Jamming-Resistant Communication: Channel Surfing without Negotiation

Shaxun Chen, Kai Zeng, Prasant Mohapatra
Department of Computer Science, University of California, Davis, CA 95616
{sxch, kaizeng, pmohapatra}@ucdavis.edu

Abstract—Channel surfing is an effective method to prevent jamming attacks in wireless communications. In traditional channel surfing schemes, two parties have to negotiate beforehand, in order to agree on the channel switching sequence. However, the negotiation process itself is vulnerable to jamming attacks. In this paper, we propose a novel channel surfing method without relying on such negotiation. Taking advantage of the reciprocity of the wireless fading channel, our method switches channels according to the random channel states observed by the two parties during their communication. Therefore, it does not introduce any extra communication overhead and can achieve strong security. To evaluate our method, we carry out extensive experiments using off-the-shelf 802.11 devices in a real indoor environment. Experimental results validate the efficiency and security of our method.

I. Introduction

With the increasing popularity of wireless networks, the security and reliability issues of wireless communication attract more and more attention. However, due to the shared medium and broadcast nature, wireless communication is especially vulnerable to jamming style deny-of-service (DoS) attacks. These attacks keep sending random packets or noise, in order to jam the channel and prevent legitimate parties from communicating.

The approaches dealing with jamming attacks mainly fall into two categories [1]. The first is spatial retreat, in which nodes try to move out of the jamming area when a jamming attack occurs. This approach requires that wireless nodes are all able to move. Moreover, spatial retreat may break the connectivity of the original wireless network. The cost of this approach is very high, so that it can only be applied to very limited scenarios. The other category of approaches is called channel surfing. Legitimate nodes change their communication frequency periodically to avoid jamming from attackers. This approach is much more feasible and widely used than the former one.

For channel surfing, the main concern is how to achieve the agreements on channel selection between two transceivers. Only when two parties select the same channel at each time slot can they successfully communicate. In highly dynamic environments, such as mobile ad hoc networks, two arbitrary parties usually do not have pre-shared secrets. They have to talk to each other beforehand, in order to decide the channel switching sequence. An alternative method is to change the channel according to a pseudo-random sequence, but two parties still have to exchange a seed so that they can generate an identical pseudo-random sequence.

Here is the controversy. Channel surfing is used to avoid jamming attacks. However, the negotiation (or key exchange) process of channel surfing itself is vulnerable to jamming and eavesdropping attacks. If the negotiation is jammed, two legitimate parties cannot even begin the channel surfing. On the other hand, if the content of negotiation is broken by the attackers, the channel surfing becomes meaningless, because the attacker is able to follow the same sequence of channels as legitimate transceivers and always jam their communication.

In this paper, we propose a novel channel surfing method which does not require any prior negotiation. Our method is based on the randomness of wireless fading channel and the theory of reciprocity in wireless communication. The reciprocity theory demonstrates that bidirectional wireless channel states should be identical between two transceivers at a given instant of time [2] [3]. We use this identical random channel state as the inherent shared secret between two communicators to generate the channel choice.

In our approach, the negotiation (as well as key exchange) process is eliminated, which breaks the circular dependency in existing works (negotiation process requires jamming-resistant communication; jamming-resistant communication requires negotiation). Therefore, our method is more robust to jamming attacks. Additionally, it gets rid of the negotiation cost and does not add any extra communication overhead.

Furthermore, as long as the attacker is more than half of the wavelength away from legitimate communicators, the channel states he observed should be independent to the channel state between the legitimate ones [4]. This means the attacker can never eavesdrop the secret (channel state) shared between legitimate communicators. To this extent, our approach provides a strong secure channel surfing method.

We conduct real-testbed experiments to evaluate our method. The result shows that without any negotiation and key exchange, we can achieve a channel agreement ratio higher than 90 percent.

The rest of the paper is organized as follows. Section 2 reviews related works. Section 3 presents our approach of channel surfing without prior negotiation. We evaluate our method comprehensively in Section 4, and discuss related issues in Section 5. Section 6 concludes the paper.

II. Related Work

Existing channel surfing methods require prior negotiation or seed exchange to achieve channel agreement [11] [13], which is vulnerable to jamming attacks. In addition, there is possibility that the content of negotiation (or seeds) suffers from interception attacks, so that the surfing sequence could be revealed to the adversary. Diffie-Hellman algorithm can be used to establish a shared secret key without prior knowledge over an insecure channel [6], but the key agreement process can still be jammed.

Frequency hopping is similar to channel surfing in that both of them change frequency during the communication. However, frequency hopping is a physical-layer technology, which requires more advanced transceivers, while channel surfing is a link-layer technology that can be applied to the existing wirelesss devices without frequency hopping features. Above all, frequency hopping also needs prior negotiation or seed exchange to achieve frequency agreement [5].

Strasser, et al. proposed a frequency hopping method that does not rely on negotiation [7] [12]. In their approach, two communication parties switch their channels at different rates. During a specific time period, they always have a chance to share a channel for a short time slot. However, the channel utilization of this method is low.

There have been some works exploiting the wireless channel randomness and reciprocity property to generate secret keys [8] [9] [10]. All of them need extra information exchange (information reconciliation) between two parties, which can be attacked by jammers. In addition, if we try to make use of these approaches and adapt them for channel agreement purpose, the extra communication overhead is considerable because information reconciliation should be performed each time the channel switches.

III. Channel Surfing without Prior Negotiation

In this section, we first outline the problem, and then introduce our channel surfing method. We present in detail how two communicating parties reach the agreement on channel selection without any prior negotiation and extra information exchange.

A. *Problem Description*

Alice and Bob are two legitimate users, who communicate with each other via wireless media. Eve is an adversary who sends random packets (or noise) in order to jam the communication between Alice and Bob as much as possible.

We assume that Alice and Bob use the same type of radio devices which can work on multiple channels. Eve can jam only one channel at a given time (which is a common assumption in traditional channel surfing [1] [11] [13]). As we know, hackers usually attack a node in the legitimate network and use it as a zombie to perform DoS attacks. Therefore, it is reasonable to assume that the attacker's wireless device has comparable capability as those of legitimate ones. Besides, if a jammer emits signal in a very wide band, it is easier to be detected.

In our settings, Alice and Bob change their channels periodically to avoid Eve's jamming attack. They do not exchange any information about channel selection, but can make their channel choice separately and still hop to the same channel almost all the time by using our method.

B. *Property of Reciprocity*

The property of reciprocity declares that bidirectional wireless channel states should be identical between two transceivers at a given instant of time, which is the basis of our method. We use this channel state as the inherent shared secret between two parties, in order to achieve channel agreement.

Let $\vec{A} = (A_1, A_2, \ldots, A_n)$ be the states of the channel between Alice and Bob observed by Alice at time $t_1, t_2, \ldots, t_n$ respectively. Similarly, $\vec{B} = (B_{1'}, B_{2'}, \ldots, B_{n'})$ are the channel states observed by Bob at time $t_{1'}, t_{2'}, \ldots, t_{n'}$, where $t_1 < t_{1'} < t_2 < t_{2'} < \ldots < t_n < t_{n'}$. According to the property of reciprocity, we have $A_i = B_i$ ($1 \leqslant i \leqslant n$), where B_i is the channel state observed by Bob at time t_i, and if $t_{i'} - t_i$ ($1 \leqslant i \leqslant n$) is shorter than the channel coherence time, we have $B_i \approx B_{i'}$ ($1 \leqslant i \leqslant n$) because the channel can be considered stable and predicable within the channel coherence time. Therefore, $A_i \approx B_{i'}$ ($1 \leqslant i \leqslant n$).

If two random variables are linearly related, the correlation coefficient of them is 1. Trivially, the correlation coefficient of two identical random variables equals 1. Since $A_i \approx B_{i'}$ ($1 \leqslant i \leqslant n$), the correlation coefficient of $\vec{A}$ and $\vec{B}$ should be very close to 1. The definition of correlation coefficient is given in formula (1).

$$\rho(\vec{A},\vec{B}) = \frac{\mathrm{E}((\vec{A}-\mathrm{E}\vec{A})(\vec{B}-\mathrm{E}\vec{B}))}{\delta\vec{A}\delta\vec{B}} \qquad (1)$$

where $\rho(\vec{A},\vec{B})$ stands for the correlation coefficient of $\vec{A}$ and $\vec{B}$. E indicates expectation and δ is standard deviation.

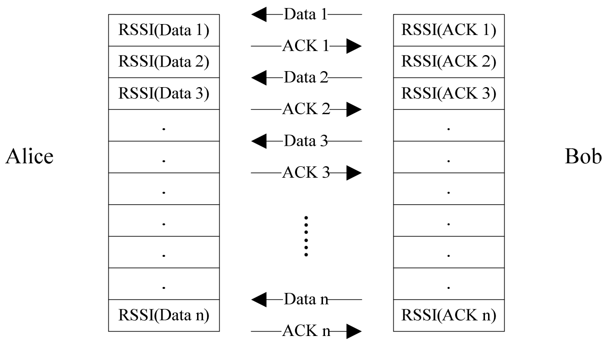

Figure 1. Sequence of RSSI recorded by Alice and Bob respectively

In this paper, A_i is instantiated by the RSSI (received signal strength indicator) of the ith data packet received by Alice (sent by Bob), while $B_{i'}$ is the RSSI of the corresponding ACK (MAC layer ACK of that ith data packet) received by Bob. These data and ACK packets are all for the data communication between Alice and Bob. Our method does not introduce any extra packets.

We choose RSSI as the indicator of the channel state because it is convenient to extract. We can read it from *radiotap* without any modification of the hardware, firmware and driver of the off-the-shelf 802.11 wireless cards. Physically, RSSI is acquired during the preamble stage of receiving an 802.11 frame. We would like to mention that our method is also applicable to any other parameters of channel states, such as phase, amplitude, etc.

We use $B_{i'}$ instead of B_i simply because wireless devices cannot transmit and receive at the same time. Since MAC layer ACK responds very quickly, the value of $t_{i'} - t_i$ is only about the data frame transmission time plus a SIFS (short inter frame space), which is much shorter than the channel coherence time of 2.4G or 5G bands. Using 802.11a as an illustrative example, assuming modulation rate is 6Mbps and packet size is 512 bytes, $t_{i'} - t_i$ is about 0.8ms. The channel coherence time of 5GHz band under walking speed (1m/s) is about 25ms.

Based on the above discussion, when Alice records the RSSI sequence of the data packets sent by Bob, and Bob records the RSSI sequence of the corresponding ACKs (shown in Figure 1), the correlation coefficient of these two sequences should be very close to 1 (i.e. $\rho(\vec{A},\vec{B}) \approx 1$). We will validate this result in Section 4B (the experimental result is about 0.98). In Section 4, we also show that the RSSI sequence collected by the adversary, Eve, is not closely related to that of either Alice or Bob. Their correlation coefficients are far less than 1.

C. Achieving Agreement on Channel Selection

As motioned in Section 3B, we have $\rho(\vec{A},\vec{B}) \approx 1$. In this section, we describe how to make use of this conclusion and enable Alice and Bob to reach the agreement on channel selection without any negotiation.

Theorem 1. If $\rho(\vec{A},\vec{B}) = 1$, then for any random variable $\vec{C}$, $\rho(\vec{A},\vec{C}) = \rho(\vec{B},\vec{C})$.

Proof. Since $\rho(\vec{A},\vec{B}) = 1$, $\vec{A} = \alpha\vec{B} + \beta$ (both α and β are constants). Then

$$\begin{aligned}
\rho(\vec{A},\vec{C}) &= \frac{\mathrm{E}((\vec{A}-\mathrm{E}\vec{A})(\vec{C}-\mathrm{E}\vec{C}))}{\delta\vec{A}\delta\vec{C}} \\
&= \frac{\mathrm{E}(\vec{A}\vec{C})-\mathrm{E}\vec{A}\mathrm{E}\vec{C}}{\delta\vec{A}\delta\vec{C}} \\
&= \frac{\mathrm{E}((\alpha\vec{B}+\beta)\vec{C})-\mathrm{E}(\alpha\vec{B}+\beta)\mathrm{E}\vec{C}}{\delta(\alpha\vec{B}+\beta)\delta\vec{C}} \\
&= \frac{\alpha\mathrm{E}(\vec{B}\vec{C})+\beta\mathrm{E}\vec{C}-\alpha\mathrm{E}\vec{B}\mathrm{E}\vec{C}-\beta\mathrm{E}\vec{C}}{\alpha\delta\vec{B}\delta\vec{C}} \\
&= \frac{\mathrm{E}(\vec{B}\vec{C})-\mathrm{E}\vec{B}\mathrm{E}\vec{C}}{\delta\vec{B}\delta\vec{C}} \\
&= \rho(\vec{B},\vec{C}) \qquad \square
\end{aligned}$$

According to Theorem 1, based on $\rho(\vec{A},\vec{B}) \approx 1$, we can infer $\rho(\vec{A},\vec{C}) \approx \rho(\vec{B},\vec{C})$ for any $\vec{C}$. This relationship inspires our distributed channel selection mechanism. We let Alice and Bob keep a common sequence $\vec{C}$, which is publicly known to everyone (including Eve). Alice now can calculate $\rho(\vec{A},\vec{C})$ using recorded RSSI sequence together with $\vec{C}$. Similarly, Bob can also calculate $\rho(\vec{B},\vec{C})$ by himself. They do not need any negotiation or information exchange.

Since the value of $\rho(\vec{A},\vec{C})$ and $\rho(\vec{B},\vec{C})$ should be approximately the same, Alice and Bob can agree on their next working channel using $\rho(\vec{A},\vec{C})$ and $\rho(\vec{B},\vec{C})$ respectively.

The value of $\rho(\vec{A},\vec{C})$ or $\rho(\vec{B},\vec{C})$ is a real number between -1 and 1. Assuming there are M channels in total, we need a function $\mathscr{F}$ which maps the correlation coefficient to the channel number (1 to M). The output of $\mathscr{F}$ should satisfy uniform distribution in that the legitimate user communicates on each channel with the same probability. We get such $\mathscr{F}$ through the following steps:

1) choose a certain $\vec{C}$;

2) collect data as training data set, for instance, RSSI sequence collected by Alice in the i^{th} run of the experiment is noted as $\vec{A}_i$ (similarly, $\vec{B}_i$ for Bob), let $\rho_{2i-1} = \rho(\vec{A}_i,\vec{C})$ and $\rho_{2i} = \rho(\vec{B}_i,\vec{C})$;

3) run the experiment for k times (evenly in each channel), then ascendingly sort ρ_i ($1 \leqslant i \leqslant 2k$);

$\mathscr{F}$ is defined as follows:

$$\mathscr{F}(x) = j$$

$$\text{where } j \text{ is the min integer that satisfies } x \leqslant \rho_{\lceil 2k*j/M \rceil} \qquad (2)$$

Now Alice and Bob are able to calculate the channel number. Above all, they only need local information to perform this calculation. Since $\rho(\vec{A},\vec{C})$ and $\rho(\vec{B},\vec{C})$ have approximately the same value, Alice and Bob will achieve the same channel selection ($\mathscr{F}(\rho(\vec{A},\vec{C}))$ euqals $\mathscr{F}(\rho(\vec{B},\vec{C}))$) with high probability. In Section 4C, experiments show that this probability is higher than 90%.

Besides, legitimate users only need to record RSSI of received data or ACK packets. Our method does not add even one extra packet. No probing or information reconciliation packets are needed. The computation overhead is also low, for that the complexity of correlation coefficient calculation is $O(n)$, where n is the sequence length of $\vec{A}$ and $\vec{B}$.

D. Protocol: Channel Surfing without Prior Negotiation

We present the methodology of our novel channel surfing scheme above. In order to make it work in practice, a concrete protocol is necessary. In this section, we describe our protocol briefly due to the limited space. It is based on the following assumptions.

1) Each legitimate user only has one antenna, or the antenna use within a certain time slot can be specified.

2) Two communication parties (Alice and Bob) both have M channels, and their clocks are synchronized.

All the channel surfing and frequency hopping methods require clock synchronization ([7] is an exception, but its channel utilization is very low). Actually, synchronization granularity of millisecond or tens of millisecond is enough for our method. Our protocol is described as follows:

1. A legitimate user initiates its communication on the first available channel (e.g. if channel one is jammed, then it begins to talk on channel two, and so forth. Jamming detection is discussed in Section 5).
2. Both parties change their own channels every t seconds (t = 0.2 in our experiments).

3. Once a user begins communicating on a new channel, the RSSIs of the first *n* packets (data or ACK) are used to calculate the next channel choice. If the number of packets received in current channel is less than *n*, the RSSIs of latest *n* packets are used (in our experiments, $n = 800$).
4. We use square wave to generate $\vec{C}$. The period is the time duration of *n* packets.
5. If two parties cannot achieve agreement (i.e. $\mathscr{F}(\rho(\vec{A},\vec{C}))$ is not euqal to $\mathscr{F}(\rho(\vec{B},\vec{C}))$), both of them go back to the last channel they successfully communicated on.

For the choice of $\vec{C}$, as indicated in Theorem 1, an arbitrary function is applicable theoretically. In practice, different choices can influence the probability of channel agreement. Generally speaking, the choice of $\vec{C}$ should follow these principles:

a) not a constant sequence. Otherwise, the denominator of the correlation coefficient will be zero.

b) not a random sequence. The correlation coefficient between a random sequence and an arbitrary sequence is close to zero. When the absolute value of the correlation coefficient becomes very small, the realtive error is enlarged.

c) does not change dramatically. The rationale is similar to b). RSSI (in the large scale) is related to the distance between communicators, and in wireless transmission, usually hundreds or even thousands of packets are sent within a second. Hence, per packet RSSI sequence changes "continuously" in the large scale. The correlation coefficient of two variables tends to be very small if one changes slowly and the other changes very quickly. If we use a periodic function to generate sequence $\vec{C}$*, the frequency of the function should not be very high.*

Although these principles enforce some constraints, we have a wide choice for $\vec{C}$. In Section 4C, we test square wave and sine function with various frequencies to generate $\vec{C}$. It is shown that in most cases, our method can reach a channel agreement ratio higher than 90%.

TABLE I. DEFAULT $\mathscr{F}$

x (Correlation Coefficient)	$\mathscr{F}(x)$ (Channel)
[-1, -0.448]	1 (Channel 36)
(-0.448, -0.281]	2 (Channel 40)
(-0.281, -0.104]	3 (Channel 44)
(-0.104, 0.114]	4 (Channel 48)
(0.114, 0.302]	5 (Channel 52)
(0.302, 0.47]	6 (Channel 56)
(0.47, 1]	7 (Channel 60)

Using our experimental data as the training set (generate $\vec{C}$ by square wave with period 800, seven channels), we calculate $\mathscr{F}$ following the steps in Section 3C. The result is shown in Table Ⅰ. This $\mathscr{F}$ is used as default in our protocol. If the scenario is significantly different from our experiment settings, this $\mathscr{F}$ can be used as a start point and trained on the fly, in order to achieve uniform channel usage.

IV. EVALUATIONS

We conduct comprehensive experiments on real testbeds to evaluate our channel surfing method.

In this section, we first introduce the experiment settings. Then we validate the property of reciprocity and demonstrate the performance of our scheme. We also show that the probability an adversary can successfully perform jamming attack is very low. Finally, we present the probability distribution of the channels in the channel surfing.

A. Experiment Settings

We use three HP nc6000 laptops acting as Alice, Bob and Eve. They are equipped with Atheros 802.11abg wireless cards and MadWiFi driver. We perform experiments on 802.11a band because it has more non-overlapping channels than 802.11b/g. Each nc6000 laptop has two antennas. We use *sysctl* command to disable one of them, so that continuous packets are transmitted in the same fading channel.

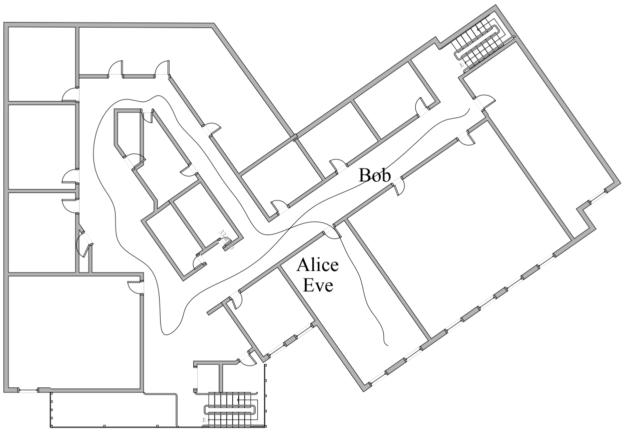

Figure 2. Layout of the experiment environment

In all of the following experiments, Alice stays static in the office and Bob walks around randomly in the hallways of the building (shown in Figure 2). Eve sits next to Alice, only about 30 centimeters away. All of them use the same transmission power. RSSI (radiotap.dbm_antsignal filed) measured by Atheros wireless cards is an integer falling into [-92, -18].

B. Reciprocity and Secrecy Validation

In this experiment, Alice records the RSSIs of the data packets sent from Bob (noted as $\vec{A}$), and Bob records the RSSIs of the corresponding ACKs ($\vec{B}$). We measured more than 1.3 million packets in seven channels (channel 36, 40, 44, 48, 52, 56 and 60 of 802.11a). The correlation coefficient of $\vec{A}$ and $\vec{B}$ is shown in Table Ⅱ.

TABLE II. RSSI CORRELATION COEFFICIENT

	Alice	Eve (Data)	Eve (ACK)
Alice		0.203	0.196
Bob	0.983	0.191	0.177

In this test, Eve is set to the *monitor* mode, so that she is able to capture all the data packets and ACKs even they are not sent to her. She records the RSSI sequence of all the data packets sent to Alice by Bob, as well as the ACKs sent to Bob by Alice. We calculated the correlation coefficient of these two sequences with $\vec{A}$ and $\vec{B}$, respectively, and the result is shown in Table II.

We can see that the correlation coefficient between $\vec{A}$ and $\vec{B}$ is very close to 1, which justifies the reciprocity property. On the other hand, the channel state observed by the adversary is far from closely related to $\vec{A}$ or $\vec{B}$ (the correlation coefficient is only about 0.2), although Eve is physically very close to Alice.

This experiment demonstrates that the state of the wireless fading channel is a random secret shared by two transceivers.

C. *Channel Agreement Ratio*

Now we evaluate the performance of our channel surfing scheme. The RSSI collecting process is the same as the experiment above. The only difference is that the RSSI sequence is now divided into fragments with the length of 800. That is, the lengths of $\vec{A}$ and $\vec{B}$ are both 800.

We choose this value as the sequence length to calculate the next working channel for the following reasons. When the length is too short, the statistical relationship between $\vec{A}$ and $\vec{B}$ cannot tolerate the measurement error and random noise, and the correlation coefficient will decrease. If the length is too long, the computing overhead increases. Of course, 800 is not the only choice. We use this value for illustration.

In this experiment, we use square wave function and sine function to generate $\vec{C}$ respectively. The amplitude of both functions is set to half of the RSSI range ((92-18)/2 = 36). The period of these two functions is shown as the y-axis. If $\mathcal{F}(\rho(\vec{A},\vec{C})) = \mathcal{F}(\rho(\vec{B},\vec{C}))$, we say that Alice and Bob agree on channel selection ($\mathcal{F}$ is the default presented in Section 3D). Channel agreement ratio = (times that Alice and Bob agree on channel selection) / (total switching times). We repeat our experiment for 1680 times, equally on seven channels. We apply the whole data set to each $\vec{C}$ and calculate the channel agreement ratio. The result is shown in Figure 3.

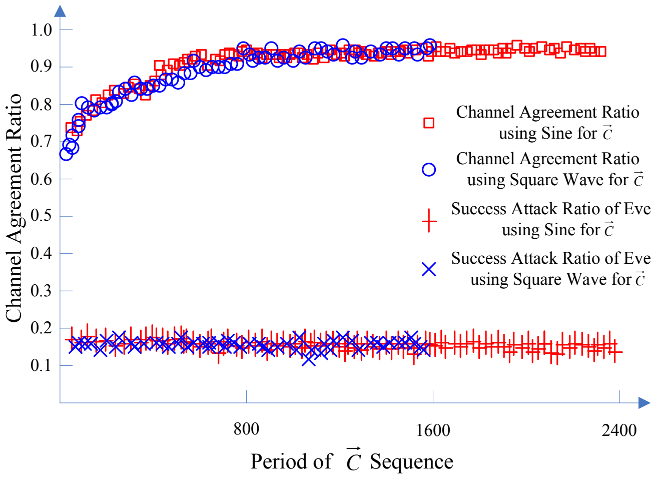

Figure 3. Channel agreement radio and successful attack ratio

From Figure 3, we can see that the channel agreement ratio keeps increasing when the period of sine (or square wave) is small. After the period gets larger than 800, channel agreement ratio becomes stably around 90%-95%. The performance of square wave and that of sine are very close. For square wave, we only test those with peroids less than 1600, because when the period gets larger than that, it may produce a constant sequence (assuming sequence length n = 800) which contradicts with principle a) in Section 3D.

The results shown in Figure 3 agrees with our principle c) in Section 3D. The choice of $\vec{C}$ does not matter much as long as its period is not very short. For quite a wide choice range, our channel surfing method is able to achieve a channel agreement ratio higher than 90%, which is much better than [7]. Compared with tranditional channel surfing and frequency hopping methods, we think it is worthwhile achieving strong security with only less than 10% performance drop.

In our protocol, we use the square wave with the period of 800 as default to generate $\vec{C}$. It has a channel agreement ratio of 93.8%, which is not the highest. However, we do not want $\vec{C}$ to overfit our data set. Moreover, our primary goal is to show the effectiveness and efficiency of our method rather than adjust parameters for trivial performance gain. Any sine or square wave with periods larger than 800 is fine to validate our method.

We also tested the probability that Eve correctly guesses the channel that Alice and Bob will hop to. Eve collects the RSSI sequences by overhearing the communication between Alice and Bob, and then follows the same steps as our method. The result is also shown in Figure 3. The average probability of successful attack is 16.9%, which is only slightly higher than the random attack (1/7 = 14.3%). This verifies that channel fading is a random shared secret between Alice and Bob, and Eve almost gains no useful information by eavesdropping.

D. *Distribution of Channel Selection*

In this experiment, we use half of our data as training data to generate $\mathcal{F}$, then apply this $\mathcal{F}$ to the other half of the data to do channel surfing.

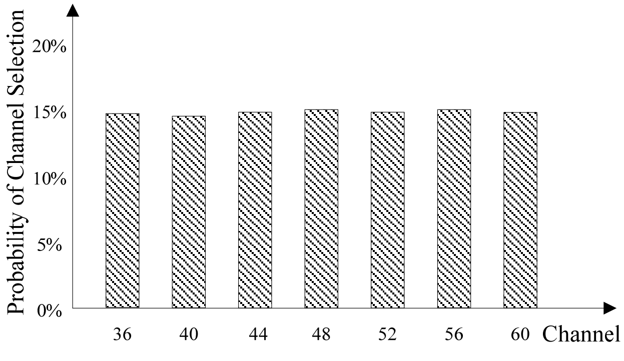

Figure 4. Distribution of channel selection

The distribution of channel selection is shown in Figure 4. The probability of hopping on each channel is almost equally distributed. The result demonstrates that $\mathcal{F}$ is stable in a particular setting and environment. However, as mentioned in

Section 3D, if the application scenario is very different from the indoor environment, the default $\mathcal{T}$can be used as a start point, and then $\mathcal{T}$can be trained on the fly.

V. Discussion

Jamming style DoS attacks can be divided into two categories, predictive jamming and reactive jamming. For predictive jamming, there are also two possible types. The first is random attack. The jammer sends noise or invalid packets on a random channel. In this case, the successful attack ratio is $1/M$, where M is the number of channels (in our experiments, $M = 7$, so successful attack ratio is about 14.3%). Another type of predictive jammers, who are more intelligent, may use the same method as legitimate communicators to select channel, such as Eve in Section 4C. But Eve's successful attack ratio (16.9%) is only slightly higher than random ones, which demonstrates that our scheme is also resistant to more sophisticated jammers. As to reactive jamming attack, jammers scan channels first and then perform attacks on the channel that legitimate users are operating on. In our settings, Alice and Bob switch their channels every 0.2 seconds, and the radio start-up cost (in a new channel) of 802.11 devices is typically tens of milliseconds. Therefore, Eve is not able to complete scanning before legitimate users change their channels. Hence, reactive jamming is not effective to our channel surfing scheme either. We should point out that no matter what form of jamming attacks the adversaries perform, the more channels legitimate users have, the less possible they are jammed.

As mentioned in Section 3D, our method needs to tell whether a jamming attack occurs on the current channel. Actually, all anti-jamming approaches require jamming detection ability. There have been many works on this topic. Common detection methods include link-layer idle time detection, ambient noise level measurement, packet delivery ratio threshold and the combinations of them [14]. These research are orthogonal to our method and we integrate any of them into our scheme.

Besides the vulnerability to jamming attacks, the negotiation (and seed exchange) process of traditional channel surfing methods also faces threatens from eavesdroppers. Although the negotiation or seeds are encrypted, they are not perfectly safe. Some encryption algorithms have been broken in the recent years. A more advanced method is to generate a shared key using Diffie-Hellman algorithm. However, this algorithm is based on the assumption that discrete logarithm problem is intractable. With rapid increase of computers' ability and development of quantum computers, this assumption might not hold true in the future. In addition, if two parties switch the channel following a pseudo-random sequence, it is possible that the adversary learns the function after a period of observation because the sequence is not truly random. In contrast, our channel surfing is based on the randomness of fading channel states, which cannot be observed by adversaries. Even legitimate parties do not know the channel surfing sequence beforehand. Furthermore, as long as attackers are more than half of the wavelength (3cm for the 5GHz band) away from legitimate users, the fading channel states they observed are independent to that between legitimate ones. Therefore, our method provides a strong secure channel surfing method.

VI. Conclusion and Future Work

In this paper, we propose a novel channel surfing method, which utilizes the wireless fading channel state as a random shared secret between legitimate parties to achieve channel agreement. Our method does not need prior negotiation or seed exchange, thus, it is more robust to jamming attacks. Real testbed experiments show that our approach achieves high channel agreement radio without introducing extra communication overhead.

When two parties use different transmission power or has different physical features (e.g. different antenna types), the problem becomes more challenging. We will focus on applying our method to heterogeneous wireless systems in the future.

References

[1] W. Xu, T. Wood, W. Trappe, and Y. Zhang, "Channel Surfing and Spatial Retreats: Defenses against Wireless Denial of Service," ACM Workshop on Wireless Security, pages 80-89, Oct 2004.

[2] R. Ahslwede and I. Csiszar, "Common Randomness in Information Theory and Cryptography – Part I: Secret Sharing," IEEE Trans. Inf. Th., July 1993.

[3] R. Wilson, D. Tse, and R. A. Scholtz,"Channel Identification: Secret Sharing Using Reciprocity in Ultrawideband Channels," IEEE Trans. On Information Forensics and Security, Sept 2007.

[4] T. S. Rappaport, "Wireless Communications: Principles and Practice," New Jersey: Prentice Hall, 2001.

[5] R. A. Poisel, "Modern Communications Jamming Principles and Techniques," Artech House Publishers, 2006.

[6] W. Diffie and M. E. Hellman, "New Directions in Cryptography, IEEE Transactions on Information Theory," vol. IT-22, Nov. 1976, pages 644-654.

[7] M. Strasser, S. Capkun, C. Popper, M. Cagalj, "Jamming-resistant Key Establishment using Uncoordinated Frequency Hopping," IEEE Symposium on Security and Privacy, pages 64-78, May 2008.

[8] H. Koorapaty, A. Hassan, and S. Chennakeshu, "Secure Information Transmission for Mobile Radio," IEEE Communications Letters, vol. 4, no. 2, pages 52–55, Feburary 2000.

[9] S. Mathur, W. Trappe, N. Mandayam, C. Ye, and A. Reznik, "Radiotelepathy: Extracting a Secret Key from an Unauthenticated Wireless Channel," in MobiCom 08, pages 128-139, 2008

[10] R. Wilson, D. Tse, and R. Scholtz, "Channel Identification: Secret Sharing using Reciprocity in Ultrawideband Channels," ICUWB 07: IEEE International Conference on Ultra-Wideband, pages 270–275, Sept 2007.

[11] W. Xu, W. Trappe, and Y. Zhang, "Channel Surfing: Defending Wireless Sensor Networks from Interference," Proceedings of the 6th International Conference on Information Processing in Sensor Networks, pages 499-508, 2007.

[12] M. Strasser, C. Popper, and S. Capkun, "Efficient Uncoordinated FHSS Anti-jamming Communication," Proceedings of the tenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, pages 207-218, 2009.

[13] V. Navda, A. Bohra, S. Ganguly, and D. Rubenstein, "Using Channel Hopping to Increase 802.11 Resilience to Jamming Attacks," InfoCom 07: Proceedings of the 26th IEEE International Conference on Computer Communications, pages 2526-2530, May 2007.

[14] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks," in MobiHoc 05: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2005, pages 46-57.