Whisper: A Unilateral Defense Against VoIP Traffic Re-Identification Attacks

Tavish Vaidya
Georgetown University

Tim Walsh
Georgetown University

Micah Sherr
Georgetown University

ABSTRACT

Encrypted voice-over-IP (VoIP) communication often uses variable bit rate (VBR) codecs to achieve good audio quality while minimizing bandwidth costs. Prior work has shown that encrypted VBR-based VoIP streams are vulnerable to *re-identification attacks* in which an attacker can infer attributes (e.g., the language being spoken, the identities of the speakers, and key phrases) about the underlying audio by analyzing the distribution of packet sizes. Existing defenses require the participation of both the sender and receiver to secure their VoIP communications.

This paper presents *Whisper*, the first unilateral defense against re-identification attacks on encrypted VoIP streams. Whisper works by modifying the audio signal before it is encoded by the VBR codec, adding inaudible audio that either falls outside the fixed range of human hearing or is within the human audible range but is nearly imperceptible due to its low amplitude. By carefully inserting such noise, Whisper modifies the audio stream's distribution of packet sizes, significantly decreasing the accuracy of re-identification attacks. Its use is imperceptible by the (human) receiver.

Whisper can be instrumented as an audio driver and requires no changes to existing (potentially closed-source) VoIP software. Since it is a unilateral defense, it can be applied at will by a user to enhance the privacy of its voice communications. We demonstrate that Whisper significantly reduces the accuracy of re-identification attacks and incurs only a small degradation in audio quality.

CCS CONCEPTS

• **Security and privacy → Network security**.

KEYWORDS

VoIP, Privacy

ACM Reference Format:
Tavish Vaidya, Tim Walsh, and Micah Sherr. 2019. Whisper: A Unilateral Defense Against VoIP Traffic Re-Identification Attacks. In *2019 Annual Computer Security Applications Conference (ACSAC '19), December 9–13, 2019, San Juan, PR, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3359789.3359807

ACSAC '19, December 9–13, 2019, San Juan, PR, USA

ACM ISBN 978-1-4503-7628-0/19/12…$15.00
https://doi.org/10.1145/3359789.3359807

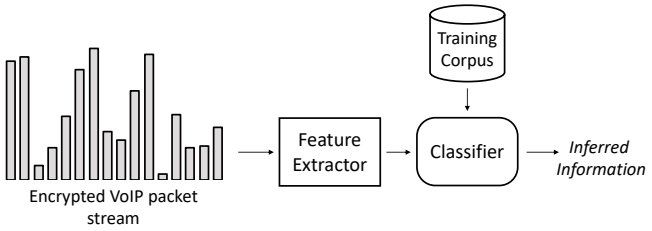

Figure 1: Overview of a traffic re-identification attack on an encrypted VoIP stream.

1 INTRODUCTION

Voice-over-IP (VoIP) systems encode voice for transmission over a network. The majority of popular VoIP systems use variable bitrate encoding (VBR) to achieve high quality audio while conserving bandwidth. The output data of VBR per unit time depends on the complexity of the input audio, resulting in audio frames (and ultimately, packets) of various sizes. To secure encoded voice data during transmission, VoIP systems often support end-to-end encryption, for example, via secure real-time transport protocol (SRTP).

Such protocols, however, while ensuring message confidentiality, still leak information about the underlying audio. Prior work has shown that significant information about the audio stream—including the identify of the speaker, the gender of the speaker, the spoken language, and even key phrases—can be inferred by analyzing the distribution of encrypted packet sizes [13, 14, 19–21]. Such re-identification attacks are possible because the size of the encrypted packets depend on the type of audio being encoded by the VBR codec; less complex audio (e.g., silence) requires fewer bits to encode.

Typically, *re-identification attacks* use machine learning techniques to infer information about the encoded audio from the encrypted VoIP stream. Figure 1 shows an overview of the attack's workflow. An adversary intercepts the encrypted VoIP stream and extracts features from the distribution of encrypted packet sizes. Using a labeled training corpus, the adversary applies machine learning techniques to build a classifier (again, using features based on the distribution of encrypted packet sizes), and applies the classifier to extract information about the underlying audio. Modern encrypted VoIP systems are surprisingly vulnerable to such attacks; for example, Wright et al. [21] showed that the spoken language can be inferred with 87% accuracy when presented as a binary classification problem and 66% accuracy using a 21-way (i.e., 21-language) classifier.

A straightforward and effective defense against re-identification attacks is to abandon VBR in favor of constant bitrate encoding (CBR). CBR offers complete protection against re-identification attacks since it eliminates information leakage due to packet size.

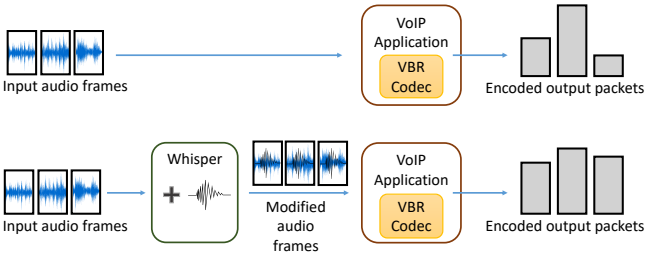

Figure 2: *(Top)* The VBR codec of a VoIP application encodes input frames to generate encoded packets that are vulnerable to traffic re-identification attacks. *(Bottom)* Whisper adds extra inaudible audio to the input audio frames before they are encoded by the VBR codec, altering the packet sizes of encoded packets and thwarting traffic re-identification attacks.

However, for the same targeted bitrate, VBR offers far better audio than CBR. Reliance on CBR incurs such a bandwidth overhead that we know of no encrypted VoIP system that has opted to use it.

There are also existing defenses that attempt to disrupt re-identification attacks while still permitting the use of VBR codecs [14, 22]. These function by modifying the size of packets generated by a VBR codec, thus hiding the underlying packet size distribution of the encoded audio. However, existing defenses either require participation of both the sender and receiver [14, 22] or require white-box access to the VBR codec [22].

In this paper, we propose Whisper, a *unilateral* defense against re-identification attacks. Whisper leverages the limits of the human audible range to alter the size of packets generated by a VBR codec in a manner that (i) obfuscates the true packet size distribution and (ii) is (ideally) imperceptible to the receiver. It allows a privacy conscious sender to secure his side of the communication without any support from the receiver.

Whisper alters the size of output packets generated by a VBR codec by overlaying *tuning audio* on the actual audio before encoding occurs. The addition of tuning audio changes the characteristics of the original audio signal to be encoded, without affecting the contents of the original audio as perceived by the human listener on the receiver side.

At first blush, it may seem that Whisper is inherently incompatible with modern audio codecs, since codecs often use band filters to remove audio outside of the human audible range. However, in practice, codecs typically err on the side of preserving audio quality and are inexact in their filtering. This leads to segments of the spectrum that are both not-filtered and either inaudible or unplayable due to the limits of commodity speakers.

Figure 2 shows an overview of Whisper. Whisper overlays tuning audio to the input audio frame before it is encoded by the VBR codec in the VoIP application. It is thus agnostic to the particular VoIP application, which we assume is unaware of the Whisper protections and merely receives audio data from a Whisper-enabled audio/microphone driver. In summary, Whisper's unilateral protections and ability to be used with any closed-source (i.e., black-box) VoIP software enable the practical protection of communication using deployed VoIP systems.

We evaluate Whisper using large voice corpora and the popular Opus VBR codec [2]. We show that Whisper significantly reduces the accuracy of re-identification attacks. For example, Whisper decreases the adversary's accuracy to correctly identify the speaker of an encoded VoIP conversation from 97.22% (without Whisper) to 31.13% (with Whisper). Whisper incurs limited bandwidth overhead and has no significant impact on the quality of actual audio.

2 RELATED WORK

VoIP re-identification attacks are instances of traffic fingerprinting, the latter of which has been richly explored (see, for example, early work by Hintz [12] and Crotti et al. [7]). Traffic fingerprinting attempts to infer characteristics about communication by examining its network attributes (e.g., the timing, sizes, and inter-arrival times of packets; and their distributions) rather than by analyzing the communication's contents. There is an active arms race between *website* traffic fingerprinting techniques and defenses [6, 18], which is especially relevant to anonymity networks such as Tor [8].

VoIP re-identification attacks apply similar fingerprinting techniques to identify attributes of the underlying call audio and/or the participants of the communication. Prior studies have found the distribution of (encrypted) packet sizes to be sufficient to infer with high accuracy the language being spoken [21], the gender and identity of the speaker [14], and even key phrases [19, 20].

Wright et al. [22] first proposed a defense against statistical traffic analysis of VoIP streams by morphing one class of traffic to look like another class. Their proposed defense alters the packet sizes of the source traffic such that the statistical distribution of its encoded packet sizes closely matches that of the target traffic. With only black-box knowledge of the codec, their defense increases the packet sizes by padding the encoded output of a VBR codec. With white-box access to the codec, Wright et al. [22] rely on the selection of the bit rate within the codec to increase or decrease the size of an encoded packet. To find a distribution closest to that of the target traffic, Wright et al. use comparison functions such as the χ^2 statistic and convex optimization to minimize the overhead due to the padding of packets.

Rather than morphing the source distribution to a particular target distribution, Moore et al. [14] calculate a new, synthetic, "superdistribution" to which all source traffic distributions can be morphed. To calculate the superdistribution, their defense considers the distributions of all potential source traffic and determines the least bandwidth-intensive distribution that can be used to map all of the source traffic. Once the superdistribution has been determined, the output of the VBR codec is padded to map it to the size described by the superdistribution. Because the padding is itself encrypted end-to-end, an attacker cannot easily infer the original, unpadded distribution of packet sizes. Whisper's approach to determine how much noise/padding to add to the baseline traffic borrows from Moore et al.'s algorithm.

Limitations of existing defenses. A straightforward defense to prevent the leakage of information due to traffic analysis of packet sizes is to use constant bitrate encoding (CBR). However, to achieve the same audio quality as VBR, CBR incurs significant bandwidth overheads. This makes CBR unsuitable for networks with limited bandwidth such as cellular networks.

The major limitation of existing traffic morphing defenses for VoIP streams [14, 22] is that they require both communicating parties to support and participate in the defenses. Defenses proposed by both Wright et al. [22] and Moore et al. [14] add padding to alter the size of the encrypted packets on the sender side, requiring the receiver to strip the extra padding. In cases where the receiver does not support the removal of the extra padding, the sender can only communicate over the vulnerable VBR channel. This essentially prohibits a privacy conscious participant from communicating with another party who does not support these defenses.

In contrast, our proposed Whisper defense is unilateral and does not require the participation of the receiver. Currently, to our knowledge, no deployed VoIP system supports a unilateral defense that can prevent traffic analysis of encrypted VoIP streams while supporting VBR encoding. Our techniques can be implemented as a virtual device driver, and are therefore compatible with existing closed-source VoIP software (e.g., Skype).

Additionally, the approach taken by Wright et al. [22] of changing the codec's bitrate to manipulate packet sizes requires white-box access to the VBR codec. In contrast, Whisper takes a black-box approach and can work with applications that do not allow access to the codec or its settings.

3 USER AND ATTACK MODELS

We assume two parties communicating via a VoIP application that uses VBR encoding. The VoIP application provides end-to-end encryption; that is, it encrypts all traffic between the communicating parties to prevent eavesdropping, but does not make any effort to hide the size of the encrypted packets. Furthermore, we assume that the communicating parties use a closed-source VoIP application such as Skype and are unable to modify the codec parameters. This assumption enforces the constraint that the defense should work with popular VoIP clients without requiring any modifications to them.

As with previous work [14], Whisper assumes that the VBR codec used by the VoIP application is publicly known. Whisper requires some per-codec tuning, which necessitates having black box access to the codec. This is a realistic assumption since popular VoIP clients use standardized codecs whose implementations are publicly available. For example, Skype uses the Silk codec [17] while WhatsApp is known to use the Opus codec [2, 9, 16], both of which have publicly available implementations. We do not require that the codec is itself open source; rather, we require only that an implementation is available for tuning our defense.

Since VoIP is typically a bidirectional channel, it should be emphasized that Whisper protects only the communication that is generated by the party applying Whisper. We do not consider correlation attacks in which the unprotected direction is used to infer information about the channel being protected by Whisper; this is likely feasible for inferring language (since typically both communicants use the same language), but may be difficult for re-identification attacks that attempt to perform speaker identification or identify key phrases. Of course, Whisper can be used by both parties to provide bidirectional protections.

Our attacker model follows existing work [14, 22] with respect to the adversary's capabilities and access to training data for performing traffic analysis. We consider a passive adversary that intercepts all encrypted VoIP traffic between the communicating parties. The adversary does not have access to the underlying plaintext audio. However, it can inspect the traffic and learn other characteristics, including the size and timing of packets.

The adversary's goal is to use the distribution of packet sizes obtained from the encrypted packet stream to discern information about the underlying audio. In this paper, we focus on the case in which the adversary attempts to learn the identity of the speaker, given a closed-world setting in which the set of candidate speakers is known apriori. We emphasize that the closed-world setting is a conservative model (for the defense). That is, a defense that successfully thwarts accurate re-identification in the (worse-case) closed-world setting is also effective in the open-world setting in which all speakers (or languages, genders, phrases, etc.) must be considered.

We chose to consider speaker identification—as opposed to re-identifying gender or language—as speaker identification has been previously shown to be highly accurate [14] and arguably more interesting to potential eavesdroppers than gender or language identification. (Presumably, learning the identity of the speaker also provides hints at gender and language.)

To conduct its attack, the adversary has access to a training corpus of unencrypted audio samples, including samples from all potential speakers in our closed-world setting. The adversary also has complete knowledge of the Whisper algorithm and its parameters, excluding the private random bits generated by the sender.

As shown in Figure 1, the adversary uses the training corpus to build a machine learning classifier to learn information about the encoded audio from the encrypted packet size distribution. All known re-identification attacks on encrypted VoIP streams [13, 14, 19–21] consider the frequency of n-grams over the size of packets as features to the machine learning classifier. We use a similar approach to show the vulnerability of the Opus codec [2] to re-identification attacks and to evaluate the effectiveness of Whisper in mitigating such attacks.

4 METHODOLOGY

Re-identification attacks on encrypted VoIP streams leverage the packet size distribution of the encrypted VoIP packets to perform traffic analysis. Whisper defeats such attacks by changing the size of the encrypted packets generated by the VoIP application before they are sent over the network. The updated sizes of these encrypted packets should be such that their packet size distribution decreases the information leaked by the encrypted VoIP stream and reduces the ability of the adversary to perform accurate traffic analysis.

Moore et al. [14] propose padding packets to achieve a particular distribution—the superdistribution—to which all classes of traffic (e.g., different speakers, genders, phrases, etc.) can be mapped. Conceptually, morphing all underlying (and revealing) distributions to the superdistribution hinders re-identification attacks since it removes the adversary's ability to discover distinguishing features within the packet size distribution.

Whisper borrows the superdistribution concept from Moore et al. [14], but uses inaudible audio to enable unilateral protections. In

what follows, we provide a brief overview of the superdistribution generation (§4.1) and mapping techniques (§4.2), and then describe how Whisper uses inaudible noise to morph traffic to the superdistribution (§4.3).

4.1 Creating the Superdistribution

Moore et al. [14] construct a superdistribution using an audio corpus, which we will refer to as the *training corpus*. (They conservatively assume that the adversary also has access to this corpus.) Without loss of generality, we will describe both the defense of Moore et al. and our Whisper system in terms of defending against re-identification attacks that aim to identify a speaker from a closed set of potential speakers. Our defenses are equally applicable to other re-identification tasks.

We assume a VBR codec that produces a sequence (vector) of L *audio frames* $\vec{a} = \langle a_1, \cdots, a_L \rangle$, where each audio frame encodes a fixed-length time period of the audio (usually 20 ms) and L is a function of the length of the source audio. That is, $\vec{a}$ is the encoding of the input audio sample produced by the VBR encoder. We consider the set of possible packet sizes over $\vec{a}$ to be the *codec alphabet* (Σ_{in}) of that codec. We note that Σ_{in} is finite, and treat it as an ordered set $\Sigma_{\text{in}} = \{\Sigma_1, \Sigma_2, \cdots, \Sigma_{|\Sigma_{\text{in}}|}\}$ where $\Sigma_i < \Sigma_j$ when $i < j$, for all $i, j \in [1, |\Sigma_{\text{in}}|], i \neq j$.

The superdistribution generation algorithm considers the distribution of all the speakers in the training corpus and calculates the least bandwidth-intensive distribution that can be used as the target distribution. To preserve audio quality, we are limited to additive modifications only: we can pad any audio sample $a_q \in \vec{a}$ of size Σ_i to any size larger than Σ_i, but cannot *decrease* the size of a_q without significantly degrading audio quality. While the defense of Moore et al. does not require that the set of padded packet sizes (Σ_{out}) equal that of the codec (i.e., Σ_{in}), Whisper necessitates that $\Sigma_{\text{out}} = \Sigma_{\text{in}}$ since the receiver should be agnostic (and potentially unaware) of the defense's use. For clarity, in what follows, we assume that $\Sigma_{\text{out}} = \Sigma_{\text{in}}$ and use Σ as shorthand.

We directly apply the superdistribution generation algorithm of Moore et al. [14, see Algorithm 1]. Briefly, the superdistribution generation algorithm considers the packet size distributions for each speaker in the training corpus, and then calculates the least bandwidth-intensive distribution to which the packet size distribution of all the speakers in the training corpus can be morphed. For the ascending list $L_z = \langle l_{1_z}, \cdots, l_{k_z} \rangle$ of k different possible lengths of output packet sizes for a packet stream z, the superdistribution algorithm calculates a target distribution L_t such that for all $1 \leq i \leq k$, $\sum(l_{i_t} + \cdots + l_{k_t}) = \max_z(\sum(l_{i_z} + \cdots + l_{k_z}))$ over all packet streams z in the training corpus.

We assume an adversary will consider not just the relative frequency of packet sizes, but also the relative frequency of n-grams of packet sizes. (This is the predominant approach used by prior work on re-identification attacks [14, 19–21].) More precisely, the adversary uses overlapping sequences of length n over the output packet sizes in Σ as features for the machine learning classifier. Like the approach of Moore et al. [14], Whisper's superdistribution generation algorithm computes a separate superdistribution for each unique sequence of $n-1$ packet lengths. Thus, there will be $|\Sigma|^{n-1}$ superdistributions. During the mapping step (see §4.2), the superdistribution matching the last $n-1$ packet length sequence is used to determine the target packet size of the next packet to be encoded.

4.2 Mapping to the Superdistribution

Whisper uses the superdistribution to determine the *desired* size of the next outgoing packet from the VoIP application. That is, given an input audio frame a_q of size Σ_i and the history of previously transmitted audio frames (including their added noise), Whisper determines the desired augmented packet size Σ'_i (where $\Sigma'_i \geq \Sigma_i$) that will cause the distribution of packet sizes to appear closest to that of the superdistribution. (As discussed above, Σ'_i cannot be less than Σ_i without incurring a significant loss of audio quality.)

We make a slight modification to the mapping algorithm proposed by Moore et al. [14] to include additional parameters to allow a trade-off between security and bandwidth overhead. (We use the term *packets* to be consistent with the terminology of Moore et al. [14]. Moore et al.'s defense added padding to the encoded packets generated by the VoIP application. In contrast, Whisper modifies audio frames before they are encoded by the VoIP application.)

Algorithm 1 describes how Whisper calculates the target packet size from an input stream such that the distribution of target packet sizes closely resembles the superdistribution. The mapping algorithm works for any level of n-grams.

In lines 5-8, we pad the initial $n-1$ packets (audio frames) to the maximum packet size for bootstrapping. Next, for each input audio frame, line 11 computes the cost of choosing each possible packet size based on the current distribution of the last $n-1$ output packet sizes and the target distribution. The cost for each potential packet size represents the distance between the target and current distribution, if that particular packet size was chosen. Line 13 modifies the cost of choosing a packet size for the next packet based on a *pktSizeWeights* weighting parameter for each target packet size. The weighted costs allow for favoring smaller packet sizes while penalizing larger packet sizes. This allows us to trade off between performance and security. Based on weighted cost, line 15 assigns the probability of selection to each packet size. The non-negative *strictness* parameter determines how strictly the target distribution adheres to the superdistribution. A smaller value means stricter adherence compared to a larger value. The strictness parameter allows the mapping algorithm to boost the selection probability of smaller packet sizes to reduce the bandwidth overhead by trading-off security. Line 17 returns either the next output packet size based on the computed probabilities or the size of the maximum packet if there are no non-zero probabilities. Lines 19-22 update the current distribution counts and the last n packets. Line 23 then modifies the input audio frame (by overlaying tuning audio) before passing it to the VoIP application such that the size of the encoded output packet generated by the VoIP application matches the desired packet size chosen in line 17.

4.3 Whisper

Whisper's overarching goal is to decrease the accuracy of re-identification attacks by modifying the size of encoded packets generated by the VBR codec of a VoIP application. The modification of audio frames to produce packet sizes that are reflective of the superdistribution minimizes information leakage and reduces the accuracy of traffic re-identification attacks.

Algorithm 1 Mapping an input distribution to output distribution determined by the superdistribution.

```
1:  procedure MorphStream(inputStream, targetStream, numPktSizes, NgramSize, pktSizeWeights, strictness)
2:      currentDistCounts ← Empty array of size numPktSizes^NgramSize
3:      lastNPkts ← Empty queue of packet sizes
4:      maxSizePkt ← Size of largest packet in inputStream
5:      for x in range(0, NgramSize) do
6:          currentPkt ← inputStream.dequeue()
7:          doWhisper(currentPkt, maxSizePkt)
8:          lastNPkts.enqueue(currentPkt.size())
9:      while currentPkt ← inputStream.dequeue() do
10:         ▷ Cost is the distance between the target & current distribution for all packet sizes, if that size was chosen.
11:         sizeCosts ← computeSizeCosts(currentPkt.size(), targetStream, maxSizePkt, currentDistCounts, lastNPkts)
12:         ▷ Weighted costs allow for favoring smaller packet size/penalize larger size.
13:         weightedSizeCosts ← computeWeightedSizeCosts(sizeCosts, pktSizeWeights)
14:         ▷ Based on weighted cost, decide probability of selection for each packet size.
15:         pktSizeProbabilities ← computeProbabilities(weightedSizeCosts, maxSizePkt, strictness)
16:         ▷ Choose the output packet size using weighted selection probabilities.
17:         chosenPktSize ← choosePktSize(pktSizeProbabilities, maxSizePkt)
18:         ▷ Update current distribution.
19:         currentDistCounts[lastNPkts][chosenPktSize] + +
20:         currentDistCounts[lastNPkts][totalPkts] + +
21:         lastNPkts.enqueue(chosenPktSize)
22:         lastNPkts.dequeue()
23:         doWhisper(currentPkt, chosenPktSize)
```

As shown in Figure 2, Whisper mitigates re-identification attacks by overlaying extra audio, called *tuning audio*, to the audio frames generated by the sender *before* they are passed to the VoIP application.

In our preliminary investigation, we observed that the addition of tuning audio to the original audio can alter the size of the encoded output generated by the VBR codec. VBR codecs are sensitive to the complexity of the audio being encoded; the output data of VBR per unit time varies with the audio complexity. Encoding an audio frame containing a high frequency (ultrasonic) signal will therefore result in a larger encoded packet size as compared to an audio frame with silence. We leverage this behavior of VBR codecs to overlay tuning audio to alter input audio frames in order to achieve the desired size of the encoded output, as determined using the superdistribution (see §4.1 and §4.2). As we discuss in the remainder of this section, we consider various forms of tuning audio.

Characteristics of tuning audio. Whisper affects packet sizes by adding tuning audio to the sender's audio messages before they are encoded by the VoIP application. On the receiver side, the VoIP application decodes the encoded audio, which includes the original audio frames intermixed with the tuning audio. Whisper is a unilateral defense and does not require any support on the receiver; put equivalently, the receiver does not attempt to actively remove the tuning audio. This restricts the types of tuning audio that can be used, since audible tuning audio could significantly degrade audio quality. In contrast, tuning audio should not introduce extraneous noise and have minimal impact on the receiver's perceived audio quality.

For example, even if using white noise as tuning audio results in the desired output packet size for a given audio frame, the white noise will be audible in the decoded audio on the receiver side and will too substantially degrade the quality of the communication.

To satisfy these requirements, we consider tuning audio that lies on and beyond the boundary of the human auditory range (20 Hz to 20 kHz [10]). Even though frequencies outside this range are imperceptible to human listeners, we found that they are not discarded by popular VBR encoders. Moreover, their inclusion as tuning audio influences the size of the encoded output, without introducing any perceptible noise in the decoded output on the receiver side.

In addition to inaudible frequencies, we also consider extremely low amplitude tuning audio signals in the audible frequency range. This was necessitated by the observed relationship between the range of input frequencies in the input audio to be encoded and the corresponding encoded packet size. We observed that when using the Opus codec [2], for instance, there were some transitions from one packet size to another, as required by the superdistribution, that we could not achieve by injecting inaudible tuning audio. These required transitions from input packet sizes to target packet sizes were such that the use of tuning audio below 20 Hz resulted in encoded packet sizes less than the desired packet size, whereas the use of tuning audio above 20 kHz resulted in encoded packet sizes greater that the desired encoded packet size. Thus, to achieve these target packet sizes, we found it necessary to inject low amplitude (volume) tuning audio. We further discuss the use of various types of tuning audio and their implications to security, audio quality, and bandwidth overhead in §5.

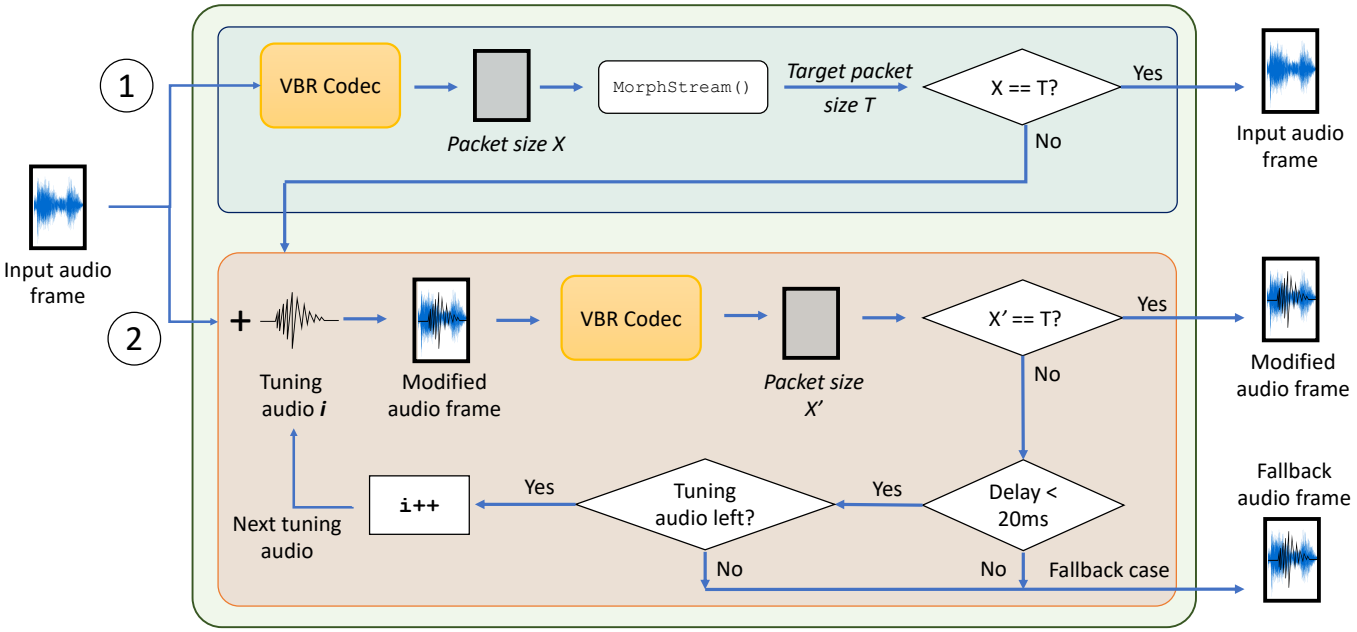

Figure 3: Whisper's workflow of modifying an input audio frame using tuning audio.

Whisper workflow. Figure 3 shows Whisper's high level workflow. To protect a speaker's VoIP communication from traffic re-identification attacks, the overlaying of tuning audio onto the outgoing audio frame should happen before it is encoded by the codec. Our user model assumes that Whisper has access to an implementation of the codec used by the VoIP application. Using the standalone codec implementation, Whisper first encodes the input audio frame a_q generated by the sender to determine the encoded packet size $X \in \Sigma$. It then uses the MorphStream procedure (Algorithm 1) to determine the target packet size $T \in \Sigma$ (where $T \geq X$) for the audio frame. If the encoded packet size X matches the target packet size T required by MorphStream (i.e., $X = T$), then no change is required to the size of the encoded packet and there is no need for any tuning audio overlay. In this case, Whisper trivially outputs the unmodified input audio frame a_q as the output.

In the case in which the encoded packet size X of the input audio frame a_q does not match the desired packet size T, Whisper overlays a single tuning audio from a predetermined candidate set (explained below) onto a_q, encodes the modified frame a'_q using the standalone codec implementation and determines the encoded packet size X' of a'_q. If the encoded packet size X' equals the desired packet size T, Whisper outputs the modified audio. Otherwise (i.e., $X' \neq T$), Whisper tries the next tuning audio.

We restrict the time Whisper can take to try different tuning audio from the set of candidates to under 20 ms to prevent gaps in audio on the receiver side, since packets usually convey 20 ms of audio. This restricts the number of tuning audio candidates that can be tried as overlays to achieve the desired packet size T.

If the encoded target packet size T is not achieved within 20 ms, Whisper outputs an audio frame according to a fallback strategy: in the *default* strategy, Whisper outputs the unmodified audio frame a_q; the *random* strategy overlays the input audio frame a_q with tuning audio selected uniformly at random from the candidate set (see below); finally, the *max* strategy outputs the input audio frame overlaid with a high frequency tuning audio such that the resulting encoded packet is maximally sized (i.e., $\Sigma_{|\Sigma|}$). We analyze the impact of the various fallback strategies in §6.1.

Generating tuning audio candidate set. To build the pool of tuning audio candidates, we use the Sox utility [4] to produce audio tones that are 20 ms in duration and are composed of one or more sine wave signals at different frequencies and amplitudes. We first consider candidates that lie outside or at the boundary of the human auditory range. In particular, we consider the infrasonic integer valued frequencies between 1 and 18 Hz, and the four ultrasonic frequencies between 20-23 kHz, at increments of 1 kHz. For each frequency, we generate multiple tuning audio candidates with different peak amplitudes, spaced uniformly, with a maximum peak amplitude factor of 0.5 (meaning, one-half the original amplitude of the sine wave).

As discussed above, the use of tuning audio in the inaudible range fails to achieve certain transitions between source and target packet sizes. Thus, we also include candidates with frequencies within the human audible range, but with peak amplitudes factors not exceeding 0.001. This ensures that the tuning audio that lie within the human audible range remain faint in comparison to the actual audio produced by the human speaker. Within the audible range, we consider 40 equally spaced frequencies between 100 Hz and 20 kHz as candidate tuning audio.

Finally, we also consider tuning audio candidates that are composed of sine waves at three to five randomly chosen frequencies.

This results in a (rather large) set of tuning audio candidates. This is undesirable since Whisper needs to identify the correct

tuning audio to overlay to achieve the desired packet size (via trial-and-error) *within 20 ms*. To prune the set of tuning audio candidates, we select a random subset of the training corpus and construct a superdistribution over all audio samples in this subset. We then morph this subset using all the tuning audio candidates in the pool of candidates. For each audio frame to be encoded, we consider every candidate tuning audio in the existing pool to encode each frame until we hit the desired packet size for that frame or run out of tuning audio to try. The candidates are tried in the order of ultrasonics, infrasonics and then those within the human audible range. For each of these categories, we try tuning audio in the increasing order of peak amplitude to prioritize *quieter* candidates. Starting with a large set of tuning audio candidates and encoding the subset of training data, we note the number of successful transitions achieved by the current pool. We also note the number of successful transitions achieved by each candidate tuning audio. To shrink this pool of tuning audio such that all candidates in the pool can be overlaid and encoded within 20 ms, we repeatedly eliminate the tuning audio with the minimum number of successful transitions each from the ultrasonic, infrasonic and audible range candidates. During our shrinking process, we found that the candidates with the same frequency (outside of the human audible range) but with peak amplitude difference of less than 0.3 resulted in the same encoded packet size for a given input packet. This allowed us to further prune the pool by eliminating tuning audio with nearby peak amplitudes for a given frequency without affecting the total number of successful transitions.

The above procedure produces a final set of 64 *inaudible* candidates that lie outside or at the boundary of human and 151 *audible* candidates (which include the 64 candidates from the *inaudible* set). All of the tuning audio are faded-in and faded-out to prevent the appearance of "clicking" noise across frame boundaries in the decoded output. This smoothing is necessary at frame boundaries to compensate for physical limitations in commodity speakers: speakers feature diaphragms with specific frequency response ranges that cause artifacts (clicks) when inter-frame transitions are insufficiently smooth.

5 EVALUATION

We next evaluate the efficacy of Whisper to defeat re-identification attacks and examine the defense's communication overheads and effects on audio quality.

Experimental setup. We use a subset of the Voxforge speech corpus [5] for evaluating Whisper. Our dataset is comprised of 21 speakers (14 male and 7 female) reading English literature recorded under different settings and with various background noises. The heterogeneity in recording environments influences the VBR codec's encoding behavior, making this a conservative (difficult) case for traffic morphing defenses such as Whisper. For each speaker, we consider 240 audio samples.

We use the Opus codec [2] to evaluate our proposed defense. The Opus codec is standardized by the IETF and is the successor of the Silk codec considered in prior work [14]. We encode our training corpus with Opus in VBR mode with its default parameters to generate encoded packets of various sizes. We note that the number of distinct packet sizes ($|\Sigma|$) generated by the Opus codec is far more than the Speex and Silk codecs considered in previous research [14, 22]. Speex and Silk produced only nine and eight distinct packet sizes respectively, whereas the Opus codec outputs a much larger range of packet sizes which is dependent on the sampling rate of the input audio. All audio samples in our dataset were sampled at 16 kHz, resulting in encoded packets with a contiguous packet size distribution between 62 to 327 bytes.

5.1 Evaluation Strategy

We evaluate the effectiveness of our proposed defense by comparing the attacker's ability to successfully perform a traffic re-identification attack on Opus-encoded VoIP streams when (i) no defense is applied and (ii) Whisper is enabled. The attacker's goal is to successfully identify the speaker (out of the 21 speakers in our dataset) from the intercepted packet stream. Figure 1 shows the high level overview of traffic re-identification attacks on VoIP streams.

Since the large number of distinct packet sizes generated by Opus makes traffic analysis difficult, we adopt a binning strategy to reduce the number of distinct packet sizes, mapping the various packet sizes into eight bins prior to performing traffic analysis. (That is, we force $|\Sigma| = 8$.) We consider the relative frequency of n-grams as features for the machine learning classifier. We considered various supervised machine learning classifiers and n-gram features during our investigation and found trigram features with an SVM classifier to provide the best accuracy. We, therefore, report results for 10-fold cross validation with an SVM classifier that uses the relative frequency of various trigrams as the feature vector.

We provide the attacker with access to the same training corpus used by Whisper to generate the superdistribution. This conservative assumption only provides more power to the attacker for improving its classifier. The attacker is also allowed to train or update its existing classifier with packet streams generated by Whisper. That is, the attacker is Whisper-aware and can apply Whisper as a preprocessor over the training corpus, allowing it to train on (labeled) Whisper-processed traffic streams. As discussed in §4.3, we make use of the *inaudible* and the *audible* sets of tuning audio in our evaluation.

5.2 Attack Accuracy

We define the *attack accuracy* to be the average accuracy across the ten folds of the cross validation. The *best case attack accuracy* (from the attacker's perspective) corresponds to the maximum accuracy achieved by the attacker using its SVM classifier, across all tested configurations (e.g., superdistribution parameters and fallback schemes).

Baseline accuracy. When no defense is applied, the attacker can perform traffic analysis of Opus-encoded VoIP packet streams and identify the speaker from the dataset with a best case attack accuracy of 97.22%, using trigrams and an SVM classifier. This shows that the Opus codec in VBR mode is vulnerable to traffic re-identification attacks.

Whisper accuracy. Whisper is able to significantly reduce the attacker's accuracy of traffic re-identification. Using candidates from the inaudible set of tuning audio as overlays, the attacker's

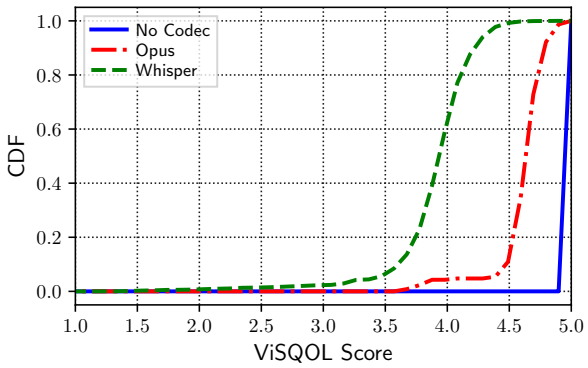

Figure 4: CDF of ViSQOL quality scores for baseline audio with no encoding ("No Codec") and audio encoded with the Opus codec without ("Opus") and with ("Whisper") the Whisper defense.

best case attack accuracy is reduced to 62.24% (compared to the baseline case of 97.22%). With the use of tuning audio from the larger audible set, the best case attack accuracy is further reduced to just 31.13%. The audible set outperforms the smaller inaudible set because the tuning audio in the inaudible set are only able to morph packets to certain packet sizes, whereas the audible tuning audio are able to cover the entire range of target packet sizes.

We also compare Whisper's effectiveness to a hypothetical technique that is able to perfectly morph the distribution of packet sizes in the input audio stream to that of the superdistribution. (The approach by Moore et al. [14] is always successful at morphing to the superdistribution, but does so at the expense of requiring bilateral cooperation between the two communicating parties.) Whisper fails to achieve ideal morphing when it cannot find a tuning audio from the candidate set of tuning audio that results in the target packet size within 20ms; in such cases, it uses one of the fallback schemes to modify the audio frame (see §4.3). Notably, however, such failures are rare and have only a modest effect on the defense's effectiveness: the hypothetical perfect morpher achieves the best case attack accuracy of 26.3%, compared to 31.13% when Whisper is used. We discuss the effects of various fallback strategies in more detail in §6.1.

5.3 Bandwidth Overhead

The bandwidth overhead incurred by Whisper stems from the increase in packet sizes necessary for hiding the underlying packet size distribution. Whisper incurs modest overheads of 34.01% and 38.43% (relative to unprotected audio) with inaudible and audible sets of tuning audio, respectively. As a point of comparison, switching to constant bitrate encoding imposes nearly a 90% overhead. Whisper allows for tunable security and performance, with one coming at the cost of decreasing the other. For example, the minimum bandwidth overhead using the audible tuning audio candidate set can be reduced from 38.43% to 18.6%, at the cost of increasing the accuracy of re-identification attacks from 31.13% to 52.94%. We discuss these tradeoffs in more detail in §6.2.

5.4 Impact on Audio Quality

We evaluate the impact of adding the tuning audio on audio quality, as measured on the receiver side. We use the following two methods to quantify VoIP quality:

5.4.1 Virtual Speech Quality Objective Listener (ViSQOL). ViSQOL [11] is a model of human sensitivity to degradations in speech quality. It uses a spectro-temporal measure of similarity between a reference and a test signal to determine the quality of speech in an audio sample and provides a mapping from an internal metric to a Mean Opinion Score (MOS) estimate. The MOS metric [3] has been commonly used to measure the quality of audio, including VoIP conversations. The metric ranges from a quality score of 1.0 to 5.0, with 1.0 being the worst. Actual VoIP calls usually lie in the range of 3.5 to 4.2 [1]. To determine the impact of tuning audio on audio quality, we use the reference implementation made publicly available by Hines et al. [11]. We refer to the MOS estimate generated by this implementation as the ViSQOL score.

We consider each audio sample from our training corpus. As a baseline, for each audio sample, we compare the raw audio without any VBR encoding to itself. Unsurprisingly, this yields an average quality score of 5.0 across the entire dataset, as the reference and test audio samples are identical.

We next assess the quality achieved after encoding with the Opus VBR codec. Equivalently, this is the audio quality that results when the Whisper defense is not used. Here, we compare each raw audio sample to the sample produced after encoding with Opus. This yields an average ViSQOL audio quality score of 4.6. We consider this a reasonable "upper-bound" for defenses against re-identification attacks.

Figure 4 shows the cumulative distribution of ViSQOL scores across all audio samples in the corpus for no encoding ("No Codec"), Opus without any Whisper protections ("Opus"), and Whisper. For the Whisper configuration, we use the audible tuning audio setting, which offers the best security (corresponding to a best case attack accuracy of 31.13%) but also intuitively should impose the greatest degradation in audio quality (since it inserts audible noise).

When Whisper uses audible tuning audio, we observe an average ViSQOL score of 3.9; the average increases slightly to 4.0 when Whisper uses inaudible tuning audio. In summary, Whisper imposes a modest degradation in audio quality, and the difference between using audible and inaudible tuning audio is minimal.

The minor difference in ViSQOL scores between the audible and inaudible tuning audio settings indicates a potential downside in using automated models to measure audio quality: such techniques do not satisfactorily filter out audio outside of the human audible range, and thus may not reflect how actual human listeners perceive audio quality. That is, they may be too conservative because they do not fully model human hearing limitations. This motivates our subjective, human-based assessment of audio quality, which we describe next.

5.4.2 User Study. To further understand the impact of Whisper on the quality of decoded audio, we conduct a small user study that asks human evaluators to rate the quality of a given audio sample. For the user study, we randomly choose eight audio samples—with four female and four males speakers—from our dataset. For each of

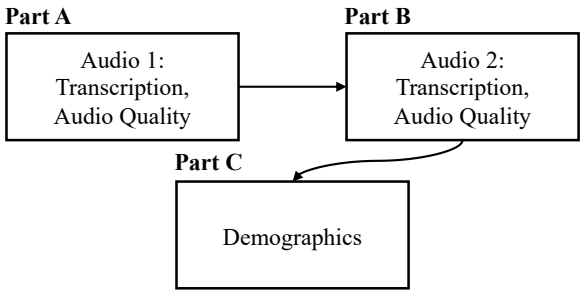

Figure 5: Sections and flow of the user study.

Metric	Percentage	Metric	Percentage
Gender		**Age**	
Female	32.9%	18-29 years	39.4%
Male	67.1%	30-49 years	53.3%
		50-64 years	6.5%
Ethnicity		65+ years	0.7%
Caucasian	75.9%		
African American	6.5%	**Education**	
Hispanic or Latino	11.6%	H.S. or below	13.1%
Asian	5.8%	Some college	24.8%
Other	4.4%	B.S. or above	62.0%

Table 1: Participant demographics for the user study. Percentages may not add to 100% due to non-response or selection of multiple options.

these eight samples, we also select the corresponding audio files produced with Opus without Whisper and with Whisper. For the Whisper-encoded version, we select the candidates encoded with the *max* fallback option, with the packet cost weight ratio between adjacent packet sizes set to 1 and the strictness parameter set to 0 (see §4.2). Thus, we use a total of 24 audio files in our user study encoded in three ways. The audio samples presented to the human evaluators are available at https://www.whisperIntoVoIP.com.

Figure 5 illustrates the design of our online user survey. In Part A of the survey, the participants first listen to an audio sample and are asked to transcribe it. This ensures that the participants actually listened to the audio and also informs us whether they are able to understand the spoken audio content. The participants are then asked to rate the overall audio quality on a five point Likert scale from Bad to Excellent (or Excellent to Bad, to minimize ordering effects). They are then asked to briefly explain their choice of rating as a free text response.

Part B of the survey asks the same questions as Part A but for an audio that differs in the encoding method and the spoken content from the audio presented in Part A. Finally, Part C concludes the survey with demographic questions about education, gender, ethnicity, age, income, and employment.

Recruitment. We used Amazon's Mechanical Turk (MTurk) crowdsourcing service to recruit participants for the user study. We required participants to be at least 18 years old, fluent in English, and located in the United States. To improve data quality, we also required participants to have at least a 95% HIT approval rate [15]. Participants were paid $1.00 for completing the study, which was reviewed and approved by our institution's ethics board. The demographics of our participants are summarized in Table 1.

	No Codec	Opus	Opus with Whisper
Responses	93	89	92

Table 2: Number of responses for each type of audio depending on how the audio was encoded.

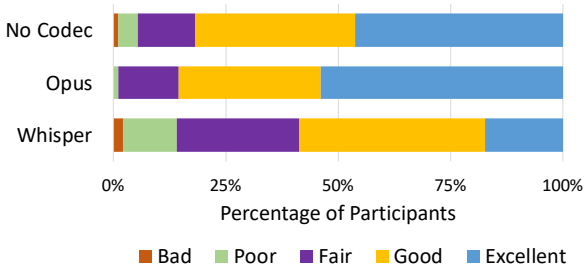

Figure 6: Audio quality as reported by human evaluators for audio with no encoding and audio encoded with the Opus codec with and without Whisper defense.

Results. In total, 150 MTurk human evaluators participated and completed our study. We exclude three responses as duplicates based on their originating IP addresses and only consider their first response. We also discard 10 responses that provided unintelligible transcriptions. For the remainder of the paper, we refer to the remaining 137 survey participants. Table 2 shows the number of responses for each type of audio presented.

Figure 6 summarizes the audio quality as reported by the survey participants. For the baseline audio with no encoding, the participants reported 4.2 as the average audio quality. When the audio was encoded with Opus (without Whisper protections), the average audio quality was 4.4. The participants, therefore, did not perceive any significant difference in the quality of audio when encoded with the Opus codec.

For audio encoded with Opus and protected using Whisper, participants reported an average audio quality of 3.6. On examining the reasoning behind the responses, one participant reported that he could hear a bit of static in the background but everything else was clear. Another participant said that there was noise in the background but could fully understand the audio. We remark that all of the study participants were able to correctly understand the contents of the spoken audio, even when they reported hearing artifacts or background noise. The perceived audio quality reported by the participants of the user study indicates that Whisper has no effect on listeners' ability to understand the audio and only introduces minimal noise.

Comparison of ViSQOL and User Study Results. The results obtained from the ViSQOL metric and the user study are largely consistent. For example, for audio encoded with the unprotected Opus codec, the audio quality reported by the ViSQOL metric (4.6) is close to that obtained from the user study (4.4). Similarly, both approaches report an average audio quality of 3.6 for audio encoded with Opus equipped with Whisper. Overall, our two techniques are consistent in showing that Whisper does not significantly impact audio quality and does not affect the perception of audio content.

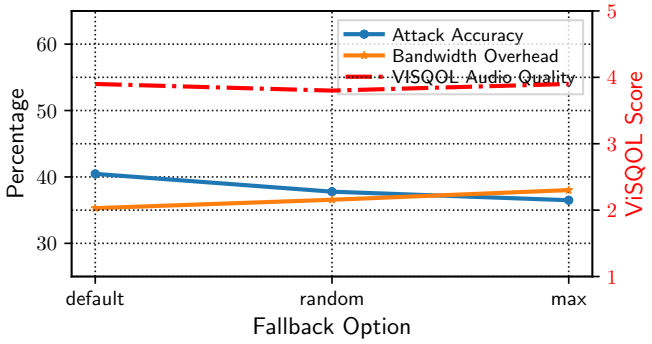

Figure 7: Effect of fallback options on various performance metrics.

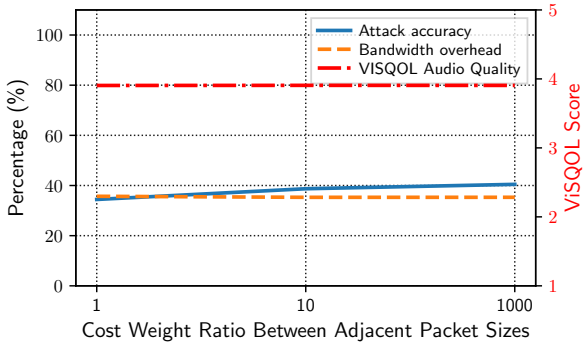

Figure 8: Effect of different *pktSizeWeights* values on various performance metrics.

6 TUNING WHISPER

Whisper has a number of configuration parameters that influence its effectiveness in thwarting re-identification attacks, its impact on audio quality, and its bandwidth overheads. In this section, we highlight some important points in this parameter space.

6.1 Effect of Fallback Options

Whisper overlays tuning audio on input audio before it is encoded by the sender's VBR codec; the choice of tuning audio is dictated by the target packet size as determined using the superdistribution. As discussed in §4.3, Whisper may fail to achieve the target packet size within the 20 ms window in which it needs to modify the audio frame. In such (rare) cases, Whisper can choose from the *default*, *random* or *max* fallback options (see §4.3 for details).

Figure 7 shows the effect of the various fallback options on attack accuracy, bandwidth overhead, and audio quality (note that lower is better for the first two performance metrics). Overall, the choice of fallback option has only a minor effect on the three performance metrics. Falling back to the maximum (*max*) packet size only slightly reduces the attack accuracy while incurring slight bandwidth overhead. Audio quality, as measured using ViSQOL, also remains almost the same across different fallback options. Thus, a user can safely configure Whisper to use any of the fallback options.

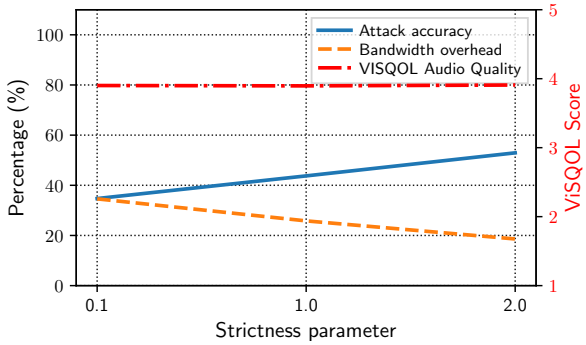

Figure 9: Effect of *strictness* parameter on performance metrics.

6.2 Effect of MorphStream Parameters

The MorphStream procedure (Algorithm 1) determines the output packet size for each input audio frame based on the superdistribution. The user specifies the *pktSizeWeights* and the *strictness* parameters to favor security or performance.

Figure 8 shows the effect of the cost weight ratio between adjacent packet sizes on various performance metrics when the default fallback scheme is used. As discussed in §4.2, the *pktSizeWeights* parameter influences the relative cost of choosing a packet size among all target packet sizes. By increasing the relative cost between packet sizes, the cost of selecting a larger packet size increases, resulting in a comparatively lower bandwidth overhead as smaller-sized output packets become more favorable. This also results in increased attack accuracy for the attacker as MorphStream may now choose a smaller packet size which can result in a packet size distribution that does not closely resemble the superdistribution. However, as shown in Figure 8, these effects are small. When *pktSizeWeights* is set to the maximum tested value, the attack accuracy rose to approximately 40% while providing little bandwidth savings. This indicates that a reasonable value of *pktSizeWeights* is 1, maximizing the efficacy of the attack while imposing little bandwidth overheads.

Figure 9 shows the effect of the *strictness* parameter on various performance metrics. The non-negative *strictness* parameter determines how strictly the target distribution should match the superdistribution. A smaller value results in stricter adherence to the superdistribution, achieving greater security. As the strictness parameter increases, the MorphStream procedure boosts the selection probability of smaller packet sizes, even though it may cause the target distribution to stray from the superdistribution. The strictness parameter allows the user to trade off between security and bandwidth savings, but (as shown in the Figure) does not affect the decoded audio quality. We consider a default value of 0 for the strictness parameter as it does not allow deviation from the superdistribution thus providing maximum security while incurring modest bandwidth overhead.

7 CONCLUSION AND DISCUSSION

In this paper, we propose the first unilateral defense, Whisper, for thwarting traffic analysis of encrypted VoIP streams. One of the major limitations of previously proposed blackbox defenses is that they require support from both the sender and receiver sides of

a VoIP stream; that is, both of the communicating parties' VoIP clients must support the defense. Unfortunately, to our knowledge, no such VoIP client has implemented existing defenses. In contrast, Whisper enables unilateral protections that can be deployed by either communicating party, without requiring the participation of the other and without modifying the VoIP client. Whisper is thus compatible with existing closed-source VoIP software.

Building on existing work, and leveraging the mechanisms of audio perception in humans, Whisper uses tuning audio at the boundaries of the human audible range to manipulate the size of the audio frames generated by VBR codecs. Our experiments demonstrate that Whisper significantly degrades the accuracy of re-identification attacks while incurring only a small loss in audio quality. Additionally, Whisper preserves much of the bandwidth savings of VBR.

Although in this paper, we focus on two-party VoIP communication, Whisper is also practical for improving the security of group communication. Here, speakers can apply Whisper to protect the privacy of their individual speech. We leave an evaluation of Whisper in the multiparty setting as a future research direction.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments. We also thank Moore et al. for providing the code for the superdistribution generation. This work has been partially supported by the National Science Foundation under grant number CNS-1718498. The views expressed in this work are strictly those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

[1] 2019. Call Quality Metrics. https://www.voip-info.org/call-quality-metrics/.
[2] 2019. Opus Interactive Audio Codec. http://opus-codec.org/.
[3] 2019. P.10 : Vocabulary for performance, quality of service and quality of experience. https://www.itu.int/rec/T-REC-P.10.
[4] 2019. SoX - Sound eXchange. http://sox.sourceforge.net/.
[5] 2019. VoxForge. http://www.voxforge.org/.
[6] Xiang Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. 2012. Touching from a Distance: Website Fingerprinting Attacks and Defenses. In *ACM Conference on Computer and Communications Security (CCS)*.
[7] Manuel Crotti, Maurizio Dusi, Francesco Gringoli, and Luca Salgarelli. 2007. Traffic Classification Through Simple Statistical Fingerprinting. *ACM SIGCOMM Computer Communication Review* 37, 1 (2007), 5–16.
[8] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. Tor: The Second-Generation Onion Router. In *USENIX Security Symposium (USENIX)*.
[9] Philipp Hancke. 2015. webrtcH4cKS: - What's up with WhatsApp and WebRTC? https://webrtchacks.com/whats-up-with-whatsapp-and-webrtc/.
[10] Henry E Heffner and Rickye S Heffner. 2007. Hearing Ranges of Laboratory Animals. *Journal of the American Association for Laboratory Animal Science* 46, 1 (2007), 20–22.
[11] Andrew Hines, Jan Skoglund, Anil C Kokaram, and Naomi Harte. 2015. ViSQOL: an objective speech quality model. *EURASIP Journal on Audio, Speech, and Music Processing* (2015).
[12] Andrew Hintz. 2003. Fingerprinting Websites Using Traffic Analysis. In *Privacy Enhancing Technologies Symposium (PETS)*.
[13] L. A. Khan, M. S. Baig, and Amr M. Youssef. 2010. Speaker Recognition from Encrypted VoIP Communications. *Digital Investigation* (2010).
[14] W. Brad Moore, Henry Tan, Micah Sherr, and Marcus A. Maloof. 2015. Multi-Class Traffic Morphing for Encrypted VoIP Communication. In *Financial Cryptography and Data Security (FC)*.
[15] Eyal Peer, Joachim Vosgerau, and Alessandro Acquisti. 2014. Reputation as a sufficient condition for data quality on Amazon Mechanical Turk. *Behavior research methods* 46 (2014).
[16] Nate Rand. 2017. Bandwidth Consumption. https://www.top10voiplist.com/bandwidth-consumption/.
[17] K. Vox, S. Jensen, and K. Soerensen. 2010. *SILK Speech Codec.* Internet-Draft draft-vos-silk-01. Internet Engineering Task Force.
[18] Tao Wang, Xiang Cai, Rishab Nithyanand, Rob Johnson, and Ian Goldberg. 2014. Effective Attacks and Provable Defenses for Website Fingerprinting. In *USENIX Security Symposium (USENIX)*.
[19] Andrew M White, Austin R Matthews, Kevin Z Snow, and Fabian Monrose. 2011. Phonotactic reconstruction of encrypted VoIP conversations: Hookt on fon-iks. In *Security and Privacy (SP), 2011 IEEE Symposium on*. IEEE, 3–18.
[20] Charles V Wright, Lucas Ballard, Scott E Coull, Fabian Monrose, and Gerald M Masson. 2008. Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*.
[21] Charles V Wright, Lucas Ballard, Fabian Monrose, and Gerald M Masson. 2007. Language identification of encrypted voip traffic: Alejandra y roberto or alice and bob?. In *USENIX Security Symposium*, Vol. 3. 43–54.
[22] Charles V Wright, Scott E Coull, and Fabian Monrose. 2009. Traffic Morphing: An Efficient Defense Against Statistical Traffic Analysis.. In *NDSS*, Vol. 9.