Deep Learning for Signal Authentication and Security in Massive Internet of Things Systems

Aidin Ferdowsi* and Walid Saad*

* Wireless@VT, Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, USA, Emails: {aidin,walids}@vt.edu.

Abstract

Secure signal authentication is arguably one of the most challenging problems in the Internet of Things (IoT) environment, due to the large-scale nature of the system and its susceptibility to man-in-the-middle and eavesdropping attacks. In this paper, a novel deep learning method is proposed for dynamic authentication of IoT signals to detect cyber attacks. The proposed learning framework, based on a long short-term memory (LSTM) structure, enables the IoT devices (IoTDs) to extract a set of stochastic features from their generated signal and dynamically watermark these features into the signal. This method enables the cloud, which collects signals from the IoT devices, to effectively authenticate the reliability of the signals. Moreover, in massive IoT scenarios, since the cloud cannot authenticate all the IoTDs simultaneously due to computational limitations, a game-theoretic framework is proposed to improve the cloud’s decision making process by predicting vulnerable IoTDs. The mixed-strategy Nash equilibrium (MSNE) for this game is derived and the uniqueness of the expected utility at the equilibrium is proven. In the massive IoT system, due to a large set of available actions for the cloud, it is shown that analytically deriving the MSNE is challenging and, thus, a learning algorithm proposed that converges to the MSNE. Moreover, in order to cope with the incomplete information case in which the cloud cannot access the state of the unauthenticated IoTDs, a deep reinforcement learning algorithm is proposed to dynamically predict the state of unauthenticated IoT devices and allow the cloud to decide on which IoT devices to authenticate. Simulation results show that, with an attack detection delay of under 1 second the messages can be transmitted from IoT devices with an almost 100% reliability. The results also show that, by optimally predicting the set of vulnerable IoTDs, the proposed deep reinforcement learning algorithm reduces the number of compromised IoT devices by up to 30%, in a massive IoT, compared to an equal probability baseline.

This research was supported by the U.S. National Science Foundation under Grants OAC-1541105 and CNS-1524634. A preliminary version of this work appears in the proceedings of IEEE ICC 2018 [1].

arXiv:1803.00916v1 [cs.CR] 1 Mar 2018
I. Introduction

The Internet of Things (IoT) will encompass a massive number of devices that must reliably transmit a diverse set of messages observed from their environment to delivery a plethora of smart city applications [2]–[9]. However, an effective deployment of these diverse IoT services require near real-time, secure, and low complexity message transmission from the IoT devices [10]. Most IoT architectures consist of four layers: perceptual, network, support, and application layer [11]. The perceptual layer is the most fundamental layer which collects all kind of information from the physical world using radio frequency identification (RFID) tags, global positioning system (GPS), accelerometer, and gyroscope data. The other three layers are centered around communication and computation of IoT signals as well as personalized services for end users. Due to simplicity of the devices and components at the perceptual layer, their resource-constrained nature, and their low computational and storage capabilities, securing the IoT signals at this layer is notoriously challenging [11].

Recently, a number of security solutions have been proposed for IoT signal authentication [12]–[18]. The work in [12] investigated physical layer security techniques for securing IoT applications. These methods include optimal sensor censoring, channel-based bit flipping, probabilistic ciphering of quantized IoT signals, and artificial noise signal transmission. In [13], the author suggested bridging the security gap in IoT devices by applying information theory and cryptography at the physical layer of the IoT. An authentication protocol for the IoT is presented in [14], using lightweight encryption method in order to cope with constrained IoT devices. Moreover, in [15], the authors developed a learning mechanisms for fingerprinting and authenticating IoT devices and their environment. Other useful signal authentication approaches are also found in [16]–[18].

To provide further security for IoT-like cyber-physical systems (CPSs), the idea of watermarking has been studied in [19]–[22]. In [19], a method was proposed to watermark a predefined signal unto a CPS input signal that enables detection of replay attacks in which an adversary repeats a sequence of past measurements. A dynamic watermarking algorithm was proposed in [20] for integrity attack detection in networked CPSs. In [21], the authors introduced a security scheme that ensures detection of attacks that add a nonzero-average-power signal to sensor measurements using a non-stationary watermarking technique. Finally, the authors in [22] analyzed the optimality of Gaussian watermarked signals against cyber attacks in linear
time-variant IoT-like systems.

Moreover, security of massive IoT systems has gained attention in recent years [23]–[26]. In [23], a blockchain-based approach was proposed to provide a distributed security solution for IoT systems with a large number of IoT devices. In [24], the authors demonstrate the feasibility of implementing existing device management protocols such as SNMP and NETCONF on resource-constrained devices. A cloud-based algorithm was proposed in [25], in which the authors designed a suspicious bucket cross-filtering to provide privacy service for resource-constrained devices in a IoT system. Furthermore, the work in [26] investigated the secure integration of cloud computing in IoT systems by addressing its scalability in massive networking scenarios.

However, the authentication solutions of [12]–[18] remain highly complex for deployment at the IoT perceptual layer, and require high computational power. Moreover, these methods do not take into account complex attacks in which the attacker collects data for a long time duration and uses it for designing undetectable attacks. Furthermore, the watermarking algorithms introduced in [19]–[22], can be detrimental to the performance of a system such as the IoT since an augmented watermark is applied in parallel to the control signal of the system. This can, in turn, cause non-optimality in the performance of the system. In addition, the input signals to IoTDs include information such as temperature, heart rate, and location, which are not controllable and require changing the IoT devices’ environment. In addition, the works in [23]–[26] do not consider resource constraints at the cloud for authentication. In a large-scale IoT system, the cloud cannot authenticate all of the transmitted signals from the IoT devices (IoTDs) due to the large amount of required computational resources as well as bandwidth. Therefore, in a practical IoT, the cloud must optimally and intelligently choose which IoTDs to authenticate.

The main contribution of this paper is a comprehensive framework that integrates together new ideas from deep learning and game theory to enable computationally-efficient authentication of IoT signals and devices, in massive IoT systems. The key contributions of this paper can be summarized as follows:

- We propose a novel framework that enables the IoT's cloud to authenticate the reliability of signals and detect the existence of a cyber attacker who seeks to degrade the performance of the IoT by changing the devices’ output signal. The proposed deep learning algorithm uses long short-term memory (LSTM) [27] blocks to extract stochastic features such as spectral flatness, skewness, kurtosis, and central moments from IoT signal and watermarks these features inside the original signal.
• This dynamic feature extraction allows the cloud to detect sophisticated eavesdropping attacks, since the attacker will not be able to extract the watermarked information. Moreover, the proposed LSTM reduces complexity and latency of the attack detection compared to other security methods such as encryption. This allows LSTM to effectively complement the cryptographic and security solutions of an IoT.

• To enable the cloud to authenticate IoTDs in a massive IoT system and under resource constraints, we formulate a noncooperative game between the cloud and the attacker to address the cloud’s optimal action in predicting vulnerable IoTDs while considering its computational resource constraints. We derive the mixed-strategy Nash equilibrium and prove the uniqueness of the cloud’s expected utility, at this equilibrium.

• In addition, we show that analytically finding the cloud’s optimal strategy is highly complicated in the massive IoT scenario. Therefore, we propose a learning algorithm based on fictitious play that converges to the mixed-strategy Nash equilibrium. Finally, we consider a practical case in which the cloud does not have complete information about all the IoTDs. To address this challenge we propose a deep reinforcement learning approach based on LSTM blocks, to learn the state of the IoTDs based on their past states. We show that the cloud’s expected utility using this proposed deep reinforcement learning method is higher than baseline scenarios in which the cloud authenticates all the IoTDs with the same probability.

• Simulation results show that, using the proposed approach and for under 1 second latency, the IoT signals can be reliably transmitted from IoT devices to the cloud. Moreover, in the massive IoT scenario the proposed learning algorithms improves the protection of the system by reducing by about 30% the number of compromised IoTDs.

The rest of the paper is organized as follows. Section II introduces the system model for IoTD-cloud signal transmission in large-scale IoT systems and proposes the deep learning-based dynamic watermarking for IoTD-cloud signal authentication. Section III analyzes the authentication scenario in the case of a massive IoT with limited resources at the cloud. Section IV presents the simulation results and their analysis while conclusions are drawn in Section V.

II. IoT SIGNAL AUTHENTICATION: SYSTEM MODEL AND DEEP LEARNING SOLUTION

Consider a massive IoT system having a set $\mathcal{N}$ of $N$ IoTDs communicating with a cloud as shown in Fig. 1. Any IoTD $i$ in the system generates a signal $y_i(t)$ at time step $t$ with sampling frequency $f_{s_i}$, and transmits this signal to a cloud, that uses the received signal for estimation and
The cloud must solve two inter-related problems: 1) How to authenticate the IoTDs and 2) Which IoTDs to authenticate. In this system, we consider an adversary who seeks to compromise the IoTD by collecting the data from communication link between the IoTD and cloud, and, then, manipulates the transmitted signal. In this case, the transmitted signal from each IoTD \( i \) will be \( \tilde{y}_i(t) \neq y_i(t) \) which will cause an estimation error at the cloud. Therefore, the cloud must implement an attack detection mechanism to differentiate between the honest and false signal received from an IoTD.

First, we analyze the signal authentication process between a single IoTD and the cloud and then we propose a dynamic watermarking framework based on a deep learning algorithm to authenticate this signal communication with a very small delay. Subsequently, in Section III, we study the case in which, due to the massive nature of the system, the cloud is unable to authenticate all IoTDs and must choose an optimal subset to authenticate.

A. Spread Spectrum Watermarking

Watermarking uses a hidden, predefined non-perceptual code (bit stream) inside a multimedia signal to authenticate the ownership of such signals. One of the the most widely used water-
marking methods is spread spectrum (SS) [28] in which a key pseudo-noise sequence is added to the original signal. The watermarked signal from each IoTD \( i \) can then be written as follows:

\[
w_i(t) = y_i(t) + \beta_i b_i p_i(t) \quad \text{for } t = 1, \ldots, n_i,
\]

where \( w_i \) is the IoTD \( i \)'s watermarked signal, \( p_i \) is a pseudo-noise binary sequence of +1 and −1 for IoTD \( i \), \( \beta_i \) is the IoTD \( i \)'s relative power of the pseudo-noise signal to the original signal, \( b_i \) is the hidden bit in the signal which can take values +1 and −1, and \( n_i \) is the number of samples (frame length) of IoTD \( i \)'s original signal used to hide a single bit. To extract the watermarked bit, the cloud receives the watermarked signal from each IoTD \( i \) and correlates it with the key pseudo-noise sequence. The extraction process will be:

\[
\tilde{b}_i = \frac{< w_i, p_i >_{n_i}}{\beta_i n_i} = \frac{< y_i, p_i >_{n_i} + \beta_i b_i < p_i, p_i >_{n_i}}{\beta_i n_i} = \tilde{y}_i + b_i,
\]

where, for \( \tilde{b}_i > 0 \) the extracted bit is 1 and for \( \tilde{b}_i < 0 \) the extracted bit is −1 and \( < w_i, p_i >_{n_i} \) is the inner production of \( n_i \) samples of \( w_i \) and \( p_i, p_i(t) \) and \( y_i(t) \) are independent stochastic variables at time \( t \) and we consider \( y_i(t) \) having mean \( \mu_i \) and variance \( \sigma_i^2 \). Next, we analyze the bit error rate of the extracted bit to evaluate the proposed watermarking scheme’s performance.

**Theorem 1.** In the proposed SS watermarking scheme, the bit extraction error for IoTD \( i \) is

\[
\frac{1}{2} \text{erfc} \left( \frac{\beta_i \sqrt{n_i}}{\sigma_i \sqrt{2}} \right).
\]

**Proof.** For \( \tilde{y}_i \), we can write:

\[
\tilde{y}_i = \frac{1}{\beta_i n_i} \sum_{t=1}^{n} y_i(t)p_i(t) = \frac{1}{\beta_i n_i} \left( \sum_{t \in \mathcal{P}_i^+} y_i(t) - \sum_{t \in \mathcal{P}_i^-} y_i(t) \right)
\]

\[
= \frac{n_{i+}}{\beta_i n_i n_{i+}} \sum_{t \in \mathcal{P}_i^+} y_i(t) - \frac{n_{i-}}{\beta_i n_i n_{i-}} \sum_{t \in \mathcal{P}_i^-} y_i(t),
\]

where \( \mathcal{P}_i^+ = \{ t|p_i(t) = 1 \} \), \( \mathcal{P}_i^- = \{ t|p_i(t) = -1 \} \), \( n_{i+} \triangleq |\mathcal{P}_i^+| \), and \( n_{i-} \triangleq |\mathcal{P}_i^-| \). Since (3) can be expressed as sum of i.i.d variables for large values of \( n_{i+} \) and \( n_{i-} \), then, using the central limit theorem, we can write (3) as linear combination of two Gaussian distributions as

\[
Y_{i_1} \sim \mathcal{N} \left( \frac{\mu_i}{\beta_i n_i}, \frac{n_{i+}}{\beta_i^2 n_i^2} \sigma_i^2 \right), \quad Y_{i_2} \sim \mathcal{N} \left( \frac{\mu_i}{\beta_i n_i}, \frac{n_{i-}}{\beta_i^2 n_i^2} \sigma_i^2 \right).
\]

Since \( \tilde{y} \) is a linear combination of two independent Gaussian distributions, then we have:

\[
\tilde{y}_i \sim \mathcal{N} \left( \frac{\mu_i}{\beta_i n_i} - \frac{\mu_i}{\beta_i n_i} + \frac{n_{i+}}{\beta_i^2 n_i^2} \sigma_i^2 + \frac{n_{i-}}{\beta_i^2 n_i^2} \sigma_i^2 \right),
\]

\[
\tilde{y}_i \sim \mathcal{N} \left( 0, \frac{1}{\beta_i^2 n_i} \sigma_i^2 \right).
\]
Now, we can show that \( \tilde{b} \) is a Gaussian variable since it is a summation of a constant value with a Gaussian variables:

\[
\tilde{b}_i \sim \mathcal{N}\left( E(\tilde{b}_i) = b_i, \sigma^2_{\tilde{b}_i} = \frac{\sigma^2_i}{\beta^2_i n_i} \right).
\]  
(6)

Then, to analyze the probability of error we consider \( b_i = 1 \). In this case an error occurs when \( \tilde{b}_i < 0 \), therefore, the probability of an error is:

\[
\Pr\left\{ \tilde{b}_i < 0 | b_i = 1 \right\} = \frac{1}{2} \text{erfc}\left( \frac{E(\tilde{b}_i)}{\sigma_{\tilde{b}_i}\sqrt{2}} \right) = \frac{1}{2} \text{erfc}\left( \frac{\beta_i \sqrt{n_i}}{\sigma_i \sqrt{2}} \right).
\]  
(7)

The same error probability can be obtained for \( b_i = -1 \).

From (7) we can observe that for large values of \( \beta_i \) and \( n_i \), the bit extraction error goes to zero. However, selecting large values for \( \beta_i \) and \( n_i \) will cause some latency and computational challenges for IoT devices which will be discussed next.

B. Static Watermarking for Attack Detection in IoTD

Now, using the SS method we present a technique for authentication of the signals transmitted from an IoTD to the cloud. We first generate a random pseudo-noise binary sequence with \( n_i \) samples. Also, for every IoTD \( i \), we define a bit stream \( s_i \) with \( n_{s_i} \) samples. Then using (1), we embed every bit of \( s_i \) in \( n_i \) samples of \( y_i \). Therefore, for any bit stream \( s_i \), we use \( n_i n_{s_i} \) samples of \( y_i \), and this embedding procedure will repeat every \( n_i n_{s_i} \) samples of \( y_i \). At the cloud, using (2), we extract the bit stream. In case of a cyber attack, the received signal in the cloud will be \( \bar{y}_i(t) \) rather than \( w_i(t) \), and, hence, the extracted bit stream will differ from \( s_i \). Thus, the cloud will trigger an alarm for declaring the existence of a cyber attack. Fig. 2 shows the block diagram of static watermarking for attack detection in an IoTD.

In our proposed watermarking scheme, \( \beta_i \), \( n_i \), and \( n_{s_i} \) play crucial roles in the security of a given IoTD. The value of \( \beta_i \) must be much smaller than the value of \( \sigma_i \). Indeed, for comparable
values of $\beta_i$, an attacker can extract the key $p_i$ and bit stream $s_i$, since $w_i(t) \approx \beta_i b_i p_i(t)$ if $|\beta_i b_i p_i(t)| = \beta_i \gg |y_i(t)|$. Therefore, we must choose small values for $\beta_i$. However, from Theorem 1, we know that a small $\beta_i$ will yield a higher bit error rate. To overcome this issue, we have to increase the pseudo-noise key length, $n_i$. Although increasing the value of $n$ will reduce the bit error rate, for large values of $n_i$, the bit extraction procedure in (2) will result in higher computation load and will also cause higher latency since the cloud must wait for $n_i n_s$ samples from the IoTD to detect the attack. Moreover, large values of $n_s$ will also cause larger delay in the cloud. Therefore, next, we propose a method to choose suitable values for these three parameters.

**Theorem 2.** To reduce the attacker’s ability to extract the hidden bit stream as well as minimize the attack detection delay of the watermarking scheme $\beta_i, n_i$, and $n_s$ must be selected to satisfy the following conditions:

$$\frac{1}{2} \text{erfc} \left( \frac{(1 + \frac{\mu_{i1}}{\beta^2 n_i}) \beta^2 n_i \sqrt{n_i}}{\sqrt{2(\sigma^2_{i1} + 2\sigma^2_i)}} \right) \geq 1 - P, \quad (8)$$

$$\frac{1}{2} \text{erfc} \left( \frac{\beta \sqrt{n_i}}{\sigma_i \sqrt{2}} \right) \leq \bar{P}, \quad (9)$$

$$n_s \leq \frac{df_{s_i}}{n_i}, \quad (10)$$

where $d$ is the acceptable delay in seconds for attack detection, $\mu_{i1}$ and $\sigma^2_{i1}$ are the mean and variance of the multiplication of random variables $y_{1i}(t)$ and $y_{2i}(t)$ with the same distribution as $y_i(t)$, $P$ is our desired probability of unsuccessful, and $\bar{P}$ is our desired bit extraction error probability attack.

**Proof.** See Appendix A

Using (8), (9), and (10), we can find the values for the three parameters which satisfy our performance and delay constraints. The proposed SS watermarking method can detect a cyber attack which can only change the transmitted data from an IoTD to the cloud. By choosing optimal values for the three parameters of watermarking, the cloud can identify the reliability of the transmitted attack. Now, consider a case in which an attacker can also collect data from the IoTDs. In this case, the attacker can launch more complex attacks by eavesdropping the legitimate transmitted data from the IoTD. For instance, if the attacker starts to record the transmitted data from the IoTD and then sums the recorded data for a long time, it can potentially reveal the
key \( p \). As an example, if the attacker collects the data for \( m \) windows of size \( m_n \) and adds this data together, it obtains:

\[
\bar{w}_{mi}(t) = \sum_{i=1}^{m} y_i^j(t) + m\beta_i b_ip_i(t),
\]

(11)

where \( y_i^j \) is the signal received in window \( j \) from an IoTD, and \( \bar{w}_{mi} \) is the summation of collected data. If we consider \( \sigma^2_{mi} \) as the variance of the sum of \( m \) random variable \( y_i^1(t), \ldots, y_i^m(t) \) then there will exists a value for \( m \) where \( m^2\beta_i^2 >> \sigma^2_{mi} \). Therefore, the attacker can use \( \bar{w}_{mi} \simeq m\beta_ib_ip_i \) as the key for watermarked signal. This attack is successful because the embedded stream \( s \) is static, at all time windows. However, if the bit stream \( s \) changes dynamically in each window of \( n_in_{si} \) samples then, the system can deter such an eavesdropping attack. To this end, next, we propose a dynamic bit stream generation using deep learning.

C. Deep Learning for Dynamic IoT Signal Watermarking

To improve our authentication scheme, we propose a novel deep learning watermarking method for dynamically generating the bit stream \( y_i \) which can thwart eavesdropping attacks. In the proposed dynamic watermarking scheme, we use the fingerprints of the signal \( y_i \) generated by an IoTD to dynamically update the bit stream, \( s_i \). Signal fingerprints can be seen as unique identifiers of a signal that can be mapped to a bit stream. Signal processing characteristics such as spectral flatness, central moments, skewness, and kurtosis can be used for extraction of fingerprints from signals \[29\]–[32]. Due to time dependence of the IoTD signal stream to past time steps, we use the powerful deep LSTM \[27\], \[33\], \[34\] framework, one of the most effective deep learning methods for sequence classification, to extract the fingerprints from the signals.

1) LSTM for Dynamic Signal Watermarking at the IoTDS: To dynamically extract fingerprints from IoTD signals, we propose an LSTM algorithm that allows an IoTD to update the bit stream based on the sequence of generated data. LSTMs are deep recurrent neural networks (RNNs) that can store information for long periods of time and thus can learn long-term dependencies within a given sequence \[34\]. Essentially, an LSTM algorithm processes an input \((y_i(1), \ldots, y_i(n_in_{si}))\) by adding new information into a memory, and using gates which control the extent to which new information should be memorized, old information should be forgotten, and current information should be used. Therefore, the output of an LSTM algorithm will be impacted by the network
activation in previous time steps and thus LSTMs are suitable for our IoT application in which we want to extract fingerprints from signals which are dependent on previous time steps.

During the training phase, the parameters of the LSTM algorithm are learned from a given training dataset of different IoTD such as accelerometer, gyroscope and positioning devices. As done in [29]–[32], we choose spectral flatness, mean, variance, skewness, and kurtosis as features that are extracted from a signal of length $n_i n_s$ and then map these values to a bit stream with length $n_i$. Next, we watermark this extracted bit stream into the original signal using a key. To train the LSTM, we use the original signal $y_i$ and a pseudo noise key $p_i$ as input stream and the watermarked signal $w_i$ as output stream. Fig. 3a shows the training phase. Next, we illustrate how we use the trained LSTM to dynamically watermark an IoT signal.

2) LSTM for Dynamic Signal Authentication at the Cloud: At the cloud, we use a dynamic watermarking LSTM (DW-LSTM) for bit extraction. To train this DW-LSTM, we use the watermarked signal $w_i$ and key $p_i$ as inputs to the neural network and the features of the original signal and extracted bit stream as outputs. The block diagram model for training phase of DW-LSTM in the cloud is shown in Fig. 3b. Using the DW-LSTM block of Fig. 3a at the IoTD and the DW-LSTM block of Fig. 3b at the cloud, we propose a dynamic LSTM watermarking scheme to implement an attack detector at the cloud.

In this method, a predefined bit stream is not used since a bit stream is dynamically generated inside the LSTM blocks at the IoTD and the cloud. This dynamic bit stream generation at the hidden layers of LSTMs solves the eavesdropping attack problem, since recording and summing the IoTD signals will not increase the power ratio of the key sequence to the signal and the attacker will not be able to extract the key and bit stream. Using this method, the IoTD inserts the generated signal and key in its LSTM block in each window of $n_i n_s$ samples and produces a watermarked signal with different bit stream $s_i$ in each window. At the cloud, the received
watermarked signal and key are passed from the LSTM. Then, the two outputs (the extracted bits, and extracted features) are compared. In the case of dissimilarity of two sequences, the attack alarm is triggered. Fig. 4 shows the block diagram of dynamic LSTM watermarking for attack detection. The proposed algorithm enables the cloud to authenticate any IoTD signal with a delay of $d$. To analyze the required computational resource at the cloud, next, we derive the computational complexity of the proposed watermarking algorithm.

**Proposition 1.** The complexity of the proposed signal authentication method at the cloud is bounded by $O(df_i^s)$.

**Proof.** From (2), we know that, in order to extract each bit in stream $s$, we need $n$ multiplications and $n-1$ summations. Therefore, the complexity of extracting one bit is $O(n_i)$ and extracting all of the bits in $s$ will have a complexity of $O(n_i n_s)$. Moreover, from Theorem 2, we know that $n_i$ is chosen based on the watermarking performance criteria, while $n_s$ is bounded by $\frac{df_i^s}{n_i}$. Thus, the complexity of signal authentication at the cloud is bounded by $O\left(\frac{df_i^s}{n_i} n_i\right) = O\left(df_i^s\right)$. ■

Proposition 1 shows that the sampling rate of an IoTD’s signal directly affects the complexity of the authentication process. However, as previously mentioned, an IoTD with higher sampling and packet transmission rates to the cloud will be more valuable for the IoT system [35]. Thus, with limited computational resources, the cloud can only authenticate a limited number of IoTD signals using our proposed method and, thus, it has to choose which subset of $\mathcal{N}$ it can authenticate. This computational limitation provides an opportunity for the attacker to choose the IoTDs with unauthenticated signals to attack and stay undetected. We assume that the amount of computations that can be done at the cloud is bounded by $O(C)$ which means that the total authentication complexity of all the received signals cannot exceed $O(C)$:

$$O\left(d \sum_{i \in S} f_i^s\right) \leq O(C), \quad (12)$$
where $S \subseteq N$ is the set of IoTDs whose signals will be authenticated by the cloud and $f^i_s$ is the sampling rate of IoTD $i$. Since the arguments on both sides of (12) are linear, we have:

$$\sum_{i \in S} f^i_s \leq \frac{C}{d}. \quad (13)$$

Since the sampling rate $f^i_s$ of each IoTD corresponds to its value for the cloud, $v_i$, by considering a value proportional to each IoTD’s sampling frequency, $v_i = \frac{f^i_s}{\sum_{i=1}^N f^i_s}$, we can rewrite (13) as:

$$\sum_{i \in S} v_i \leq \frac{C}{d} \sum_{i=1}^N f^i_s \triangleq R, \quad (14)$$

Thus, the cloud must choose a set $S \subseteq N$ that satisfies (14). In addition, the attacker has a limitation on the number of IoTDs that it can attack simultaneously due to its limited available resources. We capture this resource limitation by assuming that the attacker has only $K$ devices that it can use to eavesdrop IoTDs. Thus, the attacker must choose $K$ target IoTDs, that are not $S$, while the cloud must predict the IoTDs that the attacker will target and, then, it will authenticate them. Since the attacker’s and the cloud’s actions are interdependent, the outcome of their decisions requires analyzing their interaction. In the following, we address this interaction between the cloud and the attacker in the large-scale IoT system using a game-theoretic approach [36].

III. GAME THEORY FOR AUTHENTICATION UNDER COMPUTATIONAL CONSTRAINTS

In the considered massive IoT system, we assume that IoTDs that have more data to send will be more valuable for the system since more important applications require more frequent monitoring and control [35]. However, IoTDs with more valuable data are also more likely to be selected as an attack target by the adversary. As discussed in Section II, IoTD signal authentication using our proposed technique requires computational resources at the cloud to process the received data from all IoTDs. Therefore, the cloud must optimally predict the vulnerable IoTDs while the attacker must predict which unauthenticated IoTDs to target so as to maximize the disruption in the IoT system. We analyze this problem using game theory.

A. Game Formulation

To model the interdependent decision making processes of the attacker and the cloud, we introduce a noncooperative game $\{P, Q^i, u^j, K, C\}$ defined by five components: a) the players

\footnote{Without loss of generality, our approach can accommodate any other relationship between the IoTD’s value and sampling frequency.}
which are the attacker \(a\) and the cloud \(c\) in the set \(\mathcal{P} \triangleq \{a, c\}\), b) the strategy spaces \(\mathcal{Q}^j\) for each player \(j \in \mathcal{P}\), c) a utility function, \(u^j\) for each player, d) the number of attacker’s eavesdropping devices, \(K\) and e) the available computational resources for the cloud. For the cloud, the set of pure strategies \(\mathcal{Q}^c\) corresponds to different feasible IoTD subsets whose signals can be authenticated without exceeding the available computational resources:

\[
\mathcal{Q}^c = \left\{ S \subset \mathcal{N} \left| \sum_{i \in S} v_i \leq R \right. \right\}.
\]

On the other hand, for the attacker the set of pure strategies \(\mathcal{Q}^a\) is a set of \(K\) IoTDs that will be targeted by an attack: \(\mathcal{Q}^a = \left\{ \mathcal{K} \subset \mathcal{N} \left| |\mathcal{K}| \leq K \right. \right\}\). Moreover, the utility function of each player can be written as follows:

\[
u^c(S, \mathcal{K}) = \sum_{i=1}^{N} v_i - \sum_{i \in \mathcal{K}, i \notin S} v_i = 1 - \sum_{i \in \mathcal{K}, i \notin S} v_i, \quad u^a(\mathcal{K}, S) = \sum_{i \in \mathcal{K}, i \notin S} v_i,
\]

where \(v_i\) is IoTD \(i\)'s value. These utility functions essentially capture the fraction of secured IoTD signals for the defender, and the fraction of IoTDs that the attacker can compromise while remaining undetected. Therefore, the attacker seeks to maximize the fraction of compromised IoTDs while the defender seeks to minimize it. This coupling in the players strategies and utilities naturally leads to a game-theoretic situation \[37\]. One of the most important solution concepts for noncooperative games is that of a \textit{Nash equilibrium} (NE). The NE characterizes a state at which no player \(j\) can improve its utility by changing its own strategy, given the strategy of the other player is fixed. For a noncooperative game, the NE in pure (deterministic) strategies can be defined as follows:

**Definition 1.** A pure-strategy Nash equilibrium of a noncooperative game is a vector of strategies \((\mathcal{X}^c, \mathcal{X}^d)\) \(\in \mathcal{Q}^c \times \mathcal{Q}^d\) such that \(\forall j \in \mathcal{P}\), the following holds true: \(u^j(\mathcal{X}^c, \mathcal{X}^d) \geq u^j(\mathcal{X}^c, \mathcal{X}^d)\), \(\forall \mathcal{X}^j \in \mathcal{Q}^j\), where \(-j\) is the identifier for \(j\)'s opponent.

The NE characterizes a stable game state at which the cloud cannot improve the protection of IoTD signals by unilaterally changing its action \(S\) given that the action of the attacker is fixed. Moreover, at the NE, the attacker cannot manipulate more IoTD signals by changing its action \(K\) while the cloud keeps its action \(S\) fixed. Before analyzing the NE of our game, we introduce a useful concept called \textit{dominated strategy} to use in our further analysis.

**Definition 2.** Player \(j\)'s strategy \(\mathcal{X}^j\) is weakly dominated if there exists another strategy \(\tilde{\mathcal{X}}^j \subset \mathcal{Q}^j\) such that: \(u^j(\mathcal{X}^j, \mathcal{X}^d) \leq u^j(\tilde{\mathcal{X}}^j, \mathcal{X}^d)\), \(\forall \mathcal{K} \in \mathcal{Q}^a\), with the strict inequality for at least one
\(X^{-j}\). In case the above inequality holds strictly for all \(X^{-j} \in Q^{-j}\), \(X^j\) is said to be strictly dominated (by \(X^j\)).

Essentially, a strategy is dominated if choosing it always yields a smaller utility compared to any other strategy, given all possible strategies for other players. Using this definition, prior to finding the NE, we next derive the dominated strategies of the cloud and the attacker.

**Proposition 2.** The cloud must choose as many IoTDs as possible to authenticate and, thus, any strategy \(S \in Q^c\) is weakly dominated by \(\tilde{S} \in Q^c\) if \(S \subseteq \tilde{S}\).

**Proof.** From set theory we have:

\[
S \subseteq \tilde{S}, \Rightarrow \mathcal{K} \cap S \subseteq \mathcal{K} \cap \tilde{S}, \Rightarrow \mathcal{K} - (\mathcal{K} \cap \tilde{S}) \subseteq \mathcal{K} - (\mathcal{K} \cap S),
\]

\[
\Rightarrow \{i \mid i \in \mathcal{K}, i \notin \tilde{S}\} \subseteq \{i \mid i \in \mathcal{K}, i \notin S\}.
\]

Therefore, we have:

\[
\sum_{i \in \mathcal{K}, i \notin \tilde{S}} v_i \leq \sum_{i \in \mathcal{K}, i \notin S} v_i,
\]

\[
u^c(S, \mathcal{K}) = 1 - \sum_{i \in \mathcal{K}, i \notin S} v_i \leq 1 - \sum_{i \in \mathcal{K}, i \notin \tilde{S}} v_i = \nu^c(\tilde{S}, \mathcal{K}),
\]

which proves that \(S\) is weakly dominated by \(\tilde{S}\). \(\blacksquare\)

Proposition 2 shows that the cloud must use all of its available computational resource to authenticate the IoTD signals since any strategy that is a subset of another strategy uses less computational resources.

**Proposition 3.** The attacker must use all of its \(K\) eavesdropping devices, i.e., any strategy \(\mathcal{K} \in Q^a\) is weakly dominated by \(\tilde{\mathcal{K}} \in Q^a\) if \(\mathcal{K} \subseteq \tilde{\mathcal{K}}\) and \(|\tilde{\mathcal{K}}| = K\).

**Proof.** We know that \(\mathcal{K} \subseteq \tilde{\mathcal{K}}\), and, thus, the number of IoTDs that are attacked by the adversary when choosing strategy \(\tilde{\mathcal{K}}\) is higher than the number of attacked IoTDs when choosing strategy \(\mathcal{K}\). Therefore, if the cloud chooses any strategy \(S\), then the number of unauthenticated IoTD devices for the first case will be greater than or equal to the latter case or:

\[
\{i \mid i \in \mathcal{K}, i \notin S\} \subseteq \{i \mid i \in \tilde{\mathcal{K}}, i \notin S\},
\]

and, thus, we have:

\[
u^a(\mathcal{K}, S) = \sum_{i \in \mathcal{K}, i \notin S} v_i \leq \sum_{i \in \tilde{\mathcal{K}}, i \notin S} v_i = \nu^a(\tilde{\mathcal{K}}, S).
\]
Algorithm 1 Dynamic Programming for finding the Cloud’s Non-dominated Strategies

1: Input $\mathcal{F} \triangleq \{f^1_s, \ldots, f^N_s\}$, $C/d$,
2: Initialize $M_{N+1 \times C/d - f^1_s + 1}$ everywhere False apart from $M[0, 0] = $ True,
3: Start filling all the entities of matrix $M$:
4: for $i \leftarrow 1$ to $N$ do
5: for $j \leftarrow f^1_s$ to $C/d$ do
6: $M[i, j] = M[i - 1, j] \lor M[i - 1, j - f^i_s]$, (True, if there is a subset with the values less than $f^i_s$ that sum up to $j$.)
7: for $k \leftarrow C/d$ to $f^1_s$ do
8: if $\max_m M[m, k - f^m_s]$ do (Checks the IoTD with highest $f_s$ such that the IoTDs with less $f_s$ can sum up to $k$.)
9: Initialize $\mathcal{R} = \{\}$,
10: $S \leftarrow \text{RecPath}(m, k, \mathcal{F}, M, \mathcal{R})$: (This function follows the path in $M$ until reaching to the first column.)
11: if $M[m, k]$ do
12: $\mathcal{B} \leftarrow \mathcal{R}$,
13: $\text{RecPath}(m - 1, k, \mathcal{F}, M, \mathcal{B})$
14: if $k \geq f^m_s \land M[m, k - f^m_s]$ do
15: $\mathcal{R} \leftarrow f^m_s$,
16: $\text{RecPath}(m - 1, k, \mathcal{F}, M, \mathcal{R})$
17: Output $\mathcal{R}$
18: Output $S$

Since the attacker cannot choose more than $K$ IoTDs to attack, thus any strategy $\tilde{K}$ with $K$ members, $|\tilde{K}| = K$, dominates all the strategies that are $\mathcal{K} \subseteq \tilde{K}$. ■

Proposition 3 shows that non-dominated strategies for the attacker are those that include $K$ IoTDs to attack. Here, we define $\tilde{Q}^j \subset Q^j$ as the set of player $j$’s non-dominated strategies. The attacker’s non-dominated strategies can be interpreted as combination of $K$ IoTDs from all $N$ IoTDs. Therefore, the number attacker’s strategies is $\binom{N}{K}$. Since the number of attacker’s eavesdropping devices, $K$, is comparably less than the number of IoTDs $N$, we can easily see that the complexity of finding the attacker’s non-dominated strategy set is $O(N^K)$ which is comparably smaller than considering all the subsets of $N$, that results in a complexity of $O(2^N)$. Thus, Proposition 3 reduces the complexity of the attacker’s game significantly.

For the cloud, finding all the strategies that satisfy the condition in Proposition 2 is an NP-hard problem [38]. Thus, in Algorithm 1 we propose a novel dynamic programing approach to reduce the complexity of finding the non-dominated strategies of the cloud. The algorithm takes the set of all IoTD sampling frequencies and $C/d$ as input. In this algorithm, we first define a $(N + 1) \times (C/d - f^1_s + 1)$, matrix $M$ whose element $M[i, j]$ is set to a “True” value if there is
a subset of IoTDs with a sampling frequency less than $f_s^i$ such that the summation of all IoTD sampling frequencies in this subset equals $j$.

**Corollary 1.** At each stage of computation, the Algorithm 1 uses the solutions of previous subproblems and since the operations used to fill each entity of $M$ are similar, the complexity of finding the defender’s non-dominated strategies reduces from $O(2^N)$ to $O(NC/d)$.

Even though Algorithm 1 reduces the complexity of finding the cloud’s non-dominated strategies to a linear time, the number of these non-dominated strategies is dependent on the cloud’s available resources and distribution of sampling frequencies. Fig. 5 shows the number of non-dominated strategies for the cloud and the attacker, when $N = 1000$ IoTDs, $R \in (0, 1]$, and $K \in [0.01N, 0.1N]$. In this massive IoT scenario, from Fig. 5 we can see that, the number of strategies for both players is very large which consequently requires a complex process to find the pure-strategy NE. Moreover, even though we derived the non-dominated strategies for both players, the NE is not guaranteed to exist for our game [36]. For example, consider only three IoTDs with $\{1000, 2000, 3000\}$ as their sampling frequency, assume that $C/d = 5000$, and an attacker having 1 eavesdropping device. This example game, along with the non-dominated strategies of both players, is summarized in Table I. Any element $(i, j)$ in Table I is the outcome of playing the cloud’s $i$-th and the attacker’s $j$-th strategy. From Table I, we observe that, for any outcome of the game, at least one of the players can change its strategy to gain a better payoff. Therefore, this game cannot have a pure-strategy NE for a general case. Thus, we investigate the NE in mixed strategies which is guaranteed to exist for finite noncooperative games [36]. When using mixed strategies, each player will assign a probability for playing each one of its pure strategies. For a massive IoT, the use of mixed strategies is motivated by two facts: a) the
Table I: An example of strategies and utilities for the game between the cloud and the attacker. Cloud and the attacker must randomize over their strategies in order to make it nontrivial for the opponent to guess their potential action, and b) the procedure of choosing IoTDs can be repeated over an infinite time duration and mixed strategies can capture the frequency of choosing certain strategies for both players. Thus, next, we analyze our game’s mixed-strategy NE.

B. Mixed-Strategy Nash Equilibrium

In our game, by using mixed strategies, the attacker and defender will assign probabilities for playing each one of their non-dominated strategies \([36]\). Let \(p^a\) be the vector of mixed strategies for the attacker where each element in \(p^a\) is the probability of choosing a set of IoTDs, i.e., selecting one strategy from the attacker’s strategy set \(K\). Moreover, \(p^c\) is the vector of mixed strategies for the cloud whose elements represent the probability of choosing a certain strategy from the cloud’s strategy set, \(S\). Consequently, each player must choose its own mixed-strategy to maximize its expected utility which is defined by:

\[
U^j(p^j, p^{-j}) = \sum_{S \in Q^c} \sum_{K \in Q^a} p^c(S)p^a(K)u^j(S, K), \quad \text{for } \forall j \in \mathcal{P}. \tag{21}
\]

To solve this problem, we seek to find the mixed-strategy Nash equilibrium, defined as follows:

**Definition 3.** A mixed strategy profile \(p^*\) constitutes a mixed-strategy Nash equilibrium (MSNE) if, for each player, \(j\), we have: \(U^j(p^j, p^{-j}) \geq U^j(p^j, p^{-j'})\), \(\forall p^{j'} \in \mathcal{P}^j\), where \(\mathcal{P}^j\) is the set of all probability distributions for player \(j\) over its action space \(Q^j\).

The MSNE for our game implies a state at which the cloud has chosen its optimal randomization over authenticating the signals of its IoTDs and, therefore, cannot further improve the system security by changing this randomization. Similarly, for the attacker, an MSNE is a state at which the attacker has chosen its probability distribution over the selection of IoTDs that it will attack and, thus, cannot improve its expected utility by changing its choice. Since our game is a constant-sum two-player game, the von Neumann indifference principle can be used to find a closed-form solution for the MSNE \([36]\). Under this principle, at the MSNE, the expected utilities of the players with respect to the mixed strategies played by the opponent must be equal, for every pure strategy choice. To derive the MSNE for our game, we first define an allocation
vector $\hat{\mathbf{p}}_{N \times 1}$ for each player $j$ such that each element $i$ in this vector is the probability of choosing IoTD $i$. The relationship between the allocation vector and the mixed-strategy of our game can be written as follows:

$$\hat{p}_i^c = \sum_{S \in S_i} p^c(S), \hat{p}_i^a = \sum_{K \in K_i} p^a(K),$$

where $\hat{p}_i^c$ is element $i$ of vector $\hat{p}_i^c$. We define $K_i = \{K \in \hat{Q}^a|i \in K\}$ as the set of all attacker strategies that have IoTD $i$, and $S_i = \{S \in \hat{Q}^c|i \in S\}$ as the set of all cloud strategies that have IoTD $i$. We next derive the mapping between the expected utility of each player by playing mixed-strategy vector $\hat{p}_i^j$ and the allocation vector $\hat{p}_i^j$, then we prove that our game has infinitely many, MSNEs all of which achieve a unique expected value for both attacker and defender.

**Proposition 4.** To map the mixed strategy vectors to allocation vectors the following conditions must hold true:

$$\sum_{i=1}^{N} \hat{p}_i^a = K, \sum_{i=1}^{N} \alpha_i \hat{p}_i^c = D,$$

where $D$ is the maximum number of IoTDs in a strategy $S \in \hat{Q}^c$ and, $\forall i \in N$, $\alpha_i$ is an integer.

**Proof.** First, we analyze the mapping between attacker’s allocation vector and mixed-strategy, $\hat{p}_i$. From the definition of $K_i$, we have $\sum_{K \in K_i} p^a(K) = \hat{p}_i^a$. Moreover, we know that the summation of all the attacker’s non-dominated mixed strategies equals to 1, i.e., $\sum_{K \in \hat{Q}^a} p^a(K) = 1$. In addition, since every strategy $K \in \hat{Q}^a$ has $K$ IoTDs, then, if we make a set by $K$ times repeating the set $\hat{Q}^a$, we can build each $K_i$ from this set to obtain:

$$\sum_{i=1}^{N} \hat{p}_i^a = K \sum_{K \in \hat{Q}^a} p^a(K) = K.$$  

For the cloud, the procedure is similar. From the definition of $S_i$, we have $\sum_{S \in S_i} p^c(S) = \hat{p}_i^c$ and $\sum_{S \in \hat{Q}^c} p^c(S) = 1$. Here, the number of IoTDs in each $S \in \hat{Q}^c$ is not equal, however, to build $S_i, \forall i \in N$, we must define a set with $D$ repetitions of $\hat{Q}^c$ where $D$ is the maximum number of IoTDs in a strategy $S \in \hat{Q}^c$, i.e., $D \triangleq \max_{S \in \hat{Q}^c} |S|$. Since some of the IoTDs might not be included in the strategies which consist of $D$ IoTDs, the set $S_i$ can repeat more than once for these IoTDs, and, therefore, we will have: $\sum_{i=1}^{N} \alpha_i \hat{p}_i^c = D \sum_{S \in \hat{Q}^c} p^c(K) = D$, where $\alpha_i$ is the number of times $S_i$ shows up in $D$ repetitions of set $\hat{Q}^c$. ■

Proposition 4 uncovers a linear relationship between the allocation probabilities. Using this relationship, and given that the attacker can have a successful attack on IoTD $i$ if the defender...
does not authenticate IoT Di, we can define the expected utility of the players as follows:

\[ U^a(\hat{p}^a, \hat{p}^c) = \sum_{i=1}^{N} \hat{p}^a_i (1 - \hat{p}^c_i) v_i, \quad U^c(\hat{p}^c, \hat{p}^a) = 1 - U^a(\hat{p}^a, \hat{p}^c), \]  \hspace{1cm} (25)

with the condition in (23). Next, we derive the MSNE using (23) and (25).

**Theorem 3.** The defined game between the attacker and the cloud has infinitely many MSNEs that achieve a unique expected utility, \( V^j \), for each player \( j \). These MSNEs can be derived by solving the following minimax problem:

\[ V^j = 1 - V^{-j} = \min_{\hat{p}^j} \max_{\hat{p}^i} U^j(\hat{p}^j, \hat{p}^i), \quad \text{s.t.} \quad \sum_{i=1}^{N} \hat{p}^a_i = K, \quad \sum_{i=1}^{N} \alpha_i \hat{p}^c_i = D. \]  \hspace{1cm} (26)

*Proof.* Since the game between the players is constant-sum, then the expected utility of the game at MSNE is the solution of minimax problem in (26) [36]. Moreover, the defined expected utilities and the constraints in (23) and (25) are linear functions, therefore, the minimax problem in (26) has a single solution which we call \( \hat{p}^j \), and thus the expected utility is unique, \( V^j \).

Moreover, using the mapping between the allocation vectors and mixed-strategy vectors we can find the mixed-strategies at MSNE, by solving a set of \( N \) equations in (22). However, since the number of the attacker’s and the cloud’s strategies are greater than \( N \), then solving this set of equations will result in an infinite number of solutions, i.e., infinitely many MSNEs. \qed

To solve problem (26), one must find all the non-dominated strategies of the cloud to derive values of \( D \) and \( \alpha_i \). However, as discussed before, finding all the cloud’s non-dominated strategies is challenging in massive IoT scenarios. Therefore, in a massive IoT scenario analytically deriving the MSNE by using traditional algorithms such as minimax is computationally expensive. Moreover, the cloud will need to store the player’s massive strategy set and re-run the entire steps of the conventional algorithms to reach an MSNE as the IoT system changes or a new IoT Di joins to the system. Hence, the delay during the convergence process of such algorithms may not be tolerable for massive IoT scenarios. Also, at each time step, since the cloud cannot have complete information about the unauthenticated IoT DIs due to its resource limitation which makes the convergence of conventional algorithms not suitable for finding MSNE. Therefore, we propose two learning algorithms: a) a fictitious play (FP) for a complete information game where the cloud knows all IoT DIs’ states at each time step and b) a deep reinforcement learning (DRL) algorithm that considers the cloud’s lack of information about the unauthenticated IoT DIs.
C. Fictitious Play for Complete Information

To find the allocation vector at MSNE, \( \hat{p}_j^* \), we propose a learning algorithm based on fictitious play. Since our two-player game is constant-sum, using the results of [36] and [39], we can guarantee convergence FP to an MSNE. In the proposed algorithm, each player uses its belief about the allocation vector that its opponent will adopt. This belief stems from previous observations and is updated at every iteration. Given that the FP algorithm learns the allocation vector rather than the mixed strategies, it does not require storing the set of players’ strategies thus significantly reducing the complexity of finding MSNE compared to von Neuman’s approach. Let \( \delta_j^i(t) \) be player \( j \)'s perception of the mixed strategy that \( -j \) adopts at time instant \( t \). Each element \( \delta_j^i(t) \) of \( \delta_j^i(t) \) represents the belief that \( j \) has at time \( t \), that is the probability of attacking IoTD \( i \) (for the attacker) or authenticating an IoTD \( i \) (for the defender). Such a belief can be built based on the empirical frequency with which \( j \) has chosen IoTD \( i \) in the past. Thus, let \( \eta_j^i(t) \) be the number of times that \( j \) has observed \( -j \) choosing IoTD \( i \) up to time instant \( t \). Then, \( \delta_j^i(t) \) can be calculated as follows:

\[
\delta_j^i(t) = \frac{\eta_j^i(t)}{\sum_{i=1}^{n} \eta_j^i(t)}.
\] (27)

To this end, at time instant \( t + 1 \), based on the vector of empirical probabilities, \( \delta_j^i(t) \), that it has perceived until time \( t \), each player \( j \) chooses the IoTDs that maximize its expected utility with respect to its belief about its opponent while considering both players’ constraints, that is, the attacker chooses a set \( H^a \) of \( K \) IoTDs such that:

\[
H^a(t) = \arg \max_{K} U^a(K, \delta^a(t)), \quad \text{s.t } |K| = K.
\] (28)

Meanwhile, the cloud chooses a set of IoTDs such that the resulting computational resource constraints, as follows:

\[
H^c(t) = \arg \max_{S} U^c(S, \delta^c(t)), \quad \text{s.t } \sum_{i \in H^c(t)} v_i \leq R.
\] (29)

After each player \( j \) chooses its strategy at time instant \( t + 1 \), it can update its belief as follows:

\[
\delta_j^i(t + 1) = \frac{t}{t + 1} \delta_j^i(t) + \frac{1}{t + 1} \mathbb{1}_{\{i \in H^{-j}(t)\}}.
\] (30)

This learning process proceeds until the calculated empirical frequencies converge. Convergence is achieved when:

\[
|\delta_j^i(t + 1) - \delta_j^i(t)| = \epsilon, \quad \forall i \in \mathcal{N}, \forall j \in \mathcal{P},
\]

where \( \epsilon \) is the convergence error which is a very small number.

This algorithm assumes complete information for the players since they update their strategies based on (28) and (29) at each step, i.e., this algorithm assumes that the attacker knows which
Algorithm 2 Deep Reinforcement Learning for Authentication under Incomplete Information.

1: **Input** $F \triangleq \{v_1, \ldots, v_N\}$, $R$,
2: **Initialize** $\delta^a(0)$ and $\pi(0)$,
3: **while** not converged **do**
4: The cloud chooses its strategy such that:
5: $H^c(t) = \arg \max_S U^c(S, \pi(t))$, s.t $\sum_{i \in H^c(t)} v_i \leq R$.
6: The attacker chooses its strategy such that:
7: **if** $|h^a_q(t) == 1| == k$ **do**
8: $H^a(t) = \{i||h^a_q(t) == 1\}$
9: **else** **do**
10: $H^a(t) = \bar{H}^a \cup \{i||h^a_q(t) == 1\} \}$ **where** $\bar{H}^a$ is calculated as follows:
11: **if** rand(.) $< \varepsilon$ **do**
12: $\bar{H}^a \leftarrow$ choose randomly $k - |\{i||h^a_q(t) == 1\}|$ from $\{i||h^a_q(t) == 2\}$,
13: **else** **do**
14: choose $\bar{H}^a(t)$ using (31).
15: Each player updates its own belief about its opponent using (30).
16: Check convergence,
17: **Output** $\delta^a(t)$ and $\pi(t)$.

IoTDs the cloud authenticated at the previous step, and the cloud knows which IoTDs the attacker attacked at the previous step. This can be practical in scenarios in which even the unauthenticated IoTDs can report their security status to the cloud. However, in more realistic scenarios, the cloud can only have information about the IoTDs that it has authenticated at the previous steps. Thus, next, we propose a DRL algorithm based on LSTM blocks [40] to consider the incomplete information case.

D. Deep Reinforcement Learning for Authentication under Incomplete Information

To capture the incomplete information at the cloud, we propose a DRL method using LSTM blocks as shown in Fig. 6. The architecture consists of two components: a) a deep learning-based dynamic watermarking for IoTD signal authentication as discussed in Section II and b) a DRL algorithm based on LSTM blocks to learn which IoTDs to authenticate at each step based on the attacker’s previous actions and the cloud’s computational constraints. We consider one resource allocation LSTM (RA-LSTM) block at the cloud, which receives the action stream of the attacker at the previous time steps and estimates which IoTDs to authenticate in the next step. As discussed previously, LSTM blocks are very useful in sequence to sequence mapping and future prediction based on past sequence. In our DRL algorithm, the LSTM block collects
Fig. 6: Security architecture for massive IoT scenario with incomplete information. At the cloud, DW-LSTM blocks receive the signal from IoTDs, then they authenticate the received signal if RA-LSTM triggers them by sending a value 1. Finally, DW-LSTMs send their associated IoTD’s state to the RA-LSTM.

information from \( q \) previous steps and estimates the cloud’s next action. Thus, the components of this RA-LSTM block are:

- **Input:** The RA-LSTM receives a stream \([h_1^q(t), \ldots, h_i^q(t)]\) from each DW-LSTM at the cloud where \( h_i^t(t) \) is the action of the attacker on IoTD \( i \) at step \( t - (q - l + 1) \). This action can be any value in \( \{0, 1, 2\} \) where 0 indicates no attack, 1 indicates under attack, and 2 indicates that the cloud did not authenticate this IoTD at this time instant.

- **Output:** The output of the RA-LSTM is a vector \( \pi(t + 1) \) with each component in this vector, \( \pi_i(t + 1) \), being the probability of attacking IoTD \( i \) at step \( t + 1 \).

For training this RA-LSTM, we update the action of the cloud similar to the FP approach while considering the unknown actions of the attacker as follows: If the number of input 1 at time step \( t \) is less than \( K \), i.e. \( |\{i| h_i^q(t) = 1\}| < K \), \( \forall i \in \mathcal{N} \), which means that some IoTDs were not authenticated while they were under attack, then the cloud updates its belief about the attacker’s strategy at step \( t \) using an \( \varepsilon \)-greedy algorithm. In this algorithm, with probability \( \varepsilon \), the attacker randomly chooses a set \( \hat{\mathcal{H}}^a \) with \( K - |\{i| h_i^q(t) = 1\}| \) IoTDs from \( \{i| h_i^q(t) = 2\} \), and, with probability \( 1 - \varepsilon \), the attacker chooses a set \( \hat{\mathcal{H}}^a \) such that:

\[
\hat{\mathcal{H}}^a(t) = \arg \max_{\hat{\mathcal{K}}} U^a(\hat{\mathcal{K}} \cup \{i| h_i^q(t) = 1\}) \cdot \delta^a(t), \quad \text{s.t.} \quad \hat{\mathcal{K}} \subset \{i| h_i^q(t) = 2\},
\]

(31)

where \( \varepsilon \) is a hyper-parameter which has a small value. We consider \( \varepsilon \) to be proportional to \( \frac{\sum_{l=1}^{q} h_i^l(t)=1}{kq} \) to capture the percentage of IoTDs which were not authenticated while
being under attack at the past $q$ steps. The attacker’s strategy at step $t$ will be $\mathcal{H}^a(t) = \mathcal{H}^a \cup \{i | h^q_i(t) = 1\}$. This process is summarized in Algorithm 2. Although this algorithm will not converge to an MSNE, since it does not have complete information about all the IoTDs’ state, however, since the cloud uses an LSTM method to predict the attacker’s future actions based on the interdependence of attacker’s past actions, it can choose a set of IoTDs at each step that minimizes the number of compromised IoTDs. This, in turn, will yield a desirable solution for the system under incomplete information. In Section IV, Simulation results will show that the expected utility resulting from this DRL algorithm is higher than baseline scenarios, in which the cloud authenticates IoTDs with equal probabilities or proportional to IoTDs’ values.

IV. SIMULATION RESULTS AND ANALYSIS

For our simulations, we use a real dataset from an accelerometer with sampling frequency $f_s = 1$kHz. In each simulation, we derive the optimal values for $\beta$, $n$, and $n_s$ using the method proposed in Section II such that they satisfy the reliability and delay constraints.

Fig. 7a shows the output of the LSTM trainer with $n = 25$, $n_s = 25$, and $\beta = 0.5$. From Fig. 7a, we observe that the trained output of the LSTM, which is the dynamic watermarked signal, is very close to the training target $w$. Moreover, Fig. 7b shows that the training phase converges after 269 epochs which took 23 hours using a single CPU. Here, an epoch is a measure of the number of times all the training vectors are used once to update the weights of the neural network. Since each IoTD has its own LSTM, after training the IoTDs in parallel for approximately one day, we can use and effectively authenticate the IoTDs in the system. The training error is 0.0055 which is calculated using mean squared error. Moreover, we tested the trained LSTM on another accelerometer data, and the testing error is close to 0.02 which is acceptable for our IoT application.

Fig. 8a illustrates the higher performance of LSTM compared to the static watermarking in bit extraction. From (7), we know that higher $\beta/\sigma$ results in lower bit error. We can see from Fig. 8a that the extraction error rate for LSTM is approximately one order of magnitude lower than static watermarking when $\beta/\sigma = 1$. This ratio improves for higher $\beta/\sigma$, as Fig. 8a shows that the error rate of LSTM is almost two orders of magnitude lower than static watermarking when $\beta/\sigma = 10$. This result allows designing attack detectors with lower delay, since we can choose lower $n$ for LSTM which results in smaller window size and reduces the detection delay. In addition, Fig. 8b shows how an eavesdropping attack operates against the two watermarking
(a) Comparison of IoT signal, static watermarked signal, dynamic watermarked signal, and training error.

Fig. 7: Training phase of LSTM blocks.

(b) Training performance.

Fig. 8: Dynamic watermarking LSTM performance.

(a) Bit error rate of proposed watermarking algorithms.

(b) The power ratio of key to signal in presence of an eavesdropping attack.

schemes. We can see that, in static watermarking, the attacker records the signal and by summing the recorded data of each window, increases the ratio of pseudo-noise key power to the signal power and extracts the bit stream. However, since in LSTM the bit stream dynamically changes in each window, the summation of the recorded data will not increase the ratio of the key power to the signal power. Therefore, the attacker will not be able to extract the bit stream and key from the recorded data.

To analyze the effectiveness of our proposed watermarking schemes in attack detection, we choose a static watermarking block with $n = 10$, $n_s = 10$, and $\beta = 0.5$. We also train our LSTM
block with these features. Then, we implement two types of attacks: a) a data injection attack in which the attacker starts to change the IoT device signal and b) an eavesdropping attack, in which the attacker records the data from the IoT device, extracts the bit stream, and implements an attack with the the same watermarking bit stream. In Fig. 9, the attack detectors compare the extracted bit stream with the actual hidden bit stream and the percentage of difference between these two is considered as a metric for attack detection. In other words, a high difference between the two bit streams triggers an attack detection alarm. Fig. 9 shows that, for the first attack, both watermarking schemes can detect the attack. However, for eavesdropping, static watermarking cannot detect the existence of attack while the LSTM performs well. The reason is that, in static watermarking, the bit stream is the same for all the time windows while in LSTM the watermarked bit stream dynamically changes for each time window. In addition, we can see from Fig. 9 that the delay of attack detection is 0.1 seconds since the attacks starts at 0.5s and the attack detector triggers the alarm at 0.6s. The reason is that, for a window size of $n \times n_s/f_s = 0.1$ seconds, the cloud must wait for one window to collect the data from the IoTD.

Next, to evaluate the performance of the game-theoretic framework, we consider a system with 1000 IoTDs with sampling frequency in the range of $[1000, 15000]$ Hz. In these simulations, we compare our proposed algorithm to two baseline scenarios: baseline (A) in which the cloud
chooses all the IoTDs with equal probability and baseline (B) in which the cloud chooses the IoTDs with probabilities proportional to their values. We simulate our proposed fictitious play for full information and our proposed DRL for incomplete information at the cloud.

Fig. 10 shows a simulation in which the attacker has only $K = 100$ eavesdropping devices, while for the cloud $R$ takes values in the range $[0.1, 1]$. We can see from Fig. 10 that, as the cloud’s available resources increase from 0.1 to 1, the cloud can protect more IoTDs and thus its expected utility increases from 0.96 to 1, which indicates that when $R = 1$, the cloud can protect all of the IoTDs. Fig. 10 also show that, since the proposed FP algorithm converges to the MSNE, it outperforms both the baseline scenarios. For instance, for low available resources the attacker gains up to 40% more utility which means that the attacker can compromise approximately 20 more IoTDs when the cloud chooses any one of the baseline strategies compared to the proposed fictitious play. Moreover, from Fig. 10, we can see that, for low available resources, the DRL algorithm yields approximately up to 30% improvements compared to both baseline scenarios which is equivalent to 13 less compromised IoTDs. However, it has up to 10% less expected utility than FP, that is equivalent to 7 more compromised IoTDs, which is expected due to its operation under lack of information.

Fig. 11 shows the players’ expected utility as the number of eavesdropping devices varies, for a scenario in which the available computational resources for the cloud are such that, $R = 0.5$. Fig. 11 shows that, as the number of attacker’s eavesdropping devices increases, its expected utility increases and the attacker can disrupt more IoTDs while staying undetected. Fig. 11 also
Fig. 11: The cloud’s and attacker’s expected utility as a function of attacker’s resource, $K$, for a scenario with $R = 0.5$ and 1000 IoTDs.

shows that using the proposed FP, the cloud can gain a higher expected utility than the baseline scenarios. Note that, when the attacker can attack all the IoTDs, $K = 1000$, the proposed FP algorithm has almost 5 times higher expected utility than the baseline scenarios. Furthermore, although proposed DRL has a lower expected utility than FP, it yields a 20% improvement in the expected utility compared to the baseline scenarios for the cases where the attacker can attack all of the IoTDs.

V. CONCLUSION

In this paper, we have proposed a novel deep learning method based on LSTM blocks for enabling attack detection of data injection and eavesdropping in IoT devices. We have introduced two watermarking schemes in which the IoT’s cloud, which collects the data from the IoT devices, can authenticate the reliability of received signals. We have shown that our proposed LSTM method is suitable for IoTD-cloud signal authentication due to low complexity, small delay, and high accuracy in attack detection. Moreover, we have studied the massive IoT scenario in which the cloud cannot authenticate all the IoT devices simultaneously due to computational limitations. We have proposed a game-theoretic framework to address this problem, and we have derived the mixed-strategy Nash equilibrium and studied its properties. Furthermore, we have shown that analytically deriving the equilibrium is highly complicated in massive IoT scenarios, and, thus, we have proposed two learning algorithms two address this problem: a) a fictitious play algorithm that considers complete information about all the IoTDs’ state and converges to the mixed-strategy Nash equilibrium and b) a deep reinforcement algorithm which predicts
the set of vulnerable IoTDs for the case in which the cloud cannot have information about the unauthenticated IoTDs’ state. Simulation results have shown that the proposed dynamic LSTM watermarking can detect existence of complicated attacks in which the attacker collects data from the IoT devices and designs an undetectable attack. Furthermore, simulation results have shown that the proposed learning algorithms outperform baseline scenarios in which the cloud authenticates the IoTDs with equal probabilities.

A. Proof of Theorem 2

\( \beta_i \) must be chosen such that the attacker cannot use \( w_i \) instead of \( p_i \) to extract the embedded bit. Therefore, we have to adjust \( \beta_i \) to cause a high bit error rate during the extraction of the hidden bit using the \( w \) sequence. Thus, for two different watermarked signals \( w_{i_1} \) and \( w_{i_2} \) with equal embedded bit \( b_i = 1 \), we have:

\[
\hat{b_i} = \frac{<w_{i_1}, w_{i_2}>_{n_i}}{\beta_i^2 n_i^2} = \frac{<y_{i_1} + \beta_i b_i p_i, y_{i_2} + \beta_i b_i p_i>_{n_i}}{\beta_i^2 n_i^2} = \frac{<y_{i_1}, y_{i_2}>_{n_i} + <y_{i_1}, \beta_i b_i p_i>_{n_i}, <\beta_i b_i p_i, y_{i_2}>_{n_i}, <\beta_i b_i p_i, \beta_i b_i p_i>_{n_i}}{\beta_i^2 n_i^2}
\]

\[
= Z_{i_1} + Z_{i_2} + Z_{i_3} + b_i^2, \quad (32)
\]

where \( Z_{i_1}, Z_{i_2}, \) and \( Z_{i_3} \) are Gaussian random variables with distributions \( \mathcal{N}\left(\frac{\mu_{i_1}}{\beta_i^2 n_i^2}, \frac{\sigma_{i_1}^2}{\beta_i^2 n_i^2}\right) \), \( \mathcal{N}\left(0, \frac{\sigma_{i_1}^2}{\beta_i^2 n_i^2}\right) \), \( \mathcal{N}\left(0, \frac{\sigma_{i_1}^2}{\beta_i^2 n_i^2}\right) \), respectively (the proof is analogous to Theorem 1). Then, we can write \( \hat{b_i} \) as follows:

\[
\hat{b_i} = b_i^2 + Z_{i_4} = b_i + Z_{i_4}, \quad (33)
\]

where \( Z_{i_4} \sim \mathcal{N}\left(\frac{\mu_{i_1}}{\beta_i^2 n_i^2}, \frac{\sigma_{i_1}^2 + 2\sigma_i^2}{\beta_i^2 n_i^2}\right) \). Therefore, the bit error rate incurred during the extraction of \( \hat{b_i} \) will be:

\[
\Pr\{\hat{b_i} < 0|b_i = 1\} = \frac{1}{2} \text{erfc}\left(\frac{\sqrt{2}}{\sigma_b} \cdot \frac{E(\hat{b_i})}{\sqrt{2}}\right) = \frac{1}{2} \text{erfc}\left(\frac{(1 + \frac{\mu_{i_1}}{\beta_i^2 n_i^2})\beta_i^2 n_i \sqrt{n_i}}{2(\sigma_{i_1}^2 + 2\sigma_i^2)}\right). \quad (34)
\]

Since we want to have a high bit error rate for the attacker, we can write: \( \Pr\{\hat{b_i} < 0|b_i = 1\} \geq 1 - P \). By using (34) and (A) we can find the inequality in (8). Moreover we want to have a small extraction error for cloud as in (7). Thus, we have: \( \Pr\{\hat{b_i} < 0|b_i = 1\} \leq \bar{P} \), and using (7) we can find (9). From (8) and (9), we can derive the values for \( \beta_i \) and \( n_i \) which satisfy the security and performance requirement of the proposed watermarking scheme. Now, since we know that \( \frac{n_i \sqrt{n_i}}{\beta_i} \) seconds are needed to receive all the watermarked signal, then the maximum value for \( n_{si} \) can be found by (10).
REFERENCES

[1] A. Ferdowsi and W. Saad, “Deep learning-based dynamic watermarking for secure signal authentication in the Internet of Things,” in Proc. IEEE International Conference on Communications (ICC), Kansas City, MO, USA, May 2018.

[2] Z. Dawy, W. Saad, A. Ghosh, J. G. Andrews, and E. Yaacoub, “Toward massive machine type cellular communications,” IEEE Wireless Communications, vol. 24, no. 1, pp. 120–128, February 2017.

[3] N. Abuzainab, W. Saad, C. S. Hong, and H. V. Poor, “Cognitive hierarchy theory for distributed resource allocation in the Internet of Things,” IEEE Transactions on Wireless Communications, vol. 16, no. 12, pp. 7687–7702, Dec. 2017.

[4] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, “The Internet of Things for health care: A comprehensive survey,” IEEE Access, vol. 3, pp. 678–708, 2015.

[5] H. Farhangi, “The path of the smart grid,” IEEE Power and Energy Magazine, vol. 8, no. 1, pp. 18–28, January 2010.

[6] A. Ferdowsi, W. Saad, and N. B. Mandayam, “Colonel Blotto game for secure state estimation in interdependent critical infrastructure,” arXiv preprint arXiv:1709.09768, 2017.

[7] A. Ferdowsi, U. Challita, and W. Saad, “Deep learning for reliable mobile edge analytics in intelligent transportation systems,” arXiv preprint arXiv:1712.04135, 2017.

[8] M. Mozaffari, W. Saad, M. Bennis, and M. Debbah, “Unmanned aerial vehicle with underlaid device-to-device communications: Performance and tradeoffs,” IEEE Transactions on Wireless Communications, vol. 15, no. 6, pp. 3949–3963, June 2016.

[9] L. D. Xu, W. He, and S. Li, “Internet of Things in industries: A survey,” IEEE Transactions on Industrial Informatics, vol. 10, no. 4, pp. 2233–2243, Nov 2014.

[10] S. Ali, N. Rajatheva, and W. Saad, “Fast uplink grant for machine type communications: Challenges and opportunities,” arXiv preprint arXiv:1801.04953, 2018.

[11] H. Suo, J. Wan, C. Zou, and J. Liu, “Security in the Internet of Things: A review,” in Proc. International Conference on Computer Science and Electronics Engineering, Hangzhou, China, March 2012, pp. 648–651.

[12] A. Mukherjee, “Physical-layer security in the Internet of Things: Sensing and communication confidentiality under resource constraints,” Proceedings of the IEEE, vol. 103, no. 10, pp. 1747–1761, Oct. 2015.

[13] W. Trappe, “The challenges facing physical layer security,” IEEE Communications Magazine, vol. 53, no. 6, pp. 16–20, June 2015.

[14] J. Y. Lee, W. C. Lin, and Y. H. Huang, “A lightweight authentication protocol for Internet of Things,” in Proc. International Symposium on Next-Generation Electronics (ISNE), Kwei-Shan, Taiwan, May 2014, pp. 1–2.

[15] Y. Sharaf-Dabbagh and W. Saad, “On the authentication of devices in the Internet of Things,” in Proc. IEEE 17th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), Coimbra, Portugal, June 2016, pp. 1–3.

[16] P. N. Mahalle, B. Anggorojati, N. R. Prasad, R. Prasad et al., “Identity authentication and capability based access control (IACAC) for the Internet of Things,” Journal of Cyber Security and Mobility, vol. 1, no. 4, pp. 309–348, October 2014.

[17] M. Turkcan, B. Brumen, and M. Hlbl, “A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion,” Ad Hoc Networks, vol. 20, pp. 96 – 112, 2014.

[18] I. Butun, M. Erol-Kantarci, B. Kantarci, and H. Song, “Cloud-centric multi-level authentication as a service for secure public safety device networks,” IEEE Communications Magazine, vol. 54, no. 4, pp. 47–53, April 2016.

[19] Y. Mo, S. Weerakkody, and B. Sinopoli, “Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs,” IEEE Control Systems, vol. 35, no. 1, pp. 93–109, Feb 2015.
[20] B. Satchidanandan and P. R. Kumar, “Dynamic watermarking: Active defense of networked cyber-physical systems,” Proceedings of the IEEE, vol. 105, no. 2, pp. 219–240, Feb. 2017.

[21] P. Hespanhol, M. Porter, R. Vasudevan, and A. Aswani, “Dynamic watermarking for general LTI systems,” arXiv preprint arXiv:1703.07760, 2017.

[22] M. Hosseini, T. Tanaka, and V. Gupta, “Designing optimal watermark signal for a stealthy attacker,” in Proc. European Control Conference (ECC), Aalborg, Denmark, June 2016, pp. 2258–2262.

[23] A. Dorri, S. S. Kanhere, R. Jurak, and P. Gauravaram, “Blockchain for IoT security and privacy: The case study of a smart home,” in Proc. IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Kona, HI, USA, March 2017, pp. 618–623.

[24] A. Sehgal, V. Perelman, S. Kuryla, and J. Schonwalder, “Management of resource constrained devices in the Internet of Things,” IEEE Communications Magazine, vol. 50, no. 12, pp. 144–149, December 2012.

[25] H. Sun, X. Wang, R. Buyya, and J. Su, “Cloudeyes: Cloud-based malware detection with reversible sketch for resource-constrained Internet of Things (IoT) devices,” Software: Practice and Experience, vol. 47, no. 3, pp. 421–441, 2017.

[26] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, “Secure integration of IoT and cloud computing,” Future Generation Computer Systems, vol. 78, pp. 964 – 975, January 2018.

[27] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, “Machine learning for wireless networks with artificial intelligence: A tutorial on neural networks,” arXiv preprint arXiv:1710.02913, 2017.

[28] H. S. Malvar and D. A. F. Florencio, “Improved spread spectrum: a new modulation technique for robust watermarking,” IEEE Transactions on Signal Processing, vol. 51, no. 4, pp. 898–905, Apr. 2003.

[29] J. Haitma and A. Kalker, “A highly robust audio fingerprinting system,” in Proc. of International Symposium on Music Information Retrieval (ISMIR), Paris, France, October 2002, pp. 107–115.

[30] C. Bertocnini, K. Rudd, B. Nousain, and M. Hinders, “Wavelet fingerprinting of radio-frequency identification (RFID) tags,” IEEE Transactions on Industrial Electronics, vol. 59, no. 12, pp. 4843–4850, Dec. 2012.

[31] R. E. Learned and A. S. Willsky, “A wavelet packet approach to transient signal classification,” Applied and Computational Harmonic Analysis, vol. 2, no. 3, pp. 265 – 278, July 1995.

[32] J. B. Harley, Y. Ying, J. M. F. Moura, I. J. Oppenheim, L. Sobelman, and J. H. Garrett, “Application of mellin transform features for robust ultrasonic guided wave structural health monitoring,” Burlington, VT, USA, July 2011, pp. 1551–1558.

[33] U. Challita, L. Dong, and W. Saad, “Proactive resource management in LTE-U systems: A deep learning perspective,” arXiv preprint arXiv:1702.07031, 2017.

[34] A. Graves, A. R. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,” in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, May 2013, pp. 6645–6649.

[35] J. Zhou, Z. Cao, X. Dong, and A. V. Vasilakos, “Security and privacy for cloud-based IoT: Challenges,” IEEE Communications Magazine, vol. 55, no. 1, pp. 26–33, January 2017.

[36] T. Başar and G. J. Olsder, Dynamic noncooperative game theory. SIAM, 1998.

[37] G. Bacci, S. Lasaulce, W. Saad, and L. Sanguinetti, “Game theory for networks: A tutorial on game-theoretic tools for emerging signal processing applications,” IEEE Signal Processing Magazine, vol. 33, no. 1, pp. 94–119, Jan. 2016.

[38] Pisinger, “Dynamic programming on the word RAM,” Algorithmica, vol. 35, no. 2, pp. 128–145, Feb. 2003.

[39] J. Heinrich, M. Lanctot, and D. Silver, “Fictitious self-play in extensive-form games,” in Proc. of International Conference on Machine Learning, Lille, France, 2015, pp. 805–813.

[40] U. Challita, W. Saad, and C. Bettstetter, “Cellular-connected UAVs over 5G: Deep reinforcement learning for interference management,” arXiv preprint arXiv:1801.05500, 2018.