Hindawi
Mobile Information Systems
Volume 2018, Article ID 5032934, 9 pages
https://doi.org/10.1155/2018/5032934

Research Article

Asynchronous Channel-Hopping Scheme under Jamming Attacks

Yongchul Kim,[1] Young-Hyun Oh,[2] and Jungho Kang[1]

[1]*Korea Military Academy, Seoul, Republic of Korea*
[2]*Department of Computer Science, North Carolina State University, Raleigh, NC, USA*

Correspondence should be addressed to Yongchul Kim; kyc6454@gmail.com

Received 6 September 2017; Accepted 29 November 2017; Published 19 February 2018

Academic Editor: Francesco Gringoli

Cognitive radio networks (CRNs) are considered an attractive technology to mitigate inefficiency in the usage of licensed spectrum. CRNs allow the secondary users (SUs) to access the unused licensed spectrum and use a blind rendezvous process to establish communication links between SUs. In particular, quorum-based channel-hopping (CH) schemes have been studied recently to provide guaranteed blind rendezvous in decentralized CRNs without using global time synchronization. However, these schemes remain vulnerable to jamming attacks. In this paper, we first analyze the limitations of quorum-based rendezvous schemes called asynchronous channel hopping (ACH). Then, we introduce a novel sequence sensing jamming attack (SSJA) model in which a sophisticated jammer can dramatically reduce the rendezvous success rates of ACH schemes. In addition, we propose a fast and robust asynchronous rendezvous scheme (FRARS) that can significantly enhance robustness under jamming attacks. Our numerical results demonstrate that the performance of the proposed scheme vastly outperforms the ACH scheme when there are security concerns about a sequence sensing jammer.

1. Introduction

The unlicensed spectrum band has become overcrowded as the demand for smart phones and portable devices has increased, while the licensed spectrum band is always underutilized (e.g., the occupancy of the licensed spectrum is less than 6% [1]). To solve the spectrum scarcity issue in wireless communications, the Federal Communications Commission (FCC) allows the unlicensed users (i.e., secondary users) to make use of the licensed spectrum as long as they do not interfere with the licensed users (i.e., primary users) [2, 3]. This promising paradigm introduces the use of cognitive radio networks (CRNs) as a key technology for opportunistically exploiting the spectrum. In CRNs, secondary users (SUs) must establish a link before communicating with each other. In other words, two SUs should meet on a common channel, which is not occupied by primary users (PUs), to exchange handshake information. This is often referred to as a rendezvous process. The best known approach to solving the rendezvous problem is using a common control channel (CCC) [4, 5]. The advantages in using this approach are the simple implementation and management of a rendezvous. However, maintaining a CCC in CRNs is not feasible since the availability of the spectrum will change dynamically over time. Moreover, this approach may result in a bottleneck, and it potentially creates a single point of failure. Therefore, the rendezvous process in a distributed manner without having any centralized controller or dedicated CCCs is preferable in a more practical scenario. This process is often referred to as a blind rendezvous. To achieve blind rendezvous, a channel-hopping (CH) technique is used as a fundamental strategy to visit the available channels. Most research regarding CH algorithms consider that the time is divided into slots; thus, a successful rendezvous can be achieved when two SUs hop on the same channel in the same timeslot. The number of timeslots that are required until the successful rendezvous after all SUs have begun their CH sequences is defined as time to rendezvous (TTR). According to the geographical locations of the SUs, the number of available channels sensed by each SU might not be the same. If all SUs have the same number of available channels, we call this the symmetric system. All others are asymmetric systems. For reliable performance in CRNs, most CH schemes must guarantee rendezvous in more than one channel within

a sequence period, and the TTR value must be bounded and small. To evaluate the performance of proposed CH schemes, two critical metrics are commonly used: the maximum TTR (MTTR) and the expected TTR (ETTR).

There has been a great deal of research directed toward providing guaranteed rendezvous as well as minimizing MTTR. In particular, quorum-based rendezvous algorithms are well-known schemes for providing some level of security under hostile jamming attack scenarios with low time latency. However, these schemes remain vulnerable to sophisticated jamming attacks. In this paper, we present a noble sequence sensing jamming attack (SSJA) in which the jammer can estimate the SU's channel-hopping sequences quickly and begin jamming the rest of the timeslots. We then examine the limitations of those quorum-based rendezvous schemes including frequency quorum rendezvous (FQR) [6] and asynchronous channel-hopping (ACH) [7] schemes under a sophisticated jamming attack. Since FQR requires time synchronization between SUs, we focus more on ACH rendezvous scheme for evaluating the effectiveness of SSJA. Moreover, we introduce our proposed fast and robust asynchronous rendezvous scheme (FRARS) that can reduce TTR as well as increase robustness significantly against jamming attacks.

The contributions of this paper are twofold: first, we analyze the limitations of the well-known quorum-based rendezvous scheme under a sophisticated jamming attack by introducing a novel SSJA model. In the SSJA, a jammer can detect the sender's entire CH sequence of ACH schemes on $N/2$ time frames, where N is the number of available channels. Thus, the rendezvous success rates of ACH schemes will be dramatically decreased under SSJAs. Second, we present our proposed FRARS in order to evaluate the performance under an SSJA. We include a theoretical analysis in addition to extensive numerical analysis under SSJAs. Our numerical results demonstrate that our proposed scheme outperforms others that are recently proposed. The balance of this paper is organized as follows. In Section 2, we review related work in CRNs. In Section 3, we introduce SSJA model and present several CH schemes including FQR and ACH as well as our proposed FRARS. The numerical analysis of our jamming attack for both ACH and FRARS and the results are discussed in Section 4. Section 5 concludes this paper.

2. Related Work

Due to the drawbacks of using a centralized controller or dedicated CCC, many studies have focused on blind rendezvous systems. To solve the blind rendezvous problem in CRNs, a purely random [8] or an improved random algorithm [9] provides a trivial CH algorithm where each SU hops from one channel to another among available channels in a purely random way. That is, when two SUs hop on the same channel at the same time by chance, rendezvous occurs. These schemes can be applied to almost any system but cannot guarantee a bounded TTR between any two SUs.

2.1. Synchronous Rendezvous Algorithms. To achieve guaranteed rendezvous in finite time, several algorithms have been proposed with the assumption of global time synchronization [10–12]. Bahl et al. [10] proposed a link-layer protocol called Slotted Seeded Channel Hopping (SSCH) that increases the capacity of an IEEE 802.11 network by utilizing frequency diversity. Krishnamurthy et al. [11] proposed a two-phase autoconfiguration algorithm that enables SUs to dynamically compute the globally common channel set in a distributed manner. Bian et al. [12] introduced quorum-based two CH schemes, namely, M-QCH and L-QCH: the first design ensures ETTR by minimizing the MTTR and the second design guarantees the even distribution of the rendezvous points in terms of both time and frequency. However, these synchronous systems may not be feasible in certain types of networks, for example, ad hoc networks. Moreover, under the assumption of synchronization, the impact of a jamming attack can be significant.

2.2. Asynchronous Rendezvous Algorithms. To overcome these challenges, many asynchronous CH algorithms (i.e., one that does not require clock synchronization) have been proposed recently in the literature [13–21]. DaSilva and Guerreiro [13] proposed a sequence-based rendezvous and was later referred to as the generated orthogonal sequence (GOS) algorithm in which all SUs use the identical predefined CH sequences. This algorithm is used with interspersed permutation channels to guarantee rendezvous even when SUs are not synchronized. Theis et al. [14] showed better performance than that of GOS in terms of ETTR by presenting modular clock (MC) and modified modular clock (MMC) algorithms. In the MC system, the SUs can rendezvous any time although they independently generate their CH sequences by using prime number and rate (forward-hop). Lin et al. [15] proposed a jump-stay (JS) algorithm in which each SU has a jump and stay pattern to find a common channel. Intuitively, SUs jump on available channels during the jump pattern and stay on a specific channel during the stay pattern. The enhanced jump-stay (EJS) algorithm [16] was also proposed by the same authors to improve the MTTR and ETTR performances for asymmetric systems. Liu et al. [17] introduced the ring walk (RW) algorithm to guarantee asynchronous rendezvous by using the concept of velocity. The SUs in this scheme walk on the ring by visiting vertices of channels with different velocities. The higher velocity SU will eventually catch the lower velocity SU. Chuang et al. [18] presents a new alternate HOP-and-WAIT channel-hopping method (E-AHW) to minimize the time to rendezvous (TTR) than existing methods. In this method, each SU has a unique alternating sequence of HOP and WAIT. Most recently, Salehkaleybar et al. [22] proposed periodic jump rendezvous (PJR) and modified version of PJR (mPJR) algorithms for role-based and nonrole-based cases, respectively, in order to reduce TTR. Pu et al. [23] studied the dynamic rendezvous problem in CRNs where the status of the licensed channels varies over time by introducing the available channel probabilities. This work aims to propose more

realistic models under adversaries as a future work. Thus, the vulnerability against a sophisticated jamming attack is not addressed. The deterministic rendezvous sequence (DRSEQ) and channel rendezvous sequence (CRSEQ) algorithms proposed in [19, 20] provide fast asynchronous rendezvous under symmetric and asymmetric models, respectively. The upper bounds of MTTR of those algorithms are significantly small as shown in [24]. Yadav and Misra [21] develop an algorithm that generates a deterministic CH sequence, which guarantees rendezvous even faster than the DRSEQ algorithm. However, they are not applicable to jamming attack scenarios due to deterministic CH sequences.

2.3. Jamming Attack Scenarios. Most of the aforementioned algorithms focus on minimizing MTTR without using time synchronization in either symmetric or asymmetric scenarios. None of them are considered to be robust in jamming attack environments. In wireless communications, an adversary (enemy and jammer) is a malicious entity that can easily disrupt legitimate communications by intentionally injecting noise-like signals (or dummy packets) into the wireless medium. To alleviate this vulnerability problem in CRNs, quorum-based rendezvous schemes are proposed in [6, 7]. The FQR algorithm [6] exploits a quorum system where each SU independently constructs a random sequence by scrambling the sender's frequency quorum sequences for every frame as will be addressed in the next section. This makes jamming difficult and inefficient. An enhanced version of FQR, advanced FQR (AFQR) algorithm [6], adds more timeslots mapping random frequency channels at arbitrary positions within each frame to improve the rendezvous probability while it might degrade time overhead with the increased number of timeslots in a period. However, FQR only supports synchronous systems where any two SUs must start their sequences at the same time in order to rendezvous. Abdel-Rahman and Krunz [25] studied the rendezvous problem in the presence of an insider attack using a game-theoretic framework. This work showed that the rendezvous performance improves if the receiver and jammer are time synchronized and both have a common guess about the transmitter's strategy. However, the jammer model in this work is not a smart jammer but an insider jammer. Thus, the vulnerability against a sophisticated jamming attack is not addressed. Bian and Park [7] proposed a quorum-based ACH algorithm to ensure that the TTR is upper bounded even if the SU's clocks are asynchronous, and it maximizes the rendezvous probability between any pair of SUs by enabling rendezvous on every available channel. The sender and the receiver in an ACH algorithm independently generate their own sequences and rendezvous within a favorable amount of time compared to other schemes. Nevertheless, the ACH algorithm is significantly vulnerable to a sophisticated jamming attack. A survey paper [26] on jamming and antijamming techniques shows the classification of jammers. As a reactive channel-hopping jammer, we introduce an SSJA model in this paper to show how effectively it attacks the ACH system by adding more sophisticated capabilities such as estimating the SU's CH sequence within a short time. To overcome this vulnerability against SSJA, we proposed a FRARS algorithm that employs randomized permutation in every period. Due to the random features of FRARS, it is unfeasible for the SSJA to estimate the CH sequences. Thus, the effectiveness of the SSJA is negligible for the FRARS. Our proposed scheme is comparable to the ACH algorithm since both the sender and the receiver in FRARS independently generate their own sequences like those in ACH. The performance results of FRARS as compared with ACH will be addressed in Section 4.

3. Channel-Hopping Schemes

In this section, we present two well-known quorum-based rendezvous schemes, FQR and ACH algorithms, with a brief definition of a quorum system. Then, we introduce the SSJA model to show how effectively it attacks quorum-based rendezvous schemes. We also present our FRARS algorithm to enhance robustness against jamming attack and compare the effectiveness of the SSJA on ACH and FRARS.

3.1. The Quorum System. A quorum system has two fundamental properties, that is, the intersection property and the rotation closure property [27]. All quorum systems hold the intersection property, but the rotation closure property may not be present in some cases. The FQR exploits a cyclic quorum system [28] to design a set of hopping sequences. We provide those definitions in this subsection, and we borrow all the terminologies defined in [6, 7, 27, 28].

Definition 1. Given a finite universal set $U = Z_N = \{0, 1, \ldots, N-1\}$ of N elements, a quorum system Q under U is a collection of nonempty subsets of U, which satisfies the intersection property:

$$p \cap q \neq \varnothing, \forall p, q \in Q. \tag{1}$$

Each p or $q \in Q$ is called a quorum, and Z_N denotes the set of nonnegative integers less than n. Given a nonnegative integer k and a quorum q in a quorum system Q under the universal set U, we define

$$\text{rotate}(q, k) = \{(i + k) \bmod n \mid i \in q\}. \tag{2}$$

Definition 2. A quorum system Q under U has the rotation closure property if the following holds:

$$\forall p, q \in Q, \forall k \in [0, \ldots, N-1], \text{rotate}(p, k) \cap q \neq \varnothing. \tag{3}$$

Definition 3. A set $D = \{a_1, \ldots, a_k\} \subset Z_N$, $a_i \in \{0, \ldots, N-1\}$, and $k \leq N$ is called a cyclic (N, k) difference set if for every $d \not\equiv 0 \bmod N$ there exists at least one ordered pair (a_i, a_j), where $a_i, a_j \in D$, such that $a_i - a_j \equiv d \bmod N$, that is, d is a difference value between two elements of D.

Definition 4. Given a (N, k) difference set $D = \{a_1, \ldots, a_k\} \subset Z_N$, a cyclic quorum system constructed by D is $Q = \{C_0, \ldots, C_{N-1}\}$, where $C_i = \{a_1 + i, a_2 + i, \ldots, a_k + i\} \bmod N, i = 0, \ldots, N-1$. For a (7, 3) difference set $D = \{1, 2, 4\} \subset Z_7$, for example, the cyclic quorum system is $Q = \{C_0, \ldots, C_6\}$, where $C_0 = \{1, 2, 4\}, C_1 = \{2, 3, 5\}, C_2 = \{3, 4, 6\}$, $C_3 = \{4, 5, 0\}, C_4 = \{5, 6, 1\}, C_5 = \{6, 0, 2\}$, and $C_6 = \{0, 1, 3\}$.

3.2. FQR Base System. The authors in [25] apply the abovementioned properties of quorum systems to design a FQR system. The frequency-hopping sequence of an SU in FQR is constructed by assigning frequencies to t timeslots in one period X, which is denoted by $X = \{x_0, \ldots, x_{t-1}\} = \{(0, f_0), \ldots, (t-1, f_{t-1})\}$, where $x_i \in X$ contains a tuple of (*timeslot index, frequency index*) and $f_i \in \{0, \ldots, N-1\}$ represents the frequency index at timeslot t in a period. In FQR, an SU generates two different hopping sequences: a sending sequence and a receiving sequence. If an SU has data to transmit, it follows a sending sequence, otherwise, a receiving sequence. For example, consider that a sender and a receiver use a (7, 3) different set, that is, $N = 7$ and $k = 3$. And, a cyclic quorum system $Q = \{C_0, \ldots, C_6\}$ is constructed from $D = \{1, 2, 4\}$. Thus, the sender and the receiver can select a random number and obtain a quorum $S = C_i$ and $R = C_j$, respectively (e.g., $S = C_0 = \{1, 2, 4\}, R = C_4 = \{5, 6, 1\}$). A sender node constructs a sending sequence X by assigning a frequency index to the timeslot i using $C_0 = \{c_0, c_1, c_2\} = \{1, 2, 4\} : x_i = (i, c_m)$, where $0 \le i \le k^2 - 1$ and $m = i \bmod k$. Then, it obtains $X = \{(0, 1), (1, 2), (2, 4), (3, 1), (4, 2), (5, 4), (6, 1), (7, 2), (7, 4)\}$. A receiver node constructs a receiving sequence Y by assigning frequency index to the timeslot i using $C_4 = \{c_0, c_1, c_2\} = \{5, 6, 1\} : y_i = (i, c_n)$, where $0 \le i \le k^2 - 1$ and $n = (i - (i \bmod k))/k$. Then, it obtains $Y = \{(0, 5), (1, 5), (2, 5), (3, 6), (4, 6), (5, 6), (6, 1), (7, 1), (7, 1)\}$. Additionally, permuting frequency indexes in each frame of X is done as a last step for constructing FQR sequences.

Figure 1(a) shows FQR sequences of both sender and receiver, respectively. It also shows that the sender and receiver rendezvous on frequency 1 at time T_6. The upper bound of TTR of FQR system is only k^2 timeslots, which is approximately, equal to N. In AFQR, one more timeslot is added into each frame of the X sequence, and it is assigned a random frequency f_i such that $f_i \notin C_k$, where C_k is a quorum selected by the sender. That is, the selected frequency f_i is inserted into a random timeslot in each frame as shown in Figure 1(c). Although this will increase the length of the time period (i.e., $k(k+1)$ timeslots), the expected number of rendezvous within $k(k+1)$ timeslots also increases to $1 + (k - 1/N - k)$. The selected frequency index 5 is added to the sequence in Figure 1(c); thus, one more rendezvous on frequency 5 at time T_0 is provided. The techniques used in FQR and AFQR such as scrambling the hopping sequence and inserting additional timeslots make it difficult for a jammer to predict the hopping sequences. However, these schemes are not appropriate in an emergency or tactical scenario where time synchronization between randomly meeting nodes cannot be assumed. Figures 1(b) and 1(d) show that a clock shift nullifies the guaranteed upper bound of rendezvous.

3.3. ACH Base System. In ACH system, the TTR between any pair of CH sequences is upper bounded without requiring clock synchronization by exploiting two properties of the array-based quorum systems: the intersection property and the rotation closure property. Let u denote the CH sequence of an SU and T denote a period of the CH sequence: $u = \{u_0, u_1, \ldots, u_i, \ldots, u_{T-1}\}$, where $u_i \in [0, N-1]$ is the channel index of sequence u in the ith timeslot of a CH period. The SUs in ACH generate their CH sequences by using an $N \times N$ array-based quorum system and assigning channel index values to quorums' columns or spans. The sender and receiver construct different CH sequences for rendezvous in ACH. The sender assigns N channel indexes to the N columns of an $N \times N$ array, $S[\cdot][\cdot]$, such that each column has different channel indexes and all array elements in the same column are assigned the same channel index. If we denote h_j the channel index value assigned to the jth column, where $j \in [0, N-1]$, the sequence of the sender will be constructed by the following way: $u_{i \cdot N+j} = h_j$, where $i, j \in [0, N-1]$. The receiver assigns N channel indexes to the N spans of an $N \times N$ array, $R[\cdot][\cdot]$, such that each span has a different channel index and all array elements in the same span are assigned the same channel index. Let s_k denote the kth span and h_k denote the channel index value assigned to the s_k, where $k \in [0, N-1]$. The sequence of the receiver will be constructed in the following way: $v_{i \cdot N+j} = h_k$ if $R[i][j] \in s_k$, where $i, j \in [0, N-1]$. Therefore, the sender's sequence u and receiver's sequence v are generated by using the two $N \times N$ arrays, and the period of the sequences will be N^2. Figure 2 shows an example of the ACH system when $N = 3$.

Without loss of generality, the sender's sequence u starts first and then i timeslots later, the receiver's sequence v starts. The operation rotate (u, i) yields a new CH sequence u^*, where all the elements in the same column are assigned the same channel index. Thus, u^* and v have N distinct rendezvous channels within a sequence period $T = N^2$. It is obvious that the asynchronous rendezvous problem is solved in ACH system since the sender and receiver CH sequences satisfy the rotation closure property.

3.4. The Sequence Sensing Jamming Attacks. To evaluate the performance of the ACH system under a sophisticated jamming attack, we present a noble sequence sensing jamming attack (SSJA) model in which a jammer can estimate the SU's CH sequences. Since one period of the sender's CH sequence consists of N repeated N timeslots ($T = N^2$), a jammer can effectively nullify the whole ACH system by estimating only N channels (one subperiod). We call this subperiod a frame, that is, one period of the ACH sequence consists of N frames. We assume that the number of available N channels does not change during a sequence period T to clearly illustrate the characteristics of the SSJA. We also assume that the jammer resides in the network

	One time period								
	Frame 1			Frame 2			Frame 3		
Time slot	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
X (quorum C_0)	2	1	4	4	1	2	*1*	4	2
Y (quorum C_4)	5	5	5	6	6	6	*1*	1	1

(a)

Time slot	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_0
Sender X	2	1	4	4	1	2	1	4	2	
Receiver X		5	5	5	6	6	6	1	1	1

(b)

	One time period											
	Frame 1				Frame 2				Frame 3			
Time slot	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}
X (quorum C_0)	*5*	2	1	4	4	5	1	2	*1*	4	2	5
Y (quorum C_4)	*5*	5	5	5	6	6	6	6	*1*	1	1	1

(c)

Time slot	T_0	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}	T_{11}	T_{12}
Sender X	5	2	1	4	4	5	1	2	1	4	2	5	
Receiver X		5	5	5	5	6	6	6	6	1	1	1	1

(d)

Figure 1: (a) An illustration of rendezvous in FQR, (b) a simple clock shift can nullify FQR rendezvous, (c) an illustration of rendezvous in AFQR, and (d) a simple clock shift can nullify AFQR rendezvous.

before the communication starts, that is, the jammer waiting for the sender starts its sequence and the sender does not know of the existence of the jammer. We also assume that the jammer has capabilities similar to those of the normal user with one transmitting channel and one listening channel. Thus, the jammer knows the number of N available channels whenever it changes due to the occupation of PUs. The jammer randomly selects one distinct listening channel and waits until the sender hops to that channel. As soon as the jammer detects one channel, it chooses another channel (not previously tried) for detecting the next channel. Therefore, on average, the jammer can detect two channels within a frame (N timeslots) of the sequence. For all detected channels, the jammer immediately jams those channels for the rest of the ACH sequence. Moreover, the average time for the jammer to compute the entire ACH sequence is $N/2$ frames. When $N-1$ channels are detected among N timeslots, the last channel is automatically detected. In other words, after an average of $(N/2)\times N$ timeslots, the jammer can completely jam the remaining channels. Therefore, our SSJA model can significantly decrease the rendezvous probability of the ACH scheme.

3.5. Proposed System. In this subsection, we introduce our proposed FRARS system that can enhance robustness significantly against jamming attacks by generating new CH sequences in every period. The sender and receiver construct different CH sequences for rendezvous in a way similar to ACH. When an SU has data to send, it follows a sending CH sequence. Otherwise, it follows a receiving CH sequence. The length of an SU's CH sequence period is only $2N-1$, thus $u=\{u_0,u_1,\ldots,u_i,\ldots,u_{2N-2}\}$, where $u_i\in[0,N-1]$ is the channel index of the sequence u in the ith timeslot of a CH period. Unlike the ACH scheme, the CH sequence will change in every period. Let I denote the slot number and u and v denote two CH sequences in FRARS. When there is k slot time difference between u and v as shown in Figure 3, we say that two SUs rendezvous in the ith timeslot on channel c when $u_{\{i=I \bmod 2N-1\}}=v_{\{j=(I-k) \bmod 2N-1\}}=c$, where $c\in[0,N-1]$.

The sending CH sequence employs a random permutation channel in the first N timeslots. And then, the reverse order of the random permutation excluding the last channel will be used for the next $N-1$ timeslots. The first N timeslots and the next $N-1$ timeslots are referred to as *permutation*

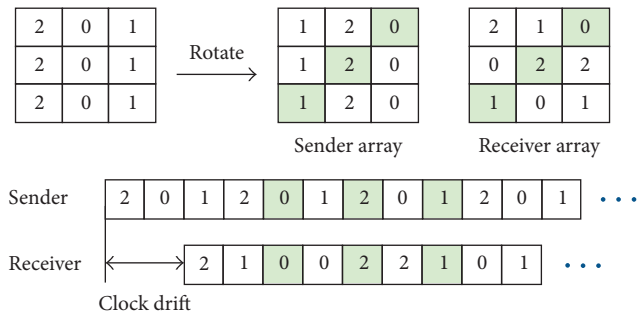

Figure 2: An example of ACH system when $N = 3$.

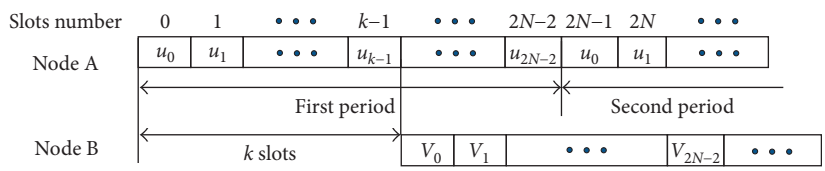

Figure 3: FRARS system model structure.

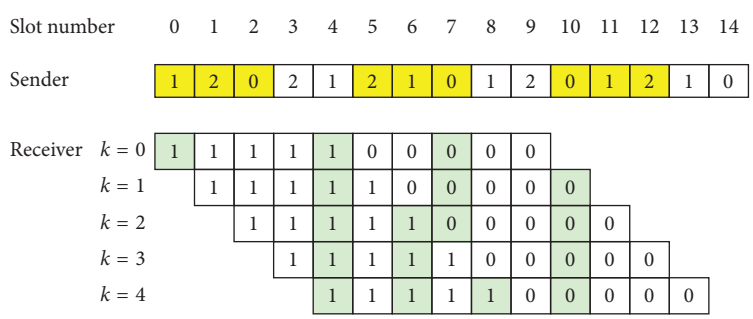

Figure 4: Illustration of FRARS when the number of available channel is 3.

parts and *reversed repetition parts*, respectively. If we represent the permutation part as $R = \{r_0, r_1, \ldots, r_i, \ldots, r_{N-1}\}$, where $r_i \in [0, N-1]$, the generation of sending CH sequence with N available channels can be expressed as

$$u_i = \begin{cases} r_i & \text{for } 0 \le i \le N-1, \\ r_{2N-2-i} & \text{for } N \le i \le 2N-2. \end{cases} \quad (4)$$

Figure 4 shows an illustration of rendezvous in FRARS when the number of available channels is 3. The sending sequence shows three periods, and the receiving sequence shows two periods. The shaded timeslots in the sending sequence represent the permutation parts, and the rest are the reversed repetition parts. For example, the sender visits channel 2 in slot number 3, which is equal to the one in slot number 1, and the sender visits channel 1 in slot number 4, which is equal to the one in slot number 0. The receiving sequence selects one random channel from N and stays on that channel during one period. The shaded timeslots in the receiving sequence indicate rendezvous. For all k, FRARS guarantees that any pair of sender and receiver nodes rendezvous within at most $2N-1$ slots. Since the sender's sequence is generated by using a randomized permutation of N channels for every period, it is not feasible for the sophisticated jammer to estimate when the sender and receiver might rendezvous.

4. Performance Evaluation

In this paper, we implemented the sequence sensing jamming attack (SSJA) model to evaluate its effectiveness against the asynchronous channel-hopping (ACH) [7] schemes. For the sake of simplicity, we did not consider multiple SU scenario, since it is more difficult to analyze the performance of rendezvous algorithms due to collision problem between SUs. Therefore, we focused on the symmetric scenario of ACH scheme in which two SUs have the same number of available channels. The two SUs do not know each other's existence, and they are not time synchronized. When they want to communicate with each other (i.e., one of them has data to send), they go through a rendezvous algorithm. During a rendezvous process, each SU has to hop every available channel according to their CH sequence until they successfully rendezvous on the same channel. In our implementation, the sender starts the communication first and the receiver can start at any timeslot within the N^2

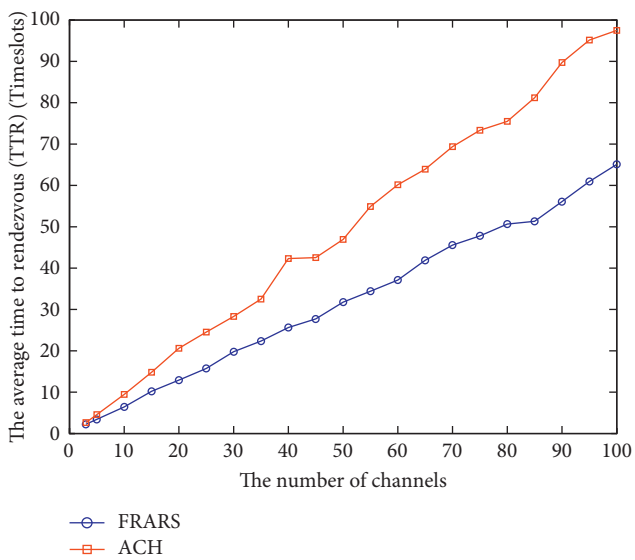

Figure 5: The average time to rendezvous (TTR) for ACH and FRARS when the available channels are $[3, \ldots, 100]$ with no jamming attacks.

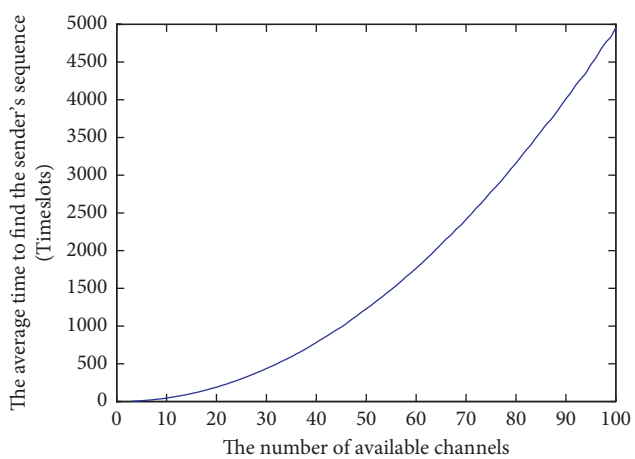

Figure 6: The average time to find the entire sender's CH sequence for ACH.

timeslots of the sender because there is no time synchronization. We also make a typical assumption that the SSJA jammer resides in the network and is listening on one channel before the communication starts. This is a normal situation since the jammer is trying to disrupt the communication and is invisible to the sender. Then, we implement our FRARS scheme to demonstrate that it is more robust against jamming attacks compared to the ACH scheme. In the jamming attack for the FRARS scheme, a jammer randomly selects one channel for each timeslot and jams the channel until the MTTR. Through our extensive numerical analysis under SSJA, we demonstrate that our proposed FRARS scheme outperforms the quorum-based state-of-the-art ACH scheme.

First, we compare the average time to rendezvous (TTR) for both ACH and FRARS schemes. Figure 5 gives the average TTR for ACH and FRARS schemes where the number of available channels N varies from 3 to 100 with no

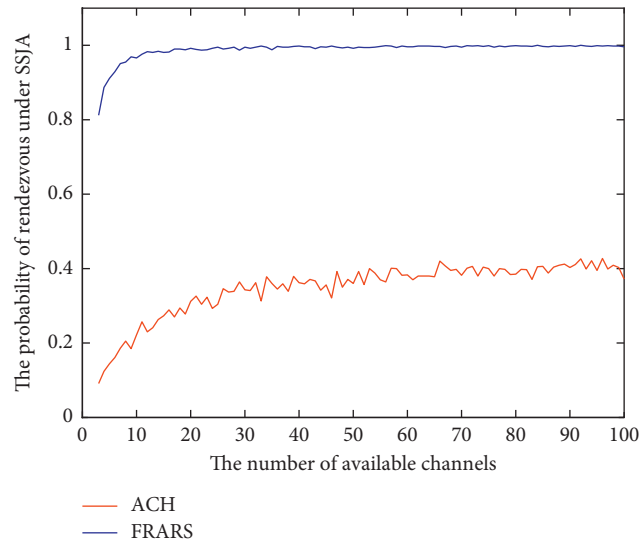

Figure 7: The probability of rendezvous for both ACH and FRARS under SSJA.

jamming attacks. We repeated the process 1000 times for each available channel N and calculated the average TTR for them. This figure shows that the average TTR for both ACH and FRARS schemes increases steadily as the number of available channels increases. Our numerical results for the ACH's TTR are close to its theoretical TTR $O(N)$ discussed in [7]. Overall, the average TTR for the FRARS system is lower than ACH system's TTR for all N.

Second, we implemented the expected time to find the entire sender's CH sequence for ACH where the number of available channels M varies from 3 to 100. We repeated the process 1000 times for each number of available channels and calculated the average time for finding the sender's CH sequence. Our numerical results show that the average time for finding the entire sender's CH sequence is close to an average of $\lfloor (N/2) \times N \rfloor$ timeslots as we discussed in Section 3.4. This means that there are no rendezvous after $\lfloor (N/2) \times N \rfloor$ timeslots so that the rendezvous success rates of ACH schemes will be dramatically decreased under our SSJA attack as we can see the average time to find the entire sender's CH sequence for ACH in (Figure 6).

Lastly, we compare the rendezvous probability for the ACH and FRARS schemes under SSJA. Figure 7 shows the probability of rendezvous for both ACH and FRARS systems. The rendezvous probability for the ACH scheme is less than 40% for almost all available channels under SSJA. Since there is no rendezvous after an average $\lfloor (N/2) \times N \rfloor$ timeslots, the rendezvous probability of the ACH scheme is dramatically decreased under the SSJA. However, the rendezvous probability for the FRARS scheme is almost steady and more than 80% for all the available channels N under jamming attacks.

5. Conclusion

In this paper, we examined the drawbacks of quorum-based state-of-the-art schemes such as FQR and ACH. In particular, the SSJA model is used to show how effectively it

attacks the ACH system. The sender's CH sequence in ACH is fully detected by the jammer within an average $(N/2) \times N$ timeslots, after that the jammer can completely jam the remaining channels. To remedy this jamming problem, we proposed a new FRARS algorithm that can significantly enhance robustness by allowing randomized permutation in every period of the sender's CH sequence. Our numerical results demonstrate that the rendezvous probability of the ACH under SSJA is dramatically decreased from 100% to less than 40%. On the other hand, the rendezvous probability of our FRARS is almost steady under jamming attacks and close to 100% as the number of available channels increases. Therefore, our FRARS vastly outperforms other recently proposed schemes under a sophisticated jamming attack. As a future work, we will continue to analyze the performance of the proposed scheme under multiple SUs and PUs by taking into account collision problems between SUs.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This work was supported by Hwarang-Dae Research Institute of Korea Military Academy under Grant no. 20130401.

References

[1] FCC, *Notice of Proposed Rulemaking: In the Matter of Unlicensed Operation in the TV Broadcast Bands, FCC 04–113*, FCC, Washington, DC, USA, 2004.

[2] 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.

[3] S. Bhattarai, J. M. J. Park, B. Gao, K. Bian, and W. Lehr, "An overview of dynamic spectrum sharing: ongoing initiatives, challenges, and a roadmap for future research," *IEEE Transactions on Cognitive Communications and Networking*, vol. 2, no. 2, pp. 110–128, 2016.

[4] Z. Htike, C. S. Hong, and S. Lee, "The life cycle of the rendezvous problem of cognitive radio ad hoc networks," *Journal of Computing Science and Engineering*, vol. 7, no. 2, pp. 81–88, 2013.

[5] C. Cordeiro, K. Challapali, D. Birru, and S. Shankar, "IEEE 802.22: the first worldwide wireless standard based on cognitive radios," *Journal of Communications*, vol. 1, no. 1, pp. 38–47, 2006.

[6] E.-K. Lee, S. Y. Oh, and M. Gerla, "Frequency quorum rendezvous for fast and resilient key establishment under jamming attack," *ACM SIGMOBILE-Mobile Computing and Communications Review*, vol. 14, no. 4, pp. 1–3, 2010.

[7] K. Bian and J. M. Park, "Asynchronous channel hopping for establishing rendezvous in cognitive radio networks," in *2011 Proceedings IEEE INFOCOM*, pp. 236–240, Shanghai, China, April 2011.

[8] M. D. Silvius, F. Ge, A. Young, A. B. MacKenzie, and C. W. Bostian, "Smart radio: spectrum access for first responders," in *Proceedings of the Wireless Sensing and Processing III*, pp. 698008–698012, Orlando, FL, USA, 2008.

[9] C. Cormio and K. R. Chowdhury, "Common control channel design for cognitive radio wireless ad hoc networks using adaptive frequency hopping," *Ad Hoc Networks*, vol. 8, no. 4, pp. 430–438, 2010.

[10] P. Bahl, R. Chandra, and J. Dunagan, "Ssch: slotted seeded channel hopping for capacity improvement in ieee 802.11 adhoc wireless networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, ser. MobiCom'04*, pp. 216–230, ACM, New York, NY, USA, 2004.

[11] S. Krishnamurthy, M. Thoppian, S. Kuppa et al., "Time-efficient distributed layer-2 auto-configuration for cognitive radio networks," *Computer Networks*, vol. 52, no. 4, pp. 831–849, 2008.

[12] K. Bian, J.-M. Park, and R. Chen, "A quorum-based framework for establishing control channels in dynamic spectrum access networks," in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking, ser. MobiCom'09*, pp. 25–36, ACM, New York, NY, USA, 2009.

[13] L. A. DaSilva and I. Guerreiro, "Sequence-based rendezvous for dynamic spectrum access," in *2008 3rd IEEE Symposium on New Frontiers in Dynamic Spectrum Access Networks*, pp. 1–7, Chicago, IL, USA, October 2008.

[14] N. C. Theis, R. W. Thomas, and L. A. DaSilva, "Rendezvous for cognitive radios," *IEEE Transactions on Mobile Computing*, vol. 10, no. 2, pp. 216–227, 2011.

[15] Z. Lin, H. Liu, X. Chu, and Y. W. Leung, "Jump-stay based channel-hopping algorithm with guaranteed rendezvous for cognitive radio networks," in *2011 Proceedings IEEE INFOCOM*, pp. 2444–2452, Shanghai, China, April 2011.

[16] Z. Lin, H. Liu, X. Chu, and Y.-W. Leung, "Enhanced jump-stay rendezvous algorithm for cognitive radio networks," *IEEE Communications Letters*, vol. 17, no. 9, pp. 1742–1745, 2013.

[17] H. Liu, Z. Lin, X. Chu, and Y. W. Leung, "Ring-walk based channel-hopping algorithms with guaranteed rendezvous for cognitive radio networks," in *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, pp. 755–760, Hangzhou, China, December 2010.

[18] I. H. Chuang, H. Y. Wu, and Y. H. Kuo, "A fast blind rendezvous method by alternate hop-and-wait channel hopping in cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 10, pp. 2171–2184, 2014.

[19] D. Yang, J. Shin, and C. Kim, "Deterministic rendezvous scheme in multichannel access networks," *Electronics Letters*, vol. 46, no. 20, pp. 1402–1404, 2010.

[20] J. Shin, D. Yang, and C. Kim, "A channel rendezvous scheme for cognitive radio networks," *IEEE Communications Letters*, vol. 14, no. 10, pp. 954–956, 2010.

[21] R. N. Yadav and R. Misra, "Periodic channel-hopping sequence for rendezvous in cognitive radio networks," in *2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1787–1792, Kochi, India, August 2015.

[22] S. Salehkaleybar and M. R. Pakravan, "A periodic jump-based rendezvous algorithm in cognitive radio networks," *Computer Communications*, vol. 79, pp. 66–77, 2016.

[23] H. Pu, Z. Gu, X. Lin, Q. S. Hua, and H. Jin, "Dynamic rendezvous algorithms for cognitive radio networks," in *Proceedings of the 2016 IEEE International Conference on Communications (ICC)*, pp. 1–6, Kuala Lumpur, Malaysia, May 2016.

[24] H. Liu, Z. Lin, X. Chu, and Y. W. Leung, "Taxonomy and challenges of rendezvous algorithms in cognitive radio networks," in *Proceedings of the 2012 International Conference*

on Computing, Networking and Communications (ICNC), pp. 645–649, Maui, HI, USA, January–February 2012.

[25] M. J. Abdel-Rahman and M. Krunz, "Game-theoretic quorum-based frequency hopping for anti-jamming rendezvous in dsa networks," in *Proceedings of the 2014 IEEE International Symposium on Dynamic Spectrum Access Networks (DYSPAN)*, pp. 248–258, Tysons Corner, VA, USA, April 2014.

[26] K. Grover, A. Lim, and Q. Yang, "Jamming and anti-jamming techniques in wireless networks: a survey," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 17, no. 4, pp. 197–215, 2014.

[27] J.-R. Jing, Y.-C. Tseng, C.-S. Hsu, and T.-H. Lai, "Quorum-based asynchronous power-saving protocols for ieee 802.11 ad hoc networks," in *Proceedings of the 2003 International Conference on Parallel Processing*, pp. 257–264, Montreal, Canada, October 2003.

[28] W.-S. Luk and T.-T. Wong, "Two new quorum based algorithms for distributed mutual exclusion," in *Proceedings of 17th International Conference on Distributed Computing Systems*, pp. 100–106, Baltimore, MD, USA, May 1997.