Eavesdropper Avoidance through Adaptive Beam Management in SDR-Based MmWave Communications

Adrian Baron-Hyppolite*, Joao F. Santos*, Luiz DaSilva*, and Jacek Kibiłda*
*Commonwealth Cyber Initiative, Virginia Tech, USA, e-mail: {adrianb, joaosantos, ldasilva, jkibilda}@vt.edu

***Abstract*—High-frequency systems use beamforming to mitigate the increased path loss. As the resulting beams become highly directional, Millimeter Wave (mmWave) radios conduct a beam sweep to probe all possible angular directions to locate each other and establish communication. In this paper, we propose an adaptive beam management strategy that leverages beam sweeping to avoid eavesdroppers and other potential attackers. Our solution employs Deep Reinforcement Learning (DRL) to dynamically select a subset of beams in the transmitter codebook. We evaluate this solution through a proof-of-concept implementation using a combination of Software-Defined Radios (SDRs) and commercial mmWave equipment, and show the improvements in the secrecy capacity.**

***Index Terms*—Beam Management, Beamforming, Millimeter-Wave, Secrecy Capacity, Deep Reinforcement Learning.**

I. Introduction

The broadcast nature of the medium makes wireless systems susceptible to over-the-air attacks. These attacks range from eavesdropping intended to acquire system or user information, to jamming that introduces interference signals and reduces the legitimate link's availability, integrity, or performance. The directionality of transmission in high-frequency bands, such as in Millimeter Wave (mmWave), or sub-Terahertz (THz) spectrum, reduces the link's exposure to attacks by limiting the angular emissions [1]. Corroborating that finding, numerous studies have shown improved secrecy that scales with the directionality of transmission [2], [3]. However, the Base Station (BS) and User Equipments (UEs) must perform an initial access procedure to establish and maintain the transmission link. During initial access, BS and UEs probe, according to a pre-defined order, all angular directions, a process referred to as beam sweeping, to determine the best transmit-receive beam pair [4]. This exhaustive probing of all angular directions reduces the security of otherwise highly directional links by exposing them to adversarial attacks like eavesdropping or jamming [5]–[8].

A few recent works have focused on reducing the susceptibility of beam management to jamming attacks by optimizing the transmit power [6], randomizing the probing sequence [7], and tracking the anomalies in the power delay profile [8]. However, a natural method to reduce the exposure to jamming and eavesdropping attacks involves reducing the probing opportunities. The vast literature on reducing the probing opportunities, e.g., [9]–[16], is focused solely on improving the efficiency of the beam sweeping process. While intuitively reducing the sweeping opportunities should result in increased secrecy of directional links, to date, there has not been a proposal to quantify the secrecy during the beam sweeping process and consequently adapt the beam management to optimize security.

In our recent work [17], we developed a demonstration that showcased Software-Defined Radio (SDR)-controlled beam sweeping adaptability in the presence of user-defined sensor nodes to improve the secrecy of mmWave communications. In this article, we propose and study, leveraging the developed experimental setting, a novel adaptation strategy for beam sweeping based on Double Deep Q-Network (DDQN). The DDQN belongs to the class of Deep Reinforcement Learning (DRL) strategies, which have been shown to perform well on different beam management tasks [15], [16], [18], [19]. One advantage of DDQN over other DRL strategies is that it provides for much-needed generalization in an experimental setting [20], [21]. To optimize for security, our proposed DDQN's reward function uses the *sweep secrecy capacity*, a new metric inspired by the secrecy pressure [22], that we propose as a means to quantify the secrecy capacity of beam sweeping-aided communication.

We validate our DDQN strategy in a real-world setting, leveraging STAMINA [23], [24], an out-of-tree GNU Radio module for experimentation in mmWave communications with flexible beam training and alignment capabilities. Our proposed implementation extends STAMINA with new building blocks required to run and implement our DDQN strategy and a new experimental scenario involving different receiver locations. Our experimental results validate our DDQN strategy and illustrate the improvement in secrecy over static beam sweeping while showing sustained beam alignment accuracy.

To the best of our knowledge, this is the first work to apply DDQNs to secure beam sweeping and quantify the secrecy of mmWave systems conducting beam sweeping. Moreover, this is one of the few works involving mmWave link with beam management adaptation that showcases the developed features in an experimental setting. The rest of this article is organized as follows. Section II describes the proposed solution. Section III details the architecture of the proof-of-concept implementation. Section IV reports on experiments and numerical results, and Section V concludes this article.

II. Proposed Solution

In this section, we present our adaptive beam management strategy. As our strategy leverages DRL, we describe it using the Markov Decision Process (MDP) framework [21].

A. State and Action Space

We assume that at a discrete time t, the DRL agent maintains a representation of the current state of the environment $s_t \in \mathcal{S}$, where $\mathcal{S}$ is the set of possible states, and selects an action $\mathbf{a}_t \in \mathcal{A}(s_t)$ to be performed on the environment, where $\mathcal{A}(s_t)$ is the action set available in state s_t. Selecting $\mathbf{a}_t$ in state s_t, incurs reward r_{t+1} and moves the environment to the new state s_{t+1}. Here, $\mathbf{a}_t$ is a beam sweeping sequence, and $\mathcal{A}(s_t)$ is the set of possible beam sweeping sequences in state s_t. Moreover, each element of $\mathbf{a}_t$ is itself a vector $\mathbf{l} \in \mathcal{L}$ which represents a given beam defined for a fixed analog beamforming codebook $\mathcal{L}$, akin to practical implementations specified by the 3GPP for the 5G NR standard [25]. Accordingly, our state at time t consists of three features $s_t = (v_t, \mathbf{a}_{t-1}, \mathbf{l}_t)$, where v_t represents the average Signal-to-Noise Ratio (SNR) of the legitimate communication link at time t, $\mathbf{a}_{t-1} \in \mathcal{A}$ represents the sweeping sequence selected at time $t-1$ and $\mathbf{l}_t \in \mathcal{L}$ denotes the ID of the best beam at timestep t.

B. Reward Function

Our reward function uses what we refer to as the sweep secrecy capacity, which we propose to quantify the secrecy capacity of communication systems based on beam sweeping. The secrecy capacity quantifies the difference in bits between the capacities of the legitimate link and an eavesdropper link. In [22], the secrecy pressure was proposed as a generalization of secrecy capacity to multiple potential eavesdropper locations or even entire areas to reflect that, in practical settings, the eavesdropper location may not be known exactly. Consequently, the secrecy pressure is an average secrecy capacity where the average is taken over an entire area, representing multiple potential locations of the eavesdropper.

Inspired by the secrecy pressure, our proposed reward function re-casts the definition of secrecy capacity to scenarios where the legitimate link uses beam sweeping and a discrete number of sensors take signal-level measurements. We, thus, average the secrecy capacity over the possible beam directions and eavesdropper locations. Formally, our proposed reward function can be expressed as:

$$r_t(\mathbf{a}_t) = \frac{1}{|\mathbf{a}_t| \times |\mathcal{X}_t|} \sum_{\mathbf{l} \in \mathbf{a}_t} \sum_{x \in \mathcal{X}_t} \max\{0, \hat{C}_B(\mathbf{l}) - \hat{C}_E(x, \mathbf{l})\}, \tag{1}$$

where $\mathcal{X}_t$ is the set of the sensor (eavesdropper) locations at time t, and $\hat{C}_B(\mathbf{l})$ denotes the rate of the legitimate link for beam $\mathbf{l}$, and $\hat{C}_E(x, \mathbf{l})$ the rate of the eavesdropper link for location $x \in \mathcal{X}_t$ when beam $\mathbf{l}$ is being used at the transmitter. In our system, $\hat{C}_B(\mathbf{l})$ and $\hat{C}_E(x, \mathbf{l})$ are rate estimates based on the measured SNR at the receiver and sensor locations, respectively. The proposed reward is expressed in bits per second, and higher values indicate increased secrecy regarding the legitimate link.

C. Action Selection Strategy

Our proposed solution relies on Q-Learning, where the agent uses Q-values to determine a matching between the particular state and action pair. The Q-value is updated as follows:

$$Q_{t+1}(s_t, a_t) \leftarrow (1-\alpha) Q_t(s_t, a_t) + \alpha \Big(r_{t+1} + \gamma \max_{a \in \mathcal{A}(s_{t+1})} Q_t(s_{t+1}, a) \Big), \tag{2}$$

where $Q_{t+1}(s_t, a_t)$ represents the updated Q-value estimate, and $Q_t(s_t, a_t)$ represents the current estimate of the Q-value at the current state and action, α is the learning rate, and γ is the future discount factor.

We assume that our agent follows an ϵ-greedy strategy in selecting its action at each time step. The value of the $\epsilon \in (0, 1)$ parameter, decided at the outset of the training, determines the preference between exploration and exploitation. The ϵ-greedy strategy can be formally expressed as [26]:

$$a_t = \begin{cases} \arg\max_{a \in \mathcal{A}(s_t)} Q_t(s_t, a), & \text{with probability } 1 - \varepsilon, \\ \text{a random action}, & \text{with probability } \varepsilon. \end{cases} \tag{3}$$

D. Q-Network and Training

To update the Q-values associated with state-action pairs, we decided to employ a DDQN, as it can leverage context to arrive at effective policies, while responding well to dynamic environments [20]. We deploy the DDQN agent and its two neural networks in the transmitter. The first network (the evaluation network), whose weights at time t are denoted as θ_t, is updated every time a new batch of experiences is stored, while the second one (the target network), whose weights at time t are denoted as θ_t^-, is a copy of the original network updated less frequently to evaluate the estimated Q-values and prevent Q-value over-estimation. The DDQN update function is as follows [20]:

$$\theta_{t+1} = \theta_t + \alpha \Bigg(r_{t+1} + \gamma Q_t \left(s_{t+1}, \arg\max_{a \in \mathcal{A}(s_{t+1})} Q_t(s_{t+1}, a; \theta_t); \theta_t^- \right) - Q_t(s_t, a_t; \theta_t) \Bigg) \nabla_{\theta_t} Q_t(s_t, a_t; \theta_t). \tag{4}$$

The target network is updated according to a soft update with the weights of the evaluation network $\theta_{t+1}^- \leftarrow \tau \theta_t + (1 - \tau)\theta_t^-$, where τ represents the target network update rate.

The proposed training algorithm is presented in Alg. 1. Each training event contains up to N episodes, each comprising T steps. At each episode, the system is initialized to follow the exhaustive sweep. The proposed training algorithm uses a replay buffer $\mathcal{D}$ that stores experiences in the form of $\{s_t, \mathbf{a}_t, r_{t+1}, s_{t+1}\}$ for each time t. The evaluate network weights are updated every time the replay buffer stores a new batch of size $\delta_{\mathcal{D}}$, while the target network weights are updated when the predetermined update period, whose value is chosen so that it is updated less frequently than the evaluation network, is reached. The training terminates either after N episodes or when the temporal difference Δ_t,

the difference between the estimated and actual Q-value at a particular state, drops below a certain threshold ϵ, indicating the agent has completed learning.

Algorithm 1 Proposed DDQN Training Algorithm

```
Initialize θ_t and θ_t^- with random weights
while (n ≤ N) ∧ (Δ_t ≥ ε) do
    Initialize the system
    for t ∈ {1, ..., T} do
        Choose action a_t using ε-greedy strategy in Eq. (3) given the state s_t.
        Sweep the chosen sequence and observe reward r_{t+1} and next state s_{t+1}.
        Store the experience {s_t, a_t, r_{t+1}, s_{t+1}} in replay buffer D.
        if new batch is available then
            Collect a batch from the replay buffer.
            Update the evaluation network θ_{t+1} using Eq. (4).
        end if
        if network update period is reached then
            Update the target network θ_{t+1}^-.
        end if
    end for
end while
```

III. Implementation

To validate our adaptive beam management strategy for avoiding eavesdroppers in mmWave communications, we leveraged STAMINA, a software-defined mmWave framework for experimentation on beam sweeping, developed in our previous works [23], [24]. In this section, we detail how we extended STAMINA to calculate the sweep secrecy capacity, train DDQN models, and use them for optimizing the beam-sweeping sequences in real time to minimize transmissions in the direction of eavesdroppers.

A. STAMINA and GNU Radio

STAMINA uses a combination of SDRs and commercial mmWave front-ends to provide a flexible beam-sweep control loop in software, while interacting with physical mmWave hardware to perform experiments with directional transmissions at high frequencies. STAMINA abstracts the interaction with the mmWave front-ends and controls them to iterate over arbitrary beam sequences, collect different Key Performance Indicators (KPIs), and select the best beam pair for data transmission. We developed STAMINA in GNU Radio, a widely known SDK for prototyping on SDRs [27].

We implemented STAMINA as a collection of GNU Radio blocks that perform different aspects of the beam-sweeping control loop. While the reader can find more information about STAMINA's operation and implementation in [23], we briefly describe some of the blocks illustrated in this section:

KPI Aggregator: Aggregates the KPIs captured by other blocks, e.g., Received Signal Strength (RSS) and SNR, and labels them using the current beam pair.

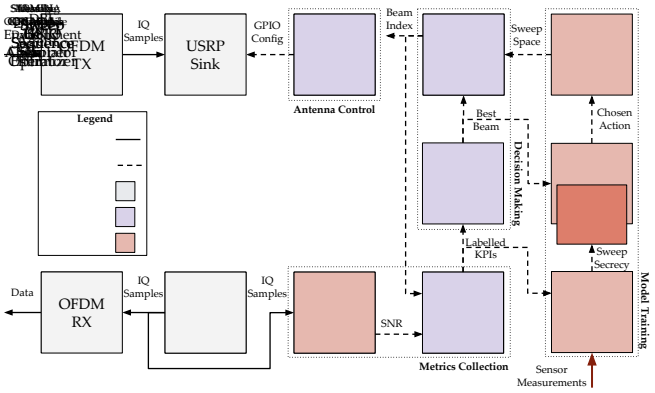

Fig. 1: STAMINA's beam sweep control loop in GNU Radio, including new blocks for training DRL models and performing adaptive beam management to avoid eavesdroppers.

Beam Selector: Parses the KPIs collected during the beam sweep and uses an arbitrary decision method, KPI, or combination of KPIs to select the best beam pair.

Sweep Sequence Iterator: Iterates over an arbitrary beam sequence and, subsequently, issues the best beam pair for data transmission.

GPIO Mapper: Converts the high-level information of a certain beam pair into the low-level parameters to configure the mmWave front-ends via their GPIO interface.

These blocks implement the control plane of the beam sweeping. For the data plane, we use vanilla GNU Radio's OFDM blocks, which allow us to transmit/receive a 5G-like PHY and USRP source/sink blocks to interface with SDRs.

B. Collecting SNR and Estimating the Sweep Secrecy

A key requirement to calculate the rates of the receiver and sensors (as discussed in Section II-B) is to measure their SNR, as per the Shannon–Hartley theorem. To do so, we created a new `SNR Calculator` block in GNU Radio, which first calculates the RSS of the received signal and then divides it by thermal noise calculated using the Johnson–Nyquist model as a function of the signal bandwidth and noise figure, which gives us the received signal's SNR. One instance of the `SNR Calculator` is part of the legitimate link's beam sweeping control loop, as shown in Fig. 1, while other instances are part of sensor nodes, which measure and report the SNR at different locations via an out-of-band channel (detailed later in Section IV).

The `KPI Aggregator` block labels the incoming SNR measurements according to the current beam pair used during the beam sweeping and forwards this information to: (*i*) the beam selector to decide the beam with the best beam for data transmission; and (*ii*) a new block, called the `Sweep Secrecy Estimator`. This new block uses the legitimate link's labeled SNR measurements and the incoming information from sensor nodes to calculate their rates and estimate the sweep secrecy capacity, using Eq. 1, as the average secrecy capacity over the possible beam directions and eavesdropper locations, which serves to train the DRL agent responsible for adapting the beam sweeping to maximize secrecy.

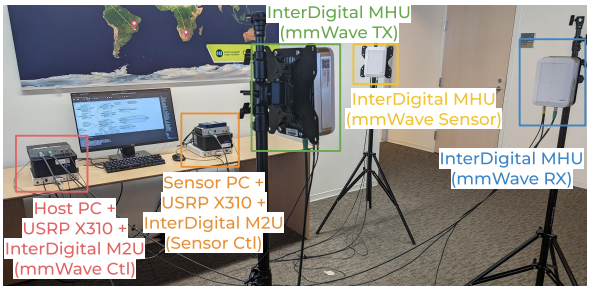

Fig. 2: Experimental setup we used to develop and evaluate our adaptive beam management strategy for avoiding eavesdroppers using a proof-of-concept implementation.

C. Embedding DRL Models

To embed our DDQN agent inside GNU Radio, we created a new block, the `DRL Environment`, which is responsible for collecting data to create the current state, as defined in Section II-A, and then feeding it to our DDQN agent.

The chosen action dictates the beam sequence for the next sweeping opportunity. The beam sequence will consist of a subset of unique beams out of the entire available codebook. `Sweep Sequence Optimizer` receives the beam with the strongest SNR from the `Beam Selector` and the beam sequence information from the `DRL Environment`. Then, it creates a sequence, i.e., a subset of the available beam codebook, centered on the beam of the strongest SNR using the sequence chosen by the DDQN agent. Finally, it outputs the sweep sequence to the `Coodebook Iterator`, which will sequentially iterate over it during the next beam sweeping procedure.

This extended version of STAMINA allows over-the-air training and evaluation of different DRL solutions for managing beams to improve communication performance and/or secrecy, using a combination of low-cost SDRs and mmWave front-ends.

IV. Evaluation

In this section, we validate the use of our DDQN-based beam management strategy for avoiding eavesdroppers in mmWave communications. First, we detail our experimental setup and the equipment used in our evaluations. Then, we examine the behavior of the sweep secrecy capacity under different spatial conditions. Then, we verify the learning and the decisions taken by our DDQN agent, and assess how our adaptive beam management strategy affects the beam sweeping accuracy.

A. Experimental Setup

To develop and evaluate our adaptive beam management strategy in STAMINA, we leveraged the radio and computational resources available for experimentation in the Commonwealth Cyber Initiative (CCI) xG Testbed, in addition to three mmWave front-ends and two mmWave controllers provided by InterDigital, creating a platform illustrated in Fig. 2. Our setup consists of a legitimate link and a sensor node, detailed below:

- The legitimate link comprises two mmWave front-ends, known as Mast Head Unit (MHU), serving as a transmitter and receiver, a mmWave control unit for controlling the front-ends, known as MHU to USRP (M2U), an USRP X310 for transmitting and receiving in Intermediate Frequency (IF), and a host PC running GNU Radio and STAMINA for baseband processing and interfacing with the mmWave equipment to perform beam sweeping.
- The sensor node comprises a single mmWave front-end, serving as a receiver to probe the environment, a mmWave control unit for controlling the front-end, a single USRP X310 for receiving in IF, and a host PC (which we refer to as sensor PC) running GNU Radio and STAMINA for baseband processing and interfacing with the mmWave equipment to collect SNR measurements to feed the host PC for estimating the eavesdropper's link capacity.

TABLE I: Our proposed DNN with four linear layers connected by three ReLU activation layers.

Layer	# Input Features	# Output Features
Linear	3	32
ReLU	-	-
Linear	32	32
ReLU	-	-
Linear	32	32
ReLU	-	-
Linear	32	3

The MHUs up/down convert signals from/to an IF in the n46 band (5.3 GHz) to/from the n257 band (28 GHz), and possess an 8×8 phased array antenna that contains a predefined codebook comprising 63 calibrated beams arranged in a 9×7 grid, ranging from $\pm 45°$ in the azimuth and $\pm 35°$ in the elevation plane. For more details about our hardware components, we refer the reader to our demonstration paper [17]. For additional details on how we interact and control the mmWave front-ends, we refer the reader to [23]. Moreover, for accessing and downloading our software package, we refer the reader to STAMINA's open-source repository (https://github.com/CCI-NextG-Testbed/gr_stamina).

For our DDQN agent, given the similarity of the problem, we adopt the neural network architecture proposed in [15]. Our Deep Neural Network (DNN) consists of 4 linear layers, with 32 neurons each, connected by three ReLU activation layers. The Table I shows a breakdown of each layer in our DNN and the associated neurons and input/output features.

To evaluate our solution, we placed mmWave front-ends 9 feet apart forming a semicircle, where we placed: (*i*) the transmitter in the center of the semicircle, facing the center of the arc; (*ii*) the receiver in different locations on the outer arc of the semicircle, always facing the transmitter; and (*iii*) the sensor node on the outer arc at a $-45°$ angle to the left of the transmitter, also facing the transmitter, as shown in Fig. 3.

We configured our adaptive beam management strategy to dynamically select between three sweeping sequences: i) 3×3 (9 beams) around the previously selected best beam, ii) 5×5 (25 beams) centered previously selected best beam, and iii) 9×7 (63 beams) spanning the entire codebook of our MHU. We chose these beam sweeping configurations for practical reasons. The action space for selecting beam sweeping sequences is effectively a powerset of the codebook

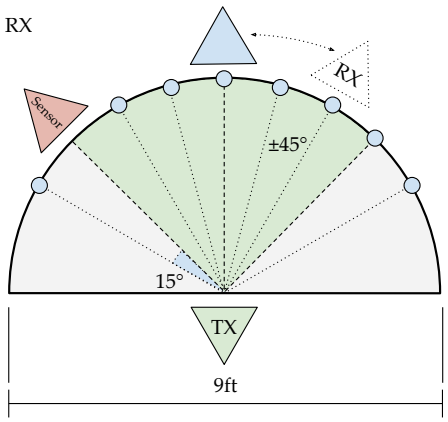

Fig. 3: Scenario used in our measurements, showing the $\pm 45°$ line of sight region in front of the transmitter (green), the placement of the sensor node at $-45°$ relative to the transmitter, and the different receiver locations (blue circles).

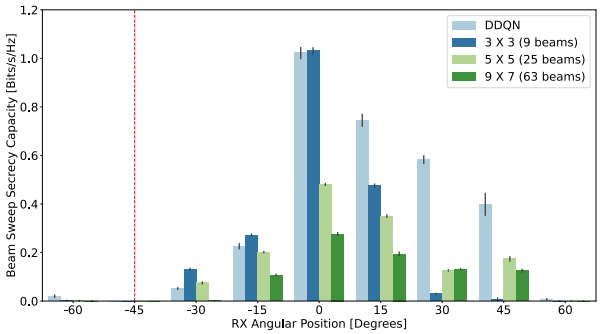

Fig. 4: Performance of the sweep secrecy capacity as we move the receiver to different positions. We can observe how the presence of the sensor node around $-45°$ compromises the security of the legitimate communication in that direction.

$\mathcal{L}$. To ensure the convergence of our algorithm in a practical setting, we must limit the available action space to sequences that exhibit different alignment and secrecy capabilities while still providing enough explanatory power.

We use this experimental setup and options of sweeping sequences in all our evaluations.

B. Assessing the Sweep Secrecy Capacity

In this analysis, we are interested in assessing the sweep secrecy capacity metric in the presence of an eavesdropper for different legitimate link configurations. Our proposed strategy is compared to three static beam sweeping sequences: i) 3×3 (9 beams) centered around the boresight, ii) 5×5 (25 beams) centered around the boresight, and iii) 9×7 (63 beams) exhaustive sweep.

Fig. 4 shows the results of our measurements. First, we observe how the sweep secrecy capacity behaves for the exhaustive beam sweep (63 beams), resulting in low secrecy essentially for all positions of the receiver with slightly better secrecy where the boresight link can be established ($0°$). Second, we observe that sweeping shorter sequences centered around the boresight is highly beneficial to secrecy, but quickly becomes inefficient as the receiver moves away from the boresight.

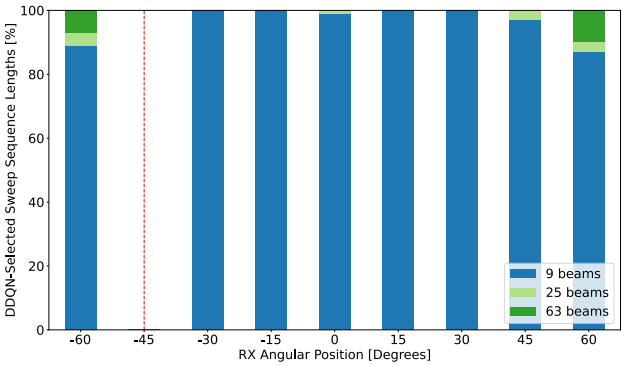

Fig. 5: The proportion of sweeping sequence lengths selected by our DDQN agent's decisions in each receiver location.

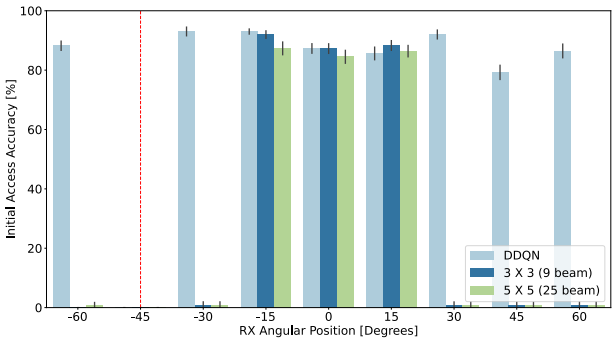

Fig. 6: Initial Access accuracy of different beam sweeping strategies for different receiver locations.

Now, our proposed DDQN strategy outperforms all the static sweeping strategies, as it is able to adapt the beam sweeping sequence relative to the orientation of the receiver while also choosing a short enough sweeping sequence to increase the secrecy of the communication. Finally, we observe, on the left-hand side of the figure, the influence of the eavesdropper sensor on the sweep secrecy capacity, as it drops dramatically in the vicinity of the sensor, indicating that in that direction, the secrecy must be optimized through additional means.

In order to corroborate the finding that the sweep secrecy capacity highly penalizes larger sweep sequences, in Fig. 5, we plot the sweep decisions of our DDQN agent when the receiver location moves on the semicircle in Fig. 2. This is also fundamental to confirming the correct training of our model and validating its operation in a dynamic environment. For each different position of the mmWave receiver, we record the length of the selected sweeping sequence for 100 beam sweeping episodes. We can observe that within the range of $\pm 45°$ degrees for the relative orientation between mmWave transmitter and receiver, our DDQN agent almost uniquely selects shorter sweeping sequences which bear substantially lower penalty in terms of the loss of secrecy. While the boresight of the receiver is still within the transmitter's codebook (see the green region in Fig. 2), the agent can track the location of the receiver by picking the shorter (9 beam) sweeping sequence around the previously determined best beam. Outside of the transmitter's codebook span, the agent can no longer track the receiver using the shorter

sequence and has to resort to exhaustive sweeps to establish and maintain communication.

Finally, we want to ensure that our proposed DDQN agent produces decisions that result in high alignment accuracy on the initial access between mmWave radios. To do so, we compared the best beam decision taken by each strategy to the best beam decision that would be produced from an exhaustive sweep. In Fig. 6, we first observe that our agent outperforms two static sweeping sequences of 3×3 (9 beams) and 5×5 (25 beams), both centered on boresight. However, the overall accuracy never exceeds 90%, which is due to the fact that in a real-life experiment, even the exhaustive sweep may not always identify the correct beam that should be used for communication. We can observe that for the region close to boresight, the different static sweeping sequences and our DDQN agent perform similarly to each other. However, as the receiver moves away from the boresight, we observe a significant drop in accuracy of the static sweep sequences which no longer allow us to track the receiver.. Conversely, our DDQN strategy iteratively tracks the direction of the highest sweep secrecy capacity (associated with high SNR) and adapts to the receiver's location.

V. Conclusion

In this paper, we proposed and experimentally validated a DDQN-based beam management solution for secure mmWave communication. We also proposed a real-life experimental implementation of the proposed solution that extends our SDR-a platform for mmWave communications. The results of experimental validation of the proposed solution showcase that our strategy improves the secrecy of the mmWave link in the presence of an eavesdropper while maintaining the beam alignment. In future works, we plan to investigate the scalability of the action space such that our agent selects an action from a much larger number of sweeping sequences while maintaining real-time capabilities. Additionally, we plan to explore different DRL design choices, such as reservoir computing, that will help us reduce our agent's training load while improving the system's generalizability.

Acknowledgements

The research leading to this paper received support from the Commonwealth Cyber Initiative, an investment in the advancement of cyber R&D, innovation, and workforce development. For more information about CCI, visit: www.cyberinitiative.org. This material is also based upon work supported by the National Science Foundation under Grants No. 2318798 and 2326599. The authors also thank InterDigital for providing them with their mmWave equipment.

References

[1] X. Chen *et al.*, "Covert Communications: A Comprehensive Survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1173–1198, 2023.

[2] L. Wang *et al.*, "Secure communication in cellular networks: The benefits of millimeter wave mobile broadband," in *Signal Processing Advances in Wireless Communications (SPAWC)*, 2014, pp. 115–119.

[3] C. Wang and H.-M. Wang, "Physical layer security in millimeter wave cellular networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 8, pp. 5569–5585, 2016.

[4] M. Giordani *et al.*, "A Tutorial on Beam Management for 3GPP NR at mmWave Frequencies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 1, pp. 173–196, 2018.

[5] B. Kim *et al.*, "Adversarial Attacks on Deep Learning Based mmWave Beam Prediction in 5G And Beyond," in *IEEE Statistical Signal Processing Workshop*, 2021, pp. 590–594.

[6] J. Zhang *et al.*, "Joint Beam Training and Data Transmission Design for Covert Millimeter-Wave Communication," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2232–2245, 2021.

[7] D. Darsena and F. Verde, "Anti-jamming beam alignment in millimeter-wave MIMO systems," *IEEE Transactions on Communications*, vol. 70, no. 8, pp. 5417–5433, Aug. 2022.

[8] J. Li *et al.*, "Secbeam: Securing mmwave beam alignment against beam-stealing attacks," *arXiv preprint arXiv:2307.00178*, 2023.

[9] T. S. Cousik *et al.*, "Deep Learning for Fast and Reliable Initial Access in AI-Driven 6G mm Wave Networks," *IEEE Transactions on Network Science and Engineering*, 2022.

[10] I. Aykin and M. Krunz, "Efficient Beam Sweeping Algorithms and Initial Access Protocols for Millimeter-wave Networks," *IEEE Transactions on Wireless Communications*, vol. 19, no. 4, pp. 2504–2514, Apr. 2020.

[11] V. S. S. Ganji *et al.*, "BeamSurfer: Minimalist Beam Management of Mobile mm-Wave Devices," *IEEE Transactions on Wireless Communications*, vol. 21, no. 11, pp. 8935–8949, Nov. 2022.

[12] C. Gan *et al.*, "Thresholded Wirtinger Flow for Fast Millimeter Wave Beam Alignment," in *Conference on Signals, Systems, and Computers (Asilomar)*, 2020, pp. 32–36.

[13] M. Polese *et al.*, "DeepBeam: Deep waveform learning for coordination-free beam management in mmWave networks," in *Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing*, 2021, pp. 61–70.

[14] H. Yan and D. Cabric, "Compressive Initial Access and Beamforming Training for Millimeter-Wave Cellular Systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 5, pp. 1151–1166, Sep. 2019.

[15] Narengerile *et al.*, "Deep Reinforcement Learning-Based Beam Training for Spatially Consistent Millimeter Wave Channels," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2021, pp. 579–584.

[16] Z. Zhang *et al.*, "Deep Reinforcement Learning Based Dynamic Beam Selection in Dual-Band Communication Systems," *IEEE Transactions on Wireless Communications*, vol. 23, no. 4, pp. 2591–2606, Apr. 2024.

[17] A. Baron-Hyppolite *et al.*, "Adaptive Beam Management for Secure mmWave Communications using Software-Defined Radios," in *IEEE Military Communications Conference (MILCOM)*, 2023, pp. 243–244.

[18] T. S. Cousik *et al.*, "Fast Initial Access with Deep Learning for Beam Prediction in 5G mmWave Networks," in *IEEE Military Communications Conference (MILCOM)*, 2021, pp. 664–669.

[19] W. Lei *et al.*, "Adaptive Beam Sweeping With Supervised Learning," *IEEE Wireless Communications Letters*, vol. 11, no. 12, pp. 2650–2654, 2022.

[20] H. van Hasselt *et al.*, "Deep Reinforcement Learning with Double Q-Learning," Mar. 2016.

[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, Massachusetts: The MIT Press, 2018.

[22] L. Mucchi *et al.*, "A New Metric for Measuring the Security of an Environment: The Secrecy Pressure," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 3416–3430, May 2017.

[23] J. F. Santos *et al.*, "STAMINA: Implementation and Evaluation of Software-Defined Millimeter Wave Initial Access," in *IEEE International Conference on Communications (ICC)*, May 2023.

[24] E. S. Fathalla *et al.*, "Beam Profiling and Beamforming Modeling for mmWave NextG Networks," in *International Conference on Computer Communications and Networks (ICCCN)*, 2023, pp. 1–10.

[25] 3rd Generation Partnership Project, "Physical Layer Procedures for Control (Rel. 17)," 3rd Generation Partnership Project, Tech. Rep. 38.213, Dec. 2021.

[26] P. Winder, *Reinforcement Learning*. O'Reilly Media, 2020.

[27] E. Blossom, "GNU Radio: Tools for Exploring the Radio Frequency Spectrum," *Linux journal*, vol. 2004, no. 122, p. 4, 2004.