Cocaine Noodles: Exploiting the Gap between Human and Machine Speech Recognition*

Tavish Vaidya
Georgetown University

Yuankai Zhang
Georgetown University

Micah Sherr
Georgetown University

Clay Shields
Georgetown University

Abstract

Hands-free, voice-driven user input is gaining popularity, in part due to the increasing functionalities provided by intelligent digital assistances such as Siri, Cortana, and Google Now, and in part due to the proliferation of small devices that do not support more traditional, keyboard-based input.

In this paper, we examine the gap in the mechanisms of speech recognition between human and machine. In particular, we ask the question, *do the differences in how humans and machines understand spoken speech lead to exploitable vulnerabilities?* We find, perhaps surprisingly, that these differences can be easily exploited by an adversary to produce sound which is intelligible as a command to a computer speech recognition system but is not easily understandable by humans. We discuss how a wide range of devices are vulnerable to such manipulation and describe how an attacker might use them to defraud victims or install malware, among other attacks.

1 Introduction

The HAL 9000, an artificial intelligence from Arthur Clark's famous 1968 novel [3], helped popularize the notion of a machine that could understand human speech. Despite the lack of voice recognition systems that approach the accuracy of the HAL 9000, speech recognition is now itself ubiquitous, present on hundreds of millions of smartphones and wearable devices, and is likely to become increasingly prevalent in the future.

Speech recognition is often the most convenient form of input for smaller devices such as portable phones, and can be the only manner of providing input for wearable devices that are too small to support either physical or on-screen keyboards. These devices tend to be ones that users carry and use on an everyday basis and which therefore contain a large amount of personal or confidential information, such as personal communications, billing or authentication credentials, or the user's location.

Speech recognition is a complex process with many techniques borrowing heavily from biological inspirations (e.g., the use of deep learning neural networks). It *mimics* how humans process language, applying complex transformations to convert analog signals into digital representations and artificial intelligence techniques to organize those representations into phonemes and hence into speech. These techniques are, at best, approximations of the physiological processes that govern how humans interpret speech, and humans are currently better at parsing speech than machines. Comical errors in speech recognition are memes on sites such as Reddit, and make it clear that there is a difference in how humans and machines interpret speech.

This paper explores this gap between the synthetic and natural and poses the question, *do the differences in which humans and computers understand spoken speech lead to exploitable vulnerabilities?* Perhaps surprisingly, our initial findings strongly indicate that the answer is "yes".

The question that motivates this work is whether there is audio that is interpreted as human speech by machines but is perceived by humans as noise other than speech. In the remainder of this paper we demonstrate that producing such audio is both possible and practical.

The man-in-the-elevator attack. The security implications of a "language" that can be understood by computer speech recognition engines but not by humans are somewhat subtle but important[1]. Current digital assistants

*As others have pointed out [2], when spoken, "cocaine noodles" is often (mis)interpreted by Google Now as "OK Google", a term used to activate Google's digital personal assistant.

[1]This paper is admittedly English-centric. Evaluating the effectiveness of our techniques for other languages—especially tonal languages

such as Siri and Google Now lack biometric authentication and make no distinction between commands issued by their owners and those issued by unauthorized individuals. However, the user's ability to hear spoken commands from others at least serves as a detection technique and allows the user to take mitigating action should others attempt to activate their device. When sounds that cannot be easily recognizable by humans as being speech are interpreted by a device, the opportunities for *undetected* unauthorized access to voice commands increases.

As one concrete example, we consider a *man-in-the-elevator* (MitE) attack. Here, the user enters an elevator with a device that supports continuous speech recognition. During the elevator ride a sound plays on the elevator's loudspeaker which is unrecognizable as speech to the user. The victim's device, however, interprets the sound as a voice command. The attacker crafts these commands to perform actions to his benefit. This might include sending a text to an SMS short code, allowing the adversary to monetize his attack by charging fees for the text message, or it might cause the device's browser to open a webpage containing a drive-by download, compromising the device itself and allowing the attacker future access to the data within.

We note that this attack has the possibility of scaling significantly outside the elevator. For example, these human unrecognizable commands could be embedded within audio or video segments on radio, television, or Internet viral videos to reach the devices of many millions of people. An authoritarian government, when faced with a large group of demonstrators, could play such sounds over loudspeakers to cause the demonstrators' devices to identify them to the authorities.

General approach. Most generally, the gap between human and machine speech recognition leads to an unmonitored channel by which an adversary can inject commands. This paper demonstrates the feasibility of such channels.

A key challenge in implementing this attack is that speech recognition is not uniform and that different systems may apply various methods to convert raw audio signals into phonemes and then into speech. These systems are usually proprietary with little public information available about how they function. The actual interpretation of audio is often performed remotely "in the cloud", leaving little opportunity for reverse engineering on the devices themselves. This makes targeting the specific recognition methods difficult.

such as Chinese, Vietnamese, and Thai—is an interesting area of future work.

This paper presents a general approach towards converting audio commands into a form that is recognized by computers but not by their users. Our techniques are agnostic to the particular machine learning approach used by a given speech recognition system. We instead target the amount of information available to the system for carrying out the speech recognition task. Conceptually, our methodology takes as input a waveform of human speech, extracts acoustic information from the input, then outputs a signal that has sufficient acoustic features for the speech recognition system to understand the intended command but which is lossy enough that it cannot be easily understood by human listeners. In the remainder of this paper, we detail this methodology and show via quantifiable metrics as well as user testing that such attacks are both effective and practical.

2 Threat Model

Our attack targets any device that is actively listening for voice input in an area where we can introduce an audio signal. In general, we expect that these are most likely to be smartphones, tablets, and wearables that the user carries into a physical space where the attacker can place a speaker.

Many devices already continuously listen for voice activation commands: when plugged in, the iPhone responds to "Hey, Siri", as does the iWatch whenever the wearer's wrist is raised; many Android smartphones have the option of continuously listening for "OK Google"; the desktop Google Chrome browser also has the ability to continuously listen for "OK Google" when the user is visiting a Google search page; Android Wear smart watches also listen for voice activation; certain GPS devices such as Garmin's nüvi line respond to voice commands by default; and the Amazon Echo is actively marketed as having the ability to always listen for its activation phrase, "Alexa". Given the rapid adoption of voice recognition systems it is reasonable to assume that the trend of devices continuously listening for spoken activation phrases will continue to expand in the future.

As discussed in §3, there are a wide variety of methods to perform speech recognition; we treat the AI components of the speech recognition process largely as a black box, and make few assumptions about the adversary's knowledge regarding the methods by which the target device interprets speech.

The adversary's goal is to exploit the target device's speech recognition system to cause the device to execute unauthorized commands. The adversary wishes to avoid detection by the user and therefore produces only man-

gled commands that are unintelligible to the device's operator but which are intelligible to the speech recognition system. Since most speech recognition systems apply band-stop filters to attenuate signals that fall outside of the range of human speech and require a minimum power level for parsing speech, the adversary does not attempt to construct a covert audio channel that cannot be perceived by the human ear. Consequently, our mangled audio signals fall within the same frequency band as human speech. We leave open the possibility that the human operator will hear the unintelligible audio produced by the attacker. We posit that in even the most quiet office environments, occasional electronic-sounding noises that cannot be easily identified are not uncommon, and are almost always ignored.

We assume that the adversary is physically proximate to the target and has the capability of playing a sound at reasonable volume ($\sim$70db, as perceived by the device's microphone(s)). We note that proximity can be extended by the use of devices such as a Long Range Acoustic Device (or LRAD), which produces the requisite sound levels over many hundreds of meters. Our proposed attack may be detected if the smartphone screen is visible to the user during the attack.

Corresponding to the vast majority of current-generation devices, our attacks target devices that do not apply biometrics or otherwise attempt to authenticate the speaker of the voice commands. We assume a human operator who is not using his device at the time of the attack and therefore may not notice any on-screen notifications that reveal the adversary's commands. Finally, we note that the attack may be targeted towards a particular device, or broadcast over a wide area to affect multiple devices.

Attack utility. An adversary who is able to cause a target device to execute voice commands may leverage this capability to achieve the following goals: (this list is not intended to be exhaustive)

- *Initiate a drive-by-download:* An attacker can issue commands to open a webpage maintained by the adversary that contains a drive-by-download. This effectively serves as a stepping stone, enabling other attacks that exploit vulnerabilities in the device's browser;
- *Earn money via pay-based SMS services:* An attacker can construct audio that causes phones to send text messages to pay-based SMS short code services that it operates;
- *Enumerate devices in a physical area:* Similarly, an attacker may use a loudspeaker to cause nearby phones to send SMS messages to a number that the adversary controls, allowing it to enumerate the devices that are physically located within "earshot" of the broadcast (e.g., those belonging to dissidents attending a rally);
- *Earn money via premium rate services:* An attacker can operate a premium rate number (i.e., a "900 number") and monetize his attack by causing nearby phones to call it (for some mobile devices and calling plans);
- *Perform a denial-of-service attack:* Using a public announcement system, an attacker can issue commands to turn on airplane mode on all devices, preventing them from receiving calls and other communications.

We note that the adversary may be required to issue multiple voice commands to active the speech recognition system ("Hey Siri") and launch his attack ("open myevilsite.com").

As more devices adopt voice activation and speech recognition capabilities, we anticipate that the security implications of our techniques will further increase.

3 Speech Recognition Overview

Here we briefly describe the steps involved in speech recognition to familiarize the reader with the topic and to better explain the attack we present in §4. We focus exclusively on speaker-independent speech recognition systems that are designed to interpret any speaker's spoken commands without requiring training data from individual users. These are the systems most often employed by the smartphones, tablets, wearables, and other devices that the attacker might target. Interested readers may refer to more detailed readings for additional background on speech recognition [9, 12, 16, 18].

Figure 1 shows the steps of a typical speaker-independent speech recognition system. The first step (*pre-processing*) usually involves removing background noise, filtering frequencies that are of little or no relevance for speech recognition, and eliminating parts of the input signal that fall below an energy threshold. This process is generally referred to as speech/non-speech segmentation. As briefly discussed in §2, a consequence of this segmentation process is that it generally disallows covert channels between the adversary and the device that cannot be perceived by the human operator. Conversely, it does *not* filter signals that are not human understandable, so long as those signals have the requisite energy level and fall within the frequency range of human speech.

The filtered audio signal is then processed to extract

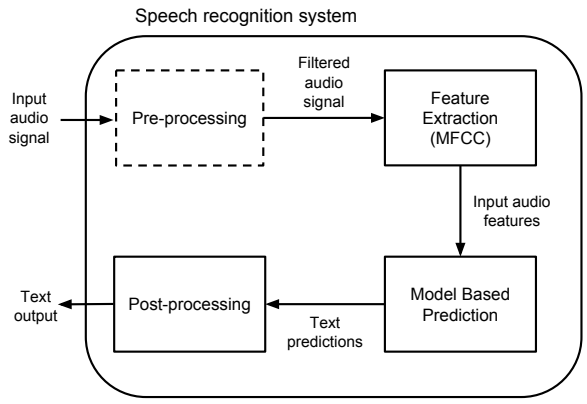


Figure 1: Workflow of a typical speech recognition system.

acoustic features useful for recognizing speech (*feature extraction*). The input signal is converted from the time domain into the spectral domain by considering uniform length time frames and performing a fast Fourier transform (FFT) over each frame. Most speech recognition systems use Mel-frequency cepstral coefficients (MFCC) to represent acoustic features of the input audio signal. MFCC closely approximates a human response to auditory sensation and allows for better representation of sound [10, 14].

The acoustic features extracted from the input signal are then matched against an existing model, built offline using training data. Speech recognition models are typically constructed using statistical approaches. In particular, Hidden Markov Models and artificial neural networks are often used to map features to phonemes, and then text. Based on the acoustic features of the input, the speech recognition model generates text predictions. A post-processing step may then be performed to rank the text predictions by employing additional sources of information—for example, enforcing grammar rules or considering the locality of words.

MFCCs. As discussed in the following section, our attack leverages the common use of MFCCs to extract features from audio signals. In what follows, we present additional detail on MFCCs that may be useful to understand our attack.

Mel-frequency cepstral coefficients (MFCCs) are used to represent the short-term power spectrum of audio on a nonlinear mel frequency scale. MFCC is based on human hearing perceptions with frequency bands equally spaced on the mel scale (as compared to linearly spaced frequency bands in normal cepstrum) [14]. MFCCs are commonly derived by first taking the Fourier transform of a windowed excerpt of the input signal and mapping the powers of the spectrum obtained onto the mel scale. Next, log of powers for each frequency on the mel scale is taken followed by the discrete cosine transform of each mel log powers. The amplitudes of the resulting spectrum are the MFCCs. (Muda et al. [14] provide a more detailed discussion on computing MFCCs.)

Various parameters can be tuned for computing MFCCs: the window length to consider for computing the Fourier transform, the spacing between two windows, the number of warped spectral bands to use, the number of cepstral coefficients to return, and the lowest and highest band edge of the mel filters.

Our proposed attack involves tuning various parameters to compute MFCCs for unmodified input audio, and then reconstructing a modified audio signal from MFCCs. The MFCC parameters are tuned in a way that they preserve enough audio features for the speech recognition system to correctly predict the corresponding text but changes the audio signal as perceived and understood by humans. We discuss this process in much more detail next.

4 Audio Mangling

Speech recognition systems rely on acoustic features extracted from input audio for generating text predictions. As long as the input audio contains enough acoustic information (above a certain threshold depending upon the sensitivity of the targeted system to noise, etc.), the system can correctly recognize the corresponding text with fairly high accuracy.

Our attack modifies the input audio signal in such a way that the mangled audio output retains enough acoustic features for the speech recognition system to correctly predict the text while making it difficult for humans to understand the mangled audio signal.

Figure 2 shows the outline of our attack. An audio command is given as an input to the *audio mangler*. The mangler then produces a morphed version of the original input. The morphed audio signal retains sufficient acoustic features of the input audio command to allow speech recognition systems to interpret it, but sounds very different than the original command to humans. This new *attack command* is then later played to the target's speech recognition system, which interprets and executes the initial audio command. The human user does not hear the command as a command, and may not notice the execution proceeding on the device.

Below, we describe in more detail the steps used to mangle audio:

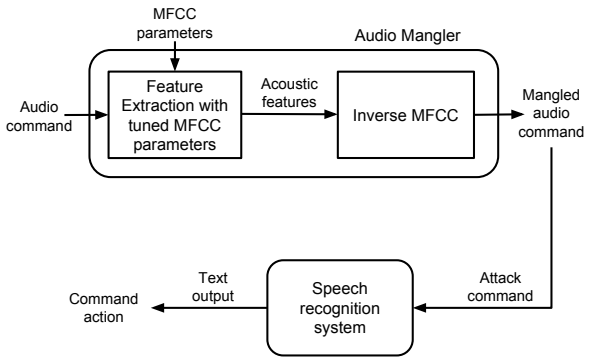


Figure 2: Attack outline.

MFCC parameters. MFCC computation requires various parameters to be specified [7]. We focus on four independent parameters: `wintime`, `hoptime`, `numcep` and `nbands`. `wintime` determines the length of the timeframe over which the signal is considered as being statistically constant; `hoptime` determines the size of the time step between successive windows; `numcep` is the number of cepstra, i.e., the number of coefficients to output; and `nbands` is the number of warped spectral bands to use while aggregating energy levels for closely spaced frequencies.

We experimentally observed the effect of changing each of these parameters independently on the perceived quality of mangled audio output. To check if the mangled audio can be understood by a speech recognition system, we submitted the outputs of the audio mangler to Google's Speech Recognition engine using a publicly available API [15]. The API returns a list of up to five possible transcriptions with a confidence value associated with each transcription. If the API cannot transcribe the audio, an error is returned.

Based on the edit distance (see §5) between the transcriptions of un-mangled and mangled audio signal, we manually narrow down the value range for each parameter. The range of parameter values, thus chosen, will produce mangled audio output with minimal but sufficient acoustic information for a speech recognition system to transcribe the audio while making it non-trivial for a human listeners to understand.

Feature extraction with tuned MFCC parameters. After experimentally determining the range of MFCC parameter values, acoustic features are extracted by computing MFCCs [7] of the input signal using the chosen MFCC parameters. Computing MFCCs is lossy: the process considers the signal to be statistically constant over a small time window and also aggregates the energy levels of closely spaced frequencies to represent the total energy in various frequency regions on the mel frequency scale. (The aggregation of energy level is motivated by the human hearing mechanism as it cannot discern differences between closely spaced frequencies and the effect becomes more dominant as the frequencies increase.) Thus, MFCCs do not retain all the information about the input audio signal. The tuned MFCC parameters used to create the attack command are intended to further increase this loss of information in way that is deterimental to human understanding.

Recall that MFCCs are representations of acoustic features of an audio signal. We use the tuned MFCC parameters to compute MFCCs of an unmodified audio command. The resulting MFCCs contain just enough acoustic information, ensured by careful selection of MFCC parameters, such that a mangled audio signal reconstructed from them will be correctly recognized by the targeted speech recognition system.

Inverse MFCC. The extracted audio features represented as MFCCs are converted back to an audio signal by reversing the steps of the MFCC computation. The inversion of MFCCs back to a waveform involves the addition of noise, since MFCC computation is lossy and aggregates energy levels of closely spaced frequencies into frequency regions. Inversion from MFCCs to audio signals *mangles* the original audio signals, making them difficult for human listeners to understand.

In summary, our approach modifies the input signal by adjusting MFCC parameters and performing feature extraction, and then reconstituting an audio signal by applying a reverse MFCC to the extracted features. A key property of our approach is that the features used in the speech recognition system remain mostly undisturbed (since they are extracted and then reconstructed), while the non-extracted-features are lost in the reconstruction.

It is worth stressing that the adversary can perform the mangling entirely offline. Recall that the adversary's job is to construct an audio file that is interpreted by computer speech recognition systems, but not easily discernible to humans. He can thus perform the procedure in a trial-and-error fashion, tuning the audio mangler's parameters to generate audio files and testing them manually to see whether the mangled output is accepted by a copy of the targeted speech recognition system and to see if the a human listener deems the mangled audio to be non-understandable. This process can be time-consuming, but producing a *single* mangled audio that achieves the above two constraints will likely lead to a successful attack.

5 Evaluation

The goal of our attack is to produce mangled audio files that activate commands on voice-recognition systems but which are difficult for humans to understand. We evaluate the effectiveness of this attack by implementing a set of mangled commands, verifying that they do activate the phone, and then performing human testing to determine how difficult the commands were for humans to interpret.

Experimental setup. We created four types of commands which cover the attacks described in §2:

- activating the voice command input (i.e., "OK Google");
- calling a number;
- sending a text message to a number; and
- opening a website (tested against two websites)

These commands are listed in Table 1. Each of the four authors provided recordings of each command to use.

We then created an experimental setup to ensure that the commands were correctly understood by the device. For this, and for testing the efficacy of the mangled commands, we tested the audio commands against the Google Now intelligent personal assistant running on a Samsung Galaxy S4 smartphone with Android version 4.4.2. The smartphone was placed on a table with a pair of speakers placed about 30 cm from the phone. The commands were played through the speakers and we determined whether the command activated the corresponding functionality on the smartphone. The experiment was carried out in a large, relatively quiet room with the background noise level averaging 50 dB.

To establish a baseline, unmangled versions of all five attack commands were tested using the described setup. All baseline candidates were successful in activating their respective functionalities on the smartphone in our test environment.

Audio mangling. We next implemented our audio mangler in MATLAB2014b using the MFCC implementation by Ellis [7] to mangle the commands described above. After fine tuning the MFCC parameters and generating multiple mangled audio candidates (see §4), two of the authors listened to around 500 potential candidates each and picked a total of 105 candidates to be evaluated by human listeners. The list of attack candidates was then further narrowed down by getting human evaluators' feedback from Amazon Mechanical Turk, with preference given to commands that the evaluators found difficult to understand.

We then tested the selected attack candidates on our test setup. Again, all of the selected attack candidates successfully activated their corresponding functionalities on the smartphone. These preliminary results support the argument that speech recognition systems can understand audio that has been specifically crafted to contain the acoustic feature information as described above. The question then becomes – how well can humans understand them?

Non-comprehension by human listeners. To test how well humans can understand the audio commands, we carried out a study using Amazon Mechanical Turk, a service that allows requesters to set up tasks for workers to complete and offer them a payment for their service.

For our study, workers were presented with a task containing links to four audio commands along with the instructions (see Appendix A). Each command in a task was unique to prevent giving the worker additional information to use to decipher the command. Some commands were the mangled commands created above, while others were unmangled to provide a baseline for measuring worker accuracy. The workers were asked to listen to the audio samples in the task and provide a transcription for each audio sample, as per instructions. Workers were paid $0.11 per transcription and a bonus of $0.05 cents for an accurate transcription. Each worker was allowed to complete only one task to avoid carrying over earlier knowledge to a new task.

While this approach is not the same as a person who might be distracted or otherwise unaware that an audio command was being played, as we assume in our attack model, we argue that this strengthens our argument that understanding a mangled command is difficult. Workers could listen to an audio command multiple times which may have increased the understanding of the audio signal, particularly provided they would receive a reward for correct transcription. The device through which the audio commands are played also affects the level of understanding. Noise cancellation headphones might allow for better perception of audio as compared to computer speakers. Such factors, not within our control, may have improved the transcription results submitted by human evaluators.

We therefore argue that these results indicate a possible best case for human understanding and that an attack victim who has only one attempt to do so at a time when they are not expecting to need to, given possibly adverse listening conditions, would likely perform less well than our evaluators.

Table 1 shows the number of human evaluators who submitted transcriptions for various audio commands. The differences in sample size are due to the order that

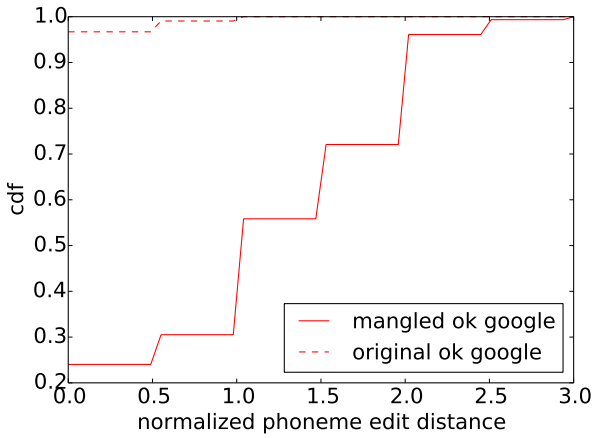


Figure 3: Cumulative distribution of phoneme edit distances, for both the original and mangled audio, for the "Ok Google" command.

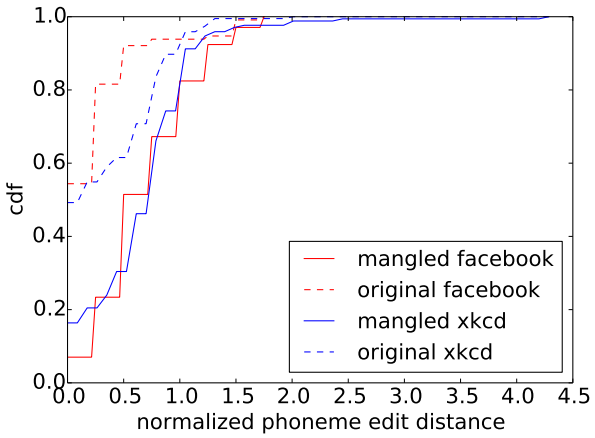


Figure 4: Cumulative distribution of phoneme edit distances, for both the original and mangled audio, for the commands that open Facebook and the xkcd website.

Command	Baseline	Mangled
ok google	212	153
open xkcd.com	195	171
show facebook.com	114	171
call 15559876543	81	114
send text to 1234567890 how are you doing	73	179

Table 1: No. of transcriptions (samples) for voice commands.

tasks were assigned by Amazon Mechanical Turk to its workers, and the number of responses we received to our survey.

Evaluation metric. To quantify human evaluators' understanding of audio commands, we use the Levenshtein edit distance where distance is based on phonemes: Phonemes for all words in the audio commands were generated for the baseline (unmodified audio) as well as mangled audio commands' transcriptions. Using phonemes instead of letters or words as a distance measure has the important advantage that it allows us to compare how close different commands sound rather than how they are spelled. To account for the varying lengths of different audio commands, we normalize the Levenshtein distance by dividing by the length (in phonemes) of the unmodified original audio sample.

Transcriptions to be compared were converted to lowercase and all non-ASCII characters and punctuation were removed. Digits were replaced with their corresponding texts (e.g., "2" became "two"). We used the CMU Pronouncing Dictionary [13], version 0.7, to determine the pronounciation (in phonemes) of words.

Results. First we determined a baseline for human evaluators by calculating the edit distance between phonemes of transcriptions for unmangled audio commands submitted by the workers and the text used to create the commands. This showed the accuracy of transcription when no mangling was present. We then measured the evaluators' understanding of the mangled attack commands by calculating the edit distance between phonemes of transcriptions submitted by the workers and the text used to create the commands. The results show clearly that the approach is promising.

Figure 3, 4 and 5 show the cumulative distribution function of phoneme edit distances for transcriptions submitted by human evaluators. The lines representing baseline and mangled responses for each command correspond to transcriptions of unmodified and mangled commands respectively. Figure 3 shows the results for the "*Ok Google*"[2] command. The *Ok Google* command is required to activate the voice input functionality and opens up the attack surface for other commands to work. A significant gap between the baseline and attack command shows that the majority of human evaluators were unable to understand the mangled command audio. The median of normalized phoneme edit distance for the unobfuscated audio is zero, indicating that most human testers (in fact, more than 95%) were able to perfectly under-

[2]We consider "Okay Google" as a valid transcription and replace *Okay* with *Ok* before computing the edit distance.

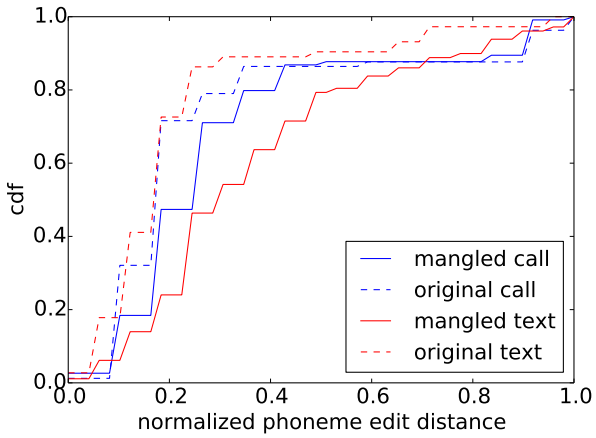


Figure 5: Cumulative distribution of phoneme edit distances, for both the original and mangled audio, for the commands that send an SMS message ("original/mangled text") and place a call ("original/mangled call").

stand it. In contrast, the median of normalized phoneme edit distance for the mangled audio is one (equivalent to two phonemes), which we note is exactly the number of phonemes in "Ok Google"—reflecting a complete misinterpretation.

We achieved similar results for our other tested commands. Figure 4 shows the level of human evaluators' understanding of "*show facebook.com*[3]" and "*open xkcd.com*[3]" commands. Figure 5 shows the result for the "*call 15559876543*" and "*send text to 1234567890 how are you doing*" commands. In nearly all cases, mangling the audio significantly increases the testers' difficulty in correctly transcribing the audio, despite having the ability to listen to the samples multiple times (and having a financial incentive, albeit a small one, to answer correctly). The gap between the baseline and mangled version for these commands is lesser, though significant, as compared to previously described commands, which can be attributed to the contextual information that may have given clues to human evaluators, as well as the length of the sample itself.

Discussion of results. The results above indicate that it is possible to create audio files that activate commands but which are more difficult for users to understand, even in the best case. For each set of commands above, there is a significant gap between the understanding of baseline and mangled command, supporting our hypothesis that is it possible to exploit the gap between humans' and machines' understanding of certain audio commands.

We also note that it appears that the popularity of website may affect results. Higher percentage of human evaluators could guess Facebook, which is ranked second in the Alexa global rankings, as compared to XKCD which is ranked 1,428th.

[3]"Dot" is replaced by "." in transcriptions.

6 Discussion

Suppressing audio feedback. Personal voice assistants like Google Now and Siri provide audio feedback to input voice commands, which might alert an attentive user. However, the audio feedback in response to our attack commands can be suppressed by playing noise via the same means as the attack command, drowning out the device's response. Timing of the audio feedback after issuing a voice command can be determined offline by the attacker and the attack command can be padded with noise such that the delay between the audio command and noise to be played is close to the time delay between the issuing of command and the audio feedback from the voice assistant.

Extending to other speech recognition systems. Our attack is very likely agnostic to any particular speech recognition system since it does not rely on specific internals of any such system and instead depends only on the use of MFCC transformations, a standard practice for speech recognition. We tested our attack on one speech recognition system to demonstrate the feasibility of such an attack. As a part of future work, we aim to extend our attack to other speech recognition systems.

Other attack vectors. In addition to the attack commands already discussed, other voice commands can be exploited to carry out attacks using the voice command input. For example, given a few keywords, Google Now automatically opens up the most relevant webpage when searched via voice commands, eliminating the need to specify a website URL. An adversary can create a web page loaded with malware that combines some number of totally unrelated topics (e.g., radioshack bankruptcy + goat farming), such that querying Google for those terms will ensure that the site is at the top of the list. To avoid Google's malware detectors, it can serve malware only when the browser is detected to be an Android device or connecting from a mobile (e.g., LTE) network.

Google also understands synonyms for commands. For example, *call* can be substituted with *ring* for dialing a number. *Vocabulary expansion*, commonly used in infor-

mation retrieval, can be employed for automatically expanding the command vocabulary and providing an attacker with more potential attack commands.

7 Related Work

Attacks that leverage audio input. The (mis)use of voice input as a vector for attacks has recently received significant attention from both industry and academia. Ben-Itzhak [1] observed that smartTVs and smartphones take synthetic voice commands without authenticating the command issuer. Diao et al. [6] demonstrated an attack in which a malicious app with zero permissions can issue privacy sensitive voice commands to Android smartphones through Google's Voice Search Service. Schlegel et al. [17] describe a sound trojan called Soundcomber on smartphones that stealthily steals private information by listening for tone- and speech-based interactions with the smartphone. Our attack differs from the above as it does not require any malware to be installed on the target device, and instead leverages the intelligent digital assistances that are installed by default on most smartphones and many wearable devices.

Closely related to our work is Jang et al.'s exploration of how accessibility features in modern operating systems can be leveraged to bypass authorization checks [11]. We also use voice to issue unauthorized commands. In this paper, we focus in particular on exploiting the voice input channel and on producing speech that is understandable by computers but not by humans.

Most similar to our work is Esteves et al.'s proposed attack of executing voice commands on smartphones by silently injecting signals via headphones plugged into smartphones [8]. The attack is only effective when the user is using wired headphones with a smartphone that has FM radio capability, and the adversary is within $\sim$2m of the target. The chances of detecting the attack are likely quite high since the phone's voice feedback will be heard by an operator who is wearing the connected headphones. In contrast, our attack plays obfuscated voice commands to the smartphones that sound as gibberish to human listeners, and does not depend on headphones or smartphones' FM capabilities.

Signal pattern matching. Researchers have also explored the use of signal features to identify individual devices and keyboard entries. Zhuang et al. [19] predict contents of typings from the sound of keyboard clicks with the help of machine learning and speech recognition techniques. Dey et al. [5] identify individual smart devices by fingerprinting accelerometer responses to motion stimulation, while Das et al. [4] achieve the same goal by fingerprinting microphones and speakers embedded in smartphones. Our work differs from these approaches in that we are essentially focusing on the opposite—i.e., the manipulation of signal pattern matching to interpret seemingly incomprehensible speech.

8 Conclusion

In this paper, we presented a proof-of-concept attack that exploits the differences in how computers and humans interpret speech. We describe a methodology for transforming speech to a form that can be understood by MFCC-based speech recognition systems while being mostly indiscernible to human listeners. Our initial findings show that it is both possible and practical to activate the voice assistant of an Android phone using mangled audio commands and carry out silent man-in-the-elevator attacks.

Acknowledgments

We thank Eric Burger, Alex Yale-Loehr, Yu Zhang, and Wenchao Zhou for several insightful conversations about this work, and the anonymous reviewers for their helpful feedback. This work is partially funded by the National Science Foundation through grants CNS-1064986, CNS-1149832, CNS-1223825, CNS-1453392, and CNS-1445967. The opinions expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] Y. Ben-Itzhak. What if Smart Devices could be Hacked with Just a Aoice? AVG.Now. Available at `http://now.avg.com/voice-hacking-devices`, 2014.

[2] "Can we change saying 'Ok Google' to '____'?" (Reddit Thread). `https://www.reddit.com/r/AndroidWear/comments/2z18ah/can_we_change_saying_ok_google_to/`.

[3] A. C. Clark. *2001: A Space Odyssey*. Hutchinson, 1968.

[4] A. Das, N. Borisov, and M. Caesar. Do you Hear What I Hear?: Fingerprinting Smart Devices through Embedded Acoustic Components. In *ACM Conference on Computer and Communications Security (CCS)*, 2014.

[5] S. Dey, N. Roy, W. Xu, R. R. Choudhury, and S. Nelakuditi. Accelprint: Imperfections of accelerometers make smartphones trackable. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2014.

[6] W. Diao, X. Liu, Z. Zhou, and K. Zhang. Your Voice Assistant is Mine: How to Abuse Speakers to Steal Information and Control Your Phone. In *ACM Workshop on Security and Privacy in Smartphones & Mobile Devices (SPSM)*, 2014.

[7] D. P. W. Ellis. PLP and RASTA (and MFCC, and inversion) in Matlab. `http://www.ee.columbia.edu/~dpwe/resources/matlab/rastamat/`, 2005.

[8] J. L. Esteves and C. Kasmi. Injection de Commandes Vocales sur Ordiphone. In *Symposium Sur la sécurité des Technologies de l'Information et des Communications (SSTIC)*, 2015.

[9] R. E. Gruhn, W. Minker, and S. Nakamura. *Statistical Pronunciation Modeling for Non-native Speech Processing*. Springer Science & Business Media, 2011.

[10] W. Han, C.-F. Chan, C.-S. Choy, and K.-P. Pun. An Efficient MFCC Extraction Method in Speech Recognition. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, 2006.

[11] Y. Jang, C. Song, S. P. Chung, T. Wang, and W. Lee. A11y Attacks: Exploiting Accessibility in Operating Systems. In *ACM Conference on Computer and Communications Security (CCS)*, 2014.

[12] J.-C. Junqua, J.-P. Haton, and H. Wakita. *Robustness in Automatic Speech Recognition: Fundamentals and Applications*. Kluwer Academic Publishers, 1996.

[13] K. Lenzo. The CMU Pronouncing Dictionary. `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`.

[14] L. Muda, M. Begam, and I. Elamvazuthi. Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques. *Journal of Computing*, 2(3), March 2010.

[15] T. Payton. Google Speech API: Information and Guidelines. `http://blog.travispayton.com/wp-content/uploads/2014/03/Google-Speech-API.pdf`.

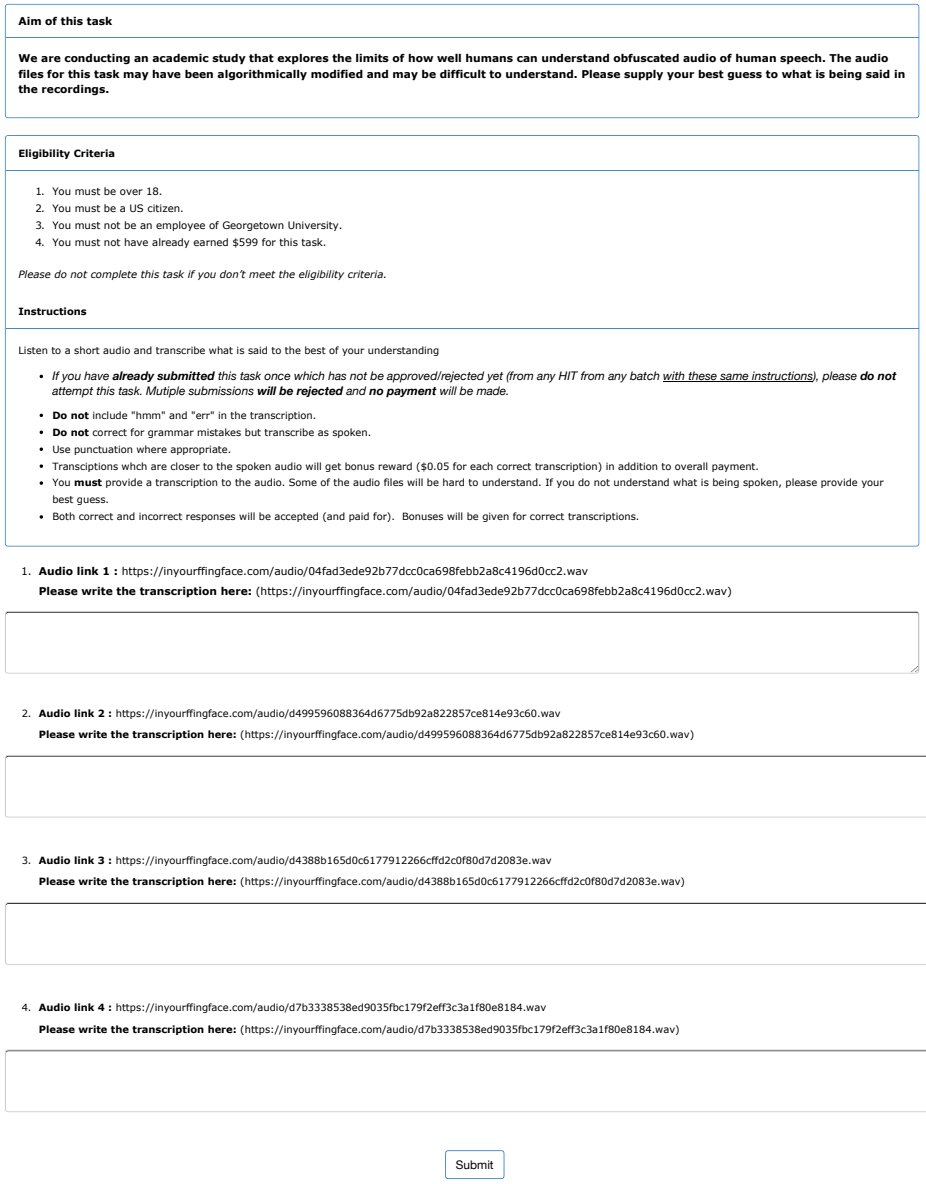
[16] L. R. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*, volume 14. PTR Prentice Hall Englewood Cliffs, 1993.

[17] R. Schlegel, K. Zhang, X.-y. Zhou, M. Intwala, A. Kapadia, and X. Wang. Soundcomber: A Stealthy and Context-Aware Sound Trojan for Smartphones. In *Network and Distributed System Security Symposium (NDSS)*, 2011.

[18] E. Trentin and M. Gori. A Survey of Hybrid ANN/HMM Models for Automatic Speech Recognition. *Neurocomputing*, 37:91–126, 2001.

[19] L. Zhuang, F. Zhou, and J. D. Tygar. Keyboard Acoustic Emanations Revisited. *ACM Transactions on Information and System Security (TISSEC)*, 13(1):3, 2009.

A Amazon Mechanical Turk Survey



Aim of this task

We are conducting an academic study that explores the limits of how well humans can understand obfuscated audio of human speech. The audio files for this task may have been algorithmically modified and may be difficult to understand. Please supply your best guess to what is being said in the recordings.

Eligibility Criteria

1. You must be over 18.
2. You must be a US citizen.
3. You must not be an employee of Georgetown University.
4. You must not have already earned $599 for this task.

Please do not complete this task if you don't meet the eligibility criteria.

Instructions

Listen to a short audio and transcribe what is said to the best of your understanding

- *If you have **already submitted** this task once which has not be approved/rejected yet (from any HIT from any batch with these same instructions), please **do not** attempt this task. Mutiple submissions **will be rejected** and **no payment** will be made.*
- **Do not** include "hmm" and "err" in the transcription.
- **Do not** correct for grammar mistakes but transcribe as spoken.
- Use punctuation where appropriate.
- Transciptions whch are closer to the spoken audio will get bonus reward ($0.05 for each correct transcription) in addition to overall payment.
- You **must** provide a transcription to the audio. Some of the audio files will be hard to understand. If you do not understand what is being spoken, please provide your best guess.
- Both correct and incorrect responses will be accepted (and paid for). Bonuses will be given for correct transcriptions.

1. **Audio link 1 :** https://inyourffingface.com/audio/04fad3ede92b77dcc0ca698febb2a8c4196d0cc2.wav
   Please write the transcription here: (https://inyourffingface.com/audio/04fad3ede92b77dcc0ca698febb2a8c4196d0cc2.wav)
2. **Audio link 2 :** https://inyourffingface.com/audio/d499596088364d6775db92a822857ce814e93c60.wav
   Please write the transcription here: (https://inyourffingface.com/audio/d499596088364d6775db92a822857ce814e93c60.wav)
3. **Audio link 3 :** https://inyourffingface.com/audio/d4388b165d0c6177912266cffd2c0f80d7d2083e.wav
   Please write the transcription here: (https://inyourffingface.com/audio/d4388b165d0c6177912266cffd2c0f80d7d2083e.wav)
4. **Audio link 4 :** https://inyourffingface.com/audio/d7b3338538ed9035fbc179f2eff3c3a1f80e8184.wav
   Please write the transcription here: (https://inyourffingface.com/audio/d7b3338538ed9035fbc179f2eff3c3a1f80e8184.wav)

Submit

Amazon Mechanical Turk survey as seen by human evaluators. Our findings assume that workers properly followed the instructions. We were able to verify that there were no duplicate submissions by any single worker.

B Survey Results

The following are the transcriptions provided by human evaluators (Amazon Mechanical Turk) for all baseline and mangled audio commands. For baseline commands, identical transcriptions have been grouped together and are shown only once followed by their respective frequency count. All transcriptions for mangled commands are shown individually.

***"Ok Google"* baseline**
Okay go go : 1 | Ok google : 10 | Okay Google : 15 | okay Google : 3 | Okay Google : 42 | Ok hoogle : 1 | ok Google : 4 | OK google : 2 | OKAY GOOGLE : 2 | Ok gogo : 1 | Ok google : 1 | OK Google : 20 | Okay google : 17 | OK Google : 3 | Ok Google : 5 | OK GOOGLE : 1 | Ok Google : 30 | Okay google : 3 | okay google : 11 | Okay gulu : 1 | MALE: Okay Google : 1 | ok google : 38

***"Ok Google"* mangled**
from hundrdd tutra | Ok Move up! | cookie coo coo | Country coo coo | I can't go back | Okay cool cool | ohay uh ha | People hall | Hoo Hee Hoo Haw | cookie crew crow | ok cocoa | Okay Todo | Open Google | Seek approval | hey cocoa | okay come on | Foo he who hall | a I U A | Okay click here | Okay google | ho he ho ha | her e ewe ah | OK let's go | coffee coo coo | okay coco | We're equal all | i hate you all | ok google | Classy supper | ho hee hoo ha (laughter) | oh-eeooh-aah | I don't need no more | call 985- 887-6543 | okay google | call jay can't come | oh hey hoogle | O e uu ah | AH HEAH HO! | No test (Sound of someone breathing) | Cookie crumble | ok google | hoo hee hoo hoh | Okay Google | ho he who ho | her he hu ha | cookie go glum | Hockey cocoa | the people all | oww he ah ha | Coochee coo coo | ku ki ku ku | Ok who hull | clue clue | effacing hulk | the evil hawk | cokeh coo coo | coe kee coo coo | FOUR PEOPLE CALLED | puchee poo poo | Ok google | Searching google | Who He Who Ha | coo- coo | OK cool cool | OK Cocoa | Oh ee oh ah ah | How'd she come home? | okay cool cool | huhehuha | ah ee ooh ahh | Okay who all | I can conclude | Sochi coo coo | hah hee ho hah | Cookies for all | hu he ho ha | Okay coo-coo | it's a good one | co kay coco | Her he you all | oo ee oo aa aa | (inaudible) | Ok Google | Cookie Kookoo | ah gay HOO haw | katay caw caw | copy google | uh hi ho ho ha | Ho eh ho ha | Her evil heart | Cutting to to | Cookie couture | cook a cocoon | okee goocal | Okay crow crow | cookie cuckoo | Claw clay clue clue | ha he ho ha | Okay Coco | HER HE OOHALL | Coochie koo koo | Okay co quo | Okay cuckoo | OKAY GOOGLE | (empty) | I'll get your coat | her gable | OK Google | ho hi ho ho | hohiho | I pick her up | Open google | Coco | ok cool cool | Audio | ka ki ko come | ooh eee ooh ough | Okay go go | merci bouque | her he hoo haw | Inaudible

***"show facebook.com"* baseline**
Show crisper conduct.co : 1 | Show Facebook.com. : 8 | the show icebuck.com : 1 | so how do you spell conduct : 1 | shawn how do you spell .com : 1 | SHOW FACEBOOK DOT COM : 1 | facebook dot com : 1 | show Facebook.com : 2 | Show facebook.com : 15 | Show facebook dot com : 1 | so facebook dot com : 1 | Show arisbook.com : 1 | so facebook.com : 1 | Go facebook dot com : 1 | ishribac.com : 1 | show icebook.com : 1 | Show ChrisBuck.com : 1 | Show icebook.com : 1 | Show Facebook dot com. : 1 | showfacebook.com : 1 | Show how do you spell conduct-com? : 1 | facebook.com : 3 | show facebook dot com : 4 | show facebook dotcom : 1 | show his book dot com : 1 | Show fistbuck.com : 1 | From facebook.com : 1 | Show adisbook.com : 1 | Show crisspoc dot com. : 1 | showaddthisbook.com : 1 | Show icebuck.com : 1 | Show frisbuck.com : 1 | Show iceberg dot com : 1 | Show icebook.com. : 1 | facebook : 1 | Show facebook dot com. : 1 | show facebook.com : 15 | show fist buck dot com : 1 | Show facebook.com. : 1 | show Facebook dot com : 1 | Show Facebook dot com : 5 | So how do you spell conductcom? : 1 | so this book can not cop : 1 | MALE: So I spoke to the doctor. : 1 | shaoicebock.com : 1 | follow facebook.com : 1 | show chrisbook.com : 1 | Facebook.com : 1 | showicebuck.com : 1 | show facebook dot com. : 1 | So I spoke it don't come : 1 | Show Facebook.com : 15 | Showicebucket.com : 1 | Show Facebook dot come : 1 | show this book.com : 1

***"show facebook.com"* mangled**
score facebook in dot com | Schwab Facebook King dot com | fwahsbuckle.com | show pressbuttle.com | Crisper conduct dock. | Facebook.com | for hush puppy .com | far sqvoo.com | show icebottle.com | sha respeckin da karr | Shaw facebooking.com | I saw a spot through that car | If I facebook dot com | small press button dot com | Fah whisper to that plan. | show facebook dot com | Wow this pretend a dog. | respond read it dot com | speaking dot gov | It's called Facebooking dot com. | Stop respecting dot com. | farpressbutton.com | It's far for wisp.com | Swah press button dot com | how do you spell .com | far is the conductor | facebook | The swamp has woken back up. | follow facebook.com | schwabwishpadotcom | for hush puppy.com | it's on baseputting.com | For face dot google dot com | for place spectrum.com | Just want to piss into that cup. | Call 1459876543. | Find Facebook.com. | Show Facebook dot com | Find icebook.com | For crisper in that cup. | swere dis button da gah | shaw facebookforyou.com | JOHN HAS GOTTEN THE GUN | Press button to play. | I'm on facebook.com. | It's one on facebook.com | The friend does not come. | Show facebook.com | (speech is garbled) essa tres quatro dot come | Show pressbutton.com | theswanisworking.com | Stop respecting the cop. | Small hairs button dot car. | the audio quatro dot com | It's on pricebooking.com. | far slooq.com | Fly and I will kingdom come | show ice bucket.com | Try bestbooking.com | Small hush puppy does take. | I was expecting that guy | From facebook.com | File facebook.com | Squirrel facebooking.com | ... facebooking.com | the car keys are in the car. | Afar expecting that call | Schwa fits the conductor. | For FaceBook-ing dot com. | Bashwhaa gastonin cua | its from pressbutton.com | The swamp has freaking got cold. | follow prospecting.com | Show Facebook.com | four three.com | file pistro es ta caro | spell facebook.com | facebooking.com | for facebooking.com | Islam does nothing but cody. | icebooking.com | shwaa prescription dot got. | best price booking dot com | Farfitsparten.com | Swab. Press button.com. | afuera es punto que esta caro | Find Krispy Kreme duck traffic. | (inaudible).com | Ha bring the car | The file. pressbutton.com | a schwa! hush puppy got frightened! | show facebook.com | show pressbutton.com | follow facebooking.com | farmfreshtoyou.com | forrestbucking.com | far press button dot com | shwapressbutton.com | Show icebuck.com | Find Facebook Dot Com | Press button .com | Find my car. | swab has button dot com | press button dot com | On facebook.com | a fall has broken that cup | *unintelligible* dot com | Cow is spotting the cow | for christmas do not come | find facebook.com | farfacebook.com | It's hard respecting the Kahl. | fasmathatude.com | slash facebook dot com | facebook dot com | If I push button top down | (Unintelligible breathing) | Show icebucket.com | For Facebook.com | FarPlaceForKing.com | facebookking.com | forum facebooking.com | schwa facebooking.com | SHOW FACEBOOK-ING DOT COM | sh facebook got it | Follow facebook.com | SchwabKrispyKreme.com | facebook.com | Fall in place. | isha frishbitu.com | scroll facebooking dot come | ?? | sawyourface.com | faforspotting dot com | shaw facbookforyou.com | For Facebooking.com | the far face protrudes through the cloud | call press button dot com | For Facebook conductor. | straw fresh booking dot com | ISH LA FRESH BUTTON DOT COM | squ hush swing co hwah | Ish maro press back to dot cottle | fshfshwfshw | Who respects the doctor! | ishrah frishbictu.com | followchristopher.com | Is shaw fice bakul dot com? | Well I'm expecting the girl. | Krisper conduct cop. | Crispycream.com | brushra pruducha man

***"open xkcd.com"* baseline**
open extracd.com : 1 | Open exkecd.com : 1 | Open XDCD.com : 1 | excessidy dot com : 1 | Open xkc.com : 1 | Open excesseedee dot com. : 1 | cd don't come : 1 | open exe cd dot com : 1 | Open X K cd dot com. : 1 | Open xk cd.com : 1 | Open ecstasy.com : 3 | XKCD.com : 1 | Open accessidy.com : 1 | open excasedy.com : 1 | open xk cd.com : 1 | openextensity.com : 1 | open ecstasy.com : 1 | Openxtc.com : 1 | open exscamecd.com : 1 | open skcd.com : 4 | Open xcedecy.com : 1 |

open xcd.vom : 1 | open xkzd.com : 1 | open xkcd dotcom : 1 | Open Excise CD.com : 1 | open exkesity.com : 1 | Open XKCD dot com. : 1 | Open xkcd.com. : 5 | Open excasity.com : 2 | Open exkay cd dot com : 1 | open excasy dot com : 1 | Open Sk cd dot com : 1 | Open X K C D .com : 1 | open xkcd dot com : 6 | Open EXTA.CD.com : 1 | Open xguesscity.com. : 1 | open xCD.com : 1 | open xkc dot com : 1 | open ectasy.com : 1 | open exscamcd.com : 1 | Open xk do dot com : 1 | open xcancd.com : 1 | Open xcd.com : 1 | open excasidy.com : 1 | Open XKZD.com : 1 | open..XKCD.com : 1 | open escascity.com : 1 | Open xkcd dot com : 2 | Open XKCD.com : 11 | Open xkcd.com : 1 | Open exticity dot com. : 1 | Open XKCD cot com : 1 | Open extesscd.com : 1 | Open. Ecstasy.com. : 1 | open skcd dot com : 2 | open XKCD dot com : 1 | open xkcd.com : 42 | MALE: Open xkcd.com. : 1 | Open XKCD.com. : 4 | open xkcd dot com. : 1 | open exkaycd.com : 1 | Open escacity.com : 1 | Open XKCD.com. : 1 | Open xcd.com. : 1 | open extasy.com : 1 | Open escasidy.com : 1 | OPEN X K C D DOT COM : 2 | Open X-K-C-D dot com : 1 | Open X K CD don't come. : 1 | Open excacd.com. : 1 | open etsy.com : 1 | openxkcd.com : 1 | open xkcd .com : 1 | Open ask a city.com. : 1 | open ecstacity dot com : 1 | open askacity.com : 1 | open escaseedy.com : 1 | open ecocity.com : 1 | open [xx].com : 1 | Open x-k c-d dot com : 1 | Open Xk cd.com : 1 | open excesscity.com : 1 | open k x cd dot com : 1 | open exuCD.com : 1 | Open XKcd.com : 1 | Open ecstacy.com : 1 | open X K CD.com : 1 | Openxkcd.com : 1 | openskcd.com : 1 | open sk cd dot com : 1 | open acces cad cd dot come : 1 | Open xkcd.com : 25 | Open ex cd dot com : 1 | Open X K C D dot com : 1 | open..xkcd.com : 1 | open: excasedy dot com. : 1 | Open eskesity.com : 1 | Open excacity.com : 1 | Open xkscd.com. : 1

"open xkcd.com" **mangled**

we're going in Eric's little car | open ecstasy.com | Okay Miss Keech did you go? | Ok Miss Casey G I'm coming. | Kitchen Key.com | open @ Stephanie.com | open skg.com | Open skcd.com | Moving. (Unintelligible breathing) | Open skgd.com | Open SKGC don't come. | Open extrecity.com | ok lets get this going | sptc.com | Put down the kcg | Open Astesity.com | Okay it's act.com. | Open xkcd.com. | well planned cassidy dot com | Open this case CD right now. | Open this case little duffel. | fine then its cause im falling down | logan at scare city dot com | Broken existing dot com. | hoping explicity.com | open s k c t dot com! | openextensity.com | In this case it did not come through. | open x-p c-d dot com | Open this kids car | open skcd.com | open explicity dot com | Open his evil coffin. | open..xkd .com | open exceeding.com | OPEN EXTA CITY DOT COM | open ecstasy.com | Open excasity.com | Open xkcd.com | Open extendcd.com. | act dot com | Look at extensity.com! | Open ecstacy dot com | Open (inaudible).com | spell extacy.com | Open this case d dot.com. | open xkcd.com | openskcd.com | open xk cd.com | open ecaxtcd.com | Open excaziti.com. | open xkc dot com | open sagd.com | Okay. SKGT.com | open exscamcd.com | 14shg.com | casey key will come | put down skgg.com | okskcg.gov | ok skcd dot com | Open escape.com | open spec .com | OPEN AS TO SEE IF I COME | Open its cage- eg.com | Open exisity cd.com. | gdot.com | Open XKCD.com | OK SPCT dot com | open the cd case or | Open skct.com | Open xttz.com. | open extacity dot com | ok eskg.com | Open this case of cd covers. | Ok it gets kingdom come | Open Excaziti.com | often he thinks about opening his mouth | Open. XKDC dot com | Open SKG.com | Open Sk cd dot com | open xkcd dot com | open xkcd.com | Open skcd.com. | open up kcg.com. | Open xfinity dot com | Okay skdg.com | open s k z d dot gov | go to skg.com | go to kgg.com | openthiskizziekid.com | OPEN S K C D DOT COM | Open expressCD.com | Okay let's get seated to cuddle | ok xxcity.com | Open sktz.com | Open Express CD dot com | Open xkcd.com | kasey k dot com | open x k c d dot com | ok skgt dot com | ok xtisida dotcom | open xkg.com | Can not understand at all | gct.com | Ok let's get Steve a car. | willpaythisgiginyourffingface.com | Open extensity dot com | ObeySAT.com | Open skgt.com | open xqcd.com | ok kis keect.com | open miss kckey.com | open s k c t dot com | oh press this keyboard come | open xgt.com | Open accessity.com. | Open extensity.com | Okay SKGT don't come. | okc.com | Open excasity.com | open this KCT dont gut you | Okay let's get to d.com | ?? | open skcd.com | open ecstasy dot com | Open xkcd.com. | open this KCT account now | Open extencity dot com. | Open XKCD.com. | open xkcd dot com | Open skeg.com | Okay xkcd.com | open shg.com | openxkcd.com | Open xkcd.com | open this casey key dot com | open extend cd then come on | Put down skgt.com | timepiece-ent.com | ok extracity.com | Open S K E G T dot com | open extacy.com | open skck.com

"call 15559876543" **baseline**

call one four five nine eight seven six five four three : 1 | call 145-987-6543 : 1 | Call 1559876543 : 6 | call one four five nine eight seven four five three : 1 | call 145 987 6543 : 1 | call one four five nine eight seven six five four three : 2 | Call 1459876453 : 1 | coal one four five nine eight seven six five four three : 1 | coal 145 987 6543 : 1 | call one four five ninety seven six five four three : 1 | Call 115 987 6543 : 1 | Call one four five nine eight seven six five four three : 1 | Coal 1559876543 : 1 | call 1559876543 : 5 | Coal 1459876543 : 1 | Coal one for five nine eight seven : 1 | Call 155 987 6543 : 1 | Call one four five nine eight seven six five four three : 1 | 41551976443 : 1 | Call 155- 987-6543 : 6 | call 1459874563 : 1 | call 1551876543 : 1 | Cole1459876543 : 1 | Call 1459876543 : 6 | CALL 145 987 6543 : 1 | cal 1559876543 : 1 | call 155 987 6543 : 3 | Call 1-4- 5 9-7 6-5-4-3 : 1 | call 15559876543 : 1 | call 155-987-6543 : 4 | Pole 1459876543 : 1 | Cole 155976543 : 1 | Paul - 1 555 987 6543 : 1 | Call 145-987-6543 : 6 | call 1459876543 : 10 | call 9876543 : 1 | Call 145 987 6543 : 1 | Coal 145 987 6543 : 1 | Call:155-987-6543 : 1 | call 145 9876543 : 1 | call 14596543 : 1 | call 1 4 5 9 8 7 6 5 4 3 : 1

"call 15559876543" **mangled**

41119876543 | who wants a five minute stab at 6493? | Call 1-4-5 97 6-5- 4-3 | show one plus five minus seven six five four three | Call 1559876543 | call155 9876543 | Fall 1459876543 | kroll one four five minus seven six five four three | fall 145986543 | 01459876543 | call one four five nine eight seven six five four three | call 8154876543 | call 15598765433 | call 155976543 | 4155976543 | 4 145 987 6543 | 145987643 | four one four five nine eight seven six five four three | Call one four five nine eight seven six five four three | Call 145-97-6543 | coal nine four five nine eight seven six five four three | fall 155976543 | Full one plus five ninety seven six five four three | call 1559876543 | 4145976543 | four one four five nine eight seven six five four three | Call 155 987 6543 | 41559976543 | call 15417996543 | Call one four five nine eight seven six five four three | VOICE: One plus five minus seven six five four three | Cole whats the 59876543 | Call 155-987-6543 | Pull 1459876543 | Four one four five nine eight seven six five four three | call 145976543 | Call 145-987-6543 | full 1459876543 | 4145697143 | Call one five five nine eight seven six five four three | Call: 155-987-6543 | Call 1459876543 | CALL 145 987 6543 | Five four three six seven | call 155 987 6543 | 41559876543 | 4-145-987-6543 | cole one plus five minus seven six five four three | Call- 155-987-6543 | call one four five one eight seven six five four three | Four One for five minus seven six five four three | four one five five nine eight seven six five four three | Call 1 4 5 9 8 7 6 5 4 3 | Call 155-987-6543 | Call 111 987 6543 | call 155657853 | Call 1-555-987-6543 | one four five nine seven six five four three | 5 minutes to heaven 6 5 4 3 | call: 1555976543 | 41459876543 | call 1459876543 | Call 145 987 6543 | 9156976543 | 414159876543 | 44159876543 | hold one four five nine eight seven six five four three | call 1 4 5 9 8 7 6 5 4 3

"send text to 1234567890 how are you doing" **baseline**

Sent text to 1234567890 how are you doing ? : 1 | Send text to 1234567890 How you doing?. : 1 | syntax 2 1234567890 how are you doing : 1 | send text to 1234567890 how you doing? : 1 | syntex 21234567890 how're you doing? : 1 | syntax is a thing to do : 1 | sintax2..12 3 4 5 6 7 8 9 0..how you doing? : 1 | send text to 1234567890. What are you doing? : 1 | Send text to 1234567890 How you doing? : 1 | Syntax 4567890 How you doing? : 1 | Syntax 2 1 2 3 4 5 6 7 8 9 0. How are you doing? : 1 | Send text to 1234567890. How are you doing? : 1 | Send text to 1 2 3 4 5 6 7 8 9 zero how you doing? : 1 | Show text to 1234567890 How are you doing? : 1 | Send text to 1234567890 How are you doing : 1 | Send text to 123-456-7890 How are you doing? : 1 | Syntax too. one two three four five six seven eight nine zero. How are you doing? : 1 | syntax one two three four five six seven eight nine zero. how you doing? : 1 | Send text to one two three four five six seven eight nine zero How you doing? : 1 | send text to 1234567890 23 : 1 | Syntax: 21234567890. How are you doing? : 1 | Send text to 1234567890: how are you doing? : 1 | sintex 2 1234567890 how you doing : 1 | syntextwo one two three four five six seven eight nine zero how are you doing? : 1 | send text to 123-456-990 how are you doing : 1 | syntax two one two three four five six seven eight nine zero how are you doing : 1 | Syntax. Two one two three four five six seven eight nine zero. What are

you doing? : 1 | sintax 1234567890 how are you doing : 1 | syntax 21234567890 how are you doing : 1 | Syntax Two: one two three four five six seven eight nine zero. How are you doing? : 1 | syntax two one two three four five six seven eight nine zero how are you doing : 1 | send text to 1-2-3-4-5-6-7-8-9-0 how are you doing : 1 | Send text to 1 2 3 4 5 6 7 9 0 how are you doing? : 1 | syntax 21234567890 how you doing? : 1 | syntax two 123 456 789 0 how you doing : 1 | send text to 1234567890 how you doing : 4 | Syntax 2 1 2 3 4 5 6 7 8 9 0 How are you doing? : 1 | Send a text to 123-456- 7890. How you doing? : 1 | Send text to 1234567890 how are you doing? : 1 | SYNTAX 21234567890 HOW YOU DOING? : 1 | Syntex 21234567890 how you doing? : 1 | Syntax 21234567890 how are you doing? : 2 | Syntax two one two three four five six seven nine zero how are you doing. : 1 | send a text to 1234567890 how you doing? : 1 | Send text to 1234567890. How are you doing? : 1 | sintex 21234567890 how are you doing? : 1 | Syntax two one two three four five six seven eight nine zero. How are you doing? : 1 | Send text to 123-456-7890 how're you doing : 1 | Send Text to 123-456-7890 How are you Doing? : 1 | Sent text to 1234567890. How are you doing? : 1 | Send text to 1234567890 how you doing : 2 | Syntax 21234567890. How are you doing? : 1 | syntax 243456790 How are you doing : 1 | syntax two one two three four five six seven eight nine zero how are you doing? : 1 | centex2123456890how are you doing : 1 | Send text to 1234567890 How are you doing? : 1 | syntax two one two three four five six seven eight nine zero how are you doing? : 1 | Send text to 1234567890: How you doing? : 1 | Send text to: 1234567890. How you doing? : 1 | Syntex two one two three four five six seven eight nine zero how are you doing? : 1 | Send text to 1234567890 why are you doing that? : 1 | send text to 1234567890 what are you doing. : 1 | sent text to 123456789 how are you doing : 1 | CENTEX 21234567890 HOW YOU DOING? : 1 | Syntex 721234567890 How are you doing? : 1 | Sending text to one two three four five six seven eight nine zero. How are you doing? : 1 | syntax 2 1 2 3 4 5 6 7 8 9 0 how you doing : 1 | Cintex 2.234567890 how are you doing? : 1

"send text to 1234567890 how are you doing" **mangled**

send pics to 1 2 3 4 5 6 7 8 9 0 | Syntax. 21234567890. Mining. | syntext 253450678 | Send a text to 1234567890. Thank you. | Syntex 23334567890 | send pegs to one two three four five six seven eight nine zero how you doing? | send text one two three four five six seven eight nine zero. | Ex 1 2 3 4 5 6 7 8 9 0 | Syntax 212-345-6990. Thank you. | send text to one two three four five six seven eight nine zero how are you doing | Syntax 2 1 2 3 4 5 6 7 8 9 0 what are you doing? | Syntax 213-345-6990 | cethrax 1234567890 try again | send text 2123567890 | Some text 2 23 4 5 6 9 9 0 by gate. | send text 25534567890 hurry up. | send fax to 1234567890 immediately | Syntax 2 3 4 5 6 7 8 9 0 | Send text to 20223799093 | Send text to 4532890 thank you. | send text 234567890 | Syntax two one two three four five six seven eight nine zero *unintelligible* | Sinfex 1234590 | syntax 2034567890 | Send text 21234567890 why again | send text to 123 456 7890 thank you | Send fax to 1234567890 while you do it. | Send text to 1234567890. How are you doing? | Send text to 123 456 7890. | syntax 2 1 2 3 4 5 6 7 8 9 0 rayboom | send text to 1234567890 what are you doing? | Sinfex 2345790303 | Send text to 1234567890 right away | Send text to- 123-456-7890 | Six zero one two three four five six seven eight nine zero call me. | Send text 12437890 | Syntax two one two three four five six seven eight nine zero. How are you doing? | syntax 2 1 2 3 4 5 6 7 8 9 0 ray com | Send text two winter three four five six seven eight nine zero How you doing? | Sent text to 1234567890 how are you doing ? | Syn-text two one two three four five six seven eight nine zero. Thank you. | send text to 12456790 right away | Send text to number 9093 | Send text to 1234567890 right away. | Send text to (123) 456-7890 wide gate | syntax 2.234567890 are you dead | sintex2 1 2 3 4 5 6 7 8 9 0 how you doing | Send Text to **030**** | syntex 21234567890 how're doing? | send text to 123456790 800 | Some texts too. One two three four five seven six nine zero. Waiting. | send text to one two three four five six seven eight nine zero how are you doing? | syntax 2.34567890 How are you doing | Send text to 1234567890 quietly | send text to 1234567890 are you doing it | Send Text to 1234567890. What're you doing too? | send text open to 345690 | syntax 1234567890 how are you doing | send text to 1 2 3 4 5 6 7 8 9 0 are you going to? | send text to 1234567890 Hi again. | two one two three five five nine zero | Syntax 21234567890 how are you doing? | Send text to 123-456-7890 | Send text 21234567890 | send text to 1234567890 what are you doing | Send text two or three four five six seven eight nine zero. How you doing? | syntax 2534567890 | syntax 2 3 4 5 6 7 8 9 0 | send text to 123 456 7890 right now | Send text. Twelve twenty three. Four five six seven eight nine zero. I adhere. | Syntax. Two. One two three four five six seven eight nine zero. Read Read. | Syntax. Twelve 1-2-3 4-5-6 7-8-9 0 hundred. | Saint Pex - 2123 456 789 0 How are you doing? | send text to 1234567890 thank you | send text to 1234567890 right now | 1234567890 | Send text to 123-456-7890 how are you doing? | Syntax 4567890 | Syntax two one two three four five six seven eight nine zero | cetax taking to 34843906 Regan | sentex2..12 3 4 5 6 7 8 9 0..how you doing | send text to 1234567890 goodbye | Syntax 2223456780 Viking | send text to 1234567890 what you are doing | 3x256789023 | syntax 234567890 | send text to 123 456 789 0 right away | syntax 2 1234567890 how are you doing | Some text two twenty-three four five six nine nine zero nine eight. | send text to 123-456-7890 right now | SEND TEXT 12123456990 THANK YOU | Send Fax through 1234567890 | Send text to 1234567890 right away. | Send text to 1234567890. What are you doing? | sent text to 1234567890 | sintax 1234567890 how are you doing | Syntax 456 | Send text to 1 2 3 4 5 6 7 8 9 0 right now. | syntax two one two three four five six seventy nine zero how are you doing? | sintex 213456790 thank you | Send text to 1234567890 | Send text to 123-456-7890. How are you doing? | send text to 1234567890 right away | send text 123456990 how are you doing | send fax to 12347890 | Send text to 123-456-7890. Later dude. | send text to 49 seconds | syntex 21234567980 123 | Send text to 1-2-3-4-5-6-7-8-9-0-1-0 | Send text to 1234567890 i'm waiting. | Send text to one two three four five six seven eight nine zero are you good? | Send text to 1234567890. What are you doing? | syntax to 1 2 3 4 5 6 7 8 9 0 alright | send text to 1234567890 play again | Send text to 1234567890 How are you doing | Send text to 1234567890 How you doing?. | syntax 2 2 3 for 5 seconds anti zero | Syntex1234567890mygame | Send text 2223456990 How you doing | Syntax. 1 2 3 4 5 6 7 8 9 0 200. | syntax 2 2 3 4 5 6 7 8 9 0 | send pics to 223 456 7890 | Send text to 123 456 990 what are you doing | syntax two one two three four five six seven eight nine zero | send text to 1234567890 hurry today | apex open to three for seven nine nine zero alright | send text to one two three four five six seven eight nine zero | Send fax to 1234567890 how you doing | Syntax one two three four five six seven eight nine zero | send text two one two three four five six seven eight nine zero right now | send text 1 2 3 4 5 6 7 8 9 10 | send text to 123456990 how are you doing | syntax 21234567890 how are you doing | send text to one to three four five six seven eight nine zero right away | Send text to 1234567890. How are you doing? | Send text to 12345678790. How you doing? | send text to 1234567890 how you doing | send text to 123456790 1800 | syntax21234567890 are you trained | Some text 1234567890 | Send text 244345699093 | Send fax to 123-456-7890 renegade. | Syntax 21234567890. How are you doing? | send text to 1234567890 | sleepex 21234567890 are you there | Send Texts to 1334569890 | Send text 2223 456 990 write it. | syntex 01234567890 lion king | syntax two one two three four five six seven eight nine zee | sintex one two three four five six seven eight nine zero | Sent text to 1234567890 Ry-game | Syntax 0123-456-7890 myerdon. | syntex 23334567890 | Syntax 2 one two three four five six seven eight nine zero. What are you doing? | Send text to one two three four five six seven eight nine zero How you doing? | send fax to 1234567890 how you doing