Wiretapping via Mimicry: Short Voice Imitation Man-in-the-Middle Attacks on Crypto Phones

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

ABSTRACT

Establishing secure voice, video and text over Internet (VoIP) communications is a crucial task necessary to prevent eavesdropping and man-in-the-middle attacks. The traditional means of secure session establishment (e.g., those relying upon PKI or KDC) require a dedicated infrastructure and may impose unwanted trust onto third-parties. "Crypto Phones" (popular instances such as PGPfone and Zfone), in contrast, provide a purely peer-to-peer user-centric secure mechanism claiming to completely address the problem of wiretapping. The secure association mechanism in Crypto Phones is based on cryptographic protocols employing Short Authenticated Strings (SAS) validated by end users over the voice medium.

The security of Crypto Phones crucially relies on the assumption that the voice channel, over which SAS is validated by the users, provides the properties of integrity and source authentication. In this paper, we challenge this assumption, and report on *automated SAS voice imitation man-in-the-middle attacks* that can compromise the security of Crypto Phones in both two-party and multi-party settings, even if users pay due diligence. The first attack, called the *short voice reordering attack*, builds arbitrary SAS strings in a victim's voice by reordering previously eavesdropped SAS strings spoken by the victim. The second attack, called the *short voice morphing attack*, builds arbitrary SAS strings in a victim's voice from a *few* previously eavesdropped sentences (less than 3 minutes) spoken by the victim. We design and implement our attacks using off-the-shelf speech recognition/synthesis tools, and comprehensively evaluate them with respect to both manual detection (via a user study with 30 participants) and automated detection. The results demonstrate the effectiveness of our attacks against three prominent forms of SAS encodings: *numbers*, *PGP word lists* and *Madlib* sentences. These attacks can be used by a wiretapper to compromise the confidentiality and privacy of Crypto Phones voice, video and text communications (plus authenticity in case of text conversations).

CCS'14, November 3–7, 2014, Scottsdale, Arizona, USA.

http://dx.doi.org/10.1145/2660267.2660274.

Categories and Subject Descriptors

K.4.1 [**Computer and Society**]: Public Policy Issues—*Privacy*; D.4.6 [**Operating System**]: Security and Protection—*Authentication*

General Terms

Security and privacy, Human-centered computing

Keywords

VoIP; Man-in-the-Middle Attack; Wiretapping; Authentication; Short Authenticated Strings

1. INTRODUCTION

Voice, video and text over IP (VoIP) systems are booming and becoming one of the most popular means of communication over the Internet. Today, VoIP is a prominent communication medium used on a variety of devices including traditional computers, mobile devices and residential phones, enabled by applications and services such as Skype, Hangout, and Vonage, to name a few.

Given the open nature of the Internet architecture, unlike the traditional PSTN (public-switched telephone network), a natural concern with respect to VoIP is the security of underlying communications. This is a serious concern not only in the personal space but also in the industrial space, where a company's confidential and sensitive information might be at stake. Attackers sniffing VoIP conversations for fun and profit (e.g., to learn credit card numbers, account numbers and passwords) as well as wiretapping and surveillance of communications by the government agencies [1, 3] are well-recognized threats. Prior research also shows the feasibility of launching VoIP man-in-the-middle (MITM) attacks [54], which can allow for VoIP traffic sniffing, hijacking or tampering.

In light of these threats, establishing secure – authenticated and confidential – VoIP communications becomes a fundamental task necessary to prevent eavesdropping and MITM attacks. To bootstrap end to end secure communication sessions, the end parties need to agree upon shared authenticated cryptographic (session) keys. This key agreement process should itself be secure against an MITM attacker. However, the traditional means of establishing shared keys, such as those relying upon a Public Key Infrastructure (PKI) or Key Distribution Center (KDC), require a dedicated infrastructure and may impose unwanted trust onto third-party entities. Such centralized infrastructure and third-party services might be difficult to manage and use in practice, and may themselves get compromised or be under the coercion of law-enforcement agencies, thereby undermining end to end security guarantees.

In this paper, our central focus is on "Crypto Phones" (Cfones), a decentralized approach to securing VoIP communications. Cfones promise to offer a purely peer-to-peer user-centric mechanism for

establishing secure VoIP connections. A prominent real-world instance of a Cfone is Zfone [8, 10], invented by Phil Zimmermann, now being offered as a commercial product by Silent Circle [9]. A Cfone involves executing a SAS (Short Authenticated Strings) key exchange protocol, such as [7, 11, 51], between the end parties. The SAS protocol outputs a short (e.g., 20-bit) string per party — if the MITM adversary attempted to attack the protocol (e.g., inserted its own public key or random nonces), the two strings will *not match*. These strings are then output, e.g., encoded into numbers or words [10], to users' devices who then *verbally* exchange and compare each other's SAS values, and accordingly accept, or reject the secure association attempt (i.e., detect the presence of MITM attack). Figure 1 depicts a traditional MITM attack scenario against Cfone.

The security of Cfones crucially relies on the assumption that the human voice channel, over which SAS values are communicated and validated by the users (Alice and Bob), provides the properties of integrity and source authentication. In other words, it is assumed that the attacker (Mallory) is not able to insert a new desired SAS value in Alice's and/or Bob's voice.

In this paper, we systematically investigate the validity of this assumption. Our hypothesis is that, although impersonating someone's voice in face-to-face arbitrarily long conversations can be significantly challenging, impersonating *short* voices (saying short and random SAS strings) in a *remote* VoIP setting may not be. Indeed, we undermine Cfones' security assumption underlying SAS validation, and report on SAS voice imitation MITM attacks that can compromise the security of Cfones in both two-party (2-Cfone) and multi-party or conferencing (n-Cfone) settings, even if users were asked to pay due diligence. Figure 2 depicts an example scenario for our short voice imitation MITM attacks against 2-Cfone.

Our Contributions: We make the following contributions:

1. Generalization and Formalization of Cfones: The secure connection establishment problem considered by Cfones, and the underlying solution approach, bear a close resemblance to the domain of "proximity-based device pairing". Based on this parallel and wealth of prior work in device pairing, we provide a generalization and semi-formalization of Cfones, considering C-fones in both two-party and multi-party settings and adopting prior device pairing methods in the context of Cfones (Section 2).

2. Voice-Centric MITM Attacks Against Cfones: We present two types of short voice imitation MITM attacks against Cfones (Section 3). The first attack, called the *short voice reordering attack*, builds arbitrary SAS strings in a victim's voice by reordering previously eavesdropped SAS strings spoken by the victim. The second attack, called the *short voice morphing attack*, builds arbitrary SAS strings in a victim's voice from a *few* previously eavesdropped sentences spoken by the victim.

3. Attack Design, Implementation and Evaluation: We design and implement our reordering and morphing attacks using publicly available, off-the-shelf speech recognition and synthesis tools (Sections 4). Next, we comprehensively evaluate our attack system with respect to both manual detection, via a *user study* with 30 participants, and automated detection (Section 5). The results demonstrate the effectiveness of our attacks against three prominently used SAS encodings: *numbers*, *PGP word lists* [10] and *Madlib sentences* [22]. These attacks can be used by a wiretapper to completely compromise the confidentiality and privacy of Cfones communications (plus authenticity in case of Cfones text conversations). Our objective evaluation shows that the shorter the SAS string, the harder it is for the user to detect voice impersonation. Our subjective evaluation shows that people can distinguish

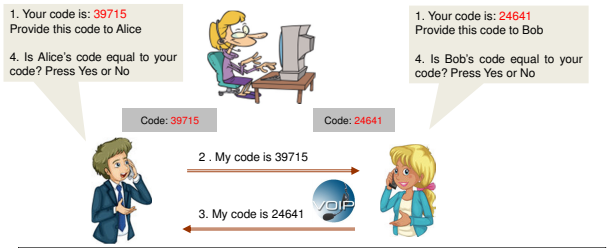

Figure 1: A traditional MITM attack scenario for 2-Cfone – attack is detected since SAS values do not match

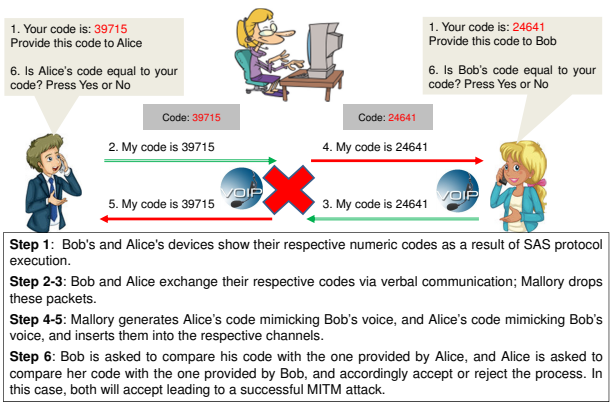

Figure 2: Our short voice imitation MITM attack scenario for 2-Cfone – attack succeeds because of voice impersonation

a *different* voice from a familiar voice with about 80% success. However, they are not as successful in detecting our reordering and morphing attacks. Moreover, we believe that in real-life, attack detection would be even more difficult due to the presence of the ambient or channel noise, and the fact that SAS validation is only a secondary user task (the primary task is establishing the call).

Cfones versus "Device Pairing": Device pairing is the process of establishing secure connection between two (or more) wireless devices communicating over a short-range channel, such as WiFi or Bluetooth. A wealth of prior work exists that uses SAS protocols and different out-of-band (OOB) channels for the purpose of device pairing [25, 30]. Device pairing involves devices and their users who are *physically nearby*. In contrast, Cfones involve devices and users which are *remote*, communicating over an open Internet channel. However, both systems assume that the SAS transfer or OOB channel provides integrity and source authentication. While this is a valid assumption in the context of physically co-located devices/users (pairing), it may not be the case for the remote VoIP setting (Cfones), since the data transmission still happens over an open *insecure* channel, not over proximity communication. This important aspect is what our work focuses on in this paper.

2. BACKGROUND & FORMALIZATION

2.1 Communication and Threat Model

A 2-Cfone SAS protocol between Alice and Bob is based upon the following communication and adversarial model, adopted from [51]. The devices being associated are connected via a remote,

point-to-point high-bandwidth bidirectional VoIP channel. An MITM adversary Mallory attacking the Cfone 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 can not 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. The latter assumption is what we are analyzing and challenging in this paper.

This approach and model can be easily extended to the VoIP group communication or conferencing scenarios (n-Cfones). Here, more than two remote participants form a group and all data (messages, video and audio) is broadcast among these participants. The same assumptions are made over this broadcast channel as the point-to-point channel in 2-Cfone. In addition, n-Cfone protocols require the participants to verbally validate the count of the group members. If undercounting happens, the attacker can simply insert itself into the group and eavesdrop over all conversation [34].

2.2 SAS Protocols

A number of SAS protocols exist [17, 32, 35, 39, 51] in the literature that a Cfone implementation may adopt. It is an authenticated key exchange protocol which allows Alice and Bob to agree upon a shared authenticated session key based on SAS validation over an auxiliary channel (such as voice channel). The protocol results in a short (e.g., 20-bit) string per party – matching strings imply successful secure association, whereas non-matching strings imply a MITM attack. These protocols limit the attack 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) using the session key. The data may include the voice, text or video data. In fact, a Cfone texting application can utilize the SAS approach to secure the text channel by means of SAS validation over the voice channel, as employed by Silent Circle [9].

SAS protocols have been extended to the group setting [33, 50], and can be utilized in the context of n-Cfones. The idea is the same: upon executing the group SAS protocol, each device outputs a SAS value; matching SAS values on all devices imply successful association whereas non-matching values indicate the presence of an attack. In addition to requiring comparison of SAS values, an n-Cfone involves the user(s) to correctly count the number of group members (i.e., the number n) taking part in the conference.

2.3 SAS Validation Mechanisms

Two-Party Setting: We consider following different 2-Cfone methods derived from [49], for associating two remote devices A and B, which encode the SAS data into decimal digits [49], PGP words [10] or Madlib phrases (grammatically correct Madlib sentences) [22]:

1. *Compare-Confirm*: A and B display SAS encoded number, PGP words, or Madlib phrase, each on their respective screens. Alice compares the number, PGP words or phrase displayed on A with the number displayed on B via verbal exchanges with Bob. Based on the comparison, both Alice and Bob accept or reject the secure association attempt.
2. *Copy-Confirm*: A displays a SAS encoded number on its screen. Alice verbally provides the number to Bob who inputs it onto B. B indicates the result of association (match or mismatch) on its screen. Bob indicates the same result to Alice verbally. Alice accepts or rejects the association process on A accordingly.

Multi-Party Setting: In case of n-Cfones, some SAS validation methods involve a centralized group member, called a *leader*, while others are *peer-based* (as discussed in the context of proximity group association [34]). For the latter, a circular topology is assumed among the group members. Recall that, in addition to comparing SAS values, n-Cfone requires the participants to correctly determine the group size. The leader, who knows the group size, will either input this number to its own device as well as announce it to others so they can enter to their respective devices, or the leader will compare the count with the one displayed by its device and announces the count to others so they can also compare with the value displayed by their respective devices. The SAS values can be validated in a leader-driven or a peer-to-peer fashion, by comparing or by copying (in case of numbers). Below is a list of methods (derived from [34]) suitable for n-Cfones.

1. *Leader-VerifySAS:* After the group size has been validated, the leader's device displays the SAS value encoded into a number, PGP words or phrase and the leader announces it to the group. Other members' devices display respective SAS values. Each member compares their respective SAS values with that announced by the leader. If SAS values do not match, a member aborts the process on its device and asks everyone else to do the same. If no one identifies an error, each member accepts group association on its device.
2. *Leader-CopySAS:* After the group size has been validated, the leader's device displays the SAS value encoded as a number and the leader announces it to the group. Other members input the announced SAS value into their devices. If the devices indicate failure (SAS value mismatch) they abort the process and warn others to do the same. Otherwise, everyone accepts.
3. *Peer-VerifySAS:* After the group size has been validated, each device displays a numeric SAS value and each member compares its SAS value with that of their neighbor on the right (predefined via a virtual circular topology). In case of a mismatch, a member aborts the process and instructs others to do the same. Otherwise, everyone accepts.

3. ATTACK OVERVIEW & BACKGROUND

We discuss why recognition of the identity of a speaker (especially from short speech) can be a complex task for human users, and provide an overview of our Cfone voice imitation attacks that exploit this inherent limitation of the human cognitive system.

3.1 Manual Speaker Recognition Limitations

In an MITM attack against the SAS protocol of a Cfone, Mallory can insert herself into a session and gain full access to the data being transferred between the Alice and Bob. To do so, Mallory needs to hijack the session and impersonate each party. As discussed in Section 2.1, Cfone's security assumption is that although Mallory has full control over the communication channel, it cannot insert voice messages that mimic Alice/Bob. Should this hypothesis be valid, the SAS value which is verbally exchanged on this channel can always authenticate Alice and Bob, foiling the MITM attack. A Cfone MITM attack seems relatively straight-forward against data communication (i.e., non verbal communication messages of the SAS protocol) [54], however, it is assumed that voice is unique to each individual, and therefore it is impossible to impersonate it. This assumption relies on special characteristic of speech which appears to make it difficult to impersonate at first glance.

Speech construction is a complex area. In simple terms, speech consists of words, each of which is a combination of speech sound units (phones). However, in reality, human voice is not as simple as this definition. Voice signal created at the vocal folds travels and gets filtered through vocal tract to produce vowels and consonants.

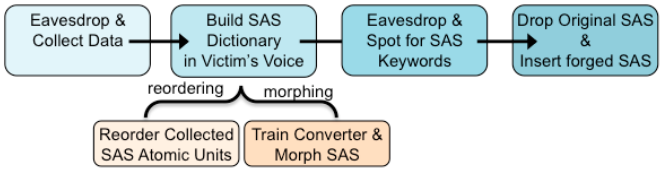

Figure 3: High-level diagram of the attack

Human body structure, vocal folds, articulators and human physiology and the style of speech provide each individual a potentially distinguished voice characteristic. Pitch, timbre and tone of speech are some of the features that may make a voice unique (for further information, we refer the reader to [15]). Therefore, the assumption that voice is unique, just like fingerprint or iris, does have some validity (although how much is a question explored in this paper).

Speech perception and recognition, the tasks that Cfone users have to perform while validating the SAS values, are even more complex than speech construction. There exists considerable literature on how speech is recognized [18, 19, 38]. Linguistics researchers have conducted various experiments and analyzed the capabilities of human speech recognition over different parameters, such as length of the samples, number of different samples, samples from familiar vs. famous people, and combinations thereof [38]. In an experiment, conducted in [31], the participants were asked to identify a voice when the sample string presented to them was "hello", which resulted in a correct recognition rate of only 31%. However, when a full sentence was presented to the participants, the recognition rate increased to 66%. In the study of [23], a 2.5 minute long passage was presented as a sample to the participants, resulting in the average recognition accuracy of 98%. Many other experiments have been performed over the years evaluating human users' performance in voice recognition [28]. They show that the shorter the sentence, the more difficult it is to identify the source.

Based on this literature survey, it appears that the task of establishing the identity of a speaker may be challenging for human users, especially in the context of short SAS, and serves as a weak-link in the security of the Cfone SAS communication.

3.2 Attack Components

Our short voice imitation attacks involve the following components (our higher-level attack is depicted in Figure 3).

Data Relaying: In a Cfone, first an unauthenticated SAS protocol performs a key exchange during session initiation or Real Time Protocol (RTP) media stream (see Figure 4). This generates a session key, which will contribute to the encryption of the media during the Secure RTP (SRTP) session. So far the protocol is unauthenticated, therefore it is susceptible to a MITM [36] attack, and the session key might have already been revealed to Mallory. To ensure that Mallory was not present during unauthenticated key exchange, Alice and Bob verbally communicate the SAS over an SRTP session. In our attack, we assume that an MITM was performed during the unauthenticated key exchange protocol, and therefore Mallory has access to the plain audio during the conversation. Mallory is now interested in manipulating the SAS to hide her presence in unauthenticated phase of the protocol (i.e., non SAS communication). Mallory is not interested to alter any conversation except for the SAS dialogue (but of course interested in listening to all). Therefore, such conversations are simply relayed by Mallory, as is, to Alice and Bob.

Training Data Collection: Mallory needs to collect some data in advance to be used as the training set for the SAS voice impersonation attacks. For the *reordering attack*, Mallory needs to build a dictionary of distinct SAS words (e.g., digits for numeric, and words for PGP word list and Madlib SAS). In contrast, in *morphing attack*, she requires a few sentences to train the system to mimic the victim's voice. To do so, Mallory can listen to several samples of the victim's voice and collect words spoken by the victim from a previous VoIP session, or even the session under attack. Alternatively, Mallory can fool the victim into recording these training samples with social engineering trickeries (discussed in Section 6).

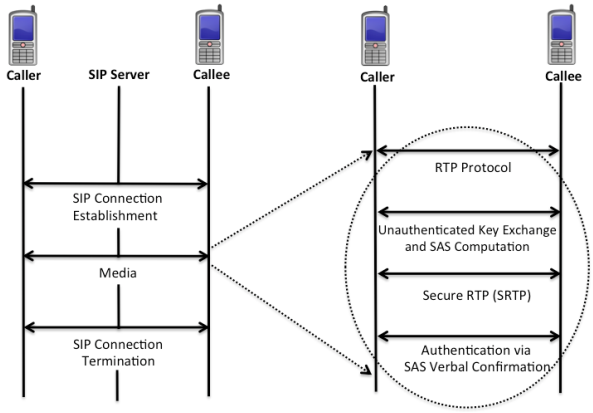

Figure 4: Cfone Protocol Flow (SIP: Session Initiation Protocol; RTP: Real-Time Transport Protocol)

Keyword Spotting: To replace a valid SAS with a new desired SAS, Mallory needs to first look up the SAS in the first few conversations over SRTP, while Alice and Bob verbally exchange the SAS. She can simply relay any non-SAS dialogue, and only manipulate the SAS at the right time. Manually performing this look-up for the presence of SAS within arbitrary conversations might be tedious. An alternative is keyword spotting, which can be performed automatically. Keyword spotting deals with identifying a specific word in an utterance. It can be performed online on an audio stream or offline on audio files in audio mining application. Several keyword spotting methods have been proposed in literatures. The work of [42] provides a comparison of different keyword spotting approaches. Any of these approaches can be used by Mallory to detect the SAS values (numbers or words). In Section 4, we will describe our implementation of a keyword spotter based on off-the-shelf voice recognition systems.

Interruption and Insertion: Once the SAS dialogue is found, Mallory should "drop" it to make it unavailable to the other party, and replace it with an forged SAS (recall that our threat model allows dropping arbitrary packets). At this point, the voice MITM on SAS happens. The forged SAS is either derived from previously recorded voice of the victim saying the same SAS, or is constructed by making a collage from the victim's speech (reordering attack), or is constructed by morphing victim's voice signal (voice morphing attack). It is not implausible to imagine that Mallory could be a professional impersonator who can speak a thousand voices ("Rich Little"). Usually this is not the case. Hence, we rely on automated reordering attack and voice conversion techniques.

Voice Reordering: An attacker who wants to insert a forged SAS into the conversation can build a dictionary of all possible words, and impersonate a legitimate party by remixing SAS in his/her voice, which we call reordering attack. After collecting SAS atomic units, she can cut pieces of the legitimate audio signal and make an offline collage of SAS messages for future use. Another attack similar to reordering attack is a text-to-speech system which is specifically trained to produce synthesized voices only on a limited domain of vocabulary. An example of such a synthesizer is presented in [16].

Voice Morphing: Although a limited domain synthesizer may produce audio with almost the same quality as a human being, de-

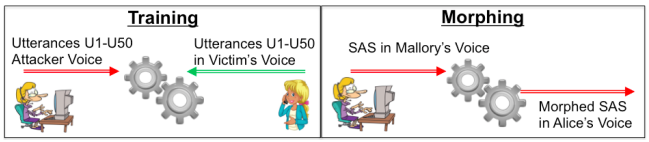

Figure 5: High-level diagram of morphing attack

pending on the SAS encoding, pre-collecting all words is not always practical. And, if the attacker does not have all possible units of SAS in the dictionary, she can not produce remixing or a limited domain synthesizer. In this situation, Mallory can try to mimic victim's voice. Attack could be successful if the adversary can "convert", for example, his own voice to the target's (victim's) voice. We call this the voice morphing or conversion attack. There are several voice conversion and transformation techniques which change the characteristic of voice such as frequency, pitch, and timbre [40, 41, 53]. Other techniques find a relation between human articulators and voice features [16, 43–46]. All these techniques work on a training system to adapt the system and can eventually convert any utterance in Mallory's own voice to the target voice even though such voice is not available in the training set. Mallory only needs to collect a *few minutes* worth of training data of the victim voice in order to perform this conversion. Later, Mallory may also produce an offline dictionary of all possible SASs in the victim voice. Figure 5 is the high-level picturization of this attack.

3.3 Attacking Different Mechanisms

The attacks presented above can be applied to undermine the security of 2-Cfones mechanisms. To attack Compare-Confirm, Mallory needs to do voice impersonation in both directions: impersonating both Alice's and Bob's voices. To compromise the Copy-Confirm mechanism, Mallory only needs to impersonate the SAS in one direction. If Mallory is interested in doing only a one-way MITM attack (e.g., Alice to Bob), it only needs to do impersonation on the channel over which the result of SAS comparison is conveyed (e.g., by Bob to Alice). Here, Mallory simply needs to impersonate "Yes" in Bob's voice thereby fooling Alice into accepting an attacked session.

With n-Cfones, the Peer-VerifySAS can be attacked in the same way as 2-Cfones, except that Mallory may need to do the attack on multiple point-to-point SAS exchanges in case of the latter. Leader-CopySAS can be attacked in the same way as Copy-Confirm in one-direction or both directions. Leader-VerifySAS can be similarly attacked. In addition, all the mechanisms can be relatively easy compromised via the "group count impersonation" attack whereby Mallory simply increases the group count by at least 1 and impersonates that "increased count" in leader's or a peer's voice.

4. DESIGN & IMPLEMENTATION

Communication Channel: Java Media Framework API (JMF) enables audio, video and other media to be captured, played, and streamed. We used JMF API to capture and transmit RTP packets at each party (Alice, Bob and Mallory). JMStudio open source code was adopted to implement the communication channel, to receive, capture and transmit media streams across the network.

Datasets Used: We used a variety of samples in different noise profiles, including samples recorded in professional recording environment as well as data collected using basic audio recorders. To have a good variety of recordings, we used three different datasets: First is the Arctic US English single speaker databases which has been constructed at the Language Technologies Institute at CMU. A detailed report on the structure and content of the database and the recording environment is available in [24]. The databases consist of around 1150 utterances include US English male and female experienced speakers. The second dataset is VoxForge, set up to collect transcribed speech for use with Free and Open Source Speech Recognition Engines. We picked US English male recordings with 16KHz sampling rate. The data samples we used are all recorded in unprofessional recording environment with free tools such as Audacity and the narrators are non expert speakers. Finally, we recorded two other voices using the basic audio recorder on an iPhone 5s, and two voices recorded by Audacity 2.0.5 on a Macbook Air laptop with internal microphone. Same as the VoxForge dataset, the narrators are not expert speakers. The Audio files in all our datasets are in WAV (Microsoft) 16 bit signed PCM format with a sampling rate of 16 KHZ in mono with single channel.

Keyword Spotting: For the purpose of keyword spotting, we used CMU's Sphinx open source speech recognition system. We utilized Sphinx4 recognizer. Sphinx-4 is very flexible in its configuration, providing a high-level interface to setup most of the components of the system. The configuration file is used to set all variables including recognizer, decoder, search manager, acoustic model, language model and dictionary components as well as configuration parameters such as: the absolute beam width that specifies the maximum number of hypotheses to consider; relative beam width that defines a trade-off between accuracy and search time; language weight or language scaling factor; insertion probability that controls word breaks recognition; and the silence insertion probability that controls how aggressive Sphinx is at inserting silences.

Sphinx takes the voice waveform as input, splits it into utterances by silences, then recognizes it based on the best matching combination of words. First, it gets a feature vector of each frame and then uses models to match this feature vector with the most probable feature vector in the model. So, it was important for us to adapt the models to fit our purpose and obtain accurate recognition results.

Three models are used in Sphinx speech recognition system. First is the *acoustic model* that contains acoustic properties for each phone. We evaluated the speech recognition with CMUSphinx acoustic models, which was quite acceptable for numeric SAS recognition. However, we adopted the acoustic model for PGP word list and Madlibs based on our speakers. As we will present in Section 5, it is enough to have 5 minutes of speech of a speaker to achieve a high accuracy. The second model involves a phonetic dictionary that contains a mapping from words to phones. We adapted the CMU's Pronouncing Dictionary (CMUdict) to cover all words available in PGP word List. The third model is language model or a language grammar that defines the structure of the language, such as the probability that two specific words come consequently. Such model is essential to restrict word matching. Compared to natural language structure, SAS language structure is very simple, it is a series of digits, or is two (or more) words from a PGP word list, or a sentence based on a Madlib phrase. Therefore, we built a grammar for our specific design.

There are some implementations of CMU Sphinx as keyword spotter. We changed Sphinx Audio Alignment code to transcribe and retrieve the time information for certain words. Once the appearance time of a SAS is captured, we snip that frame and add it as a single SAS to our SAS attack dictionary.

Reordering Attack: To implement the reordering attack, we developed a simple Java application that reads individual SAS words and produce any SAS combinations. Later, any of these combinations are picked and inserted in the voice MITM attack. Obviously, rather than building an offline dictionary of all combinations, remixing can be performed on the fly at the time of the attack. An alternative is a limited domain synthesizer speaking in the victim voice. Festival [2] limited domain synthesizers can produce voices

very similar to the target voice, but its performance is optimized whenever all SAS atomic units are pre-recorded at least one time.

Morphing Attack: A text-to-speech tool that can speak with a victim's voice might be suitable for a morphing attack. Such systems usually require a large collection of good quality training data to capture features, style, and articulation of the source voice. For example, AT&T Natural Voices [5] and Model Talker [6] promises good quality synthesis. However, still after hours of training, voices generated by such systems may sound synthesized and unnatural.

An alternative is voice converters that convert a source voice to a target voice by mapping features between the two voices. The voice conversion framework we used in our morphing attack is CMU's Festvox [4] voice transformation. Festvox gets trained by only a few sentences (less than three minutes) spoken in both the source (attacker or default Text-to-Speech, TTS voices) and the target (victim). Therefore it requires much less effort than other synthesizers. Once trained, a synthesized voice is built based on the target system. It can either act as a TTS tool in the victim's voice or can be used to convert any utterance spoken by Mallory to the same utterance in the victim's voice. Rather than converting properties of the voice, Festvox predicts the position of articulators from the speech signal and maps between the speakers and create voices in the target voice.

We trained the system with less than 50 sentences from the victim and an attacker voice, and converted all the possible SAS atomic units from the attacker voice to the victim voice. Using this system, we built an offline dictionary of all possible SAS combinations even though the SAS atomic units have not been spoken by the victim before. The offline dictionary can be queried at the MITM attack time to insert new forged SAS.

Attack Implementation – Putting the Pieces Together: To evaluate the feasibility of our attack, we setup an RTP communication channel between Alice and Bob with Mallory acting as the router in the middle. We used the JMF framework to send audio captured from the built-in microphone of Alice's computer to Mallory. Mallory receives the RTP stream and stores it in WAV format audio file. Duration of each audio file is set to be 3 seconds.

In the attack, Mallory's initial goal is to search for the presence of a SAS in the regular conversation between Alice and Bob. To this end, after receiving the first audio file, our application on Mallory's node calls CMU Sphinx keyword spotter to look up possible SAS in the captured audio files. We evaluated the performance of the attacking application with two different grammars. The first grammar looks up all possible SAS combinations. For example, a two word PGP word list SAS could be "dashboard liberty" which is included in our keyword spotter grammar. The second grammar just looks up some possible phrases that Alice and Bob might say just before confirming the SAS such as "SAS shown on my side is ...", or "My SAS is ...". The second grammar makes the keyword spotting faster but it is not completely predictable, as there are many ways which users can confirm their SASs.

We assume that SAS atomic units (e.g., digits or words) have already been forged and are stored on Mallory's system for further use following the morphing or reordering attack discussed earlier in this chapter. The keyword spotter works in parallel with RTP receiver, audio files are stored and processed in a First-In-First-Out (FIFO) order. Those files containing SAS are replaced with same size (bit-wise) recording matching the MITM desired SAS, and files not containing the SAS are simply relayed to Bob.

SAS might split in two audio files in some situations. This means that one part of the SAS is located at the end of one file and the other part is located at the beginning of the second file. To detect such combinations, our keyword spotter concatenate current captured file with the previous file and looks for the SAS in the mixed file. The performance of this attack is provided in Section 5.

5. EXPERIMENTS AND EVALUATION

In order to measure the effectiveness of our attacks, we performed objective and subjective evaluations. The objective evaluation quantitatively measures the similarity of a forged voice to the original voice. The subjective evaluation measures human users' qualitative capability of differentiating a forged voice and the original voice so as to detect our attacks. In this section, we present both forms of evaluations and the respective results. Moreover, we report on the delay introduced by our attacks.

5.1 Objective Evaluation

In speech and speaker recognition systems, it is common to extract a multi-dimensional vector of components of the underlying audio to identify the linguistic features of the signal. We used Mel-Cepstral Distortion (MCD) to measure the similarity between a forged (converted) SAS and an original SAS by calculating the Euclidean distance between feature vector of the forged SAS and that of the original SAS. A similar strategy has been used in several speech conversion and synthesis systems [20,27,29] to measure the distance between a synthesized and an original version of the same utterance. If the difference between feature vector of the original SAS and the forged SAS is minimized, the forged voice is close to the original voice and detecting the attack would be inherently difficult. Lower MCD shows better conversion (a forged SAS is so similar to the original one that it is not easy to distinguish the two).

To compute MCD, we extract features of the forged SAS (fSAS) built from attack engine (morpher) and features of the original SAS (oSAS) spoken by the victim, and calculate the difference between the two. MCD computation is defined in Equation 1, where v_d^{fSAS} denotes the d-th MCEP[1] of fSAS and v_d^{oSAS} denotes the d-th MCEP of oSAS. In TTS applications, typical parameters are 25 dimensional mel-frequency scaled cepstral coefficients (d between 0 to 24). The 0-th dimension represents the overall signal power (loudness). Therefore, to eliminate the effect of speaker loudness, MCD has been calculated for $d = 1..24$.

$$MCD(v^{fSAS}, v^{oSAS}) = \frac{10}{ln(10)} \sqrt{2 \sum_{d=1}^{24} \left(v_d^{fSAS} - v_d^{oSAS}\right)^2} \quad (1)$$

In our objective evaluation, we trained the voice conversion engine to convert between pairs of 8 different male and female speakers from our voice dataset (Section 4) representing victims and attackers. A combination of 20 different conversions was built. To first test the effect of the training set size on the conversion performance, we trained the system with 50, 100, and 200 sentences. We noticed an average MCD improvement of only 1.6% by increasing the size of the training set from 50 to 100 and an average MCD improvement of only 1.1% by increasing the size of the training set from 100 to 200. This means that increasing the training set size beyond 50 sentences does not significantly improve the performance of conversion. As a result, in the rest of the experiments, we use 50 first sentences of Arctic dataset in the training phase. The average duration of each utterance of the training set is 5 seconds, and the average duration of the whole set of 50 sentences is 2 minutes and 30 seconds. This means that in order to train a system to speak in the victim's voice, we are required to collect less than 3 minutes of her voice. This is quite short and therefore training does not seem to impose a challenge for the attacker in the conversion process.

Table 1 presents the results of our objective evaluation. We present the results of only 4 conversions between same genders and

[1]Weighted average of the magnitudes of cepstral peaks.

Table 1: Objective evaluation results for the morphing attack

Row	Utterances	Conversion Name	Source (Attacker)	Female 1	Male 1	Female 2	Male 2
			Target (Victim)	**Female 2**	**Male 2**	**Male 2**	**Female 1**
1	First 50 Sentences of Arctic	Single Speaker	MCD Before Conversion (dB)	5.28	5.65	5.65	6.15
2			MCD After Conversion (dB)	2.11	2.34	2.34	2.39
3		Source to Target	MCD Before Conversion (dB)	7.27	8.28	8.94	9.41
4			MCD After Conversion (dB)	4.64	4.91	4.97	4.96

			SAS Size (bits)	**20**	**80**	**128**	**20**	**80**	**128**	**20**	**80**	**128**	**20**	**80**	**128**
5															
6	20 SASs of Different Sizes	Single Speaker	MCD After Conversion (dB)	1.95	1.97	2.01	2.17	2.21	2.27	2.17	2.21	2.27	2.23	2.31	2.34
7		Source to Target	MCD After Conversion (dB)	4.18	4.33	4.44	4.53	4.61	4.68	4.72	4.77	4.86	4.45	4.63	4.70

different genders. The other 16 conversions yielded similar results and are not reported here due to space constraints.

To obtain a measure of how good the conversion process is, we first performed a conversion between utterances of a single speaker (the victim) spoken and recorded in two different noise profiles. We call this the "Single Speaker" conversion. Rationally, such conversion would gain the optimum conversion result. Row 1 of Table 1 shows MCD between the two set of 50 recordings (in different noise profiles) of the victim before the conversion, averaged across all recordings, which can be used as a reference of what MCD values are acceptable. Row 2 of the table shows the result of conversion. The Single Speaker conversion gives us a baseline MCD to measure the quality of other conversions from attacker voice to the same speaker as the victim.

Row 3 depicts the MCD between an utterance in the training dataset spoken by the source and the same utterance spoken by the target before the conversion, averaged across all utterances. This parameter characterizes the actual similarity/dissimilarity between the attacker and victim voice before conversion. Recall that we used 50 utterances spoken by the source and the target to train the system to convert from attacker to the victim. We refer to this conversion as the "Source to Target" conversion. Row 4 shows result of this conversion. By comparing row 3 and row 4, it can be seen that after conversion, the distance between source and converted voice becomes less than the initial distance. This demonstrates the effectiveness of the converter.

Comparing the result of Single Speaker and Source to Target conversions (row 2 and 4 of the table), we can observe that the MCD between converted voice and the original voice is higher in the Source to Target conversion (which is the real attack scenario) than the Single Speaker conversion (the optimum conversion result). This is intuitive. However, by comparing row 1 and 4, it is interesting to note that the MCD values after conversion are slightly less than MCD values of single speaker before conversion, which suggests that Source to Target conversion produces a voice that is comparable to the voice of the victim in a different noise profile.

Subsequently, we tested the performance of the converter for the purpose of our attack (i.e. SAS morphing). Here, we used the trained system (described in the above two paragraphs) to convert 60 utterances from our potential attackers to victims representing 20 short, medium and long SAS with size of 20 bits, 80 bits and 128 bits respectively. Average duration of saying short, medium and long SAS is approximately 1.2, 2.1 and 4.4 seconds respectively. Row six of Table 1 shows the distance between the converted SAS (resulted from Single Speaker converter) and the original SAS (spoken by the victim). And finally last rows show the average distance between the forged SAS (resulted from Source to Target converter) and the original SAS (spoken by the victim).

For all pairs of speakers, we see a clear pattern of increase in MCD with increase in the SAS size. This suggests that the quality of SAS conversion degrades as the SAS size increases, which may make longer SASs more detectable than shorter ones. Comparing rows six and seven, we see that the quality of SAS conversion degrades only slightly when source and target are different speakers (similar to the case of non-SAS samples as discussed above).

Unlike the morphing attack that maps between features of the attacker and the victim, in reordering, filtering characteristics of vocal cords of the speaker are not changed. As the name suggests, reordering simply remixes the ordering of words or digits. Therefore, unlike morphing attack, no new voice is generated in reordering and in fact the attacked voice has the same features as that of the victim's voice [13, 47]. Hence, we did not conduct objective evaluation test on the reordered SASs.

5.2 Subjective Evaluation: User Study

We report on a user study we conducted to measure users' capability to detect our voice imitation attacks against Cfones. Specifically, we conducted a survey, approved by our University's IRB, and requested 30 participants to answer several multiple choice as well as open-ended questions about the quality and (speaker) identity of certain recordings. There were basically two categories of questions: one related to the quality of the forged SAS (9 questions) and another related to speaker identification (9 questions).

Survey and Participant Details: The survey[2] was created using the Question Pro online survey software which gave us the flexibility in designing multiple choice *multimedia* questions. Participants were recruited by word of mouth and were only told that the purpose of the study is to assess speech recognition quality. Following best practices in usable security research, we did not give details about the security purposes behind the survey in order to prevent explicit priming which may have biased their responses. However, in real-life, users should be warned that an incorrect SAS validation may harm the security of their communications. Moreover, our attack study was targeted towards average users, and therefore we can not deduce the performance of more or less security-aware users.

The demographic information of the participants is presented in Table 2. They were mostly young and well-educated, with almost equal gender split. Such a sample is suitable for our study because if the study results indicate that it is hard for young and educated participants to detect our attacks, it may be harder for older and less educated (average) people. The survey took each participant about 15 minutes to complete.

The *quality test* in the survey is similar to the Mean Opinion Score (MOS) test [37]. It consisted of 9 questions, each asks the participants to listen to the embedded SAS recording and rate the quality, in terms of genuineness (naturalness) on a scale of one to five (5: excellent; 4: good; 3: fair; 2: poor; 1: bad). Each question presents two SAS recordings, that could be the original speaker recording in different noise profiles, reordered SAS or morphed/converted SAS. Different set of original recordings were played when subjecting the participants to reordered SAS and morphed SAS.

[2]Available at: http://surveys.questionpro.com/a/t/AKvTXZQtoV

Table 2: Demographic Info: User Study

	N = 30
Gender	
Male	53%
Female	47%
Age	
18-24 years	34%
25-34 years	62%
35-44 years	3%
Education	
High school graduate, diploma or the equivalent	5%
Some college credit, no degree	7%
Bachelor's degree	55%
Master's degree	24%
Doctorate degree	9%
English as First Language	
Yes	28%
No	72%
Hearing Impairment	
Yes	10%
No	90%

The *speaker identification test* contained 9 questions, each presents three sentences spoken by the same speaker and asks the participant to listen to these three recordings so as to first get familiar with the voice. Then, the participants should listen to another two recordings (forged or real by the same or different speaker), and answer "yes" if they think any of the recordings are of the same person, and "No" if they think it is a different person, and "Maybe" if they can not make a distinction. The participants were asked to ignore any dissimilarity in the quality of the recordings.

We collected different types of SAS recordings, including four 16-bit numerical SAS, eight PGP word lists and four Madlibs. We also presented four longer SASs including 32-bit PGP word list and 32-bit Madlibs. Generally, 32-bit numeric SAS is not secure against reordering attack (since in only one transmission of SAS, all 10 distinct digits might appear). Therefore 32-bit numeric SAS was not questioned. In addition, the two samples of morphed version of "Yes" and "No" phrases that can be used to attack Copy-Confirm approach (Section 3.3) and launch Denial-of-Service attacks (Section 6) were played. Finally, to evaluate the group count impersonation attacks (Section 3.3), we played two recordings of individual forged numbers in victim's voice to represent a group leader who is announcing the (increased) group count.

As mentioned in Section 4, the voice dataset for the evaluation consists of four collections from CMU Arctic, four collections from VoxForge and four unprofessional recordings collected by us. For the morphing attack, we trained the system with 50 sentences from each speaker. The survey audio samples consist of male and female speakers in different noise profiles.

Results for Quality Test: Table 3 summarizes the result for the quality test, showing the average ratings provided by the participants assessing the quality of original and forged SAS recordings.

As the table shows, in all the cases, participants rated original recordings between "fair" and "good" except for 32-bit Madlib, which is rated as "poor." Participant rated reordered SAS as "fair" or "good" except for 32-bit Madlib, which is rated as "poor". The results also show that they did not notice much difference between the reordered SAS (rated mostly between good and fair) and the original SAS. The participants rated morphed SAS somewhere between "poor" and "fair". Interestingly none of the forged voices were rated as "bad" quality, while none of the original voices were rated as "Excellent" quality. No statistically significant differences emerged between the two types of ratings when tested with the Wilcoxon Signed-Rank. We observed a relatively high standard deviation in all answers that we believe originates from different interpretation each person has of the word "quality". Answers to our open-ended question reveal that participants had a different definition of quality and genuineness, and therefore their quality rating is affected by their own definition. For example, participants mentioned "background noise", "loudness", "clarity", and "speed" as their measure for quality. Only three participants rated the recordings based on "genuineness", "naturalness", and "machine generated versus human."

Moreover, we can see that the difference between the ratings for morphed SAS and original SAS is generally more than the difference between the ratings for reordered SAS and original SAS (only exception is 32-bit Madlibs). This suggests that reordering attack might generally be harder to detect for the participants than morphing attack. It is interesting to note that for PGP words, participants rated the reordered SAS higher than the original recording. This implies if the attacker collects enough data to perform the reordering attack on PGP words, the quality of the forged SAS may even be perceived better than the original one. However, the same was not true for Madlibs as the participants rate the attacked samples slightly lower than the original ones. Madlibs have a correct grammatical structure and therefore people usually read them following a sentence flow, which may make it difficult for the attacker to split and remix.

Results for Speaker Identification Test: We next evaluated the speaker identification test. Recall that in each question of the speaker identification test, participants first get familiar with a voice, then they are asked if any of the two subsequent SASs is spoken by the same person or not. In all of our calculations, we treated half of the uncertain answers ("Maybe") as "Yes" and half of them as "No".

We define the "Yes" answer as the Positive class and the "No" answer as the Negative class. By this definition, True Positive (TP or hit) is the instance of recognizing a legitimate familiar voice correctly (higher values show Cfone system works well under benign, non-attack, setting). False Positive (FP or false alarm) is the instance of considering an attacked voice as a familiar voice (higher values represent that the attack is working and participants are not able to detect it). True Negative (TN or miss) is the instance of not recognizing a legitimate familiar voice. And, False Negative (FN) is the instance of correctly recognizing that an attacked voice is unfamiliar (higher value represents that the attack is not successful and participants can detect it). To evaluate Cfones, we calculated *Accuracy* and *False Discovery Rate* (FDR) in the presence of different types of attacks. Accuracy is defined as $(TP + TN)/(TP + TN + FP + FN)$ (the proportion of true results), and FDR is defined as $FP/(TP + FP)$ (the proportion of the false positive against all the positive results). In presence of an effective attack, FP increases, which is reflected in lower accuracy and higher FDR values.

Table 4 depicts our evaluation metrics corresponding to a SAS spoken by a "different" person (second column, representing the naive attack), a SAS generated by converting attacker voice to victim voice (third column), and a SAS spoken by the same person but reordered (fourth column). The results are shown for different

Table 3: Mean (Std. Dev) ratings for original and attacked SAS

		Numeric	PGP Words	16-Bit Madlib	32-Bit Madib
1	*Original SAS*	4 (0.95)	3.05 (1.21)	4.15 (0.9)	3.28 (1.28)
2	*Reordered SAS*	3.67 (1.08)	3.23 (1.22)	3.64 (1.35)	2.68 (1.30)
3	*Original SAS*	3.51 (1.19)	3.74 (1.09)	3.34 (1.30)	2.56 (1.41)
4	*Morphed SAS*	2.33 (1.20)	3.18 (1.25)	2.75 (1.39)	2.26 (1.35)

type of SAS. Also shown is the overall aggregated result among all three types of SASs (the last matrix).

First of all, the table illustrates a relatively high TN for SASs played in a "different voice" (row 2, column 2 – bold fonts in dark gray shade), which means that when a totally different voice is presented, people successfully detect the difference with a high chance (about 80%). This demonstrates that if the (naive) attacker just inserts a different voice in its MITM attack, it would be detected by the users with a high probability. This provides an important quantitative benchmark to compare the performance of attacks with.

The effectiveness of our voice imitation (morphing and reordering) attacks is represented by FP (first row results for column 3 and 4 – bold fonts in light grey shade), which is also reflected in FDR (last row results for column 3 and 4). Although FDR is not very high (somewhere around 50-60%), it is important to look at the corresponding Accuracy of the Cfone system under our attacks (row 3, column 3 and 4), which is roughly around 50% or lower, and shows that people are not accurate in recognizing the familiar voice saying SAS even in non-attack (benign) scenarios, and almost 50% of the times participants detect original voice in a different noise profile as fake voice. That is, even in non-attack scenario, participants are making almost random guesses to decide whether the voice is real or fake. Thus, we can conclude that, under our attacks, users do not perform any better than a random guess in recognizing an forged SAS, and in fact the result is very similar to recognizing original similar voice in a different noise profile. In short, people are as successful as recognizing a forged SAS as they are successful in recognizing an original SAS in a different noise profile.

Similar to the quality test, the speaker identification test shows that reordering attack generally works better (e.g., has higher FDR) than the morphing attack. The performance metrics, however, do not indicate any significant differences in the way users may detect the attacks against different SAS types (numeric, PGP or Madlibs). They all seem almost equally prone to our attacks. In Section 3, we referred to the linguistic studies that demonstrate people are more successful in recognizing familiar voices when they are presented with long sentences rather than short sentences. Our experiment for short SAS confirms this insight.

Copy-Confirm and Group Count Attacks: Copy-Confirm SAS validation mechanisms work by Alice reading the SAS and Bob accepting or rejecting by saying a "Yes" or "No" phrase. In our evaluation, we converted two type of Accept/Yes and Reject/No phrases (i.e. "Yes, It's a match") to represent Mallory who drops a reject response from Bob and rather injects an accept response in Bob's voice to authenticate a connection. We repeated our speaker identification test for Yes and No phrases. The results show that this conversion follows the same pattern as SAS conversion. The TN is relatively high (between 70-80%) for different voices (people can detect a different voice), but precision and FDR are around 50% for non-attacked and attacked scenario. That means reordering and morphing attack on the yes and no phrases is as successful as SAS conversion. Reordering works the best in such situation as the attacker only needs to replay a previously spoken phrase.

The group count attack is an attack in n-Cfone where Mallory announces the (increased) group count in the leader's voice. Similar to numeric SAS conversion, Mallory need to generate and insert a number. However, here it is much easier as the number is only one digit long. Due to similarity between Numeric SAS conversion and group count conversion, we did not evaluate group count conversion, and rely on the result of Numeric SAS morphing and reordering attack to prove effectiveness of this attack.

Open-Ended Feedback: As part of the survey, we also requested the participants to provide open ended feedback as to how they found the experiment overall. Majority of the participants felt that recognizing even a familiar voice is difficult and confusing. Some of them believed it is the background noise in the recordings that makes the recognition task difficult. The answers to the open-ended question in these series of question such as, "In noisy data, it is more difficult to compare the tracks.", or "Challenging" and "Confusing" show that people find it difficult to recognize familiar voices especially in the presence of some background noise, which may be common in telephonic VoIP conversion.

Video-Audio Synchronization Test: Our final question in the survey was designed to test the effectiveness of video SAS, i.e., whether users can detect the asynchrony between the forged verbal SAS and the lip movement associated with the original SAS in the video stream. For example, under our voice MITM attacks, the audio SAS may be "1234" but the lip movement in the video may correspond to "8604". If the users can detect the presence of this asynchrony, the attack could be detected. We recorded a one-minute video of a person and later replaced part of the audio on this recording with a different SAS from another recording of the same speaker. We asked participants to watch the video and provide their opinion about the quality and the genuineness of the video. Only two participants, recognized the mismatch between the audio and video, while others found the video to be "genuine", "excellent", and "good". This experiments shows that even the use of the video channel would be vulnerable to the audio MITM attack.

Study Limitations: Two attack variants, three SAS types, and Yes/No conversions, and video SAS, had to be included in the survey. Moreover, in our speaker identification questionnaire, participants had to listen to three additional recordings of a person (probably multiple times) to get familiar with the voice. Therefore, to keep the survey concise, we limited ourselves to play one or two samples of each of the attacks. While ideally more samples would give a better judgment, it would also make the survey long, and may reduce the user experience and accuracy. Furthermore, in quality test, results seem biased due to the definition of "quality". The core idea was to rate "genuineness" of the recordings, while peoples' answers seemed affected by the parameters such as noise and loudness. Finally, all the samples were drawn from US English, while the first language of a majority of participants was not English (Table 2). The familiarity with English might affect the result. To our knowledge, most Cfone applications are developed in English, so we did not perform a language-centric study.

5.3 Delay of the Attack

The voice MITM attack naturally introduces a delay associated with the MITM attack on the non-voice, non-SAS channel communication, and with the voice impersonation on the SAS channel communication. Prior work [54] shows that MITM attack on non-voice channel can be efficiently performed and therefore we focus on the delay related to the SAS voice impersonation. The dominating delay in voice impersonation could be because of the keyword spotting procedures Therefore it is critical to analyze the spotting delay in our attack implementation (discussed in Section 4).

Using simpler grammars (i.e. the SAS confirmation phrases) can improve the keyword spotting method. In offline keyword spotting (such as the one that we used), duration of each stored audio file can affect the performance, since we are running the RTP receiver and the keyword spotter in parallel. So if the duration of the stored audio file is less than the execution of keyword spotting method, no delay would be introduced by the keyword spotter.

We evaluated execution time of the attack with different keyword spotting grammar sizes and different duration of the audio file in offline keyword spotting. Table 5 summarizes our attack timing ex-

Table 4: Results of subjective evaluation for different attacks and SAS types. TP (row 1, column 1) and TN (row 2, column 1) show answers to benign setting (instances that are successful or not successful in detecting a familiar voice). FP (row 1, column 2-4) and FN (row 2, column 2-4) show effectiveness of each attack (naive different voice attack; reordering and morphing attacks). Higher FP (lower FN) shows more powerful attack. Accuracy is the accuracy of Cfones under different attacks (lower values show the system is not working well under the attack). FDR represents the overall effectiveness of the attacks (higher values mean better attack).

	Presented Numeric SAS voice			
	Original	*Different*	*Morphed*	*Reordered*
Detected as: Yes	57.50%	14.52%	**61.25%**	**87.50%**
Detected as: No	42.50%	**85.48%**	38.75%	12.50%
Accuracy		71.49%	48.13%	35.00%
FDR		20.16%	51.58%	60.34%

	Presented 16-Bit PGP SAS voice			
	Original	*Different*	*Morphed*	*Reordered*
Detected as: Yes	51.67%	17.71%	**50.00%**	**68.75%**
Detected as: No	48.33%	**82.29%**	50.00%	31.25%
Accuracy		66.98%	50.83%	41.46%
FDR		25.53%	49.18%	57.09%

	Presented 16-Bit Madlib SAS voice			
	Original	*Different*	*Morphed*	*Reordered*
Detected as: Yes	50.83%	21.67%	**51.39%**	**81.67%**
Detected as: No	49.17%	**78.33%**	48.61%	18.33%
Accuracy		64.58%	49.72%	34.58%
FDR		29.89%	50.27%	61.64%

	Aggregated Rates on All SAS Types			
	Original	*Different*	*Morphed*	*Reordered*
Detected as: Yes	55.42%	17.96%	**50.58%**	**78.23%**
Detected as: No	44.58%	**82.87%**	49.42%	21.77%
Accuracy		68.86%	52.42%	38.59%
FDR		24.48%	47.72%	58.53%

Table 5: Attack Timing (highlighted cells denote only cases where delay is introduced by the attack: file duration $<$ attack duration)

# Words in Grammar	10	10	10	10	20	20	20	20	256	256	256	256
Audio File Duration (s)	1	3	5	10	1	3	5	10	1	3	5	10
Average Attack Duration (s)	1.28	1.63	2.04	2.08	1.48	1.86	2.26	2.35	3.15	4.5	6.59	9.38

periment. Number of words in the grammar is defined as 10 words for numerical SAS, 256 for 16-bit PGP word list and Madlibs, and 20 words for SAS confirmation phrases. For an audio recording with an average length of 3 and 5 seconds, and a grammar of 10 and 20 words, the attack duration is computed to be less than the audio recording duration itself, and therefore the attack does not introduce any delay in such cases. For a longer grammar of 256 words, the keyword spotting produces an average delay of less than 2 seconds, for an audio file of 1, 3 and 5 seconds. However, importantly, for all tested grammar sizes, the attack does not produce any delay if the offline keyword spotter stores and processes 10 second audio file. Grey-colored columns show the combinations that introduce delay.

As mentioned earlier, in cases the delay exists in our attack, it is mostly due to keyword spotting, particularly because the keyword spotter is looking for the SAS in the current file as well as a 1s file created by concatenating the current file and the previous file (to find those SAS that are distributed in two files). Real-time keyword spotters such as [21, 48] might be helpful in further improving the performance of the attack in such cases.

6. DISCUSSION AND SUMMARY

Evaluation Summary: Our objective evaluation shows that the distortion between the original SAS and morphed SAS increases with the size of the SAS. In other words, shorter SAS values show less distortion, which means that shorter forged SAS are more similar to the original SAS, and they would be more difficult to distinguish (and more prone to our attacks). This supports our hypothesis that short voice impersonation is easier for the attacker (harder for the users to detect) compared to long speech impersonation. We also observed that if attacker voice is similar to the victim voice, the result of conversion would be better.

Our subjective evaluation shows that in a non-attack scenario, almost only 50% of the times participants can detect familiar voices and 50% of the times they can not detect familiarity of a voice (played in a different background noise). This means that participant are making almost random guesses in normal, benign situation. However, people can distinguish a *different* voice from a familiar voice with about 80% success, and therefore a naive attack, where the attacker simply inserts her own voice (or that of another user), is not successful, and more complex attack is needed. This is where our reordering and morphing attacks are a good candidates, as 50-80% of observed instances can not detect such attacks, which means that, in the worst case, our attack works as good as the non-attack condition. Unlike our evaluation, in real-life, users may not pay due diligence when asked to validate the identity of the other speaker (secondary task) when making a call with Cfone (primary task). It would result in higher true negatives (i.e., fewer rejections in non-attack cases, or better usability) than what we observed, but would also lead to higher false positives (i.e., weaker security) especially when a reordered/morphed SAS is inserted. In other words, the Cfone system may be more usable in practice but less secure against our attacks.

Both evaluations support that a reordered SAS is more effective for the attacker (harder to detect) compared to a morphed SAS.

Acquiring Training Data: Our attacks require collecting prior audio samples from the victim. While the reordering attack requires previously spoken SASs to build a dictionary, the morphing attacks only require a few previously spoken sentences from which victim's voice features can be derived. Building training sets for the latter case is quite easy. The attacker can eavesdrop prior unprotected VoIP sessions of a victim. Since only a few sentences are needed, eavesdropping only a few minutes of conversation would be sufficient. The attacker can also record such samples from a victim by being physically close to the victim while the victim is talking in a public place or giving a public presentation.

As far as building training sets for reordering attack is concerned, the difficulty depends on the underlying SAS encoding type. While eavesdropping all (10) digits for numeric SAS is relatively easy (e.g., waiting for the victim to speak phone numbers, zip codes, and other numeric utterances), learning all PGP words or Madlib words might be challenging given these words may not be commonly spoken in day to day conversations. However, it is possible for the attacker to use social engineering techniques to address this challenge. Number of possibilities exists to this end. For example, the attacker can create crowd sourcing tasks on online websites (e.g., freelancer or Amazon Mechanical Turk) which asks the users to auditize proses which contain all PGP words or Madlib phrases. Similarly, the attacker can create audio CAPTCHAs, and use them on its own websites or other compromised sites, that challenge the

users to auditize words from books (similar in spirit to the idea of reCAPTCHAs).

Moving forward, we believe that our work also raises a more broader and general threat of "voice privacy." The malicious actors may use various approaches to record someone's voice samples and use these samples to compromise the security and privacy in another application (such as Cfones or voice recognition systems). While people seem quite concerned about their "visual privacy" in today's digital world (e.g., someone taking their picture), they may not consider their voice to be so sensitive (e.g., people often talk out loud in a restaurant and even talk to strangers). Given that audio sensors are very common and do not require explicit efforts from an attacker to record audio (unlike camera, for example), we believe that voice privacy can have several implications that may need careful attention.

Potential Defenses and Challenges: In light of our attacks against Cfones, a natural question is what can be done to improve the security of the underlying SAS validation process. One possibility is to rely upon multiple preceptory channels rather than just audio. For example, users may be asked to pay attention to the video (assuming video is available) while validating verbal SAS. In other words, if the attacker performs the voice impersonation against SAS, users may be able to detect this attack by looking at and analyzing the accompanying video of the communicating party – the lip movement of the person stating the SAS would not match with the spoken SAS. This could serve as a potentially useful defense to our attacks. However, it may present significant challenges in practice. First, the users may not be in a position to look at the video or may simply not pay enough attention to spot the lack of audio-visual SAS synchronization. In fact, our voice-audio synchronization test (Section 5), shows that only 2 out of 30 (only 7%) survey participants were able to detect such an audiovisual synchronization. Second, it is not hard to imagine that the attacker can manipulate the video packets in addition to the audio packets so the spoken audio matches with the video stream. The video impersonation attacks are feasible due to the same underlying weakness of the VoIP channel with respect to manipulation as the audio attacks. There exists some prior work that suggests image/video morphing attacks are feasible [52]. The need to influence both audio and visual channels at the same time may increase the complexity of the attack, however.

Another potential defense to our attacks could be integration of an automated voice recognition or voice biometrics system to Cfones. That is, in place of, or addition to, human voice recognition, a software component may be used to detect potential SAS forgeries. While voice biometrics is a rich area, achieving robust detection rates (i.e., low false negatives and low false positives) is still a challenging problem. In addition, existing voice biometrics system may not work well to thwart active voice impersonation and synthesis attacks [26]. Furthermore, given that the SAS challenge being authenticated in Cfones application is short (only few seconds worth of audio sample), it may not provide sufficient "knowledge" to the biometric system to extract features from the voice using which detection can take place. Our objective evaluation showed that the MCD distortion level increases with the length of SAS, which means shorter converted SASs may not be distinguishable from the original SASs even at a quantitative level.

Yet another potential solution to thwart the voice impersonation attacks against Cfones is to perform the SAS validation over an auxiliary channel that can be more resistant to voice and packet manipulation. PSTN communication is believed to offer such properties, and, when available, may be used to secure VoIP communication. For example, if the communicating devices support both VoIP capability (Internet connection) and PSTN connectivity (e.g., cellular connection), the non-SAS communication can take place over the former and SAS validation can take place over the latter. This mechanism is suitable for mobile phones – the Cfone app switches to a PSTN call when SAS comparison is performed by the user (Android, e.g., allows making VoIP and cellular calls simultaneously). A limitation of this defense mechanism is that it is only applicable to devices which have PSTN capability (such as cell phones).

An independent defense could be increasing the dictionary size to make reordering difficult, and to reduce the efficiency of automatic keyword spotting. Moreover, if the dictionary is not fixed, reordering will be impossible. An idea suggested in [12] is to choose words from a large dynamic space (e.g., front pages of today's newspapers). The dictionary can be chosen by users, or programmatically during key exchange. However, the security and user experience of this approach needs further investigation.

Another possibility is to employ the approach proposed by Balasubramaniyan et al. [14], which identifies and characterizes the network route traversed by the voice signal and creates a detailed fingerprints for the call source. For VoIP connection, this method is based on network characteristics, and therefore may only be effective if the attacker and the victim reside in different networks.

7. CONCLUSIONS

Crypto Phones aim to solve an important problem of establishing end-to-end secure communications on the Internet via a purely peer-to-peer mechanism. However, their security relies on the assumption that the voice channel, over which short checksums are validated by the users, provides integrity/authenticity. We challenged this assumption, and developed two forms of short voice impersonation attacks, reordering and morphing, that can compromise the security of Crypto Phones in both two-party and multi-party settings. Our evaluation demonstrate the effectiveness of these attacks, when contrasted with a trivial attack where the attacker impersonates with a totally different voice. We suggested potential ways and associated challenges to improve the security of Crypto Phones against the voice MITM attacks. A comprehensive future investigation is needed to develop a viable mechanism to thwart such attacks.

Acknowledgments

This work was supported in part by a Cisco grant. We would like to thank Patrick Traynor (our shepherd) and anonymous CCS'14 reviewers for their constructive comments and guidance. We are grateful to Dhiraj Rajani for his help with our subjective study setup. We are also thankful to N. Asokan, Steve Bethard, Raman Bhati, Jason Britt, Hugo Krawczyk, and all members of the UAB SPIES lab, for feedback on previous versions of this paper.

8. REFERENCES

[1] Infosecurity - Microsoft Expands Encryption to Foil Government Snooping. http://www.infosecurity-magazine.com/view/36034/microsoft-expands-encryption-to-foil-government-snooping.

[2] Limited Domain Synthesis. http://festvox.org/ldom/.

[3] NSA and All Major Intelligence Agencies Can Listen in to Encrypted Cell Phone Calls. http://www.nbcnews.com/tech/security/nsa-can-listen-encrypted-phone-calls-theyre-not-only-ones-f2D11744226.

[4] TRANSFORM: Flexible Voice Synthesis Through Articulatory Voice Transformation . http://festvox.org/transform/transform.html.

[5] AT&T Labs Text-to-Speech. http://www2.research.att.com/~ttsweb/tts/index.php.

[6] ModelTalker Speech Synthesis System. http://www.modeltalker.com.

[7] Open Whisper Systems. https://whispersystems.org/.

[8] PGPfone - Pretty Good Privacy Phone. http://www.pgpi.org/products/pgpfone/.

[9] Silent Circle – Private Communications. https://silentcircle.com/.

[10] The Zfone Project. http://zfoneproject.com/.

[11] ZORG - An Implementation of the ZRTP Protocol. http://www.zrtp.org/.

[12] N. Asokan, T. Chan, and G. Krishnamurthi. Authenticating Security Parameters, Feb. 8 2007. US Patent App. 11/672,900.

[13] B. B. Voice Conversion Uning Articulatory Features. Master's thesis, International Institute of Information Technology Hyderabad, 2012.

[14] V. A. Balasubramaniyan, A. Poonawalla, M. Ahamad, M. T. Hunter, and P. Traynor. PinDr0p: Using Single-Ended Audio Features to Determine Call Provenance. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 109–120. ACM, 2010.

[15] J. Benesty, M. M. Sondhi, and Y. Huang. *Springer Handbook of Speech Processing*. 2008.

[16] A. W. Black and K. A. Lenzo. Limited Domain Synthesis. Technical report, 2000.

[17] M. Cagalj, S. Capkun, and J. Hubaux. Key agreement in peer-to-peer wireless networks. In *Proceedings of the IEEE (Special Issue on Cryptography and Security)*, 2006.

[18] J. P. Campbell Jr. Speaker Recognition: A Tutorial. *Proceedings of the IEEE*, 85(9), 1997.

[19] M. Chevillet, M. Riesenhuber, and J. P. Rauschecker. Functional Correlates of the Anterolateral Processing Hierarchy in Human Auditory Cortex. *The Journal of Neuroscience*, 31(25), 2011.

[20] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad. Voice Conversion Using Artificial Neural Networks. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, 2009.

[21] A. Ganapathiraju and A. N. Iyer. Method and System for Real-Time Keyword Spotting for Speech Analytics, July 20 2012. US Patent App. 13/554,937.

[22] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun. Loud and Clear: Human-Verifiable Authentication Based on Audio. In *International Conference on Distributed Computing Systems (ICDCS)*, July 2006.

[23] H. Hollien, W. Majewski, and E. T. Doherty. Perceptual Identification of Voices Under Normal, Stress and Disguise Speaking Conditions. *Journal of Phonetics*, 1982.

[24] A. W. B. John Kominek. CMU ARCTIC Databases for Speech Synthesis, 2003.

[25] 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.

[26] T. Kinnunen, Z.-Z. Wu, K.-A. Lee, F. Sedlak, E.-S. Chng, and H. Li. Vulnerability of Speaker Verification Systems Against Voice Conversion Spoofing Attacks: The Case of Telephone Speech. In *Acoustics, Speech and Signal Processing (ICASSP)*, 2012.

[27] J. Kominek, T. Schultz, and A. W. Black. Synthesizer Voice Quality of New Languages Calibrated with Mean Mel Cepstral Distortion. In *Proc. Inte. Workshop Spoken Lang. Technol. for Under-Resourced Lang.(SLTU)*, 2008.

[28] J. Kreiman and D. Sidtis. *Foundations of Voice Studies: An Interdisciplinary Approach to Voice Production and Perception*. John Wiley & Sons, 2011.

[29] R. F. Kubichek. Mel-Cepstral Distance Measure for Objective Speech Quality Assessment. In *Communications, Computers and Signal Processing, 1993., IEEE Pacific Rim Conference on*, volume 1, 1993.

[30] A. Kumar, N. Saxena, G. Tsudik, and E. Uzun. Caveat Emptor: A Comparative Study of Secure Device Pairing Methods. In *International Conference on Pervasive Computing and Communications (PerCom)*, March 2009.

[31] P. Ladefoged and J. Ladefoged. The Ability of Listeners to Identify Voices. *UCLA Working Papers in Phonetics*, 49, 1980.

[32] S. Laur and K. Nyberg. Efficient mutual data authentication using manually authenticated strings. In *Cryptology and Network Security (CANS)*, 2006.

[33] S. Laur and S. Pasini. SAS-Based Group Authentication and Key Agreement Protocols. In *Public Key Cryptography*, 2008.

[34] R. Nithyanand, N. Saxena, G. Tsudik, and E. Uzun. Groupthink: Usability of Secure Group Association of Wireless Devices. In *International Conference on Ubiquitous Computing (Ubicomp)*, September 2010.

[35] S. Pasini and S. Vaudenay. An Optimal Non-Interactive Message Authentication Protocol. In *CT-RSA*, 2006.

[36] Petraschek, Martin and Hoeher, Thomas and Jung, Oliver and Hlavacs, Helmut and Gansterer, Wilfried N. Security and Usability Aspects of Man-in-the-Middle Attacks on ZRTP. *J. UCS*, 14(5), 2008.

[37] I. Rec. P. 800: Methods for Subjective Determination of Transmission Quality. *International Telecommunication Union, Geneva*, 1996.

[38] P. Rose. *Forensic Speaker Identification*. CRC Press, 2003.

[39] Stanislaw Jarecki and Nitesh Saxena. Auhenticated Key Agreement with Key Re-Use in the Short Authenticated Strings. In *Conference on Security and Cryptography for Networks (SCN)*, September 2010.

[40] D. Sundermann, H. Hoge, A. Bonafonte, H. Ney, A. Black, and S. Narayanan. Text-Independent Voice Conversion Based on Unit Selection. In *Acoustics, Speech and Signal Processing (ICASSP)*, 2006.

[41] D. Sundermann, H. Ney, and H. Hoge. VTLN-Based Cross-Language Voice Conversion. In *Automatic Speech Recognition and Understanding, 2003. ASRU'03. 2003 IEEE Workshop on*, 2003.

[42] I. Szoke, P. Schwarz, P. Matejka, L. Burget, M. Karafiát, M. Fapso, and J. Cernocky. Comparison of Keyword Spotting Approaches for Informal Continuous Speech. In *Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, 2005.

[43] T. Toda, A. W. Black, and K. Tokuda. Acoustic-to-Articulatory Inversion Mapping with Gaussian Mixture Model. In *INTERSPEECH*, 2004.

[44] T. Toda, A. W. Black, and K. Tokuda. Spectral Conversion Based on Maximum Likelihood Estimation Considering Global Variance of Converted Parameter. In *Proc. ICASSP*, volume 1, 2005.

[45] 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 Transactions on*, 15(8), 2007.

[46] T. Toda, A. W. Black, and K. Tokuda. Statistical Mapping Between Articulatory Movements and Acoustic Spectrum Using a Gaussian Mixture Model. *Speech Communication*, 50(3), 2008.

[47] A. R. Toth and A. W. Black. Using Articulatory Position Data in Voice Transformation. *ISCA SSW6*, 2007.

[48] H. Tsuboi and Y. Takebayashi. A Real-Time Task-Oriented Speech Understanding System Using Keyword-Spotting. In *Acoustics, Speech, and Signal Processing*, 1992.

[49] E. Uzun, K. Karvonen, and N. Asokan. Usability analysis of secure pairing methods. In *Financial Cryptography and Data Security*, 2007.

[50] J. Valkonen, N. Asokan, and K. Nyberg. Ad Hoc Security Associations for Groups. In *Security and Privacy in Ad-Hoc and Sensor Networks (ESAS)*, 2006.

[51] S. Vaudenay. Secure Communications over Insecure Channels Based on Short Authenticated Strings. In *Advances in Cryptology - CRYPTO 2005*, 2005.

[52] F. Yang, E. Shechtman, J. Wang, L. Bourdev, and D. Metaxas. Face Morphing Using 3D-aware Appearance Optimization. In *Proceedings of Graphics Interface 2012*, GI '12, 2012.

[53] H. Ye and S. Young. High Quality Voice Morphing. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 1, 2004.

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