Security Analysis of FHSS-type Drone Controller *

Hocheol Shin[1(✉)], Kibum Choi[1], Youngseok Park[2], Jaeyeong Choi[1], and Yongdae Kim[1,2]

[1] School of Electrical Engineering, KAIST, Republic of Korea
{h.c.shin,kibumchoi,go1736,yongdaek}@kaist.ac.kr
[2] Graduate School of Information Security, KAIST, Republic of Korea
raccoon7@kaist.ac.kr

Abstract. Unmanned Aerial Vehicles (UAVs), or drones, have attracted a considerable amount of attentions due to their utility to civilian as well as military applications. However, the security issues involved in UAV technology have not been extensively discussed in the literature. As a first step toward analyzing these security issues, we investigate security in drone controllers, especially controllers that adopt Frequency Hopping Spread Spectrum (FHSS). In order to affect an FHSS-type controller, an attacker first has to access its physical layer. This is difficult because of the pseudorandomness of the hopping sequence and the rapidly changing channels. However, these difficulties can be relaxed when the attacker acquires the hopping sequence and when the hopping speed of the target system is not significant. In this paper, we propose a general scheme to extract the hopping sequence of FHSS-type controllers using a software-defined radio (SDR). We also propose a method to address the issue of the limited bandwidth of the SDR. We implemented our scheme on a Universal Software Radio Peripheral (USRP), successfully extracted the hopping sequence of the target system, and exposed the baseband signal.

Keywords: Attack, Drone, Physical layer security, Blackbox system

1 Introduction

Because of their extensive range of application, from military airstrikes [20] to automated package delivery platforms [2, 4], unmanned aerial vehicles (UAVs) or drones have lately been the subject of increasing interest. As they become more popular, drones are frequently flown in noisy environments, and are thus exposed to intentional/unintentional interferences. It is therefore necessary for drones to

* This research was supported by (1) Next-Generation Information Computing Development Program through the NRF (National Research Foundation of Korea) funded by the MSIP (Ministry of Science, ICT & Future Planning) (No. NRF-2014M3C4A7030648), Korea, and (2) the MSIP, Korea, under the ITRC (Information Technology Research Center) support program (IITP-2015-R0992-15-1006) supervised by the IITP (Institute for Information & communications Technology Promotion)

secure their control systems against such interference. As exemplified by the case of the capture by Iranian forces of a United States Air Force (USAF) drone in 2011 [21], even military drone control systems are not adequately secure.

Wireless remote controllers for radio-controlled (RC) model aircraft in the past employed fixed frequencies of tens of megahertz [1]. However, due to a shortage of spectral resources, and in order to protect against interference, current remote controllers adopt spread spectrum technology on industrial, scientific, and medical (ISM) radio bands of 2.4 GHz [1, 11].

FHSS is a spread spectrum technology that continuously changes carrier frequency for anti-jamming/sniffing/spoofing capabilities. A theoretically unique hopping sequence is pre-shared by every transmitter-receiver pair through binding, which can prevent issues of mutual interference. However, even FHSS cannot completely protect links against all jamming/sniffing/spoofing threats. High-energy wideband jammers can block the entire hopping space used by an FHSS system [17]. A random jammer with a much greater hopping speed can deteriorate the signal-to-noise ratio (SNR) [17]. Furthermore, FHSS is vulnerable to reactive jammers with a sufficiently high reaction speed.

Although the above-mentioned attack vectors against FHSS are quite expensive for attackers, an exposed hopping sequence can drastically reduce the required complexity of attacks. Using the hopping sequence, attackers can proactively react to the changing center frequency, which enables the implementation of low-cost reactive jammers or baseband extractors.

In this paper, we propose a general scheme to extract the hopping sequences of FHSS-type drone controllers by using a software-defined radio (SDR). The versatility of SDRs makes it possible to deal with most FHSS-type drone controllers, which are mostly incompatible with one another [5]. We also propose a method to overcome the problem of limited SDR bandwidth when treating controllers with larger bandwidths. We applied our proposed scheme for a real-world FHSS-type controller, where the bandwidth of the target controller was approximately three times larger than that of the SDR, successfully extracted the total hopping sequence of the target system, and exposed the baseband signal.

The remainder of this paper is structured as follows: Sec. 2 provides the requisite background for our research here, whereas Sec. 3 is dedicated to a description of our proposed scheme to extract hopping sequences. In Sec. 4, we describe the implementation of the attack platform as well as the results. Sec. 5 summarizes related work in the area, and Sec. 7 contains our conclusions.

2 Background and Attack Model

2.1 Frequency Hopping Spread Spectrum

FHSS is a major spread spectrum technology along with Direct Sequence Spread Spectrum (DSSS). In wireless communications, it rapidly switches channels using a pseudorandom sequence, which makes it difficult to eavesdrop. Rapidly changing carrier frequency also renders the system highly resistant to narrowband interference, and enables it to share the frequency band with other systems using different communication technologies.

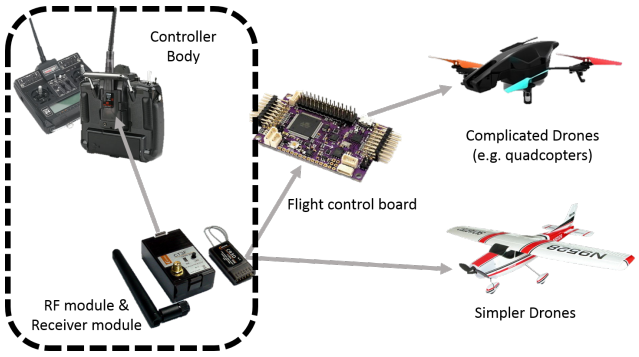

Fig. 1: RC aircraft system composition

FHSS has drawbacks as well. First, FHSS systems occupy much wider bandwidth than it actually requires. For example, a typical 10 channel FHSS system occupies bandwidth ten times wider than it actually uses. Second, a transmitter-receiver pair has to be finely synchronized, which is achieved by several ways. A transmitter may transmit duplications of a packet for all channels, while the receiver listens to a randomly selected channel. For another way, a transmitter-receiver pair can share a frequency table, repeating the predefined sequence.

Most of drone controllers utilize 2.4 GHz band, which is shared by many other wireless devices. Therefore, drone controllers can occupy only a fraction of assigned bandwidth. Furthermore, a transmitter-receiver pair does not have additional communication channel. Once bound at the initial stage, the pair communicates each other without any prior pairing steps after they are turned on. This means a consistent frequency table is shared by the pair.

2.2 Radio control system for RC aircraft

Drone controllers interface humans and drones. Although drones differ in their level of autonomy, controllers always take up the most critical functions, which makes controllers one of the most important component. Controllers vary as the drones are. In this paper we focus on controllers for civilian RC aircraft.

A typical RC controller consists of three components: a transmitter body, an RF module, and a receiver. The transmitter body provides the user interface, and converts user control into electrical signals. The RF module modulates and upconverts the control signal. It characterizes the wireless link between a transmitter and a receiver, whose robustness against interferences according to its wireless characteristics. The receiver reconverts the wireless signal into Pulse-width modulation (PWM) pulses. Fig. 1 shows the composition of the overall drone system, where components in the dashed rectangle indicate the drone controller.

Frequently, multiple RC aircraft are flown together, where multiple control signals interferes one another. In this case, control signal interference is critical for drones. They can fall into uncontrollable state, which is critical for fast moving aircraft. Therefore, RC controllers are required to resist high level of inter-

ference. To both share the band and resist mutual interference, spread spectrum technologies (FHSS and DSSS) are widely adopted.

Currently, no industrial standard for RC controllers exists. Therefore, controllers from diverse manufacturers are usually not compatible. Furthermore, the absence of the standard makes manufacturers hide details of their products. This makes the RC controller a blackbox system for third party analyst.

2.3 Attack Model

Our attack model is as follows. First, the target system is considered to be a blackbox. Though the attacker can analyze the system on her own capabilities, she cannot access any confidential information on the system a priori. Second, we assume the controller signal has at least one exclusively distinguishing characteristic. Furthermore, the attacker can exploit this characteristic to differentiate the target signal on air from other signals. The attacker can easily purchase such popular controllers and analyze their signal to reveal exclusive characteristics.

3 Methodology

3.1 Extracting the hopping sequence

Measuring channel information Typical FHSS systems have identical channel widths. Thus, we can derive channel center frequencies and the number of channels from measurements at the lowest and the highest channels. The center frequency of the remaining channels can be identified by repeatedly adding channel bandwidth to the first channel until the last channel is reached.

Detecting channel activeness In order to extract the hopping sequence through measurement, we first need to detect channel activeness. A considerable amount of past research in the area has dealt with this topic, since it is intimately related to cognitive radio [9, 18, 19, 23, 24]. Yücek et al. [24] listed various methods to determine spectral activeness according to the amount of available information regarding the target signal. The more accurate the method is, the more detailed the prior information that it requires.

Since the target system is considered to be a blackbox, applicable channel sensing methods are quite limited. Energy detection [19] and cyclostationarity-based detection [9] methods are representative techniques used to detect blackbox signals. We use energy detection to detect channel activeness for the sake of simplicity. However, this can be altered without affecting the remainder of our work here. Once we can detect channel activeness, we can record the history of the activated channels. Finally, assuming constant hopping speed, measurements of the continuous signal can be converted into a discretized sequence.

Searching the period The easiest way to predict the future sequence is to extract the period in hopping sequence. This can be achieved by choosing a part of the sequence and search for repetitions. While this scheme works well in most cases, there are instances of error. If the length of the chosen part is greater than twice the actual period, the period appears longer than it is. Moreover, if the chosen part is shorter than the half the actual period, the period can appear

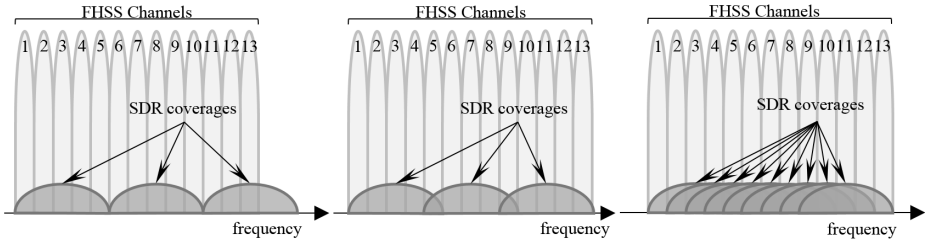

Fig. 2: Examples of SDR coverage arrangements

shorter than it is in some bad cases. These erroneous cases can be settled by searching repetitions by choosing multiple parts.

If the history has numerous measurement errors, the aforementioned exact matching-based search will fail. In such cases, we can find the period with similarity-based matching. This is identical to exact matching-based search except that repetitions are detected by similarity scores. Whenever the similarity score exceeds a certain threshold during the search, the relevant points are marked as repeating points. If a sufficient number of points are acquired, the intervals between any two points are aligned and compared to identify the most frequently appearing interval, which can be considered the hopping period.

Various pattern matching algorithms can be used to derive similarity scores. Matched filters are largely adopted for pattern detection [7, 8]. Algorithms used to solve sequence alignment problems [22] can also be considered, since that problem is quite similar to the one here.

3.2 Overcoming limited SDR bandwidth

In some cases, the bandwidth of the target system can exceed the maximum bandwidth of the SDR used. In such cases, the SDR can only monitor a part of each channel. This makes it impossible to detect the activeness of channels beyond the tuned SDR bandwidth. To solve this issue, the attacker can simultaneously utilize multiple, tightly synchronized SDRs, or a more powerful SDR that can cover the entire range of the target bandwidth. However, these approaches are expensive. We suggest an alternative that enables a single narrowband SDR to acquire the full hopping sequence of the target system.

Measuring the number of channels and their center frequencies is not challenging, even with a narrowband SDR, since we simply need to measure the first and the last channels. However, we can only acquire a number of partial sequences with a narrowband SDR, and such partial sequences should be uniquely merged to obtain the actual total hopping sequence. In order to uniquely combine partial sequences, SDR coverage should be carefully arranged. We explore various arrangements to show that a careful arrangement can yield the total sequence without ambiguity. Note that in all examples of partial sequences presented in this subsection, all channels in a period are activated equally frequently. Although this condition is not essential to our method, typical FHSS systems meet this condition in order to uniformly utilize bandwidth.

Fig. 2 shows three examples of coverage arrangement. In the left arrangement, the SDR coverages span the entire hopping space but do not overlap with one another. In this arrangement, partial hopping sequences are combined only with the duration of slots of no activity, which can lead to multiple combinations. For example, if channels under each coverage are activated in a series: "1 5 2 3 4 / 10 6 9 8 7 / 13 11 12," the partial sequences for each coverage are "1 5 2 3 4," "10 6 9 8 7," and "13 11 12," respectively. Since the channels in each coverage are contiguously activated, these partial sequences can also be combined as, for example, "1 5 2 3 4 / 13 11 12 / 10 6 9 8 7." This leads to multiple combinations.

The middle arrangement can also lead to multiple solutions when each channel is activated more than once in a period. The two example hopping sequences below show one of such cases. Overlapped channels are marked with hats and differences are bolded. It is easily verified that the two hopping periods, (1) and (2), are different, although the corresponding partial sequences are identical.

$$\cdots 1\ \hat{5}\ 2\ 3\ 4\ \mathbf{10}\ 6\ \hat{9}\ 8\ 7\ \mathbf{13\ 11\ 12} \parallel 2\ \hat{5}\ 3\ 4\ 1\ \mathbf{11}\ 6\ \hat{9}\ 8\ 7\ \mathbf{10\ 12\ 13} \cdots \quad (1)$$
$$\cdots 1\ 5\ 2\ 3\ 4\ \mathbf{11}\ 6\ 9\ 8\ 7\ \mathbf{10\ 12\ 13} \parallel 2\ 5\ 3\ 4\ 1\ \mathbf{10}\ 6\ 9\ 8\ 7\ \mathbf{13\ 11\ 12} \cdots \quad (2)$$

By contrast, the last arrangement, which is maximally overlapped, does not lead to multiple solutions. In this arrangement, every channel, except the one at each end, overlaps with another, i.e., they are all entangled. Therefore, any rearrangement of channels different from the original will always interfere with other partial sequences. However, maximal overlap is not always optimal. In most cases, a loosely overlapped arrangement will suffice. Indeed, we can uniquely combine partial hopping sequences with a non-maximally overlapped arrangement. Therefore, repeated trials with increasing overlaps are required to find the optimally overlapped arrangement.

3.3 Possible attack vectors

Once the total hopping sequence has been acquired, the basic requirements of catching up the ongoing FHSS signal are met. With the hopping sequence of the target system in hand, the attack cost can be greatly reduced, and the hopping sequence can be applied to the following attack platforms.

Baseband extractors receive and record the baseband signal while continuously following FHSS signal stream. The extracted baseband signal can later be analyzed to yield information regarding the modulation, encoding, or the packet structure of the baseband. **Reactive sniffer**s operate similar to baseband extractors, except that they can demodulate and decode the baseband signal to expose the bitstream or meaningful information from target systems. **Reactive jammer**s transmit narrowband interfering signals whenever a channel is activated. Using the extracted hopping sequence, the level of difficulty of implementing reactive jammers can be drastically reduced, since attackers can proactively wait for the channel to be activated.

4 Implementation and Results

4.1 Equipment

Software-defined Radio - We used a USRP N210 to receive signals from the target system. USRP N210 has a gigabit Ethernet interface, and can provide

Fig. 3: FrSKY DJT Radio Telemetry (RF module, left of left figure), FlySky FH-TH9X Transmitter (transmitter body, right of left figure), and FrSKY D4R-II 4Ch Receiver (receiver, right figure)

up to 25 million 16-bit pair (I & Q) samples per second ($= 2 \times 2B \times M/s = 100MB/s$) [6]. USRP N-series devices require a separate RF frontend, called a daughterboard. We used a CBX daughterboard [3] with full duplex capability with 40 MHz of instantaneous bandwidth. It can cover 1.2∼6 GHz.

Host PC - We used a general desktop with Intel Core i5-3570 and 16 GB of DDR3 memory. To interface with the USRP, we used Intel PRO/1000 PT Dual Port Server Adapter. As an OS, we used Ubuntu 12.04 LTS 64-bit.

4.2 Test Target Selection and Basic Analysis

Selected Test Target We chose a real-world radio controller to verify our attack scheme. The target controller is composed of three components: a transmitter body, an RF module, and a receiver. The detailed brands and names are shown in Fig. 3. It adopts Advanced Continuous Channel Shifting Technology (ACCST), which is FrSKY's commercial name for FHSS. ACCST devices shift channels more than a hundred times per second for security and stability.

We first conducted basic examinations of the target system in order to apprehend its mechanism. We analyzed only the body and the RF module, since the analysis of the receiver is not required to verify the attack model.

Analysis of Transmitter Body For the selected target, the transmitter body only output a series of PWM pulses and passed them to the RF module. The PWM pulses were further modulated and up converted in the RF module. Having examined the transmitter body, we concluded that the transmitter body was not related to generating FHSS signals.

Analysis of the RF module The RF module was powered by a pin connecting it to the transmitter body, and modulated the input PWM pulses to generate the FHSS signal output. In order to analyze only the output signal of the RF module, we connected the module's output port and the CBX input port directly using a SubMiniature version A (SMA) cable. We then ran uhd_fft to view the spectrogram of the RF module's output signal. uhd_fft is a GUI application that makes USRP work as a simple spectrometer. Since we already knew that the RF module used 2.4 GHz bands, we first tuned uhd_fft to 2.4 GHz, and gradually changed the frequency. As a result, the center frequency of the first and the last FHSS channel were found to be 2.40517 GHz and 2.41415 GHz, respectively.

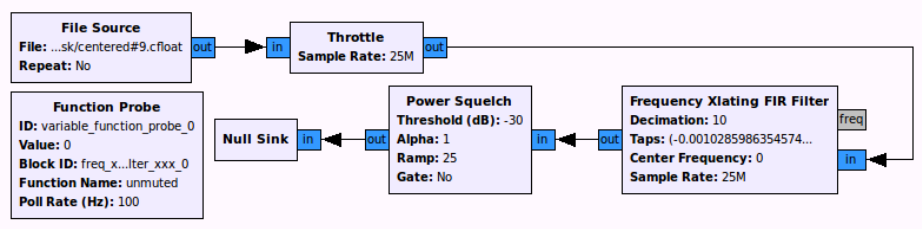

Fig. 4: GNU radio flow graph for partial sequence extraction

From these observations, we identified the total number of FHSS channels and their center frequencies.

To summarize, there were 47 channels in total, and each channel was 1.5 MHz wide. The total bandwidth was calculated as below.

$$\left[(2.40517 \times 10^9 - 1.5 \times 10^6/2) - (2.47415 \times 10^9 + 1.5 \times 10^6/2)\right] \text{Hz} \approx 70\text{MHz}$$

It was approximately three times larger than the maximum bandwidth of USRP N210 (25 MHz), which was the case described in Section 3.2

4.3 FHSS Sequence Extraction

Hopping Speed The hopping speed of the target system is important because USRP has limited agility. If the hopping speed is too high, it is impossible to follow the changing frequency of the target system.

The simplest method of measuring hopping speed is to measure the duration of a hop, since typical FHSS systems have a constant hopping speed. To measure the duration, we first tuned the USRP to one of the channels and recorded the signal into a file. We subsequently browsed the recorded file to measure the duration of a hop, which was 0.0058 s. Converting this duration directly into hopping frequency, we derived $1/0.0058$ s ≈ 172 Hops/s. Note that this was the upper bound of the hopping speed, since no FHSS channel is typically changed without a delay.

Based on work by Nychis et al. [13], the hopping speed was in a range that USRP can readily follow without any modifications to the field-programmable gate array (FPGA) or the firmware. In their study, the overall round trip time between the host and the USRP was measured to be 612 μs on average with a standard deviation of 789 μs. With the measured hopping speed, only more than $+7\sigma$ cases would lead to missing activated hops.

Partial Sequence Extraction Following the basic analysis of the target system, we extracted the partial hopping sequences. As described in the previous subsection, the total bandwidth of the target system was approximately 70 MHz, much higher than the maximum bandwidth of the USRP (25 MHz). Therefore, we first acquired the partial hopping sequences of the target system.

In order to record the sequence, we built a GNU Radio flow graph as in Fig. 4. The flow graph was mainly composed of a Frequency Xlating FIR filter, Power Squelch, and Function Probe.

```
build flow graph structure;
while running flow graph do
    for each interval do
        for each channel do
            if squelch active then
                record(channel_number);
            end
            if no activation then
                record(0);
            end
        end
    end
end
```

Algorithm 1: Partial sequence extraction algorithm

Frequency Xlating FIR filter first operates as a channel selection filter. It tunes to the target channel and filters out other signals. Power Squelch and Function Probe are core parts of this flow graph. Power Squelch allows input signals to pass though only when the power level exceeds a preset threshold, and Function Probe monitors the state of Power Squelch to determine if it is open.

We parallelized the flow graph in Fig. 4 to simultaneously record multiple channels under USRP coverage. We set seven USRP coverages, and ran Algorithm 1 for each coverage to record the corresponding partial sequence. As a result, we finally acquired all partial hopping sequences, as listed in Tab. 1. From the table, we see that all channels were identically activated three times for each partial period. This confirmed that the target system uniformly unitizes its bandwidth.

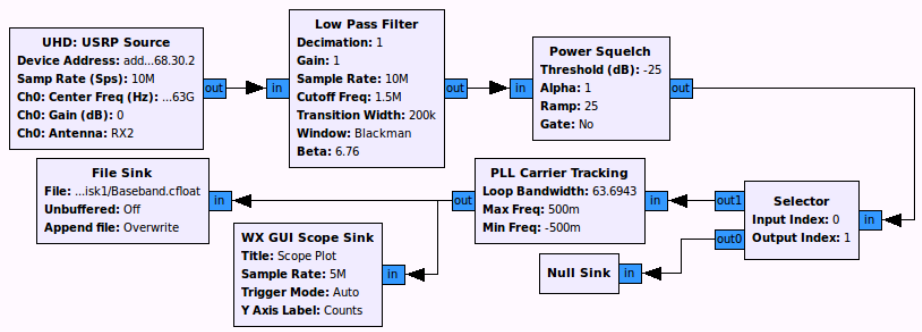

Fig. 5: GNU radio flow graph for the baseband extractor

Combining Partial Sequences In the final step, we combined acquired partial sequences. This step was not automated, since partial sequences can be woven manually due to their short lengths. We arranged partial sequences in a spreadsheet, and fit the activations of the overlapped channels together to find the complete hopping sequence. Though the coverages did not overlap maximally, a unique combination could be found. Tab. 2 shows the combined sequence.

Coverage #	Partial hopping sequence	Length
1 (Ch1∼Ch9)	7, 1, 6, 5, 4, 9, 3, 8, 2, 7, 1, 6, 5, 4, 3, 2, 1, 9, 8, 7, 6, 5, 4, 9, 3, 8, 2	27
2 (Ch1∼Ch17)	7, 1, 12, 6, 11, 5, 10, 4, 9, 3, 8, 2, 7, 1, 6, 17, 5, 16, 4, 15, 3, 14, 2, 13, 1, 12, 17, 11, 16, 10, 15, 9, 14, 8, 13, 7, 12, 6, 17, 11, 5, 16, 10, 4, 15, 9, 3, 14, 8, 2, 13	51
3 (Ch9∼Ch25)	12, 23, 11, 22, 10, 21, 9, 20, 25, 19, 24, 18, 23, 17, 22, 16, 21, 15, 20, 14, 25, 19, 13, 24, 18, 12, 23, 17, 11, 22, 16, 10, 21, 15, 9, 20, 14, 19, 13, 18, 12, 17, 11, 16, 10, 15, 9, 14, 25, 13, 24	51
4 (Ch17∼Ch33)	26, 31, 25, 30, 24, 29, 23, 28, 22, 33, 27, 21, 32, 26, 20, 31, 25, 19, 30, 24, 18, 29, 23, 17, 28, 22, 27, 21, 26, 20, 25, 19, 24, 18, 23, 17, 22, 33, 21, 32, 20, 31, 19, 30, 18, 29, 17, 28, 33, 27, 32	51
5 (Ch25∼Ch41)	41, 29, 40, 28, 39, 27, 38, 26, 37, 25, 36, 41, 35, 40, 34, 39, 33, 38, 32, 37, 31, 36, 30, 41, 35, 29, 40, 34, 28, 39, 33, 27, 38, 32, 26, 37, 31, 25, 36, 30, 35, 29, 34, 28, 33, 27, 32, 26, 31, 25, 30	51
6 (Ch33∼Ch47)	44, 43, 42, 47, 41, 46, 40, 45, 39, 44, 38, 43, 37, 42, 36, 47, 41, 35, 46, 40, 34, 45, 39, 33, 44, 38, 43, 37, 42, 36, 41, 35, 40, 34, 39, 33, 38, 37, 36, 47, 35, 46, 34, 45, 33	45
7 (Ch39∼Ch47)	44, 43, 42, 47, 41, 46, 40, 45, 39, 44, 43, 42, 47, 41, 46, 40, 45, 39, 44, 43, 42, 41, 40, 39, 47, 46, 45	27

Table 1: Extracted partial sequences for each coverage

Combined partial periods	7, 1, 36, 30, 24, 12, 6, 47, 35, 29, 23, 11, 5, 46, 34, 28, 22, 10, 4, 45, 33, 27, 21, 9, 3, 44, 32, 26, 20, 8, 2, 43, 31, 25, 19, 7, 1, 42, 30, 24, 18, 6, 47, 41, 29, 23, 17, 5, 46, 40, 28, 22, 16, 4, 45, 39, 27, 21, 15, 3, 44, 38, 26, 20, 14, 2, 43, 37, 25, 19, 13, 1, 42, 36, 24, 18, 12, 47, 41, 35, 23, 17, 11, 46, 40, 34, 22, 16, 10, 45, 39, 33, 21, 15, 9, 44, 38, 32, 20, 14, 8, 43, 37, 31, 19, 13, 7, 42, 36, 30, 18, 12, 6, 41, 35, 29, 17, 11, 5, 40, 34, 28, 16, 10, 4, 39, 33, 27, 15, 9, 3, 38, 32, 26, 14, 8, 2, 37, 31, 25, 13 (Length = 47×3 =141)

Table 2: Acquired total hopping sequence

4.4 Baseband Extractor

Having successfully extracted the hopping sequence, we programmed the USRP to follow and record the target FHSS signal. In order to do that, we built a GNU radio flow graph as in Fig. 5. In the flow graph the incoming signal is first filtered by Low Pass Filter block. Power Squelch then senses the activeness of the current channel. Selector is initially headed to the null sink in order not to record meaningless signals, and is switched to the remaining flow graph when the USRP catches the FHSS signal. Once Selector is switched, PLL Carrier Tracking block finely tunes on the signal stream. Finally, the signal stream is recorded to an output file and visualized in real time. The flow graph is dynamically controlled using a Python script. It first commands the USRP to monitor one of

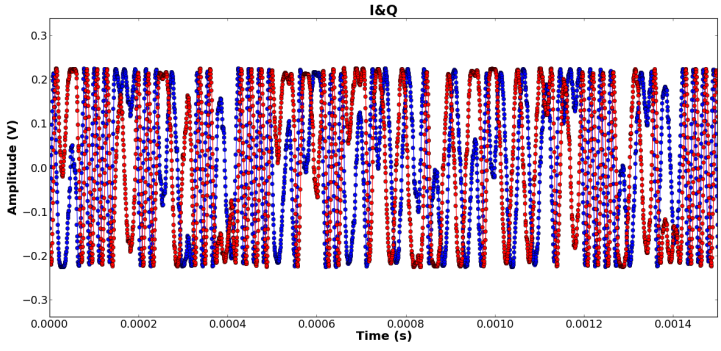

Fig. 6: Part of the extracted baseband

the channels. When the channel being awaited is activated, the script compares the state of its internal counter at the moment with the incoming signal. If they match, the script switches Selector and starts recording the signal stream. Otherwise, it is reset. As a result, we successfully extracted the raw baseband signal of the target system. Fig. 6 shows a part of the extracted signal.

5 Related Work

5.1 Drone Security

Several attack trials have shown that drones are quite vulnerable. With regard, for instance, to the RQ-170 USAF drone mentioned in the introduction, the Iranian government claimed that it had captured the drone through its cyber-warfare unit [21]. Although some debates on the attack means exist, the captured drone seems quite intact, which means it was not shot down by projectiles.

With regard to civilian drones, Todd Humphreys et. al. insisted that civilian drones are threatened by GPS spoofing [10]. It was even demonstrated that drones are hijacked by spoofing GPS signals with a custom GPS spoofer [10]. Kamkar recently announced SkyJack, a specialized drone hijack platform that targets only Parrot AR drones [12]. It exploits a WiFi hotspot vulnerability in AR drones to acquire control over them.

5.2 FHSS Security

FHSS is widely adopted to various communication devices for the motive of securing transmission, as its rapid pseudorandom frequency shift apparently makes FHSS systems resilient against eavesdroppers or jammers to a certain extent. However, not a few researches indicate FHSS alone cannot completely secure the contents being transferred. Song et. al. presented several algorithms for breaking the pseudorandom FHSS sequence with external observation [16]. Presented algorithms were theoretically analyzed, and some were simulated with C++ software, which is different from our work where the attack scheme is implemented and verified in reality. Furthermore, Song et. al. assume omniscience in receiving the target signal, and thus limited receiver bandwidth was not considered in their work. Q et. al. utilized a low-cost hardware equipped with commercial RF Integrated Circuits (RFIC) to implement a hopping sequence analyzer [14]. With the tool implemented, they successfully extracted hopping pattern in the 902-928MHz spectrum. However, their work is different from ours in some aspects. First, the presented approach can only be applied to spectrum to which the RFIC used can be tuned, whereas our SDR approach is much more flexible. Second, overcoming the limited receiver bandwidth was not covered in their work.

5.3 Bluetooth Security

Bluetooth is among the best-known communication standards that use FHSS. The wide adoption of Bluetooth in input devices suggests the likelihood of critical attacks. Especially, Bluetooth Low Energy (BLE) adopts a much simpler hopping and key sharing mechanism than classic Bluetooth. Mike Ryan has claimed [15] that the hopping sequence of BLE can be identified by collecting

empty data packets, and attackers can sniff ongoing links. He used Ubertooth, a programmable BLE sniffer, to extract parameters required to acquire the hopping sequence, and brute-forced the encryption key, which enabled BLE sniffing.

6 Discussion and Future Works

Attack research on drones has not only the meaning of attacking a system. It is also highly related to privacy protection, infrastructure security, and defense, since drones are becoming severe threats against them. Drone control system is apparently one of the major attack vectors against drones. This work deals with the very first step of attacking FHSS-type control system by acquiring the hopping sequence, which is essential to realize attacks.

For future works, first, we will analyze the baseband signal to reveal its structure. If it is not encrypted, our attack platform can operate as a sniffer, which can monitor control signals. This will enable the attacker to predict the movement of the target drone. Additionally, carefully crafted spoofing waveform can take control of the target drone, which will give the defender safely capture the target drone. Second, we will automate the process of combining partial hopping sequences to make the presented attack scheme applicable to general wideband FHSS devices.

7 Conclusion

In this paper we proposed a general scheme to extract the hopping sequences of FHSS-type RC drone controllers and showed its effectiveness using an SDR. We also proposed a scheme to overcome the issue of the limited bandwidth of the SDR and showed that it was effective by successfully extracting the baseband signal of a target system in an experiment. Our work can be extended to be implemented on jammers, sniffers, and spoofers against RC controllers.

References

1. 2.4GHz Radio Control Explained, `http://www.rcmodelreviews.com/spreadspectrum01.shtml`
2. Amazon Prime Air, `http://www.amazon.com/b?node=8037720011`
3. CBX 1200-6000 MHz Rx/Tx (40 MHz), `http://www.ettus.com/product/details/CBX`
4. DHL launches first commercial drone 'parcelcopter' delivery service, `http://www.theguardian.com/technology/2014/sep/25/german-dhl-launches-first-commercial-drone-delivery-service`
5. How compatible are 2.4GHz RC systems?, `http://www.rcmodelreviews.com/rxcompatibility.shtml`
6. USRP N210 Datasheet, `http://www.ettus.com/content/files/~07495_Ettus_N200-210_DS_Flyer_HR_1.pdf`
7. Chaudhuri, S., Chatterjee, S., Katz, N., Nelson, M., Goldbaum, M.: Detection of blood vessels in retinal images using two-dimensional matched filters. IEEE T-MI 8(3), 263–269 (1989)
8. Chen, Q., Defrise, M., Deconinck, F.: Symmetric phase-only matched filtering of fourier-mellin transforms for image registration and recognition. TPAMI 16(12), 1156–1168 (1994)

9. Gardner, W., et al.: Exploitation of spectral redundancy in cyclostationary signals. IEEE Signal Processing Magazine 8(2), 14–36 (1991)
10. Humphreys, T.E., Ledvina, B.M., Psiaki, M.L., OHanlon, B.W., Kintner Jr, P.M.: Assessing the spoofing threat: Development of a portable gps civilian spoofer. In: ION GNSS+. vol. 55, p. 56 (2008)
11. James, M.: What are DSM RC Controllers and Receivers and What Do They Do?, http://rcvehicles.about.com/od/frequency/f/dsmtechnology.htm
12. Kamkar, S.: SkyJack, http://www.samy.pl/skyjack/
13. Nychis, G., Hottelier, T., Yang, Z., Seshan, S., Steenkiste, P.: Enabling mac protocol implementations on software-defined radios. In: NSDI. vol. 9, pp. 91–105 (2009)
14. Q, Atlas, Cutaway Smash, Slugs on Toast: Hop hacking hedy (2011), https://www.youtube.com/watch?v=aMBaO94Q49U
15. Ryan, M.: Bluetooth: With low energy comes low security. In: WOOT (2013)
16. Song, M., Allison, T.: Frequency hopping pattern recognition algorithms for wireless sensor networks. In: ISCA. pp. 264–269 (2005)
17. Stahlberg, M.: Radio jamming attacks against two popular mobile networks. In: Tik-110.501. vol. 3 (2000)
18. Tang, H.: Some physical layer issues of wide-band cognitive radio systems. In: DySPAN. pp. 151–159. IEEE (2005)
19. Urkowitz, H.: Energy detection of unknown deterministic signals. Proceedings of the IEEE 55(4), 523–531 (1967)
20. Wikipedia: General Atomics MQ-1 Predator (2015), http://en.wikipedia.org/wiki/General_Atomics_MQ-1_Predator
21. Wikipedia: Iran-U.S. RQ-170 incident (2015), http://en.wikipedia.org/wiki/Iran%E2%80%93U.S._RQ-170_incident
22. Wikipedia: Sequence alignment (2015), http://en.wikipedia.org/wiki/Sequence_alignment
23. Yücek, T., Arslan, H.: Spectrum characterization for opportunistic cognitive radio systems. In: MILCOM. pp. 1–6. IEEE (2006)
24. Yücek, T., Arslan, H.: A survey of spectrum sensing algorithms for cognitive radio applications. IEEE Communications Surveys & Tutorials 11(1), 116–130 (2009)