auxiliary channel (e.g., the human voice channel as in Crypto Phones). The protocol results in a Short Authenticated String (SAS) checksum per party, commonly encoded into words or numbers (e.g., "skydive amulet"). *Compare-Confirm* is the most popular SAS Checksum Comparison method [60]. In this method, the checksum is displayed on each party's screen, they verbally exchange their respective checksums, and both accept or reject the connection by comparing the checksums.

Copy-Confirm, is another approach in which one party reads the checksum to the other party, who types it onto his/her device and gets notified whether the checksum is correct or not.

In the security model of Crypto Phones, devices are connected via a remote, high-bandwidth bidirectional (Vo)IP channel, and are assumed to be trusted or uncompromised. An MITM adversary Mallory attacking the protocol has full control over the channel, namely, she can eavesdrop and tamper with messages transmitted.

Due to the inherent nature of the Crypto Phones key exchange protocol, matching checksums imply the successful secure association, whereas non-matching checksums imply an MITM attack. The MITM attacker's goal is to intercept or tamper with the communications; not to prevent the users from communicating (or denial of service). The protocol limits the success probability of the attack to 2^{-k} for k-bit checksums[1].

The simplest form of attack against the Crypto Phones key exchange protocol is a data-based man-in-the-middle or *data MITM* attack. The data MITM attacker acts as an MITM on the data channel and interferes with the key exchange in an attempt to establish impersonated sessions with the two parties (Figure 3). As a result of the attack, the generated checksums do not match at the two parties. However, if the users erroneously accept mismatching checksums, the data MITM attack will succeed.

Another type of attack against the Crypto Phones key exchange protocol was introduced in [58], in which the attacker can tamper with the voice channel (apart from the data channel). We refer to this attack as *voice MITM* (Figure 4). The voice MITM attack utilizes current advancement in voice synthesis/conversion [5, 21]. In this attack, after tampering with the key exchange protocol (i.e., running the data MITM attack), the attacker inserts his/her own voice (i.e., "different speaker attack"), or a morphed/converted voice of the

[1] Current implementation of Crypto Phones usually keeps the checksum short. This is because: (1) short checksums give practical level of security (i.e., 2^{-16} success probability of the attack for a 16-bit checksum), and (2) verifying long checksums is harder for the users. Some Crypto Phones use a different variation of the key exchange protocol, where the checksum is long, like a 160-bit collision-resistant hash of the public keys of two parties [36, 37]. Nevertheless, the main functionality remains the same in terms of the human tasks.

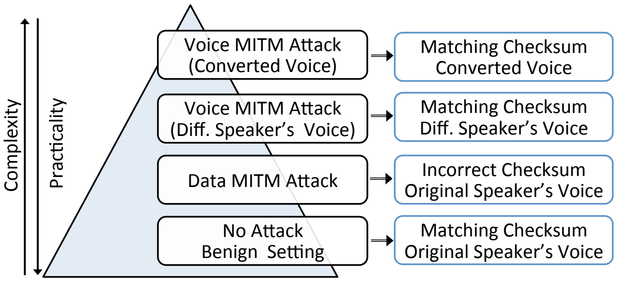

Complexity
Practicality
Voice MITM Attack (Converted Voice)
Matching Checksum Converted Voice
Voice MITM Attack (Diff. Speaker's Voice)
Matching Checksum Diff. Speaker's Voice
Data MITM Attack
Incorrect Checksum Original Speaker's Voice
No Attack Benign Setting
Matching Checksum Original Speaker's Voice

Figure 5: Crypto Phones attacks ordered by complexity/practicality

other user (i.e., "voice conversion attack") into the communication, attempting to fool the user into accepting the speaker as valid. In the different speaker attack, the adversary does not perform any voice synthesis, while in the voice conversion attack the adversary transforms his voice into the victim's voice based on some prior recordings of the victim's voice.

Data MITM is the most straight-forward and hence a common form of attack in practice, as Figure 5 shows. Compared to data MITM attack, voice MITM attack is more complex to establish. Firstly, it involves manipulation of both voice and data channels. Secondly, it imposes a delay to look up the checksum in the audio stream, to drop the legitimate checksum, and to insert an imitated checksum [58]. Lastly, in the case of the voice conversion attack, it requires training of the voice conversion tool based on previously collected audio samples spoken by the victims. Collecting the samples in victim's voice may not always be easy or possible. Clearly, a different speaker attack is simpler than a voice conversion attack since the attacker does not need to collect voice samples and train the voice converter, but can easily use his/her voice [2].

Given the hierarchy of the threat model, as a *mandatory* goal, real-world Crypto Phones implementations must attempt to make data MITM attacks as difficult (or infeasible) as possible. *Optionally*, it may attempt to resist voice MITM attacks. Indeed, most current Crypto Phones only ask the users to perform the Checksum Comparison task to detect data MITM attacks [34, 64], and do not explicitly ask the users to perform the Speaker Verification task to detect voice MITM attacks. Among this class of attacks, different speaker attack should be considered a more practical threat followed by conversion attack, which is the most powerful attack. This same *tiered* threat model is what we consider in this paper.

2.2 Limitations of Related Work

Problem 1—Susceptibility to Human Errors and MITM Attacks: Crypto Phones checksum validation protocol involves the essential task of Checksum Comparison (to defeat data MITM). However, it has been shown that the errors committed by human users in comparing the checksums lead to false acceptance of an MITM attack session or false rejection of a valid checksum [49, 57, 59]. This is a serious vulnerability of the current Crypto Phones that CCCP aims to primarily address.

Crypto Phones checksum validation also involves the second optional task of Speaker Verification (to defeat voice MITM). However, manual speech perception and recognition is a complex task [54]. Therefore, Speaker Verification is challenging even in benign settings. On top of that, voice conversion and reordering attacks are possible against Crypto Phones, which make Speaker Verification even harder [58].

The results of prior studies show that current designs of Crypto Phones offer a *weak level of security* (significantly weaker than that guaranteed by the underlying protocols), and their usability is low. Quantitatively, the overall average likelihood of users failing to detect an attack session is about 25-50%, while the average likelihood of accepting a legitimate session is about 75% [58, 59]. These drawbacks with the currently deployed approach in Crypto Phones provide a sound motivation to investigate other checksum validation models.

Problem 2—Security Degradation with Increase in Checksum Size: Checksum size is a crucial security parameter for Crypto Phones. Theoretically, the security of Crypto Phones should increase exponentially in presence of a data MITM attacker with increase in the checksum size.

However, [59] shows that increasing the checksum size makes the Checksum Comparison task more difficult for human users, eventually decreasing the usability and the security of the system. Based on this prior study, while the theory guarantees that increasing the checksum size, from 2-word to 4-word increases the security exponentially, by a factor of 65536 (2^{16}), the attacker success probability increased (from about 30% to 40%). This situation emerges because, as the checksums became longer, Checksum Comparison became much harder.

In this light, there is a need to design new validation models, which preserve the increase in system security with increase in checksum size, to be consistent with the theoretical bounds of the protocols.

3 OUR APPROACH: CLOSED CAPTION CRYPTO PHONES

We introduce a novel Crypto Phone checksum validation design, with the goal of making the Checksum Comparison and Speaker Verification tasks highly robust (significantly more robust compared to the traditional design). The introduced CCCP model (Figure 2) is built using the speech transcription technology, and carefully leverages the strengths of both humans and machines.

Transcription Primer: Automated Checksum Comparison in our suggested schemes is based on a Speech to Text (STT) tool. STT, which we also refer to as transcriber, takes the voice waveform as input and recognizes it based on the best matching combination of words. STT tools use machine learning techniques to incorporate information about grammar and language structure to generate a transcription. First, it gets a feature vector of each word and then uses models to match this feature vector with the most probable feature vector in the model. Transcription is a fairly mature technology with extensive applications in various domains involving human

[2] In practice, it is often assumed that the voice MITM attack is very difficult to perform [34, 64], and therefore, traditional Crypto Phones usually do not explicitly ask the user to perform the task of Speaker Verification.

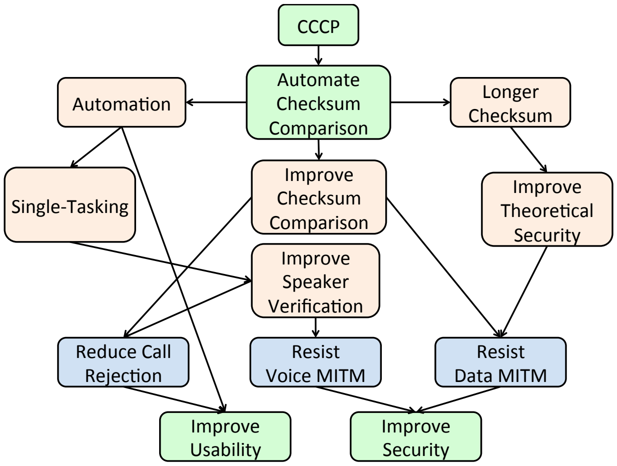


Figure 6: Path for improved security and usability in CCCP

speech, including closed captioning of videos [10], journalism, and medical transcription [23].

There are several open-source transcription tools available for different platforms. These tools are designed for general use, hence they incorporate the language model and optionally the speaker model to improve the accuracy of the transcription. They also benefit from signal processing algorithms, neural networks, deep learning, and big data to provide high accuracy. There are built-in Speech to Text tools for dictation, voice commands and accessibility on smartphones (e.g., Siri on iPhone [8] and "Ok Google" command engine on Android phones [32]). Other apps, such as Nuance Dragon Mobile Assistant [13], are also available and are gaining popularity. These tools and apps are built on top of powerful Speech to Text APIs, such as iOS Speech Recognition API [7], android.speech [6], and Dragon NaturallySpeaking [14]. Other systems such as IBM Watson Speech to Text [29] and Google Cloud Speech API [17] are available for cloud and web platforms.

Increasing the Robustness of Checksum Comparison through Transcription: Our key idea in CCCP is to automate the process of Checksum Comparison by using the automated human speech transcription technology. We propose to place automated checksum comparison tool in the Crypto Phones, which receives the audio checksum from one end (Alice's checksum referred to as SAS_A in Figure 2), and transcribes it at the other end (Bob) followed by comparing the transcribed output with the local checksum (Bob's checksum referred to as SAS_B in Figure 2). Alice only needs to verbally announce the checksum (like in current Crypto Phones) but does not need to compare the checksum (unlike current Crypto Phones). To initiate the transcription process, similar to speech recognition systems, CCCP can spot a specific keyword (e.g., "Go Secure"). Alternatively, tapping a "checksum matching" button embedded within the UI could trigger the transcriber.

An indirect advantage of using the transcription technology is the capability to use long checksums since Checksum Comparison is being performed by a machine, not a human. To recall, the longer the checksum, the better the theoretical security offered by the checksum validation protocol against MITM attacks. That is, we can push towards achieving a nearly negligible probability of success for the data MITM attacker.

In addition to enhancing security against the data MITM attack, CCCP promises to improve usability by taking the human user out of the loop of the Checksum Comparison task, except for verbally announcing the checksum, and thus, reduces the chances of false negatives (i.e., disconnecting valid calls).

As part of CCCP, we build a Checksum Comparison tool suitable for our purpose, based on the off-the-shelf transcription systems. The current transcription technology [42] is known to be robust and is in wide use as discussed above (primer), and thus we hope to have excellent accuracy in the Checksum Comparison task based on the transcription systems, thereby offering a high level of resistance to data MITM attack.

Optionally Increasing the Robustness of Speaker Verification through Single-Tasking: To optionally resist against voice MITM, similar to current Crypto Phones, CCCP relies on the human user to verify the speaker and judge if the received checksum is spoken by the original speaker. In particular, the user should decide if the voice that speaks the checksum belongs to the person he/she is calling, either based on pre-familiarity with the speaker, or, if the speaker is not already familiar, by assuring that the person who speaks the checksum is the one who takes part in the rest of the conversation [34].

However, there is a crucial difference between traditional Crypto Phones and CCCP in Speaker Verification. Relieving the user from the task of Checksum Comparison (through transcription as described above), may improve the overall performance of the user since the user is now only involved in a "single task" of Speaker Verification. Therefore, by automating the Checksum Comparison, we may also improve the performance of Speaker Verification under benign and voice MITM attack settings. In contrast, current Crypto Phones require the user to "multi-task", which could be detrimental to users' performance [46–48, 51].

Our hypothesis is that in CCCP, original and different speakers will be reasonably well-recognized, and even the converted voice samples will be fairly recognized.

Summary of Projected Advantages of CCCP: Based on the above discussion, CCCP could significantly improve the security and usability of the traditional design. Figure 6 illustrates how CCCP strives to increase security and usability of Crypto Phones, as a direct or indirect result of automating Checksum Comparison, the use of longer checksums, and single-tasking. We summarize the advantages of our scheme (CCCP) compared to the current scheme (traditional Crypto Phones) in Table 1.

4 CCCP EVALUATION STUDY DESIGN

4.1 Objectives

Our study is designed to measure the security and usability of CCCP, based on the threat model depicted in Figure 5. The goals of the studies are outlined below.

(1) **Robustness against Data and Voice MITM Attacks:** For security assessment against the data MITM attack, we are interested in determining how often the transcriber accepts a

Table 1: The projected usability and security properties of CCCP contrasted with the traditional design. The highlighted cells represent the key security and usability improvements offered by CCCP over the traditional designs under the required task of Checksum Comparison.

		SECURITY			USABILITY	
	Checksum Size	Data MITM	Voice MITM (Optional)		Matching Checksums	Original Speaker (Optional)
			Different Voice	Converted Voice		
Traditional Design	Short	Poor	Poor	Poor	Poor	Poor
CCCP	Short (and Long)	Excellent	Good	Fair	Very Good	Good

mismatching checksum due to potential transcription errors. *False Positive Rate* of Checksum Comparison (FPR$_{cc}$) denotes the probability of accepting such instances.

Moreover, for security assessment against the voice MITM attack, we are interested in determining how often the user fails to detect a different speaker's voice or a converted voice. *False Positive Rate* of Speaker Verification (FPR$_{sv}$) denotes the probability of accepting such attacked instances.

(2) **Accuracy in Benign Settings:** For usability assessment, we are interested in finding how often the system rejects matching checksums spoken in an original speaker's voice. *False Negative Rate* of Checksum Comparison (FNR$_{cc}$) represents the probability of rejecting a valid checksum by the transcriber (due to failure of the transcriber, and potential failure of users in correctly speaking the checksum).

In addition, as an optional task in CCCP, we would like to find out how often the users reject the caller announcing the matching checksum spoken in the original speaker's voice. *False Negative Rate* of Speaker Verification (FNR$_{sv}$) represents the probability of the listener rejecting a valid speaker.

(3) **Efficiency:** The delays incurred in performing the Checksum Comparison and/or Speaker Verification tasks, referred to as completion time, may impact the overall usability of the system. This delay might arise due to: (1) users speaking the checksum (referred to as "Duration of Checksum"), (2) users' delay in verifying the speaker (referred to as "T$_{sv}$"), and (3) users requesting the other party to repeat the checksum (referred to as "Replay Rate" or "RR"). The delay may prolong the process and establishment of the phone call. Perhaps less significant is the time taken by the transcriber (referred to as "T$_{cc}$" in our study).

(4) **Comparison with Traditional Crypto Phones**: As a baseline for our study, we intend to compare the performance and accuracy of CCCP with traditional 2-word and 4-word Crypto Phones.

4.2 System Setup

To show the feasibility of our CCCP model in practice, and to support the security and usability study, we developed an application for web-based clients to make web-based VoIP to our system. Next, we describe the main components of this setup in more detail as listed in Figure 7.

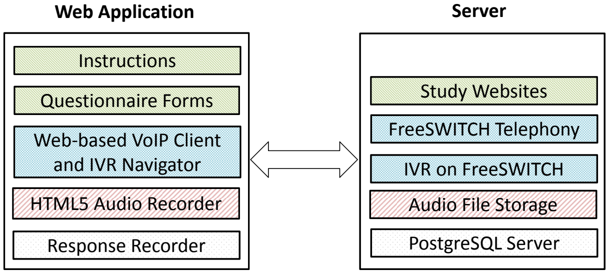


Figure 7: The study implementation and setup

Web-based Interface: The web-based interface was developed in PHP, JavaScript, and HTML5 and was the entry point for the participants in the study. It consisted of web-based WebRTC (Web-based Real-time Communications) voice client, and a database client to connect to the database server to read questions and store participants' responses. It also included a web-based audio recorder to record the voices of the participants when they spoke the checksum. The web-server was hosted on an Amazon Elastic Compute Cloud (Amazon EC2) t1.micro instance with up to 3.3Ghz CPU, 1GB of Memory, and Debian 8 Jessie operating system.

Web-based Voice Application: We set up a softswitch server on the same EC2 machine as the webserver. We installed and configured FreeSWITCH 1.6.7 as the softswitch [16]. We configured the security group (firewall) on the EC2 instance to accept web and voice communication protocols (HTTP, HTTPS, WS, WSS, SIP, and RTP). The open source FreeSWITCH software supports VoIP protocols including Session Initiation Protocol (SIP), IVR, and WebRTC that are essential components to connect the web-based clients to the switch.

We designed and implemented the web-based VoIP client based on SIP session initiation protocol, and WebRTC transport protocol. We used the SIP server on our cloud-based telephony platform to initiate the session. The web-based voice client uses sipML5 open source HTML5 SIP client API [39], and supports Dual Tone Multi-Frequency signaling (DTMF).

In our study (described in Section 4.3), we configured the IVR system on FreeSWITCH system to play the instructions, voice recordings of speakers and checksum challenges (from the prerecorded audio files of the original speaker, different speaker and converted voice), based on the DTMF signals it receives from the web- based application (i.e., clicking buttons on the web-page is

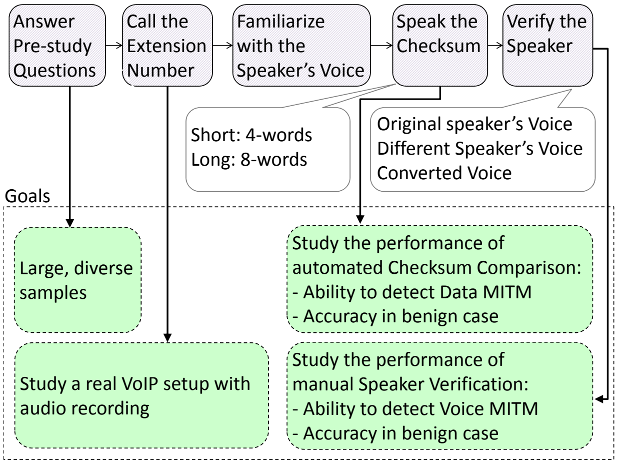

Answer Pre-study Questions
Call the Extension Number
Familiarize with the Speaker's Voice
Speak the Checksum
Verify the Speaker
Short: 4-words
Long: 8-words
Original speaker's Voice
Different Speaker's Voice
Converted Voice
Goals
Large, diverse samples
Study the performance of automated Checksum Comparison:
- Ability to detect Data MITM
- Accuracy in benign case
Study a real VoIP setup with audio recording
Study the performance of manual Speaker Verification:
- Ability to detect Voice MITM
- Accuracy in benign case

Figure 8: The protocol flow and methodology of our CCCP human factors study

translated to DTMF and is sent to the softswitch). To randomize the ordering of the displayed and played checksums, we generated a 18×18 Latin Square[3] using which we set the IVR to play the audio samples. The same ordering was used to display the checksum that the participant should speak.

Response Database: We set up a PostgreSQL database on our Amazon EC2 server to store the answers that participants provide to the demographic questionnaire, and to the Speaker Verification .

Audio Storage: We recorded the audio collected from the participants in the same EC2 server.

Voice Dataset We picked one male speaker from the CMU_ARCTIC US English dataset [11] as the original speaker of the study (victim of the attack). As a different speaker (simple voice MITM attacker), we picked another male speaker from the same dataset. We converted voice of the different speaker (attacker) to the voice of the original speaker (victim) using the Festvox voice transformation tool [5]. This type of voice synthesis was used in [58, 59] to perform the conversion attack against traditional Crypto Phones. We used 100 sentences spoken by the victim and the attackers to train the voice conversion system.

4.3 Study Protocol

The study design and protocol flow are shown in Figure 8. The study is in line with [59]–the participants are asked to speak the checksum and verify the speaker, but, unlike [59], they do not compare the checksum.

In the pre-study step, we asked the participants to follow a link to our web-based VoIP application. The participants were first asked to fill out a demographic questionnaire. These questions poll for each participant's age, gender and education. An additional question was asked for participants' familiarity with VoIP applications. Also, they were asked if their first language is English, and whether they suffer from any hearing impairments (relevant to the task of Speaker Verification). Then the participants were given instruction on how to perform the main study tasks, i.e., how to establish a call, speak the checksum, and verify the speaker.

In the main study phase, *first*, participants were asked to make a web-based VoIP call to our soft-switch through the web-based application. Once they get connected to the telephony system, they could listen to the instructions through the IVR and read the displayed instructions about how to proceed in each step of the study. *Second*, the voice of a speaker was played for 2 minutes and participants were instructed to get familiar with the speaker's voice[4]. *Third*, participants were asked to speak a displayed checksums picked from CMU_ARCTIC sentences (listed in Section A.1) ("Checksum Speaking" task), and *fourth* to verify a speaker who speaks the same checksum (Speaker Verification task). The manual Speaker Verification task was performed online and the automated Checksum Comparison task was performed offline.

In the Checksum Speaking task, participants were asked to speak the displayed checksum. The spoken audio was recorded and uploaded to our system for offline transcription and analysis. In the Speaker Verification task, they were asked to listen to a voice that speaks the checksum and verify if the voice is the one that they originally got familiar with. In this part of the experiment, the samples of the original speaker's voice, the different speaker's voice, and the converted voice, were played randomly one at a time.

The total number of challenges presented in the main study phase consisted of 18 samples. The samples included: 9 samples of short-size checksums (4 words) and 9 samples of long-size checksums (8 words). An equal number of samples of the original speaker, different speaker and converted voices (9 samples each) were played.

We published a Human Intelligence Task (HIT) on Amazon Mechanical Turk (MTurk) and recruited 66 subjects. The study was approved by our university's IRB. Participants who completed the HIT were compensated $1.25 each. The average duration of the experiment was around 20 minutes. We chose the incentive based on similar MTurk HITs (e.g., [59]). Appendix A.3 shows a screenshot of several other similar studies.

4.4 Statistical Analysis Methodology

All results of statistical significance of our data analysis pursued in the next section are reported at a 95% confidence level. The non-parametric Friedman test is used to test for the existence of statistical differences within the groups, and, if it succeeded, Wilcoxon Singed-Rank test is used to examine in which pairs the differences occurred. The statistically significant pairwise comparisons are reported with Bonferroni corrections.

5 ANALYSIS AND RESULTS

5.1 Demographic Information

There were 55.8% males and 44.2% females among the 66 participants in our study. Most of the participants were between 18 and 44 years old (33.8% 18-24 years, 32.5% 25-34 years, 23.1% 35-44

[3] Latin square is an n × n array filled with n different symbols, each occurring exactly once in each row and exactly once in each column.

[4] In line with [58, 59], our study was designed to test a scenario involving a unfamiliar speaker's voice. Current Crypto Phones also ask the users to verify that the voice speaking the checksum matches the voice used in the rest of the conversation for an unfamiliar speaker.

years, 8.0% 45-54 years, and the rest 55-64 years). 26.1% of participants were high school graduates, 32.7% had a college degree, 33.4% had Bachelor's degree and 7.9% had Master's degree. This analysis shows that participants represent a diverse population by gender, age, and education. More than 96.7% of the participants declared that they did not have any hearing impairment[5].

5.2 Design of Checksum Comparison with Standard Transcribers

Using the collected data, we ran a basic analysis of several speech to text tools including Google Speech API [17], Apple Mac Dictation [35], IBM Watson Speech to Text Service, and CMUSphinx [12]. We transcribed 100 audio samples checksums (about 800 words) using each of the tools.

For CMUSphinx, we used the Sphinx4 transcriber demo developed in Java, which showed very high latency. Therefore, we discarded it from the study. To test Google Speech, we played back the audio files using an iPhone 7 and used the dictation add-on on Google Docs to transcribe the audio. Similarly to test Apple Mac dictation, we enabled the Dictation and Speech on a MacBook laptop, played back the audio files on the phone and transcribed the audio. To test IBM Watson tool, we set up the environment on Watson Developer Cloud and developed a Java application that executes curl requests to the transcriber.

The average Word Error Rate (WER) from this pilot experiment was 12% for Google Speech API, 10% for Apple Mac Dictation, and 10% for IBM Watson service. This preliminary evidence demonstrates the strong capability of the existing transcription tools applied to SAS checksum transcription, which can yield to robust Crypto Phones systems.

We selected the IBM Watson Speech to Text service for the rest of our analysis, due to its high accuracy and simplicity of development. We developed a Java application that uses IBM Watson Speech to Text service to build an automated comparison tool. After receiving the result of transcription, we processed the output (generated in JSON format) to compare the transcribed audio against the checksum. We stored the result of the analysis in the database.

Although IBM Watson Speech to Text is a cloud-based tool, we are *not suggesting a cloud-based service*. We note that our idea is *not limited* to this tool, considering the high performance of many transcribers (built on top of large training models). Other on-board tools with similar accuracies, such as Nuance [23], can be integrated into the apps.

5.3 Resistance to Attacks

Robustness against Data MITM: We first analyze the instances of accepting mismatching checksums (FPR$_{cc}$), which leads to the success of a data MITM attack as shown in column 3 of Table 2. Given the frequency and simplicity of data MITM (as per Figure 5), FPR$_{cc}$ is the most important parameter in CCCP.

To measure FPR$_{cc}$, we compared the incorrectly transcribed words against all the words in our checksum dictionary. The results show that regardless of checksum size, FPR$_{cc}$ for all samples is 0%. Knowing that none of the incorrectly transcribed sequences matched *any possible checksum* shows that it is highly unlikely or impossible for the transcriber to decode an attacked checksum to the valid checksum for a particular session. This result is the most encouraging outcome of our CCCP solution that proves how automated Checksum Comparison can eliminate the possibility of data MITM attacks.

Table 2: Analysis of the data MITM attack. FNR$_r$ (the actual FNR of the system) shows the error when at most half of the words in the checksum are transcribed incorrectly. FPR$_{cc}$ remains the same with or without the relaxed mode.

Checksum Size	Checksum Duration	FPR$_{cc}$	FNR$_r$	FNR$_{cc}$
4-word	5.79s (1.81)	0%	4.38%	24.57%
8-word	10.28s (2.87)	0%	7.44%	63.17%

Table 3: Analysis of Speaker Verification for the Different Speaker attack

Checksum Size	Checksum Duration	FPR$_{sv}$	T$_{sv}$	RR
4-word	2.72s	6.56%	4.33s	5.56%
8-word	6.07s	12.13%	5.54s	4.04%

Table 4: Analysis of Speaker Verification for the Voice Conversion attack

Checksum Size	Checksum Duration	FPR$_{sv}$	T$_{sv}$	RR
4-word	3.85s	18.43%	5.27s	6.57%
8-word	5.50s	20.45%	5.91s	1.51%

Robustness against Voice MITM: Our study shows that FPR$_{sv}$ for the different speaker attack is at most 12%, which suggests that CCCP can resist this attack in a large majority of cases. The result is shown in column 3 of Table 3.

For the voice conversion attack, our study shows a higher FPR$_{sv}$ (about 20%) compared to different speaker's voice (participants were less successful in detecting the attack since the voice is now more similar to the victim's voice). The result is shown in column 3 of Table 4.

Again, since FPR under data MITM attack is 0%, FPR$_{sv}$ in Table 4 captures the rate with which the voice conversion attack will succeed. These results suggest that even the sophisticated voice conversion attack can be detected reasonably well for practical purposes. We recall that this attack is not easy to launch due to the delays introduced in the voice channel and the need to collect prior voice samples [58].

5.4 Accuracy in Benign Setting

Accuracy of Checksum Comparison in Benign Case: The errors that happened in transcribing might be the result of incorrect pronunciation by the human user or incorrect transcription by the transcriber tool. The collected audio samples in our experiment were over 1100 files (over 200 minutes) and we could not manually

[5] Given a small fraction of potential hearing impaired users, we report our analysis results in aggregate corresponding to all participants.

verify if the participants had spoken all the words accurately. However, in a random selection of 50 audio files, we did not notice any incorrect pronunciation of the words. Therefore, we assume that the reported error rate is related to incorrect automated transcription and not incorrect pronunciation by the users.

We analyze FNR$_{cc}$ based on the number of incorrect words in each checksum. FNR$_{cc}$ is 24.57% and 63.17% for 4-word and 8-word checksum, respectively, when *at least one word* is incorrectly transcribed. Although these results by themselves may seem high, we show how we can effectively improve them by orders of magnitude.

The higher error rate in the 8-word checksum compared to the 4-word checksum is not surprising. As one may expect, there is a higher probability of generating a single-word error in an 8-word sentence compared to a 4-word sentence. In theory, if the probability of incorrectly transcribing at least one word in a 4-word checksum is p_1, such probability increases to $2p_1 - p_1^2$ in a 8-word checksum.

Wilcoxon signed-rank test conducted using alpha levels of 0.05. and showed statistical significance ($p = 0.008$) for the comparison between FNR$_{cc}$ of 4-word and FNR$_{cc}$ of 8-word checksum. This analysis confirms that FNR$_{cc}$ significantly increases when the checksum becomes longer.

Relaxing Automated Checksum Comparison to Significantly Reduce Rejection Rate: Since FNR$_{cc}$ leads to the rejection of a benign call (which may degrade the usability of the system), our goal is to reduce this error. To decrease the error rate, we propose to relax the assumption of accepting the checksums. For example, we suggest accepting the checksum even if at least half of the words in the checksum are incorrectly transcribed (such as for a 4-word checksum, the transcriber accepts it even if one or two words are transcribed incorrectly).

FNR$_r$ in Table 2 (to be read as FNR of Checksum Comparison in "r"elaxed mode) shows the result of such relaxation. Using this approach, FNR$_{cc}$ significantly reduces from around 25% and over, to around 5%. With this approach the usability of the system may increase since rejecting the calls due to the incorrect matching of the valid checksum will be less frequent. Figure A.1a and A.1b in Appendix A.3, further show the effect of "number of tolerated incorrect words" on FNR$_r$.

We did not find a statistically significant difference in FNR$_r$ between the two checksum sizes, which implies that longer checksums may not change the usability of the system induced by an unwanted rejection of calls.

Relaxing the Automated Checksum Comparison has an impact on the theoretical security of CCCP. Using this approach, the security of a k-word checksum (2^{-k}) is reduced to the security provided by a $k/2$-word checksum ($2^{-k/2}$). For example, in our study, the security offered by a 4-word checksum reduces to that of a 2-word checksum, similarly, 8-word to 4-word. Although, the security of CCCP will be reduced in the "relaxed" mode, still if more than 4-word checksum is incorporated, CCCP can offer significantly higher security compared to traditional Crypto Phones with 2-word checksum as argued next.

Since FPR$_{cc}$ is essentially 0% in CCCP, the security provided by CCCP is close to the theoretical security promised by the underlying protocol. Therefore, in a relaxed mode (if transcription error in up to half of the checksum is accepted), a 4-word checksum offers a security level close to the security of 2-word checksum, that is 2^{-16}. However, in the traditional Crypto Phones although the security provided by a 2-word checksum is expected to be 2^{-16}, due to human errors in Checksum Comparison the security degrades to around 30% [59]. In practice, we can pick any number of tolerated incorrect words to optimize accuracy and security.

Table 5: Analysis of Speaker Verification for the original speaker

Checksum Size	Checksum Duration	FNR$_{sv}$	T$_{sv}$	RR
4-word	3.66s	25.76%	5.71s	10.61%
8-word	6.79s	21.72%	5.96s	2.02%

Accuracy of Speaker Verification in Benign Case: We are also interested in investigating FNR$_{sv}$, which represents the instances where the user mistakenly rejects the CCCP call due to the failure in recognizing a legitimate user's voice. Our study shows that FNR for the 4-word and 8-wordchecksums were 25.76% and 21.72% respectively as shown in column 3 of Table 5. In analyzing this higher error rate we should recall that the voice MITM is a less plausible attack (especially with the converted voice).

5.5 Efficiency

SAS Length: Column 2 of Table 2 shows the average duration of speaking the checksum. As expected, the duration of the checksum increases as the number of words increases. Wilcoxon signed-rank test showed statistical significance when comparing the 4-word and 8-word ($p = 0$).

Time Taken by Speaker Verification (T$_{sv}$) and Replay Rate: The average duration of making a decision to accept or reject the different speaker, conversion attack, and original speakers voice are shown under T$_{sv}$ in column 4 of Table 3, 4, and 5, respectively. We did not find any statistically significant difference in T$_{sv}$ between 4-word and 8-word checksums in any of the attack and benign settings.

Note that the average T$_{sv}$ in some cases is less than the average duration of the samples, which shows that users did not fully listen to the samples before accepting or rejecting the call. However, in all cases, T$_{sv}$ increases by the increase in the duration of the sample. We will discuss how we should incorporate this result in designing checksums in Section 6.

The results of Replay Rate (RR) is shown in column 5 of Table 3, 4, and 5. Replay Rate is around 5% when averaged over all instances of attack and benign cases. The maximum Replay Rate is around 10% in the benign case for 4-word checksum. It seems that the participants did not frequently replay the samples more than once before deciding to accept or reject them. We did not find statistically significant difference in RR between the two checksum sizes in any of the attack and benign settings.

Time Taken for Checksum Comparison (T$_{cc}$): Since in our study, Checksum Comparison analysis is performed offline, the efficiency of the transcriber (T$_{cc}$) does not play a major role in the imposed delay. However, for the sake of completeness, we report

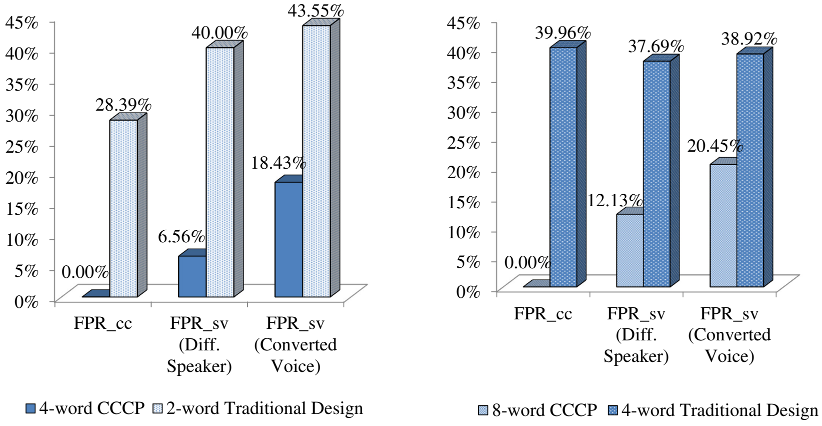


Figure 9: Comparison of FPR$_{cc}$ and FPR$_{sv}$ between CCCP and traditional design

T$_{cc}$ analysis here. Our analysis shows that IBM Watson Speech to Text tool can operate at near real-time speed. Using our dataset, T$_{cc}$ is 59 words per minute, which seems to be efficient for all practical checksum sizes. Therefore, the transcriber can run in parallel with receiving the checksum (i.e., while the user speaks the checksum). This performance analysis was performed on a MacBook Air with 1.3GHz Intel Core i5 processor with 4GB of DDR3 Memory, over a 300Mbps Internet link. There are faster Speech to Text tools that have reported the speed of 450 words per minute [9], which may be utilized in future real-life implementations of CCCP.

5.6 CCCP vs. Traditional Designs

Although our study was not designed to directly compare CCCP with the traditional designs, we summarize an indirect quantitative comparison and illustrate it in Figure 9. The MTurk participants in the study of [59] had demographic characteristics similar to the participants in our study, which allows for us to compare the results between the two studies meaningfully. Therefore, this represents a relatively fair, between-subjects comparison with a study involving a similar design/dataset and similar set of participants. In this comparison, we consider CCCP in the relaxed mode (i.e., when at most half of the words in the checksum are transcribed incorrectly), and therefore compare our 4-word and 8-word results against 2-word and 4-word results of [59], respectively. This is a fair comparison at the equivalent level of security offered by CCCP relaxed mode versus the traditional design tested in [59].

Mandatory Threat Model: Data MITM: The most appealing contribution of our work is that CCCP eliminates the chances of data MITM — the most commonly occurring attack in practice ([19, 30, 56]) (Figure 9). With respect to Checksum Comparison, our study shows that if the only applicable attack is data MITM, then the FPR for CCCP is close to 0%[6]. In contrast, the results of [59] show that in the traditional Crypto Phones, with manual Checksum Comparison, due to human errors, the average FPR of accepting a mismatching checksum is on average around 28% and 39% for 2- and 4-word checksums, respectively. Our automated Checksum Comparison basically eliminates this error.

Also with respect to the average FNR, [59] reports the error rates of about 22% and 25% for 2- and 4-word checksums, respectively, while CCCP reduces the error rates to around 4% and 7% in the relaxed mode for the 4- and 8-word checksums. Therefore, CCCP not only drastically reduces FPR, but also eliminates the click-through effect by automating Checksum Comparison. We believe these improvements constitute an important breakthrough in Crypto Phones' security. Moreover, CCCP's time duration is low (about 5s).

Optional Threat Model: Voice MITM: Even considering more sophisticated, i.e., less common, voice MITM attacks, CCCP provides a significant security improvement (Figure 9), which we attribute to CCCP's "single tasking" feature, since, in our study of CCCP, users are only involved in one task of Speaker Verification, whereas in [59], participants had to perform both Speaker Verification and Checksum Comparison. The Speaker Verification task in [59] shows FPR$_{sv}$ of about 40% for 4-word checksum for the different speaker attack (compared to 6% in our study) and 43% for voice conversion attack (compared to 18% in our study). For the 8-word checksum, the Speaker Verification task in [59] shows FPR$_{sv}$ of about 38% for the different speaker attack (compared to 12% in our study) and 39% for voice conversion attack (compared to 20% in our study).

In terms of usability, although our FNR$_{sv}$ are in line with those of traditional designs (21-25%), we believe that single tasking would make the CCCP system more usable by reducing the cognitive burden on the users, which might also prevent them from clicking through speaker verifications in practice. Therefore, with respect to Speaker Verification, our study also shows that users generally performed better in detecting the voice MITM attack in CCCP compared to the reports of [59].

6 SUMMARY AND KEY INSIGHTS

(1) Defeating Data MITM Attacks due to Human Errors and Click-Through: By automating the Checksum Comparison task, we effectively reduced FPR$_{cc}$ to 0%. This result implies that CCCP drastically improves security against data MITM attack, compared

[6] Precisely, FPR would be no more than the success rate of the random attack against the protocol, that is, 2^{-k}, for a k-bit checksum

to traditional Crypto Phones. This is a notable result, considering that data MITM attack is the most dominant and practical form of attack against Crypto Phones.

In current Crypto Phones and even CCCP, user's primary task is to have a conversation, while the establishment of the secure channel is their secondary task. Therefore, users may skip the security task or accept a mismatching checksum without fully verifying it to proceed with the call. This click-through or rushing user behavior has also been reported in the context of localized device pairing schemes [49], security warnings [41] and end-to-end encrypted messaging apps [57].

CCCP seems naturally more robust to such a click-through behavior in detecting data MITM attacks compared to traditional designs, due to the automation of Checksum Comparison. Since the FPR_{cc} for existing designs is high, they can not be sure when the users are under attack and therefore can not reliably inform the users of the presence of the attack. In contrast, since the FPR_{cc} for CCCP is 0%, the Crypto Phones built on the CCCP model can very reliably alert the users about an ongoing attack, therefore, the users can make an informed decision to drop the call. The app can also optionally drop the call if the automated Checksum Comparison detects an ongoing data MITM attack. Automatically dropping the calls may have some usability price as CCCP's FNR_{cc} is around 5% and therefore around 5% of the valid calls may have to be rejected. However, it is important to notice that CCCP's FNR_{cc} is still much lower compared to that of the current Crypto Phones designs (around 25%) (as reported in Section 5.6), which means that current designs can not at all afford to automatically drop the calls under the suspicion of the attack.

(2) Other Significant Benefits: Automating the Checksum Comparison task offers two additional/optional benefits that further improves security indirectly:

***2a) Longer Checksums*:** Automating the Checksum Comparison task facilitated the use of longer checksums and as a result increases the security of the system significantly. For example, the probability of the attack success in CCCP (relaxed mode) reduces from 2^{-16} for a 4-word (32 bit) checksum to 2^{-32} for a 8-word (64 bit) checksum.

***2b) Single-Tasking*:** By automating Checksum Comparison in CCCP, users are only involved in one task of Speaker Verification, while in current Crypto Phones models, users are involved in two tasks of Speaker Verification and Checksum Comparison. There are several research work, which studied and argued that multitasking reduces efficiency and performance since the brain can focus on one task at a time (e.g., [46–48, 51]). Therefore, automating Checksum Comparison might have an implicit effect on improving the manual Speaker Verification task by reducing the number of the simultaneous tasks that users need to perform at a given time.

(3) Efficiency: Longer checksums improve the security of CCCP. However, longer checksums increase the time taken to speak the checksum and the time taken to reject or accept the speaker's voice (T_{sv}). This may impact the usability of the system. For example, our study showed that on average it takes about 10s to speak the 8-word checksum and users took on average about 6s to accept or reject the speaker's voice. Assuming that the transcriber works in real-time, and the transcription and Speaker Verification runs in parallel, in practice, the checksum validation would take about 10s to complete. For a 4-word checksum, this time is around 6s.

The analysis also shows that the users do not wait for the whole checksum to be spoken before accepting or rejecting the voice. Hence, it seems the dominant delay in CCCP checksum validation is in speaking the checksum.

We infer that while the size of the checksum in bits increases the theoretical security, the duration of the checksum in seconds may affect usability. Since our study shows that users take no more than 6s for Speaker Verification (regardless of the checksum size and duration), a practical CCCP may design a checksum dictionary to incorporate this result to limit any n-word checksum (e.g., 8 words to provide 2^{-32} theoretical security against MITM) to a certain duration (e.g., 6s, since users take no more than 6s to verify the speaker).

7 DISCUSSION AND FUTURE DIRECTIONS

7.1 Defeating Voice Reordering Attacks

In Copy-Confirm checksum validation, where the users are asked to copy the checksum spoken by the other party into their device (and let the device compare it), typically numerical checksums are used. This is because the use of phrases is not practical due to the limitation of users in typing the (random) words. However, numerical checksum is highly susceptible to *reordering attacks* [58]. Since in CCCP, Checksum Comparison is now automated, several types of checksums become possible (e.g., words from a large, dynamic dictionary) making reordering attacks practically impossible.

7.2 Study Strengths and Limitations

In our human factors study designed to evaluate CCCP, we collected data set from a large and diverse sample of users, operating from their own computers (potentially with diverse hardware and software, including microphones to record the spoken samples). Our collected data included over 1100 audio samples, and responses to around 400 questions in each category of checksum challenges (attacked and benign setting for 4- and 8-word). This study was supported by our online setup that emulated a realistic VoIP call in a web-based setting. The web-based VoIP architecture helped us to gather data that would have not been easily collected if the setup were based on a mobile VoIP app or a lab study.

Similar to any study that involves human users, our study also had certain limitations. First, we recruited native American English speakers through Amazon Turk and our study did not cover any other accent. At this stage of the study, we preferred to focus on one language to show the promising feasibility of the CCCP notion. Future research may be needed to evaluate other accents. Second, some of the recorded data samples in our study had very poor quality and were not recognized by the speech recognition tool or were generating unusual number of errors. After manually listening to and checking these samples with high error rates, we noticed that the quality of these samples was so poor, due to excessive noise, that they were not even easily recognized by the human user. Therefore, we discarded these data samples collected from 6 users from our analysis. We assume that such low audio quality will be discarded by the users in a phone call and users will drop the call.

7.3 Future Work

Integrating with Existing Crypto Phones: Our study showed that automated Checksum Comparison is a practical and feasible approach that can effectively eliminate the data MITM attacks. We also derived several insights for Crypto Phones designs regarding the duration of the checksums and the number of words in the checksum, using which the performance of transcription and the performance of humans user in – now the only – task of Speaker Verification (if required) is improved.

In our future work, we plan to develop an SDK that can independently be used by Crypto Phones. Since in current Crypto Phones, Checksum Comparison is a human task independent of the key exchange protocol, integration of automated Checksum Comparison tool with the current Crypto Phones would be straight-forward. We will incorporate the insights drawn from our study into designing such tool.

We also plan to develop a real-time automated Checksum Comparison tool using off-the-shelf STT engines designed for mobile platforms (e.g., Nuance Dragon Mobile SDK) and we will customize it for the specific needs of checksum transcription. To activate the transcriber for automated Checksum Comparison, we suggest Crypto Phones to ask the user to speak a fixed preamble phrase during the checksum validation phase (e.g., similar to "Ok Google" in Android) through their checksum validation screen. Traditional Crypto Phones, such as Silent Circle, can then use our tool to convert the received checksums to text and automatically compare it with the checksum generated by the protocol locally.

Further Improving the Accuracy of the Transcriber:

1) Limited-Domain Transcribers: The existing off-the-shelf STT tools are designed for the *natural language* grammar (i.e., arbitrary speech communications). This allows the tools to predict the words based on the context of the speech. However, this may not be ideal for the Crypto Phones Checksum Comparison functionality where a string of *isolated words* (not meaningful text) should be transcribed. Since a general purpose transcriber may not always perform accurately in the context of Crypto Phone application domain, designing a specialized grammar for the tool may improve the accuracy of the transcriber. A future research avenue is to consider the requirements of the Crypto Phones to design a special-purpose transcriber that fulfills the narrow and specific need of the automated Checksum Comparison task by designing a limited-domain transcriber on top of the existing off-the-shelf speech-to-text engines.

2) Optimal Checksum Dictionary: The checksum vocabulary is very compact compared to the natural language. For example, PGP word list [26], which is commonly used in Crypto Phones, consists of two lists of 256 phonetically distinct words. We envision that limiting the dictionary of the spoken words would have a significant impact on the accuracy of the transcriber.

We manually analyzed several samples of the transcription and observed that for instance, homophones that sound alike but have different meanings and different spellings (e.g., ate and eight), are not transcribed accurately in our application. By removing the homophones from our data set, we observed that we may improve the overall transcription WER from about 10% to about 9%. In this light, as part of future research, we suggest designing a dictionary of words that are generally transcribed more accurately by the tools (e.g., avoiding homophones in the dictionary).

Integrating with Other Defenses: An orthogonal defense against data or voice MITM attack is to detect the caller source. An interesting caller detection approach is PinDr0p, which determines the source of the call and the path taken by a call [43]. This technique detects and measures audio features to identify the voice codecs, packet loss and noise profiles to identify the caller. However, such approach on its own may not fully detect the MITM attacker which resides on the same network as the victim. Another recent approach to identify the entities in a call is AuthLoop [53], which provides authentication within the voice channel. Unlike Crypto Phones, AuthLoop does not require an Internet connection to exchange the keys, and it authenticates the callers solely within the voice channel. We believe that CCCP is an independent solution and may work in conjunction with these prior techniques. Such integration may be explored in future work.

Potential Sophisticated Attacks on Transcription: Sophisticated attacks against transcription technology have been proposed in recent work [45, 61]. Such attacks produce audio samples unintelligible (though audible) to the human user but interpretable by the transcriber, and may be used to compromise virtual personal assistant apps by running potentially hidden commands given by the attacker. Although such attacks may be conceptually applicable to the automated Checksum Comparison task in our scheme, we assume that in a phone conversation, the user who attends the call (and optionally verifies the speaker) can supposedly detect such suspiciously malformed (robotically sounding) audio samples. A careful further study is needed to evaluate the effect and practicality of such attacks in the context of CCCP.

8 CONCLUSIONS

In this paper, we introduced and studied CCCP, a novel approach to Crypto Phones built on top of *speech transcription*. CCCP works by automating the checksum comparison decisions, thereby reducing the reliance on human users who are prone to making errors, or even clicking-through, such decisions. Our work shows that CCCP can fully detect mismatching checksums and therefore defeat man-in-the-middle attacks that only tamper with the data channel (the most realistic form of attack against any secure communication protocol, including Crypto Phones). CCCP can also drastically reduce the chances of rejecting matching checksums compared to the traditional approach, thereby improving the rates at which secure calls/connections can be established. CCCP can *optionally* facilitate the use of longer checksums. Longer checksums also increase the security level provided by the underlying cryptographic protocols.

As an important side-effect of automating Checksum Comparison, CCCP unburden the users to only perform the single task of validating the identity of the checksum announcing speaker. Our work shows that this fundamental attribute may help increase the robustness of human users in detecting even more sophisticated forms of man-in-the-middle attacks that tamper with both data *and* voice channels, especially when contrasted with currently deployed Crypto Phones, although at the cost of increased delay in speaking longer checksums.

Ackowledgments

We thank anonymous CCS'17 reviewers for their constructive comments and guidance. We are also thankful to Ahana Roy Choudhury, Jesvin James George, Hugo Krawczyk, and all members of the UAB SPIES lab for feedback on previous versions of this paper.

REFERENCES

[1] 2006. *Legal authorities supporting the activities of the national security agency described by the president.* Technical Report. U.S. Department of Justice.

[2] Accessed: 05/12/2017. How The NSA Pulls Off Man-In-The-Middle Attacks: With Help From The Telcos. (Accessed: 05/12/2017). https://goo.gl/Kg4ysn.

[3] Accessed: 05/12/2017. Infosecurity - Microsoft Expands Encryption to Foil Government Snooping. (Accessed: 05/12/2017). http://goo.gl/Ta4H0x.

[4] Accessed: 05/12/2017. NSA and All Major Intelligence Agencies Can Listen in to Encrypted Cell Phone Calls. (Accessed: 05/12/2017). http://goo.gl/KJgoIv.

[5] Accessed: 05/12/2017. TRANSFORM: Flexible Voice Synthesis Through Articulatory Voice Transformation . (Accessed: 05/12/2017). http://festvox.org/transform/transform.html.

[6] Accessed: 05/12/2017. android.speech. (Accessed: 05/12/2017). https://developer.android.com/reference/android/speech/package-summary.html.

[7] Accessed: 05/12/2017. Apple iOS Speech Recognition API. (Accessed: 05/12/2017). https://developer.apple.com/videos/play/wwdc2016/509/.

[8] Accessed: 05/12/2017. Apple SiriâĂŐ. (Accessed: 05/12/2017). hhttp://www.apple.com/ios/siri/.

[9] Accessed: 05/12/2017. Can Dragon Speech Recognition beat the world touch typing record? (Accessed: 05/12/2017). http://goo.gl/PXb4gg.

[10] Accessed: 05/12/2017. Captioning Tools. (Accessed: 05/12/2017). http://goo.gl/Lkncp6.

[11] Accessed: 05/12/2017. CMU Arctic Databases. (Accessed: 05/12/2017). http://festvox.org/cmu_arctic/index.html.

[12] Accessed: 05/12/2017. CMU Sphinx speech Recognition ToolkitâĂŐ. (Accessed: 05/12/2017). http://cmusphinx.sourceforge.net/.

[13] Accessed: 05/12/2017. Dragon Mobile Assistant. (Accessed: 05/12/2017). http://www.dragonmobileapps.com/.

[14] Accessed: 05/12/2017. Dragon Software Developer Kit | Nuance. (Accessed: 05/12/2017). www.nuance.com/for-developers/dragon/index.htm.

[15] Accessed: 05/12/2017. Fact Sheet 9: Wiretapping and Eavesdropping on Telephone Calls. https://goo.gl/tPLhCh. (Accessed: 05/12/2017).

[16] Accessed: 05/12/2017. FreeSWITCH. https://freeswitch.org. (Accessed: 05/12/2017).

[17] Accessed: 05/12/2017. Google Cloud Speech API âĂŐ. (Accessed: 05/12/2017). https://cloud.google.com/speech/.

[18] Accessed: 05/12/2017. Google details how it cut Google Voice transcription error rates. (Accessed: 05/12/2017). http://goo.gl/pYZ9mW.

[19] Accessed: 05/12/2017. iOS and Android OS Targeted by Man-in-the-Middle Attacks. (Accessed: 05/12/2017). https://goo.gl/R3KW40.

[20] Accessed: 05/12/2017. London newspaper wiretapped royals. http://goo.gl/BCzxNM. (Accessed: 05/12/2017).

[21] Accessed: 05/12/2017. ModelTalker Speech Synthesis System. (Accessed: 05/12/2017). http://www.modeltalker.com.

[22] Accessed: 05/12/2017. No More Typing: How To Prepare For The Next Wave Of Voice Search. (Accessed: 05/12/2017). http://goo.gl/GX6DY1.

[23] Accessed: 05/12/2017. Nuance Voice Biometrics. (Accessed: 05/12/2017). http://www.nuance.com/for-business/customer-service-solutions/voice-biometrics/index.htm.

[24] Accessed: 05/12/2017. Open Whisper Systems. (Accessed: 05/12/2017). https://whispersystems.org/.

[25] Accessed: 05/12/2017. Paranoid much? Demand for secure CryptoPhone is so high. http://goo.gl/HkewdZ. (Accessed: 05/12/2017).

[26] Accessed: 05/12/2017. PGP Word List. (Accessed: 05/12/2017). http://philzimmermann.com/docs/PGP_word_list.pdf.

[27] Accessed: 05/12/2017. Recent Advances in Conversational Speech Recognition. (Accessed: 05/12/2017). https://goo.gl/zja4UJ.

[28] Accessed: 05/12/2017. Silent Circle – Private Communications. (Accessed: 05/12/2017). https://silentcircle.com/.

[29] Accessed: 05/12/2017. Speech to Text | IBM Watson Developer Cloud. (Accessed: 05/12/2017). www.ibm.com/watson/developercloud/speech-to-text.html.

[30] Accessed: 05/12/2017. T-Mobile Wi-Fi Calling App vulnerable to Man-in-the-Middle attack. (Accessed: 05/12/2017). http://goo.gl/EQ0gT3.

[31] Accessed: 05/12/2017. The end of typing? Speech recognition technology is getting better and better. (Accessed: 05/12/2017). http://goo.gl/esV53V.

[32] Accessed: 05/12/2017. The Google AppâĂŐ. (Accessed: 05/12/2017). https://www.google.com/search/about/.

[33] Accessed: 05/12/2017. The Zfone Project. (Accessed: 05/12/2017). http://zfoneproject.com/.

[34] Accessed: 05/12/2017. The ZRTP Project - Frequently Asked Questions. (Accessed: 05/12/2017). http://zfoneproject.com/faq.html.

[35] Accessed: 05/12/2017. Use your voice to enter text on your Mac. (Accessed: 05/12/2017). https://support.apple.com/en-us/HT202584.

[36] Accessed: 05/12/2017. Viber Encryption Overview. (Accessed: 05/12/2017). https://www.viber.com/en/security-overview.

[37] Accessed: 05/12/2017. WhatsApp Security. (Accessed: 05/12/2017). https://www.whatsapp.com/security/.

[38] Accessed: 05/12/2017. Why Our Crazy-Smart AI Still Sucks at Transcribing Speech. (Accessed: 05/12/2017). goo.gl/M4oEq4.

[39] Accessed: 05/12/2017. World's first HTML5 SIP client. http://sipml5.org. (Accessed: 05/12/2017).

[40] Accessed: 05/12/2017. ZORG - An Implementation of the ZRTP Protocol. (Accessed: 05/12/2017). http://www.zrtp.org/.

[41] Devdatta Akhawe and Adrienne Porter Felt. 2013. Alice in Warningland: A Large-Scale Field Study of Browser Security Warning Effectiveness. In *Proceedings of the 22th USENIX Security Symposium, Washington, DC, USA, August 14-16, 2013.*

[42] Kartik Audhkhasi, Panayiotis Georgiou, and Shrikanth S Narayanan. 2011. Accurate transcription of broadcast news speech using multiple noisy transcribers and unsupervised reliability metrics. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE.

[43] Vijay A Balasubramaniyan, Aamir Poonawalla, Mustaque Ahamad, Michael T Hunter, and Patrick Traynor. 2010. PinDr0p: using single-ended audio features to determine call provenance. In *Proceedings of the 17th ACM conference on Computer and communications security.* ACM, 109–120.

[44] Mario Cagalj, Srdjan Capkun, and J-P Hubaux. 2006. Key agreement in peer-to-peer wireless networks. *Proc. IEEE* 94, 2 (2006), 467–478.

[45] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. 2016. Hidden voice commands. In *25th USENIX Security Symposium (USENIX Security 16), Austin, TX.*

[46] Diane Damos. 1991. *Multiple task performance.* CRC Press.

[47] Diane L Damos, Thomas E Smist, and Alvah C Bittner. 1983. Individual differences in multiple-task performance as a function of response strategy. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 25, 2 (1983).

[48] Diwas Singh KC. 2013. Does multitasking improve performance? Evidence from the emergency department. *Manufacturing & Service Operations Management* 16, 2 (2013), 168–183.

[49] Cynthia Kuo, Jesse Walker, and Adrian Perrig. 2007. Low-cost manufacturing, usability, and security: an analysis of bluetooth simple pairing and Wi-Fi protected setup. In *International Conference on Financial Cryptography and Data Security.*

[50] Sven Laur and Kaisa Nyberg. 2006. Efficient Mutual Data Authentication Using Manually Authenticated Strings. In *Cryptology and Network Security (CANS).*

[51] Harold Pashler. 1994. Dual-task interference in simple tasks: data and theory. *Psychological bulletin* 116, 2 (1994), 220.

[52] Sylvain Pasini and Serge Vaudenay. 2006. An Optimal Non-Interactive Message Authentication Protocol.. In *CT-RSA.*

[53] Bradley Reaves, Logan Blue, and Patrick Traynor. 2016. AuthLoop: End-to-End Cryptographic Authentication for Telephony over Voice Channels. In *25th USENIX Security Symposium (USENIX Security 16).*

[54] Phil Rose. 2003. *Forensic Speaker Identification.* CRC Press.

[55] George Saon, Hong-Kwang J Kuo, Steven Rennie, and Michael Picheny. 2015. The ibm 2015 english conversational telephone speech recognition system. *arXiv preprint arXiv:1505.05899* (2015).

[56] Sebastian Schrittwieser, Peter Frühwirt, Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Markus Huber, and Edgar R Weippl. 2012. Guess Who's Texting You? Evaluating the Security of Smartphone Messaging Applications.. In *NDSS.*

[57] Svenja Schröder, Markus Huber, David Wind, and Christoph Rottermanner. 2016. When SIGNAL hits the Fan: On the Usability and Security of State-of-the-Art Secure Mobile Messaging. (2016).

[58] Maliheh Shirvanian and Nitesh Saxena. 2014. Wiretapping via Mimicry: Short Voice Imitation Man-in-the-Middle Attacks on Crypto Phones. In *ACM CCS 2014.*

[59] Maliheh Shirvanian and Nitesh Saxena. 2015. On the Security and Usability of Crypto Phones. In *Proceedings of the 31st Annual Computer Security Applications Conference.* ACM.

[60] Ersin Uzun, Kristiina Karvonen, and Nadarajah Asokan. 2007. Usability analysis of secure pairing methods. In *Financial Cryptography & Data Security.*

[61] Tavish Vaidya. 2015. Cocaine noodles: exploiting the gap between human and machine speech recognition. *Presented at WOOT* 15 (2015), 10–11.

[62] Serge Vaudenay. 2005. Secure Communications over Insecure Channels Based on Short Authenticated Strings. In *CRYPTO.*

[63] Ruishan Zhang, Xinyuan Wang, Ryan Farley, Xiaohui Yang, and Xuxian Jiang. 2009. On The Feasibility of Launching the Man-in-the-Middle Attacks on VoIP from Remote Attackers. In *ASIACCS.*

[64] Phil Zimmermann, A Johnston, and J Callas. 2011. ZRTP: Media path key agreement for unicast secure RTP. *Internet Engineering Task Force (IETF)* (2011).

A APPENDIX

A.1 Evaluated Checksum Words

across, adrift, adverb, advisedly, again, Alberta, ally, almost, already, amount, anguish, announce, anyway, appreciation, approach, argument, around, articulate, artist, artistic, associate, Australia, back, bankruptcy, bargain, beady, before, behind, bell, beside, beyond, Billy, biologist, black, bless, bored, bourgeois, bow, branch, breed, bright, brought, burst, bursting, business, butchers, California, candidate, canoe, canyon, carefully, carried, case, cash, catch, cease, certainly, chair, challenge, change, charcoal, charge, charm, chattering, cheerful, chivalry, choking, class, clothes, club, cluster, command, commissionaire, commit, companion, compound, conduct, confidence, consider, contemplating, continue, contribute, convince, criticize, cryptic, curious, cutter, day, daylight, dead, delicate, delight, determine, devil, devotion, diameter, die, direction, disgust, disturb, diversion, document, dog, dominate, door, doubling, down, dreadfully, drop, duality, each, earth, editorial, eighteen, employer, encourage, escape, ethic, every, everywhere, exclamation, existence, expect, expectancy, experience, express, face, fact, fail, faith, fascinate, father, feeling, fellow, fifty, fighting, finality, finger, fingertip, fire, fish, fix, flaming, flash, follow, foolish, forever, forgot, forgotten, forth, forward, free, fresh, gaunt, general, giant, girl, gloom, glorious, go, gone, graduate, grain, great, greatly, growth, happen, here, hesitate, hiding, himself, Honolulu, hoof, hour, humanity, hundred, hyena, hypothesis, individualism, instinct, intention, intermittent, interurban, irritation, joy, judge, junk, jury, lake, last, laugh, life, linger, lips, long, luxury, magnificent, maintain, man, manage, market, match, meat, minute, miracle, mirth, moist, more, much, muzzle, nauseating, nice, nope, normal, now, obedient, object, once, oppression, oppressive, organization, orthodox, oursel, outsider, oversee, pain, pal, pan, part, partnership, pass, passionately, path, people, Philips, physique, place, plantation, plausible, player, pleasure, point, preferring, present, price, princess, proceed, produce, promise, property, prostrate, puzzle, quadrupling, quarrel, quiet, quivering, quotation, rapid, recollection, recover, refugee, refuse, release, resident, resist, restaurant, return, review, revolver, reward, rhythm, riffle, rising, road, rock, round, scare, scream, sensation, service, seven, seventeen, shaking, shoulder, Siberia, sign, simple, singing, situation, sketchy, slave, sleep, smash, smile, smoothly, snap, soft, solicitor, sound, speech, spite, spokesman, stable, state, States, stick, stranger, strength, struck, suddenly, sufficient, sugar, sunshine, suppress, surprise, table, teeth, temptation, terribly, terror, themsel, there, thirty, thousand, thrill, time, tobacco, today, together, tomorrow, tremendous, trouble, trout, turn, Unconsciously, understand, United, untoward, urge, value, view, violate, vital, vitality, vocabulary, voice, walk, way, weapon, week, weight, widely, wildly, willing, wolf, woman, word, worth, yard, year, yet, yield

A.2 Study Instruction

Making the call Instruction

1) Click "Start Call" to initiate the call.
2) Allow your browser to access microphone.
3) Wait for the call to get connected.
4) After hearing the prompt, click "Next" to listen to the instruction.

Speaking Instruction

1) Click "Start Recording" to record your voice.
2) Speak the displayed words slowly and deliberately.
3) Once you are done speaking click "Stop Recording" to stop recording.
4) Click "Next" to move forward.
5) System is set to disconnect you from the call if you don't answer within 10 seconds.

Familiarization Instruction

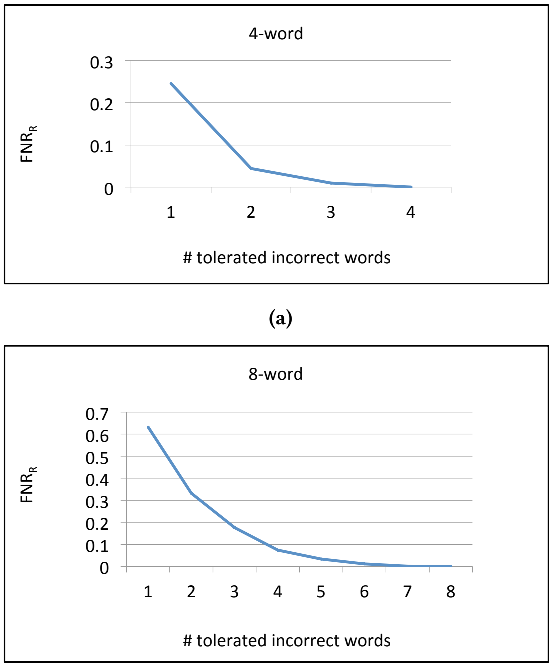
Listen to the story to get familiarize with the voice. Click "Listen Again" to hear the voice again and click "Next" to move forward.

Speaker Verification Instruction

Regardless of the quality of recordings, you should answer "Yes" if you can recognize the speaker's voice. Answer "No" if you think it is not the speaker's voice. Click "Replay" if you need to hear the voice again.

1) You can replay the voice once.
2) System is set to disconnect you from the call if you don't answer within 10 seconds.

A.3 Additional Figures and Tables



(a)

(b)

Figure A.1: Effect of the Number of tolerated incorrect words on FNR$_r$