You Can Jam But You Can't Hide: Defending Against Jamming Attacks for Geo-location Database Driven Spectrum Sharing

Haojin Zhu, *Senior Member, IEEE,* Chenliaohui Fang, Yao Liu, *Member, IEEE,* Cailian Chen, *Member, IEEE,* Mengyuan Li, and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—The emerging paradigm for dynamic spectrum sharing is based on allowing secondary users (SUs) to exploit white space frequencies that are not occupied by primary users (PUs). White space database provides an opportunity for SUs to obtain spectrum availability information (SAI) by submitting a location based query. However, this new paradigm can also be exploited by the attackers to significantly enhance their jamming capability due to the available channel information from spectrum queries, which is expected to block SUs at an increased successful rate. The challenge is that the unique characteristics (e.g., lack of the wide range frequencies or continuous broadband) make existing anti-jamming techniques (e.g., Direct-Sequence Spread Spectrum and Frequency Hopping Spread Spectrum) difficult to be applied. In this paper, we present a novel Jammer Inference based Jamming Defense (jDefender) Framework. The main idea of jDefender is inferring the likelihood of a user being a jammer based on the observed jamming events and then utilizing the inferred attack likelihood to enhance the effectiveness of a series of proposed anti-jamming strategies. In specific, we first propose Channel Allocation based Jammer Inference (CAJI) scheme to infer the likelihood of an SU being jammer based on the channels occupied by SUs even under the collusion attack performed by multiple jammers. The strength of the anti-jamming strategies (e.g., puzzle difficulties, available spectrum resources) will be correlated with the possibility of an SU being jammer to achieve the tradeoff between system performance and jamming tolerance. We then implement the proposed scheme on Universal Software Radio Peripheral (USRP) and PC. Extensive evaluations are performed to validate the effectiveness of the attacks and countermeasures.**

***Index Terms*—Database Driven Cognitive Radio Networks, Dynamic Spectrum Access, Jamming Attack.**

I. Introduction

The ever increasing demand of spectrum for wireless applications has inspired the emerging concept of Cognitive Radio Networks (CRNs), which is considered as a promising way to improve the utilization of the scarce radio spectrum [1]. In CRNs, there are two types of users: Primary Users (PUs) and Secondary Users (SUs). PUs are licensed users that are pre-assigned with certain channels to operate while SUs are unlicensed users that are allowed to use PUs' channels only when the channels are not occupied by the PUs [2], [3].

The landmark FCC ruling in November 2011 mandated the use of spectrum databases in USA, with rules of access for stationary and mobile cognitive radio (CR) nodes, as well as the consideration of specific capabilities such as geo-location [4]. In white space database, an SU queries a central database to obtain Spectrum Availability Information (SAI) at its location [5], [6]. Until August 2014, eleven white space database systems have been proposed and 6 of them have completed the testing and are expected to achieve the approval of FCC. The testing completed white space administrators include Google, Comsearch, Spectrum Bridge, Telcordia and etc [7]. Currently, Google Spectrum Database has provided API services to allow the SUs to query the database and find available TV-band white space spectrum at a given geo-location [8].

While the database-driven CRNs facilitate the detection of unoccupied spectrum by providing SAI to SUs, this new paradigm can be exploited by the jammers to amplify their ability of launching attacks. Specifically, by using this available spectrum knowledge, the jammers could significantly increase the successful rate of jamming the SUs. In this paper, we propose a novel jamming attack in database-driven CRNs. Different from the conventional jamming attacks in sensing based cognitive radio networks which employs limited sensing ability to detect the unoccupied channels [9], [10], [11], this kind of jammer submits an inquiry to the database instead of acquiring all the SAI at its position. With this information, the jammer could block the SUs with a high successful probability without sensing the spectrum. What's worse, the elimination of sensing process deduces the cost of the jamming attack and could save time for the jammer to emit more jamming signals. By successfully launching the jamming attack, the jammer could block the communication of SUs, achieving the denial-of-service (DoS) on the spectrum. Moreover, the jamming signal can also be incorrectly identified as the PU's waveform, or could lead to a low signal-noise-ratio (SNR), both of which will make SUs discard the utilization of the spectrum and leave the spare channels to the jammers.

Unfortunately, the traditional well-known anti-jamming techniques such as Direct-Sequence Spread Spectrum (DSSS)

This work is supported by National Science Foundation of China (No. 61272444, 61672350, 61622307, U1401253, U1405251) and National Science Foundation (No. 1527144 and 1553304).

Haojin Zhu, Chenliaohui Fang and Mengyuan Li are with Shanghai Key Laboratory of Scalable Computing and Systems, Department of Computer Science and Engineering, Shanghai Jiao Tong University, China. E-mail: zhu-hj@cs.sjtu.edu.cn, fclh1991@gmail.com, limengyuan@sjtu.edu.cn.

Yao Liu is with Department of Computer Science and Engineering, University of South Florida, 4202 E Fowler Ave, ENB 118, Tampa, FL 33620.

Cailian Chen is with Department of Automation, Shanghai Jiao Tong University, China. E-mail: cailianchen@sjtu.edu.cn.

Xuemin (Sherman) Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada, N2L 3G1. Email: xshen@bbcr.uwaterloo.ca.

and Frequency Hopping Spread Spectrum (FHSS) [12] will face the following challenges when being applied to address this new attack. First, these spread-spectrum techniques assume that the receivers should share the secrets (spreading codes or hopping sequences) with the sender prior to their anti-jamming communication. It may be quite difficult to have this assumption in the case that mobile SUs may never meet each other before the start of the transmission. Secondly, although some variants of these anti-jamming schemes, such as uncoordinated frequency hopping (UFH) and uncoordinated DSSS (UDSSS), are proposed to eliminate the requirement of pre-shared secrets [13] [14] [15] [16], the wide range frequencies or continuous broadband needed by these schemes may not be available in CRNs. Thirdly, the conventional anti-jamming techniques assume that the jammer does not know all the communication channels of legal users or can only jam a part of the spectrum. However, due to the limited number of unoccupied channels in CRNs, the jammers with a knowledge of all the available spectrum can be powerful enough to block all the channels and overcome the spreading gain.

This paper presents a new approach called Jammer Inference based Jamming Defense framework (or jDefender in short), which aims to infer the likelihood of a node being a jammer and exploit multiple of defending strategies based on the inference results. Our intuition is that, though it is difficult to prevent jamming attacks, it is possible to mitigate the impact of the jamming attacks. To achieve this, we introduce a novel Channel Allocation based Jammer Inference (CAJI) scheme.The basic idea of CAJI is assigning different channel sets to different SUs (including both of the normal SUs and the jammers). By exploiting the diversity of assigned available channels, a jammed event including a specific jammed channel will help increase the likelihood of being jammers for users which are assigned with this channel while ruling out those which are not assigned with jammed channel. The proposed CAJI scheme can identify the jammers from a set of the normal SUs even under the collusion attack performed by multiple jammers. The experimental results show that jDefender can infer all of the jammers exactly with a small false positive ratio. With this likelihood, jDefender adopts a series of countermeasures, including client puzzles [17], reducing allocated spectrum resource, as well as DSSS and FHSS to defend against the jamming attack. We perform the USRP based real-world experiments to demonstrate the practicality of the proposed jamming attack in database-driven CRNs. Extensive experiments have been performed to demonstrate the effectiveness of jDefender.

The contribution of our paper is summarized as below:

1) We identify a novel attack in Database-driven CRNs, in which the jammer can exploit white space database query to launch the jamming attack efficiently at the reduced cost even without the spectrum sensing.
2) We introduce a novel CAJI scheme, which exploits the diversity of the assigned available channels to distinguish the jammers from the normal SUs based on jamming events detected by SUs. To further improve the detection reliability, we introduce an Aggregated Jamming Attack Detection Algorithm.
3) We propose a novel Jammer Inference based Jamming Defense framework (jDefender), which is comprised of multiple anti-jamming strategies on spectrum query phase including *puzzle solving, channel allocation*, and other anti-jamming strategies during the wireless transmission.
4) We use USRP based real-world experiments to demonstrate its effectiveness and efficiency of jDefender.

The remainder of paper is organized as follows. In Section II, we formulate the concerned problem, including the adversary model and assumptions. In Section III, we propose the compositions of jDefender framework and the CAJI scheme. We present the submodules of jDefender including Jammer Inference Module, Channel Allocation Module, Anti-jamming Module respectively in Sections IV. We evaluate the jDefender framework based on USRP and PC in Section V. We provide the related work in Section VI, and conclude the paper in Section VII.

II. A Novel Jamming attack in Database-driven Cognitive Radio Networks

In this section, we first introduce the database driven CRNs, and then present a novel jamming attack, namely Query based Jamming Attack, which is based on the white space database queries. Based on the attack, we give the adversary model and security assumptions.

A. Overview of Database-driven CRNs

According to IETF RFC 7545 standard, "Protocol to Access White-Space (PAWS) Databases" [5], Database-driven CRNs typically consist of three components: 1) a set of primary users $\mathcal{P}$, 2) a set of secondary users $\mathcal{S}$, 3) the database $\mathcal{DB}$ which stores the SAI. $\mathcal{P}$ and $\mathcal{S}$ share the same set of channels $\mathcal{CH}$. A secondary user su_i ($1 \leq i \leq |\mathcal{S}|$) should query the $\mathcal{DB}$ to obtain the available channels at a specific location which are not occupied by $\mathcal{P}$ before using the channels. A typical database query process is comprised of three phases [18]:

1) *Query phase*: su_i ($1 \leq i \leq |\mathcal{S}|$) delivers a query containing his location loc to the $\mathcal{DB}$ at time t;
2) *Retrieval phase*: $\mathcal{DB}$ responds to su_i with the SAI containing available channels and the corresponding expected service time of each channel;
3) *Commitment phase*: after receiving the SAI from $\mathcal{DB}$, su_i determines the channels that he is going to use and registers the selected channels in $\mathcal{DB}$.

B. Jamming Attacks in CRNs

Radio jamming is a kind of Denial of Service (DoS) attack which aims at disrupting communications at the physical and link layers of wireless networks. The goal of the jammers is to block the packets of SUs as many as possible in the interference range. Similar to the previous researches in [9], we assume that the jammers cannot jam the PUs when PUs are active due to the following two facts.

- Firstly, the PU can impose a heavy penalty on the attacker if its identity is identified by the PUs;

- Secondly, the jammers may not reach the interference range of the PUs, especially in the case of PUs are formed by TV towers [9].

According to the existing studies [20], jammers can attack the channels by the following two ways:

1) Keeping the wireless spectrum busy (e.g. constantly injecting packets to a shared spectrum), which will always collide with the packets transmitted by the legitimate SUs using the same channels;
2) Injecting high interference power in the vicinity of a victim, so that the Signal to Noise Ratio (SNR) deteriorates heavily and no data can be received correctly.

To launch the jamming attack described above, the jammers could choose the following four strategies: constant jamming, deceptive jamming, random jamming and reactive jamming. Different from the other three strategies, the reactive jammers solely target at packets that are already *on the air*, which means the jammer starts the jamming actions only when detecting the transmission of the SUs.

C. Adversary Assumptions

According to IETF RFC 7545 standard, "Protocol to Access WS database", FCC rules require that a mobile device should register its owner and operator contact information, its device identifier, its location, and its antenna height to the spectrum database [21]. Therefore, before accessing to the spectrum database, each SU should register to DB for obtaining its public/private key according to its identity. When SU is accessing to spectrum database, an SU should perform Transport Layer Security (TLS) Protocol to authenticate itself as suggested by IETF RFC 7545 standard. It is also assumed that the number of jammers is much less than the number of SUs. It is further assumed that each $su_i(1 \leq i \leq |\mathcal{S}|) \in \mathcal{S}$ could transmit packets on k_i channels simultaneously, which is known by the $\mathcal{DB}$.

In this study, the attacker is assumed to launch a random jamming or sweep jamming attack towards the channels assigned by the DB. We made this assumption due to the following reasons. Firstly, FCC has removed the location-sensing capabilities for SUs in future dynamic spectrum sharing system. Secondly, the individual or few multiple sensing nodes may face the serious problem of unreliable detection [22]. The random jamming based on unreliable spectrum sensing may potentially introduce the interferences to the PU, which is strictly prohibited by FCC[21]. Lastly, in this study, we only consider the jamming attack arising from database driven cognitive radio networks rather than the traditional sensing based jamming. We believe that the traditional random jamming attack is an important topic and deserves the separate research. Similar to other client puzzle solutions [17], each SU could compute hash function and is willing to sacrifice some computational resources to avoid being jammed. The available channels are shared by multiple SUs by using media access control (MAC) protocol (e.g. CSMA/CA). In this study, we do not consider the privacy issues arising from dynamic spectrum access which have been well studied in the previous works such as [18], [19], which deserves the separate research.

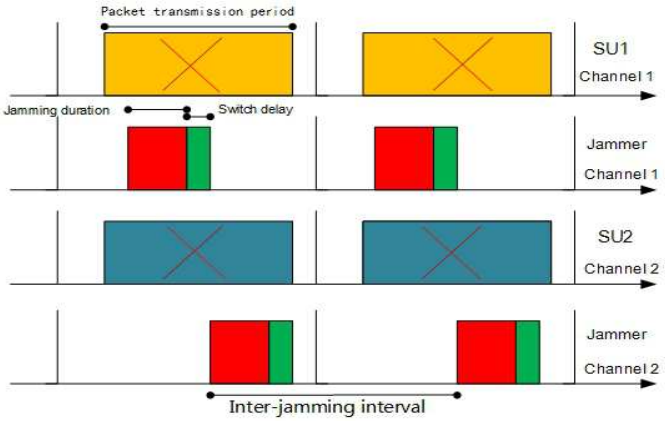


Fig. 1: Instance of query based jamming on two channels

D. A Novel Database Assisted Query based Jamming Attack

The traditional jamming attacks assume that the white space availability is unknown to both of jammers and SUs. Therefore they have to sense the channels to detect the presence of PUs independently before utilizing or jamming the spectrum in each time slot [9] [10] [11]. Since the jammers and the SUs may choose the totally different sets of channels to sense, this kind of jamming attack has a low success rate to block the transmission of SUs.

However, in database-driven CRNs, jammers could obtain the SAI through database query instead of spectrum sensing. Generally, the $\mathcal{DB}$ cannot distinguish jammers from the normal SUs. Thus, $\mathcal{DB}$ may respond all the SAI to the jammers. The database query process eliminates the requisite of the sensing process and save the energy and time consumption of the jammers. Therefore, this kind of attack does not require a strong sensing capability of the device, making it much easier to launch.

1) Impact of Query based Jamming Attack: Query based jamming attack amplifies the capability of the jammer by loosing the assumption made by the previous research works [9] [10] [11], which assume that both the jammers and the SUs only sense and transmit/jam on the spectrum (if available) only once in one time slot. In query based jamming attack, the jammer can jam one channel and switch to another within a specific time slot, which means that the jammers can block more channels in one time slot. As a result, equipped with the fast channel switching property of cognitive radio devices, the jammers are able to magnify the jamming capability for several times.

We estimate the impact of query based jamming attacks as follows. As shown in [23], even with error coding scheme, the jammer can cause the packet unreadable by jamming 15% of the message size. We assume the jammer could jam one channel and the channel switching time is $0.5ms$[24]. Moreover, the transmission rate of the transmitters is set to $1Mbps$ and the packet size is 1500 $bytes$, which means the time for communication is merely $12ms$ (regardless of the data sampling and processing time). Then, the time required by the jammer to successfully block one channel is $12 \cdot 15\% + 0.5 = 2.3ms$, which means an attacker could be four

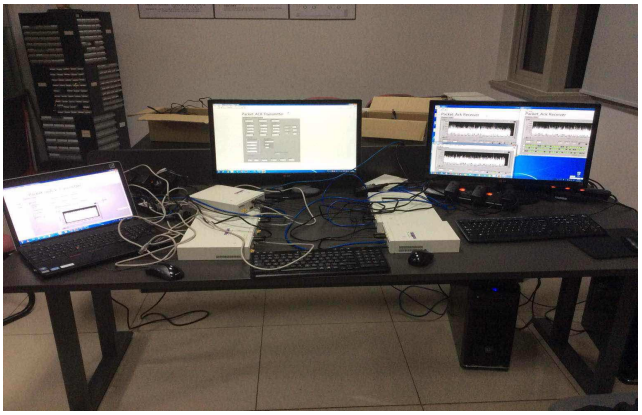


Fig. 2: Experiment prototype system with 7 USRPs

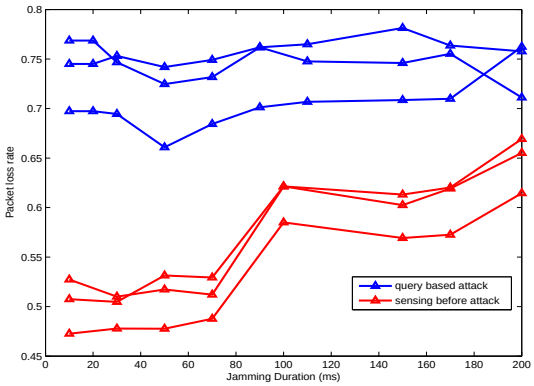


Fig. 3: Jamming efficiency of two kinds jammers

Algorithm 1: Channel Allocation based Jammer Inference (CAJI) Algorithm

Input: A series of jamming events detection results $\mathbf{JE_n} = \{je_1,\ je_2,\ \cdots,\ je_n\}$

Output: The identities of jammers

for $i \leftarrow 1$; $i \leq n$; $i \leftarrow i+1$ **do**

- $t \leftarrow\ time\ of\ je_i$;
- $loc \leftarrow\ location\ of\ je_i$;
- $ch \leftarrow\ channel\ of\ je_i$;
- $\mathcal{DB}$ retrieves $\mathcal{S}^{ch}_{t,loc}$, which denotes the SUs who know channel ch is available in location loc at time t;
- **for** *each* $su_j \in \mathcal{S}^{ch}_{t,loc}$ **do**
  - **if** $je_i = 1$ **then**
    - Increase $Pr\{su_j\ is\ a\ jammer\}$;
  - **else**
    - Decrease $Pr\{su_j\ is\ a\ jammer\}$;
  - **if** $Pr\{su_j\ is\ a\ jammer\}$ *excesses a threshold* **then**
    - Print su_j is a jammer;

times more powerful than the conventional jamming attacker. Generally, given the packet transmitting time T_t, the minimal jamming period for one packet T_j, the channel switching time T_w and the number j of channels the jammer can block simultaneously, the total number of channels the jammers can jam in one time slot is $j \cdot \lfloor \frac{T_t - T_j}{T_j + T_w} \rfloor + j$. Fig. 1 shows an instance that a query-based jamming attack on two channels simultaneously.

2) USRP based Experimental Evaluation: We have implemented an USRP (Universal Software Radio Peripheral) based experiment to demonstrate the effectiveness of query based jamming attacks.

Experiment Setup: The prototype based on USRP is constructed to evaluate the effectiveness of our proposed query based jamming attack in database-driven CRNs. As shown in Fig. 2, the prototype system consists of 7 USRP devices, one of which is a jammer and the other six are three pairs of SUs. Each pair includes one sender and one receiver. Moreover, every pair occupies a channel that is different from the others. The software we use to implement our experiments is LABVIEW.

Evaluation Results: In the evaluation, it takes the conventional sensing based jammer about 10 ms to finish one round of sensing. Fig. 3 shows the package loss ratio of three pairs of wireless communications under the query based jamming attack and the conventional sensing based jamming attack, respectively. It is observed that the query based jamming can incur a higher package loss ratio (e.g., 78%) than the conventional jamming attack (e.g., 52%) under the similar parameter settings. Note that, the above evaluation on query based jamming attack only provides a lower bound estimation of the capability of the jammers. In practice, along with improvement of channel switch speed by adopting more advanced USRP equipments, the effectiveness of the query based jamming attack can be further enhanced.

Based on above evaluation, it is concluded that the query based jamming attack poses a real threat to the wireless communications in database-driven CRNs, which motivates us to propose the corresponding defending solutions in the subsequent section.

III. Channel Allocation based Jammer Inference in Database Driven Spectrum Sharing

The first step towards effectively thwarting the jamming attacks in database driven spectrum sharing is identifying the targeted jammers, which will make the anti-jamming strategies more effective. In this section, we will introduce a novel Channel Allocation based Jammer Inference (CAJI) algorithm, which aims to identify the jammers based on the channels jammed and the allocated channel set for each SU in database-driven CRNs. The basic idea of CAJI scheme is motivated by the following observations:

- Firstly, different from the conventional jammer who can arbitrarily jam the channels, a jammer in database-driven CRNs cannot jam the channels which are occupied by the PUs to avoid introducing the interference to the PUs
- Secondly, in database-driven CRNs, the channels that a specific SU or jammer can access are determined by the $\mathcal{DB}$.

TABLE I: Summary of notations

Symbol	Meaning
$\mathcal{DB}$	$\mathcal{DB}$ is the database furnishing the SAI.
$\mathcal{S}$, $\mathcal{P}$	$\mathcal{S}$ is SUs in CRNs and $\mathcal{P}$ is PUs in CRNs
su_i, k_i	$su_i \in \mathcal{S}$, su_i is the identity of an SU and su_i can use at most k_i channels simultaneously
$je^{ch}_{t,loc}$	$je^{ch}_{t,loc}$ is jamming event aggregated detection result on channel ch in location loc at time t
	$je^{ch}_{t,loc} = 1$ means jamming event is detected; otherwise $je^{ch}_{t,loc} = 0$
p^k_i	su_i's likelihood of being a jammer after k rounds inference
$\mathcal{S}^{ch}_{t,loc}$	a set of SUs who know the channel ch is available in location loc at time t.
$\mathcal{AC}^t_{loc}$	$\mathcal{AC}^t_{loc}$ is a set of total available channels in location loc at time t.
$\mathcal{C}^i_{t,loc}$	$\mathcal{C}^i_{t,loc}$ is the set of available channels allocated to su_i in location loc at time t
$et^i_{t,loc}$	$et^i_{t,loc}$ is a array, $et^i_{t,loc}[ch](ch \in \mathcal{C}^i_{t,loc})$ means the expected service time of channel ch allocated to su_i.
un^t_{loc}	un^t_{loc} is a array, $un^t_{loc}[ch]$ means the number of SUs using channel ch in location loc at time t
$los^{max,db}_{loc,i}$	$los^{max,db}_{loc,su_i}$ is the maximum network service quality loss can be imposed to su_i

Therefore, based on the above two observations, $\mathcal{DB}$ is able to identify SUs by allocating different available channel sets and different corresponding expected service time, and conversely, these allocated channels can be used as the hints to determine if a specific user is a jammer in the presence of a jamming event. More specifically, $\mathcal{DB}$ maintains two key variables for each su_i: 1). *identity of* su_i; 2). $Pr\{su_i\ is\ a\ jammer\}$*, the likelihood of being a jammer*. Let $je^{ch}_{t,loc}$ represent a jamming event detected on channel ch in location loc at time t, and $\mathcal{S}^{ch}_{t,loc}$ represent the set of SUs who are assigned with channel ch with the same location loc and time t. If $je^{ch}_{t,loc} = 1$, which denotes channel ch is jammed, based on the first observation, it can be inferred that it is the nodes $\in \mathcal{S}^{ch}_{t,loc}$ that launch the jamming attack. Since $\mathcal{DB}$ is spectrum allocator, according to the second observation, $\mathcal{DB}$ knows all the $su \in \mathcal{S}^{ch}_{t,loc}$. Then $\mathcal{DB}$ can increase the likelihood of being a jammer of each SU among $\mathcal{S}^{ch}_{t,loc}$. On other hand, if $je^{ch}_{t,loc} = 0$, which denotes channel ch is not jammed, then $\mathcal{DB}$ could decrease the value of the likelihood of being a jammer of each SU in $\mathcal{S}^{ch}_{t,loc}$. We summarize the proposed CAJI algorithm in Algorithm 1.

There are two challenges that need to be addressed to implement the CAJI scheme. Firstly, though a node losing its sending ability is a clear signal that it is being jammed, a weak reception capability (i.e. a low Packet Delivery Ratio, or PDR) can also be caused by several factors besides jamming, such as a low link quality due to large distance between the sender and the receiver. Therefore, the first challenge is how to differentiate a jamming event from an observed low PDR due to natural causes of poor link quality. Secondly, the database should update the jammer likelihood of SUs after jamming events detection. The smart jammer can launch the collusion attack to avoid being inferred by the CAJI directly. Thus, the second challenge is how to update the jammer likelihood even for the the collusion attack. In the below, we will introduce *Jamming Event Aggregated Detection* and *Jammer Inference via Sequential Bayesian Model* to address the above two challenges.

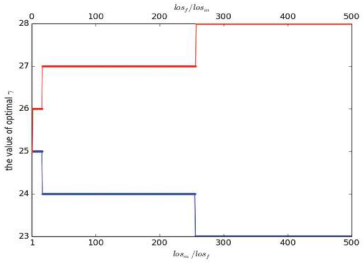
A. Jamming Event Aggregated Detection

When some jammers launch the jamming attack, the victims can detect the presence of jamming event (e.g the Packet Delivery Ratio and Energy Detection) by using single-statistics-based and consistent-check-based algorithms according to the previous research [20]. However, a single node's detection may face the challenge on how to differentiate a jamming event from an unintentional wireless network interference. Further, a smart jammer can choose a more sophisticated strategy (e.g., random jamming) to launch a jamming attack, which will also increase the difficulty of jamming attack detection. In this subsection, we propose a *Jamming Event Aggregated Detection* scheme. In particular, in database-driven CRNs, by collecting the individual detection results performed by the multiple SUs, $\mathcal{DB}$ can perform an aggregated detection on a jamming event. Without loss of generality, we assume that n SUs are involved in the jamming detection and k out of n will judge the channel as "Jammed" during a jamming event.

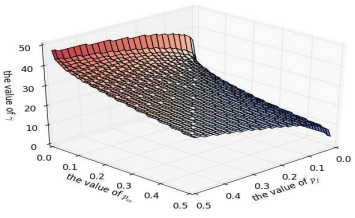
We define γ as the *Minimum Number of voters* needed to accept a channel ch is "Jammed" in a jamming event by checking the detection observations from n SUs. Obviously, a smaller γ [25] will lead to a more prompt judgment for jamming event at the sake of a higher false positive rate. On the other hand, a larger threshold of γ will increase the false negative rate at the cost of a delayed detection of an observed jamming event. Therefore, there is a tradeoff between the jamming event detection speed and the detection robustness. In the following section, we will give a detailed discussion on choosing an appropriate selection on γ. For the simplicity of presentation, we adopt the following notations:

- J_0: jamming attack is absent
- J_1: jamming attack is existent
- R_0: the $\mathcal{DB}$ judge the jamming attack is absent
- R_1: the $\mathcal{DB}$ judge the jamming attack is existent

1) Choosing Optimal γ for Aggregated Jamming Detection: In this subsection, we will discuss how to choose the optimal γ for aggregated jamming detection by $\mathcal{DB}$. Without loss of generality, we assume that each SU has the same average false positive rate of jamming detection p_f , and the same average false negative rate of jamming detection p_m. Thus, the false positive on $\mathcal{DB}$'s aggregated jamming detection can be given



(a) Relation between γ and los_f/los_m, los_m/los_f



(b) Relation between γ and p_f , p_m

(c) Experiment prototype system with 14 USRPs

Fig. 4: Simulation and Experimental Evaluation for Jammer Inference Algorithm

by:

$$\begin{aligned} D_f &= Prob\{R_1|J_0\} \\ &= \sum_{l=\gamma}^{n} \binom{n}{l} p_f^l (1-p_f)^{n-l} \end{aligned} \tag{1}$$

Similarly, the false negative of $\mathcal{DB}$'s aggregated jamming detection can be given by:

$$\begin{aligned} D_m &= Prob\{R_0|J_1\} \\ &= 1 - \sum_{l=\gamma}^{n} \binom{n}{l} p_m^{n-l} (1-p_m)^l \end{aligned} \tag{2}$$

For $\mathcal{DB}$, let los_f be the loss caused by false positive and los_m be the loss caused by false negative. The expected loss is defined as follows:

$$E[\gamma]_{loss} = los_f \cdot D_f + los_m \cdot D_m \tag{3}$$

The below theorem gives how to choose an optimal γ to minimize the expected loss.

Theorem I: *In a jamming event located in a specific cell, n SUs in this cell are potential observer for this jamming event. Given los_f be the loss caused by false positive, los_m as the loss caused by false negative, the optimal value of γ can be chosen as following equation to minimize the expected loss*

$$\gamma = min\{n, \lceil \frac{\ln(los_f/los_m) + n\ln((1-p_f)/p_m))}{\ln((1-p_m)(1-p_f)/(p_f p_m))} \rceil\} \tag{4}$$

Proof: We will prove the **Theorem I** in Appendix A.

According to **Theorem I**, the optimal γ is determined by three parameters, los_f/los_m, p_m and p_f. In the below, we give a numerical analysis of the optimal γ by setting $n = 50$. According to existing works [20], $p_f \lll 1 - p_f$, and $p_m \lll 1 - p_m$. In Fig. 4(a), we fix $p_f = 0.20$, $p_m = 0.20$ and plot γ under different los_f/los_m and los_m/los_f settings respectively. The blue line shows that the optimal γ decreases slowly from 25 to 23 when los_m/los_f increases from 1 to 500, and the red line shows that the optimal γ increases slowly from 25 to 28 when los_f/los_m increases from 1 to 500. Thus we can conclude that the loss of false positive los_f and the loss of the false negative los_m are not the major factors contributing to the value of optimal γ. In Fig. 4(b), when los_f to los_m ratio is set to 1, the value of optimal γ are increased significantly along with the increase of p_m and p_f ($p_m, p_f < 0.5$). It is clear that false positive ratio p_f and false negative ratio p_m play a more important role in optimal threshold selection. Further, a larger false positive p_f leads to a larger optimal γ, which means that more voting SUs are necessary to make a jamming event detected, while a high false negative p_m leads to a smaller optimal γ, which means that more veto SUs are necessary for the jamming detection.

B. Jammer Inference via Sequential Bayesian Model

As mentioned above, $\mathcal{DB}$ maintains a table recording jammer likelihood for all of the SUs. We use $Prob\{su_i \; is \; a \; jammer\} = p_i^k$ to represent how likely su_i is a jammer after k rounds of inferences. According to the CAJI algorithm 1 proposed in the previous section, we can update the value of p_i^k by using the result of aggregated jamming detection. We assume that $\mathcal{DB}$ has no prior knowledge about SUs. Thus $\mathcal{DB}$ sets $p_i^0 = \frac{1}{2}$ for su_i who just joins the network. In the below, we will propose a novel jammer inference scheme based on Sequential Bayesian Inference Models (SBIM)[26].

Sequential Bayesian Inference [26] is the Bayesian estimation of a dynamic system which is changing in time. Let $\mathbf{s_t}$ be the state of the system at time t. The goal of the Sequential Bayesian inference is to estimate a posteriori probability density function (pdf) $p\{\mathbf{s_t}|\mathbf{E_t}\}$, where $\mathbf{E_t} = \{\mathbf{e_t}, \mathbf{e_{t-1}}, \cdots \mathbf{e_0}\}$ represent a sequence of arriving events before time t. In this work, the goal of $\mathcal{DB}$ is to estimate the value of p_i^k for su_i according to $\mathbf{JE_k}$, where $\mathbf{JE_k}$ denotes the results of k jamming detection. Thus, p_i^k is defined as follows:

$$p_i^k = Prob\{su_i \textbf{ is a jammer } | \; \mathbf{JE_k}\}\} \tag{5}$$

Let $je_{t,loc}^{ch}$ denote the result of jamming detection on channel ch in location loc at time t. Note that for a specific $su_i \in \mathcal{S}_{t,loc}^{ch}$, $je_{t,loc}^{ch}$ represents the jamming event detection result of the $(k+1)_{th}$ inference round. Therefore, in terms of the value of $je_{t,loc}^{ch}$, we can update the value of p_i^{k+1} in the following two cases.

Case I: Jamming Attack Detected for channel ch. According to the CAJI scheme, assume that channel ch is jammed in location loc at time t and $\mathcal{S}_{t,loc}^{ch}$ is the set of SUs (including the jammers) who have been assigned with available channel ch in location loc at time t. When $\mathcal{DB}$ detects the presence of the jamming attack, $\mathcal{DB}$ has no idea of who implements

the jamming attack or the number of jammers. However, $\mathcal{DB}$ can infer that one or multiple jammers should belong to $\mathcal{S}^{ch}_{t,loc}$. Therefore, $\mathcal{DB}$ can update the value of p_i^{k+1} of $su_i \in \mathcal{S}^{ch}_{t,loc}$ by following equation:

$$p_i^{k+1} = p_i^k(\cdot \mid je^{ch}_{t,loc} = 1) = \frac{p_i^k(\cdot,\ je^{ch}_{t,loc} = 1)}{p(\ je^{ch}_{t,loc} = 1)} = \frac{p(je^{ch}_{t,loc} = 1 \mid \cdot) \cdot p_i^k(\cdot)}{p(je^{ch}_{t,loc} = 1)} \quad (6)$$

$p_i^k(\cdot,\ je^{ch}_{t,loc} = 1)$ denotes the probability that su_i is a jammer and jamming attack on channel ch is detected after k rounds of inferences; $p(je^{ch}_{t,loc} = 1 \mid \cdot)$ denotes the probability that jamming attack on channel ch is detected when su_i is a jammer.

Case II: Jamming Attack Not Detected for channel ch. When jamming attack is not detected for channel ch in location loc at time t, there are two possibilities: 1) there is no jammer in $\mathcal{S}^{ch}_{t,loc}$; 2) the jammers who have been assigned with available channel ch have not launched the jamming attack. Let $\overline{p_i^k} = 1 - p_i^k$ denote that the probability that su_i is not a jammer. Then the value of $\overline{p_i^{k+1}}$ can be updated by the following equation:

$$\overline{p_i^{k+1}} = \overline{p_i^k}(\cdot \mid je^{ch}_{t,loc} = 0) = \frac{\overline{p_i^k}(\cdot,\ je^{ch}_{t,loc} = 0)}{p(\ je^{ch}_{t,loc} = 0)} = \frac{\overline{p}(je^{ch}_{t,loc} = 0 \mid \cdot) \cdot \overline{p_i^k}(\cdot)}{p(je^{ch}_{t,loc} = 0)} \quad (7)$$

$\overline{p_i^k}(\cdot,\ je^{ch}_{t,loc} = 0)$ denotes the probability that su_i is not a jammer and jamming attack on channel ch is not detected after k rounds of inferences. $\overline{p}(je^{ch}_{t,loc} = 0 \mid \cdot)$ denotes the probability that the jamming attack is not detected when su_i is not a jammer. Note that updating the value of $\overline{p_i^{k+1}}$ also brings to the updating of p_i^{k+1}'s value since $p_i^{k+1} = 1 - \overline{p_i^{k+1}}$.

Note that, we use equation (6) and equation (7) to update SUs' likelihood of being a jammer recursively. To calculate these two equations, we need to calculate $p(je^{ch}_{i,loc} = 1 \mid \cdot)$, $\overline{p}(je^{ch}_{t,loc} = 0 \mid \cdot)$, $p(je^{ch}_{t,loc} = 1)$ and $p(je^{ch}_{t,loc} = 0)$ Before calculating these notations, we will introduce two new notations:

- $p^k_{atk,i}$ denotes the estimated probability probability that su_i will implement the jamming attack after k rounds of inference. Since $\mathcal{DB}$ has no prior knowledge about each SU, thus $\mathcal{DB}$ does not know the value of $p^k_{atk,i}$. However, $\mathcal{DB}$ can estimate $p^k_{atk,i}$ by the probability that su_i associated with jamming events.

- $p^k_{natk,i}$ denotes the estimated probability that su_i will not implement the jamming attack after k rounds of inference. We can estimate $p^k_{natk,i}$ from two aspects: 1) su_i is not a jammer, 2) su_i is a jammer but has not implemented the attack. Thus, $p^k_{natk,i}$ can be derived by the following equation:

$$p^k_{natk,i} = \overline{p_i^k} + p_i^k(1 - p^k_{atk,i}) \quad (8)$$

With $p^k_{atk,i}$ and $p^k_{natk,i}$, $p(je^{ch}_{t,loc} = 1 \mid \cdot)$ can be expressed as:

$$p(je^{ch}_{t,loc} = 1 \mid \cdot) = p^k_{atk,i} + (1 - p^k_{atk,i})\Big(1 - \prod_{su_i \in \mathcal{S}^{ch}_{t,loc} \setminus su_i} p^k_{natk,j}\Big) \quad (9)$$

$\overline{p}(je^{ch}_{t,loc} = 0 \mid \cdot)$ can be derived as:

$$\overline{p}(je^{ch}_{t,loc} = 0 \mid \cdot) = \prod_{su_j \in \mathcal{S}^{ch}_{t,loc} \setminus su_i} p^k_{natk,j} \quad (10)$$

$p(je^{ch}_{t,loc} = 1)$ and $p(je^{ch}_{t,loc} = 0)$ can be expressed as:

$$p(je^{ch}_{t,loc} = 1) = 1 - \prod_{su_i \in \mathcal{S}^{ch}_{t,loc}} p^k_{natk,i} \quad (11)$$

$$p(je^{ch}_{t,loc} = 0) = \prod_{su_i \in \mathcal{S}^{ch}_{t,loc}} p^k_{natk,i} \quad (12)$$

Therefore, equation (6) and equation (7) can be calculated by recursively applying equations (8) – (12).

C. Security Analysis

In the section, we will discuss how the proposed Jammer Inference algorithm can thwart forged jamming report attack as well as the collusion attack.

1) Thwarting forged jamming reports: The jammers can forge two types jamming reports to mislead the $\mathcal{DB}$:

- Type 1: The jammer reports that a channel ch is jammed. However, $\mathcal{DB}$ has not allocated channel ch to the jammer;
- Type 2: The jammer is allocated with channel ch and reports channel ch is jammed. However channel ch is not actually jammed.

$\mathcal{DB}$ can easily address the type 1 forged reports since $\mathcal{DB}$ is the spectrum allocator and it has the clear idea of the global spectrum allocation, which will make $\mathcal{DB}$ easily identify and thwart this attack. For Type 2 forged report, if jammers report type 2 forged jamming report to $\mathcal{DB}$, it will accelerate the process of its being identified as the jammers according to the proposed CAJI algorithm. Therefore, it can prevent the jammers from launching Type 2 forged jamming report attack. Note that, the existing solutions on thwarting falsified spectrum sensing attack such as [40], [41], [42] can be also adopted to enhance the jamming event aggregated detection.

2) Thwarting collusion attacks: Collusion attack represents a great challenge for jamming inference. Colluded jamming attack in database-driven CRNs is defined as that two or more jammers share their available channels (SAI) and launch the jamming attacks. The proposed Jammer Inference algorithm can successfully thwart this colluded jamming attack due to the following reason. Without loss of the generality, a channel ch is assumed to be assigned to jammer j_b, who shares this SAI with another jammer j_a. Then, j_a launches the jamming attack towards channel ch. Based on the proposed CAJI scheme, j_b will not evade the inference since his jammer likelihood will still be increased as long as the channels assigned to him are under jamming attack. This simple case can be also generalized to a more complicated case of collusion attack

TABLE II: jammers' and normal SUs' likelihood of being a jammer for different inference rounds

Rounds / SU	0	5	10	15	20	25	30
$jammer_0$	0.500	0.328	0.899	0.989	0.998	0.999	0.999
$jammer_1$	0.500	0.365	0.935	0.985	0.998	0.999	0.999
$jammer_2$	0.500	0.193	0.527	0.632	0.999	0.999	0.999
su_0	0.500	0.309	0.096	0.071	0.032	0.011	0.002
su_1	0.500	0.539	0.499	0.438	0.358	0.189	0.089
su_2	0.500	0.191	0.116	0.375	0.497	0.499	0.476
su_3	0.500	0.527	0.346	0.162	0.065	0.023	0.008
su_4	0.500	0.270	0.126	0.033	0.012	0.007	0.003
su_5	0.500	0.255	0.105	0.039	0.014	0.005	0.001
su_6	0.500	0.255	0.105	0.027	0.009	0.003	0.001
su_7	0.500	0.548	0.401	0.192	0.076	0.027	0.007
su_8	0.500	0.566	0.341	0.179	0.087	0.032	0.011
su_9	0.500	0.608	0.408	0.191	0.075	0.027	0.009

launched by multiple jammers. In other words, no matter how many jammers share the channel ch, if one of the jammers launch the attack, the original channel holder (the jammer with assigned channel ch) will be assigned with a higher likelihood of being a jammer. Therefore, after several rounds of inference, the jammers can be identified.

D. USRP based Experimental Evaluation

We have implemented an USRP based experiment to demonstrate the effectiveness and the feasibility of the CAJI scheme with *Sequential Bayesian Inference Models*.

Experiment Setup: The USRP based prototype is constructed to examine the effectiveness of our proposed *Jammer Inference Algorithm* in database-driven CRNs. As shown in Fig. 4(c), the prototype system consists of 14 USRP devices, 3 of them are jammers, 10 of them are normal SUs and the last one is implemented as a $\mathcal{DB}$. Jammers and SUs can communicate with $\mathcal{DB}$ to obtain SAI. Each USRP device is driven by a PC with LABVIEW.

Evaluation Results: Table II shows that 3 jammers' and 10 SUs' likelihood of being a jammer will be updated for different inference rounds. The initial likelihood of each jammer and SU are 0.500 since the $\mathcal{DB}$ has no prior-knowledge of jammer and normal SU. From the table II, we can conclude that each jammer's likelihood of being a jammer is larger than 0.99 after 20 inference rounds. As for the normal SUs, only the su_2 is still suspected as a jammer with probability 0.476 after 30 rounds inference, and for the other normal SUs, the likelihood of being a jammer is less than 0.1 after 30 rounds inference. Thus, we can conclude that our proposed CAJI scheme with *Sequential Bayesian Inference Models* is effective and efficient.

IV. The Proposed Jammer Inference based Jamming Defense (jDefender) Framework

In the previous section, we have introduced the Jammer Inference Module based on the proposed CAJI algorithm. In this section, based on this Jammer Inference Module, we will introduce a novel anti-jamming framework called jDefender (Jammer Inference based Jamming Defense framework) framework.

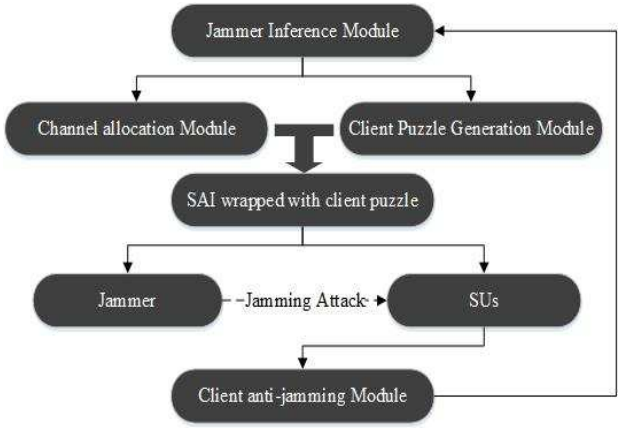


Fig. 5: Framework architecture of jDefender

A. Overview of jDefender

The jDefender exploits a series of anti-jamming strategies including channel allocation, client puzzle and FHSS/DSSS, and utilizes the jammer inference results to drive the jamming countermeasures. In particular, the system will maintain the inference results (or likelihood of being a jammer) for each SU, which will be updated by *Jammer Inference Module* whenever a jamming event is observed. The resources that SUs can enjoy (e.g., allocated available channels) and the cost paid for anti-jamming (e.g., client puzzles) will be determined based on how likely this user is a jammer by *Channel Allocation Module* and *Client Puzzle Generation Module*, respectively. The SUs can also implement the traditional anti-jamming methods such as FHSS or DSSS via *Client Anti-jamming Module*. The *Client Puzzle Generation Module* and the *Client Anti-jamming Module* are two components of the *Anti-jamming Module*. The detailed framework architecture is given in *Fig. 5*.

B. Mitigation at Channel Allocation Module

$\mathcal{DB}$ utilizes the *Channel Allocation Module* to allocate the spectrum resource (or available channels) for SUs according to their likelihoods of being a jammer. This kills two birds with one stone. On the one hand, it can mitigate the impact of the jamming attacks especially for suspicious SUs with a high jammer likelihood. On the other hand, allocating different channel set to different SUs can help to infer the jammers as described in CAJI scheme.

Let $\mathcal{AC}^t_{loc}$ denote the total available channels in location loc at time t. Since $\mathcal{DB}$ knows su_i can use at most k_i channels simultaneously, $\mathcal{DB}$ randomly picks a subset of channels $\mathcal{C}^i_{t,loc}$ of size k from $\mathcal{AC}^t_{loc}$ as the response to su_i's spectrum query. Further, $\mathcal{DB}$ also needs to determine the expected service time for each channel in $\mathcal{C}^i_{t,loc}$, which is denoted as $et^i_{t,loc}$. Here, $et^i_{t,loc}[ch](ch \in \mathcal{C}^i_{t,loc})$ means the corresponding expected service time of channel ch. $\mathcal{DB}$ cannot distinguish the jammers from the normal SUs with the full confidence. If su_i is a jammer, long expected service time may cause a significant network loss. On the other hand, if the su_i is a normal SU, short expected service time may significantly decrease the quality of the service of database CRNs. Fortunately, $\mathcal{DB}$

maintains the likelihood of being a jammer for each SU. By using the likelihood, an optimal service time allocation scheme is proposed to provide an acceptable quality of service for the normal SUs and, at the same time, minimize the impact of jamming attack caused by the potential jammers. Before proposing the channel allocation scheme, we will firstly estimate the potential network loss caused by jamming attack and network service quality loss caused by channel allocation respectively. After that, we will discuss how to optimize the channel allocation problem.

1) Estimating The Loss of Jamming Attack: Jamming attack is a DoS attack which will devastate the wireless channels. It is obvious that jamming attack will cause the network loss. In this section, we will consider the worst situation: the wireless communications on some channels will be interrupted when some jammers implement jamming attack on the same channels in the same location and the jammers will attack all the available channels as long as possible.

According to the above description, the jammer will attack all the channels in $\mathcal{C}^i_{t,loc}$ and the jamming duration for each channel is equal to the corresponding expected service time of each channel, which is denoted by $et^i_{t,loc}$. Let array un^t_{loc} refer to the number of SUs using each channel in $\mathcal{C}^i_{t,loc}$. We further assume un^t_{loc} will not change. Then, we can use the following equation to estimate the network loss caused by this jamming attack if $\mathcal{DB}$ allocates $\mathcal{C}^i_{t,loc}$ channels and $et^i_{t,loc}$ to a jammer.

$$los^{jm,i}_{t,loc} = \sum_{ch \in \mathcal{C}^i_{t,loc}} un^t_{loc}[ch] \cdot et^i_{t,loc}[ch] \tag{13}$$

2) Estimating The Loss of Network Service Quality: Let $\mathcal{RS}(\mathcal{C}^i_{t,loc})$ be the maximum of available spectrum resource the $\mathcal{DB}$ can allocate to su_i according to the spectrum resource usage of $\mathcal{C}^i_{t,loc}$ in location loc at time t. $\mathcal{DB}$ can allocate all the $\mathcal{RS}(\mathcal{C}^i_{t,loc})$ to su_i if $\mathcal{DB}$ has full confidence that su_i is a normal SU. However, based on the likelihood of being a jammer, $\mathcal{DB}$ may responde a part of $\mathcal{RS}(\mathcal{C}^i_{t,loc})$ to su_i.

To simplify the problem, it is assumed that every channel has the same spectrum resource rs which will be divided equally by each SU who shares the channel and the un^t_{loc} will not change. Thus, the spectrum quality obtained by su_i can be defined as following equation:

$$RS^i_{t,loc} = \sum_{ch \in \mathcal{C}^i_{t,loc}} \frac{rs}{un^t_{loc}[ch]} \cdot et^i_{t,loc}[ch] \tag{14}$$

For su_i, the loss of the network service quality caused by the expected service time allocation can be defined by the following equation:

$$los^{db,i}_{t,loc} = \mathcal{RS}(\mathcal{C}^i_{t,loc}) - RS^i_{t,loc} \tag{15}$$

We define $los^{max,db}_{loc,i}$ as the maximal loss of network service loss can be imposed on su_i, and the value of $los^{max,db}_{loc,i}$ is related with su_i's likelihood of being a jammer. We will define a monotonic increasing function $f(p)$, whose range is $[0, 1]$ and the parameter p is $Pr\{su_i \ is \ a \ jammer\}$. By using the function $f(p)$, we define $los^{max,db}_{loc,i}$ by following equation:

$$los^{max,db}_{loc,i} = f(p) \cdot \mathcal{RS}(\mathcal{C}^i_{t,loc}) \tag{16}$$

We can learn that larger p results in more maximal loss of network service quality according to equation (16). The least network service quality obtained by su_i:

$$Q^i_{t,loc} = (1 - f(p)) \cdot \mathcal{RS}(\mathcal{C}^i_{t,loc}) \tag{17}$$

We can allocate the expected service time of the corresponding channel $ch \in \mathcal{C}^i_{t,loc}$ by a simple method: $\mathcal{DB}$ allocates $Q^i_{t,loc}$ spectrum resource to su_i and each $ch \in \mathcal{C}^i_{t,loc}$ is allocated with the same spectrum resource. The array $et^i_{t,loc}$ allocated in this way can be expressed by the following equation:

$$et^i_{t,loc}[ch] = \frac{Q^i_{t,loc}}{k_i} \cdot un^t_{loc}[ch] \tag{18}$$

By using the above approach, we can learn that the loss of network service quality is exactly $los^{max,db}_{loc,i}$. However, this method may cause a great jamming attack loss since the more SUs use the channel, the longer expected service time is allocated. In the below, we will propose an improved expected service time allocation scheme by introducing the below optimization problem.

3) Optimization problem statement and solution: In this section, we consider the following problem settings.

1) The probability that su_i is a jammer: p^k_i
2) The maximum network quality loss: $los^{max,db}_{loc,i}$
3) A set of channels of size s_i: $\mathcal{C}^i_{t,loc}$
4) A set of SUs using the channel in $\mathcal{C}^i_{t,loc}$: un^t_{loc}

The aim of the optimization problem is to calculate the array of $et^i_{t,loc}$, namely the corresponding service time of channels in $\mathcal{C}^i_{t,loc}$, which will minimize the expected network loss caused by the attack from su_i by exploiting the SUs' jammer likelihood information. The expected network loss can be represented by the following equation:

$$E[los^{jm,i}_{t,loc}] = p^k_i \cdot los^{jm,i}_{t,loc} \tag{19}$$

The optimization problem can be expressed by the following formula:

$$\begin{aligned}
&\textbf{Minimize} && E[los^{jm,i}_{t,loc}] && (20)\\
&\textbf{Subject to} && los^{db,i}_{t,loc} \le los^{max,db}_{loc,i} && (21)\\
&&& a \ set \ channels \ \mathcal{C}^i_{t,loc} && (22)\\
&&& channels \ usage \ un^t_{loc} && (23)\\
&&& likelihood \ of \ a \ jammer \ p && (24)
\end{aligned}$$

According to the above equation(19), it can be concluded that to minimize the value of $E[los^{jm}_{t,loc}]$, we need to minimize the value of $los^{jm}_{t,loc}$ since the value of p^k_i has been inferred by the $\mathcal{DB}$. The following **Theorem II** discusses the solution of this optimization problem.

Theorem II*: Given a set of available channels $\mathcal{C}^i_{t,loc}$, the maximal loss of network service $los^{max,db}_{loc,i}$, the usage of each channel un^t_{loc}. Let $\mathcal{L} = Q^i_{t,loc}/rs$ and $\mathcal{O} = \sum_{ch \in \mathcal{C}^i_{t,loc}} \frac{1}{un^t_{loc}[ch]^2}$. The expected network loss $los^{jm,i}_{t,loc}$ can achieve the minimum value as follows: $los^{min,jm}_{t,loc} = \frac{\mathcal{L}}{\sum_{ch \in \mathcal{C}^i_{t,loc}} 1/(un^t_{loc}[ch]^4 \mathcal{O})}$. Further, for each channel $ch \in$*

$\mathcal{C}^i_{t,loc}$, *the expected service time can be defined as* $et^i_{t,loc}[ch] = \frac{los^{min,jm}_{t,loc}}{un^t_{loc}[ch]^3\mathcal{O}}$.

Proof: We will prove **Theorem II** in Appendix B.

We will evaluate the efficiency of our proposed improved channel allocation method in section VII.

C. Anti-jamming Module

In addition to the above described *Jammer Inference Module*, it is more important to propose a method which can alleviate the damage to the network caused by the jamming attack. In this section, we will propose the *Anti-jamming Module* to handle this challenge. In particular, the *Anti-jamming Module* has two sub-modules. The first one is *Client Puzzle Generation Module*, which is established in $\mathcal{DB}$'s side. And the other one is *Client anti-jamming Module*, which is established in SUs' side. They will be introduced in the below.

1) Client Puzzle Generation Module: The client puzzle scheme is commonly proposed as a solution to DoS attack in Internet. The basic idea of client puzzle is to construct a client puzzle for client to force the inquirer to commit his computational and energy resources. In our proposed jDefender, we use the client puzzle protocol to prevent the potential jammer from getting SAI easily. However, though client puzzle can effectively thwart query based jamming attack, it will inevitably incur a significantly computational overhead and access delay. Therefore, how to achieve the tradeoff between the security and the efficiency represents a major challenge for thwarting the anti-jamming in dynamic spectrum access system. To address this challenge, we introduce a novel client puzzle construction scheme, in which the difficulty of the solving client puzzle is determined not only by the system parameters but also the likelihood of a SU being a jammer. In other words, a normal user can access to the dynamic spectrum system efficiently by resolving a simple puzzle while a suspicious jammer has to solve a difficult client puzzle, which can effectively filter the jammers without introducing a high overhead.

Typically, the process of client puzzle consists of the following four steps in a Client/Server system.

1) Client initiates a connection request to server;
2) Server generates a puzzle, and then sends to the client;
3) Client solves the puzzle, and then sends the solution of the puzzle to server;
4) Server verifies the solution of the puzzle, and then permits the connection if the solution is correct

However, compared with the non-client puzzle protocol C/S system, the client puzzle protocol needs two additional package transmission in step 2) and step 3), which will increase the networks load. To relieve the network load and the server load, in our proposed *Client Puzzle Generation Module*, $\mathcal{DB}$ uses the puzzle to wrap SAI, and SU can get the SAI from the solution of the puzzle. What is more important, $\mathcal{DB}$ maintains the likelihood of being a jammer for each SU, and the difficulty of the puzzle is related with the likelihood. Therefore, for su_i with a less possibility of being a jammer, it only needs to solve a simple puzzle while for su_i with a larger likelihood, it has to solve a more difficult puzzle (probably computational impossible) and sacrifice enormous computational and energy resources. In the below, we will introduce the process of *Client Puzzle Generation Module* in details.

Puzzle Generated at Database: Assume *Client Puzzle Generation Module* generates a client puzzle Pz_i for su_i, Pz_i can be represented as follows:

$$Pz_i = (Y^*||f||r^*, h(Y||f||r)) \tag{25}$$

Y^* is a fixed k-length string derived from Y and f is a l-length random digit. The value of k and l are the common knowledge of all the SUs. The notation $||$ refers to the concatenation. Here, f is only used to extend the searching space to prevent the pre-computation. We concatenate the string r with a variable-length random digit x, and r^* is obtained by replacing all the bits of x with 0. After receiving the request from su_i who wants to query the available spectrum in location loc at time t, $\mathcal{DB}$ retrieves the white space availability spectrum. According to the *Channel Allocation Module*, $\mathcal{DB}$ has known that su_i can use at most k_i channels simultaneously. Thus, $\mathcal{DB}$ selects k_i available channels ($\mathcal{C}^i_{t,loc}$) from available channels ($\mathcal{AC}^t_{loc}$) randomly, and then selects n_i unavailable channels from the rest channels. Assume TC_{loc} is string whose length is k, where k refers to the number of channels (including the available channel and unavailable channel) in location loc. For k_i channels in the selected subset $C^i_{t,loc}$, $\mathcal{DB}$ sets the corresponding selected k_i bits of TC_{loc} as 1 and puts expected service time in array $et^i_{t,loc}$. For the rest $k - k_i - n_i$ bits of TC_{loc}, $\mathcal{DB}$ sets them as 0, and sets corresponding positions in $et^i_{t,loc}$ as a random number. We denote this new TC_{loc} as Y. Next, we replace the chosen $k_i + n_i$ bits of Y with 1 and denote this string as Y^*. Thus, $\mathcal{DB}$ can construct the puzzle according to the string Y^*, Y, f, r, r^* and the hash function h.

Puzzle Solved at SUs: After receiving the puzzle Pz_i, su_i should solve the Pz_i to obtain the SAI. According to [27], the most efficient way to compute the inverse of hash function is the brute-force testing of all the possible inputs. Therefore, su_i will check whether $\mathrm{h}(Y'||f||r') = h(Y||f||r)$ for any candidate value Y' and r' to find the solution.

Adjusting Puzzle Difficulty: The design of client puzzle should enable the difficulty of the puzzle to be adjusted according to the SUs' likelihood of being a jammer. We can modify the length of r and the parameter n_i to adjust the puzzle difficulty[27]. To measure the difficulty of the puzzle, we calculate the average number of hash function computations needed to successfully find a solution and give the following theorem.

Theorem III: *Given the puzzle Pz_i constructed from k_i available channels and n_i occupied spectrum, the average number of hash function computations needed by the secondary users to obtain the solution is:*

$$E[Pz_i] = 2^{x-1}\binom{n_i + k_i}{k_i} \tag{26}$$

Proof: To solve the puzzle, the user must obtain the exact availability of all the $(k_i + n_i)$ channels. The total number

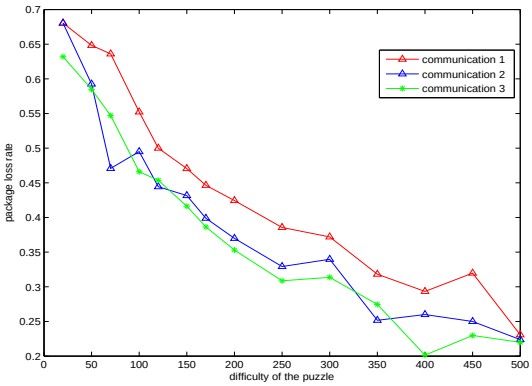


Fig. 6: Efficiency of client puzzle scheme

of candidates of Y is $w = 2^{k_i+n_i}$. However, su_i knows that there are k_i available channels and n_i unavailable channels. Therefore, the number of candidates of Y is reduced to $w = \binom{n_i+k_i}{k_i}$. On other hand, since r a x-length random digit, thus, the total number of candidates of r is 2^x . Since the most effective way to compute the inverse of hash function is the brute-force testing of all the possible inputs. Thus, the expected trial number $E[Pz_i]$ is:

$$E[Pz_i] = 2^x \sum_{j=1}^{w} \left(\frac{j}{w-j+1} \prod_{z=1}^{j-1} \frac{w-z}{w-z+1}\right) = 2^{x-1} \cdot (1+w) \tag{27}$$

According to *Theorem III*, since k_i is a constant, we can adjust the difficulty of Pz_i by modifying the n_i and the length of r according to SU's likelihood of being a jammer.

USRP based Experimental Evaluation We have implemented an USRP based experiment to demonstrate the defense efficiency of our proposed client puzzle scheme. *Fig. 2* shows the USRP prototype used in the experiment. There are 7 USRP devices used in the experiment, one of them is jammer and the other 6 devices are used to constitute 3 pair wireless communications. Moreover, each pair occupies a unique channel to prevent interfering other wireless communications. Each USRP device is driven by a PC with LABVIEW. Besides, one PC is used to simulate the DB who generates the SAI wrapped by the client puzzle.

Experiment Result: To verify the anti-jamming effect of client puzzle, we gradually increase the difficulty of the puzzle (measured as the average time(ms) to solve the puzzle) and check the packet loss rate of the three sessions. *Fig. 6* shows that the packet loss ratio decreases from nearly 70% to 20% when the difficulty of the puzzle is increased from 10ms to 500ms. Therefore, we can conclude that the client puzzle scheme is an effective way to mitigate the network loss caused by jamming attack.

2) Client anti-jamming Module: The *Client anti-jamming Module* is established in the SUs' side. SUs use the module to detect the jamming attack and submit the jamming event report to $\mathcal{DB}$, and SUs also can use this module to solve the client puzzle for the SAI. The senders and the receivers of the wireless communication can use the module to implement the traditional anti-jamming strategies such as FHSS or DSSS when they share the spreading codes or hopping sequences.

Traditional anti-jamming techniques like FHSS and DSSS [12] can enable the wireless communications in the presence of jammers. On the other hand, these spread spectrum technologies can serve as the complement of the proposed jammer inference and jamming attack detection for the following reasons:

- **Accelerating Jammer Inference:** When FHSS is adopted by the SUs, the transmitting radio signals will switch a carrier among multiple frequency channels. For a reactive jammer, it has to follow the sender to jam the switched channels to ensure a successful jamming attack towards a particular SU. The jammed channels will leak more information of the jammer, which will help to infer the jammer's identify.
- **Increasing Difficulty of Launching Jamming Attack:** Similarly, when DSSS is adopted, bits or symbols at the transmitter are spread to higher-order chip sequences while maintaining the same signal power. Therefore, to achieve the same jamming effect, the jammer has to use the higher jamming power, which makes the jamming detection easier and more accurately performed by the aggregated jamming detection proposed in this paper.

Note that, for any communication systems, the use of spread spectrum techniques inevitably brings extra cost in bandwidth and complicates the implementation. Thus, depending on the system operation focus (e.g., trade-off between detection accuracy and overhead), the network administrator may decide whether or not to turn on the spread spectrum option during the jammer detection phase.

TABLE III: False positive and False negative for different rounds of inference

Rounds	FP(%)	FN(%)	Rounds	FP(%)	FN(%)
50	0.816	100	300	3.88	20
100	4.08	90	350	3.67	0
150	4.69	60	400	3.06	0
200	5.10	50	450	1.83	0
250	5.71	40	500	1.42	0

V. Evaluations of The Proposed jDefender

We have implemented the *Query based Jamming Attack*, *Jammer Inference Model* and *Client Puzzle Generation Module* based on USRP. In this section, we will implement the total jDefender in PC to demonstrate the defense efficiency.

Experiment Setup: We implement jDefender based on a 64-bit PC with Intel i5 CPU of 2.8 GHz and 4G memory. The programming environment is Python 2.7.6. We assume that the spectrum is divided into 50 different channels $\{ch_0, ch_2, \cdots, ch_{49}\}$, and there are 500 SUs sharing these channels in the database-CRNs, and 10 cooperative jammers hiding among these SUs. In our experiment, the jammers are assumed to be rational and they will implement a random jamming strategy with a probability of 0.5. Besides, we will implement two channel allocation methods in jDefener to show

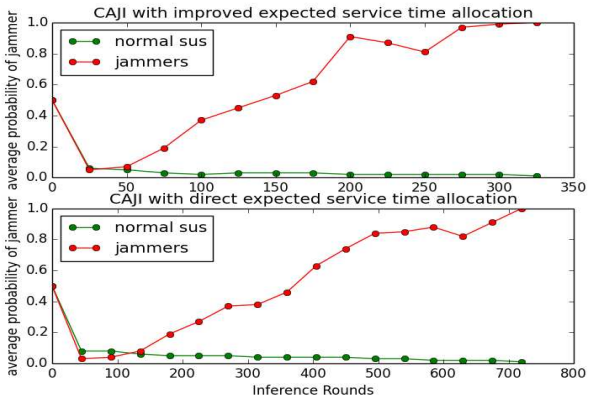


Fig. 7: Performance of CAJI scheme with different channel allocation method

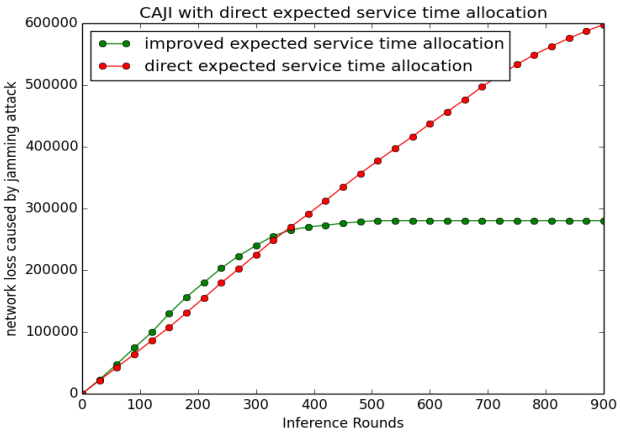


Fig. 8: Network loss caused by jamming attack during jDefender with different channel allocation method

the efficiency of our proposed improved channel allocation method.

Experiment Results: $Fig.7.$ shows the performance of CAJI scheme with different channel allocation methods. It can be derived that the average likelihood of being a jammer inferred by CAJI scheme with improved expected service allocation is exactly 1.0 after 350 rounds of inference, which means all the jammers are identified successfully. Table III. shows the false positive ratio and false negative ratio vary with the rounds of inference. It is observed that the false negative ratio decreases monotonously along with the increase of the inference rounds, and the value of false negative will reach 0 after 350 rounds of inference. It is also pointed out that the maximal false positive ratio is only 5.71%, which means that CAJI is friendly to the normal SUs, and the value of false positive will decrease gradually after 250 rounds of inference.

$Fig.8$ shows the network loss caused by jamming attack when jDefender adopts different channel allocation methods. We compare the improved channel allocation with directed channel allocation in terms of the network loss under the jamming attack. It is observed that jDefender with directed allocation can cause nearly double network loss compared to jDefender with optimal allocation. Therefore, it can be concluded that the improved channel allocation method can significantly reduce the total network loss caused by the jamming attack. On the other hand, according to **Theorem II**, it can be derived that CAJI can also benefit from the improved channel allocation method since it always allocates shorter expected service time when the channel is shared by more SUs. $Fig.8$ shows that CAJI with the improved channel allocation only needs less than 350 rounds of inference to identify all jammers while CAJI with directed allocation needs more than 750 rounds of inference. In Table III, we summarize the False positive and False negative for different rounds of inference.

VI. Related Works

The existing research on cognitive radio networks mainly include thwarting Primary User Emulation (PUE) Attack [28], [29], Spectrum Sensing Data Falsification (SSDF) Attack [30], Location Privacy issues and stimulating selfish behaviors in collaborative sensing [31], [32], [33], [34], [35]. The Internet Engineering Task Force (IETF) is currently working on the development of a Protocol for Access to Whitespace Spectrum (PAWS), which is used to request resources from the white space database [21]. It addresses the security issues of impersonation and man-in-the-middle attacks by using digital certificates and strong SU authentication. It also suggested using Transportation Layer Security (TLS) protocol to provide user authentication and data transportation.

Location privacy issue in the context of Database Driven Cognitive Radio Networks is gaining the interest from the society. In [18], the issue of location information leakage of SUs during the spectrum query process have been firstly pointed out and a novel privacy-preserving spectrum query and allocation framework has been proposed. In [36], [37], it extends the location privacy issues from SUs to primary users. In particular, the malicious secondary users, through seemingly innocuous queries to the database, can determine the types and locations of incumbent systems operating in a given region of interest, and thus compromise the incumbents operational privacy. It proposes a series of privacy-preserving mechanisms that allow the operation of white space database while preserving the operational privacy of the primary users.

In [38], it investigates the location spoofing attacks in database-driven CRNs, in which the adversary can compromises SUs GPS localization system, which results in SUs querying the database with false locations and obtaining incorrect spectrum information. It proposes corresponding spoofing attack detection and countermeasure solutions. In [39], it proposes a novel infrastructure-based approach that relies on the existing WiFi or Cellular network Access Points (or AP) to provide privacy-preserving location proof.

Different from the existing researches, we present a novel jamming attack in database-driven CRNs and then propose an anti-jamming framework to thwart this jamming attack based on the propposed jammer inference algorithm.

VII. Conclusion

In this paper, we have identified a novel jamming attack, Query based Jamming Attack, in database-driven CRNs. To thwart this attack, we have proposed a novel Jammer Inference based Jamming Defense jDefender, which is comprised of anti-jamming on spectrum query phase including client puzzle, channel allocation, and other anti-jamming strategies during the wireless transmission. We have also introduced a novel Channel Allocation based Jammer Inference (CAJI) with Sequential Bayesian Jammer Inference to infer the identities of jammers even under collusion jamming attack. For our future work, we will further study how to improve the security and privacy of database-driven CRNs under various attacks.

References

[1] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, and S. Mohanty, "Next generation dynamic spectrum access cognitive radio wireless networks: a survey," *Computer Networks*, vol. 50, no. 13, pp. 2127-2159, 2006.

[2] N. Zhang, H. Liang, N. Cheng, Y. Tang, J.W. Mark, and X. Shen, "Dynamic Spectrum Access in Multi-Channel Cognitive Radio Networks", *IEEE J. Selected Areas of Communications*, vol.32, no.11, pp.2053-2064, 2014.

[3] N. Zhang, N. Cheng, N. Lu, H. Zhou, J.W. Mark, and X. Shen, "Risk-aware Cooperative Spectrum Access for Multi-Channel Cognitive Radio Networks", *IEEE J. Selected Areas in Communications*, vol.32, no.3, pp.516-527, March 2014.

[4] F. C. Commission et al., "Third memorandum opinion and order," FCC 12, 2012.

[5] R. Murty, R. Chandra, T. Moscibroda, and P. Bahl, "Senseless: A database-driven white spaces network," *IEEE Trans. on Mobile Computing*, vol. 11, no. 2, pp. 189-203, 2012.

[6] H. Zhou, N. Cheng, N. Lu, L. Gui, D. Zhang, Q. Yu, F. Bai, and X. Shen, "WhiteFi Infostation: Engineering Vehicular Media Streaming with Geolocation Database", *IEEE J. Selected Areas of Communications*, to appear in 2016.

[7] H. Venkataraman and G.-M. Muntean, "Cognitive radio and its application for next generation cellular and wireless networks," Springer, 2012.

[8] [Online]. Available: www.google.com/get/spectrumdatabase/

[9] Q. Wang, K. Ren, and P. Ning, "Anti-jamming communication in cognitive radio networks with unknown channel statistics," in Proc. of *IEEE ICNP'11*, 2011.

[10] Y. Wu, B. Wang, K. R. Liu, and T. C. Clancy, "Anti-jamming games in multi-channel cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 1, pp. 4-15, 2012.

[11] H. Li and Z. Han, "Dogfight in spectrum: Combating primary user emulation attacks in cognitive radio systems, part i: Known channel statistics," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3566-3577, 2010.

[12] B. Sklar, Digital communications. Prentice Hall NJ, vol. 2, 2001.

[13] M. Strasser, S. Capkun, and M. Cagalj, "Jamming-resistant key establishment using uncoordinated frequency hopping," in Proc. of *IEEE Symposium on. Security and Privacy*, 2008.

[14] M. Strasser, C. Popper, and S. Capkun, "Efficient uncoordinated fhss anti-jamming communication," in Proc. of *ACM MobiHoc'09*, 2009.

[15] A. Liu, P. Ning, H. Dai, Y. Liu, and C. Wang, "Usd-fh: Jamming-resistant wireless communication using frequency hopping with uncoordinated seed disclosure," in Proc. of *IEEE MASS'10*, pp. 41-50, 2010.

[16] Y. Liu, P. Ning, H. Dai, and A. Liu, "Randomized differential dsss: Jamming-resistant wireless broadcast communication," in Prof. of *IEEE INFOCOM'10*, 2010.

[17] X. Wang and M. K. Reiter, "Mitigating bandwidth-exhaustion attacks using congestion puzzles," in Proc. of *ACM CCS'04*, 2004.

[18] Z. Gao, H. Zhu, Y. Liu, M. Li, and Z. Cao, "Location privacy in database-driven cognitive radio networks: Attacks and countermeasures," in Proc. of *IEEE INFOCOM'13*, 2013.

[19] Y. Dou, K. Zeng, H. Li, Y. Yang, B. Gao, C. Guan, K. Ren, S. Li, "P2-SAS: Preserving Users Privacy in Centralized Dynamic Spectrum Access Systems," in Proc. of *ACM MobiHoc16*, 2016.

[20] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in Proc. of *ACM MobiHoc'05*, 2005.

[21] P. J. McCann, L. Zhu, V. Chen, J. Malyar, and S. Das, "Protocol to access white-space (paws) databases," 2014.[Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-paws-protocol/

[22] S. M. Mishra, A. Sahai, and R. W. Brodersen, "Cooperative sensing among cognitive radios," in Proc. of *IEEE ICC06*, 2006.

[23] G. Lin and G. Noubir, "On link layer denial of service in data wireless lans," *Wireless Communications and Mobile Computing*, vol. 5, no. 3, pp. 273-284, 2005.

[24] A. Sampath, H. Dai, H. Zheng, and B. Y. Zhao, "Multi-channel jamming attacks using cognitive radios," in Proc. of *IEEE ICCCN'07*, 2007.

[25] W. Zhang, R. K. Mallik, and K. Letaief, "Optimization of cooperative spectrum sensing with energy detection in cognitive radio networks," *IEEE Trans. on Wireless Communications*, vol. 8, no. 12, pp. 5761-5766, 2009.

[26] S. Martin, "Sequential bayesian inference models for multiple object classification," in Proc. of *IEEE Information Fusion (FUSION'11)*, 2011.

[27] T. Aura, P. Nikander, and J. Leiwo, "Dos-resistant authentication with client puzzles," in *Security Protocols*, LNCS 2133, pp. 170-177, 2001.

[28] R. Chen, J. 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, Jan. 2008.

[29] Y. Liu, P. Ning, H. Dai, "Authenticating Primary Users Signals in Cognitive Radio Networks via Integrated Cryptographic and Wireless Link Signatures," in Proc. of *IEEE Symposium on Security and Privacy 2010*, Oakland, CA, May 2010.

[30] O. Fatemieh, A. Farhadi, R. Chandra, and C. Gunter, "Using Classification to Protect the Integrity of Spectrum Measurements in White Space Networks," in Proc. of *NDSS'11*, 2011.

[31] Z. Gao, H. Zhu, S. Li, S. Du and X. Li, "Security and Privacy of Collaborative Spectrum Sensing in Cognitive Radio Networks," *IEEE Wireless Communications*, vol.19, no.6, 2012.

[32] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing and X. (Sherman) Shen, "Location Privacy Preservation in Collaborative Spectrum Sensing," in Proc. of *IEEE INFOCOM12*, Orlando, Florida USA, March 25-30, 2012.

[33] S. Li, H. Zhu, Z. Gao, X. Guan and K. Xing, "YouSense: Mitigating Entropy Selfishness in Distributed Collaborative Spectrum Sensing," in Proc. of *IEEE INFOCOM13*, 2013.

[34] Yunlong Mao, Tingting Chen, Yuan Zhang, Tiancong Wang, Sheng Zhong, "Protecting Location Information in Collaborative Sensing of Cognitive Radio Networks," in Proc. of *ACM MSWiM '15*, 2015.

[35] X. Jin and Y. Zhang, "Privacy-Preserving Crowdsourced Spectrum Sensing," in Proc. of *IEEE INFOCOM'16*, 2016.

[36] B. Bahrak, S. Bhattarai, A. Ullah, J-M Park, J. Reed, D. Gurney,"Protecting the Primary Users Operational Privacy in Spectrum Sharing," In Proc. of *IEEE DySPAN14*, 2014.

[37] Matthew Clark, Konstantinos Psounis, "Can the Privacy of Primary Networks in Shared Spectrum be Protected?," in Proc. of *IEEE INFOCOM'16*, 2016.

[38] K. Zeng, S. Ramesh and Y. Yang, "Location Spoofing Attack and Its Countermeasures in Database-Driven Cognitive Radio Networks," in Proc. of *IEEE CNS'14*, 2014.

[39] Y. Li, L. Zhou, H. Zhu and L. Sun, Privacy-preserving Location Proof for Securing Large-scale Database-driven Cognitive Radio Networks, IEEE Internet of Things Journal, to appear in 2016.

[40] R. Chen, J. M. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in Proc. of *INFOCOM 2008*, Phoenix, AZ, 13-18 Apr. 2008.

[41] G. Ding, J. Wang, Q. Wu, L. Zhang, Y. Zou, Y. D. Yao, and Y. Chen,"Robust spectrum sensing with crowd sensors," *IEEE Trans. Commun.*, vol. 62, no. 9, pp. 3129-3143, Sep. 2014.

[42] P. Kaligineedi, M. Khabbazian, and V. K. Bhargava, "Malicious user detection in a cognitive radio cooperative sensing system," *IEEE Trans. Wireless Commun.*, vol. 9, no. 8, pp. 2488-2497, Aug. 2010.

Appendix A
Proof of Theorem I

$$E[\gamma]_{loss} = los_f \cdot D_f + los_m \cdot D_m \tag{28}$$

$$= los_f \cdot \sum_{l=\gamma}^{n} \binom{n}{l} p_f^l (1-p_f)^{n-l} + los_m \cdot (1 - \sum_{l=\gamma}^{n} \binom{n}{l} p_m^{n-l} (1-p_m)^l)$$

We can regard the $E[\gamma]_{loss}$ as a function of γ. Thus, to find the optimal γ, we have:

$$\frac{\partial E[\gamma]_{loss}}{\partial \gamma} \approx E[\gamma+1]_{loss} - E[\gamma]_{loss} \tag{29}$$

$$= \binom{n}{\gamma}(los_m \cdot p_m^{n-\gamma}(1-p_m)^{\gamma} - los_f \cdot p_f^{\gamma}(1-p_f)^{n-\gamma})$$

The optimal γ can be obtained when:

$$\frac{\partial E[\gamma]_{loss}}{\partial \gamma} \approx E[\gamma+1]_{loss} - E[\gamma]_{loss} = 0 \tag{30}$$

$$\Leftrightarrow \quad los_m p_m^{n-r}(1-p_m)^r = los_f p_f^r (1-p_f)^{n-r} \tag{31}$$

Thus, we have got the value of optimal γ:

$$\gamma = \lceil \frac{\ln(los_f/los_m) + n\ln((1-p_f)/p_m))}{\ln((1-p_m)(1-p_f)/(p_f p_m))} \rceil \tag{32}$$

Since $\gamma \leq n$, thus we have

$$\gamma = min\{n, \lceil \frac{\ln(los_f/los_m) + n\ln((1-p_f)/p_m))}{\ln((1-p_m)(1-p_f)/(p_f p_m))} \rceil\} \tag{33}$$

Appendix B
Proof of Theorem II

Assume $\mathcal{C}_{t,loc}^i$ is the available channels corresponding to su_i, the defined optimization problem is to find the expected service time $et_{t,loc}^i$ to minimize the value of $los_{t,loc}^{jm}$ under the constraint condition expressed equations(21)-(24).

For the ease of presentation, we denote $et = et_{t,loc}^i$, $\mathcal{C} = \mathcal{C}_{t,loc}^i$, $un = ud_{loc}^t$, $Q = Q_{t,loc}^i$. Since we have $et[ch] > 0$ and $un[ch] > 0 (ch \in \mathcal{C})$, according to the fundamental inequality, we have the following equations:

$$\sum_{ch\in\mathcal{C}} un[ch]\cdot et[ch] \sum_{ch\in\mathcal{C}} \frac{1}{un[ch]^2} \geq (\sum_{ch\in\mathcal{C}} \sqrt{\frac{et[ch])}{un[ch]}})^2 \tag{34}$$

According to the properties of the fundamental inequality, the right side of equation (34) is equal to the left side if and only the following equation holds

$$\forall ch, ch'(ch \neq ch') \in \mathcal{C}$$
$$\frac{un[ch]\cdot et[ch]}{un[ch']\cdot et[ch']} = \frac{1/un[ch]^2}{1/un[ch']^2} \Leftrightarrow \quad \frac{et[ch]}{et[ch']} = \frac{un[ch']^3}{un[ch]^3} \tag{35}$$

Let ch_1 be a specific channel in $\mathcal{C}$. Then we have the following equations derived from Equation (35)

$$\forall ch \in \mathcal{C}, \frac{\sqrt{et[ch]/un[ch]}}{\sqrt{et[ch_1]/un[ch_1]}} = \frac{un[ch_1]^2}{un[ch]^2} \tag{36}$$

By combining with equation (35), we have

$$\forall ch \in \mathcal{C}, \sqrt{\frac{et[ch]}{un[ch]}} = \frac{un[ch_1]^2}{un[ch]^2}\sqrt{\frac{et[ch_1]}{un[ch_1]}} \tag{37}$$

We plug equation (37) into equation (34). To facilitate the computations, we let $\mathcal{O} = \sum_{ch'\in\mathcal{C}} \frac{1}{un[ch']^2}$. Then we can obtain the following equation:

$$\sum_{ch\in\mathcal{C}} un[ch]\cdot et[ch]\cdot\mathcal{O} = \frac{et[ch_1]}{un[ch_1]} un[ch_1]^4\cdot\mathcal{O}^2$$
$$\Rightarrow \frac{et[ch_1]}{un[ch_1]} = \frac{\sum_{ch\in\mathcal{C}} un[ch]\cdot et[ch]}{un[ch_1]^4\cdot\mathcal{O}} \tag{38}$$

By combining the equation (37) with equation (38), we have the following equation:

$$\forall ch \in \mathcal{C}, \frac{et[ch]}{un[ch]} = \frac{\sum_{ch'\in\mathcal{C}} un[ch']\cdot et[ch']}{ud[ch]^4\cdot\mathcal{O}} \tag{39}$$

We let $\mathcal{L} = Q/rs$. With the equation (14) and the constraint condition (21), we have the following inequation:

$$\sum_{ch\in\mathcal{C}} \frac{et[ch]}{un[ch]} = \sum_{ch\in\mathcal{C}} \frac{\sum_{ch'\in\mathcal{C}} un[ch']\cdot et[ch']}{un[ch]^4\cdot\mathcal{O}} \geq \mathcal{L} \tag{40}$$

$$\Rightarrow \sum_{ch'\in\mathcal{C}} un[ch']\cdot et[ch'] \geq \frac{\mathcal{L}}{\sum_{ch\in\mathcal{C}} 1/(un[ch]^4\cdot\mathcal{O})} \tag{41}$$

According to equation (13), equation(41) gives the minimum expected loss caused by the jammer in the worst condition.

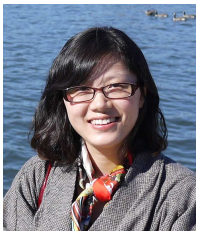
Let $los_{min}^{jm} = \frac{\mathcal{L}}{\sum_{ch\in\mathcal{C}} 1/(un[ch]^4\cdot\mathcal{O})}$. Given equation (39), if the loss achieves los_{min}^{jm}, the allocated expected service time of each channel can be expressed by the following equation.

$$\forall ch_{\in}\mathcal{C}, et[ch] = \frac{los_{t,loc,min}^{jm,i}}{un[ch]^3\cdot\mathcal{O}} \tag{42}$$

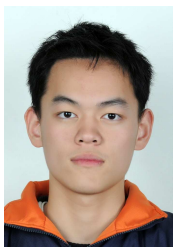
Haojin Zhu (S'16, M'09) received his B.Sc. degree (2002) from Wuhan University (China), his M.Sc.(2005) degree from Shanghai Jiao Tong University (China), both in computer science and the Ph.D. in Electrical and Computer Engineering from the University of Waterloo (Canada), in 2009. His current research interests include network security and data privacy. He published 33 international journal papers, including IEEE Trans. on Parallel and Distributed Systems, IEEE Trans. on Mobile Computing, IEEE Trans. on Wireless Communication, IEEE Trans. on Vehicular Technology, IEEE Wireless Communications, IEEE Communications, and 50 international conference papers, including ACM CCS, ACM MOBICOM, ACM MOBIHOC, IEEE INFOCOM, IEEE ICDCS, IEEE GLOBECOM, IEEE ICC, IEEE WCNC. He received a number of awards including: IEEE ComSoc Asia-Pacific Outstanding Young Researcher Award (2014), Top 100 Most Cited Chinese Papers Published in International Journals (2014), Supervisor of Shanghai Excellent Master Thesis Award (2014), Distinguished Member of the IEEE INFOCOM Technical Program Committee (2015), Outstanding Youth Post Expert Award for Shanghai Jiao Tong University (2014), SMC Young Research Award of Shanghai Jiao Tong University (2011). He was a co-recipient of best paper awards of IEEE ICC (2007) and Chinacom (2008) as well as IEEE GLOBECOM Best Paper Nomination (2014). He serves as the Associate/Guest Editor of IEEE Internet of Things Journal, IEEE Wireless Communications, IEEE Network, and Peer-to-Peer Networking and Applications.

Chenliaohui Fang received his B.S. degree from Xidian University, Xi'An in 2013. He is currently a master student in computer science department of Shanghai Jiao Tong University. His research interests are wireless network security and network privacy.

Yao Liu received the Ph.D. degree in computer science from North Carolina State Univ. in 2012. She is now an assistant professor at the Dept. of Computer Science and Engineering, Univ. of South Florida, Tampa, FL. Dr. Lius research is related to computer and network security, with an emphasis on designing and implementing defense approaches that protect emerging wireless technologies from being undermined by adversaries. Her research interest also lies in the security of cyber-physical systems, especially in smart grid security. She was the recipient of Best Paper Award for the 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems.



Cailian Chen (S03CM06) received the B.Eng. and M.Eng. degrees in automatic control from Yanshan University, China, in 2000 and 2002, respectively, and the Ph.D. degree in control and systems from City University of Hong Kong, Hong Kong SAR in 2006. She joined Department of Automation, Shanghai Jiao Tong University, in 2008 as an Associate Professor. She is now a Full Professor. Her research interests include distributed estimation and control of network systems, wireless sensor and actuator network, multi-agent systems, and intelligent control systems. She has authored and/or coauthored 2 research monographs and over 80 referred international journal and conference papers. She is the inventor of 20 patents. Dr. Chen received the IEEE TRANSACTIONS ON FUZZY SYSTEMS Outstanding Paper Award in 2008. She was one of the First Prize Winners of Natural Science Award from the Ministry of Education of China in 2007. She was honored as New Century Excellent Talents in University of China and Shanghai Rising Star in 2013, Shanghai Pujiang Scholar, Chenguang Scholar, and SMC Outstanding Young Staff of Shanghai Jiao Tong University in 2009. She is a Member of IEEE. She serves as an Associate Editor of Peer-to-peer Networking and Applications (Springer), ISRN Sensor Networks, and Scientific World Journal (Computer Science), and TPC member of several flagship conferences including IEEE Globecom, IEEE ICC and IEEE WCCI.



Mengyuan Li is currently an undergraduate student in Shanghai Jiao Tong University. His research interests are wireless network security and network privacy.

Xuemin (Sherman) Shen (IEEE M97-SM02-F09) received the B.Sc.(1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering. He is a Professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. He is also the Associate Chair for Graduate Studies. Dr. Shens research focuses on resource management in interconnected wireless/wired networks, wireless network security, social networks, smart grid, and vehicular ad hoc and sensor networks. He is an elected member of IEEE ComSoc Board of Governor, and the Chair of Distinguished Lecturers Selection Committee. Dr. Shen served as the Technical Program Committee Chair/Co-Chair for IEEE Globecom16, Infocom14, IEEE VTC10 Fall, and Globecom07, the Symposia Chair for IEEE ICC10, the Tutorial Chair for IEEE VTC'11 Spring and IEEE ICC08, the General Co-Chair for ACM Mobihoc15, Chinacom07 and QShine06, the Chair for IEEE Communications Society Technical Committee on Wireless Communications, and P2P Communications and Networking. He also serves/served as the Editor-in-Chief for IEEE Network, Peer-to-Peer Networking and Application, and IET Communications; a Founding Area Editor for IEEE Transactions on Wireless Communications; an Associate Editor for IEEE Transactions on Vehicular Technology, Computer Networks, and ACM/Wireless Networks, etc.; and the Guest Editor for IEEE JSAC, IEEE Wireless Communications, IEEE Communications Magazine, and ACM Mobile Networks and Applications, etc. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004, 2007, 2010, and 2014 from the University of Waterloo, the Premiers Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, and a Distinguished Lecturer of IEEE Vehicular Technology Society and Communications Society.