ADVoIP: Adversarial Detection of Encrypted and Concealed VoIP

Paolo Addesso, Michele Cirillo, Mario Di Mauro, and Vincenzo Matta

***Abstract*—A network attacker wants to transmit Voice-over-IP (VoIP) traffic streams covertly. He tries to evade the detection system by manipulating the VoIP streams through padding, shifting and splitting operations, so as to conceal them amidst the Internet traffic. A defender wants to detect the manipulated VoIP streams. Tackling this problem from an adversarial perspective, we provide two contributions: i) we obtain a highly stylized representation of VoIP streams in terms of transmission frequency $\mathscr{F}$ and packet length $\mathscr{L}$, and characterize the $(\mathscr{F},\mathscr{L})$ region achievable by the attacker's transformation; ii) we formulate the VoIP detection game, and find both theoretical conditions as well as a practical algorithm to find the Nash equilibrium of the game. As a result, we are able to design an optimal (from the adversarial perspective) algorithm for VoIP detection, which is nicknamed as ADVoIP. Simulations over real network traces, and comparison with existing approaches, show the effectiveness of the proposed approach.**

***Index Terms*—Adversarial Detection, Nash Equilibrium, VoIP traffic, Network Security.**

I. Introduction and Motivation

CYBER-SECURITY ranks among the biggest challenges of modern times.

Disclosing the presence of concealed exchange of information between network users is a fundamental issue in cyber-security, which has tremendous impact on many application domains, such as lawful interception, prevention of terrorist attacks, network safeguarding. Therefore, the development of powerful techniques for detection of encrypted Voice-over-IP (VoIP) traffic is a crucial achievement.

The early traffic identification methods, such as port detection or signature recognition, are nowadays no longer effective. This is because recent protocols often exploit randomly picked ports and/or use an encrypted packets to obfuscate the application layer content. In order to overcome these issues, various traffic classification techniques based on statistical and/or learning methodologies have been recently proposed — see, e.g., [1], [2]. With reference to VoIP traffic, some techniques based on classic machine learning tools have been advanced in [3]–[5], where either supervised or unsupervised schemes are applied to suitable statistical features that characterize VoIP sessions (e.g., delay, jitter). The bottom line emerging from the aforementioned studies is that i) effective detection of VoIP traffic is in fact possible; and ii) the main traffic features for a successful discrimination are the packet lengths (typically small, to accommodate pieces of conversation and silence), and the packet inter-arrival times (typically short, to guarantee an uninterrupted conversation flow).

The authors are with DIEM, University of Salerno, via Giovanni Paolo II, I-84084, Fisciano (SA), Italy (e-mails: paddesso@unisa.it, michelecirillo1993@gmail.com, mdimauro@unisa.it, vmatta@unisa.it).

The situation changes dramatically if we give to a malicious user (the *attacker*) the power of manipulating the VoIP streams, in such a way that a conversation can still take place, while the stream is camouflaged as a non-VoIP activity. As a matter of fact, encrypted VoIP calls are nowadays used for many illegal purposes, i.e., for mere money interests (for instance, users interested in eluding government taxes), or even criminal interests (for instance, terrorists interested in communicating covertly). Noticeably, it has been documented that VoIP calls can be forwarded through the so-called *low-latency intermediate networks*[1] in order to make them untraceable [6], [7]. Differently from classic anonymizing networks, these low-latency networks must guarantee real-time communication. To this aim, they implement suitable manipulations on the incoming packets, subject to low-latency constraints. This is one particular example of manipulation that can be performed on VoIP streams.

Attacks of this type are often referred to as *evasion attacks*, because the malicious user manipulates the data in order to evade some detection mechanism [8]. In order to contrast the aforementioned forms of cyber-threats, the defender must devise proper detection strategies that are able to work *in the presence of manipulated VoIP streams*, with the main focus of optimizing the detection performance. At the other side, the attacker should concurrently select his manipulation strategy in order to minimize the probability of being detected by the defender. The conflicting objectives of the two players suggest that one should take an *adversarial* perspective.

One of the classic tools to tackle inference problems in the presence of adversaries is game theory [9]–[11], which has been productively exploited to solve various inference problems and, in particular, to solve adversarial detection problems. In [12], the adversarial detection game, where the attacker can modify a statistical source up to some tolerable distortion, is solved in the asymptotic (information theoretic) setting. Recent works address the relevant case where the statistical nature of the data is unknown, or only partially known. In [13], [14], the results of [12] are extended to account for the presence of training data, while adversarial machine learning techniques have been advocated in [8], [15]–[17]. The case of a two-side attack (i.e., with an attacker capable to alter the data under both hypotheses) is addressed in [18], whereas in [19] the optimal mass transport theory is used to address the source identifiability issue. A classic two-side attack is the so-called Byzantine attack, where malicious network nodes can alter the statistical nature of the data through insertion of fake

[1]Examples of implementations can be found, e.g., on torfone.org or on opaq.com/wp-content/uploads/OPAQ_Cloud_VOIP_Use_Case_Data_Sheet.pdf.

measurements [20], so as to confuse the detector.

In the present work, we address the VoIP detection problem from an adversarial perspective, where the attacker tries to conceal the VoIP traffic amidst the Internet through padding/shifting/splitting operations, subject to some physical constraints in terms of packet lengths and transmission frequency. These constraints are necessary to meet the low-latency requirements characterizing a VoIP session. We adopt here a non-asymptotic approach, where each individual traffic stream is represented through a pair of descriptive indicators, borrowed from the existing literature on VoIP detection, namely, the transmission "frequency" $\mathscr{F}$ (number of packets sent over the observation interval), and the packet length $\mathscr{L}$ (number of bytes transmitted over the observation interval). In this representation, a traffic stream corresponds to a single realization of a certain $(\mathscr{F}, \mathscr{L})$ pair. Thanks to the proposed representation, we arrive at a highly stylized model, which allows obtaining i) a neat geometric interpretation of the admissible attacker's transformation, *given the constraints proper of the particular VoIP application*; and ii) the characterization of the Nash equilibrium of the game, along with an algorithm, nicknamed as ADVoIP, to determine the attacker's and defender's strategies at equilibrium.

It is useful to remark an essential property of the proposed solution. As typical in the security framework, designing a countermeasure that is specifically tailored to one attack would give rise to the endless cat-and-mouse game where the defender runs after the attacker, forever. In contrast, thanks to the adversarial framework, the proposed analysis provides the best countermeasure (in the game-theory jargon, the best defender's response) *without relying on a particular form of attack*. In fact, we define an admissible class of transformations by considering the unavoidable constraints in terms of bandwidth and latency that the attacker must obey to guarantee real-time conversation. In this way, traditional VoIP applications (as we will explicitly show in the our experiments, see Sect. V further ahead) VoIP with intermediate low-latency networks, as well as many types of manipulated VoIP streams that one can conceive (given the bandwidth/latency constraints) can be automatically handled within our framework.

II. Related Work

The present article falls under the umbrella of adversarial detection. We now contrast our approach and results to related works, in order to highlight the main common points as well as the distinguishing features.

Recent works have formalized the problem of detection in the presence of adversaries — see, e.g., [12]–[14]. The general paradigm introduced in these articles relies on two fundamental ingredients: i) the attacker's power is formalized through some type of transform that he can apply to manipulate the data streams, subject to some constraints (e.g., the streams can be modified up to a maximum tolerable distortion); ii) the optimal solution to the adversarial detection problem is found by solving a game (i.e., looking for a Nash equilibrium) between the attacker and the defender. The present work matches exactly this paradigm. However, there exist substantial differences between the VoIP problem and the models used in the aforementioned works, as we now detail.

— *Limited time-horizon and ergodicity*. The theoretical framework considered in [12]–[14] focuses on sources (in our context, traffic streams) that are ergodic, and devises optimal detection strategies and attacks under the asymptotic framework of infinite source streams. In our peculiar case, we are faced with at least two main difficulties that make these approaches not applicable. First, the traffic streams (both VoIP and non-VoIP) are usually not well represented by sequences of i.i.d. realizations coming from the same distribution, nor, more in general, by ergodic sequences. Second, in typical VoIP applications, the observation window is limited (e.g., in the order of few minutes) and providing quick detection is critical.

— *Shape of attacker's strategies*. Another important difference that prevents us from using the existing results in the VoIP context pertains to the attacker's strategies. The classic choice is assuming that the attacker can manipulate the sequence up to a certain average distortion computed between the transformed streams and the original ones. In the VoIP application, the attacker should keep under control basically two factors: i) the delay, in order to grant almost real-time communication; ii) the packet conservation, in order to preserve the total information. We see therefore that the VoIP application is sensitive to a form of *local* constraints that do not match well global metrics such as the average-distortion. Ignoring such constraints could lead to an inefficient detector design.

— *Limitations of our approach*. In order to manage the VoIP transformations according to the aforementioned physical constraints, in the present work we limit ourselves to examine two simple features, namely, the aggregate packet length and the average transmission frequency. For the same reasons, most of the existing works on adversarial detection rely on first-order, i.e., marginal statistics. For example, in [12] the empirical pmfs are employed (which are richer descriptive indicators than the average indicators used in this work).

In order to investigate the impact of using a limited set of features, in the section devoted to numerical simulations we compare the performance of our strategy with the performance of existing detectors that leverage enlarged sets of features. As we will see, the bottom line of the conducted analysis is that the enlarged set of features cannot compensate the disadvantages produced by the fact that the corresponding detectors are not designed to operate under the VoIP constraints. On the other hand, extending our theoretical analysis to enlarged set of features requires the evaluation of the best attacker's transformation in more complicated feature spaces, which seems to be a highly non-trivial task.

III. VoIP Traffic Features

A. Physical Model for Primitive Voice Streams

We start by describing the essential characteristics of a voice stream. In particular, it is useful to identify the peculiarities of a *primitive* voice stream, namely, of a *raw-data* stream that has undergone only minimal encoding operations. Characterizing a primitive stream is useful because the packetized streams measured across the network can be always seen as

a transformation of the primitive streams, subject to some fidelity/latency constraints. In this respect, both legitimate transformations (e.g., arising from existing VoIP encoders) as well as malicious transformations (e.g., arising from users that want to conceal the existence of the voice stream) can be regarded as manipulations of the primitive streams.

The construction of the primitive stream is realized through the following steps, which represent the preliminary procedure that lies at the heart of virtually any VoIP encoder.

— The vocal signal is sampled and quantized.

— A Voice Activity Detection (VAD) algorithm is implemented to identify the silence intervals. The resulting samples are then deleted at the transmission stage.

— At regularly-spaced time intervals, the samples corresponding to active periods are encoded into an IP/UDP/RTP packet with a certain overhead.

As a result, the skeleton of a packetized voice stream will be made of ON/OFF (i.e., talk and silence) periods. During a talk period, the voice is encoded into a sequence of almost regularly spaced (in time) packets, with almost constant size.

In view of the considered application, in the forthcoming treatment it will be particularly convenient to work with a *slotted* system. Accordingly, we partition the time axis into elementary slots that, for the sake of concreteness, have a duration of 1 ms. The interval between subsequent transmissions (a.k.a. *packetization time*) is usually determined by the sampling theorem and by the chosen level of buffering, and the corresponding number of slots will be denoted by Δ. Typical values for Δ are in the order of few tens.

The packet size depends upon a number of factors, including reproduction fidelity, overhead size (needed, e.g., to manage standard network operations, or for cryptography purposes), additional bits for improving "analog" voice perception (e.g., comfort noise). Clearly, different applications give rise to different behaviors as regards the final VoIP streams. However, we recall that a *primitive* VoIP stream corresponds to only minimal encoding operations, and that the extra-bytes used for the aforementioned purposes can be always incorporated in the transformation of a primitive stream. Therefore, a VoIP stream constructed according to such a procedure does not transmit during a silence period, while during the talking periods it transmits packets of almost-constant length, equal to $\mu = b\Delta$, where b is the chosen bit-rate.

In Fig. 1 we display the pipeline corresponding to the aforementioned description of VoIP primitive streams. In particular, we display the result of a real experiment that we conducted to construct the voice stream. The blue curve represents the sampled-and-quantized output of the analog signal recorded by a PC microphone, whereas the stepwise red curve represents the ON/OFF skeleton of the output of the VAD algorithm.

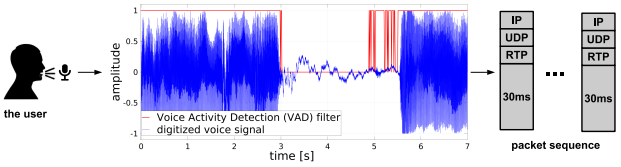

Fig. 1. An exemplification of basic VoIP encoding and packetization.

B. Streams Representation in the Feature Space

In order to perform VoIP traffic identification, the network analyst collects traffic streams from across the monitored network. Each (VoIP or non-VoIP) traffic stream can be partitioned into time-windows of duration equal to the packetization time, Δ. In practice, each window is composed by discrete time-slots of some small duration, which, as said, will be here set to 1 ms. We assume that the minimum (nonzero) packet length in a single slot is 1 byte. Likewise, we assume that the maximum packet length in a single slot is equal to the maximum transfer unit, denoted by mtu.

Motivated by existing works [3]–[5], we select the following two features to represent the VoIP traffic streams: i) the aggregate transmission frequency $\mathscr{F}$ (number of packets sent over the observation interval), and ii) the aggregate packet length $\mathscr{L}$ (number of bytes transmitted over the observation interval). Formally, let f_i be the number of the packets sent into the i-th window, let ℓ_i be the number of bytes transmitted during the i-th window, and let n be the number of windows. The selected features can be accordingly written as:

$$\boxed{\mathscr{F} \triangleq \sum_{i=1}^{n} f_i, \quad \mathscr{L} \triangleq \sum_{i=1}^{n} \ell_i} \tag{1}$$

We are now ready to see how the *primitive* VoIP streams will look like in the feature space. Given a particular VoIP stream, $\boldsymbol{v}$, we denote respectively by $f_i(\boldsymbol{v})$ and $\ell_i(\boldsymbol{v})$ the number of bytes and the number of packets related to stream $\boldsymbol{v}$, within the i-th window. Likewise, we introduce the aggregate quantities:

$$F(\boldsymbol{v}) \triangleq \sum_{i=1}^{n} f_i(\boldsymbol{v}), \quad L(\boldsymbol{v}) \triangleq \sum_{i=1}^{n} \ell_i(\boldsymbol{v}). \tag{2}$$

According to the above description, if in the i-th window there is silence, we have $f_i(\boldsymbol{v}) = \ell_i(\boldsymbol{v}) = 0$, while if there is talking activity, we have $f_i(\boldsymbol{v}) = 1$, and $\ell_i(\boldsymbol{v}) = \mu$ (we recall that each window has a duration corresponding to the packetization time). We remark that, since we assumed that the packet length in any slot is at least of 1 byte, the model $\ell_i(\boldsymbol{v}) = \mu$ naturally entails the condition $\mu \geq 1$. This condition is widely fulfilled by the most popular VoIP codecs — see, e.g. [21].

In terms of the aggregate features in (1), we get:

$$\boxed{L(\boldsymbol{v}) = \mu F(\boldsymbol{v})} \tag{3}$$

namely, by varying the talking frequency, the collection of primitive VoIP streams covers a "segment" with slope equal to the size of a transmitted VoIP packet, μ.

C. Attacker's Transformations

We consider an attacker (actually, a pair of attackers, the sender and the receiver) interested in communicating by VoIP covertly, i.e., in such a way that a traffic analyst (the defender) cannot detect the presence of the call. To achieve this goal, the attacker has the possibility of manipulating the primitive VoIP streams to some extent in order to evade the detection

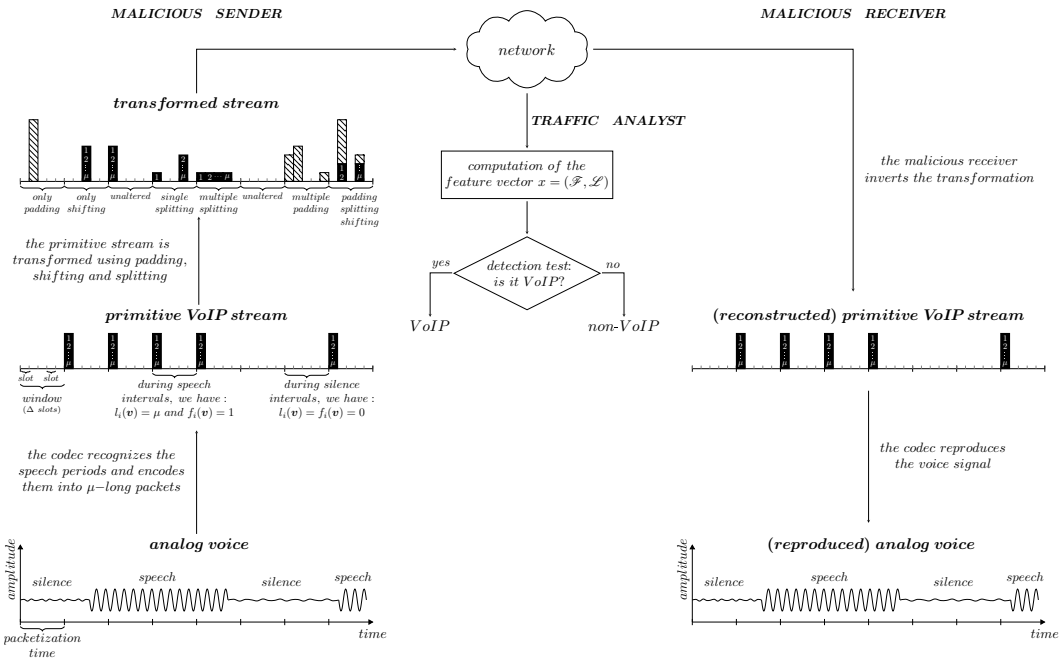

Fig. 2. Graphical sketch of the ADVoIP scenario.

system. For example, a VoIP stream could be transformed to make it similar to another type of stream (e.g., http). The manipulations permitted over a packet are the classic ones: padding, shifting and splitting. The amount of perturbation that can be implemented is essentially dictated by the fact that the sender and the receiver must be able to set up an effective VoIP call, namely, that suitable real-time/low-latency constraints must be fulfilled. The overall pipeline corresponding to the considered setting is illustrated in Fig. 2: starting from the lowermost/leftmost panel, the sender: first digitizes and packetizes the analog voice signal, giving rise to the primitive VoIP packet stream, and then applies some transformation (padding and/or shifting and/or splitting) before sending it over the communication channel. All these operations can be implemented in a *sequential (i.e., online)* manner, meaning that the transformation acts *window-by-window*.

It is useful to remark some important characteristics of the formal model adopted for the transformation. First, mere content encryption (i.e., modification of the bit values) is simply accounted for by imposing that the traffic analyst cannot decode, and, hence, that his decision rule must be based on content-independent features. Second, all protocol overheads related to specific operations (e.g., routing, encryption) can be mathematically modeled through the padding operation. Third, we do not impose a constraint on a *minimal* overhead that the transmitted packets must contain. Thanks to the latter choice, our analysis does not rely on a specific protocol, and is conservative because more power is given to the attacker.

While the transformed packets travel across the network, they are eavesdropped by a traffic analyzer, see middle panel of Fig. 2. After having accumulated a certain number of packets, the traffic analyzer attempts to detect the presence of the VoIP call. At the end of the pipeline, rightmost panels in Fig. 2, the intended (malicious) receiver inverts the transformation in order to reconstruct the primitive VoIP stream and to reproduce the conversation.

Let us now focus on the inversion of the transformation. In principle, padding, shifting and splitting are invertible operations. However, in order to perform the inversion, the receiver needs to know which particular mapping has been implemented by the sender. Since the number of possible mappings on a finite data streams is finite, the transformations can be indexed and the communicating parties can agree about a particular transformation beforehand. However, owing to the high dimensionality of the space of transformations, enumeration of the transformations might be wasteful, e.g., in terms of overhead and processing delay. These factors are critical for real-time communication, and therefore, they should be kept as small as possible.

In order to circumvent these complexity issues, in this section we show how to implement a simple and efficient invertible transformation. To this aim, let us consider the two transformations shown in Fig. 3. These transformations are identical in terms of packet lengths and inter-arrival times. They differ only in how the original and the extra bytes are distributed across the packet. In particular, we notice that the stream in the middle panel of Fig. 3 presents an irregular mix of the two types of bytes, whereas the stream in the rightmost panel obeys the following regular scheme: $i)$ for any packet, the original bytes (if any) are always put in the final part of the packet; $ii)$ for any window, the original bytes are sent in the same order as they were in the original stream. Thanks to these two properties, such a regular transformation can be readily inverted, provided that one knows how many bytes are contained in any packet. The remaining bytes can be removed, and packet reconstruction becomes then automatic in view of the bytes placement and ordering (points $i)$ and $ii)$ above). The information about the content of each transformed packet

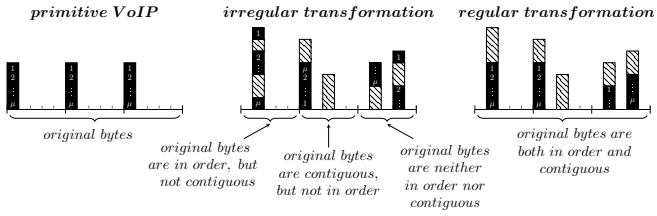

Fig. 3. A primitive stream (leftmost panel) can be transformed either into a generic irregular stream (middle panel) or into a regular one (rightmost panel). In the regular case the original bytes are placed contiguously at the end of the packets, in the same order as they appear in the primitive stream.

can be incorporated in the header with minimal overhead[2].

Once a particular set of padding/shifting/splitting operations has been chosen, we get a certain point in the $(\mathscr{F},\mathscr{L})$ feature space. While there exist in general several transformations corresponding to the same point, it is important to remark that any point in the admissible feature space can be in particular reached through a *regular* transformation.

D. Achievable $(\mathscr{F},\mathscr{L})$ Region

The attacker is allowed to manipulate the primitive VoIP streams through padding, shifting and splitting operations, namely, he can add extra-bytes, move or divide packets across the slots — see Fig. 2 for a graphical illustration.

In order to do this he can manipulate the quantities f_i's and ℓ_i's introduced in the previous section. However, such manipulations are unavoidably subject to some constraints, which can be classified as *physical constraints* or *VoIP-application constraints*. Violating a physical constraint leads to a stream that is not physically realizable. In contrast, violating a VoIP-application constraint leads to a stream that is realizable, but corresponds to an unsatisfactory quality of the resulting VoIP call. The physical constraints can be formally abstracted as follows:

$$0 \le f_i \le \Delta, \qquad f_i \le \ell_i \le f_i\,\mathsf{mtu}. \tag{4}$$

The leftmost constraint in (4) imposes that the number of packets per window cannot be negative and cannot exceed the number of slots, Δ. The rightmost constraint in (4) signifies that, when the attacker transmits f_i times in a window, he has to transmit at least 1 byte in the single slot, and at most mtu bytes per slot.

Let us now switch to the analysis of the VoIP-application constraints. The first constraint relates to the necessity of preventing channel overload, in order to cope with the *real-time/low-latency* requirement that is unavoidable for the considered (voice) application. Such a requirement can be formalized in terms of a maximum allowable number of bytes per window that the transformed sequence can contain. Denoting such a value by $\ell_{\max}$, the constraint is formally expressed as $\ell_i \le \ell_{\max}$ and $f_i \le \ell_{\max}$.

The second constraint relates instead to the necessity of preserving the full information associated to the original VoIP sequence, $\boldsymbol{v}$, and can be expressed by imposing that the number of bytes delivered in each window is at least equal to the number of bytes of the original sequence, namely, $\ell_i \ge \ell_i(\boldsymbol{v})$ and $f_i \ge f_i(\boldsymbol{v})$. In summary, the VoIP-application constraints can be expressed as:

$$f_i(\boldsymbol{v}) \le f_i \le \ell_{\max}, \qquad \ell_i(\boldsymbol{v}) \le \ell_i \le \ell_{\max}. \tag{5}$$

For later use, it is also necessary to introduce the following inequalities:

$$\Delta < \ell_{\max} < \mathsf{mtu}. \tag{6}$$

The leftmost inequality in (6) holds true in our setting since the typical value of Δ is in the order of a few tens, while the maximum packet length per window is usually in the order of some hundreds of bytes. The rightmost inequality in (6) holds true in our setting since mtu is usually in the order of thousands of bytes, which are typically not compatible with the real-time/low-latency requirement given the short time-duration Δ — see, e.g., [21].

One natural question at this point is: How the malicious transformations will affect the quality of a VoIP call? More or less obviously, the padding/shifting/splitting operations can impact to some extent on the bandwidth requirements, i.e., on factors like congestion, jitter, and packet loss. According to our attack model, the bandwidth cannot increase more than $\ell_{\max}$ bytes over Δ milliseconds. Typical values for these parameters are $\Delta = 30$ and $\ell_{\max} = 1000$, which have little impact on the bandwidth.

In addition, a critical factor influencing the quality of a VoIP call is the so-called "mouth-to-ears delay", namely, the time taken by the sound to reach the listener's ears after leaving the speaker's mouth [22]. However, any admissible transformation cannot move a packet outside its original time window, which means that the delay introduced by the transformation cannot go beyond tens of milliseconds. Since, for typical applications, the mouth-to-ears time can safely reach the order of some hundreds of milliseconds [23], we conclude that the delay introduced by the transformation does not lead to significant reduction of the perceived quality.

Coming back to the mathematical formulation of the transformation constraints, it is useful to express in a more convenient form the various inequalities obtained so far. To this aim, we start by observing that the physical constraints in (4) and the VoIP-application constraints in (5) can be condensed in the following compact form:

$$\max\{f_i(\boldsymbol{v}), 0\} \le f_i \le \min\{\ell_{\max}, \Delta\}, \tag{7}$$

and

$$\max\{l_i(\boldsymbol{v}), f_i\} \le \ell_i \le \min\{\ell_{\max}, f_i\,\mathsf{mtu}\}. \tag{8}$$

By recalling that *i)* either $f_i(\boldsymbol{v}) = l_i(\boldsymbol{v}) = 0$ or $f_i(\boldsymbol{v}) = 1$ and $l_i(\boldsymbol{v}) = \mu$, and *ii)* that $\Delta < \ell_{\max} < \mathsf{mtu}$, the constraints can be finally written as:

$$\boxed{f_i(\boldsymbol{v}) \le f_i \le \Delta} \tag{9}$$

$$\boxed{\begin{cases} \max(f_i, \mu) \le \ell_i \le \ell_{\max}, & \text{if } f_i(\boldsymbol{v}) = 1 \\ f_i \le \ell_i \le \min(f_i, 1)\,\ell_{\max}, & \text{if } f_i(\boldsymbol{v}) = 0 \end{cases}} \tag{10}$$

[2]The length of a primitive VoIP packet within the packetization window is 0 or μ. Accordingly, the bytes in any *transformed* packet vary from 0 to μ, and, hence, they can be safely indexed by a few bits.

In summary, *the attacker can transform a VoIP stream $\boldsymbol{v}$ into any stream having f_i's and l_i's that cope with the constraints (9) and (10)*. The ensemble of admissible transformation leads to the concept of achievable region: in the feature space the original primitive VoIP streams $\boldsymbol{v}$ occupies the point $(F(\boldsymbol{v}), L(\boldsymbol{v}))$, whereas the generic transformation of $\boldsymbol{v}$ occupies the point $(\mathscr{F}, \mathscr{L})$ with $\mathscr{F} \triangleq \sum_{i=1}^{n} f_i$ and $\mathscr{L} \triangleq \sum_{i=1}^{n} \ell_i$. This means that in the feature space a transformation of the stream $\boldsymbol{v}$ consists of the point translation $(F(\boldsymbol{v}), L(\boldsymbol{v})) \rightarrow (\mathscr{F}, \mathscr{L})$. We define the attacker's achievable region for the stream $\boldsymbol{v}$ as *the set of all points $(\mathscr{F}, \mathscr{L})$ that are reachable from $(F(\boldsymbol{v}), L(\boldsymbol{v}))$ by such a translation*.

The following theorem provides closed-form expressions for the achievable region.

Theorem 1 (Achievable region): Consider a primitive VoIP sequence $\boldsymbol{v}$. Under the constraints (9) and (10), with $\ell_{\max}$ fulfilling (6), the admissible range of frequencies is:

$$\boxed{F(\boldsymbol{v}) \leq \mathscr{F} \leq n\,\Delta} \tag{11}$$

Let now $\zeta \triangleq \min(\Delta, \mu)$. Then, for $\mathscr{F}$ spanning this range, the minimum and maximum achievable lengths, $\mathscr{L}_{\min}(\mathscr{F})$ and $\mathscr{L}_{\max}(\mathscr{F})$, are, respectively:

$$\boxed{\mathscr{L}_{\min}(\mathscr{F}) = \begin{cases} \mu F(\boldsymbol{v}), & \text{if } \mathscr{F} \leq \zeta F(\boldsymbol{v}) \\ \mathscr{F} + (\mu - \zeta) F(\boldsymbol{v}), & \text{if } \mathscr{F} > \zeta F(\boldsymbol{v}) \end{cases}} \tag{12}$$

and

$$\boxed{\mathscr{L}_{\max}(\mathscr{F}) = \begin{cases} \mathscr{F}\,\ell_{\max}, & \text{if } \mathscr{F} \leq n \\ n\,\ell_{\max}, & \text{if } \mathscr{F} > n \end{cases}} \tag{13}$$

Furthermore, if

$$\frac{F(\boldsymbol{v})}{n} \geq \frac{1}{\ell_{\max} - \mu + 1}, \tag{14}$$

then the attacker's achievable region for the considered VoIP stream $\boldsymbol{v}$ is the set of all the points lying inside the region described by the aforementioned boundaries, including the boundaries themselves.

Proof: See Appendix A. ∎

We now explain why condition (14) is relevant in practice. As a matter of fact, $F(\boldsymbol{v})$ measures the number of windows containing speech activity, and since n is the number of the existing windows, the fraction $\frac{F(\boldsymbol{v})}{n}$ measures the fraction of time that the user has actually spoken. Now, in practice the value of $\ell_{\max}$ is typically $\gtrsim 500$ bytes, whereas μ is typically $\lesssim 40$ bytes [21]. Accordingly, assumption (14) holds for all the VoIP calls such that $\frac{F(\boldsymbol{v})}{n} \geq \frac{1}{\ell_{\max}-\mu+1} \approx 10^{-2}$, namely, for all the VoIP calls where a user speaks at least one hundredth of the available time (corresponding to about a half second per minute).[3]

We now introduce and examine a special property of the transformed region.

REMARK I (*Transformed points cannot move backward*). Let us consider the primitive VoIP cells in the feature space. As said, they are spread over a segment. Let us also assign progressive numbers to the cells of the feature space that compose such segment, in the direction where the frequency increases. Accordingly, a particular point along the segment can be denoted by x_j, with $j = 1, 2, \ldots, n$. In Fig. 4 we depict the admissible region corresponding to the points lying over the segment. In particular, we examine the situation corresponding to a pair of consecutive points, x_{j-1} and x_j. We see that there is an L-shaped region that is reachable from x_{j-1}, but not from x_j. In this sense, we shall say that "*points cannot move backward*". This property will be very useful in the forthcoming analysis.

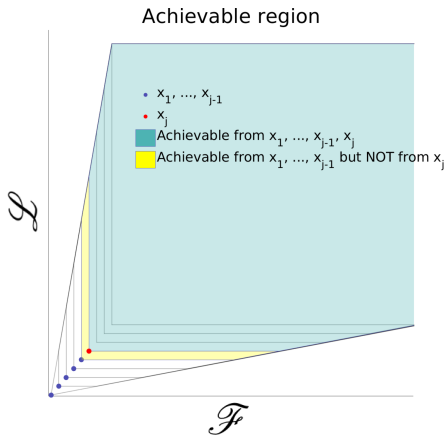

Fig. 4. Region achievable by the attacker's transformation (Theorem 1).

In the following, we denote the j-th L-strip by $\mathcal{S}_j$. For later use, it is useful to introduce the union of consecutive L-strips. Formally, we let $\mathcal{S}_{j_1:j_2} \triangleq \bigcup_{j=j_1}^{j_2} \mathcal{S}_j$, for all $j_1, j_2 = 1, 2, \ldots, n$, with $j_1 < j_2$. The overall region achievable by the admissible transformation is accordingly represented as $\mathcal{S}_{1:n}$. We remark that, since: i) each L-strip corresponds to a single primitive VoIP stream; and ii) in a primitive VoIP stream only one slot per window is filled, we conclude that the number of strips corresponds to the number of windows, n.

IV. The VoIP Detection Game

The feature space is a two-dimensional grid. We denote by $x = (\mathscr{F}, \mathscr{L})$ a single point of such grid. The statistical properties of the traffic streams in the feature space will be described through a probability mass function (pmf), defined over the points of the grid. More specifically, the pmf of primitive VoIP streams will be denoted by $P(x)$, whereas the pmf of non-VoIP streams will be denoted by $Q(x)$. To avoid trivial cases, we assume that non-VoIP streams can in principle occupy any portion of the feature space, namely, we assume $Q(x) > 0$ for all x.

We now introduce the game between the attacker (who tries to manipulate a primitive VoIP stream to disguise it as non-VoIP), and the defender (who tries to detect the VoIP stream). In our work, we adhere to the classic adversarial (i.e., game-theoretic) paradigm, where it is assumed that each player knows the set of admissible strategies of the other player, and both players act rationally, pursuing a noncooperative equilibrium in the following sense: each player will choose the best strategy taking in due account the presence of the other

[3]The proposed design makes sense even when (14) is not met, since considering the attacker's region in terms of its boundaries, even when some internal points are not achievable, leads clearly to a conservative design.

player (who is optimizing his own choice as well). From a mathematical viewpoint, a pair of equilibrium strategies guarantees that no player has convenience in deviating unilaterally from the chosen strategy.

As said, a strategy for the attacker consists of choosing a transformation rule that is realized in practice through padding/shifting/splitting operations acting on the primitive VoIP stream. Such a transformation corresponds, in the feature space, to move a primitive VoIP point x_j to another point belonging to the admissible region $\mathcal{S}_{1:n}$. In order to grant generality, the transformation is allowed to be random, i.e., there is some transfer probability $\tau(x|x_j)$, which produces the following transformed pmf $P_\tau(x)$:

$$\boxed{P_\tau(x) = \sum_{j=1}^{n} P(x_j)\tau(x|x_j)} \tag{15}$$

Randomness of the transformation implies that, for a given primitive VoIP stream (i.e., for a given point x_j corresponding to the strip $\mathcal{S}_j$ into the feature space), the sets of padding/shifting/splitting operations that move x_j to x is selected with probability $\tau(x|x_j)$. In particular, since, from Remark I, we know that the generic point x_j cannot move backward, we have $\tau(x|x_j) = 0$ for any $x \in \mathcal{S}_{1:j-1}$.

A strategy for the defender consists of a *decision rule* for accepting/rejecting the VoIP hypothesis. We shall consider the classic Neyman-Pearson (NP) detection setting. Under this formulation, the decision rule is described through the probability of accepting the VoIP hypothesis after observing x, which will be denoted by $\delta(x)$. The statistical performance of the detector will be characterized through the *false-alarm* and *miss-detection* probabilities, defined respectively as:

$$\boxed{\alpha_\delta = \sum_{x\in\mathcal{S}_{1:n}} Q(x)\delta(x), \quad \beta_{\tau,\delta} = 1 - \sum_{x\in\mathcal{S}_{1:n}} P_\tau(x)\delta(x)} \tag{16}$$

We start by examining the false-alarm probability α_δ, which quantifies the probability of mistakenly declaring a non-VoIP stream as VoIP. Accordingly, this type of error depends on: i) the pmf of the non-VoIP streams, $Q(x)$; ii) the decision rule, δ, which is active (i.e., nonzero) only in the points where VoIP is declared. As a result, we see that α_δ depends only on the defender's strategy, δ.

Next we move on to examine the miss-detection probability, $\beta_{\tau,\delta}$, which quantifies the probability of mistakenly declaring a (possibly transformed) VoIP stream as non-VoIP. Clearly, $\beta_{\tau,\delta}$ depends on the decision rule, δ. Moreover, the miss-detection probability must depend also on the actual transformation, τ, that has been applied to the primitive VoIP stream. As a matter of fact, the distribution $P_\tau(x)$, corresponding to the transformed streams, will be heavily dependent on the particular transformation. The corresponding miss-detection probability should be accordingly computed over such a modified pmf, $P_\tau(x)$.

Let us now explain how these two performance indices are combined in a classic adversarial framework. First of all, under the NP formulation a constraint is put on the false-alarm probability. Formally, given a false-alarm constraint $\overline{\alpha}$, the set of admissible defender's strategies is given by: $\{\delta : \alpha_\delta \leq \overline{\alpha}\}$.

Given the set of admissible strategies, the adversarial NP formulation leads to two opposite, i.e., conflicting, requirements for the defender and the attacker: the defender is interested in minimizing the miss-detection error, whereas the attacker is interested in letting the VoIP call propagate covertly, which translates into the mathematical requirement of maximizing the miss-detection probability.

In summary, the VoIP adversarial NP detection game can be formulated as follows. Given a pmf $Q(x)$ for the non-VoIP streams, an admissible defender's strategy consists of a decision rule δ ensuring that the false-alarm probability α_δ in (16) does not exceed a tolerable level $\overline{\alpha}$. Given a pmf $P(x)$ for the primitive VoIP streams, a transformation τ yields the pmf $P_\tau(x)$ in (15), which, along with the decision rule δ, determines the miss-detection probability $\beta_{\tau,\delta}$ in (16). The attacker's strategy τ and the defender's strategy δ are chosen by the players in a noncooperative way, and the conflicting objectives (the defender wants to minimize $\beta_{\tau,\delta}$, while the attacker wants to maximize it) must be managed according to an equilibrium criterion.

We are ready to introduce the *costs* associated to the attacker and to the defender. A straightforward and classic choice is as follows. For a given pair of admissible strategies, (τ, δ), the defender's cost is given by the probability of missing the attacker, $\beta_{\tau,\delta}$, whereas the attacker's cost is given by the probability of being discovered, $1 - \beta_{\tau,\delta}$. Following a standard formulation for adversarial detection under the NP framework [12], the latter cost can be obviously modified into the opposite of the miss-detection probability, i.e., $-\beta_{\alpha,\delta}$, yielding a *zero-sum game* [25].

We now provide a necessary and sufficient condition for reaching a Nash Equilibrium (NE) of the game. By definition, a couple of strategies (s_1, s_2) is a NE of the game *iff* the strategies are the best responses one for each other, namely, *iff* strategy s_1 is optimal for player 1 given that s_2 is selected by player 2, and vice versa. Put another way, if the pair (s_1, s_2) is chosen, no player has convenience in deviating unilaterally from the strategy he has chosen.

Accordingly, we compute first the defender's best response to a *generic attacker's transformation* (Lemma 1), and then the attacker's best response to a *generic defender's decision rule* (Lemma 2). The roadmap to find a NE amounts then to find a pair of strategies that meet simultaneously the two lemmas.

Lemma 1 (Optimal δ for a given τ — classic NP lemma): Let τ be an attacker's transformation — see (15). Then, an optimal defender's strategy (decision rule) $\delta^\star_\tau(x)$, is given by:

$$\boxed{\delta^\star_\tau(x) = \begin{cases} 1, & \text{if } P_\tau(x) > \eta_\tau Q(x) \\ \rho_\tau(x), & \text{if } P_\tau(x) = \eta_\tau Q(x) \\ 0, & \text{if } P_\tau(x) < \eta_\tau Q(x) \end{cases}} \tag{17}$$

In (17), the detection threshold η_τ can be chosen as the smallest value z ensuring that:

$$\sum_{x:P_\tau(x)>zQ(x)} Q(x) \leq \overline{\alpha}. \tag{18}$$

Actually, when (as happens in our setting) the likelihood ratio assumes only a finite set of values, the detection threshold η_τ can be chosen as the minimum of these values that fulfills (18). Moreover, assuming that the detection probability is not equal to 1 (which would be a trivial case), the randomizing probability $\rho_\tau(x)$ determines the false-alarm probability through the following equation [24]:

$$\sum_{x:P_\tau(x)>\eta_\tau Q(x)} Q(x) + \sum_{x:P_\tau(x)=\eta_\tau Q(x)} \rho_\tau(x)Q(x) = \overline{\alpha}. \quad (19)$$

Proof: The claim is the well-known NP lemma, whose proof can be found in several classic textbooks [24]. ∎

Before introducing Lemma 2, it is necessary to introduce some useful quantities. Let $\delta_{\min}(j)$ be the minimum value of the decision probability in the set $\mathcal{S}_{j:n}$, namely,

$$\delta_{\min}(j) \triangleq \min_{\xi\in\mathcal{S}_{j:n}} \delta(\xi). \quad (20)$$

The set containing these minima is denoted by:

$$\mathcal{S}_{j:n}^{(\min)} \triangleq \{x \in \mathcal{S}_{j:n} : \delta(x) = \delta_{\min}(j)\}. \quad (21)$$

Lemma 2 (Optimal τ for a given δ): Let δ be a defender's decision rule. Then, an optimal attacker's strategy (pmf transformation), $\tau_\delta^\star$, must fulfill the following condition, for all $j = 1, 2, \ldots, n$, and for all $x \in \mathcal{S}_{j:n}$:

$$\boxed{\delta(x) > \delta_{\min}(j) \Rightarrow \tau_\delta^\star(x|x_j) = 0} \quad (22)$$

Proof: See Appendix B. ∎

A. Useful Algorithm

We introduce preliminarily the following useful quantity:

$$\Lambda_{i:j} \triangleq \frac{\sum_{x\in\mathcal{S}_{i:j}} P(x)}{\sum_{x\in\mathcal{S}_{i:j}} Q(x)}. \quad (23)$$

The above equation represents in a sense the "quantized" likelihood ratio corresponding to the coarser information of knowing that x belongs to the union of strips $\mathcal{S}_{i:j}$, without details about the specific location x. Accordingly, the ratio in (23) will be referred to as the quantized likelihood.

Lemma 3 (Identification of a special attack): Let a_k and b_k, for $k \geq 1$, be defined recursively as follows (in presence of multiple solutions, argmin selects the lowest index):

$$a_1 = 1, \qquad b_1 = \operatorname*{argmin}_{j\geq a_1} \Lambda_{a_1:j}, \quad (24)$$

$$a_2 = b_1 + 1, \qquad b_2 = \operatorname*{argmin}_{j\geq a_2} \Lambda_{a_2:j}, \quad (25)$$

and so on, until we reach the condition $b_m = n$ (we recall that n is the total number of strips) for a certain integer m. Moreover, let the union-of-strips between a_k and b_k be $\Sigma_k \triangleq \mathcal{S}_{a_k:b_k}$. Then, there always exists an attack $\tau^\star$ with the shape in Fig. 5, namely, an attack featuring the following properties:
A. **Closed-door behavior**. For any $k = 1, 2, \ldots, m$, the attack does never transfer mass outside each region Σ_k, namely, it only moves mass from points in Σ_k to other points in Σ_k.
B. **Stepwise behavior**. For any x in the region Σ_k, the attack likelihood ratio $\frac{P_{\tau^\star}(x)}{Q(x)}$ is constant and equal to the *quantized* likelihood ratio in that region, which will be denoted by:

$$\gamma_k \triangleq \Lambda_{a_k:b_k}. \quad (26)$$

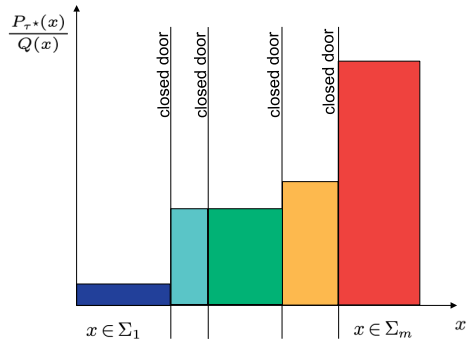

Fig. 5. Illustration of the special type of attack described in Lemma 3.

C. **Monotonic behavior**. The attack likelihood ratio is non-decreasing across the regions Σ_k, i.e., $\gamma_1 \leq \gamma_2 \leq \ldots \leq \gamma_m$.
Premise. We now describe an algorithm that always (i.e., for any configuration of the considered problem) transforms the original pmf $P(x)$ into a new one satisfying the three properties above. This algorithm acts independently for any region Σ_k. Therefore, it suffices to illustrate its behavior only with reference to the first region Σ_1. For the sake of clarity, it is convenient to omit the subscript 1, namely, we refer to the region $\Sigma = \Sigma_1$ which is implicitly identified by the indices $a = a_1$ and $b = b_1$. Likewise, the quantized likelihood ratio for this region will be denoted as $\gamma = \gamma_1$.

Proof: The proposed algorithm iterates over the L-strips that form the set Σ. The iteration index, say it h, spans over the interval $a, \ldots, b$ (remember that we are focusing on the region $\Sigma = \Sigma_1$, and therefore $a = a_1 = 1$). To avoid trivial cases we assume that $b > 1$.

It is useful to introduce preliminarily the following quantities. First, let x_h be the point of the VoIP segment belonging to the h-th strip, and let $P(x_h)$ the corresponding probability mass under the VoIP hypothesis. Second, let $\tilde{P}(x; h)$ be the transformed pmf arising from the h-th algorithm iteration, evaluated in a generic point x. The output of the algorithm will be the transformed pmf $P_{\tau^\star}(x) \triangleq \tilde{P}(x; h)$, for any $x \in \mathcal{S}_h$ and for $h = a, \ldots, b$. Now we are ready to describe the algorithm behavior for the region Σ.

Let us consider the iteration $h = 1$. In order to meet property B for the points in $\mathcal{S}_1$, the algorithm moves from $x_h = x_1$ to any $x \in \mathcal{S}_1$ a probability mass equal to:

$$\tilde{P}(x; 1) = \gamma Q(x). \quad (27)$$

This is always possible because the amount of probability present at the point x_1 is sufficiently high, namely,

$$P(x_1) > \sum_{x\in\mathcal{S}_1} \tilde{P}(x; 1) = \gamma \sum_{x\in\mathcal{S}_1} Q(x). \quad (28)$$

In fact because, by definition, the index b minimizes $\Lambda_{1:j}$ over j, and because we supposed $b > 1$, we have:

$$\frac{P(x_1)}{\sum_{x\in\mathcal{S}_1} Q(x)} \triangleq \Lambda_{1:1} > \Lambda_{1:b} \triangleq \gamma. \quad (29)$$

In particular, the strict inequality follows from the fact that b is defined as the *smallest* minimizer of $\Lambda_{1:j}$, namely, there is not a value $j < b$ that minimizes $\Lambda_{1:j}$.

As a second step, the algorithm moves from x_1 to x_2 the excess mass that has not been used for the operation in (27), which, according to (28), is equal to:

$$P(x_1) - \gamma \sum_{x \in \mathcal{S}_1} Q(x). \tag{30}$$

As a third step, the algorithm updates the pmf taking into account the transfer in (30), yielding:

$$\begin{aligned} \tilde{P}(x_2;1) &= P(x_2) + \left[P(x_1) - \gamma \sum_{x \in \mathcal{S}_1} Q(x) \right] \\ &= \sum_{x \in \mathcal{S}_{1:2}} [P(x) - \gamma Q(x)] + \gamma \sum_{x \in \mathcal{S}_2} Q(x), \end{aligned} \tag{31}$$

where, in the last equality: i) we added and subtracted the term $\gamma \sum_{x \in \mathcal{S}_2} Q(x)$, and ii) we used the fact that, since $P(x)$ is the distribution over the primitive streams, the only points where $P(x)$ is nonzero within $\mathcal{S}_{1:2}$ are x_1 and x_2.

At step $h = 2$, the algorithm proceeds to meet property B in $\mathcal{S}_2$, by moving from $x_h = x_2$ to any point $x \in \mathcal{S}_2$ a probability mass equal to $\tilde{P}(x;2) = \gamma Q(x)$. Now we show that even in this case the probability present in $x_h = x_2$ is sufficiently high, namely,

$$\tilde{P}(x_2;1) \geq \sum_{x \in \mathcal{S}_2} \tilde{P}(x;2) = \gamma \sum_{x \in \mathcal{S}_2} Q(x). \tag{32}$$

To this aim, recalling the expression in (31), we need to show that the excess mass term

$$\tilde{P}(x_2;1) - \gamma \sum_{x \in \mathcal{S}_2} Q(x) = \sum_{x \in \mathcal{S}_{1:2}} [P(x) - \gamma Q(x)] \tag{33}$$

is either positive or zero. The latter possibility (zero excess mass) occurs when $b = 2$, i.e., when the minimum in (24) is reached in the second strip. In this case, we have that

$$\frac{\sum_{x \in \mathcal{S}_{1:2}} P(x)}{\sum_{x \in \mathcal{S}_{1:2}} Q(x)} \triangleq \Lambda_{1:2} = \Lambda_{1:b} \triangleq \gamma, \tag{34}$$

which nullifies the excess mass in (33). In this case, the routine pertaining to the single region $\Sigma = \Sigma_1$ ends at step $h = 2$.

The former possibility (strictly positive excess mass) occurs when $b > 2$, yielding to a conclusion similar to (29), namely,

$$\frac{\sum_{x \in \mathcal{S}_{1:2}} P(x)}{\sum_{x \in \mathcal{S}_{1:2}} Q(x)} \triangleq \Lambda_{1:2} > \Lambda_{1:b} \triangleq \gamma, \tag{35}$$

which makes the excess mass in (33) strictly positive. Such excess mass is moved forward to $x_{h+1} = x_3$, and the pmf $\tilde{P}(x_3;2)$ is updated accordingly.

Iterating the above procedure, the algorithm can be compactly represented for the generic step $1 \leq h \leq b$ as follows:

$$\tilde{P}(x;h) = \gamma Q(x), \quad x \in \mathcal{S}_h, \tag{36}$$

$$\tilde{P}(x_{h+1};h) = \sum_{x \in \mathcal{S}_{1:h}} [P(x) - \gamma Q(x)] + \gamma \sum_{x \in \mathcal{S}_h} Q(x). \tag{37}$$

We are now in the position of establishing two useful properties. First, since the equalizing operation in (36) is feasible for all steps $1 \leq h \leq b$, the algorithm always meets property B. Second, since at the last iteration there is no excess mass, in the region Σ there is *exactly* the amount of positive mass necessary for equalization. This means that no other mass needs to be transferred outside Σ, and property A is automatically met.

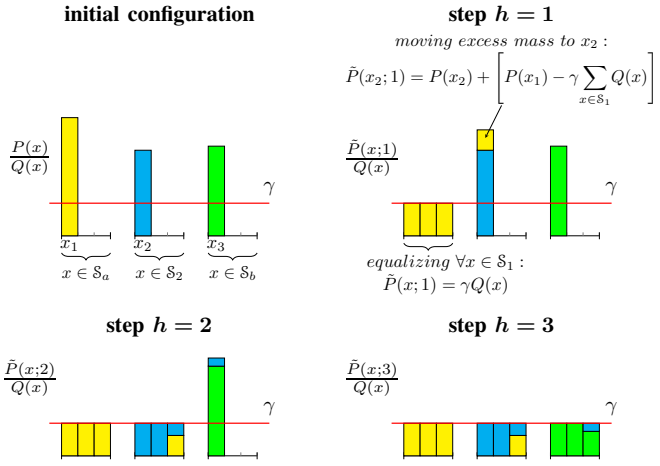

Fig. 6. A step-by-step illustration of the algorithm described in Sect. IV-A.

It remains to prove property C, namely, to show that $\gamma_k \triangleq \Lambda_{a_k:b_k} \leq \Lambda_{a_{k+1}:b_{k+1}} \triangleq \gamma_{k+1}$. We start by proving the following inequality:

$$\Lambda_{a_k:b_k} \leq \Lambda_{a_k:b_{k+1}}, \tag{38}$$

which follows readily from the definition of b_k as the smallest minimizer of $\Lambda_{a_k:j}$. Moreover, since $a_{k+1} = b_k + 1$, we can also write:

$$\begin{aligned} \sum_{x \in \mathcal{S}_{a_k:b_{k+1}}} P(x) &= \underbrace{\sum_{x \in \mathcal{S}_{a_k:b_k}} P(x)}_{A} + \underbrace{\sum_{x \in \mathcal{S}_{a_{k+1}:b_{k+1}}} P(x)}_{B}, \\ \sum_{x \in \mathcal{S}_{a_k:b_{k+1}}} Q(x) &= \underbrace{\sum_{x \in \mathcal{S}_{a_k:b_k}} Q(x)}_{C} + \underbrace{\sum_{x \in \mathcal{S}_{a_{k+1}:b_{k+1}}} Q(x)}_{D}, \end{aligned} \tag{39}$$

which gives:

$$\Lambda_{a_k:b_k} = \frac{A}{C}, \ \Lambda_{a_k:b_{k+1}} = \frac{A+B}{C+D}, \ \Lambda_{a_{k+1}:b_{k+1}} = \frac{B}{D}. \tag{40}$$

From (38) we have $(A/C) \leq (A+B)/(C+D)$, which implies that $(A/C) \leq (B/D)$, which in turn corresponds to say that $\Lambda_{a_k:b_k} \leq \Lambda_{a_{k+1}:b_{k+1}}$, and the proof of the lemma is complete. ■

In Fig. 6 we illustrate step-by-step the effects of the algorithm transformations on the likelihood, with reference to the region $\Sigma = \mathcal{S}_{a:b}$, with $a = 1$. Let us start from the initial situation, depicted into the uppermost-and-leftmost panel. Since the VoIP streams are originally distributed over the segment points $x_1, x_2, \ldots, x_n$, the likelihood ratio can take nonzero values only in such points. At the first step, after performing the equalizing operation in (27) and moving the excess mass forward as described in (31), the resulting situation is depicted into the uppermost-and-rightmost panel.

Here we observe that the likelihood ratio, within the region $\mathcal{S}_1$, is equal to γ everywhere, whereas the likelihood ratio is increased at $x_2 \in \mathcal{S}_2$, due to excess mass that has been here transferred. The same procedure applies to the other steps. It is worth noting that, without the excess mass coming from $x_1 \in \mathcal{S}_1$, the algorithm would not have been able to perform equalization within $\mathcal{S}_2$; in fact, in the considered example we see that the original quantized likelihood ratio in $\mathcal{S}$ is smaller than what would be necessary for an equalization.

B. Nash Equilibrium

We are now ready to prove the existence of a Nash Equilibrium for the VoIP detection game. We will show that the attacker's transformation illustrated in the previous section, along with the NP decision rule corresponding to the resulting transformed pmf, yields a pair of NE strategies.

We recall that the expression of the NP rule, for a given general attack, is described by Lemma 1, in particular in (17)–(19). Now, given the stepwise property illustrated in Fig. 5, the transformed likelihood ratio can assume only a finite number of values, $\gamma_1 \leq \gamma_2 \leq \ldots, \leq \gamma_m$. Therefore, according to Lemma 1, the detection threshold η_τ can be chosen as the minimum of these values fulfilling (18). Let $\gamma^\star$ be such value, and let $\Sigma_{k^\star}, \ldots, \Sigma_{h^\star}$ be all the regions where the likelihood ratio is equal to $\gamma^\star$. Using again the stepwise and monotonicity properties, these regions are such that:

$$\gamma_{k^\star-1} < \gamma^\star, \qquad \gamma_{k^\star} = \ldots = \gamma_{h^\star} = \gamma^\star, \qquad \gamma^\star < \gamma_{h^\star+1}, \tag{41}$$

which implies that:

$$\begin{cases} P_{\tau^\star}(x) < \gamma^\star Q(x), & \text{for } x \in \Sigma_{1:k^\star-1}, \\ P_{\tau^\star}(x) = \gamma^\star Q(x), & \text{for } x \in \Sigma_{k^\star:h^\star}, \\ P_{\tau^\star}(x) > \gamma^\star Q(x), & \text{for } x \in \Sigma_{h^\star+1:m}, \end{cases} \tag{42}$$

where we have introduced the definition:

$$\Sigma_{k_1:k_2} \triangleq \bigcup_{k=k_1}^{k_2} \Sigma_k = \mathcal{S}_{a_{k_1}:b_{k_2}}, \tag{43}$$

for all $k_1, k_2 = 1, 2, \ldots, m$, with $k_1 \leq k_2$. It is now readily seen that the three conditions on the likelihood and the threshold summarized in (17) remap into (42). The corresponding decision rule, illustrated in the rightmost panel of Fig. 7, is:

$$\boxed{\delta^\star(x) = \begin{cases} 0, & \text{for } x \in \Sigma_{1:k^\star-1}, \\ \rho^\star, & \text{for } x \in \Sigma_{k^\star:h^\star}, \\ 1, & \text{for } x \in \Sigma_{h^\star+1:m} \end{cases}} \tag{44}$$

where a classic choice for $\rho^\star$ is [24]:

$$\rho^\star \triangleq \frac{\overline{\alpha} - \sum_{x \in \Sigma_{h^\star+1:m}} Q(x)}{\sum_{x \in \Sigma_{k^\star:h^\star}} Q(x)}. \tag{45}$$

Theorem 2 (Nash equilibrium for ADVoIP): The pair of strategies given by: the attacker's transformation described in Lemma 3 and the defender's strategy corresponding to the NP rule in (44), is a (pure-strategy) Nash equilibrium.

Proof: In order to prove the claim, it suffices to show that Lemmas 1 and 2 are simultaneously met by the attacker's and the defender's strategies mentioned in the claim of the theorem. First of all, we note that Lemma 1 is automatically satisfied by the considered strategies, because the defense in (44) is explicitly defined as the NP *given the considered attack*. Therefore, to prove equilibrium it remains to show that such attack fulfills Lemma 2 *given the considered defense*. We will now show that (22) holds true by *reductio ad absurdum*. Assume that the attacker's strategy, $\tau^\star$, does not fulfill (22). Then, for a certain $i \in \{1, \ldots, n\}$ there exists $\overline{x}$ such that:

$$\delta^\star(\overline{x}) > \delta^\star_{\min}(i) \quad \text{and} \quad \tau^\star(\overline{x}|x_i) > 0. \tag{46}$$

According to property A (closed doors), the considered attack does never transfer masses between two different regions Σ_j and Σ_k. Therefore, in order to have $\tau^\star(\overline{x}|x_i) > 0$, namely, in order to have a non-zero probability of moving x_i to $\overline{x}$, we need that both $\overline{x}$ and x_i belong to the same region Σ_k, for a certain $k = 1, \ldots, m$. But in this case, according to the definition in (44), we have $\delta^\star(\overline{x}) = \delta^\star_{\min}(i)$, in view of the monotonicity of the decision rule, which is clearly illustrated in the rightmost panel of Fig. 7. Therefore, condition (22) is never violated by the considered couple of strategies, and the claim of the theorem is met. ∎

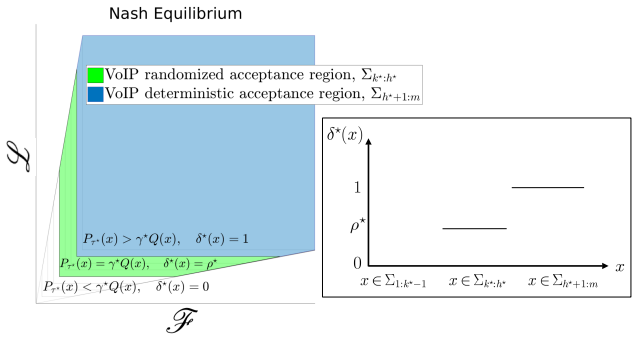

Fig. 7. Illustration of NE for the VoIP detection game (Theorem 2).

In Fig. 7, leftmost panel, we provide an illustration of the NE in the feature space. In particular, we note that the equilibrium is completely characterized by partitioning the region $\mathcal{S}_{1:n}$ into the three sub-regions $\Sigma_{1:k^\star-1}$, $\Sigma_{k^\star:h^\star}$ and $\Sigma_{h^\star+1:m}$. Within the central region, $\Sigma_{k^\star:h^\star}$, the NP decision rule $\delta^\star(x)$ is equal to $\rho^\star$ and the likelihood ratio is equal to $\gamma^\star$. Otherwise stated, when a measurement falls into this region, the defender will classify it as a VoIP with probability $\rho^\star$. Within the region $\Sigma_{h^\star+1:m}$, the decision rule is equal to 1 and, according to (42), the likelihood ratio is strictly greater than $\gamma^\star$. Therefore, when a measurement falls here the defender will always judge it as a VoIP. Finally, within the region $\Sigma_{1:k^\star-1}$, the decision rule is equal to 0 and the likelihood ratio is smaller than $\gamma^\star$. As a result, all measurements falling here are classified as non-VoIP.

REMARK II (*Practical uniqueness of Nash Equilibrium*). The players of the game choose their strategies in a *non-cooperative* way. For this reason, the existence of multiple equilibria can be a critical aspect, because it is not guaranteed that both players will decide to play for the *same* equilibrium pair. Another issue is that different equilibria might yield different costs. Luckily, these issues are not relevant in our case, because zero-sum games (as the one we consider) have

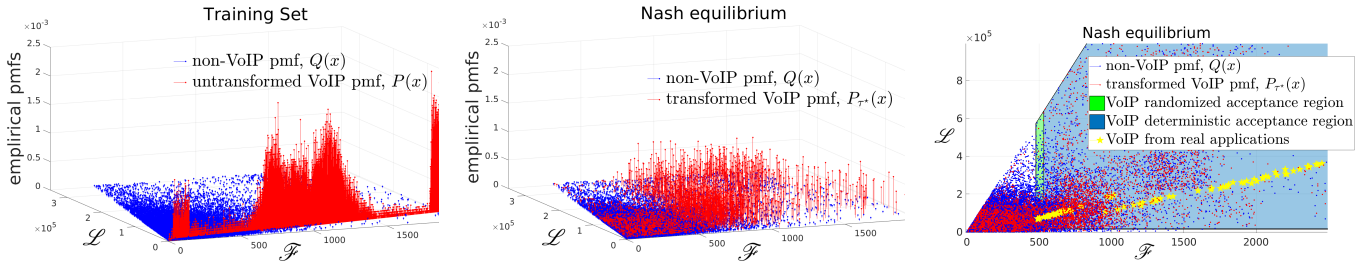

Fig. 8. **Leftmost panel**: *training* set containing non-VoIP streams and *primitive* VoIP streams, represented in the $(\mathscr{F}, \mathscr{L})$ feature space. **Middle panel**: *test* set containing non-VoIP streams, and VoIP streams *transformed according to the equilibrium attack*. **Rightmost panel**: 2D view of the middle panel, along with the VoIP acceptance regions, and with VoIP streams coming from standard applications. The VoIP acceptance regions (defender's NE strategy), as well as the VoIP transformation rule (attacker's NE strategy), have been computed over the training set using the algorithm described in Lemma 3.

the following properties. First, all equilibria yield the same costs. Second, the class of equilibrium pairs can be obtained as the Cartesian product of an ensemble A of attacker's strategies and an ensemble B of defender's strategies. This means that the pair obtained by picking any strategy from A and any strategy from B is an equilibrium [25]. In particular, this property allows the attacker and the defender to consider only the equilibrium strategies computed in the previous treatment.

REMARK III (*Costs at equilibrium*). Let us recall that property B entails a probability-mass conservation: after transformation, all the probability mass of primitive VoIP points lying in any region Σ_k must remain confined into Σ_k, for any $k = 1, \ldots, m$. This leads to the equality

$$\sum_{x \in \Sigma_k} P_{\tau^\star}(x) = \sum_{x \in \Sigma_k} P(x). \tag{47}$$

As a result, recalling the definition of $\delta^\star(x)$ in (44), the miss-detection probability at equilibrium can be written as:

$$\begin{aligned} \beta_{\tau^\star, \delta^\star} &= 1 - \sum_{x \in \mathcal{S}_{1:n}} P_{\tau^\star}(x) \delta^\star(x) \qquad (48) \\ &= 1 - \left[\rho^\star \sum_{x \in \Sigma_{k^\star : h^\star}} P(x) + \sum_{x \in \Sigma_{h^\star + 1 : m}} P(x) \right], \end{aligned}$$

which shows how the performance at equilibrium depends only upon the *original, untransformed* pmf, $P(x)$.

V. EXPERIMENTS

A. Real Traffic Capturing

We start by describing the procedure adopted for building our dataset. We collected real-world traffic streams arising from typical applications like website navigation, file downloading, streaming services, and obviously VoIP calls. Each individual traffic stream lasts one minute. In particular, part of such measurements comes from existing online repositories, such as [27] and [26], whereas another part has been collected from scratch in our laboratory. In both cases, the measurements have been organized using the *pcap* file format, which is commonly adopted by many software tools for packet capturing and inspection.

The collected streams have been labeled (VoIP vs. non-VoIP) using the information available from such software tools. However, as discussed also in the introductory section, VoIP applications implement certain operations on the primitive voice stream before transmitting it (due to, e.g., proprietary software constraints, quality and/or security requirements). As a result, a VoIP stream obtained from a particular application is a transformed versions of a primitive voice stream, and, in addition, the type of transformation may vary across different applications.

However, in order to span the set of the achievable transformations, we need to start from primitive voice streams. To this end, in our experiments we construct primitive voice streams starting from the gathered VoIP streams, through the following three-step procedure.

First, we empty all packets corresponding to the silence periods. Second, as packet size μ for a primitive stream we consider the minimum size corresponding to plain voice encoding (no overhead). Third, to set Δ, we choose the maximum packetization time found for existing VoIP applications.

B. Performance of ADVoIP

The obtained dataset is then split in two equal-size portions. The first portion is used as training set, in the following sense. Since implementation of ADVoIP requires using the pmfs $P(x)$ and $Q(x)$ (i.e., the pmf in the $(\mathscr{F}, \mathscr{L})$ plane corresponding to primitive VoIP and non-VoIP streams, respectively), we use the training set to learn these pmfs. The algorithm to find NE is then run using such empirical estimates.

In Fig. 8, leftmost panel, we report the training set arising from the aforementioned procedure. We note that, as predicted by (3), the primitive VoIP streams cover a segment in the $(\mathscr{F}, \mathscr{L})$ plane. The pmf of the primitive VoIP streams exhibits a certain symmetry, a behavior that is ascribed to the fact that, per each conversation, we have communication in two directions, with the talking periods in one direction being (almost) equal to the silence periods in the opposite direction. Moreover, we see that the non-VoIP streams are spread over a much wider region, in accordance with the huge variety of flows traveling across the network. We remark that the VoIP and non-VoIP streams occupy only a focused portion of the available region (namely, the portion that is shown in Fig. 8). This happens because, typically, the transmitter is forced to use a small portion of the maximum achievable frequency, $n\delta$. We

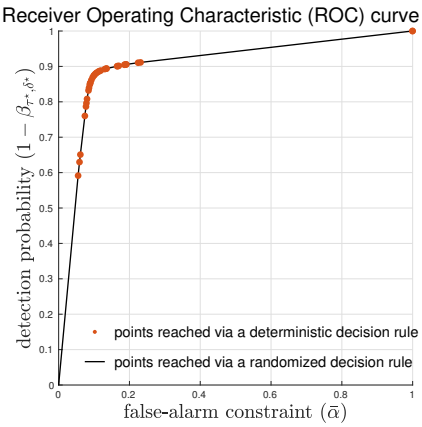

Fig. 9. ROC curve for the decision rules at equilibrium.

remark also that the proposed analysis can be applied, *mutatis mutandis*, by including a further constraint on the maximum rate in the model. On the other hand, considering the widest achievable region (as we do in this work) is a conservative choice that avoids introducing additional tuning parameters.

The second portion of the dataset is used to construct a test set. To this end, the non-VoIP streams are left unaltered, whereas the primitive VoIP streams are transformed using the equilibrium attack computed at the previous step. Finally, the detection performance is estimated over the test set.

In Fig. 8, middle panel, we report: i) the non-VoIP points in the $(\mathscr{F},\mathscr{L})$ plane corresponding to the test set (blue points); ii) the primitive VoIP points, transformed according to the NE attacker's transformation *that has been determined over the training set* (red points).

In the rightmost panel, we display the NE defender's VoIP deterministic acceptance region $\Sigma_{h^\star+1:m}$ (shaded light-blue area) and the randomized acceptance region $\Sigma_{k^\star:h^\star}$ (shaded green area) *that have been determined over the training set*.

For comparison purpose, we show also the (non-primitive) VoIP points corresponding to standard VoIP applications (yellow points). It is useful to examine these points in some detail, since we know that non-primitive VoIP streams are transformation of primitive VoIP streams, and that VoIP applications must certainly fulfill the real-time/low-latency requirements when manipulating the primitive voice streams. First, we see from the rightmost panel of Fig. 8 that all the non-primitive VoIP traces fall inside the admissible transformed region. This behavior reveals that the constraints we imposed are automatically satisfied by standard applications. Even more remarkably, almost all the *VoIP traces are correctly detected by ADVoIP*. This is perhaps not unexpected, since, in a sense, standard VoIP applications have (obviously) no malicious purposes, and, hence, it is reasonable to assume that their transformation are neither too invasive, nor optimized for hiding the traffic.

In Fig. 9 we show the Receiver Operating Characteristic (ROC) curve for the decision strategy at equilibrium. The ROC exhibits a marked knee-point, and around such knee-point, the rate of correct classification ranges from $85-90\%$ for a false-alarm rate ranging from $10-15\%$.

It is useful to remark that the computational complexity of the detection procedure resides essentially in the algorithm that must find the Nash equilibrium, i.e., that determines the boundaries of the decision region. As a result, we observe that: i) for static (or moderatley varying) training sets, this procedure is run only once, and does not constitute a problem even for online implementations; ii) even for dynamic training sets that need to be periodically updated, the cost seems affordable, because the algorithm has a time-complexity in the order of minutes for training sets of cardinality in the order of thousands traffic streams.

C. Experimental Comparison with Existing Approaches.

As far as we can tell, there are no works that consider adversarial detection of VoIP streams. For this reason, in this section we are going to consider i) existing *VoIP-specific*, but *non-adversarial* detectors, that are known to give excellent results in absence of malicious transformations, and ii) existing *adversarial*, but *not VoIP-specific* detectors, which are known to be optimal when the attacker's possibilities match some general-purpose distortion criteria. In order to avoid misunderstanding, we remark that the comparison we make is *not* intended to judge the merits of the aforementioned strategies, which are proved to be efficient for the class of problems they are conceived. More properly, the comparison illustrates why taking into account the specific forms of the transformation in the VoIP application is crucial to boost the detector's performance in the considered VoIP application.

In particular, as regards the VoIP-tailored, non-adversarial strategies, we consider the C4.5 algorithm, which is commonly employed for VoIP detection [28].

With regard to adversarial general-purpose strategies, we consider two methods. The first method is the asymptotically-optimal detector developed in [12]. The second one is an adversarial Support-Vector-Machine (SVM) developed in [29].

Let us now focus on the C4.5 algorithm. The algorithm takes advantage from a superset of the two features $(\mathscr{F},\mathscr{L})$ that are employed in our approach. As expected, the algorithm works very well (with negligible error) using only two features, that are the minimum packet length and the average inter-arrival time. However, the decision rule is easily crackable by the attack model described in Sect. IV, which can heavily alter these two features and attain a complete evasion. A sample of this performance is reported in Fig. 10.

We now move on to examine the detector developed in [13]. In a nutshell, this detector takes as input a source of data (in our setting, the traffic stream), assumes that such source is ergodic, and takes its (marginal) empirical pmf as a fundamental statistical descriptor. We remark here that the empirical pmfs under consideration are not the pmfs $P(x)$ and $Q(x)$, for $x\in(\mathscr{F},\mathscr{L})$, which have been considered in the previous discussion. These latter are in fact obtained by computing the relative frequency of occurrence, *across several realizations of traffic streams*, of a particular point x in the feature space. In contrast, following an ergodic assumption, the empirical pmf considered in [13] is the relative frequency of a particular source value (a packet length in our setting) within a single traffic stream, and is computed *across time*.

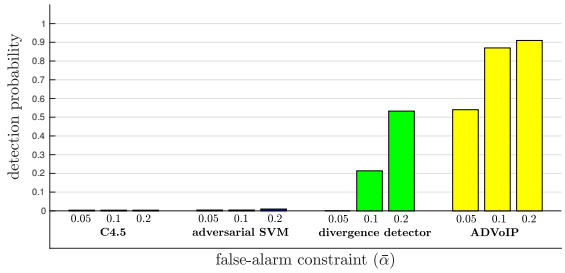

Fig. 10. Bar plot comparing the performance of ADVoIP against the performance of the different strategies considered in Sect. V-C.

The detection game formulated in [13] is solved exactly through an adversarial generalized likelihood ratio test, which relies on the evaluation of Kullback-Leibler divergences between suitably defined sequences taken from the test set and/or the training set. For the sake of brevity, we shall refer to this detector as *divergence detector*.

We have implemented the divergence detector on our dataset, and the resulting performance is shown in Fig. 10, where we can see that, as expected, ADVoIP exhibits a better performance. This behavior comes perhaps as no surprise, since, as we have already anticipated, comparing our system to general-purpose, even though adversarial, detectors might be not fair. As a matter of fact, the divergence detector, applied to the VoIP setting, is penalized in at least two ways. First of all, it is penalized when the observed stream is a primitive VoIP. In this case, the divergence detector is impaired if the primitive stream is manipulated in such a way that the empirical pmf of the transformed stream is close to the pmf of the non-VoIP streams, which affects adversely the detection probability. Second, the divergence detector is penalized when the observed stream is non-VoIP. In this case, the empirical pmf of a non-VoIP stream (e.g., an http flow) is compared against the pmf computed over the aggregate of different non-VoIP applications. As a result, empirical pmf of such single non-VoIP stream would seldom converge to the aggregate pmf, which affects adversely the false alarm probability. We see therefore that, as already mentioned in the introduction, the lack of ergodicity is a critical aspect.

We finally consider the adversarial SVM technique described in [29]. As for the C4.5 algorithm, we train the algorithm with a superset of our two features $(\mathscr{F}, \mathscr{L})$.

We now remark three limitations of the considered adversarial SVM for our VoIP setting. First of all, many features employed in [29] (e.g., the average packet length) are global descriptors, since they do not take into account the local real-time constraints. Second, the attack model used in [29] does not match the specific VoIP scenario. In particular, it is assumed that the attacker can "perturb" the feature vector of a VoIP stream along any direction of the feature space, namely, he can increase and/or decrease the features *independently* one from each other, up to some given boundaries. However, some features relevant for VoIP cannot be transformed independently. For instance, adding a new packet affects both the length and the inter-arrival statistics. Considering attack models that are not applicable to the VoIP domain may have a huge impact on the classifier performance, because resources are wasted to combat attacks that are not feasible. We note also that the linear decision model adopted by the SVM is perhaps not the best one for the problem at hand, due to the non-perfect linear separability of the sample data. This notwithstanding, the SVM approach might in principle perform better than our detector, since it takes advantage of a richer set of features. In order to show that this is not the case, we have conducted an experimental test, whose results are summarized in Fig. 10. From the detection probabilities reported in Fig. 10 we observe that the performance of the adversarial SVM is significantly worse than the performance of ADVoIP. The observed gap in performance should be mostly ascribed to the fact that the available adversarial SVMs are not designed to capture the peculiarities of the addressed VoIP identification problem.

VI. Conclusions

Voice calls over telecommunications media are undergoing a significant shift of paradigm. Thanks to the proliferation of VoIP and related applications, the usage of network protocols for sustaining reliable calls is now becoming more and more popular, with a trend that is expected to increase in the next few years. While VoIP applications offer new opportunities for ubiquitous communications, they concurrently give rise to new challenges in terms of cyber-security. One notable instance of these challenges is the clandestine propagation of VoIP calls, which is of interest for several illegal/criminal scopes. In this domain, VoIP packets are manipulated, e.g., through intermediate low-latency networks, so as to grant at the same time anonymity and satisfaction of real-time constraints.

The practical applications where these issues arise can be summarized in the following model: there are two parties that want to communicate through a VoIP application, trying not to be identified by a third-party monitoring agency. To this aim, the sender performs suitable transformations on the packet before transmitting it, in such a way that the traffic analyst can confuse the monitored stream with some other type of traffic. The role of the third party is to devise a proper countermeasure (detection rule) to decide: is the observed stream VoIP or not? The combined interaction between the two parties and the third-party agency gives rise to a classic adversarial framework, where an attacker (the two communicating parties), and a defender (the agency) play the same game, each one trying to maximize their respective utility.

In this work, we have conducted a detailed analysis of the aforementioned model, providing the following contributions.
— We introduced a class of attacker's transformations that apply padding, shifting and splitting operations on the packet, guaranteeing that the call is reconstructed by the receiver *within tolerable real-time/low-latency margins*. The transformation is used to hide the VoIP traffic stream amidst the general traffic stream flowing all across the network.
— Our analysis shows how the attacker should design the optimal transformation, and how the defender should design the optimal detector to discriminate VoIP from non-VoIP streams. In both cases, the adjective "optimal" is intended in a game-theoretic sense.

— The result of the latter two points is a *Nash equilibrium* solution of the game: once that the two players choose their own best strategies, *no one of them has convenience in deviating unilaterally from the selected strategy*.
— We tested the usefulness of the method over real traffic traces, and we provided a comparison with existing VoIP detection methods. In particular, we showed that the strategies available in the literature are either i) not designed to cope with the presence of a malicious attacker; or ii) consider attacker's transformations that do not match the VoIP paradigm.

The proposed strategy offers remarkable performance gains w.r.t. existing methods. Moreover, the adversarial formulation leads to a solution that, besides being optimal from a theoretical standpoint, has the following practical implications. The defender does not assume a general attack model but, coping with the classic adversarial framework, he assumes that a *rational* attacker has always convenience in playing at equilibrium. Accordingly, in the presence of a rational attacker, the defender's rule provides *the best response* yielding the minimum defender's cost. What if the attacker were not acting rationally? Since we deal with a zero-sum game, the choice of the defender would be conservative, because the defender's cost can be only reduced by a wrong attacker's choice.

Finally, for the conducted theoretical study to be efficiently declined in practical contexts, several additional analyses can be useful, including: validation over larger datasets; classification with enlarged set of features; testing over VoIP streams subject to specific manipulations that have been documented in real-world applications.

ACKNOWLEDGMENT

The authors express their deep gratitude to Prof. Maurizio Longo for the stimulating discussions and for the endless encouragement offered during the writing of this work.

APPENDIX A
PROOF OF THEOREM 1

For later use, it is expedient to introduce the following quantities:

$$X \triangleq \sum_{f_i(\boldsymbol{v})=1} \min(f_i-\mu,0), \quad Y \triangleq \sum_{f_i(\boldsymbol{v})=1} \max(f_i,\mu), \tag{49}$$

$$W \triangleq \sum_{f_i(\boldsymbol{v})=0} \max(f_i-1,0), \quad Z \triangleq \sum_{f_i(\boldsymbol{v})=0} \min(f_i,1). \tag{50}$$

Since, for any $x, y \in \mathbb{R}$, we have that

$$\min(x,y)+\max(x-y,0)=x, \tag{51}$$

using the definition of $\mathscr{F}$ in (1), it is readily seen that:

$$\boxed{\mathscr{F} = X+Y+Z+W} \tag{52}$$

We start by computing the boundaries of the achievable region. First, by applying (9) to (1), we get:

$$F(\boldsymbol{v}) = \sum_{i=1}^{n} f_i(\boldsymbol{v}) \le \mathscr{F} \le \sum_{i=1}^{n} \Delta = n\Delta, \tag{53}$$

which corresponds to (11). Likewise, by observing that $\mathscr{L} = \sum_{f_i(\boldsymbol{v})=1} \ell_i + \sum_{f_i(\boldsymbol{v})=0} \ell_i$ and by using (10), we get

$$\begin{aligned} \mathscr{L} &\ge \sum_{f_i(\boldsymbol{v})=1} \max(f_i,\mu) + \sum_{f_i(\boldsymbol{v})=0} f_i \\ &= Y+Z+W \end{aligned} \tag{54}$$

and

$$\begin{aligned} \mathscr{L} &\le \sum_{f_i(\boldsymbol{v})=1} \ell_{\max} + \sum_{f_i(\boldsymbol{v})=0} \min(f_i,1)\,\ell_{\max} \\ &= F(\boldsymbol{v})\,\ell_{\max} + Z\ell_{\max}, \end{aligned} \tag{55}$$

or:

$$Y+Z+W \le \mathscr{L} \le (F(\boldsymbol{v})+Z)\ \ell_{\max}. \tag{56}$$

In order to find the bounds of $\mathscr{L}$, we proceed by computing the lower bound of the quantity $Y+Z+W$, and the upper bound of the quantity Z, corresponding to a given $\mathscr{F}$.

Before proceeding, we recall that:

$$\zeta \triangleq \min(\Delta,\mu), \quad \mu \ge 1, \tag{57}$$

and that:

$$F(\boldsymbol{v}) = \sum_{i=1}^{n} f_i(\boldsymbol{v}) \Rightarrow \sum_{f_i(\boldsymbol{v})=0} 1 = n - F(\boldsymbol{v}). \tag{58}$$

Using (57) and (58), in view of (9) we can write:

$$\begin{aligned} (1-\mu)F(\boldsymbol{v}) \le\ & X \le (\zeta-\mu)F(\boldsymbol{v}), \\ \mu F(\boldsymbol{v}) \le\ & Y \le [\Delta-(\zeta-\mu)]F(\boldsymbol{v}), \\ 0 \le\ & Z \le n-F(\boldsymbol{v}), \\ 0 \le\ & W \le [n-F(\boldsymbol{v})](\Delta-1). \end{aligned} \tag{59}$$

Using now (52), we can conclude that:[4]

$$\begin{aligned} Y+Z+W &\ge \max\{\mu F(\boldsymbol{v}),\ \mathscr{F}-(\zeta-\mu)\,F(\boldsymbol{v})\} \\ &= \mu F(\boldsymbol{v}) + \max\{0, \quad \mathscr{F}-\zeta F(\boldsymbol{v})\}, \end{aligned} \tag{60}$$

and that:

$$\begin{aligned} Z &\le \min(n-F(\boldsymbol{v}), \quad \mathscr{F}-(1-\mu)F(\boldsymbol{v}))-\mu F(\boldsymbol{v}))) \\ &= \min(n-F(\boldsymbol{v}), \quad \mathscr{F}-F(\boldsymbol{v})). \end{aligned} \tag{61}$$

Equations (60) and (61) can be rewritten in a more convenient form as, respectively:

$$Y+Z+W \ge \begin{cases} \mu F(\boldsymbol{v}), & \text{if } \mathscr{F} \le \zeta F(\boldsymbol{v}) \\ \mathscr{F}+(\mu-\zeta)F(\boldsymbol{v}), & \text{if } \mathscr{F} > \zeta F(\boldsymbol{v}) \end{cases} \tag{62}$$

and:

$$Z \le \begin{cases} \mathscr{F}-F(\boldsymbol{v}), & \text{if } \mathscr{F} \le n \\ n-F(\boldsymbol{v}), & \text{if } \mathscr{F} > n \end{cases} \tag{63}$$

The boundary relations for $\mathscr{L}$ provided in (12) and (13), are now obtained by combining (62) and (63) with (56).

It remains to prove that, under assumption (14) any point $(\mathscr{F},\mathscr{L})$ internal to the aforementioned boundaries can be reached from the point $(F(\boldsymbol{v}),L(\boldsymbol{v}))$, by making a suitable

[4]Given $u_{\min} \le u \le u_{\max}$, $v_{\min} \le v \le v_{\max}$, and $u+v=a$, then we have: $\max(u_{\min}, a-v_{\max}) \le u \le \min(u_{\max}, a-v_{\min})$. In (60) we consider $Y+Z+W$ in place of u and X in place of v. In (61) we consider Z in place of u and $X+Y+W$ in place of v.

assignment of the f_i's and ℓ_i's. The difficulties related to this part of the proof arise from the following facts: a generic point can be attained with more than one assignment and different points can be attained by different assignments. For this reason, it is convenient to split the analysis into distinct cases, and to consider a particular attack for each separate case. In particular, we will partition the whole region

$$F(\boldsymbol{v}) \leq \mathscr{F} \leq n\Delta, \qquad \mathscr{L}_{\min}(\mathscr{F}) \leq \mathscr{L} \leq \mathscr{L}_{\max}(\mathscr{F})$$

into the four sub-regions examined next (Cases 1–4), as exemplified in Fig. 11.

— *Case* 1. Firstly, we prove the claim for the sub-region:

$$F(\boldsymbol{v}) \leq \mathscr{F} \leq \zeta F(\boldsymbol{v}), \qquad \mathscr{L}_{\min}(\mathscr{F}) \leq \mathscr{L} \leq F(\boldsymbol{v})\ell_{\max}. \tag{64}$$

Let us consider the set of all the attacks assigning the f_i's with the following criterion:

$$f_i(\boldsymbol{v}) \leq f_i \leq \zeta \;\text{ if } f_i(\boldsymbol{v}) = 1, \qquad f_i = 0 \;\text{ if } f_i(\boldsymbol{v}) = 0. \tag{65}$$

Otherwise stated, the attacker can set the f_i's *independently* one from each other, by picking their values within the ranges specified in (65). Then, after having chosen the value for the generic f_i, he can set, again *independently* from the others, the related ℓ_i to any value within the range given by (9), which in this case becomes (we recall that $\zeta \leq \mu$ by definition):

$$\mu \leq \ell_i \leq \ell_{\max} \;\text{ if } f_i(\boldsymbol{v}) = 1, \qquad \ell_i = 0 \;\text{ if } f_i(\boldsymbol{v}) = 0. \tag{66}$$

Using now the definition of $\mathscr{F}$ in (1), we see that the aforementioned set of attacks spans any value of $\mathscr{F}$ within the range $F(\boldsymbol{v}) \leq \mathscr{F} \leq \zeta F(\boldsymbol{v})$ of the pertinent sub-region in (64). In fact, the lower bound $\mathscr{F} = F(\boldsymbol{v})$ can be attained by imposing $f_i = 1$ for all the f_i's where $f_i(\boldsymbol{v}) = 1$. Moreover, by incrementing only one of these f_i's at a time by 1, until all the f_i's reach the maximum allowed value ζ, we obtain $\mathscr{F} = F(\boldsymbol{v}) + 1, \mathscr{F} = F(\boldsymbol{v}) + 2, \ldots, \mathscr{F} = \zeta F(\boldsymbol{v})$.

Likewise, using the definitions of X, Y, Z, W in terms of the f_i's, and since $\mathscr{F} = X + Y + Z + W$, from (65) we get:

$$X = \mathscr{F} - \mu F(\boldsymbol{v}), \quad Y = \mu F(\boldsymbol{v}), \quad Z = W = 0, \tag{67}$$

and therefore, for any $\mathscr{F}$ within the sub-region in (64), the boundary relation in (56) becomes:

$$\mu F(\boldsymbol{v}) \leq \mathscr{L} \leq \ell_{\max} F(\boldsymbol{v}). \tag{68}$$

We now show that the value $\mathscr{L}$ spans the whole range in (68). In fact, the lower bound $\mathscr{L} = \mu F(\boldsymbol{v})$ can be attained by imposing $\ell_i = \mu$ for all the ℓ_i's where $f_i(\boldsymbol{v}) = 1$. Then, by incrementing only one of these ℓ_i's at a time by 1, until all the ℓ_i's reach the value $\ell_{\max}$, we obtain $\mathscr{L} = \mu F(\boldsymbol{v}) + 1, \mathscr{L} = \mu F(\boldsymbol{v}) + 2, \ldots, \mathscr{L} = \ell_{\max} F(\boldsymbol{v})$. Finally, since if $\mathscr{F} \leq \zeta F(\boldsymbol{v})$ then $\mathscr{L}_{\min}(\mathscr{F}) = \mu F(\boldsymbol{v})$, the range in (68) is the same as in (64), and the proof for the first sub-region is complete.

— *Case* 2. We now focus on the sub-region:

$$\zeta F(\boldsymbol{v}) \leq \mathscr{F} \leq n\Delta, \qquad \mathscr{L}_{\min}(\mathscr{F}) \leq \mathscr{L} \leq F(\boldsymbol{v})\ell_{\max}. \tag{69}$$

Let us now consider the set of all the attacks assigning the f_i's with the following criterion:

$$\zeta \leq f_i \leq \Delta \;\text{ if } f_i(\boldsymbol{v}) = 1, \qquad 0 \leq f_i \leq \Delta \;\text{ if } f_i(\boldsymbol{v}) = 0. \tag{70}$$

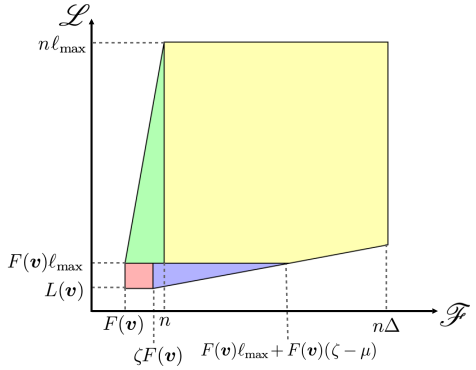

Fig. 11. Illustration of the four sub-regions used in the proof of Theorem 1.

The rule for choosing the ℓ_i's is given by (66). As done in the previous case, using this set of attacks it is possible to span the entire range of $\mathscr{F}$ within the sub-region in (69). However, for this sub-region we need to make an important remark. For some values of $\mathscr{F}$ within the range in (69), the condition

$$\mathscr{L}_{\min}(\mathscr{F}) > F(\boldsymbol{v})\ell_{\max} \tag{71}$$

can be met. In particular, making explicit the definition of $\mathscr{L}_{\min}(\mathscr{F})$ in (12), Eq. (71) reduces to:

$$\mathscr{F} > F(\boldsymbol{v})\ell_{\max} + F(\boldsymbol{v})(\zeta - \mu). \tag{72}$$

Inspecting the range of $\mathscr{L}$ appearing in (69), Eq. (71) reveals that, for the values of $\mathscr{F}$ in (72), the second sub-region of interest is in fact limited to the smallest range:

$$\zeta F(\boldsymbol{v}) \leq \mathscr{F} \leq \min\{F(\boldsymbol{v})\ell_{\max} + F(\boldsymbol{v})(\zeta - \mu),\ n\Delta\}, \tag{73}$$

which is obtained by removing the values of $\mathscr{F}$ in (72) from the range in (69) — see Fig. 11 for a graphical illustration.

Moreover, from (70) we see that:

$$X = (\zeta - \mu)F(\boldsymbol{v}), \qquad Y + Z + W = \mathscr{F} - (\zeta - \mu)F(\boldsymbol{v}), \tag{74}$$

and therefore, for any $\mathscr{F}$ ranging in (73), Eq. (56) becomes:

$$\mathscr{L}_{\min}(\mathscr{F}) = \mathscr{F} - (\zeta - \mu)F(\boldsymbol{v}) \leq \mathscr{L} \leq (F(\boldsymbol{v}) + Z)\ell_{\max} \tag{75}$$

Since from (59) we have $Z \geq 0$, we conclude that $(F(\boldsymbol{v}) + Z)\ell_{\max} \geq F(\boldsymbol{v})\ell_{\max}$. This means that the set of attacks proposed in (77) permits to span a superset of the considered sub-region, that in effect proves the claim.

The next two cases are handled reasoning as done in the previous cases, in particular:

— *Case* 3. The third sub-region,

$$F(\boldsymbol{v}) \leq \mathscr{F} \leq n, \qquad F(\boldsymbol{v})\ell_{\max} \leq \mathscr{L} \leq \mathscr{L}_{\max}(\mathscr{F}) \tag{76}$$

can be covered by the set of attacks such that:

$$f_i = 1 \;\text{ if } f_i(\boldsymbol{v}) = 1, \qquad 0 \leq f_i \leq 1 \;\text{ if } f_i(\boldsymbol{v}) = 0, \tag{77}$$

where the rule for choosing the ℓ_i's is given by (66).

— *Case* 4. The fourth sub-region,

$$n \leq \mathscr{F} \leq n\Delta, \qquad F(\boldsymbol{v})\ell_{\max} \leq \mathscr{L} \leq \mathscr{L}_{\max}(\mathscr{F}), \tag{78}$$

can be covered by the set of attacks such that $1 \leq f_i \leq \Delta$, where the rule for choosing the ℓ_i's is given by (66). ■

APPENDIX B
PROOF OF LEMMA 2

Assume that τ' is an attacker's strategy that does not fulfill (22). Then, for a certain $i \in \{1,\ldots,n\}$, there exists a point $\overline{x} \in \mathcal{S}_{i:n}$ such that

$$\delta(\overline{x}) > \delta_{\min}(i) \quad \text{and} \quad \tau(\overline{x}|x_i) > 0, \tag{79}$$

where $\delta_{\min}(i)$ is defined in (20). Let also $x_{\min} \in \mathcal{S}_{i:n}^{(\min)}$, where $\mathcal{S}_{i:n}^{(\min)}$ is defined in (21). Now, from (15) and (16) we have:

$$\begin{aligned}\beta_{\tau',\delta} &= 1 - \sum_{x\in\mathcal{S}_{1:n}} \sum_{j=1}^{n} \delta(x)\tau'(x|x_j)P(x_j) \\ &= 1 - [\delta(x_{\min})\tau'(x_{\min}|x_i) + \delta(\overline{x})\tau'(\overline{x}|x_i)]P(x_i) \\ &+ \text{extra terms.}\end{aligned} \tag{80}$$

The second equality follows by isolating from the two summations the terms corresponding to $j = i$, and to $x = \overline{x}$ and $x = x_{\min}$. We now show that the attacker can always increase the miss-detection error by a transformation τ'' that obeys:

$$\tau''(\overline{x}|x_i) = 0, \qquad \tau''(x_{\min}|x_i) = \tau'(\overline{x}|x_i) + \tau'(x_{\min}|x_i), \tag{81}$$

and is elsewhere equal to τ'. In practice, the "mass" that τ' transports from x_i to $\overline{x}$, is now all transported to $x_{\min}$ by τ''. The corresponding miss-detection probability is:

$$\begin{aligned}\beta_{\tau'',\delta} &= 1 - [\delta(x_{\min})\tau''(x_{\min}|x_i) + \delta(\overline{x})\tau''(\overline{x}|x_i)]P(x_i) \\ &+ \text{same extra terms as in (80),}\end{aligned}$$

which, using (81), becomes:

$$\begin{aligned}\beta_{\tau'',\delta} &= 1 - [\delta(x_{\min})\tau'(x_{\min}|x_i) + \delta(x_{\min})\tau'(\overline{x}|x_i)]P(x_i) \\ &+ \text{same extra terms as in (80).}\end{aligned} \tag{82}$$

Then, the miss-detection probability in (82) differs from the one in (80) only in the term $\delta(x_{\min})\tau'(\overline{x}|x_i)$ that replaces the term $\delta(\overline{x})\tau'(\overline{x}|x_i)$. Now, since from the first relationship in (79) we have $\delta(\overline{x}) > \delta(x_{\min})$, it is immediately seen that the miss-detection probability in (82) is (strictly) greater than the miss-detection probability in (80), implying that the transformation τ' is *not* optimal. ■

REFERENCES

[1] Y. Zeng, H. Gu, W. Wei, and Y. Guo, "*Deep-Full-Range*: A deep learning based network encrypted traffic classification and intrusion detection framework," *IEEE Access*, vol. 7, pp. 45182–45190, 2019.

[2] M. E. Ahmed, S. Ullah, and H. Kim, "Statistical application fingerprinting for DDoS attack mitigation," *IEEE Trans. Inf. Forensics and Security*, vol. 14, no. 6, pp. 1471–1484, Jun. 2019.

[3] T. T. Nguyen, G. Armitage, P. Branch, and S. Zander, "Timely and continuous machine-learning-based classification for interactive IP traffic," *IEEE/ACM Trans. Netw.*, vol. 20, no. 6, pp. 1880–1894, 2012.

[4] J. Zhang, X. Chen, Y. Xiang, W. Zhou, and J. Wu, "Robust network traffic classification," *IEEE/ACM Trans. Netw.*, vol. 23, no. 4, pp. 1257–1270, 2015.

[5] M. Di Mauro and M. Longo, "Skype traffic detection: a decision theory based tool," *IEEE ICCST*, Rome, 2014, pp. 1–6.

[6] S. Chen, X. Wang, and S. Jajodia, "On the anonymity and traceability of peer-to-peer VoIP calls," *IEEE Network*, vol. 20, no. 5, pp. 32–37, 2006.

[7] S. Shintre, V. Gligor, and J. Barros, "Anonymity leakage in private VoIP networks," *IEEE Trans. Depend. Sec. Comput.*, vol. 15, no. 1, pp. 14–26, Jan.-Feb. 2018.

[8] T. Alpcan and T. Başar, *Network Security. A Decision and Game-Theoretic Approach*, 1st ed. Cambridge University Press, 2011.

[9] J. Von Neumann, "Zur theorie der gesellschaftsspiele," *Mathematische Annalen*, vol. 100, pp. 295–320, 1928.

[10] J. Nash, "Equilibrium points in n-person games," *Proceedings of the National Academy of Sciences*, vol. 36, no. 1, pp. 48–49, 1950.

[11] J. Von Neumann and O. Morgenstern, *Theory of Games and Economic Behavior*, Princeton University Press, 2007.

[12] M. Barni and B. Tondi, "The source identification game: an information theoretic perspective," *IEEE Trans. Inf. Forensics and Security*, vol. 8, no. 3, pp. 450–463, Mar. 2013.

[13] M. Barni and B. Tondi, "Binary hypothesis testing game with training data," *IEEE Trans. Inf. Theory*, vol. 60, no. 8, pp. 4848–4866, Aug. 2014.

[14] M. Barni and B. Tondi, "Adversarial source identification game with corrupted training," *IEEE Trans. Inf. Theory*, vol. 64, no. 5, pp. 3894–3915, May 2018.

[15] P. McDaniel, N. Papernot, and Z. B. Celik, "Machine learning in adversarial settings," *IEEE Security Privacy*, vol. 14, no. 3, pp. 68–72, 2016.

[16] S. Rota Bulò, B. Biggio, I. Pillai, M. Pelillo and F. Roli, "Randomized prediction games for adversarial machine learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. PP, no. 99, pp. 1–13, 2016.

[17] Q. Wang and M. Liu, "Learning in Hide-and-Seek," *IEEE/ACM Trans. Netw.*, vol. 24, no. 2, pp. 1279–1292, 2016.

[18] B. Tondi, M. Barni and N. Merhav, "Detection games with a fully active attacker," *IEEE WIFS*, Rome, 2015, pp. 1–6.

[19] M. Barni and B. Tondi, "Source Distinguishability Under Distortion-Limited Attack: An Optimal Transport Perspective," *IEEE Trans. Inf. Forensics and Security*, vol. 11, no. 10, pp. 2145–2159, Oct. 2016.

[20] S. Marano, V. Matta, and L. Tong, "Distributed detection in the presence of Byzantine attacks," *IEEE Trans. Signal Process.*, vol. 57, no. 1, pp. 16–29, Jan. 2009.

[21] Cisco Technical Report, "Voice Over IP - Per call bandwidth consumption,", Document ID:7934, 2016.

[22] V. Joseph and B. Chapman, *Deploying QoS for Cisco IP and Next Generation Networks: The Definitive Guide*, 1st ed. Elsevier Science, 2009.

[23] ITU-T, Recommendation G.114, General Recommendations on the transmission quality for an entire international telephone connection, International Telecommunication Union, 2003.

[24] E. L. Lehmann and J. P. Romano, *Testing Statistical Hypotheses*, Springer, 2005.

[25] F. Vega-Redondo, *Economics and the Theory of Games*, Cambridge University Press, 2003.

[26] NETRESEC's public repository of network traffic traces. Available at: https://www.netresec.com/?page=PcapFiles

[27] Tstat's public repository of network traffic traces. Available at: http://tstat.polito.it/traces.shtml

[28] R. Alshammari and A. N. Zincir-Heywood, "An investigation on the identification of VoIP traffic: Case study on Gtalk and Skype,", in *2010 International Conference on Network and Service Management*, (Niagara Falls, ON), pp. 310–313, 2010.

[29] Y. Zhou, M. Kantarcioglu, B. Thuraisingham, and B. Xi, "Adversarial support vector machine learning," in *Proc. of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (New York, NY), pp. 1059–1067, 2012.