# Garage Door Openers: A Rolling Code Protocol Case Study

Ahmed Ghanem
*ECE Dept. University of Victoria*
*British Columbia, Canada*

Riham AlTawy
*ECE Dept. University of Victoria*
*British Columbia, Canada*

2022 19th Annual International Conference on Privacy, Security & Trust (PST) | 978-1-6654-7398-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/PST55820.2022.9851991

***Abstract*—Rolling code is a keyless access protocol used prominently for garage doors and vehicles entry. In this work, we examine the security of three garage door opener systems which are widely used in the north American markets. Such openers are electronically controlled by wireless remotes and mobile applications. We reverse engineer their rolling code protocol and demonstrate practical attacks that enable an adversary to open the garage door after wirelessly sniffing only one open/close signal produced by the remote control device owner. Our security analysis reveals that such attacks are due to vulnerabilities in the deployment of the rolling code protocol in two out of the three investigated brands. Responsible disclosure procedures have been followed prior to the dissemination of our results.**

***Index Terms*—Garage door openers, Rolling code, Keeloq**

## I. Introduction

These days garage door openers become widely used in our houses and most apartment buildings to secure parking areas. Almost all of the people's houses can be accessed from their garages. For this reason, the security of the garage door openers is becoming crucial as breaking the security would make people and their assets vulnerable. In what follows we discuss the different types and techniques of garage door openers.

### *A. Fixed code garage door opener*

This generation of garage door openers used no cryptography. The remote control sends a fixed code to the gate, and if the gate recognizes the code, it opens. In the transmitter, there is a DIP switch that has a group of small switches controlling the transmitting code, which means a limited number of codes can be used, for example, the 12 DIP switches produce 4096 different codes that can be used to program a garage door opener. This type of code is vulnerable to replay attacks. Furthermore, it can be broken by brute-forcing all possible combinations of the DIP switch [1].

### *B. Rolling code*

Sometimes also called hoping code, this system uses cryptography for encryption. The plaintext for generating a new encrypted rolling code signal consists of a counter value that increases with every button press and other inputs. At the gate opener's side, after decryption, if the Unique Identifier (UID) of the remote control is known to the gate opener, the counter value is compared to the last valid counter value previously received (i), and if the value is somewhere between i+1 and i+n, it is considered new and therefore accepted, where n is the validity window for counter values that can be any defined number by the manufacturer. And it is used just in case the button is pressed away from the gate. The rolling code mechanism protects against replay attacks since each transmitted code is no longer valid once it has been received by the gate.

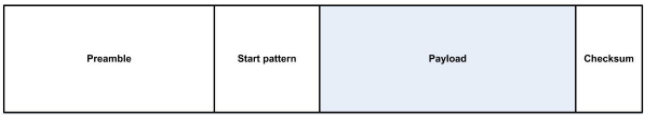


Fig. 1. Packet structure of a rolling code. The Payload part is encrypted

As shown in figure 1, the rolling code packet is divided into: The preamble which is a regular alternating sequence of zeros and ones. Start pattern that may be a sequence of ones or some fixed bits. The actual cryptographic payload which usually contains the Unique Identifier (UID) of the remote control, the rolling counter value, and the pressed button and finally, a checksum. Usually, the same packet is sent multiple times per button press for redundancy. Taking into consideration that many systems are slightly different from this general structure [2].

### *C. Example of a rolling code system based on keeloq cipher*

In most cases, garage door openers manufacturers are not interested in developing their own rolling code algorithm. Instead, they use the design of a specialized company. For example, Microchip developed a code hopping system based on keeloq cipher [3]. It is widely used for vehicle Remote Keyless Entry Systems (RKE) and garage door openers. The manufacturer only needs to feed the counter value into the encoder chip to be encrypted. On the gate's side, a similar chip does the decryption.

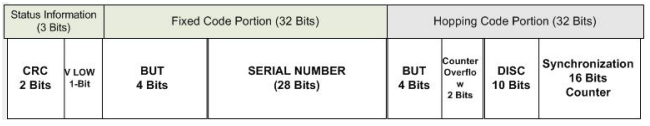


Fig. 2. Code word format produced by a Microchip HCS370 encoder.

As depicted in figure 2, the typical code word format produced by a Microchip encoder consists of a hoping portion

and a fixed code portion. The encrypted portion of the message or the hoping code portion contains the 16 bits counter value, the discrimination bits, and details of the button pressed on the remote. The discrimination bits are usually the lower 10 bits of the serial number, they are compared to the serial number on the decryption side to check the correctness of the decoded message. In the fixed code portion, the serial number of the key fob is transmitted without encryption.

The manufacturers may not follow the exact hoping algorithm as discussed above. However, they follow a similar format [3].

### D. Related work

The primary security goal of garage door opener systems is to prevent unauthorized access. Precisely, the gate should only open when an authorized user intends for the action to occur. In this section, we talk about previous analyses and attacks on garage door openers.

The replay attack is the most simple and easy, where an intruder would intercept and record the signal when someone presses the button to open their garage, then simply replay the signal and the garage door would open. This attack can work only with fixed code garage door openers. Another attack was demonstrated by Samy Kamkar [4] against fixed code garage door openers. Since they are using code set by DIP switches and there are not that many combinations, it can be brute-forced in about 30 minutes. He further reduced that to three minutes by removing the waits between transmissions. Finally, he was able to crack any fixed code garage door opener in eight seconds using a children's toy. In the literature, there are some successful cryptanalytic attacks on Keeloq [5] [6] and differently by side-channel attacks [7] [8].

A different, simple but effective method used by criminals to break the rolling code is the "rolljam attack". The RF signal transmitted from a modern key fob and received by the associated garage door opener is only used once. However, the code must be received by the garage door opener before it can be added to the list of used codes. The trick is to jam the receiver while one captures the code from the transmitter, doing this two times will give two valid codes, the first one can be re-transmitted to open the gate, which the user thinks is normal behavior. Practically one may have a device mounted near the garage that is always one code behind the most current. When the attacker wants to open the gate, the button on this device is all it takes. This attack is valid for rolling code-based RKE vehicles and garage door openers [9].

### E. Contribution and results

In this paper, we study some widely used garage door opener systems. In our analysis, we assume a man-in-the-middle adversary that can intercept, modify and create wireless signals between the user and the garage door opener. We reveal some vulnerabilities in their deployment of rolling code RKE which affects thousands of people. We look at brands sold at big Canadian stores like Home Depot and Canadian tire and analyze the captured RF signals from the remote controls of the three garage door openers, Skylink ATR-1722CK, Mastercraft 046-0265-2 1/2HP, and Chamberlain 050ACTWF. Specifically, those brand models lie at the middle of the price spectrum of automatic garage openers which suggests that they are widely desired by consumers. Skylink and Chamberlain are two of the top automatic garage door openers players in the north American market according to [10]. Mastercraft is the store brand of Canadian tire which makes it preferred and easy to buy for a wide range of consumers. We attempt to reverse engineer the format of the transmitted signals and produce new valid signals. In two out the three tested garage door openers, we are able to break the code and generate new valid signals from a single recorded keypress using minimal computations. Our results show the invalidity of the claims of such brands that they employ rolling code protocol [11].

*Experimental setup*. For our analysis, we use the Software-Defined Radio (SDR) HackRF to emulate a key and to eavesdrop and record rolling codes. We also use an open-source tool, the Universal Radio Hacker (URH), which is designed for RF protocols analysis. [12].

When studying remote controls of the three brands, We could not identify the type of microcontroller from the markings on the main IC, which complicates the reverse engineering

## II. Analysis of Skylink garage door opener

In the beginning, we identify the transmitting frequency of the remote control. It can be found precisely by following the FCC ID of the remote control which is 318MHz [13]. Then, we use the spectrum analyzer of the URH to find the actual operating frequency. Next, we start to capture several keypresses of the wireless remote using (Record Signal) in URH. The signal contains eight packets, the first packet is the smallest one and the following are similar. For some captured signals the last seven packets appeared to be different in shape due to both noise and interference as shown in figure 3. Regarding the bit representation, the packets looked random, not similar, and with different lengths, which was unexpected. We realized that we need to do some fine adjustments with the noise, center, samples/symbol, and Error tolerance. Also, we need to properly choose the modulation type in the interpretation tab.

By inspecting the signals, we can see that the used modulation is Amplitude Shift Keying (ASK) (See figure 4). However, changing the other parameters in the interpretation tab did not improve the bit representation, the recorded signals seemed to be affected by the recording environment, which was not ideal.

To get the actual bit representation, we apply a bandpass filter to the recorded signals to reduce the noise and interference then we re-adjust the other parameters in the interpretation tab. Finally, the bit representation started to be more logical.

Now we can see that the bit representation of the demodulated signals consists of 235 binary bits (59 hex digits), except the first packet consists of 33 binary bits (9 hex digits).

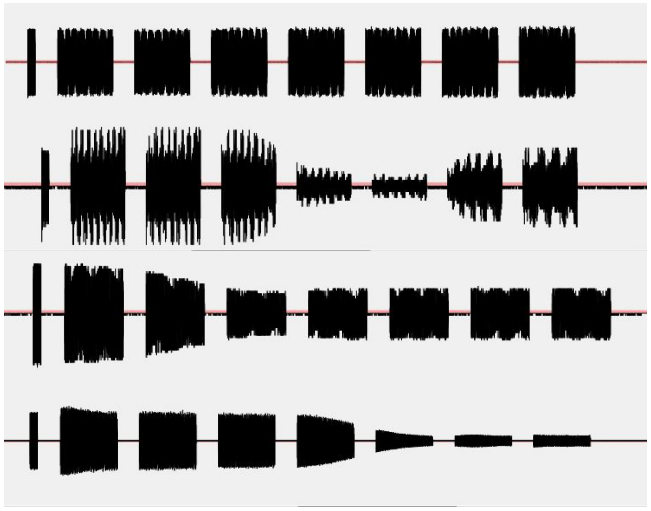

Fig. 3. 8 packets RF signals.The packets are different in shape due to noise and interference.

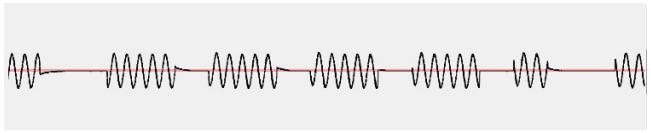

Fig. 4. Part of the signal showing that the used modulation is ASK.

### A. Reverse engineering

By comparing 15 captured signals from the same remote we notice that the first packet is 33 binary bits that are always ones, which may be used as an indication of the start of the signal. The last seven packets are identical for all the signals with the following structure: The first 72 binary bits (18 hex digits), and the last 64 binary bits (16 hex digits) are all the same for all the signals, and the bits from 73 binary (19 hex) to 172 binary (43 hex) except the binary bits from 125 to 132 (32 and 33 hex) are changing from signal to signal. By using a different wireless remote and comparing the captured signals to the previous analysis, we found that the binary bits 61, 67, 178, 184, 187, 190, and 193 (16, 17,45,46,47,48, and 49 hex) are changing by using a different wireless remote, Also, the binary bits 43, 46, and 214 (11, 12, and 54 hex) are changing by changing the pressed button in the same wireless remote. We deduce that the wireless remote transmits data in the form shown in figure 5. To test the success of our preprocessing

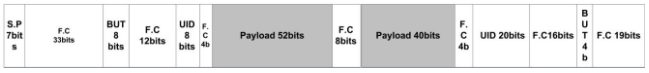


Fig. 5. Packet structure of a rolling code for Skylink. S.P: Start pattern, F.C: Fixed code, BUT: Details of the button pressed, UID: Identifier of the remote control.

procedure, we record some keypresses out of range from the gate opener, then we transmit them back in range, causing the motor to respond successfully. We also get the same result by sending the second packet only. Furthermore, to prove that the bandpass filter has no impact on the validity of the signals, we play back some filtered signals with the same results. It was interesting to find that the opener did not respond to the second playback of any of the previous signals, and that is one of the main properties of the rolling code.

### B. Our attack

Our goal is to build a valid 235 bits signal that can make the garage door opener respond, we can assume with certainty all of the 235 bits except the bits in the payload range, the total length of the payload range is 92 binary bits and this is the most challenging part of the signal.

It is not practical to brute-force all the 92 bits in the payload range, the challenge now is to find any relation between the different signals to produce any valid signal. We also have another challenge, even if we find a valid signal, it should be in the validity window of the gate opener, otherwise, the signal will be discarded.

When we look at the payload range in different signals, we find that the 92 bits are not completely different in all the signals, sometimes the difference is only 14 bits. So, by brute-forcing only 14 bits, we may get a valid signal.

In the “Generator” tab at the URH, we choose a recorded signal that has only 14 bits difference with another valid one to generate all the possible ”16384” combinations of those 14 bits. We use 65ms pauses between signals. Unfortunately, it is not practical to transmit the signals back to the gate opener as it take a long time to generate, modulate and transmit the signals. We observe that if we look at the hexadecimal representation of the 15 signals in figure 6, we find that the difference between two successive signals is in no more than 15 hex digits. Furthermore, we notice that the changing bits are taking two or four values only instead of the full hexadecimal range as shown in table I.

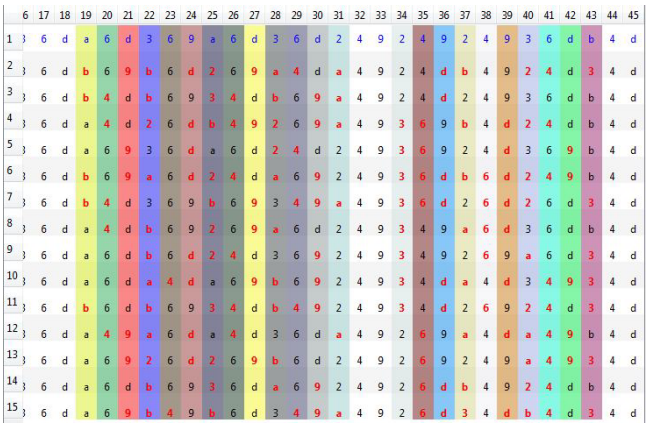


Fig. 6. The hexadecimal representation of the payload range in 15 recorded signals.

Now it is more clear, there are 5 hex digits in the encryption range that are taking four possible values, and the rest of the digits 18 are taking two possible values.

TABLE I
THE POSSIBLE VALUES OF THE ENCRYPTION RANGE

| Hex digit | (Possible values) | Hex digit | (Possible values) |
|---|---|---|---|
| 19 | a, b | 31 | a, 2 |
| 20 | 4, 6 | 34 | 2, 3 |
| 21 | d, 9 | 35 | 4, 6 |
| 22 | a, 3, 2, b | 36 | d, 9 |
| 23 | 4, 6 | 37 | a, 3, 2, b |
| 24 | d, 9 | 38 | 4, 6 |
| 25 | a, 3, 2, b | 39 | d, 9 |
| 26 | 6, 4 | 40 | a, 3, 2, b |
| 27 | d, 9 | 41 | 4, 6 |
| 28 | a, 3, 2, b | 42 | d, 9 |
| 29 | 4, 6 | 43 | b, 3 |
| 30 | d, 9 | - | - |

The total number of the possible combinations = $2^{18}X4^5 = 268,435,456$ possible codes.

Starting from one of the previously used signals in figure 3, we decide to choose randomly 7 ( 2 values) hexadecimal digits from the encryption range, to exhaustively generate all possible combinations resulting in 129 packets and send them to the gate opener, this can be done in the "Generator" tab at the URH.

Surprisingly, the gate opener responded several times during the first transmission of the 129 packets, and it was still responding when we repeated the transmission. The total number of working packets were 9 out of 129 packets with a probability of 0.06976, and this is a large number compared to the total number of trials 129 packets. We calculate the probability, $P$ of finding a valid signal code by $P$ = The number of valid codes in the receiving window / The total number of possible codes.

The number of valid codes in the receiving window is usually a small number around 250 codes when divided by the total number of possible codes 268,435,456 the result is $9.3 * 10^{-7}$. In other words, about 9 valid codes in every ten million trials. Accordingly, we conclude that there is no receiving window in this type of gate opener.

To get a better understanding of the attack and to find out why the gate opener is still responding when we repeated the same transmission, we keep sending only two from the working packets several times with 4 seconds pauses between them, the gate opener responded in approximately 50 percent of the total number of playbacks at least one time. This violates the concept of the rolling code, a previously used code should not be able to open the gate one more time. In our experiment, the gate opener failed to recognize the previously used codes in around 50 percent of the total number of trials.

For the analysis of the Mastercraft garage door opener, we found that it is exactly similar to Skylink. Thus we report the same results with the same vulnerabilities as the ones presented in this section on the Skyline model.

## III. ANALYSIS OF CHAMBERLAIN GARAGE DOOR OPENER:

To start the analysis, we capture some signals from the remote control to compare them and to inspect their bit representation to get a deep understanding of the code.

### A. Preprocessing

The transmitting frequency of the remote control was found to be 315 MHz from the FCC ID [14]. Next, we capture some keypresses using URH and hack rf (SDR).

We can see that each of the recorded signals contains eight packets figure 7, the packets look similar in shape and (ASK) modulation, the modulation type can be noticed clearly, by getting a closer look at the signals in the interpretation tab of the URH. After recording several signals, ignoring a lot of distorted ones, and adjusting noise, center, samples/symbol, and error tolerance, we get the actual bit representation. The first packet consists of 126 binary bits (32 hex digits), the last seven packets, three of them are 124 binary bits (31 hex digits), and the rest are 123 binary bits (31 hex digits).

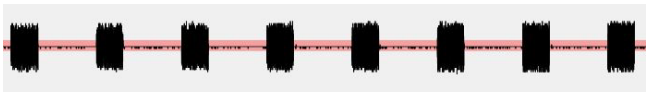

Fig. 7. A recorded signal contains eight packets

Figure 8 shows the bit representation of one of the recorded raw signals, we notice that the first packet is slightly different from the rest of the packets. However, the rest of the packets are not all the same, the odd rows are identical, and the even rows are the same. We also notice that there are no more than two consecutive zeros or ones, which indicates that the bits are encoded using Manchester encoding [15].

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | a | a | a | a | a | a | a | 9 | 5 | 6 | 6 | a | 9 | a | 6 | 9 | 9 | 9 | 5 | 5 | a | 6 | 5 | 9 | 9 | 5 | a | 5 | 6 | 5 | 4 |
| 2 | a | a | a | a | a | a | a | a | 5 | 5 | a | a | 6 | 6 | 6 | 9 | 6 | a | 6 | 9 | 6 | 6 | 9 | 6 | a | 9 | 6 | 6 | a | 9 | a | |
| 3 | a | a | a | a | a | a | a | a | 5 | 5 | 9 | a | a | 6 | 9 | a | 6 | 6 | 5 | 5 | 6 | 9 | 9 | 6 | 6 | 5 | 6 | 9 | 5 | 9 | 5 | |
| 4 | a | a | a | a | a | a | a | a | 5 | 5 | a | a | 6 | 6 | 6 | 9 | 6 | a | 6 | 9 | 6 | 6 | 9 | 6 | a | 9 | 6 | 6 | a | 9 | a | |
| 5 | a | a | a | a | a | a | a | a | 5 | 5 | 9 | a | a | 6 | 9 | a | 6 | 6 | 5 | 5 | 6 | 9 | 9 | 6 | 6 | 5 | 6 | 9 | 5 | 9 | 5 | |
| 6 | a | a | a | a | a | a | a | a | 5 | 5 | a | a | 6 | 6 | 6 | 9 | 6 | a | 6 | 9 | 6 | 6 | 9 | 6 | a | 9 | 6 | 6 | a | 9 | a | |
| 7 | a | a | a | a | a | a | a | a | 5 | 5 | 9 | a | a | 6 | 9 | a | 6 | 6 | 5 | 5 | 6 | 9 | 9 | 6 | 6 | 5 | 6 | 9 | 5 | 9 | 5 | |
| 8 | a | a | a | a | a | a | a | a | 5 | 5 | a | a | 6 | 6 | 6 | 9 | 6 | a | 6 | 9 | 6 | 6 | 9 | 6 | a | 9 | 6 | 6 | a | 9 | a | |

Fig. 8. The bit representation of one of the recorded raw signals

### B. Reverse engineering

To understand the bit representation and why the packets are not identical, we record some keypresses out of range

from the gate opener then we do the following: Sending a complete signal to the gate opener, and the gate responded. Sending the first packet only, but the gate did not respond. Sending each one of the following seven packets individually, with no response. Sending the first two packets, then the gate responded. Sending packets three and four, again the gate responded. Sending packets two and four, but the gate did not respond. We conclude that at least two consecutive packets should be transmitted to open the gate. Any single packet or two even/odd packets are not sufficient. Equally important, we notice that the first 48 binary bits (12 hex digits) are the same for all the recorded signals, and the rest of the bits are different.

Furthermore, it is important to decode the signal to study the bit representation of the decoded signal. This enables us to look at the actual ciphertext before encoding. In the analysis tab at the URH, we can choose the decoding to be (Manchester reverse) in this encoding system, a 1 bit is represented as high for the first half of the bit period and low for the second half, and a 0 bit is represented as low in the first half and high for the second half. It is also called Manchester II or Biphase-L code [15]. The bit representation of the decoded signal consists of 64 binary bits (16 hex digits) as shown in figure 9. Again, the same as before, the first packet is slightly different, and the even rows are identical, and the odd rows too. The first 24 binary bits (6 hex digits) are the same for all the signals, and the rest of the bits are changing. By using the Generator tab

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 2 | f | 6 | d | 4 | 1 | a | 5 | 1 | 8 | 8 |
| 2 | f | f | f | f | 0 | f | 5 | 6 | 7 | 6 | 5 | 9 | e | 5 | e | c |
| 3 | f | f | f | f | 0 | b | d | b | 5 | 0 | 6 | 9 | 4 | 6 | 2 | 0 |
| 4 | f | f | f | f | 0 | f | 5 | 6 | 7 | 6 | 5 | 9 | e | 5 | e | c |
| 5 | f | f | f | f | 0 | b | d | b | 5 | 0 | 6 | 9 | 4 | 6 | 2 | 0 |
| 6 | f | f | f | f | 0 | f | 5 | 6 | 7 | 6 | 5 | 9 | e | 5 | e | c |
| 7 | f | f | f | f | 0 | b | d | b | 5 | 0 | 6 | 9 | 4 | 6 | 2 | 0 |
| 8 | f | f | f | f | 0 | f | 5 | 6 | 7 | 6 | 5 | 9 | e | 5 | e | c |

Fig. 9. The bit representation of the decoded signal consists of 64 binary bits (16 hex digits)

at URH to send a signal, we can choose the encoding of the transmitted packets which is used to make sure that we chose the right decoding type, so we apply Manchester decoding to one of the out-of-range recorded signals, then we use the Generator tab at the URH to Manchester encode and transmit the signal back to the gate opener, and the gate responded successfully.

To get a better understanding of the signal structure and to find the bits of the button pressed, we define a different button in the remote control. Then we record some keypresses using the newly defined button, to compare the signals. However, we did not notice any difference which indicates that the pressed button bits are encrypted.

### C. Attempted attack

We look at the variable hex digits in 15 different signals to find all possible values of these digits. The analysis was made on both the recorded raw signals and the decoded ones. For the raw signals, we find that there are 19 hex digits which take only one of the following values (9, 6, 5, a) which is logical since the signal is Manchester encoded. Thus only the combination of (9, 6, 5, a) will prevent the occurrence of three successive zeros or ones. Overall, the variable hex digits are 10 (from 7 to 16) in the decoded signals for the last two packets, taking one of the values in Table II. We can choose some hex digits from the payload range of two consecutive packets to do an exhaustive search. However, URH can not search two packets at the same time, we had to find some way to go around that. If we concatenate two consecutive packets thus, we get a single packet with 62 hex digits. Next, we have to prove that the gate opener is responding to the signal after the modification. Again, we used one of the out-of-range recorded signals to concatenate the last two packets then, transmit them back, and fortunately, the gate opener responded. The same steps were repeated with a Manchester decoded signal with the same results, the packet size, in this case after concatenation, the signal was 32 hex digits. Accordingly, we deduce that the remote control is transmitting a signal with the structure shown in figure 10 for two concatenated packets. As a result,

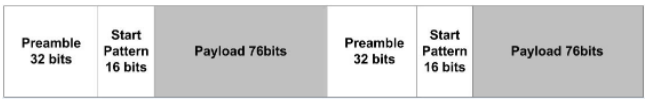


Fig. 10. Two concatenated packets structure of a rolling code for Chamberlain.

we have one (62 hex digits) packet with two payload ranges. Now, we can choose some digits from the payload range to generate all possible combinations using the values in table II and transmit them back to the gate opener. We selected up to 10 hex digits several times, but the gate opener did not respond which indicates that this brand of garage door openers is indeed enforcing the receiving window contrary to the other two brands.

## IV. Final Discussion

We note that we physically opened the remote and opener casing but found that the chip information wiped out by the manufacturers. From table I, we can see that the hex digits in the payload range of both Skylink and Mastercraft take only a few values of the hexadecimal range. Consequently, the encrypted bits are not fully random. Furthermore, the process of comparing the received codes or the counter value with the previous codes or the last stored counter value is strongly affected when removing the first packet which leads to failure to recognize the previously received codes. More tests on the products should take place. Using different packet

TABLE II
THE POSSIBLE VALUES OF THE PAYLOAD RANGE

| no/p.v | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|
| | 5 | 7 | e | 9 | 2 | d | 9 | c | 4 | 8 |
| | b | 5 | c | 5 | 7 | 5 | 1 | 0 | d | c |
| | d | 9 | 1 | f | f | 9 | 4 | 6 | 2 | 0 |
| | 9 | f | a | c | b | e | e | 5 | e | 4 |
| | f | d | 3 | 2 | a | 4 | f | 1 | 5 | 2 |
| | 7 | e | 6 | 0 | 4 | b | 5 | e | 7 | e |
| | a | 6 | 7 | 4 | e | 0 | 0 | 2 | 1 | - |
| | e | a | 5 | 1 | 8 | 2 | d | 7 | 8 | - |
| | 6 | b | 8 | b | 5 | 6 | c | 8 | c | - |
| | - | - | - | 6 | 3 | 7 | 7 | b | b | - |
| | - | - | - | 8 | 1 | 1 | b | a | a | - |
| | - | - | - | a | c | 8 | 3 | d | 6 | - |
| | - | - | - | d | 6 | f | 6 | - | 9 | - |
| | - | - | - | - | 0 | - | 8 | - | - | - |
| | - | - | - | - | - | - | a | - | - | - |

rolling codes in Chamberlain makes it more difficult for intruders. Only the receiving of two successive packets is sufficient to open the gate. We believe that the payloads in any two successive packets are related by a certain function which makes it harder to guess them both. Also, the receiving window of the rolling code makes it more complex for the exhaustive search to find a valid and within-range signal. From table II, we can see that the range of possible values of the payload for 15 recorded signals is wide which suggests a reasonable good encryption output.

## V. CONCLUSION

In this paper, we investigated the security of some widely spread garage door opener systems. We reverse engineered their rolling code protocols, and presented practical attacks on two brands out of three tested brands. Such attacks enable adversaries to gain unauthorized access with minimal interaction from the home owner's end. More preciesely, an adversary only has to eavesdrop on a single signal from a target remote control. Afterward, they can modify this signal to generate a number of valid signals to open the gate of the target home. This approach is considerably more stealthy and harder to prevent than the currently known methods of theft, e.g., using physical force or jamming the rolling code. Both Skylink and Mastercraft companies were informed by the results of this work and we received no response back.

## REFERENCES


[1] M. Brain, "How remote entry works," *HowStuffWorks, Inc*, 2001.

[2] F. D. Garcia, D. Oswald, T. Kasper, and P. Pavlidès, "Lock it and still lose it—on the ({In) Security} of automotive remote keyless entry systems," in *25th USENIX Security Symposium (USENIX Security 16)*, 2016.

[3] Microchip, "Hcs370 keeloq code hopping encoder data sheet." [Online]. Available: https://ww1.microchip.com/downloads/en/DeviceDoc/41111d.pdf

[4] S. Kamkar, "Opensesame." [Online]. Available: https://samy.pl/opensesame/

[5] A. Bogdanov, "Attacks on the keeloq block cipher and authentication systems," in *3rd Conference on RFID Security*, vol. 2007. Citeseer, 2007.

[6] S. Indesteege, N. Keller, O. Dunkelman, E. Biham, and B. Preneel, "A practical attack on keeloq," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2008, pp. 1–18.

[7] T. Eisenbarth, T. Kasper, A. Moradi, C. Paar, M. Salmasizadeh, and M. T. M. Shalmani, "On the power of power analysis in the real world: A complete break of the keeloq code hopping scheme," in *Annual International Cryptology Conference*. Springer, 2008, pp. 203–220.

[8] M. Kasper, T. Kasper, A. Moradi, and C. Paar, "Breaking keeloq in a flash: on extracting keys at lightning speed," in *International Conference on Cryptology in Africa*. Springer, 2009, pp. 403–420.

[9] S. Kamkar, "Drive it like you hacked it: New attacks and tools to wirelessly steal cars," *Presentation at DEFCON*, vol. 23, 2015.

[10] W. Market, "Automatic garage door openers market manufacturers, suppliers, vendors sales, revenue, market share 2022 to 2028," May 2022. [Online]. Available: https://www.marketwatch.com/press-release/automatic-garage-door-openers-market-manufacturers-suppliers-vendors-sales-revenue-market-share-2022-to-2028-2022-05-12

[11] Fred, Dcfar, HomeDepotCustomer, Debra, Brenda, Mike, Lisa, and CharInWA, "Skylink 3/4 hpf garage door opener chain drive with built-in led, remote control amp; keypad," Oct 2021. [Online]. Available: https://www.homedepot.ca/product/skylink-3-4-hpf-garage-door-opener-chain-drive-with-built-in-led-remote-control-keypad/1001182800

[12] J. Pohl and A. Noack, "Universal radio hacker: A suite for analyzing and attacking stateful wireless protocols," in *12th USENIX Workshop on Offensive Technologies (WOOT 18)*, 2018.

[13] F. ID, "Capital prospect keychain type transmitter mk318 fcc id kutmk318," Jul 2011. [Online]. Available: https://fccid.io/KUTMK318

[14] G. I. Chamberlain, "Chamberlain group , the remote control transmitter 7359 fcc id hbw7359," Jun 2011. [Online]. Available: https://fccid.io/HBW7359

[15] A. S. Tanenbaum, *Computer networks*. Pearson Education India, 2002.