DETECTABILITY OF DENIAL-OF-SERVICE ATTACKS ON COMMUNICATION SYSTEMS

Holger Boche[*]*, Rafael F. Schaefer*[†]*, and H. Vincent Poor*[‡]

[*]Theoretische Informationstechnik
Technische Universität München
München, Germany
boche@tum.de

[†]Information Theory and Applications
Technische Universität Berlin
Berlin, Germany
rafael.schaefer@tu-berlin.de

[‡]Dept. Electrical Engineering
Princeton University
Princeton, USA
poor@princeton.edu

ABSTRACT

Wireless communication systems are inherently vulnerable to adversarial attacks since malevolent jammers might jam and disrupt the legitimate transmission intentionally. Accordingly it is of crucial interest for the legitimate users to detect such adversarial attacks. This paper develops a detection framework based on Turing machines and studies the detectability of adversarial attacks. Of particular interest are so-called denial-of-service attacks in which the jammer is able to completely prevent any transmission. It is shown that there exists no Turing machine which can detect such an attack and consequently there is no algorithm that can decide whether or not such a denial-of-service attack takes place, even if there are no limitations on computational complexity and computing capacity of the hardware.

Index Terms— Communication system, adversarial attack, Turing computability, Entscheidungsproblem

1. INTRODUCTION

Digitalization has been identified as a disruptive technology to change everyone's lives. Information processing including transmission and storing of information and data is one of the key enablers and accordingly, there is the need to study information processing from a fundamental point of view. This includes privacy of the users, guaranteed secrecy requirements on the data, security against adversarial attacks, efficiency of communication systems to ensure that scarce resources are not wasted, and many other issues. Such requirements for example will be imposed by the Tactile Internet [1]. For the latter, there are currently standardization efforts [2] as this is expected to be a key enabler for future systems beyond the fifth generation (5G) mobile networks, particularly 6G.

Reliable communication between legitimate users is the indispensable basis for information processing. And particularly wireless communication systems are inherently vulnerable to adversarial attacks since malevolent jammers might jam and harm the legitimate communication intentionally. Communication under adversarial attacks has been studied under various aspects, cf. the overview papers [3] and [4]. This paper considers the simplest communication scenario in which adversarial attacks can happen. It consists of one transmitter Alice, one receiver Bob, and one Jammer. The aim of Alice is to transmit a message reliably to Bob, while the intention of the Jammer is to disrupt this transmission as much as possible. In the worst case, the Jammer is able to perform a denial-of-service attack which means that no communication is possible at all.

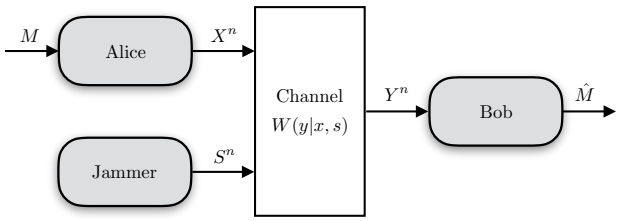

Fig. 1. Communication system with a Jammer, who tries to disrupt the communication by transmitting an own jamming sequence s^n.

Accordingly, it is of utmost importance for the legitimate users to detect such adversarial attacks, in particular denial-of-service attacks. However, to date, it is not clear how to effectively decide whether or not such adversarial attacks take place. This has drawn surprisingly little attention and to address this issue, we use the concept of a *Turing machine* [5–7]. This is a mathematical model of an abstract machine that manipulates symbols on a strip of tape according to certain given rules. It can simulate any given algorithm and therewith provides a simple but very powerful model of computation. Turing machines have no limitations on computational complexity, unlimited computing capacity and storage, and execute programs completely error-free. They are further equivalent to the von Neumann-architecture without hardware limitations and the theory of recursive functions, cf. also [8–12]. Accordingly Turing machines provide fundamental performance limits for today's digital computers and they are the ideal concept to decide whether or not such a verification task is possible at all. With the latter we mean that we are interested in understanding whether this task can in principle be solved algorithmically (without putting any constraints on the computational complexity of such an algorithm). In particular, we show that there exists no Turing machine that can detect a denial-of-service attack. Accordingly, there is no algorithm that can decide whether or not such an attack takes place.[1]

2. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we introduce the model of our communication system as shown in Fig. 1. It consists of a transmitter Alice, a receiver Bob, and a Jammer, who tries to disrupt the transmission.

The work of H. Boche and R. F. Schaefer was supported by the Gottfried Wilhelm Leibniz program of the German Research Foundation (DFG) under Grant BO 1734/20-1. The work of H. V. Poor was supported by the U.S. National Science Foundation under Grants CCF-093970, CCF-1513915, and CNS-1702808. The research direction of Turing computability for communication systems was initiated by the German Research Foundation (DFG) under Grants BO 1734/24-3 and BO 1734/25-1.

[1] *Notation:* Discrete random variables are denoted by capital letters and their realizations and ranges by lower case and script letters, respectively; $\mathbb{N}$, $\mathbb{R}$, and $\mathbb{R}_c$ are the sets of non-negative integers, real numbers, and computable real numbers; $\mathcal{P}(\mathcal{X})$ denotes the set of all probability distributions on $\mathcal{X}$.

Let $\mathcal{X}$ and $\mathcal{Y}$ be finite input and output alphabets and $\mathcal{S}$ a finite state (jamming) alphabet of the Jammer. Then the channel between Alice and Bob is given by a stochastic matrix $W : \mathcal{X} \times \mathcal{S} \to \mathcal{P}(\mathcal{Y})$ which we interchangeably also write as $W \in \mathcal{CH}(\mathcal{X}, \mathcal{S}; \mathcal{Y})$. For a fixed jamming sequence $s^n \in \mathcal{S}^n$, the discrete memoryless channel is given by $W^n_{s^n}(y^n|x^n) = W^n(y^n|x^n, s^n) = \prod_{i=1}^n W(y_1|x_i, s_i)$ for all input and output sequences $x^n \in \mathcal{X}^n$ and $y^n \in \mathcal{Y}^n$.

Definition 1. *The* arbitrarily varying channel (AVC) $\mathfrak{W}$ *is given by*

$$\mathfrak{W} = \big\{W(\cdot|\cdot, s)\big\}_{s\in\mathcal{S}}.$$

For characterizing the capacity of an AVC, we need the definition of an averaged channel. For any probability distribution $q \in \mathcal{P}(\mathcal{S})$, the *averaged channel* is given by

$$W_q(y|x) = \sum_{s\in\mathcal{S}} W(y|x, s)q(s)$$

for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. We further need the concept of *symmetrizability* [13] which describes the ability of the Jammer to "simulate" a valid channel input making it impossible for the receiver to decide on the correct codeword sent by the transmitter. This concept can be generalized as e.g. in [14] by introducing the function

$$F(\mathfrak{W}) = \min_{U\in\mathcal{CH}(\mathcal{X};\mathcal{S})} \max_{x\neq\hat{x}} \sum_{y\in\mathcal{Y}} \Big| \sum_{s\in\mathcal{S}} W(y|\hat{x}, s)U(s|x) - \sum_{s\in\mathcal{S}} W(y|x, s)U(s|\hat{x})\Big|. \tag{1}$$

In particular we observe that the AVC $\mathfrak{W}$ is symmetrizable if and only if $F(\mathfrak{W}) = 0$.

With

$$\max_{p\in\mathcal{P}(\mathcal{X})} \min_{q\in\mathcal{P}(\mathcal{S})} I(p, W_q) = \min_{q\in\mathcal{P}(\mathcal{S})} \max_{p\in\mathcal{P}(\mathcal{X})} I(p, W_q) = \min_{q\in\mathcal{P}(\mathcal{S})} C(W_q)$$

the capacity of an AVC is given as follows.

Theorem 1 ([13, 15])**.** *The capacity $C(\mathfrak{W})$ of an AVC $\mathfrak{W}$ is*

$$C(\mathfrak{W}) = \begin{cases} \min_{q\in\mathcal{P}(\mathcal{S})} C(W_q) & \text{if } F(\mathfrak{W}) > 0 \\ 0 & \text{if } F(\mathfrak{W}) = 0. \end{cases}$$

Intuitively, one would expect that the Jammer chooses the jamming sequence that minimizes $\min_{q\in\mathcal{P}(\mathcal{S})} C(W_q) = C(W_{\hat{q}})$, i.e., it chooses the jamming sequence that results in the worst channel corresponding to $\hat{q} \in \mathcal{P}(\mathcal{S})$. However, Theorem 1 reveals that even if the entropic quantity $C(W_{\hat{q}}) > 0$ is positive, a denial-of-service attack is possible in which case no communication is possible at all. This is exactly the case when $F(\mathfrak{W}) = 0$. In the sequel, we are interested in studying whether or not it is possible to algorithmically detect whether or not such a denial-of-service attack is possible.

Assume that Alice, Bob, and the Jammer all know the channel $\mathfrak{W}$. The question is now: Is there an algorithm A that takes the underlying channel $\mathfrak{W}$ as an input and outputs $A(\mathfrak{W}) = 1$ if the Jammer is able to perform a denial-of-service attack so that $C(\mathfrak{W}) = 0$ in this case, and otherwise the algorithm outputs $A(\mathfrak{W}) = 0$?

Remark 1. *We assume that the Jammer knows the encoder and decoder. For public communication systems, this is a reasonable assumption as the encoder is usually standardized. This might not be necessarily true for the decoder, but the Jammer can simply assume the best possible decoder for the communication system at hand. If the Jammer is able to launch a denial-of-service attack for this decoder, then this will be true for any other decoder as well. However, we do not assume that the Jammer knows the actual message or codeword (which are equivalent for a deterministic encoder).*

Remark 2. *AVCs provide a basic model for secure communication with passive eavesdroppers and active jammers. They account for the communication between the legitimate users; cf. also [16–18].*

3. DETECTION FRAMEWORK

Here, we formally introduce the detection framework based on Turing machines. For this we need some basic definitions and concepts of computability. The concept of computability and computable real numbers was first introduced by Turing in [5] and [6].

A sequence of rational numbers $\{r_n\}_{n\in\mathbb{N}}$ is called a *computable sequence* if there exist recursive functions $a, b, s : \mathbb{N} \to \mathbb{N}$ with $b(n) \neq 0$ for all $n \in \mathbb{N}$ and

$$r_n = (-1)^{s(n)} \frac{a(n)}{b(n)}, \qquad n \in \mathbb{N},$$

cf. [19, Def. 2.1 and 2.2] for a detailed treatment. A real number x is said to be computable if there exists a computable sequence of rational numbers $\{r_n\}_{n\in\mathbb{N}}$ such that

$$|x - r_n| < 2^{-n}$$

for all $n \in \mathbb{N}$. We denote the set of computable real numbers by $\mathbb{R}_c$. Based on this, we define the set of computable probability distributions $\mathcal{P}_c(\mathcal{X})$ as the set of all probability distributions $P \in \mathcal{P}(\mathcal{X})$ such that $P(x) \in \mathbb{R}_c$, $x \in \mathcal{X}$. Further, let $\mathcal{CH}_c(\mathcal{X}; \mathcal{Y})$ be the set of all computable channels, i.e., for a channel $W : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ we have $W(\cdot|x) \in \mathcal{P}_c(\mathcal{Y})$ for every $x \in \mathcal{X}$. This is important since a Turing machine can only work with computable real numbers.

Definition 2. *A function $f : \mathbb{R}_c \to \mathbb{R}_c$ is called* Borel computable *if there is an algorithm that transforms each given computable sequence of a computable real x into a corresponding representation for $f(x)$.*

We note that Turing's definition of computability conforms to the definition of Borel computability above. There are weaker forms of computability known as *Markov computability* and *Banach-Mazur computability*, of which the latter one is the weakest form of computability. In particular, Borel or Markov computability implies Banach-Mazur computability, but not vice versa. For an overview of the logical relations between different notions of computability we again refer to [8] and the introductory textbook [7].

We further need the concepts of a recursive set and a recursively enumerable set as defined e.g. in [19].

Definition 3. *A set $\mathcal{A} \subset \mathbb{N}$ is called* recursive *if there exists a computable function f such that $f(x) = 1$ if $x \in \mathcal{A}$ and $f(x) = 0$ if $x \notin \mathcal{A}$.*

Definition 4. *A set $\mathcal{A} \subset \mathbb{N}$ is* recursively enumerable *if there exists a recursive function whose domain is exactly $\mathcal{A}$.*

We have the following properties; cf. for example [19]

- $\mathcal{A}$ is recursive is equivalent to $\mathcal{A}$ is recursively enumerable and $\mathcal{A}^c$ is recursively enumerable.
- There exist recursively enumerable sets $\mathcal{A} \subset \mathbb{N}$ that are not recursive, i.e., $\mathcal{A}^c$ is not recursively enumerable. This means there are no computable, i.e., recursive, functions $f : \mathbb{N} \to \mathcal{A}^c$ with $[f(\mathbb{N})] = \mathcal{A}^c$.

Now we are in the position to introduce the concept of a Turing machine. It accounts for all those problems and tasks that are algorithmically solvable on a classical (i.e., non-quantum) machine. This is equivalent to the von Neumann-architecture without hardware limitations and the theory of recursive functions, cf. [9–12].

Within our detection framework, the task of a Turing machine $\mathfrak{T}$ is to detect denial-of-service attacks on the legitimate communication. This is an *Entscheidungsproblem*, since for a given channel $\mathfrak{W}$, the Turing machine $\mathfrak{T}$ should answer the question of whether or not a denial-of-service attack takes place.

A hypothetical algorithm (or Turing machine) for answering this question is given by taking the set of all computable channels $\mathcal{CH}_c(\mathcal{X},\mathcal{S};\mathcal{Y})$ and by partitioning this set into two disjoint subsets $\mathcal{M}_{\text{denial}}$ and $\mathcal{M}^c_{\text{denial}}$. The set $\mathcal{M}_{\text{denial}}$ corresponds to all those channels $\mathfrak{W}$ for which a denial-of-service attack is possible, i.e., $\mathfrak{W} \in \mathcal{CH}_c(\mathcal{X},\mathcal{S};\mathcal{Y})$ with $C(\mathfrak{W}) = 0$. From Theorem 1 we immediately see that

$$\mathcal{M}_{\text{denial}} = \big\{\mathfrak{W} \in \mathcal{CH}_c(\mathcal{X},\mathcal{S};\mathcal{Y}) : F(\mathfrak{W}) = 0\big\}.$$

Since $\mathcal{M}_{\text{denial}}$ is characterized by the continuous function $F(\cdot)$, the set is well defined. The question is now whether or not this set is also well defined from an algorithmic point of view, i.e., does a Turing machine exist that decides whether or not $\mathfrak{W} \in \mathcal{M}_{\text{denial}}$.

4. DETECTABILITY

In this section, we consider the detectability of denial-of-service attacks. First, we study whether or not there exists a Turing machine $\mathfrak{T} : \mathcal{CH}_c(\mathcal{X},\mathcal{S};\mathcal{Y}) \to \{0,1\}$ that can detect a denial-of-service attack, i.e., $\mathfrak{T}(\mathfrak{W}) = 1$ if and only if $\mathfrak{W} \in \mathcal{M}_{\text{denial}}$. The following result shows that such a Turing machine does not exist, i.e., this question is algorithmically not decidable.

Theorem 2. *For all $|\mathcal{X}| \geq 2$, $|\mathcal{S}| \geq 2$, and $|\mathcal{Y}| \geq 3$, there is no Turing machine $\mathfrak{T} : \mathcal{CH}_c(\mathcal{X},\mathcal{S};\mathcal{Y}) \to \{0,1\}$ with $\mathfrak{T}(\mathfrak{W}) = 1$ if and only if $\mathfrak{W} \in \mathcal{M}_{denial}$.*

Sketch of the Proof. It is sufficient to prove the result for $|\mathcal{X}| = 2$, $|\mathcal{S}| = 2$, and $|\mathcal{Y}| = 3$. It extends to the case of arbitrary alphabet sizes in a straightforward way and the details are omitted for brevity.

To construct a suitable AVC for this link, we make use of an example that first appeared in [20] and that was later also discussed in [15, Example 1]. We define the AVC $\hat{\mathfrak{W}} = \{\hat{W}\}$ where $\hat{W} \in \mathcal{CH}_c(\mathcal{X},\mathcal{S};\mathcal{Y})$ is given by

$$\hat{W}(y|1,1) = \begin{pmatrix}1\\0\\0\end{pmatrix}, \qquad \hat{W}(y|1,2) = \begin{pmatrix}0\\0\\1\end{pmatrix},$$
$$\hat{W}(y|2,1) = \begin{pmatrix}0\\1\\0\end{pmatrix}, \qquad \hat{W}(y|2,2) = \begin{pmatrix}1\\0\\0\end{pmatrix}.$$

We easily observe that the symmetrizability condition

$$\begin{aligned}&\hat{W}(y|1,1)q_1 + \hat{W}(y|1,2)(1-q_1)\\ &\quad = \hat{W}(y|2,1)q_2 + \hat{W}(y|2,2)(1-q_2) \quad \forall y \in \mathcal{Y}\end{aligned}$$

is true for the choice $q_1 = 1$ and $q_2 = 0$. This means the channel $\hat{W}$ is symmetrizable, i.e., $F(\hat{\mathfrak{W}}) = 0$, cf. (1). However, we have $\min_{q\in\mathcal{P}(\mathcal{S})} C(\hat{W}_q) > 0$, which means that a denial-of-service attack is possible here.

Furthermore, for $n \in \mathbb{N}$ we consider the AVC $\mathfrak{W}_n = \{W_n\}$ specified by

$$W_n(y|1,1) = \begin{pmatrix}1\\0\\0\end{pmatrix}, \qquad W_n(y|1,2) = \begin{pmatrix}0\\0\\1\end{pmatrix},$$
$$W_n(y|2,1) = \begin{pmatrix}0\\1\\0\end{pmatrix}, \qquad W_n(y|2,2) = \begin{pmatrix}1-\frac{1}{n}\\ \frac{1}{n}\\ 0\end{pmatrix}$$

which can be shown to be non-symmetrizable, i.e., $F(\mathfrak{W}_n) > 0$.

We prove the desired result by contradiction. To do so, we assume that there exists a Turing machine that can solve the task, i.e., the characteristic function f of the set $\mathcal{M}_{\text{denial}}$ is Turing computable. Then for all computable sequences $\{W_n\}_{n\in\mathbb{N}}$ of computable channels, the sequence $\{f(W_n)\}_{n\in\mathbb{N}}$ is a computable sequence of computable numbers as well. Next, we need a concept of distance.

For two DMCs W_1 and W_2 we define the distance between W_1 and W_2 based on the total variation distance as

$$d(W_1,W_2) = \max_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} \big|W_1(y|x) - W_2(y|x)\big|.$$

The distance between two AVCs is then given by

$$D(\mathfrak{W}_1,\mathfrak{W}_2) = \max\big\{G(\mathfrak{W}_1,\mathfrak{W}_2), G(\mathfrak{W}_1,\mathfrak{W}_2)\big\}$$

with

$$G(\mathfrak{W}_1,\mathfrak{W}_2) = \max_{s_1\in\mathcal{S}_1} \min_{s_2\in\mathcal{S}_2} d\big(W_1(\cdot|\cdot,s_1), W_2(\cdot|\cdot,s_2)\big).$$

Note that $\mathcal{S}_1$ and $\mathcal{S}_2$ can be arbitrary finite state sets and we do not need to have $|\mathcal{S}_1| = |\mathcal{S}_2|$.

From the definitions of $\hat{\mathfrak{W}}$ and $\mathfrak{W}_n$ we have

$$D(\hat{\mathfrak{W}},\mathfrak{W}_n) = \frac{2}{n}.$$

Then for $n \geq \varphi(m)$ with $\varphi(m) = 2^{m+2}$, $\varphi : \mathbb{N} \to \mathbb{N}$ a computable function, we have

$$D(\hat{\mathfrak{W}},\mathfrak{W}_n) = \frac{2}{n} \leq \frac{2}{2^{m+2}} = \frac{1}{2^{m+1}} < \frac{1}{2^m}.$$

This means we have effective, i.e., computable, convergence of the sequence $\{W_n\}_{n\in\mathbb{N}}$ of computable channels to the channel $\hat{W}$, which is, accordingly, computable. But $\hat{W}$ is trivially computable anyhow as it consists only of the numbers $\{0,1\}$.

Let $\mathcal{A} \subset \mathbb{N}$ be an arbitrary recursively enumerable set such that $\mathcal{A}$ is not recursive, i.e., $\mathcal{A}^c$ is not a recursively enumerable set. With the definition of recursively enumerable sets, cf. Definition 4, we can construct a total function g, i.e., domain$(g) = \mathbb{N}$, such that $[g(\mathbb{N})] = \mathcal{A}$ and g is recursive and therewith a computable function. Furthermore, without loss of generality, we can require that $g : \mathbb{N} \to \mathcal{A}$ is a one-to-one mapping from $\mathbb{N}$ to $\mathcal{A}$.

Now, for every $(n,m) \in \mathbb{N}\times\mathbb{N}$ we define the computable function $q : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ as

$$q(n,m) = \begin{cases} 2^{m+2} & n \notin \{g(0),...,g(2^{m+2})\}\\ r & n \in \{g(0),...,g(2^{m+2})\} \text{ and } g(r) = n.\end{cases}$$

Note that r above is unique. Since $\mathcal{A}$ is recursively enumerable, the function q is recursive and therewith computable.

Next, we extend a construction of Pour-El, cf. Case I on page 336 in [21]. We consider the double sequence $\{\mathfrak{W}_{q(n,m)}\}_{n\in\mathbb{N},m\in\mathbb{N}}$ of AVCs. Note that this is only a suitable variation of the sequence $\{\mathfrak{W}_n\}_{n\in\mathbb{N}}$ which is effectively computable since q is recursive function. The idea is to show that for all $n \in \mathbb{N}$ the double sequence $\{\mathfrak{W}_{q(n,m)}\}_{n\in\mathbb{N},m\in\mathbb{N}}$ effectively converges to a channel $\hat{\mathfrak{W}}_n$. Then the sequence $\{\hat{\mathfrak{W}}_n\}_{n\in\mathbb{N}}$ is a computable sequence as well. For this purpose we have to construct a suitable function φ_n for each $n \in \mathbb{N}$, for which we then show that $\{\mathfrak{W}_{q(n,m)}\}_{m\in\mathbb{N}}$ converges effectively to $\hat{\mathfrak{W}}_n$. We can show that this sequence is a computable sequence of computable channels with

$$\hat{\mathfrak{W}}_n = \begin{cases} \hat{\mathfrak{W}} & \text{if } n \in \mathcal{A}^c \\ \mathfrak{W}_r & \text{if } n \in \mathcal{A} \text{ and } g(r) = n. \end{cases}$$

This implies that the sequence $\{f(\hat{\mathfrak{W}}_n)\}_{n\in\mathbb{N}}$ is a computable sequence. It holds that

$$f(\hat{\mathfrak{W}}_n) = 0 \;\Leftrightarrow\; n \in \mathcal{A} \quad \text{and} \quad f(\hat{\mathfrak{W}}_n) = 1 \;\Leftrightarrow\; n \in \mathcal{A}^c.$$

Accordingly, the set $\mathcal{A}^c$ must be recursively enumerable so that the set $\mathcal{A}$ is recursive, which is a contradiction to the initial assumption that $\mathcal{A}$ is recursively enumerable but not recursive. This implies that the assumption that there exists a Turing machine that can solve this task is wrong. This completes the sketch of the proof.

In Theorem 2 we required that the Turing machine $\mathfrak{T}$, if it exists, needs to stop for every channel $\mathfrak{W} \in \mathcal{CH}_c(\mathcal{X}, \mathcal{S}; \mathcal{Y})$ and that $\mathfrak{T}(\mathfrak{W}) = 1$ if and only if $\mathfrak{W} \in \mathcal{M}_{\text{denial}}$. This would imply that the corresponding characteristic function of the set $\mathcal{M}_{\text{denial}}$ would be Turing computable. However, we have seen this function is not computable and accordingly, there is no algorithm, i.e., Turing machine $\mathfrak{T}$, that can decide whether or not a given computable channel $\mathfrak{W}$ is in $\mathcal{M}_{\text{denial}}$.

Next, we are interested in dropping the requirement that the Turing machine needs to stop. For this purpose, we first study the question of whether or not there is a Turing machine that stops if and only if $\mathfrak{W} \in \mathcal{M}^c_{\text{denial}} = \mathcal{CH}_c(\mathcal{X}, \mathcal{S}; \mathcal{Y}) \backslash \mathcal{M}_{\text{denial}}$. Intuitively, $\mathfrak{W} \in \mathcal{M}^c_{\text{denial}}$ means that the Jammer is able to degrade the performance of the legitimate transmission, but is not able to completely prevent it. Accordingly, the question is addressed next.

Theorem 3. *There is a Turing machine $\mathfrak{T} : \mathcal{CH}_c(\mathcal{X}, \mathcal{S}; \mathcal{Y}) \to \{\text{stop}, \text{run forever}\}$ that stops if and only if $\mathfrak{W} \in \mathcal{M}^c_{\text{denial}}$, i.e., no denial-of-service attack is possible.*

Sketch of the Proof. The proof can be done similarly to the proof of Theorem 2 and is omitted due to space constraints.

A similar approach for $\mathcal{M}_{\text{denial}}$ (as done for $\mathcal{M}^c_{\text{denial}}$ in Theorem 3) is not possible. Otherwise we would have an immediate contradiction to Theorem 2, since it would then be possible to simply run both machines in parallel; one for $\mathcal{M}_{\text{denial}}$ and one for $\mathcal{M}^c_{\text{denial}}$. The combined Turing machine would stop whenever one of the Turing machines stops and one of these would always stop: If $\mathfrak{W} \in \mathcal{M}_{\text{denial}}$ the second machine always stops and the first one never. Likewise, if $\mathfrak{W} \in \mathcal{M}^c_{\text{denial}}$ the first machine always stops, cf. Theorem 3, and the second one never. This would imply that we would be able to compute the characteristic function of $\mathcal{M}_{\text{denial}}$ which is a contradiction.

5. DISCUSSION

The set of all channels $\mathcal{CH}_c(\mathcal{X}, \mathcal{S}; \mathcal{Y})$ can be divided into two subsets: $\mathcal{M}_{\text{denial}}$ for which a denial-of-service attack is possible and $\mathcal{M}^c_{\text{denial}}$ for which no such attack is possible. We have seen that there exists an algorithm or Turing machine that takes the channel $\mathfrak{W}$ as an input and comes to a stop if and only if $\mathfrak{W} \in \mathcal{M}^c_{\text{denial}}$. On the other hand, if $\mathfrak{W} \in \mathcal{M}_{\text{denial}}$, then Turing machine does not stop and does not output any answer. But it is important to see that we then cannot conclude that $\mathfrak{W} \in \mathcal{M}_{\text{denial}}$ must be true, since we cannot say whether the algorithm does not stop because $\mathfrak{W} \in \mathcal{M}_{\text{denial}}$ or if it would stop in the near future (e.g. after some more iterations).

A possible solution would be to consider a more general class of Turing machines that address the issue of not stopping. Assume that the Turing machine is required to always stop and is allowed to output a third option "*?*" in this case, i.e., $\mathfrak{T} : \mathcal{CH}_c(\mathcal{X}, \mathcal{S}; \mathcal{Y}) \to \{0, 1, ?\}$ with $\mathfrak{T}(\mathfrak{W}) = 0$ if and only if $\mathfrak{W} \in \mathcal{M}^c_{\text{denial}}$. But we show next that such a Turing machine cannot exist either. For such a Turing machine we could define a new Turing machine $\mathfrak{T}'$ with

$$\mathfrak{T}'(\mathfrak{W}) = \begin{cases} 0 & \mathfrak{T}(\mathfrak{W}) = 0 \\ 1 & \mathfrak{T}(\mathfrak{W}) = 1 \text{ or } \mathfrak{T}(\mathfrak{W}) = ?. \end{cases}$$

For this, we have $\mathfrak{T}'(\mathfrak{W}) = 0$ if $\mathfrak{W} \in \mathcal{M}^c_{\text{denial}}$. Accordingly, we must have $\mathfrak{T}'(\mathfrak{W}) = 1$ if $\mathfrak{W} \notin \mathcal{M}^c_{\text{denial}}$, i.e., $\mathfrak{W} \in \mathcal{M}_{\text{denial}}$. The latter implies that the Turing machine $\mathfrak{T}'$ computes the characteristic function of the set $\mathcal{M}_{\text{denial}}$, but this is impossible. Accordingly, such a Turing machine cannot exist.

Another possible solution would be to study the class: $\mathfrak{T}''(\mathfrak{W}) \in \{0, 1\}$ or $\mathfrak{T}''$ does not stop. We again require that $\mathfrak{T}''(\mathfrak{W}) = 0$ if and only if $\mathfrak{W} \in \mathcal{M}^c_{\text{denial}}$. Then for $\mathfrak{W}$ with $\mathfrak{T}''(\mathfrak{W}) = 1$ we must have $\mathfrak{W} \in \mathcal{M}_{\text{denial}}$, i.e., in this case the Turing machine $\mathfrak{T}''$ provides the correct answer. However, the set of channels $\mathfrak{W}$ with $\mathfrak{T}''$ does not stop must be large. To see this, consider the following example: If we take two channels $\mathfrak{W}^1 \in \mathcal{M}^c_{\text{denial}}$ and $\mathfrak{W}^2$ such that $\mathfrak{T}''(\mathfrak{W}^2) = 1$, then the set

$$\left\{\mathfrak{W} = \{W_s\}_{s\in\mathcal{S}} : \exists \lambda \in (0, 1) \cap \mathbb{R}_c : W_s = (1 - \lambda)W^1_s + \lambda W^2_s\right\}$$

must necessarily contain elements of the set

$$\left\{\mathfrak{W} = \{W_s\}_{s\in\mathcal{S}} : \mathfrak{T}''(\mathfrak{W}) \text{ does not stop}\right\}$$

since otherwise the function $f(\lambda) = \mathfrak{T}''(\mathfrak{W})$ would be continuous in $\lambda \in \mathbb{R}_c \cap [0, 1]$. But this cannot be true since $\{f(\lambda)\} \subset \{0, 1\}$ for $\lambda \in \mathbb{R}_c \cap [0, 1]$ and we have seen that the function takes both values. We conclude that such a Turing machine can also not exist.

6. RELATION TO PRIOR WORK AND OUTLOOK

Communication under adversarial attacks has been studied under various aspects, cf. for example [22] and [23] as well as the overview papers [3] and [4]. However, communication from a computability or algorithmic point of view has attracted much less attention. Such studies are crucial for the standardization and verification of secure applications for the Tactile Internet [1, 2].

To the best of our knowledge, there is only one work that falls into this category. In [24] the computability of the capacity function of the wiretap channel under adversarial attacks is studied. The computability of the identification capacity under feedback is studied in [25] and that of the identification capacity of the correlation-assisted channel is studied in [26]. The computability of secret key generation with a rate-limited public channel is addressed in [27]. It is evident that algorithmic computability, i.e., Turing computability, plays a major role in system design and system analysis.

7. REFERENCES

[1] Gerhard Fettweis, Holger Boche, Thomas Wiegand, *et al.*, "The Tactile Internet," Tech. Rep., ITU-T Tech. Watch Rep., Aug. 2014.

[2] IEEE ComSoc Tactile Internet Emerging Technical Subcommittee. [Online]. Available: http://ti.committees.comsoc.org/standardisation/

[3] Amos Lapidoth and Prakash Narayan, "Reliable communication under channel uncertainty," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2148–2177, Oct. 1998.

[4] Rafael F. Schaefer, Holger Boche, and H. Vincent Poor, "Secure communication under channel uncertainty and adversarial attacks," *Proc. IEEE*, vol. 103, no. 10, pp. 1796–1813, Oct. 2015.

[5] Alan M. Turing, "On computable numbers, with an application to the Entscheidungsproblem," *Proc. London Math. Soc.*, vol. 2, no. 42, pp. 230–265, 1936.

[6] Alan M. Turing, "On computable numbers, with an application to the Entscheidungsproblem. A correction," *Proc. London Math. Soc.*, vol. 2, no. 43, pp. 544–546, 1937.

[7] Klaus Weihrauch, *Computable Analysis - An Introduction*, Springer-Verlag, Berlin Heidelberg, 2000.

[8] Jeremy Avigad and Vasco Brattka, "Computability and analysis: The legacy of Alan Turing," in *Turing's Legacy: Developments from Turing's Ideas in Logic*, Rod Downey, Ed. Cambridge University Press, Cambridge, UK, 2014.

[9] Kurt Gödel, "Die Vollständigkeit der Axiome des logischen Funktionenkalküls," *Monatshefte für Mathematik*, vol. 37, no. 1, pp. 349–360, 1930.

[10] Kurt Gödel, "On undecidable propositions of formal mathematical systems," *Notes by Stephen C. Kleene and Barkely Rosser on Lectures at the Institute for Advanced Study, Princeton, NJ*, 1934.

[11] Stephen C. Kleene, *Introduction to Metamathematics*, Wolters-Noordhoffv, Van Nostrand, New York, 1952.

[12] Marvin Minsky, "Recursive unsolvability of Post's problem of 'tag' and other topics in theory of Turing machines," *Ann. Math.*, vol. 74, no. 3, pp. 437–455, 1961.

[13] Imre Csiszár and Prakash Narayan, "The capacity of the arbitrarily varying channel revisited: Positivity, constraints," *IEEE Trans. Inf. Theory*, vol. 34, no. 2, pp. 181–193, Mar. 1988.

[14] Rafael F. Schaefer, Holger Boche, and H. Vincent Poor, "Super-activation as a unique feature of secure communication in malicious environments," *Information*, vol. 7, no. 2, May 2016.

[15] Rudolf Ahlswede, "Elimination of correlation in random codes for arbitrarily varying channels," *Z. Wahrscheinlichkeitstheorie verw. Gebiete*, vol. 44, no. 2, pp. 159–175, June 1978.

[16] Moritz Wiese, Janis Nötzel, and Holger Boche, "A channel under simultaneous jamming and eavesdropping attack—Correlated random coding capacities under strong secrecy criteria," *IEEE Trans. Inf. Theory*, vol. 62, no. 7, pp. 3844–3862, July 2016.

[17] Janis Nötzel, Moritz Wiese, and Holger Boche, "The arbitrarily varying wiretap channel—Secret randomness, stability, and super-activation," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3504–3531, June 2016.

[18] Holger Boche and Christian Deppe, "Secure identification under passive eavesdroppers and active jamming attacks," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 2, pp. 472–485, Feb. 2019.

[19] Robert I. Soare, *Recursively Enumerable Sets and Degrees*, Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 1987.

[20] David Blackwell, Leo Breiman, and A. J. Thomasian, "The capacities of certain channel classes under random coding," *Ann. Math. Statist.*, vol. 31, no. 3, pp. 558–567, Sept. 1960.

[21] Marian Boykan Pour-El, "A comparison of five "computable" operators," *Math. Logic Quart.*, vol. 6, no. 15-22, pp. 325–340, 1960.

[22] Holger Boche and Rafael F. Schaefer, "Capacity results and super-activation for wiretap channels with active wiretappers," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 9, pp. 1482–1496, Sept. 2013.

[23] Holger Boche and Rafael F. Schaefer, "Optimal transceiver design for wiretap channels with side information," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, Vancouver, Canada, May 2013.

[24] Holger Boche, Rafael F. Schaefer, and H. Vincent Poor, "Performance evaluation of secure communication systems on Turing machines," in *Proc. 10th IEEE Int. Workshop Inf. Forensics Security*, Hong Kong, Dec. 2018, pp. 1–7.

[25] Holger Boche, Rafael F. Schaefer, and H. Vincent Poor, "Identification capacity of channels with feedback: Discontinuity behavior, super-activation, and Turing computability," submitted.

[26] Holger Boche, Rafael F. Schaefer, and H. Vincent Poor, "Identification capacity of correlation-assisted discrete memoryless channels: Analytical properties and representations," submitted.

[27] Holger Boche, Rafael F. Schaefer, and H. Vincent Poor, "On the computability of the secret key capacity under rate constraints," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, Brighton, UK, May 2019.