On the Security and Usability of Crypto Phones

Maliheh Shirvanian and Nitesh Saxena
University of Alabama at Birmingham
Birmingham, AL, USA
maliheh@uab.edu, saxena@cis.uab.edu

ABSTRACT

Crypto Phones represent an important approach for end-to-end VoIP security, claiming to prevent "wiretapping" and session hijacking attacks without relying upon third parties. In order to establish a secure session, Crypto Phones rely upon end users to perform two tasks: (1) *checksum comparison*: verbally communicating and matching short checksums displayed on users' devices, and (2) *speaker verification:* ascertaining that the voice announcing the checksum is the voice of the legitimate user at the other end. However, the *human errors* in executing these tasks may adversely affect the security and usability of Crypto Phones. Particularly, failure to detect mismatching checksums or imitated voices would result in a compromise of Crypto Phones session communications.

We present a human factors study, with 128 online participants, investigating the *security* and *usability* of Crypto Phones with respect to both checksum comparison and speaker verification. To mimic a realistic VoIP scenario, we conducted our study using the WebRTC platform where each participant made a call to our IVR server via a browser, and was presented with several challenges having matching and mismatching checksums, spoken in the legitimate user's voice, different speakers' voices and automatically synthesized voices. Our results show that Crypto Phones offer a *weak level of security* (significantly weaker than that guaranteed by the underlying protocols), and their usability is low (although might still be acceptable). Quantitatively, the overall average likelihood of failing to detect an attack session was about 25-50%, while the average likelihood of accepting a legitimate session was about 75%.

Moreover, while the theory promises an exponential increase in security with increase in checksum size, we found a degradation in security when moving from 2-word checksum to 4-word checksum.

1. INTRODUCTION

Internet-based voice, video and text communication, collectively referred to as VoIP, is one of the most popular mechanisms of online communication deployed today. Unlike the traditional PSTN (public-switched telephone) networks, VoIP communication may be more easily susceptible to various forms of attacks, including eavesdropping [1, 2] and session hijacking or man-in-the-middle (MITM) [34] attacks. As a prime example, governments, intelligence agencies, private organizations, and even cyber criminals, often "wiretap" VoIP calls, legally or illegally [5], for a variety of purposes including crime investigation, political or military endeavors [16], and theft of private information from the victims, especially those who are famous, rich or powerful [7].

In light of these vulnerabilities, a fundamental goal is to secure, that is, encrypt as well as authenticate all VoIP communication. Ideally, this objective should be achieved without relying upon a third-party (such as an online server) or a dedicated infrastructure (such as a PKI) because such centralized services may themselves get compromised or be under the coercion of law enforcement authorities. Crypto Phones, such as Zfone [12], Silent Circle [11], RedPhone and Signal [9], are mobile apps (or hardware devices) claiming to offer precisely such end-to-end VoIP security guarantees based on a purely *peer-to-peer*, *user-centric* mechanism. Recent media reports seem to suggest that Crypto Phones are in high demand both in the commercial and personal domains [10].

To secure the data (voice, video or even text) communication, Crypto Phones require a cryptographic key, which is agreed between the end parties using a special-purpose key exchange protocol [15, 33]. This protocol results in a *short* (e.g., 16-bit or 2-word) checksum, called a *Short Authenticated String (SAS)*, per party, with the inherent property that if an MITM attacker is "present" during the protocol, the *checksums will not match*. As a result, to ensure that the MITM attacker did not interfere with the protocol messages and compromise the protocol security, Crypto Phones rely upon end users to perform two crucial tasks (Figure 1 visualizes the benign setting):

1. *Checksum Comparison*: Verbally communicating and matching short checksums displayed on each user's device, *and*
2. *Speaker Verification:* Ascertaining that the voice announcing the checksum is the voice of the legitimate user at the other end of the call.

Theoretically, the SAS protocol deployed in a Crypto Phone application limits the MITM attack success probability to 2^{-k} for a k-bit SAS checksum. For instance, for a 16-bit checksum, the protocol suggests that the MITM attacker cannot succeed with a probability better than 0.0015% (2^{-16}). However, in practice, the *human errors* in executing the checksum verification and speaker verification tasks may adversely affect the security of Crypto Phones. Particularly, failure to detect *mismatching checksums*, or *imitated voices* would result in a compromise of Crypto Phones session communications. In the first attack scenario (Figure 2), the MITM attacker only manipulates the data communication during the protocol – this results in mismatching checksums as an inherent characteristic of the protocol. If the users erroneously accept mismatch-

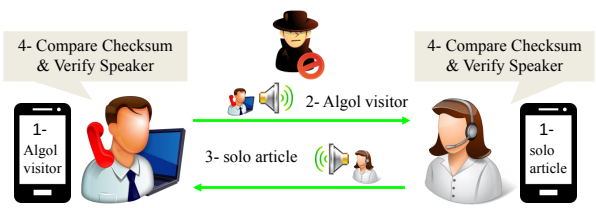
ACSAC '15, December 07-11, 2015, Los Angeles, CA, USA

Step 1: Bob's and Alice's devices show their respective checksums as a result of SAS protocol execution.

Step 2-3: Bob and Alice exchange their respective checksums via verbal communication.

Step 4: Bob is asked to compare his checksum with the one provided by Alice plus verify Alice's voice, and Alice is asked to compare her checksum with the one provided by Bob plus verify Bob's voice, and accordingly accept or reject the process. In this case, both should be accepting thereby establishing a secure session. However, if users erroneously reject matching checksums, the protocol will be aborted and need to be restarted.

Figure 1: Crypto Phones benign setting – original voices; matching SAS (this is the setting subject to our usability assessment).



Step 1: Bob's and Alice's devices show their respective checksums as a result of SAS protocol execution.

Step 2-3: Bob and Alice exchange their respective checksums via verbal communication.

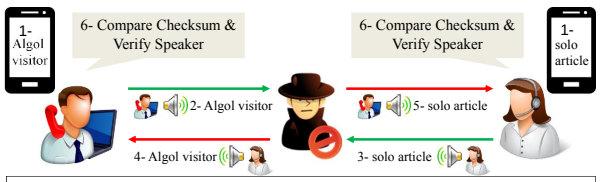
Step 4: Bob is asked to compare his checksum with the one provided by Alice plus verify Alice's voice, and Alice is asked to compare her checksum with the one provided by Bob plus verify Bob's voice, and accordingly accept or reject. In this case, both should be rejecting because checsums do not match (although voices are valid) thereby preventing the attack. However, if users erroneously accept mismatching checksum, the attack will succeed.

Figure 2: Crypto Phones MITM attack scenario 1 – original voices; mismatching checksums. If users accept mismatching checksums, security is compromised.

ing checksums, the security will be compromised. In the second attack scenario (Figure 3), the MITM attacker manipulates the data communication as well as the SAS/voice communication. It inserts its *own voice*, or an *automatically generated voice mimicking the user's voice*, so that the SAS checksums *appear to be matching* to the users. If users accept these imitated voices, the security will be compromised. For automatic voice generation, *voice morphing* [3] could be used whereby an attacker collects a few minutes of victim's speech, and uses a voice converter tool to create arbitrary checksums that were not earlier spoken by the victim.

In addition to undermining security, the human errors may negatively impact the user experience of Crypto Phones in the benign settings. That is, rejecting matching checksums spoken in legitimate users' voices would degrade overall usability, as users will have to re-execute the protocol. Further, perhaps more seriously, repeated executions may indirectly hamper overall security. Repeated protocol runs could frustrate the users to the point they may start accepting even MITM attack instances, or may give up using the Crypto Phones apps altogether, and rather resort to apps that do not at all protect the communications. Similar (negative) security consequences of poor usability have been noted in prior *device pairing* research [24].

OUR CONTRIBUTIONS: In this paper, we investigate and empirically quantify the *security* and *usability* of Crypto Phones with re-



Step 1: Bob's and Alice's devices show their respective checksums as a result of SAS protocol execution.

Step 2-3: Bob and Alice exchange their respective checksums via verbal communication; Mallory drops these packets.

Step 4-5: Mallory generates Alice's checksum mimicking Bob's voice, and Alice's checksum mimicking Bob's voice (Mallory's own voice or auto-generated voice), and inserts them into the respective channels.

Step 6: Bob is asked to compare his checksum with the one provided by Alice plus verify Alice's voice, and Alice is asked to compare her checksum with the one provided by Bob plus verify Bob's voice, and accordingly accept or reject. In this case, both should be rejecting because checksums match but they are not spelled in original users' voices, thereby preventing the attack. However, if users erroneously accept imitated voices, the attack succeeds.

Figure 3: Crypto Phones MITM attack scenario 2 – imitated (attacker's or auto-generated) voice; matching checksums. If users accept imitated voices, security is compromised.

spect to users' performance in the aforementioned checksum comparison and speaker verification tasks. Our human factors study was conducted with a total of 128 Amazon Mechanical Turk participants, and brings about three main contributions outlined below:

1. **Study Design Emulating Real-World VoIP Setting:** To mimic a realistic VoIP scenario, we conducted our study using the WebRTC (web-based real-time communications) platform where each caller made a call from a web browser to an extension on our soft telephony switch, and got connected to an IVR (interactive voice response) server, which acted as the callee and played back the short authenticated checksums. Throughout the experiment, the user was guided using spoken instructions (via the IVR prompt) and displayed instruction on her screen. All user interaction with the callee was through the web server which received the web clicks and transformed them into DTMF (dual-tone multi-frequency) tones acceptable by the softswitch. Our study design and implementation is described in *Section 3*.
2. **Extensive Evaluation of Benign and Attack Scenarios:** In our study, we emulated the benign setting as well as different attack scenarios for Crypto Phones (Figures 1–3). Each participant was presented with several challenges corresponding to matching and mismatching checksums, spoken in a legitimate user's voice, different speakers' voices and automatically synthesized voice. For the latter scenario, we used the voice morphing technique to convert an attacker's voice to a victim's voice. Our results show that Crypto Phones offer only a weak level of security, significantly weaker than that guaranteed by the SAS protocols. The overall average likelihood of failing to detect an attack session was about 25-50%. On the positive side, we found that the usability of Crypto Phones might be acceptable in practice (although not very high). The average likelihood of accepting a legitimate session was about 75%, and users took less than 5 seconds to complete their tasks and generally provided positive ratings to the system. Our results are presented, statistically analyzed and interpreted in *Section 4*.
3. **Effect of SAS Checksum Size:** The length of the SAS checksum is a crucial security parameter for Crypto Phones. Theoretically, the security of Crypto Phones should increase exponentially with increase in the size of the checksum. For example, the security level should increase by a factor of 65536 when

moving from 16-bit (2-word) checksums to 32-bit checksums. In our study, we investigated the practical implications of the checksum size on the security (and usability) level of Crypto Phones. To achieve this, we compared the performance of two groups of participants (64 per group), following a *between-subjects* study design, who performed the checksum comparison and speaker verification tasks on 2-word and 4-word checksums. The results show *degradation* in overall security due to the increase in human errors specifically in comparing longer checksums. Our results are described in *Section 4.2.3*.

2. BACKGROUND AND RELATED WORK

2.1 Threat Model and SAS Protocols

A Crypto Phone SAS protocol between Alice and Bob is based upon the following communication and adversarial model, adopted from [33]. The devices being associated are connected via a remote, point-to-point high-bandwidth bidirectional VoIP channel. An MITM adversary Mallory attacking the SAS protocol is assumed to have full control over this channel, namely, Mallory can eavesdrop and tamper with messages transmitted. However, an additional assumption is that Mallory cannot insert voice messages on this channel that mimic Alice's or Bob's voice. In other words, the voice channel (over which the SAS values are validated) is assumed to provide integrity and source authentication.

A number of SAS protocols exist in the literature (e.g., [28, 33]) that a Crypto Phone implementation may adopt. SAS protocol is an authenticated key exchange protocol, which allows Alice and Bob to agree upon a shared authenticated session key after validating a short string over an auxiliary channel (such as the voice channel used in Crypto Phones). The protocol results in a short checksum (e.g., 16-bit) per party – matching checksums imply successful secure session establishment, whereas mismatching checksums imply an MITM attack. As mentioned earlier, these protocols limit the MITM attack success probability to 2^{-k} for k-bit SAS data. Once the SAS protocol and SAS validation process completes, all data between Alice and Bob is secured (e.g., using authenticated encryption). The session data may include voice, text or video data. In fact, a Crypto Phone texting application may utilize the SAS approach to secure the text channel by means of SAS validation over the voice channel, as employed by Silent Circle [11]. This means, if the attack succeeds, all subsequent communication would be revealed, and all text communications could also be manipulated.

2.2 SAS Encodings and Comparison

There are two types of SAS checksums commonly used in Crypto Phones and device pairing applications. The first is the numerical encoding where the checksum is usually presented in the form of 6-8 digits numbers. The second is PGP words where SAS is mapped to words (similar to NATO phonetic alphabet).

Compare-Confirm and *Copy-Confirm* are the two popular SAS checksum comparison methods as introduced in [32]. In Compare-Confirm, the SAS checksum is displayed on each party's screen, they verbally exchange their respective checksums, and both accept or reject the connection by comparing the displayed and spoken checksum. In Copy-Confirm, one party reads the encoded checksum to the other party, who types it onto his/her device, and get notified whether the checksum is correct or not. The inaccuracy of the users in reading, and typing the codes might lead to false acceptance of an MITM attack session, or false rejection of a legitimate session. In this study, we are studying unidirectional Compare-Confirm checksum comparisons, given this is the most commonly deployed approach on Crypto Phones.

2.3 Speaker Verification Task

Speaker verification is the task of authenticating a claimed identity by means of analyzing a spoken sample of the claimant's voice. Manual speech perception and recognition is a complex task, which depends on many parameters, including: the length of the samples, the number of samples, the source of the samples (familiar vs. famous people), and combinations thereof [29]. There exists an extensive literature in linguistics, analyzing human speech recognition capabilities over different parameters [19, 22, 25, 29]. This line of research shows that the shorter the sentence, the more hard it may be to identify the source.

2.4 Voice Conversion

With the advancement in speech technology, there exists automated systems that can reproduce someone's voice. Examples are: text-to-speech tools, such as AT&T natural voices [4], voice synthesis tools, such as ModelTalker [8], voice manipulation tools, such as Voxal Voice Changer [13], and voice converters (or transformers), such as Festvox [3]. Voice converters have the advantage of producing more natural voices with less training data. They get trained with relatively small training data, when compared, for example, with the limited domain tools, which require samples of all possible words or phrases or phoneme spoken by the target.

One of the most popular voice conversion tools is Festvox that synthesizes the voice by modifying speech characteristic based on the conversion rules. Festvox conversion uses Gaussian Mixture Model (GMM) on joint probability density of source and target features. The optimum mixture sequence is then determined by maximizing the likelihood function as described in [31]. We created automated voices using the Festvox voice transformation tool and we assume that in real life, the attacker can gain a similar morphed voice quality as in our work, by collecting only a few minutes of the victim's voice using such off-the-shelf voice conversion tools.

2.5 Related Work

There exists prior work that studied the security or usability of the checksum comparison or speaker verification tasks.

An extensive amount of research concentrates on the checksum comparison task in the context of the *proximity-based device pairing* application. A wealth of prior works exists that uses SAS protocols and different out-of-band (OOB) channels for the purpose of device pairing (see survey study [23]). However, there is a significant difference between checksum comparisons in our work and checksum comparisons in prior work. Device pairing involves devices and their users who are *physically nearby*, whereas Crypto Phones pertain to *remote* devices and users communicating over a VoIP channel. Such remote interactions are clearly different from physical interactions, and therefore users' performance in Crypto Phones checksum comparisons *cannot be deduced* from users' performance in device pairing checksum comparisons.

The speaker verification task in security applications has also been studied previously. A recent work [30], investigated the feasibility of a *voice morphing*[1] and a *voice reordering*[2] attack against Crypto Phones. The user study reported in [30], evaluated the security of Crypto Phones with a web-based *survey* that presented samples of the original speaker, different speakers and morphed voices

[1]A voice morphing attack is an attack against human voices in which the attacker creates a synthesized voice that mimics a speaker's voice to fool the victims into accepting it as the original speaker's voice

[2]A reordering attack is an attack in which the attacker records individual words spoken by the victim from previous voice conversations and remixes them to build any utterance in the victim's voice.

to the participants. The subjects were asked to rate the quality of the recordings in terms of genuineness. They were also asked to listen to some recordings to get familiarized with a given speaker's voice, and then recognize if a new recording corresponds to the same speaker or not.

Although we study a similar attack in our study, there are significant differences between our study and the "attack-only study" of [30]. *First*, usability of the system was not assessed in [30], while we evaluate both security and usability of Crypto Phones. *Second*, the focus of their study was on the evaluation of the security of Crypto Phones only with respect to the speaker verification task, *but not* the checksum comparison task. Since the security of Crypto Phones relies upon *both tasks in conjunction*, assessing users' security behavior in just one task is not sufficient. The *third* characteristic of the study of [30] pertains to its survey-based nature, where the primary task of the participants is speaker verification. In contrast, in real world, the primary task of Crypto Phones users is to make a phone call, and the secondary task is speaker verification (plus checksum comparison). Therefore, the attacks might perform differently in real world where the user may want to complete the lengthy security tasks rapidly (or ignore them completely if possible), and move forward to the main primary task (voice conversation). Moreover, the noise associated with a VoIP channel is different from a static clip on a web-page, even if the same recording is played. Hence, users might perceive the voices differently when played over the phone call. The *fourth* limitation of the prior study stems from the relatively small sample sizes (only 30 participants, each presented with 10 quality assessment and 10 speaker verification challenges). Our study is performed with a total of 128 participants, each of whom was presented with 64 challenges. Larger sample sizes provide larger confidence in the results.

An additional important difference from all of the related prior work is that we study the security and usability of *both tasks*, checksum comparison and speaker verification, *in conjunction*. Prior works either evaluated the former alone [23] or the latter alone [30]. Therefore, we investigate the usability of Crypto Phones as well as the security of Crypto Phones as a whole (with respect to both checksum comparison and speaker verification tasks) in an environment where a real VoIP call was made by each participant, and all the tests are performed over this call. Based on the discussion above, the current work cannot not be directly derived from related prior work, but we briefly compare our results with prior results in Section 5.1.

3. STUDY PRELIMINARIES AND DESIGN

In this section, we describe our study goals and design based on morphing attacks on word-based SAS. As reported in [30], the reordering attack is effectively implementable against numerical SAS. Therefore, we focus only on the morphing attack on word-based SAS which is the most difficult attack for the users to detect.

3.1 Objectives and Metrics

Our study is designed to measure the security and usability of Crypto Phones. The specific goals of the study are outlined below:

1. **Robustness:** *How well do the users perform at the tasks of checksum comparison and speaker verification together?* For usability assessment, we are interested in finding out how often users accept matching checksums spoken in an original speaker's voice. *False Negative Rate (FNR)* represents the probability of rejecting such instances. False rejections force the users to restart the protocol affecting the overall usability. That is, the lower the FNR, the better the usability.

   For security assessment, we are interested in determining how often users accept mismatching or matching checksums in a different speaker's voice or a morphed voice. *False Positive Rate (FPR)* denotes the probability of accepting such instances. False acceptance implies the success of the MITM attacker and a compromise of the security of Crypto Phones session communications. The lower the FPR, the better the security. The theoretical FPR for Crypto Phones, with a SAS of size k bits, is 2^{-k}, as stated earlier. For example, with a 16-bit SAS, the theoretical FPR is 0.0015%, which should serve as a baseline for the user-centric (practical) FPRs resulting from our study.

2. **Efficiency:** *How long it takes for the users to complete the checksum comparison and speaker verification tasks (benign setting or attack scenario)?* The delays incurred in performing these tasks, referred to as *time* to completion, may impact the overall usability of the system. In making their decisions, users might hesitate or request the other party to repeat the checksum, which may prolong the process and delay establishment of the phone call. To capture this notion, we define a metric called *Replay Rate (RR)*, which is the fraction of the times the test challenges were replayed by the participants while making their decisions. Higher RR indicates lower efficiency.

3. **User Perceptions:** *How usable do the users find the overall Crypto Phone application requiring the checksum comparison and speaker verification tasks?* We are interested in determining as to how comfortable and confident people are in using the application with respect to system complexity, need for training, support or maintenance. In our study, we measure user perceptions using a standard scale questionnaire, called the Simple Usability Scale (SUS) [18].

4. **Effect of Checksum Size:** *How much does the checksum size affect the security and usability of Crypto Phones?* As mentioned earlier, theoretically, the security of Crypto Phones should increase exponentially with increase in the size of SAS. For example, with a 16-bit SAS, the probability of the success of an MITM is 0.0015%, while for a 32-bit SAS, the probability degrades down to 2.33×10^{-8}%. We are interested in quantifying the *practical* change in FPR (and FNR) as well as completion time with increase in SAS size. In our study, we measure this effect for 2-word (16-bit) vs. 4-word (32 bit) checksums.

3.2 Study Design and Implementation

In our study, we considered different aspects pertaining to the usability and security of Crypto Phones, and designed and implemented a system that closely simulates Crypto Phones voice call initiation. Due to the popularity of web-based voice applications, we implemented a web-based voice telephony system as the platform for our study. Our system emulates Crypto Phones unidirectional call establishment/authentication (such as a scenario where a customer service call is being made by a user). Our results, however, easily extrapolate to a bidirectional setting, as discussed later in Section 5.1.

3.2.1 Design Components

The main components of our study design are the telephony platform and the web-based application, described below.

Telephony Platform: We set up a softswitch on an Amazon Ubuntu Server 14.04 LTS (HVM) t2.micro[3] instance. We ran a FreeSWITCH 1.5.14b as the softswitch [6]. The open source FreeSWITCH software supports VoIP protocols including Session

[3] 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GB memory

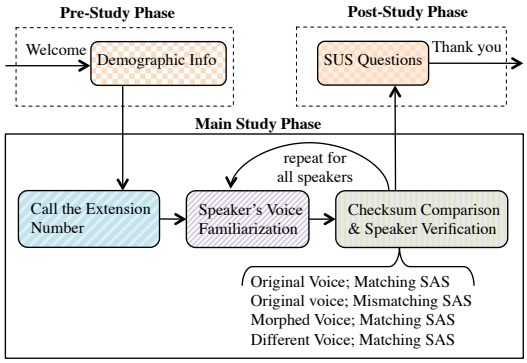


Figure 4: The study involves three phases, pre-study, main study and post-study. Extension number and IVR menu is participant-specific.

Initiation Protocol (SIP), IVR, and WebRTC (Web-based Real-time Communications) that are essential components to connect the web based clients to the switch. We modified the security group (virtual firewall) of the Amazon EC2 instance to allow traffic to/from the FreeSWITCH, and configured NAT functionality to support different type of clients (participants machines) regardless of their ISP.

We configured the IVR system on FreeSWITCH to play the instructions, voice recordings of speakers and SAS challenges, based on the commands it receives from the web-based application.To better simulate the practical Crypto Phone application, we did not use synthesized text-to-speech voice for IVR commands. That is, IVR voice commands are either audio recordings of human speakers or audio recordings of morphed voices.

Web-based Application: The web server that supports the application was hosted on a Debian 7.4 with 2 x Intel(R) Xeon(TM) CPU 3.20GHz and 3.87 GB of memory. The web-based application developed in PHP, JavaScript and HTML5 was the connection point of the participant and the experimental setup. It consisted of demographical and SUS surveys, web-based WebRTC voice client supporting DTMF, and a database client to connect to the database server to read questions and store participants' responses. The PostgreSQL database was located on a Debian 7.4 machine with 1 x Intel(R) Xeon(R) CPU E5-2620 0 @ 2.00GHz and 512 MB of memory. The database stored the list of SAS challenges, usability (SUS) questions, answers to demographic and SUS questions, and answers to the SAS challenges, time taken to answering each questions, and number of replays required by each user to decide to accept or reject a given SAS challenge.

The web-based voice client uses sipML5 open source HTML5 SIP client API [14]. SIP/SDP stack of the API is written in JavaScript and the network transport uses WebSockets. Its media stack depends on WebRTC which is natively provided by the web browser. We developed our web-based voice client with JavaScript using this API. Our program is supported on Chrome and Firefox and is extensible to other browsers. To make and receive calls, first, the SIPml media and signaling engine is initiated. Then, a SIP stack is created that is the base object through which a user is registered to the FreeSWITCH and can make/receive calls.

3.2.2 Study Flow and User Interactions

We published our study on Amazon Mechanical Turk, and recruited 128 subjects who were asked to follow a link to our *web-based voice application* (described in Section 3.2.1). To investigate the effect of SAS size on usability and security, we grouped the participants into two sets. 64 participants answered challenges regarding 2-word (16-bit) SAS, and the other 64 participants answered questions about 4-word (32-bit) SAS. Each participant was assigned a unique ID which was later utilized for the payment and data analysis purposes. The average duration of the experiment was around 20 minutes. Our study was approved by the Institutional Review Board (IRB) of our university and the participation in the experiment was voluntary. The flow of the experiment and the tasks that the users had to complete is depicted in Figure 4, and described below.

- **Pre-Study phase**

  First, the participants were presented with a welcome message and instructed to use a Chrome or Firefox browser, and grant the web application with access to their microphone for the duration of the experiment in order to establish the call.

  Second, the participants were asked to fill out a demographic questionnaire. These questions polled 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.

- **Main Study Phase**

  First, each participant was registered to a VoIP softswitch (described in Section 3.2.1) through the web-based application, and dialed an extension number to get connected to an interactive voice response (IVR) module of the softswitch. All user interaction with the switch was performed through DTMF tones via clicking buttons on the web-page.

  Second, once the participants get connected to the IVR menu, they were asked to get familiar with a single speaker's voice. All voice recordings and instructions were arranged as part of the IVR prompts. There was no restriction on replaying the instructions and voice recording for familiarization. Written instructions were provided in each page to further support the voice instructions in the IVR prompts.

  Third, during the *challenge phase*, the participants were posed with checksum comparison and speaker verification challenges. A set of 2-word or 4-word SAS challenges (described in the next subsection) related to that speaker were displayed on the user's web page (one at a time), and the IVR menu corresponding to the displayed SAS challenge was played. The participant task was to match the spoken words with the displayed words, and click on a "yes" button, if the words matched and voice was of the original speaker (previously familiarized voice). If the words did not match, or the voice did not match the original speaker's voice, they should click the "No" button. The participant could replay the SAS challenge for the second time, if he/she could not validate the SAS in the first attempt.

- **Post-Study Phase**

  First, the participants were presented with the SUS questionnaire to rate their experience with the system and the underlying tasks.

  Second, upon answering the SUS questions, the users were instructed to enter their Unique ID back in Amazon Turk for the payment purpose. The answers and number of replays, as well as the time taken by the user to complete each challenge was stored in the database for later analysis.

Randomness with Latin Square Design: To provide randomness throughout the experiment so as to minimize potential learning biases, the unique ID assigned to each participant was used to connect them to a participant-specific IVR menu. We developed a Java application that creates the IVR config file based on a 64×64 *Latin Square* [20], to set unique IVR menus for each user.

Voice Dataset: To create the voice samples used in the main study phase, we used the CMU ARCTIC US English single speaker database. We picked two female speakers (CLB and SLT), which we call female 1 and female 2, and two male speakers (BDL and RMS), which we call male 1 and male 2, and made male to male and female to female speaker conversions using the Festvox transformation tool. In our experiment, we trained the system with first 100 sentences of ARCTIC data set and created morphed SAS challenges using the rest of the dataset. We performed 4 conversions (male 2 to male 1, male 1 to male 2, female 1 to female 2, and female 2 to female 1). In each conversion, the source voice is considered as the attacker's voice and target's voice is considered as the victim's voice. For the familiarization purpose, we played a 1-minute recording of the target speaker (the victim) which was not available in training and test dataset. We used the same dataset as a possible large dictionary for SAS encoding.

3.3 SAS Challenges and Related Metrics

In the challenge phase of our study, we randomly displayed and played sixteen checksum comparison and speaker verification challenges. These challenges emulate the benign scenario as well as the MITM scenario for Crypto Phones. We define four categories of challenges, each including 4 instances, as defined below and repeated the challenges for the four different speakers.

1. **Original Voice; Matching SAS**: In this set of challenges, the original user says the same SAS as the one displayed on the participant's current screen. This set captures the success of participants in recognizing a familiar (original speaker's) voice speaking a matching SAS (the benign case – Figure 1). A "yes" answer to this challenge shows that the participant could match the SAS and detect the familiar voice correctly. A "no" answer shows the failure of the user in detecting a familiar voice/matching SAS, and defines the FNR. Recall from Section 3.1 that high FNR reduces overall usability.
2. **Original Voice; Mismatching SAS**: In this scenario, the SAS that the original user says is different phonetically distinct from the SAS that is displayed on the screen. This test mostly shows the accuracy of participants in comparing the checksums, i.e., detecting an MITM attack on the data channel that does not manipulate the voice (Figure 2). A "yes" answer to this challenge shows that the participant could not detect the mismatch between the two SAS values. Such instances of incorrectly accepting a wrong SAS contribute to the FPR, which defines attack success rate (higher the FPR, more successful the attack). A "no" answer, on the other hand, shows success of the participant in comparing the checksums. We also studied how the placement and number of incorrect SAS words in the checksum would affect the FPR (i.e., 1st word mismatch, 2nd word mismatch, etc.).
3. **Different Voice; Matching SAS**: In these challenges, the attacker says the same SAS, as the displayed SAS, in his/her own voice without any conversion. This test shows the success of participants in distinguishing a different speaker's voice (Figure 3). A "yes" answer to this challenge shows failure of the participant in distinguishing the speaker (considering a different speaker's voice as the familiar voice) and contributes to the FPR. A "no" answer shows success of the user in detecting the different voice.
4. **Morphed Voice; Matching SAS**: In this set, the attacker says the same SAS, as the displayed SAS, in the victim's voice by using the voice converter tool (morphing attack). This test captures the success of participants in detecting a voice-based MITM attack (Figure 3). A "yes" answer to this challenge shows that the participants failed to detect the attack (considering a morphed voice as the familiar voice), and contributes to the FPR. A "no" answer, in contrast, shows the success of the participant in detecting the attack.

In our study, we considered errors related to both speaker verification and checksum comparison tasks. To study the errors that are most probably related to speaker verification, we displayed a SAS on the user's web-based application, and played the same SAS in the original speaker's voice, in a different speaker's voice and in the morphed voice. Here, since the displayed and spoken SAS match, most probably the users' answers are based on the voice (and not the SAS). However, it could be the case that the user rejects a voice because of the failure in matching the words. Similarly, to study the errors that are most probably related to checksum comparison, we displayed the SAS word on the web-based application and played a different SAS spoken by the original speaker. Here, since the voice is familiar (the original speaker's voice) and only the words do not match, most probably users' responses are based on the SAS (and not the voice). However, there might be some rejections that are related to speaker verification, which means the user failed in rejecting the mismatched words, but by mistake rejected the familiar voice. Although we could have clearly asked users their reasoning behind rejecting the challenge (mismatch in the words, or presence of an imitated voice), we preferred not to make such a distinction. The reason is that we wanted to closely simulate a real Crypto Phone application in which the only task of the user in the SAS validation procedure is "accept" or "reject" (i.e., based on both checksum comparison and speaker verification tasks together).

4. STUDY RESULTS

4.1 Participant Demographics

We recruited 128 M-Turk workers (as discussed in Section 3.2.2) who reside in the US. Table 1 summarizes the demographic information of the participants, collected via our demographic questionnaire. In both groups, the first language of most of the participants is English. Most participants do not have any hearing impairment. More than 50% of them use VoIP applications daily, and only a negligible fraction has never used VoIP applications before. A majority of the participants are young, and well-educated with at least a diploma. The various demographic attributes of our two groups of participants are very similar, which allows us to meaningfully compare the experimental results for 2-word and 4-word SAS following a between-subjects methodology.

4.2 Results and Analysis

4.2.1 2-word SAS Checksum

We first analyze the benign case, i.e., the one corresponding to our first set of challenges (Original Voice; Matching SAS). The first column of Table 2 summarizes these results. The FNR (averaged over all participants and all the four speakers) is about 22%, which may be acceptable in real-world applications. The time taken by the participants to respond to these challenges is quite low, only 3.05 seconds on an average. If we look at each speaker individually, we see that the FNR, shown in the first column of Table 4, is relatively low, varying between 17% to 26% for all speakers. Such FNRs may be indicative of an acceptable (but not high) level of usability in practice. Based on the Friedman test[4], we did not find

[4]The Friedman test is a non-parametric statistical test used to detect differences in treatments across multiple matched test samples.

Table 2: Speaker verification and checksum comparison results for the 2-word SAS experiment.

Speaker →	Original		Morphed	Different	Original		
SAS →	Matching		Matching	Matching	Position of the Mismatched Word:		
					1st	2nd	1st & 2nd
FNR	22.31%	FPR	43.55%	40.00%	30.00%	29.23%	25.96%
Time(s) mean	3.05	Time (s) mean	3.05	2.86	2.82	2.97	3.49
(std dev)	(1.93)	(std dev)	(1.36)	(1.16)	(1.44)	(1.55)	(2.42)
Replay Rate	3.94%	Replay Rate	2.69%	2.31%	4.62%	4.23%	2.88%

Table 3: Speaker verification and checksum comparison result for the 4-word SAS experiment.

Speaker →	Original		Morphed	Different	Original			
SAS →	Matching		Matching	Matching	Not Matching in:			
					1 word	2 words	3 words	4 words
FNR	25.00%	FPR	38.92%	37.69%	51.51%	44.32%	32.20%	31.82%
Time(s) mean	3.85	Time(s) mean	3.50	3.64	3.97	3.82	4.07	3.98
(std. dev)	(1.83)	(std dev)	(1.70)	(1.68)	(2.08)	(2.33)	(2.34)	(3.43)
Replay Rate	7.77%	Replay Rate	5.02%	5.40%	4.69%	4.69%	4.30%	7.03%

Table 1: Demographic information of users participating in the 2-word and 4-word SAS experiments.

	2-word	4-word
	N = 64	N = 64
Gender		
Male	58%	76%
Female	42%	24%
Age		
18-24 years	28%	46%
25-34 years	40%	25%
35-44 years	22%	13%
45-54 years	9%	11%
55-64 years	1%	5%
Education		
High school graduate or diploma	12%	32%
Some college credit, no degree	29%	16%
Bachelor's degree	35%	37%
Master's degree	22%	14 %
Doctorate degree	2%	2%
English as First Language		
Yes	94%	92%
No	6%	8%
Hearing Impairment		
No	97%	97%
Yes	3%	3%
Voice App Usage Frequency		
Daily	54%	59%
Sometimes	42%	37%
Never	4%	4%

any statistically significant differences in the FNR corresponding to different speakers in the first challenge.

Table 4: Speaker Verification result for each individual speaker in the 2-word SAS experiment.

	Original FNR	Morphed FPR	Different FPR
Male 1	17.69%	32.69%	27.69%
Male 2	22.31%	49.23%	31.54%
Female 1	26.92%	45.38%	48.08%
Female 2	22.31%	43.08%	52.69%

To measure users' performance in the second scenario (Original Voice; Mismatching SAS), we placed incorrect word(s) in the first and/or second word of a 2-word SAS. The last three columns of Table 2 summarize the results for this scenario. We see that the average FPR of accepting an incorrect SAS is around 30%, which may be considered high for applications demanding high security.

Users detected the checksums with two incorrect words slightly better than those with one incorrect word, which means the number of incorrect words in a SAS reduces the FPR. However, the Friedman test did not report any significant changes in the error rates with the change in the number or position of the incorrect word (e.g., whether the incorrect word is the first or the second word of the checksum). The average time to answer these set of checksum comparison questions is around 3.09 seconds, and the average rate of replaying the samples (replay rate: RR) while performing the comparison was around 3%. This low rate shows that the users do not generally show interest towards requesting a replay.

Through our third category of challenges (Different Voice; Matching SAS), we look at the instances of accepting a different speaker's voice. In a perfect world, the assumption of Crypto Phones is that the users can distinguish a "different voice" 100% of the times. However, the results of our study, presented in the third column of Table 2, show that, on an average, the FPR is 40%. Moreover, this result depends on the similarity between the attacker's voice and the victim's voice. It varies, from 53% for the two female voices which are more similar to each other, to around 28% for the two male voices which are less similar. Column three of Table 4 shows the FPRs corresponding to each speaker's voice. Friedman test did not report any significance in comparing FPRs of the speakers with one another.

In our final set of challenges (Morphed Voice; Matching SAS), the matching SAS is played-back in a morphed voice. The average FPR across all users and all speakers is around 43%, as shown in the column two of Table 2. Column two of Table 4 shows FPR rates for morphing attack on different speaker's voice. The FPR varies between 32% for the conversion from male 2 to male 1 voice to 49% for the conversion from male 1 to male 2 voice. Similarity between the attacker's and the victim's voices before the conversion highly affects quality of the conversion and as a result the related FPR. For voices that are more similar, such as the two female voices in our study, FPR of the different voice attack is better than the FPR of the morphed voice. Ideally if the attacker can mimic the victim, the attacked voice would sound more natural and more acceptable by human users. However, if the attacker's voice and the victim's voice are not similar to each other, such as in the case of the two male speakers in our study, the attacker can take advantage of voice morphing tools to construct a voice that mimics the victim's voice.

Friedman test was conducted to compare the FPR result among multiple speakers, and it rendered a Chi-square value of 0.002, which was significant. Further, Wilcoxon signed-rank test[5], conducted using Bonferroni adjusted alpha levels of 0.0125 per test (0.05/4), showed statistical significance with p-values of 0, 0.001 and 0.001 for the comparison between male 1 and other speakers,

[5]Wilcoxon signed-rank test is a non-parametric test suitable for comparing matched samples which may not be normally distributed. All results of statistical significance in this paper are reported at a 95% confidence level.

with small effect size for male 2 (r = 0.225) and trivial effect size for female 1 and female 2 (r = 0.072 and r = 0.030, respectively). However, the Wilcoxon signed-rank test did not report any statistical significance when comparing FPRs for other set of speakers with one another. This shows that different voices and conversions might result in a better, or worse, attack success rate.

The result of the Wilcoxon signed-rank test to compare the two voice imitation attacks averaged over multiple speakers (different speaker voice vs. morphed voice) was not statistically significant.

Based on the Wilcoxon signed-rank test, we did not find any statistical significance in the replay rate and the time taken to complete different type of challenges in the 2-word SAS.

4.2.2 4-word SAS Checksum

The error rates for the 4-word SAS experiment follows the same pattern as 2-word SAS, as summarized in Table 3. FNR for rejecting a benign setting (Original Voice; Matching SAS) is 25% (column one of Table 3). FPR under the voice imitation attacks (Morphed/Different Voice; Matching SAS) is 38.92% and 37.69%, respectively, varying among different speakers as shown in Table 5. The results re-iterate that if the voice of the two speakers is more similar before the conversion, the attacker can simply insert his/her own voice into the SAS conversation. However, if the two voices differ (such as in the case of our two male speakers), the attacker can benefit from voice conversion.

Table 5: Speaker Verification result for each individual speaker in the 4-word SAS experiment.

	Original FNR	Morphed FPR	Different FPR
Male 1	23.11%	36.74%	27.65%
Male 2	19.32%	40.53%	28.03%
Female 1	25.76%	40.15%	42.80%
Female 2	31.82%	38.26%	52.27%

The Friedman and Wilcoxon signed-rank test report on statistical significance in 4-word SAS experiment follows the same pattern as the 2-word SAS experiment. That is, the statistically significant result was just reported for FPRs of the morphing attack when comparing male 1 speaker with other speakers.

We displayed one to four incorrect words in a 4-word SAS, and compared the FPRs (Original Voice; Mismatching SAS). The last four columns of Table 3 summarize the results for this scenario. They show that the overall average FPR for accepting an incorrect SAS is around 40% in a 4-word SAS. Similar to the 2-word SAS experiment, users detected checksums with higher number of incorrect words slightly better than those with lesser number of incorrect words. Friedman test was conducted to compare the FPR for different number of incorrect words in SAS, and it rendered a Chi-square value of 0, which was significant. Further Wilcoxon signed-rank test, conducted using Bonferroni adjusted alpha levels of 0.0125 per test (0.05/4), showed statistical significance with p-values of 0.001, 0.007 when comparing 1 to 4, and 2 to 4, incorrect words in a 4-word SAS, with small effect sizes (r = 0.273 and r = 0.172, respectively), while other comparisons were not significant.

The average time to answer the SAS mismatch questions is around 3.96 seconds, and on average the replay rate was only around 5%. Similar to 2-word SAS, we did not find any statistically significant differences in the replay rate and the time taken to complete different type of challenges in the 4-word SAS.

4.2.3 2-word vs. 4-word SAS Performance

Theoretically, the security of Crypto Phones should increase (exponentially) with increase in the size of the SAS. This is based on the assumption that human users' accuracy in checksum comparison and speaker verification tasks is perfect. However, in real life, users make mistakes in these tasks as our results presented so far confirm. Intuitively, it is expected that people perform better in the speaker verification task when faced with a longer SAS checksum. This is because when they are presented with longer speech, they may obtain more/better features of the speaker's voice. This perspective is in line with prior literature in linguistics research [22] which shows that people can recognize familiar voice samples with a better accuracy when the duration of the sample is longer. In this case, the security and usability of the system should increase, since people can recognize the original and attacked samples with a better accuracy. On the other hand, when the size of the SAS checksum increases, people need to match a larger number of words, which intuitively should increase the possibility of making a mistake in the checksum comparison task.

Our results for the 2-word and 4-word experiments show the above effects for the speaker verification and checksum comparison tasks. With reference to Tables 2 and 3, we see that while the FPR in detecting morphed voice is around 43% in a 2-word SAS, it is decreased to about 39% in a 4-word SAS. A similar effect is seen for the different voice attack (about 40% for 2-word vs. about 38% for 4-word). This shows that users are slightly more successful in detecting the morphed voice when the SAS becomes longer. The result of the Mann-Whitney U test[6], was not statistically significant when comparing FPRs of morphing attack in the 2-word and 4-word. However, users are less successful in the checksum comparison as we move from a 2-word to a 4-word SAS. This is confirmed via the Mann-Whitney U test, which shows statistical significance yielding a p-value of 0.0002. Therefore, the overall FPR – averaged over the speaker verification and checksum comparison tasks – increases when transitioning from 2-word to 4-word SAS. Although we theoretically expected the length of the SAS to have a positive impact on the security, in fact it decreases the security since the users make more errors in detecting the mismatch.

The average time to answer a 2-word and a 4-word SAS checksum is 3.04 seconds and 3.83 seconds, respectively. This difference is because of the longer speech duration in the 4-word SAS. Mann-Whitney U test, comparing these average timings, shows statistical significance with a p-value of 0.0244. However, in real life, this difference may not matter much as both scenarios take less than 5 seconds for the SAS validation task. We do not notice a significant difference in the replay rate for the two SAS sizes, although on average the number of replays in the 4-word SAS is slightly more.

4.3 SUS Feedback

The usability of a system involves several aspects such as effectiveness and efficiency of the system, and users' experience with, and satisfaction of, the system. In general, usability shows how much effort and time users should expend to achieve their desired objectives and have a satisfactory experience. Based on this definition, a simple questionnaire, called System Usability Scale (SUS) [18], was designed to measure the usability of an engineered system. SUS is a 5-point Likert scale consisting of ten questions, each with 5 possible answers (1 represents strong disagreement and 5 represents strong agreement), covering various aspects of the usability of the system, such as the need for support and training, and system complexity. SUS score is calculated between 0 and 100, and a higher score means better usability.

At the end of our study, each participant filled out the SUS questionnaire, rating his/her perception of the experiment underlying our web-based VoIP system. The average SUS score for the 2-word

[6]Mann Whitney U test is a non-parametric test suitable for data which may not be normally distributed.

SAS was 72.23 (std dev = 18.04), and the average SUS score for the 4-word SAS was 75.04 (std dev = 19.52). Considering that industry averages for SUS scores tend to hover in the 60–70 range [26], our results show that users found both systems to be usable. The standard deviation in our study shows that the SUS scores almost fall between 55 and 95 which is generally considered to be "good" [17]. The result of comparing 2-word and 4-word SUS was not statistically significant based on the Mann-Whitney U test.

5. DISCUSSION

5.1 Summary and Key Insights

Our study provides several insights into the security and usability of Crypto Phones. The first insight pertains to the implication of SAS sizes. Although, theoretically, a longer SAS should result in higher security ($1/2^{32}$ versus $1/2^{16}$ probability of the attack success for 2-word and 4-word SAS), our evaluation shows that in real-life, human errors in recognizing the speaker and comparing the checksums translate into a much lower security. Although longer SAS slightly decreased the error rate (FPR) in the speaker verification task, it significantly increased the error rates in the checksum comparison task. Therefore, the overall security of a 4-word SAS actually decreased compared to a 2-word SAS. Moreover, the time taken to validate the 4-word SAS is longer, which might negatively affect the usability of the system.

Our second key result relates to the difference between the two types of voice imitation attacks (morphing attack and different speaker attack). Although, on average, the morphing attack is only somewhat better than the different speaker attack, the result varies significantly among multiple speakers. The attacker, whose voice is more similar to the victim's voice, has a better chance in performing the voice imitation attack using his/her own voice than using voice conversion tools. The reason is that, in spite of all the advancement in speech synthesis, the flow and naturalness in the "human voice" is still better than that produced by the machines. In our experiment, the FPRs of the morphed voice and the different speaker voice for the two male speakers are in line with the ones reported in [30] (i.e., FPR is lower for the different speaker attack compared to the morphing attack).

Third, the original speakers' voices were shown to be fairly accurately recognizable by the users (as demonstrated by our relatively low FNRs). However, we should mention that we picked all the recordings of the same speaker in a single session (with the same ambient noise and volume) for the familiarization clip as well as for the SAS challenges. The idea behind this selection was the that, if the end parties do not know each other before the call to authenticate the other party (such as a customer service call, like in our scenario where amazon Mechanical Turk users called the IVR), they should rely only on the current session and ensure that the voice that speaks the SAS checksum is the same voice that speaks the rest of the conversation. The users who are already familiar with each other prior to the call might perform better in real life.

Fourth, from a qualitative perspective, the SUS responses show that both 2-word and 4-word SAS checksums are both generally acceptable and usable. Our participants found the systems to be easy to use and less complex, and did not feel they would require high level of training and support in order to use them in practice. The user-friendly GUI, clearly written and spoken instructions, and reliable voice and web servers, as well as the demographics of the subjects may have contributed to this high usability. The demographic information shows that our subjects were mostly young and well-educated, and had prior experience with VoIP applications. Other users (older, less educated, or those with less expertise in VoIP application) might find the system less or more usable.

Although we evaluated Crypto Phones SAS validation process on a unidirectional channel, our results extend to the bidirectional case. In a bidirectional setting, both parties need to confirm the SAS checksums, hence the security of the system would actually increase, while the usability of the system will decrease. If we assume that the probability of the attack failure in one direction is 60% (a realistic number based on our evaluation), then the probability that the system may defeat the attack in both directions is 84%. However, based on the current study, around 75% of the legitimate calls were accepted, and, therefore, if we want both sides to accept the connection, only around 56% of the valid sessions will be accepted. Recall that the more frequently people reject valid calls, the more frequently they need to redial, which may further reduce the usability level directly, and the security level indirectly. Although a two-sided attack in a bidirectional case is less successful, note that the attacker can still compromise only one side for a successful attack. In this case, the probability of attacking either Alice or Bob is increased to 64%. After establishing the connection (with Bob for example), the attacker might continue with voice conversation (by creating arbitrarily long speech in Alice's voice using voice morphing techniques [27]), or insert text (in Crypto Phones text communications) on behalf of Alice over the established channel.

The results show that while out-of-band device pairing is somehow successful in a face to face setting (around 15% failure as reported in [23]), SAS validation is more challenging in Crypto Phones as "remote" users should compare the checksum over the same VoIP channel. Also, the results show that in our setting combining the two task of checksum comparison and speaker verification (which is similar to a practical Crypto Phone application) makes the verification slightly more difficult compared to [30] where the only task of the participants is speaker verification.

5.2 Potential Limitations

The popularity and ease of use of the web-based applications might have given our setup a high usability. We have not studied other forms of Crypto Phones applications, but we believe that their security and usability should be similar to the web-based application. Unless the user performs a "hands-free" call, she may not be able to easily compare the SAS on her phone's display with the one spoken by the other party. This in practice might reduce the security (increase the possibility of making errors) and usability (looking at the screen and holding the phone next to the ear might be difficult). One possible limitation of our work relates to the use of IVR rather than a human user. One real-life application of IVR in Crypto Phones is support or service centers in which the customer calls a number, is responded by IVR machines and might be asked to validate a SAS in order to secure the communication. Apart from this application, we believe using natural human voices as IVR commands (rather than mechanical text-to-speech voices) may address this limitation of our work.

Similar to other studies which recruit remote participants, we could not fully monitor the authenticity of the responses. We did not include dummy questions and trusted the subjects that their answers were honestly provided as per the given instructions. We did discard a few instances where the participants clicked on the same answer for all type of challenges and SUS questions, and collected new data to replace such participants. Also, we noticed that average time to complete the study is more or less similar among all participants, which could be a sign of validity of the responses. Nevertheless, the high number of subjects involved in our study still help us to ignore possible "random" or "dishonest" answers, and may provide a high confidence in the final aggregated results.

5.3 Potential Defenses and Future Work

Improving *both* the security and usability of Crypto Phones seems like a challenging endeavor. One natural goal should be to improve the users' performance in the checksum comparison task. In our study, we used the Compare-Confirm approach for checksum comparisons, given that this is the most popular approach currently employed in Crypto Phones applications. An alternative approach is Copy-Confirm [32]. The Copy-Confirm method may reduce FPR (enhance security) but may increase FNR (degrade usability) [32] especially for word-based SAS (as opposed to numeric SAS) because typing the words may be more error-prone.

Another direction to improve the checksum comparison task performance might be to make the task more engaging for the users. The work presented in [21] explored a scoring-enhanced comparison approach and showed that it improved the security, and usability in the context of device pairing application. Further investigation is necessary to evaluate this system in a remote VoIP setting.

Since the attacks against Crypto Phones are largely a consequence of human errors, it seems reasonable to provide sufficient instructions and training to those who use these applications. The users who are trained and aware of the risks associated with the leakage of their sensitive information might protect themselves better against such attacks. However, administering such training might pose a significant challenge in practice.

Alternatively, the human reliance in the Crypto Phones applications could be reduced. As suggested in [30], the task of speaker verification could be performed automatically using a voice biometrics system. However, current voice biometrics systems may not offer viable usability and security especially when working with *short* spoken words. Further research is necessary to evaluate such automated solutions in the context of Crypto Phones.

6. CONCLUSIONS

In this paper, we comprehensively evaluated the security and usability of mobile Crypto Phones, a popular decentralized approach to establish end-to-end VoIP security. These apps are being rapidly deployed in many real-world personal and business settings in a hope to protect users from the prying eyes of eavesdroppers. On the negative side, the results of our study suggest that – because of the human errors associated with the checksum comparison and speaker verification tasks – the security offered by Crypto Phones falls significantly short of the theoretical guarantees provided by the underlying cryptographic protocols. For instance, for a 2-word (16-bit) checksum, the attacker would succeed with a probability about 30% in practice as shown by our study, while the theory suggests that the attacker cannot do better than 0.0015% – a security degradation by a factor of 20,000. Moreover, while the theory guarantees that increasing the checksum size, from 2-word to 4-word, will increase the security exponentially, by a factor of 65536 (2^{16}), we saw that the attacker success probability increases (from about 30% to 40%). This situation emerged because, as the checksums became longer, validating speakers became slightly easier (as users could get more cues to identify a malicious voice), but at the same time comparing checksums became much harder.

On the promising side, the relatively positive user perception (SUS scores over 70 out of 100) bode well for the usability and acceptability of Crypto Phones. However, a 75% of accuracy under the benign setting does not show high performance of the users.

Acknowledgments

This work has been supported in part by a grant from Cisco Systems, entitled "Establishing Peer-to-Peer Secure VoIP Connections", 2013-2016. The authors thank Zachary Peterson (our shepherd) and ACSAC 2015 anonymous reviewers for their helpful feedback and guidance.

7. REFERENCES

[1] Infosecurity - Microsoft Expands Encryption to Foil Government Snooping. http://goo.gl/lOZuik.
[2] NSA and All Major Intelligence Agencies Can Listen in to Encrypted Cell Phone Calls. http://goo.gl/rs3UWO.
[3] TRANSFORM: Flexible Voice Synthesis Through Articulatory Voice Transformation . http://goo.gl/ZrRtXG.
[4] AT&T Natural Voices,. http://goo.gl/Nwz3Q2.
[5] Fact Sheet 9: Wiretapping and Eavesdropping on Telephone Calls. https://goo.gl/w94ciL.
[6] FreeSWITCH. https://freeswitch.org.
[7] London newspaper wiretapped royals. http://goo.gl/jTm04H.
[8] ModelTalker Speech Synthesis System. http://www.modeltalker.com.
[9] Open Whisper Systems. https://whispersystems.org/.
[10] Paranoid much? Demand for secure CryptoPhone is so high. http://goo.gl/r2zxnQ.
[11] Silent Circle – Private Communications. https://silentcircle.com/.
[12] The Zfone Project. http://zfoneproject.com/.
[13] Voxal Voice Changer. http://goo.gl/Ets8SO.
[14] World's first HTML5 SIP client. http://sipml5.org.
[15] ZORG - An Implementation of the ZRTP Protocol. http://www.zrtp.org/.
[16] Legal authorities supporting the activities of the national security agency described by the president. Technical report, U.S. Department of Justice, 2006.
[17] A. Bangor, P. Kortum, and J. Miller. Determining what individual sus scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3), 2009.
[18] J. Brooke. Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 1996.
[19] J. P. Campbell Jr. Speaker Recognition: A Tutorial. *Proceedings of the IEEE*, 85(9), 1997.
[20] J. Dénes and A. Keedwell. *Latin squares and their applications*. London, 1974.
[21] A. Gallego, N. Saxena, and J. Voris. Exploring extrinsic motivation for better security: A usability study of scoring-enhanced device pairing. In *Financial Cryptography and Data Security*. 2013.
[22] H. Hollien, W. Majewski, and E. T. Doherty. Perceptual Identification of Voices Under Normal, Stress and Disguise Speaking Conditions. *Journal of Phonetics*, 1982.
[23] R. Kainda, I. Flechais, and A. W. Roscoe. Usability and Security of Out-Of-Band Channels in Secure Device Pairing Protocols. In *SOUPS: Symposium on Usable Privacy and Security*, 2009.
[24] C. Kuo, J. Walker, and A. Perrig. Low-cost manufacturing, usability, and security: An analysis of bluetooth simple pairing and wi-fi protected setup. In *Workshop on Usable Security, (USEC)*, 2007.
[25] P. Ladefoged and J. Ladefoged. The Ability of Listeners to Identify Voices. *UCLA Working Papers in Phonetics*, 1980.
[26] J. R. Lewis. Ibm computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. *Int. J. Hum.-Comput. Interact.*, 7(1), 1995.
[27] D. Mukhopadhyay, M. Shirvanian, and N. Saxena. All your voices are belong to us: Stealing voices to fool humans and machines. In *ESORICS*. 2015.
[28] S. Pasini and S. Vaudenay. An Optimal Non-Interactive Message Authentication Protocol. In *CT-RSA*, 2006.
[29] P. Rose. *Forensic Speaker Identification*. CRC Press, 2003.
[30] M. Shirvanian and N. Saxena. Wiretapping via mimicry: Short voice imitation man-in-the-middle attacks on crypto phones. In *ACM conference on Computer and communications security*, 2014.
[31] T. Toda, A. W. Black, and K. Tokuda. Voice Conversion Based on Maximum-Likelihood Estimation of Spectral Parameter Trajectory. *Audio, Speech, and Language Processing, IEEE Trans.*, 2007.
[32] E. Uzun, K. Karvonen, and N. Asokan. Usability analysis of secure pairing methods. In *Financial Cryptography & Data Security*. 2007.
[33] S. Vaudenay. Secure Communications over Insecure Channels Based on Short Authenticated Strings. In *CRYPTO*, 2005.
[34] R. Zhang, X. Wang, R. Farley, X. Yang, and X. Jiang. On The Feasibility of Launching the Man-in-the-Middle Attacks on VoIP from Remote Attackers. In *ASIACCS*, 2009.