Cross-Layer Attacks on Cognitive Radio Networks and Defenses for them

Jonathan Ebert Munich, Germany
jonathan.ebert@tum.de

***Abstract*—Cognitive Radio Networks are an uprising radio technology, which tries to solve the problem of frequency scarcity by allowing secondary users to utilize licensed frequencies while no primary user is sending on them. However the cross-layer nature of these networks creates novel vulnerabilities, which can be exploited by different Cross-Layer Attacks while being hard to detect. Therefore new defenses against such malicious actions need to be developed, in order to make Cognitive Radio Networks ready for real world usage.**

In this paper we give an overview of the details of multiple different Cross-Layer attacks for Cognitive Radio Networks, as well as possible defenses against them. We find similarities in the approaches of the attacks, namely that they all exploit the sensing mechanism of Secondary Users and assess the viable defense strategies for their cost in performance and benefits for the security of a network. We deduce that the most valuable improvement of the current state of Cognitive Radio Networks is the implementation of specific cross-layer protocols for these special networks.

***Index Terms*—cognitive radio networks, cross-layer attacks, wireless networks**

I. Introduction

In the last decades wireless communication established itself as an important part of our lives. Currently there are a lot of applications and services, e.g. the Global System for Mobile Communications (GSM), provided by wireless networks, with their number rapidly rising. As their security is essential, especially in their private and military uses, the security of protocols for wireless networks were researched in great detail. The growth in usage of wireless networks showed the main problems of wireless communications. Namely the limitation of resources like time, power and frequency. Especially the frequency spectrum is getting more and more scarce.

To combat this development new approaches of wireless communication are explored, in order to utilize the limited spectrum more efficiently and replace the currently widely used static spectrum allocation. One of these new approaches is Cognitive Radio Networks (CRNs). The ongoing research about CRNs made great progress in increasing their efficiency by using new algorithms and other means, but a relatively small effort was made to increase their security against attacks tailored to their own specific properties. Some attacks like the primary user emulation attack (PUE-attack) [1] were discovered and methods to defend against them or mitigate their effects were developed. Additionally many traditional attacks against wireless networks are effective against cognitive radio networks. Although the research of these vulnerabilities is valuable, the dynamic nature of CRNs as well as their cognition of their environment enables more sophisticated vulnerabilities, which can be exploited. Particularly cross-layer attacks are able to exploit the characteristics of cognitive radio networks with great effect, while being hard to detect.

In this paper, we investigate different cross-layer attacks on cognitive radio networks and explore possible strategies to defend or mitigate their effects on a cognitive radio network. We also deduce similarities between the different cross-layer attacks and assess their defenses in their cost and security benefits.

Namely we discuss:

- Lion attack
- MAC-TCP Cross-Layer attack
- Off-sensing and Route Manipulation Attack

The rest of this paper is organized as follows. Section II gives an overview of the functionality and characteristics of Cognitive Radio Networks as well as cross-layer attacks. In Section III through Section V the above listed attacks and viable defenses against them are explained respectively. Common aspects of the discussed cross-layer attacks and their defenses as well as reasons for or against the defense methods are given in Section VI. The importance of the results are discussed in Section VII, followed by the conclusion in Section VIII.

II. Background

Cognitive Radio Networks are an upcoming type of wireless networks with primary and secondary users. These types of networks try to solve the overutilization of spectrum under current spectrum management policies. Primary users are license holders of specific spectrum ranges (e.g. Television Broadcasting), which federal commissions assign, and therefore are free to use these spectrum ranges as they wish. Contrary to that, secondary users of non cognitive radio networks are only allowed to communicate over the limited unlicensed spectrum frequencies, which are rapidly filling up with users. An example for this trend would be the rise in users of the GSM. Fortunately most licensed spectrum are not constantly under use, in fact they are unused for 90% of the time [2]. To make this unused spectrum available for secondary users and thereby not waste it, the concept of cognitive radios was introduced.

These special devices are equipped with software and hardware which is able to reconfigure their communication parameters and protocols based on their current environment.

more subsections
would be nice

This enables them to communicate on licensed spectrum ranges while guaranteeing a specific performance quality for primary users. Cognitive radios can accomplish this by means of sensing the spectrum or by training a neural network to make educated guesses about current utilization of the spectrum.

Most commonly cognitive radios use the spectral-gap filling approach, which means that they only utilize licensed bands for communication as long as no primary user is using them. If a primary users starts communicating on the same spectrum the Cognitive Radio is currently using, it then chooses, with the help of the gathered data about their environment, a new not used spectrum range with the best properties for its own communications. After it found a fitting frequency the radio pauses its current work, adjusts its own communication parameters to the new frequency and then resumes its communication to other devices in the network. This process is called a frequency handoff. It is necessary to minimize the time which is needed to find a fitting frequency, in order to make this approach effective and obey the FCC regulations, which give a SU at most two seconds to detect a PUs transmission and switch the frequency. Cognitive radios also exchange information about their current environment between each other over control channels, in order to increase the performance and precision of frequency handoffs.

Cross-layer attacks are a group of network attacks, which are initialized at one layer of the OSI-model[1] but effect a different layer or which combine vulnerabilities across multiple protocol layers to attack a system [3]. This trait also makes them hard to detect with conventional detection schemes as the attacks point is not on the targeted layer and currently used defense schemes inspect each layer individually. In order to detect and mitigate cross-layer attacks, more communication between the different protocols operating on different layers needs to be implemented and one needs new cross-layer detection schemes.

III. Lion Attack

Juan Hernandez-Serrano, Olga León et al. [4] introduced a new cross-layer attack on cognitive radio networks, which they named the Lion Attack. Although an attacker does not need to be part of the network, the attack can increase the percentage of inactive time of the transport layer up to 98.13% [5]. In order to understand the Lion attack one needs to understand the PUE-attack on which it is based and how the TCP congestion control works and under which circumstances it is initiated.

A. PUE-attack

As a cognitive radio needs to distinguish between primary and secondary users, so that a frequency handoff can be performed when a primary user starts sending on the currently used channel, a spectrum sensing scheme has to be used. We will assume such a device uses a simple energy-detection based transmitter verification scheme. Under such circumstances the device is only able to recognize the signals of other secondary users and will categorize any other signals as primary user transmissions. Therefore for the device any unrecognized signal originates from a primary user and a frequency handoff is going to be performed. An attacker may exploit this behavior in order to block the channel for secondary users or to force all secondary users, which are currently using the channel, to perform a frequency handoff. Under such circumstances an attacker may simply send random data on the channel, in order to perform a PUE-attack. It is also possible to execute this attack on networks with cognitive radios utilizing spectrum sensing schemes, which are able to analyze the specific properties of primary users. E.g. commercial available TV UHF transmitters may be used in order to send primary user signals on the channel, which are falsely classified as valid primary user signals by the cognitive radios [1].

B. TCP congestion control

After a package was sent out via TCP a retransmission timer is initialized with a value based on the Round-Trip Time (RTT) for the established connection. A sent out package is considered to be lost if the retransmission timer expires, while no matching acknowledgment has been received. If a package is lost a Retransmission Time Out (RTO) starts. The sender can not send out any additional packages or start retransmissions until the RTO expires. As TCP assumes that the cause for the lost of the package is congestion in the network, it retransmits the original package after the RTO and reduces the congestion window to the size of one segment. The size of the congestion window determines the number of bytes that can be sent out on a connection without waiting for the acknowledgment of past messages, therefore a reduction of the congestion window to the size of one segment also reduces the throughput of the TCP connection. Additionally the congestion control backs off[2] the retransmission timer each time a transmission is unsuccessful, further increasing the time the sender remains inactive.

C. Performing the Lion attack

When an attacker is performing the Lion attack, he exploits the missing communication between the physical layer and TCP on the transport layer. If the attacker performs a PUE-attack in a cognitive radio network, all secondary users on the attacked channel are forced to perform an frequency handoff. Since TCP is not aware that the handoff is in progress, it continues to hand down packages to the physical layer, which are then queued. Since in most cases the retransmission timer is smaller than the time it takes to perform a frequency handoff, TCP assumes the package is lost and starts its congestion control, therefore decreasing its throughput. This attack can be extended to a Denial of Service (DoS), if the attacker can predict or know the frequency to which the secondary user switches and then repeat the attack on the new frequency. This can easily be known if the CRN uses insecure control channels or the attacker knows the state of the network and

[1] In most cases the physical or MAC layer

[2] backing off here means doubling the retransmission timer

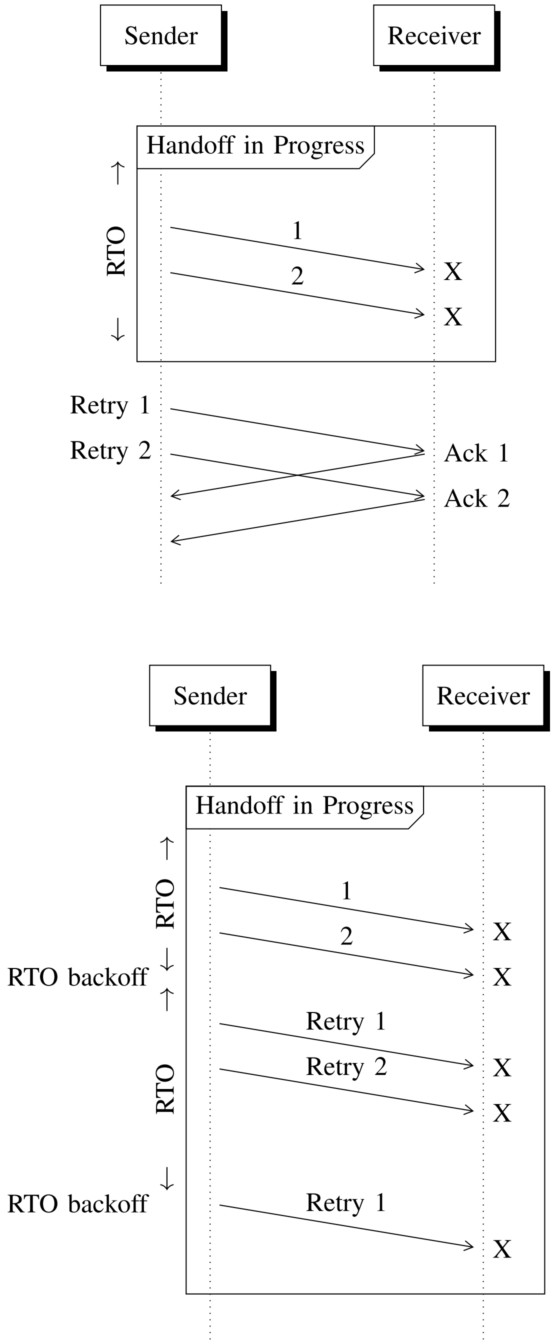

Fig. 1. Frequency Handoff effect on TCP

can therefore predict suitable channels for the victim to switch to.

Figure 1 shows the effects of a spectrum handoff in the TCP. In the first diagram a sender sends two segments to a receiver during a frequency handoff. Because of the handoff the two segments are never sent out and the sender perceives them as lost after the RTT has run out and the RTO backs off. Fortunately the frequency handoffs finishes before the retransmission is started, so that the retransmissions can be successfully sent out. The second diagram shows a to a DOS extended Lion attack from the point of view of the TCP layer. Here the same or a new handoff is in progress after the RTT for the first two packages has ran out. Therefore the retransmissions are also perceived as lost and the RTO will be backed off another time, further interrupting the communication. One should especially notice how the RTO increases in length with every new failed transmission.

D. *Mitigating the Lion attack*

Mainly the vulnerability which the Lion attack exploits is the missing cross-layer communication between the transport layer and the physical layer. As TCP is not aware when a frequency handoff is in progress, it will detect a connection loss and starts its congestion control. Therefore it is necessary to give more information from the physical layer to TCP. Hernandez-Serrano, León et al. [4] suggested a new TCP version based on Freeze-TCP. In this suggested variation of TCP the receiver monitors the signal strength and tries to predict disconnects. If a disconnects is expected by the receiver it sends a Zero Window Probe (ZWP) to the sender before the disconnect occurs. Upon receiving a ZWP the sender is not able to make changes to its transmission parameters, excluding parameters regarding the next available band to switch to. Also the sender can not send out new packages, except window probes. As the sender can not start new transmissions or retransmissions, the congestion control is not initiated and the congestion window remains at a sensible size. When the receiver detects that the connection is resumable, it advertises a non-zero window. Upon receiving the non-zero window the sender returns to normal operations. If the nodes of a cognitive radio network share their information, the receiver does not need to advertise the ZWP, as the sender should already be aware of the incoming handoff and start ’freezing’ its parameters. Even if the named changes are implemented on a CRN, an attacker may still DoS the network with the Lion attack by continuously forcing frequency handoffs.

If the attacker utilizes local spectrum sensing to find out the new frequency, one can utilize a wider part of the unlicensed spectrum. This will not prevent local spectrum sensing but it will make a local probe of the network differ more from the global. Therefore the chance that the attacker predicts the same new frequency as the cognitive radio will decrease. Obviously this mitigation is only viable more unlicensed frequencies are available.

Now the attacker may fall back to eavesdropping the normally unsecured control channel in order to get the required information to continue his attack. To defend against such sniffing, one needs to use a way of encrypting the control data, while keeping it accessible for all network members as well as making it possible for members to authenticate themselves to the network. As the nodes of the network are already sharing information with each other, a efficient and simple solution would be to use a group key. A group key would provide all necessary function for secure communication in the network, so a group key management protocol has to be implemented for CRNs.

All aforementioned solutions only mitigate the Lion attack as they can not handle the DoS or the channel degradation inflicted by the attack. In order to actually defend against the

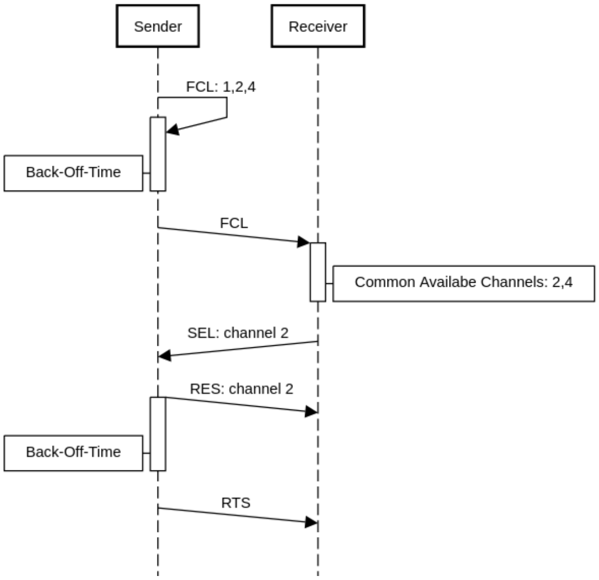

Fig. 2. Channel Negotiation of Cognitive Radios

attack the authors of the paper suggest using an intrusion detection system (IDS). Such an IDS needs to have two features so that it can be effectively used in CRNs: distributed traffic monitoring over the network in a cooperative way and inter-layer interaction to detect cross-layer attacks. As the two techniques for state of the art IDS, misuse and anomaly detection, are not fitting for CRNs, the authors note that a new IDS schemes for cognitive radio networks are needed. These new scheme needs to have cross-layer interaction, a reputation system and a global analysis of the live monitoring mechanisms. The cross-layer interaction additionally needs to define a way to collect information from different layers for a node in order to profile his behavior and therefore make it possible for the IDS to detect cross-layer attacks with a lower percentage of false positives. The reputation systems prevents malicious nodes to forge information to disrupt the networks collaborative way of working and the global analysis of the live monitoring makes sure that nodes which are not cooperating are assigned a lower trust level.

IV. MAC-TCP Cross-Layer Attack

Dileep Nagireddygari and Johnson Thomas [6] identified a new cross-layer attack for CRNs. Simulations showed that the attack can let the TCP throughput rate of the attacked node decrease by about 40% of the original rate. The attack is similar to the Lion attack but differs in one major aspect, namely the layer on which the attack is launched. While the Lion attack is executed on the physical layer the MAC-TCP cross-layer attack executes at the MAC-layer like the name suggests.

A. Channel Negotiation in MAC-layer

As the attack exploits the distributed channel negotiation used by CRNs in the MAC-layer, a sufficient understanding of the negotiation process is needed. In order to prevent two transmissions from occurring on the same channel and disrupting each other, a sender needs to reserve a channel before starting its transmission. First the sender identifies free spectrum bands and allots them into its Free Channel List (FCL), which then is send to the receiver after the back-off time has run out. The neighbors of the sender defer from transmitting on the channels by updating their network allocation vector with the header of the FCL Frame. The receiver computes common available channels with the help of the received FCL and then sends a Selection Frame (SEL) to the sender with one of the free channels as a data channel. If the sender agrees with the SEL Frame it informs its neighbors with a Reservation Frame (RES) about the selected channel and then reserves the channel with a Request-To-Send Frame (RTS) after the back-off period which is selected from $[0, CW)$ (CW being the congestion window size). Malicious nodes may use a small CW size to reserve a channel first and therefore prevent others from using the channel.

Figure 2 shows the channel negotiation between two SU in a CRN. The Sender wants to communicate with another node in the network namely the Receiver. Therefore he first checks his environment for free channels and finds out that channels 1,2 and 4 are free and saves them into his FCL. Now he sends his FCL to the Receiver, who compares it with his own FCL and finds that channel 2 and 4 are available to both nodes. Then the Receiver chooses channel 2 for the communication, as it has the best performance out of the two, and notifies the Sender about his selection with the SEL frame. After he received the SEL frame, the Sender accepts channel 2 for the communication and informs all neighbors (including the Receiver) that he will use that channel now with the RES frame. When the back-off time has ran out, the Sender now sends out a RTS and upon receiving the CTS from the Receiver the communication starts.

B. TCP Spectrum Sensing

In order to detect the presence of primary users, secondary users in CRNs need to switch between spectrum sensing and transmission mode in TCP. As the node can not receive any packages while it is in spectrum sensing mode and will receive a number of Acknowledgment Frames (ACK) upon returning to transmission mode, a variation in the RTT will occur. Therefore if the spectrum sensing duration combined with the RTT is greater than the RTO, the RTO timer will expire and the TCP congestion control is started.

C. Performing the MAC-TCP Cross-Layer attack

As the Lion attack the MAC-TCP Cross-Layer attack aims to degrade the TCP layers throughput by exploiting the TCP congestion control. In this attack the adversary preempts periodically the wireless channels in his range in a round-robin manner. This can be accomplished by e.g. manipulating its own back-off mechanism. After the attacker preempts a channel, the cognitive radios currently operating on this channel, will change to spectrum sensing mode, further increasing the RTT. After the attacker diverts to preempting another channel, the cognitive radio will switch back to transmission mode and send his possible communication partners an ACK frame. If the RTO timer of his communication partners has ran out before the ACK frame was received, their congestion control will be started and the TCP throughput of the connection will decrease. The attacker can repeat the attack as he wishes.

D. *Mitigating The MAC-TCP Cross-Layer attack*

As the attacker needs to modify MAC-frames in order to perform the attack, some sort of authentication in the MAC layer is needed to mitigate the attack. Since standard client/server key management protocols are not suited for CRNs, as there is not a trusted server available. Dileep Nagireddygari and Johnson Thomas [6] suggested the use of a deterministic key pre distribution algorithm. In order to create the initial key chain for the individual nodes, they decided to utilize the Chinese Remainder Theorem. These key chains need to be distributed to the nodes prior the network deployment. Now if two nodes want to communicate they first search in their key chain for common keys. If they have a common key they start their communication normally. But when they do not have a common key, the nodes start checking their neighboring nodes for common keys and then relay the message over the neighbor if they have a common key. This process is repeated until the nodes which want to communicate have found a key path. Additionally the nodes build a sorted channel list, which includes all channels in the network sorted in such a way that the most common channel is first, and a common channel list, which includes the channels common to a communication pair sorted after the rarity of a channel. The nodes use these lists in order to make the process of finding a key path more efficient. If an attacker now launches the MAC-TCP Cross-Layer attack, the attacked intermediate node can not send an ACK before its timeout. With the new information the source node now can know identify the attacked node and reroute the communication path around it.

The simulations of Nagireddygari and Thomas have shown that, although a slight decrease of the throughput can be noticed directly after the execution of the attack, the throughput returns to its original rate in under 0.5 seconds, if the defensive measures they proposed were taken.

V. Off-sensing and Route Manipulation Attack

Moinul Hossain and Jiang Xie [7] found a cross-layer attack in CRNs, which can manipulate the traffic flow of a CRN by attacking the physical layer. In detail this attack is a combination of multiple Off-sensing DoS attacks (OS-DoS), which when coordinated correctly are able to change the routing of a CRN. In simulations a 50% increase in traffic flow to the target node is noticed. If the targeted node is compromised by the attacker, the increased traffic sent through it can be exploited e.g. to initiate a blackhole attack. The OS-DoS exploits the fact that CRs need to periodically sense for new or existing PU transmissions to use the licensed frequencies without disturbing the communications of PUs.

In more detail an attacker will interfere on the currently used band only when the SU is transmitting. As now two nodes are sending on the same frequency simultaneously, the throughput of both will be decreased. Furthermore the SU accounts this decrease in throughput to himself interfering with the transmission of a PU and therefore will initiate a frequency handoff in at least the next 2 seconds to not harm the FCC regulations. Therefore the channel availability of the targeted node will be reduced. As state of the art routing protocols optimize their decisions heavily by the spectrum availability, repeated attacks create a DoS for the targeted node as neighboring nodes redirect their traffic through other nodes, effectively changing the decisions of the routing layer. If an attacker chooses the right nodes, he can therefore direct the traffic of a CRN to any node in it.

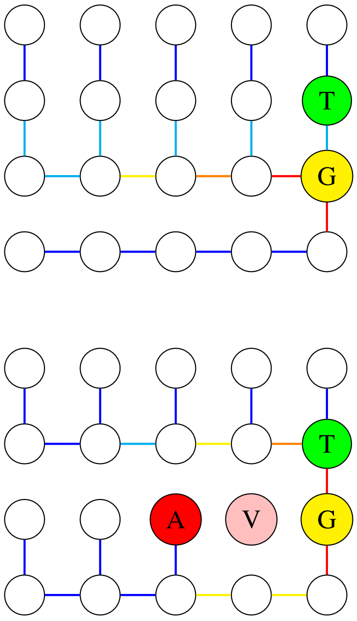

Fig. 3. Traffic heat map with and without OS-RM attack [7]
traffic intensity from low to high: blue, cyan, yellow, orange, red
G: Gateway; T: Target; A: Attacker; V: Victim

Figure 3 illustrates how the routing of the traffic of a network can be manipulated by the OS-RM attack. The first graph shows the traffic and routes of a CRN with is not under attack. It is noticeable that most of the traffic of the network is handled by the two neighbors of the Gateway and only a small amount of traffic flows through the target note. The second graph depicts the same CRN in which now node V is attacked by the malicious node A with a OS-DoS attack. As the attack is successful, V is now effectively separated from all other nodes and new routes are used. Especially the whole upper two columns nodes are now routing their packages to the gateway through node T, thus increasing the traffic load T needs to handle.

Theoretically the attacker needs to know the channel hopping sequence of his target to perform a OS-DoS. As in practice most attackers do not have access to such information, a specific strategy namely the random-OS-DoS [8] is used. When using this strategy multiple coordinated attackers hop semi-randomly around the channels each time slot, until one detects a starting transmission of a victim by listening to the Request-to-Send/Clear-to-Send (RTS/CTS) messages sent out on the current channel. If a transmission of a victim is detected, the OS-DoS is performed. In order to maximize the chances of encountering a victim multiple strategies are used. The attackers search for victims on non-overlapping channels to increase the count of channels observed each slot. Also the attackers generate a new channel hopping sequence each time an attack was successful, however the channels on which the attack was successful are appended to the end of the sequence, as the victim can not stay on this channel. Obviously the effectiveness of the random-OS-DoS increases with the number of attackers, as it will increase the number of channels observed each time slot.

The attack utilizing the OS-DoS in such a way is called a off-sensing and route manipulation attack (OS-RM). In detail the attack can structured in three parts, the node selection for the OS-DoS, the prediction of the state of the channels and the estimation of channel parameters with the help of a Hidden Markov Model (HMM). In the following sections we explain these three task, which are needed to perform a OS-RM in a CRN.

A. *OS-DoS Node Selection*

In order to maximize the performance of the OS-RM the attacker needs to carefully choose the right targets for the OS-DoS. Hence the attacker tries to find neighbors to attack with the OS-DoS with the goal to route the network traffic to the target node of the OS-RM. To archive this goal an attacker can use a shortest-path algorithm e.g. the Dijkstra-Algorithm. Thus he first calculates the number of nodes, which route their traffic to the target node under no attack. Then he checks for each neighbor if the number of nodes routing their traffic through the target node increases, when he attacks the neighbor. If that is the case he will start attacking the neighboring node or if not he will wait for the next network update.

B. *Channel State Prediction*

As the attack needs the current network graph, a method to predict it is needed. One possible approach to predict the network state is based on the routing updates sent out through the network. The updates enable the attacker to make predictions about the channel availability and estimate their link costs before the next routing update arrives. As the details of the prediction model are too detailed for the scope of this paper, we will only give a rough overview of its functionality. The model is based on two criteria for deciding if a channel is active or inactive. Firstly a probability that a given channel is idle or busy and secondly the expected length of the activity or inactivity. This information enables the attacker to calculate the probability of a channel being active. If the probability is greater than the threshold given by the predictor model the channel is considered to be active. Additionally this method requires the knowledge of the channel statistics to calculate the probability, which are hard to get. Thus a Hidden Markov Model (HMM) is used to estimate these statistics.

C. *Hidden Markov Model*

In order to estimate the channel statistics a specific kind of slotted discrete time Markov Model, the Hidden Markov Model, is used. HMMs consist of two states: a hidden state X which follows a normal Markov Model but is not observable and Z which is observable and generates symbols based on the hidden state X. The states of process X represent the real states of PU transmissions and the states of Z represent the routing updates. The iterative Baum-Welch algorithm is used to estimate the needed parameters for the HMM, which gives the maximized probability of observing a specific sequence in Z under the given parameter e.g. the probability which a PU sends on an unoccupied channel. Lastly the attacker only needs to calculate the state probability matrix given by the Baum-Welch algorithm to a transition rate matrix and then map it into the transition probability matrix to get the channel statistics.

D. *Mitigating the OS-RM attack*

To defend against the OS-RM attack one needs to mitigate the frontend of the cross-layer attack namely the random-OS-DoS attack. Moinul Hossain and Jiang Xie [8] proposed a detection technique based on Q-Learning to detect occurring OS-DoS attacks and a fitting strategy to avoid the impacts of OS-DoS attacks based on a Markov Game when an attack is detected. This approach has the advantage that it considers the absence of attackers and therefore avoids using unnecessary defense measures, hence reducing the network performance.

First we will describe the defense strategy. As already mentioned the attack and defense problem is modeled as a Markov Game which we will then use to develop a single-agent defense method for the random-OS-DoS attack. The game will include three actions: *stay*, *hop* and *extra-sense*. The actions *stay* and *hop* represent the decisions to stay on the current channel or perform a frequency handoff and hop to a new channel accordingly. When performing the *extra-sense* action, a SU tries to detect attackers by fine-sensing the current channel instead of transmitting. Fine-sensing enables the SU to differentiate between PU and an attacker. Additionally the attacker is included as the environment in the Markov game, hence reducing it to single-agent game for the victim SU. With the help of the defined Markov Game it is now possible to deduce an optimal policy, which maximizes the possible result for the SU for each action at each state of the game. As the complete details of the game are out of the scope of this paper we will now present the resulting strategy to follow for a targeted SU in order to minimize the effects of a random-OS-DoS attack. The SU uses a underutilized channel for k time slots and then performs a frequency handoff; if g successive transmissions fail the SU takes the *extra-sense* action. k and g are thereby determined by the network parameters. The defense strategy can hence be described as a game of 'hide and seek' with the attackers, trying to escape from the multiple attackers of the random-OS-DoS attack to different channels.

As we now defined a way to defend against the random-OS-DoS attack, we now need to get the network parameters and find a way to detect the presence of an attacker. As it is impossible for the victim to know the exact network parameters for the Markov Game, we need to use a learning system for the SU to learn the game. As it can learn without the transition probabilities and adapt fast to sudden changes in the Markov Game, a Q-Learning based approach is well fitting. Q-Learning approximates the unknown transition probabilities with the help of empirical distribution of already experienced states by iteratively calculating Q-values for the state-action transition tuples. As Q-Learning is not policy based in the learning process, the agents take random actions with the goal of taking the action with the highest Q-value, hence reducing the randomness The learning process ends when the Q-value converges. Since the agent makes errors while learning, a decrease in performance is noticeable but as the defender learns the rate of mistakes and therefore the decrease in performance reduces.

VI. Common Aspects of Cross-Layer Attacks and their Defenses

All three discussed attacks exploit the PU detection mechanism of CRNs. While the attack point is the same the concrete method differs from each attack. The Lion attack uses imitation with the PUE-attack to ultimately decrease the TCP throughput, while the MAC-TCP Cross-Layer Attack simply periodically preempts the channel. The in this regards more sophisticated Off-Sensing Attack on the other hand directly interferes with the transmissions of SU to force a reduction in throughput. As also the primary goal of the MAC-TCP Cross-Layer Attack and the Lion Attack are the same,therefore they can be grouped together as Cross-Layer Attacks for Cognitive Radio Networks attacking the sensing mechanism of SUs with the goal of decreasing the throughput of the TCP layer. Although one notable difference between these two attacks is that the attacker needs to be part of the CRN in order to perform a MAC-TCP Cross-Layer Attack, while it is not required for the execution of the Lion Attack. The Off-sensing and Route Manipulation Attack differs in the goal as it reroutes the traffic of a network through a node and thus is more useful as a starting point for following attacks on the network.

The defense or mitigation methods against the discussed attacks differ heavily in kind an effort to integrate them into a network. For one mitigating the Lion Attack implies making major changes to the network or the protocols used in it, which brings a decrease in performance in the whole network. Hence one needs to assess if the network could be a valid target for an attack and set up defensive measures accordingly. Though one needs to keep in mind that some proposed defense mechanism like the usage of the proposed new TCP variant could also mitigate similar attacks targeting the TCP congestion control like the MAC-TCP Cross-Layer Attack. Also enforcing the encryption of the control communication in CRNs would be a sane new standard, which does not only defend against existing attacks but would also harden the network against new kind of attacks of similar nature. Regarding the Off-sensing attack even as the defense mechanisms are only activated upon detection of an attack, it would still be advisable to implement them only if there is a measurable danger of being targeted by such an attack, as the underlying learning model may impact the whole network in unexpected ways and therefore may be the source of new problems in the network. The defense against all attacks would also heavily profit from new cross-layer mechanism between the network layers, as their impact on the network and their property of being hard to detect all originate from the fact that the upper layer protocols do not have information about the current status of lower protocols.

To summarize these attacks are best categorized by their attack point. The different defense mechanisms are still best grouped by their normal categories (e.g. encryption, IDS etc.) as they do not target the cross-layer interaction of the attacks but defend against the frontend attack of the cross-layer attacks. Additionally a new kind of protocols needs to be designed, which would better integrate into the cross-layer nature of CRNs.

VII. Discussion

As cognitive radio networks currently are rising in popularity, the interest of malicious parties to exploit them is also rising. Therefore it is important to get an overview of the different currently identified weaknesses in their designs in order to find or implement new methods to secure these networks against them. To better identify valid defenses we compared three noteworthy exploits and described how they managed to abuse the current designs to perform malicious activities on a CRN. These observations are important as in CRNs the different network layers depend more heavily on each other without having the knowledge of each others state. This shortcoming of the current protocol designs can be especially acknowledged, when observing the exploitation of the TCP congestion control in the Lion or MAC-TCP Cross-Layer attack, as the TCP protocol misinterprets the policy-abiding behavior of the lower layers as congestion in the network.

We also assessed the different defense methods regarding their benefits for securing the network and costs, which helps to make decisions for production network deployment.

We restricted the research on the theoretical concepts behind these attacks and their effectiveness in simulations, as well as already known defensive measures against them.

VIII. Conclusion

Cognitive Radio Networks have risen as a solution to the scarcity of radio spectrum. The members of these networks are equipped with hardware and software, which enables them to access the best spectrum bands for their communication, licensed or unlicensed, without harming the PUs on the licensed bands.

In this paper, we discussed multiple cross-layer attacks on CRNs, as well as viable defenses against them. We described how the different attacks archive their malicious cross-layer interaction and which effects it has on the targeted network. All of the discussed attacks exploited the sensing mechanisms of CRs, hence a further investigation of the concrete policies for SUs and possible changes to them could provide valuable improvements to the currently defined ones.

We noted that most of these attacks only work because there is a lack of inter-layer communications between the protocols (most notable the lack of communication between lower layers and the TCP congestion control) as well as a lack of cross-layer defense mechanisms. Although some progress was already made in the regards of finding specific methods for securing CRNs, it is largely of theoretical nature. Thus more research and especially development in the topic of defensive methods for CRNs is needed. As we outlined it is critical that new or modified protocols for CRNs, with cross-layer interaction are designed to make CRNs a viable solution to the frequency shortage problem. Possible starting points for this research are the concretion of IDS for CRNs as well as methods to encrypt the control channels and authenticate the members of the network. We believe that although there are still some security challenges to overcome, that CRNs would provide great benefits of the modern wireless communication landscape.

References

[1] R. Chen, J.-M. Park, and J. H. Reed, "Defense against primary user emulation attacks in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 26, no. 1, pp. 25–37, 2008.

[2] N. Devroye, M. Vu, and V. Tarokh, "Cognitive radio networks," *IEEE Signal Processing Magazine*, vol. 25, no. 6, pp. 12–23, 2008.

[3] A. Klein, "Cross layer attacks and how to use them (for dns cache poisoning, device tracking and more)," in *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 1179–1196, 2021.

[4] O. Leon, J. Hernandez-Serrano, and M. Soriano, "A new cross-layer attack to tcp in cognitive radio networks," in *2009 Second International Workshop on Cross Layer Design*, pp. 1–5, 2009.

[5] J. Hernandez-Serrano, O. León, and M. Soriano, "Modeling the lion attack in cognitive radio networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, pp. 1–10, 2011.

[6] D. Nagireddygari and J. P. Thomas, "Mac-tcp cross-layer attack and its defense in cognitive radio networks," in *Proceedings of the 10th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, Q2SWinet '14, (New York, NY, USA), p. 71–78, Association for Computing Machinery, 2014.

[7] M. Hossain and J. Xie, "Off-sensing and route manipulation attack: A cross-layer attack in cognitive radio based wireless mesh networks," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, pp. 1376–1384, 2018.

[8] M. Hossain and J. Xie, "Hide and seek: A markov-based defense strategy against off-sensing attack in cognitive radio networks," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 3028–3041, 2020.