Security Analysis of Rolling Code-based Remote Keyless Entry Systems

by

Ahmed Ghanem
B.Sc., Ain Shams University- Egypt, 2000
A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

Security Analysis of Rolling Code-based Remote Keyless Entry Systems

by

Ahmed Ghanem
B.Sc., Ain Shams University- Egypt, 2000

Supervisory Committee

Dr. Riham AlTawy, Supervisor
(Department of Electrical and Computer Engineering)

Dr. T. Aaron Gulliver, Department Member
(Department of Electrical and Computer Engineering)

ABSTRACT

Providing privacy and security is a critical issue in smart- homes. Many residents are concerned about unauthorized access to their homes. This work focuses on the security analysis of remote keyless entry systems (RKES) for automatic garage door openers.

Many of the RKES are unidirectional, but some are also bidirectional allowing for challenge-response authentication. The unidirectional RKES nowadays usually feature a rolling code. A rolling code is calculated on both the receiver (e.g., garage door) and the transmitter (e.g., key fob) and if there is a match of a received signal, the gate opens/closes. This way, the transmission is different every time, eliminating a simple replay attack. A widely used encryption algorithm for rolling codes is the Keeloq block cipher with is used to encrypt the value of the rolling code to prevent the generation of future valid codes.

To obtain a picture of the level of security that current rolling code-based automatic garage door openers systems provide, a selection of three of them are analyzed. The research uncover security vulnerabilities in two of them that enable an adversary to open the garage door after wirelessly sniffing only one open/close signal produced by the remote control device owner. In our analysis, we use the Software-Defined Radio (SDR) HackRF to emulate a key, and to eavesdrop and record rolling code signals. We also use the open-source tool Universal Radio Hacker (URH), which is designed for RF protocol analysis. Using these tools, we reverse engineer the structure of the signal used in the protocol, identify the encrypted code bits, and successfully pin out some bits that exhibit low randomness. By iterating over such bits, we successfully generate new signals that opens the garage door.

We also analyze the KeeLoq block cipher with respect to related key attack and present a chosen ciphertext attack for keys related by rotation. Our attack recovers the used 64-bit key with a negligible time complexity and data complexity of 66 chosen ciphertexts decrypted under 34 related keys.

Contents

List of Tables

List of Figures

List of Acronyms

RKE	Remote Key less Entry
URH	Universal Radio Hacker
SDR	Software Defined Radio
MITM	Man In The Middle
RKES	Remote Key less Entry Systems
ASK	Amplitude Shift Keying
PSK	Phase Shift Keying
FSK	Frequency Shift Keying
FSR	Feedback Shift Register
NLFSR	Non Linear Feedback Shift Register
NLF	Non Linear Function

Chapter 1

Introduction and motivation

Remote Keyless Entry (RKE) systems have become a common part of many people's daily lives. They are often used in vehicle remote controls, wireless garage door openers, and similar applications. Because they are intended to protect access to valuable assets and radio frequency communications are susceptible to eavesdropping and other possible manipulations, they typically use cryptographic measures to ensure the security of transmissions. Remote controls are usually low-power battery-powered devices so they commonly implement lightweight cryptosystems which are usually less resource-intensive than the cryptosystems used in traditional infrastructure. Some of them use KeeLoq cipher which is used in the rolling codes protocol [3, 8, 9].

The first part of this thesis investigates the security of the used RKE systems in garage door openers systems and identify vulnerabilities in their deployment. Using the Software-Defined Radio (SDR) HackRF and the open-source tool, the Universal Radio Hacker (URH), 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.

The second part of the thesis describes two different attacks against the Keeloq block cipher. The first one is a chosen ciphertext attack using keys related by rotation which recovers the key with a negligible time complexity and data complexity of 66 chosen ciphertexts decrypted under 34 related keys. The second one is fault attack against the encryption hardware that recovers the key by forcing the system to run more than 528 times with a negligible time complexity.

1.1 Problem statement and motivation

Garage door opener RKE systems are cyber-physical devices attached to our homes that can be electronically controlled by wireless remotes, and mobile applications [10]. These days they become widely used in our houses and most apartment buildings to secure parking areas. Consequently, the security of those systems is becoming crucial as breaking the security would make people and their assets vulnerable.

We examine the security of modern garage door opener RKE systems. Our goal is to analyze, and reverse engineer the wireless protocols, and look for vulnerabilities in their design that might lead to successful attacks.

Our security analysis mainly targets the systems which are widely used in the North American markets. We also go deeper and analyze the security of the underlying KeeLoq block cipher, the small and low-energy cipher mostly used in the design of the authentication mechanism. We review the previous attacks and look for implementation weaknesses.

1.2 Research objectives and organization

The goal of this research is to study the security of the garage door opener RKE systems, this will include the keeloq cipher and the rolling code protocol and provide useful results

to help the manufacturers to improve the security of their products and make those systems safer for people to use in their homes. The organization of this thesis is as follows:

Chapter 2 (Background). This chapter provides an overview on the role of cryptography, the basic definition of block ciphers and their security properties. We also the adversarial models for analyzing block ciphers. We define radio frequency signals, their digital modulation schemes, and data encoding. Finally, we talk about the tools used in our research.

Chapter 3 (Rolling codes in garage door openers). This chapter describes the different types of garage door openers RKES including the rolling code system and the previous attacks against such systems. The chapter also provides a case study of the security of three different brands which are widely used in the North American markets and demonstrates practical attacks against two of them. The contributions of this chapter are published in [11].

Chapter 4 (Keeloq block cipher cryptanalysis). This chapter provides the specifications of the Keeloq block cipher, authentication systems that employ it, and the previous attacks that were conducted against it. The chapter also presents two new different attacks against the Keeloq block cipher.

Chapter 5 (Conclusion). The conclusion and future work are given in this chapter.

Chapter 2

Background

This chapter provides general background about the concepts and tools that are used in this thesis.

2.1 The role of cryptography

The use of a cryptosystem is to provide the basic security features required by communicating entities. For example, if two users, Alice and Bob, exchange messages remotely, there is no guarantee that the exchanged messages will not be intercepted, modified, or fabricated. Furthermore, if they communicate wirelessly, it is quite easy for adversaries to either passively intercept and read their messages, or actively engage in the communication by editing, deleting, or inserting messages. Cryptography provides a framework of various cryptographic algorithms to provide basic security requirements for secure communication between Alice and Bob [12]. Such requirements include:

- Confidentiality. Keeping information secret from all but those authorized to see it.

- Integrity. Ensuring that information has not been modified by unauthorized or un-

known means.

- Entity Authentication. Confirming the identity of a communicating entity.

- Non-repudiation. Avoiding rejection of previous commitments or actions.

2.2 Block ciphers

Block ciphers belong to a class of cryptographic algorithms called symmetric encryption algorithms that aim to ensure data confidentiality by sharing a secret information between communicating parties. Such an information is used in the conversion of plaintexts into ciphertexts and vice versa so that the adversary is unable to retrieve plain text.

Due to the need to formalize secrecy in communication systems. in 1949, Shannon defined perfect secrecy [13]. A cipher provides perfect secrecy if the ciphertext does not provide the adversary with any additional information about plain text other than what is a priori known, e.g., language, length.

Shannon proved that such ciphers require a key that must be at least in the length of plaintext and cannot be reused for different plaintexts. This makes the ciphers with perfect secrecy impractical in most environments where large amounts of data are needed to be encrypted. So one needs other designs of encryption algorithms and other approaches to the definition of security concepts not so strict. Block ciphers represent one of such more efficient ways to construct encryption algorithms [9].

2.3 Security requirements

Block ciphers are expected to resist attacks where the adversary can have additional knowledge and capabilities than only observing random ciphertext. Extra knowledge may include known-plaintext, chosen-plaintext, chosen-ciphertext, and adaptive attacks [14].

The cryptographic community also considers block ciphers to resist additional attack scenarios [9]. For example, current block ciphers are analyzed for related-key attacks [15] in which the adversary is assumed to adopt a chosen-plaintext model while assuming that encryption is performed using different unknown keys that have a known/chosen difference. Also, to the utilization of block ciphers in constructing hash functions, analysis of block ciphers using the known-key and chosen-key models have become popular [12, 16].

2.4 Notions of security and adversary models

The most important question in cryptology is how secure a specific design is. The expected time and memory complexities of concrete attacks are often part of the answer and other two factors:

- The attack outcome: What goal the adversary can achieve with a certain attack?
- The adversary model assumed: what the adversary is allowed to undertake to perform a certain attack.

2.4.1 Attack outcomes

A classification of attack outcomes for block ciphers are given in [17] and are summarized by:

Key recovery. The attacker can determine the encryption key K. This is the most powerful attack outcome.

Global deduction. The adversary can compute the encryption and decryption without knowing the key K.

Instance deduction. The adversary can compute the encryption and decryption without knowing the key K for some plaintext p or ciphertext c.

Information deduction. The adversary can obtain some information about the key K or about unknown inputs to the encryption or decryption.

Distinguishing. The adversary can distinguish the encryption of a message p from a randomly drawn permutation [9].

2.4.2 Adversary models

The main adversary models of attacks on block ciphers are as follows:

Known plaintext attacks. The adversary knows some plaintexts processed by the cipher and their corresponding ciphertexts [9].

Chosen plaintext attacks. The adversary can choose plaintexts to be encrypted by the cipher before the attack and gets the corresponding ciphertexts during the attack [9].

Chosen ciphertext attacks. The adversary can choose ciphertexts to be decrypted by the cipher before the attack and gets the resulting plaintexts during the attack [9].

Adaptive chosen plaintext attacks. Based on the intermediate results of the attack, the adversary can choose plaintexts during the attack and gets the corresponding ciphertexts [9].

Adaptive chosen ciphertext attacks. Based on the intermediate results of the attack the adversary can choose ciphertexts during the attack and gets the corresponding plaintexts [9].

Related key attacks: The adversary can encrypt plaintexts or decrypt ciphertexts with the attacked key and with keys related to it by choosing some differences between the original key and the related keys [18].

2.5 Radio frequency signals

A radio frequency (RF) signal refers to a wireless electromagnetic signal used as a form of communication when talking about wireless electronics. Radio waves are a form of electromagnetic radiation with identified radio frequencies ranging from 9 kHz to 300

GHz [19]. The following function describes a signal over time (t)

$$\mathrm{Y(t)} = \mathrm{A} \times \sin(2\pi f t + \phi)$$

Where A is the amplitude of the sine wave, f is the frequency and ϕ is the phase shift. Basic modulation methods are based on modifying one of these components over time [1].

2.6 Digital modulation methods

Digital modulation is defined as the modulation algorithm in which discrete signals are used for modulating carrier waves and it is used for removing noise from the waves.

2.6.1 Amplitude Shift Keying

As depicted in figure 2.1, in amplitude-shift keying (ASK), the signal is modulated by changing the carrier wave's amplitude over time.

$$Y_{ASK}\ \mathrm{(t)} = \mathrm{A(t)} \times \sin(2\pi f t + \phi)$$

For binary ASK, 1 is represented by amplitude A_1, and 0 is represented by amplitude A_2. There is a particular variation of binary ASK called on-off keying (OOK), where A_2 is zero. It is widely used in short-range wireless applications, this includes home automation, industrial networks, wireless base stations, and remote keyless entry (RKE). OOK is often employed in battery-powered portable applications, as such systems can save power in the transmission by not sending the '0' bits [1, 20].

2.6.2 Frequency Shift Keying

As depicted in Figure 2.1, in frequency-shift keying (FSK), the signal is modulated by changing its frequency over time.

$$Y_{FSK}\ \mathrm{(t)} = \mathrm{A} \times \sin(2\pi f(t) t + \phi)$$

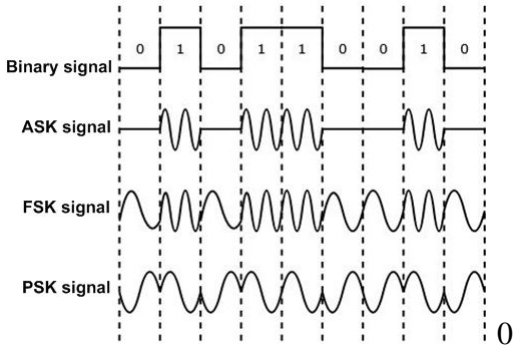

Figure 2.1: Basic digital modulation techniques [1].

For binary FSK, 0 is typically represented by f-δ, and 1 is typically represented by f+δ, where f is the center frequency of the carrier wave [1].

2.6.3 Phase Shift Keying

As depicted in figure 2.1, in phase-shift keying (PSK), the signal is modulated by changing the phase of the carrier wave over time.

$$Y_{PSK}\ (\mathrm{t}) = \mathrm{A} \times \sin(2\pi f t + \phi(t))$$

For binary PSK, 0 is represented by phase shift ϕ_1, and 1 is represented by phase shift ϕ_2, typically 180 degrees apart [1].

For higher data rates, more complex modulation schemes may be used. The modulation can be improved by encoding more bits into one state of a particular scheme. For example, quadrature PSK (QPSK) can be represented by four constellation points, meaning four possible combinations of 2 bits, with four distinct phase shifts (0, 90, 180, and 270 degrees) [1].

However, these complex modulations are not used in RKE applications. Remote controls are usually cost-effective battery-powered devices, so they usually use one of the binary modulation methods, such as ASK-OOK [20].

2.7 Data encoding

Transmitted binary data must be encoded before transmission over the air to avoid overlap and distortion. Several encoding schemes are commonly used. They are called line codes because they were designed to carry digital data on telecommunications transmission lines [2]. As depicted in Figure 2.2, the most common line codes include Non-return-to-zero (NRZ), Return-to-zero (RZ), Manchester (Biphase-L), Differential Manchester (Biphase-M, Biphase-S, and D.M.), and Bipolar code [2].

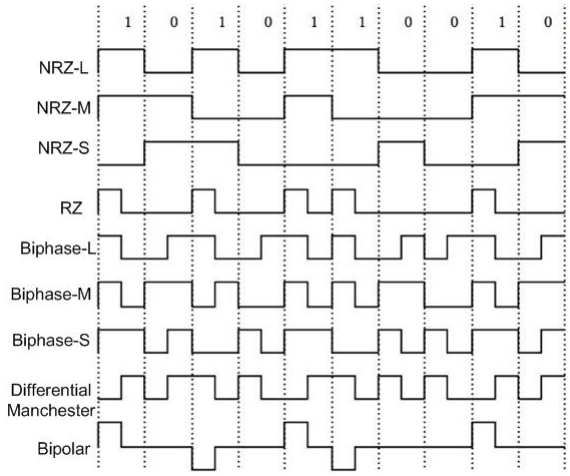

Figure 2.2: Mostly used line codes techniques [2].

2.8 HackRF one

The HackRF One from Great Scott Gadgets is a Software Defined Radio (SDR) peripheral that transmits or receives radio signals from 1 MHz to 6 GHz. It is designed to allow testing and development of modern radio technology. It is an open-source hardware platform that is often used as a USB peripheral or programmed for stand-alone operation. HackRF One acts as a computer sound card. It processes digital signals to the radio processes enabling the integration of extensive communication networks. It is intended for testing, developing,

improvising, and modifying current radio frequency systems [21].

2.9 Universal Radio Hacker

Universal Radio Hacker (URH) is a free tool that enables developers and security professionals to explore wireless protocols with a high level of accuracy. This tool is compatible with many SDR devices, including HackRF One, and several others [22].

As depicted in Figure 2.3, the interpretation screen provides basic RF processing of the captured signal via the capture tool that allows setting frequency, bandwidth, sampling rate, gain, and other relevant parameters. It is also possible to filter signal to noise cancellation.

Basic RF processing also consists of setting the center of the signal wave, the number of samples per period, noise level, and modulation type. URH supports only amplitude, frequency, and phase shift binary modulation. Also, built-in algorithms enable the automatic detection of some of the necessary parameters. Based on experiments, they provide reliable results when the modulation scheme is set manually. After the initial post-processing is complete, it creates a binary stream equivalent to the captured signal [22].

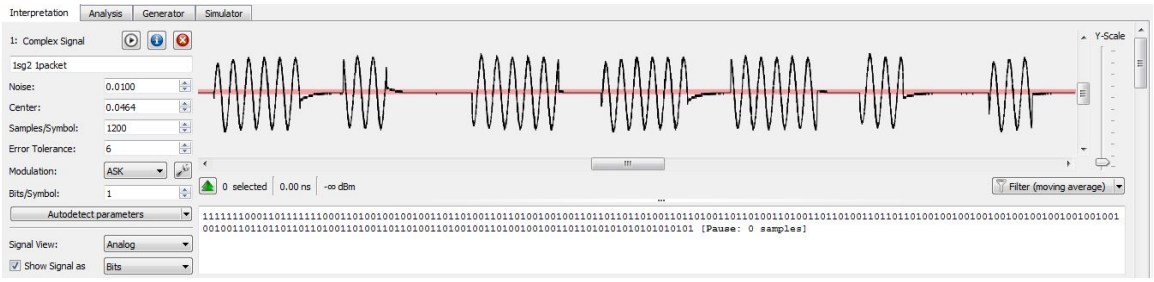

Figure 2.3: Part of the interpretation screen of the URH with ASK modulated signal loaded

As depicted in Figure 2.4, the analysis screen helps to explore the structure of the protocol. The binary data from the first screen is divided into individual message blocks, and the user can choose the correct encoding and mark the messages according to their type.

URH allows the user to choose from ready-made decoding algorithms such as Manch-

ester, NRZ, and so on, or generate your own decoding algorithm. Marked messages can be further tagged at the level of their fields [22].

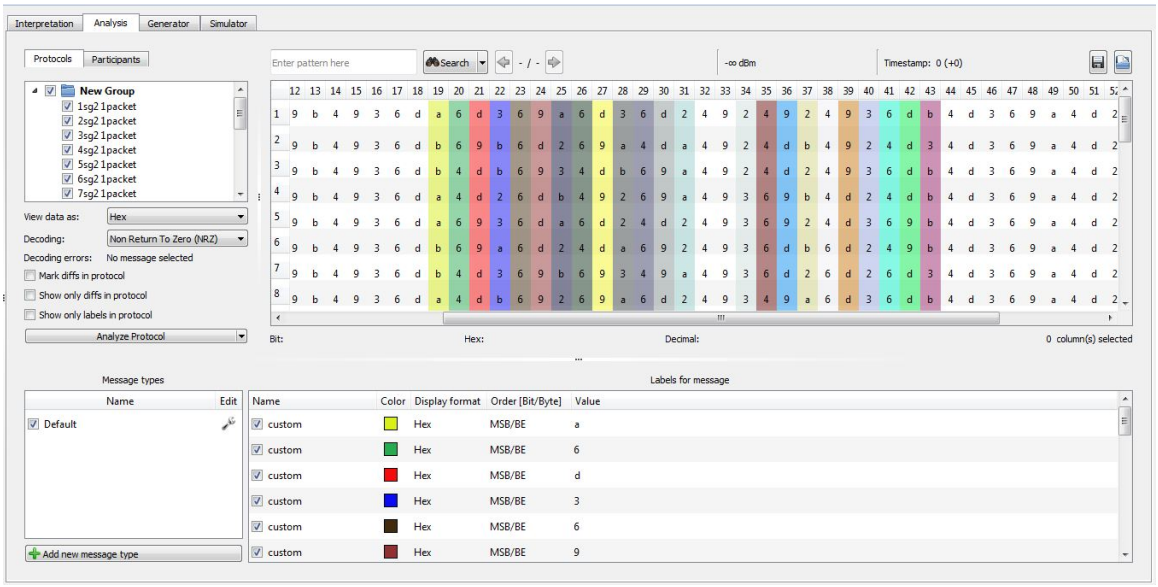

Figure 2.4: The analysis screen of the URH showing labeled messages

The generator screen allows the user to generate data for transmission. It can be used to replay data that was previously parsed. This feature has been used in our work, which increased work productivity because every data packet generated could be transmitted reliably.

Chapter 3

Garage Door Openers: A Rolling Code Protocol Case Study

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.

Rolling code is a keyless access protocol used prominently for garage doors and vehicles entry. In this chapter, 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. The contributions of this chapter are published in [11]. Responsible disclosure procedures have been followed

prior to the dissemination of our results.

In what follows we discuss the different types and techniques of garage door openers.

3.1 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 Figure3.1, 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.

Figure 3.1: Fixed code transmitter

This type of code is vulnerable to replay attacks. Furthermore, it can be broken by brute-forcing all possible combinations of the DIP switch [23].

3.2 Rolling code

Also known by hoping code, this system uses cryptography for encryption [3]. 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.

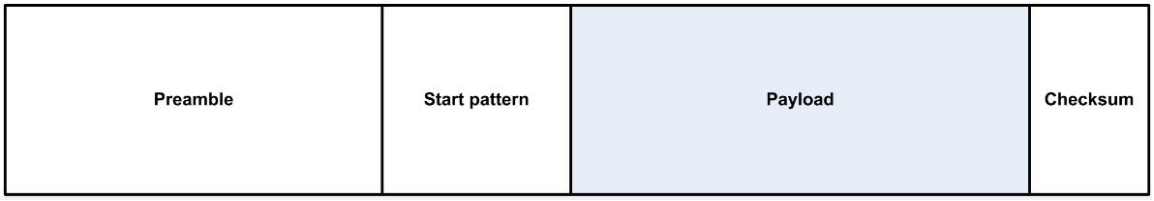

Figure 3.2: Packet structure of a rolling code. The Payload part is encrypted [3].

As shown in Figure 3.2, the rolling code packet is divided into:

- The preamble which is a regular sequence, meaning a sequence of ones, sequence of zeros, or alternating sequence of zeros and ones.

- Start pattern that may be a sequence of one 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.

- Checksum which is a value that represents the number of bits in a transmission packet.

Take into consideration that many systems are slightly different from this general structure [3].

3.3 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 [4]. 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.

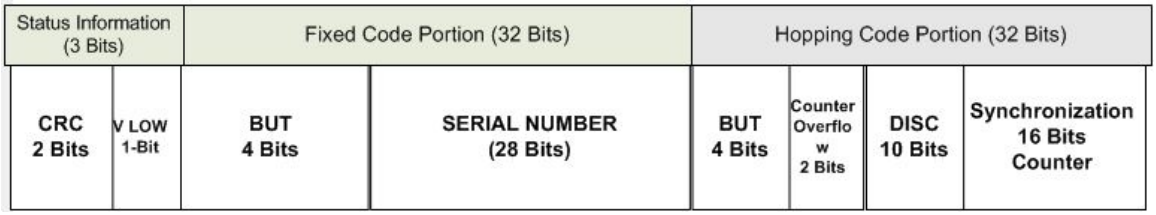

Figure 3.3: Code word format produced by a Microchip HCS370 encoder [4].

As depicted in Figure 3.3, the typical code word format produced by a Microchip encoder consists of a hopping portion and a fixed code portion.

The encrypted portion of the message or the hopping 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 [4].

3.4 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 discuss previous analysis and attacks on garage door openers.

The replay attack is the most simple and easy, when the attacker has hardware resources that allow capturing the signal, saving it, and transmitting it afterward. 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 [24] against fixed code garage door openers. Since they are using code set by DIP switches, there aren't that many combinations. Consequently, 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 [10].

A different simple but effective method used 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 the attacker captures the code from the transmitter. Doing this two times will give the attacker two valid codes, the first one can then be re-transmit to open the gate, which the user thinks is normal behavior and the second one can be later used for unauthorized access. Practically, the attacker would have a device mounted

near the garage that is always one code behind the most current. When the attacker wants to open the gate, pressing a button on this device is all it takes to send the last code which opens the gate.

This attack combines jamming and replays to provide a way to break most rolling code RKE systems. The idea is based on the fact that the frequency bandwidth of the receiver is quite broad.

As depicted in Figure 3.4, the jamming signal is transmitted at frequency f+δ, where f is the center frequency of the attacked RKE system. The reception of the signal is set to frequency f but is set to accept only frequencies in a narrow band around f, to filter out the jamming signal at frequency f+δ, which is intended only to jam the legitimate receiver.

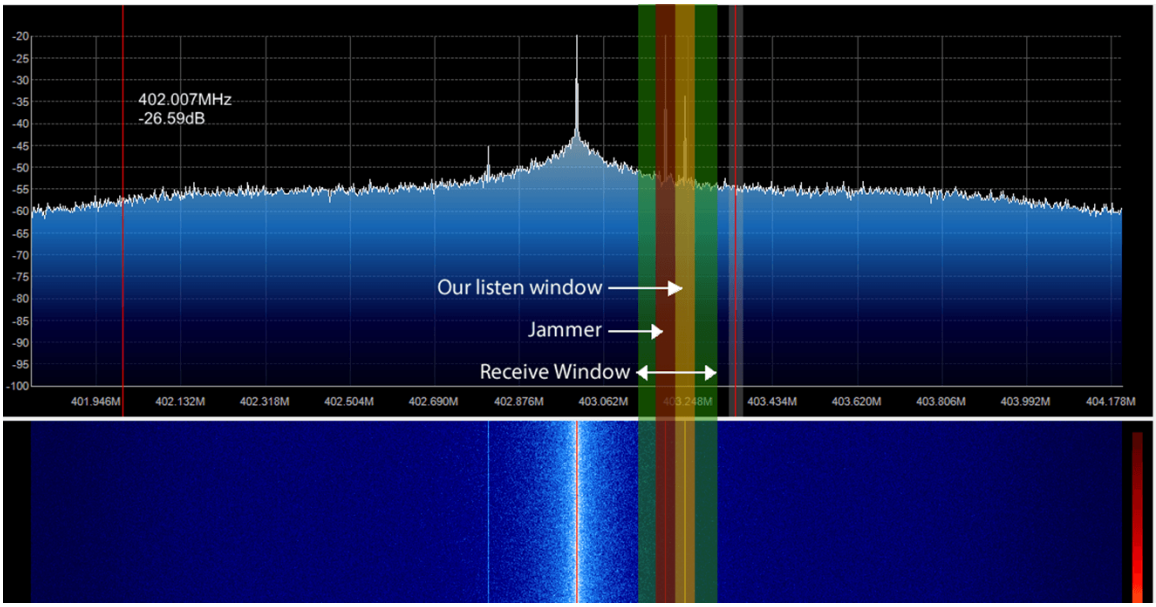

Figure 3.4: Rolljam attack [5]

At the beginning of the attack, the attacker starts with signal jamming. When the attacker accepts the first message, it's saved until it receives the second message, which is also saved. Right by that time, the attacker stops jamming and replays the first message which did not go through at the time of its original transmission because of the jamming signal interfering

with the receiver.

The legitimate user thinks that the second message went through as intended, but it is the first one that is accepted by the receiver. This allows the attacker to replay the second message later as its counter is still fresh for the receiver counter value, as the last message delivered contained a counter lower by one.

This attack is valid for rolling code-based RKE vehicles and garage door openers [25].

3.5 Contribution and results

In what follows, 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 [26]. 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 [27].

3.6 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 [22].

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.

3.7 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 [28].

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 3.5.

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.

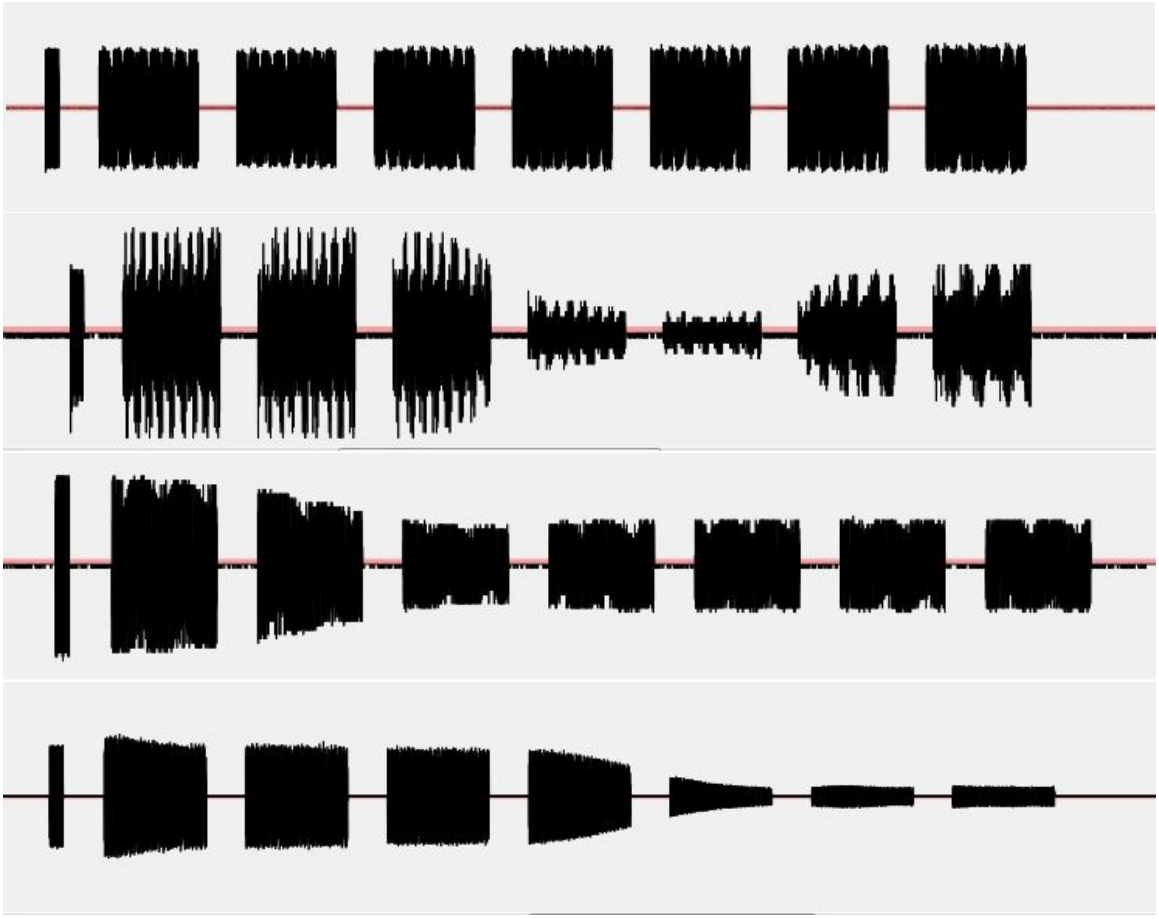

Figure 3.5: 8 packets RF signals.The packets are different in shape due to noise and interference.

By inspecting the signals, we can see that the used modulation is Amplitude Shift Keying (ASK) (See Figure 3.6). However, changing the other parameters in the interpretation tab did not improve the bit representation as the recorded signals seemed to be affected by the recording environment, which was not ideal.

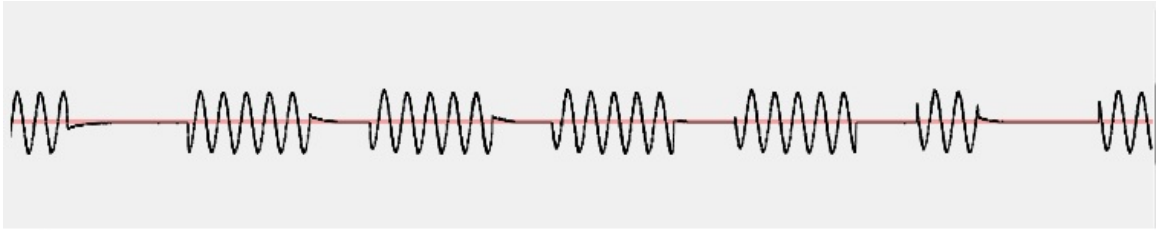

Figure 3.6: Part of the signal showing that the used modulation is ASK

To get the actual bit representation, we apply a band pass 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 and we found 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).

3.7.1 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 3.7.

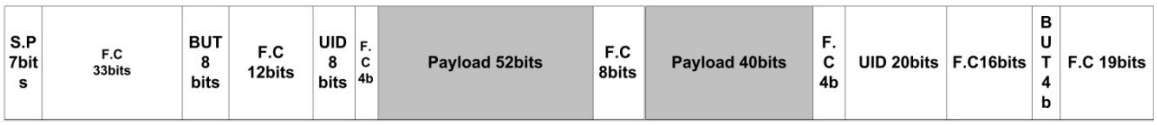

Figure 3.7: Packet structure of a rolling code for Skylink. S.P: Start pattern, F.C: Fixed code, BUT: Details of the button pressed, UID: Identifier of the remote control.

To test the success of our preprocessing procedure, we record some keypresses out of range from the gate opener and then transmit them back to the gate, 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 notice the opener did not respond to the second playback of any of the previously received signals, which is one of the main properties of the rolling code.

3.7.2 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 3.8, 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 3.1.

	6	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
1	3	6	d	a	6	d	3	6	9	a	6	d	3	6	d	2	4	9	2	4	9	2	4	9	3	6	d	b	4	d
2	3	6	d	b	6	9	b	6	d	2	6	9	a	4	d	a	4	9	2	4	d	b	4	9	2	4	d	3	4	d
3	3	6	d	b	4	d	b	6	9	3	4	d	b	6	9	a	4	9	2	4	d	2	4	9	3	6	d	b	4	d
4	3	6	d	a	4	d	2	6	d	b	4	9	2	6	9	a	4	9	3	6	9	b	4	d	2	4	d	b	4	d
5	3	6	d	a	6	9	3	6	d	a	6	d	2	4	d	2	4	9	3	6	9	2	4	d	3	6	9	b	4	d
6	3	6	d	b	6	9	a	6	d	2	4	d	a	6	9	2	4	9	3	6	d	b	6	d	2	4	9	b	4	d
7	3	6	d	b	4	d	3	6	9	b	6	9	3	4	9	a	4	9	3	6	d	2	6	d	2	6	d	3	4	d
8	3	6	d	a	4	d	b	6	9	2	6	9	a	6	d	2	4	9	3	4	9	a	6	d	3	6	d	b	4	d
9	3	6	d	a	6	d	b	6	d	2	4	d	3	6	9	2	4	9	3	4	9	2	6	9	a	6	d	3	4	d
10	3	6	d	a	6	d	a	4	d	a	6	9	b	6	9	2	4	9	3	4	d	a	4	d	3	4	9	3	4	d
11	3	6	d	b	6	d	b	6	9	3	4	d	b	4	9	2	4	9	3	4	d	2	6	9	2	4	d	3	4	d
12	3	6	d	a	4	9	a	6	d	a	4	d	3	6	d	a	4	9	2	6	9	a	4	d	a	4	9	b	4	d
13	3	6	d	a	6	9	2	6	d	2	6	9	b	6	d	2	4	9	2	6	9	2	4	9	a	4	9	3	4	d
14	3	6	d	a	6	d	b	6	9	3	6	d	a	6	9	2	4	9	2	6	d	b	4	9	2	4	d	b	4	d
15	3	6	d	a	6	9	b	4	9	b	6	d	3	4	9	a	4	9	2	6	d	3	4	d	b	4	d	3	4	d

Figure 3.8: 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.

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 3.8, 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

Table 3.1: 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	-	-

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 which equals to the number of valid codes in the receiving window divided by 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.

3.8 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.

3.8.1 Preprocessing

The transmitting frequency of the remote control was found to be 315 MHz from the FCC ID [29]. Next, we capture some keypresses using URH and Hack Rf (SDR).

We can see that each of the recorded signals contains eight packets (Figure 3.9), 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).

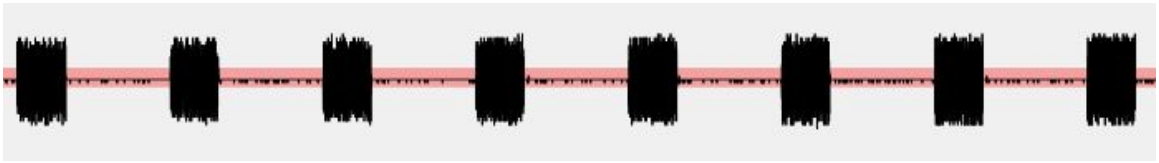

Figure 3.9: A recorded signal contains eight packets

Figure 3.10 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 [30].

	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	

Figure 3.10: The bit representation of one of the recorded raw signals

3.8.2 Reverse engineering

To understand the bit representation and why the packets are not identical, we recorded 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 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 noticed

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 [30].

The bit representation of the decoded signal consists of 64 binary bits (16 hex digits) Figure 3.11.

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 at URH to send a signal, we can choose the encoding of the transmitted packets, this can be used to make sure that we chose the right decoding type, so we applied Manchester decoding to one of the out-of-range recorded signals, then we used 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, hence, we record some keypresses using the newly defined button, to compare the signals. However, we did not notice any difference. It seems that the pressed button bits are encrypted.

	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

Figure 3.11: Bit representation of the decoded signal consists of 64 binary bits (16 hex digits)

3.8.3 Attempted attacks

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 3.2

We can choose some hex digits from the payload range of two consecutive packets to do an exhaustive search. However, the URH can not search two packets at the same time,

Table 3.2: 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	-	-	-

we had to find some way to go around that.

If we concatenate two consecutive packets thus, we will get a single packet with 62 hex digits, next, we have to prove that the gate opener will still be responding to the signal after the modification.

Again, we use 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 repeated with a Manchester decoded signal with the same results, the packet size, in this case after concatenation, was 32 hex digits. We can say that the remote control is transmitting a signal with structure in Figure 3.12 for two concatenated packets.

As a result, 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 table2 and transmit them back to the gate opener.

Preamble 32 bits	Start Pattern 16 bits	Payload 76bits	Preamble 32 bits	Start Pattern 16 bits	Payload 76bits

Figure 3.12: Two concatenated packets structure of a rolling code for Chamberlain

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.

3.9 Final discussion

We note that we physically opened the remote and opener casing but found that the chip information wiped out by the manufacturers. See Figure 3.13.

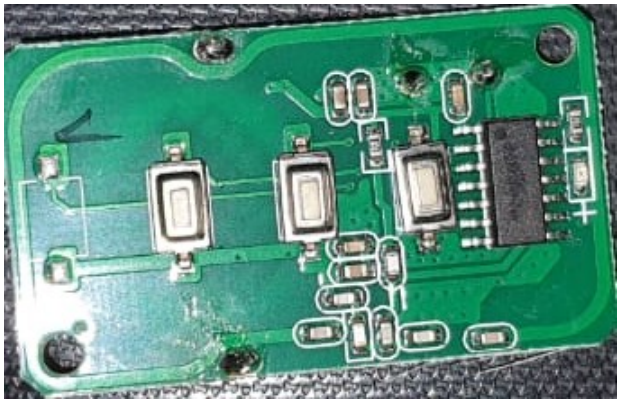

Figure 3.13: One of the tested remote controls, the chip information wiped out

From table 3.1, 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 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 3.2, 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.

Chapter 4

Keeloq cryptanalysis

The keeloq block cipher is based on a non linear feedback shift register NLFSR with a nonlinear boolean feedback function of 5 variables [6]. It consists of a 32-bit plain-text/cipher-text register, a 64-bit key register, and a nonlinear function (NLF). The registers are rotated in each of 528 encryption cycles where one bit of the key is XORed to the output of the NLF in each cycle. After the 528 cycles, the cipher-text is read from the 32-bit register.

Due to its lightweight architecture, the keeloq cipher is a low-cost and efficient for hardware implementation. The keeloq cipher is widely used among designers of remote keyless entry systems, automotive and burglar alarm systems, automotive immobilizers, and gate and garage door openers [31].

There are some published cryptanalytic attacks on the keeloq cipher [32] [33] [7]. The time complexity of the most efficient attack is 2^{48} and needs 2^{16} plain- ciphertext pairs. Therefore, from a cryptographic point of view, the keeloq cipher should be considered insecure. However, those mathematical attacks are impractical for the RKE systems that use rolling codes [34]. Nonetheless, they shed light on the security of its employed structure.

A power analysis attack that describes how to extract the encryption key of commercial RKE systems based on KeeLoq [35] is proposed using differential power analysis (DPA) [36]. The ideal platform for doing a DPA attack would be simply the Microchip HCS301 keeloq rolling code encoder [37]. The timing behavior of the chip can be expected preciesely since it always performs the same digital operations independent of the secret key. Consequently, the power consumption at each point in time of the acquired traces is always related to the same step of the KeeLoq cipher, and extracting the key with DPA is relatively simple [34].

A related-key attack using keys related by rotation is a chosen plaintext attack that exploits the simplicity of mixing the key into the state during encryption by encrypting related plaintexts with two related keys by rotation. This process recovers two key bits and can be repeated to recover the 64 key bits of the original key [7]. The attack needs 66 plaintexts encrypted under 34 related keys and negligible time complexity. The number of plaintexts and related keys could be reduced in exchange for higher time complexity by switching to an exhaustive key search after recovering a suitable number of key bits.

This chapter presents two different attacks against the keeloq cipher. The first one is a chosen ciphertext attack using keys related by rotation. We also discuss the tradeoff between the time complexity and the used number of ciphertexts and related keys.

In the second attack, we assume that we have access to the encryption hardware and we can control the clock to recover the secret key.

4.1 Keeloq algorithm

keeloq is the widely used encryption module in remote keyless entry systems. It is a proprietary hardware block cipher based on NLFSR [35]. KeeLoq is a block cipher with

a key size of 64-bits that operates on 32-bit plaintext and ciphertext. The design is built on a nonlinear feedback shift register NLFSR of length 32 bits with a nonlinear feedback function NLF of 5 inputs. The feedback depends linearly on two register bits, the next key bit taken from the shifted key register of length 64 bits, and the output of the non-linear function NLF.

4.2 Keeloq encryption

A data register that is used to store the 32-bit plaintext and a key register that is used to store the 64-bit key. Also, the cipher defines a nonlinear function (NLF) that has five input variables and one bit output.

Keeloq encryption algorithm process is depicted in Figure 4.1. The bits L_{31}, L_{26}, L_{20}, L_9, and L_1 from the data register are fed as inputs to the (NLF) to produce an output bit. Then, the output bit is xored with L_{16}, L_0, and the key bit K_0 to produce the first output that is fed back to the data register. The key register and the data register are shifted by one-bit to the right every clock cycle. After 528 clock cycles, a 32-bit cipher text is read from the data register [6].

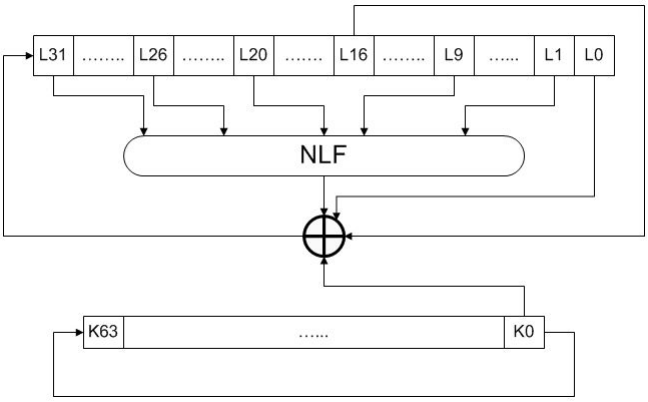

Figure 4.1: Keeloq encryption diagram [6].

The equations describing the encryption algorithm are given by:

$\alpha_i = NLF(L^i_{31}, L^i_{26}, L^i_{20}, L^i_9, L^i_1) \oplus L^i_{16} \oplus L^i_0 \oplus k_{imod64}$

$L^{i+1} = (\alpha_i, L^i_{31}, \cdots, L^i_1)\ 0 \geq i \geq 527,$

where the nonlinear function NLF is defined by

NLF(a,b,c,d,e) = $abc \oplus abd \oplus ace \oplus ade \oplus de \oplus cd \oplus be \oplus bc \oplus ae \oplus ac \oplus e \oplus d$

4.3 Keeloq decryption

As depicted in Figure 4.2, the decryption process of the keeloq is almost the same as the encryption process. However, the inputs to the (NLF) are different, and the registers are shifted in the opposite direction [6].

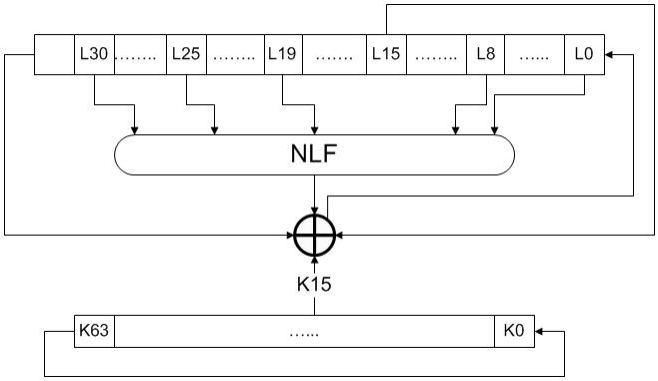

Figure 4.2: Keeloq decryption diagram [6].

The equations of the decryption algorithm are as follows:

$\alpha_i = NLF(L^i_{30}, L^i_{25}, L^i_{19}, L8^i, L^i_0) \oplus L^i_{15} \oplus L^i_{31} \oplus k_{i-1mod64}$

$L^{i-1} = (L^i_{30}, \ldots.., L^i_0, \alpha_i) \qquad i\ from\ 528\ to\ 1$

4.4 Previous attacks

Literature focusing on the cryptanalysis of the KeeLoq block cipher include [7, 32, 33]. In what follows, we discuss those attacks.

4.4.1 Slide attack

The main idea of the slide attack is to reduce the complexity of ciphers with the self-similar structure of their functioning. Commonly used for cryptanalysis of cipher with a large number of rounds.

As depicted in Figure 4.3, this attack needs to function as a known plaintext attack and the cipher must be able to break down to several iteration steps of the same function F.

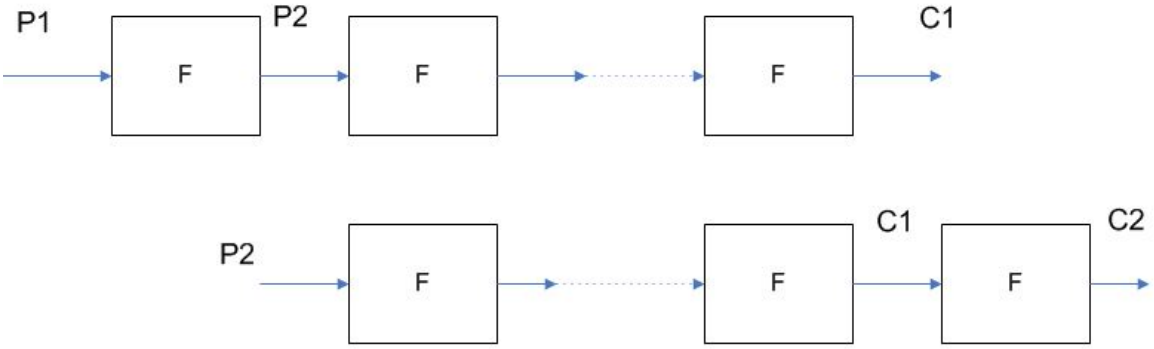

Figure 4.3: Slide attack [7]

To successfully perform a slide attack, a slid pair must be found. A slid pair is a pair of two plaintext-ciphertext pairs (P_1, C_1) and (P_2, C_2) that satisfies the condition that (F(P_1), F (C_1)) = (P_2, C_2), which means that this pair is shifted or slid by one iteration of the function F . Once this slid pair is found, an attack on the entire cipher could be simplified and broken in a single iteration function F because the slid pair serves as a known plaintext-ciphertext pair for the function F.

Finding a slid pair is not an easy process, but with the birthday paradox, the amount of

required plaintext/ciphertext pairs is limited to the square root of the ciphertext size. In the case of the KeeLoq cipher, this means that statistically a slide pair should be found in 2^{16} plaintext / ciphertext pairs.

The KeeLoq cipher does not meet the general assumptions of an attack, such as the number of rounds (528) is not divisible by 64, which is the size of the key and its shift register period. A solution to this problem is to guess the remaining 16 rounds specifically to reduce the number of rounds to attack to 512, which can effectively transform the problem to be solvable as a series of 8 iterations of 64 cipher rounds to meet the slide attack requirements [7].

4.4.2 Algebraic slide attack

An algebraic attack using the sliding property can successfully recover the key in an equivalent of 2^{53} KeeLoq encryptions [33]. To successfully execute the attack, one slide pair over 64 rounds of KeeLoq encryption of function F needs to be found. Once the slid pair (P_1, P_2) is found their corresponding ciphertexts (C_1, C_2), can be also considered as slide pair of function F, but with a key rotated by 16 positions due to the extra 16 rounds. Then a system of boolean equations are solved for regular 64 rounds of the encryption and 64 rounds with shifted key by 16 positions. The key can be recovered algebraically without guessing any key bits it can be done in overall complexity of 2^{64} CPU clocks.

4.4.3 Meet-in-the-Middle slide attack

In this attack, the key is recovered when there are two states separated by 32 rounds at maximum. Due to the cipher structure, the transition between these two states is dependent only on particular key bits, which can be recovered in linear time [7].

Figure 4.4 shows a description of the attack. The key is divided into 4 equal parts

each one is 16 bits, at the start first 16 key bits are guessed and used to partially encrypt plaintexts and partially decrypt ciphertexts. The values of the partially encrypted plaintext, partially decrypted ciphertext in the middle, and part from the key are saved on a hash table. During the iterations, the values are compared to the hash table to find a collision. Once the collision is found, then P_i and P_j are a slide pair.

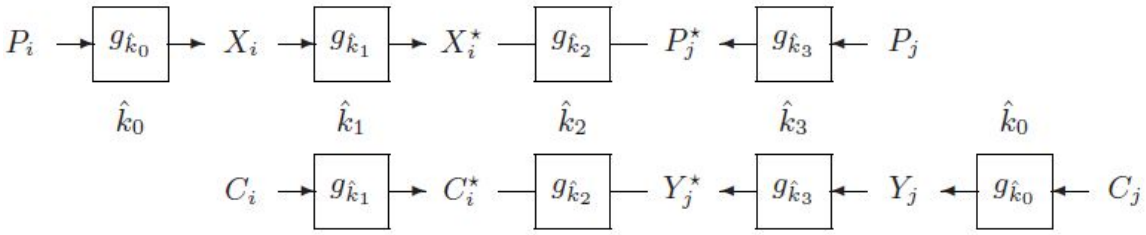

Figure 4.4: Meet in the middle slide attack [7]

The complexity of this attack is equivalent to 2^{45} KeeLoq encryptions. A generalized form of this attack has even slightly reduced complexity, equivalent to $2^{44.5}$ KeeLoq encryptions.

4.5 Our contribution

We present two different attacks against the Keeloq cipher. The first one is a chosen ciphertext attack against the decryption of the keeloq cipher using keys related by rotation. The attack requires 66 chosen ciphertexts decrypted under 34 related keys and negligible time complexity.

We also discuss the tradeoff between the time complexity and the used number of ciphertexts and related keys. In the second attack, we assume we have access to the encryption module and can control its clock, so we force the system to operate more than 528 clock cycles and show a method by which we recover the secret key.

4.6 Our attack

In this attack, we assume that we have access to the decryption circuit of a Keeloq cipher. We use chosen ciphertext attack with related keys to retrieve the original key. The attack exploits the simplicity of mixing the key into the state during decryption. Denote full decryption of a Keeloq cipher C with the key K by $D_K(C)$, and decryption through a single round with the subkey bit k by $f_k(C)$.

Consider a pair of related keys (K,K') such that $K' = (K <<< 1)$, where $K <<< l$ denotes the left rotation of the key K by l bits. If for a pair of ciphertexts (C, C') we have: $C' = f_{k15}(C)$, where k_{15} is the 15^{th} bit of K, then $D_{K'(C')} = f_{k63}(D_K(C))$ It means that the decryption of C' under the key K' is equal to the decryption of C under K shifted by one round (see Figure 4.5).

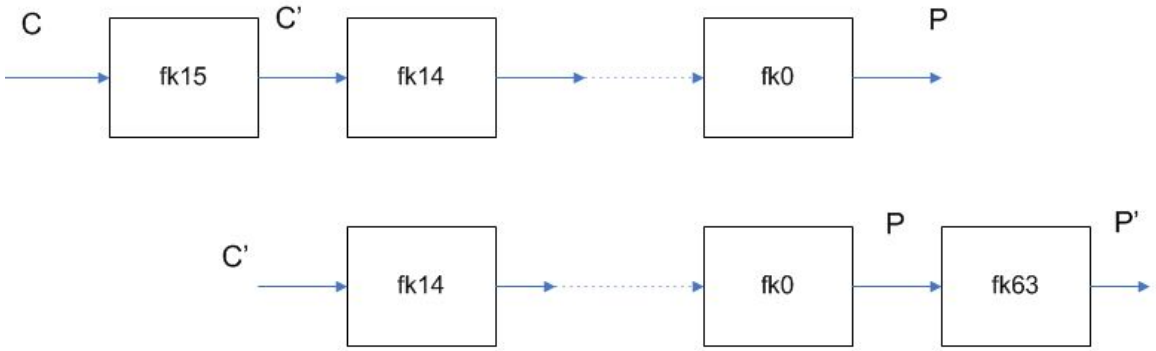

Figure 4.5: A related-key attack.

Using this fact, we can retrieve two bits of the secrete key K in the first stage of the attack. The next step is to consider a ciphertext C. Since C' is one more round further than C, then $C' = f_{k15}(C)$. Therefore, C' is one of two possibilities C'_1 =$(C << 1)||1$ or C'_2 =$(C << 1)||0$, where $C << 1$ denotes shifting the ciphertext C by one bit to the left. We request the decryption of C under the key K, and the decryption of C'_1 and C'_2

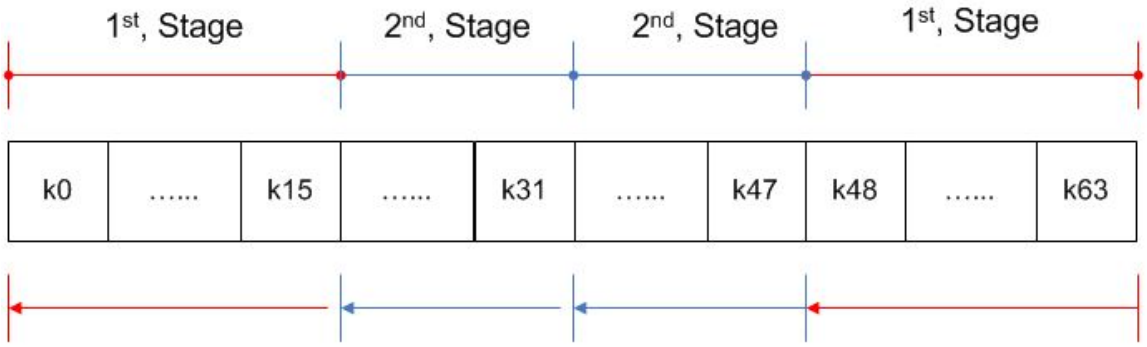

Figure 4.6: The two stages of the related-key attack.

under the related key K'. Then we check which of the following two equations is valid: $D_{K'}(C'_1)$=$f_{k63}(D_K(C))$ or $D_{K'}(C'_2) = f_{k63}(D_K(C))$. Note that $f_{k63}(D_K(C))$ has 31 bits in common with $D_K(C)$. So, we can easily decide which of C'_1, and C'_2 satisfy the relation.

Since both C and C' are known, we can get k_{15} using the relation $C' = f_{k15}(C)$. Also, the decryption of both C and C' are known. Consequently, we can get k_{63} from the relation $D_{K'}(C') = f_{k63}(D_K(C))$.

We repeat the same procedure with the pair of related keys (K', K'' <<< 1) and the ciphertext C' from the previous step. Again, we ask for the decryption of two ciphertexts C''_1 =$(C' << 1)||1$ and C''_2 =$(C' << 1)||0$ under the key K'', as a result, we get the two key bits k'_{15} and k'_{63}, which are equal to k_{14} and k_{62}.

This procedure can be repeated 16 times to get the key bits $k_0, k_1, \cdots, k_{15}$ and $k_{48}, k_{49}, \ldots, k_{63}$. That is the end of the first stage of the attack (see Figure 4.6).

Proceeding beyond than 16 times is not efficient because we get some of the previously recovered key bits. Accordingly, in the second stage of the attack, we repeat the same procedure with another 16 related keys of the form (K <<< 32), (K <<< 33),$\cdots$,(K <<< 47) to retrieve the remaining 32 key bits.

4.6.1 Attack complexity

In the first stage of the attack, we use three chosen ciphertexts and two related keys to get the first two key bits. Then, we use two chosen ciphertexts and one related key for each of the remaining 15 steps to get 30 key bits. And the same for the second stage of the attack.

The attack then requires a total of 66 chosen ciphertexts decrypted under 34 related keys and a negligible time complexity.

We may reduce the number of ciphertexts and related keys for a higher time complexity by switching to an exhaustive key search after recovering some key bits.

4.7 Another attack

We assume that we have access to the encryption hardware, and we can force the system to run more than 528 times by controlling the clock or by moving the key register to different hardware that can be controlled. First, we run the encryption system 528 times which is the normal number of rounds to get the ciphertext C, then we reset the states and force the system to run 529 times to get C'.

We know that C' is one more round than C. So, $C' = f_{k16}(C)$ and we have both C and C' so we can get k_{16}. We repeat the same procedure and force the system to run 530 times and get $C'' = f_{k17}(C')$ then we recover another key bit.

We can repeat this procedure 64 times to recover the original key with negligible time complexity. Note that this attack is theoretical and has not been demonstrated in a lab environment.

Chapter 5

Final discussion and conclusion

5.1 Conclusion

In the first part of this thesis, we investigated the security of some widely used 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.

In the second part of the thesis, we have presented a chosen cipher text attack against the decryption of the Keeloq cipher using keys related by rotation. We discussed the two

stages of the attack and how to reduce the number of ciphertexts and related keys against time complexity. We also presented another attack on the encryption algorithm by forcing the system to operate more than 528 clock cycles to recover the secret key.

From our work and the previous attacks discussed in this paper, we conclude that the Keeloq block cipher is vulnerable to attacks that can recover its secret key.

5.2 Future work

Although we investigated the security of three brands of garage door openers in our work, It seems interesting and promising to investigate more brands. Moreover, we could extend the security investigation to include some brands of keyless vehicles.

Since KeeLoq is a lightweight block cipher that is deployed in mobile devices, e.g., key fobs and immobilizers, another type of security investigation that could be carried out is fault attacks against the key and plaintext/ciphertext registers. Such attacks can then be combined with differential analysis to recover the secret key efficiently. Also, the use of Physical unclonable functions (PUFs) could be investigated as such functions provide a unique finger print to the transmitter that can not be replicated by any impersonating entity.

References

[1] J. D. McKinney, I. S. Lin, and A. M. Weiner, "Shaping the power spectrum of ultra-wideband radio-frequency signals," *IEEE Transactions on Microwave Theory and Techniques*, vol. 54, no. 12, pp. 4247–4255, 2006.

[2] G. C. Clark Jr and J. B. Cain, *Error-correction coding for digital communications*. Springer Science & Business Media, 2013.

[3] 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.

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

[5] W. b. admin and W. by, "Steve mould hacks into his car with a hackrf," Dec 2020. [Online]. Available: https://www.rtl-sdr.com/tag/rolljam/

[6] D. Li, W. Xiao, Z. You, and Y. Wang, "The analysis and improvement of keeloq algorithm."

[7] 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.

[8] T. Glocker, T. Mantere, and M. Elmusrati, "A protocol for a secure remote keyless entry system applicable in vehicles using symmetric-key cryptography," in *2017 8th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2017, pp. 310–315.

[9] A. Bogdanov, *Analysis and design of block cipher constructions*. Citeseer, 2010.

[10] S. Kamkar, "Opensesame," Jun 2015. [Online]. Available: https://samy.pl/opensesame/

[11] A. Ghanem and R. AlTawy, "Garage door openers: A rolling code protocol case study," in *2022 19th Annual International Conference on Privacy, Security & Trust (PST)*. IEEE Computer Society, 2022, pp. 1–6.

[12] R. AlTawy, "Cryptanalysis of some aes-based cryptographic primitives," Ph.D. dissertation, Concordia University, 2016.

[13] C. E. Shannon, "Communication theory of secrecy systems," *The Bell system technical journal*, vol. 28, no. 4, pp. 656–715, 1949.

[14] A. J. Menezes, S. A. Vanstone, and P. C. Van Oorschot, "Handbook of applied cryptography (special indian edition)," 2010.

[15] A. Biryukov, D. Khovratovich, and I. Nikolić, "Distinguisher and related-key attack on the full aes-256," in *Annual International Cryptology Conference*. Springer, 2009, pp. 231–249.

[16] L. R. Knudsen and V. Rijmen, "Known-key distinguishers for some block ciphers," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2007, pp. 315–324.

[17] L. R. Knudsen, "Contemporary block ciphers," in *School organized by the European Educational Forum*. Springer, 1998, pp. 105–126.

[18] E. Biham, "New types of cryptanalytic attacks using related keys," *Journal of Cryptology*, vol. 7, no. 4, pp. 229–246, 1994.

[19] J. Scarpati, "What is radio frequency (rf, rf)? - definition from whatis.com," Feb 2021. [Online]. Available: https://www.techtarget.com/searchnetworking/definition/radio-frequency

[20] M. Integrated, "I'm ook. you're ook?" Apr 2009. [Online]. Available: https://www.maximintegrated.com/en/design/technical-documents/app-notes/4/4439.html

[21] I. Martoyo, P. Setiasabda, H. Y. Kanalebe, H. P. Uranus, and M. Pardede, "Software defined radio for education: Spectrum analyzer, fm receiver/transmitter and gsm sniffer with hackrf one," in *2018 2nd Borneo International Conference on Applied Mathematics and Engineering (BICAME)*, 2018, pp. 188–192.

[22] 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.

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

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

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

[26] 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-sha

[27] 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

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

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

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

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

[32] N. T. C. V. Bard and A. Bogdanov, "Periodic ciphers with small blocks and cryptanalysis of keeloq," *Tatra Mt. Math. Publ*, vol. 41, pp. 167–188, 2008.

[33] N. T. Courtois, G. V. Bard, and D. Wagner, "Algebraic and slide attacks on keeloq," in *International Workshop on Fast Software Encryption*. Springer, 2008, pp. 97–115.

[34] 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.

[35] G. Agosta, A. Barenghi, and G. Pelosi, "High speed cipher cracking: the case of keeloq on cuda," 2012.

[36] 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.

[37] H. Microchip, "Hcs301 keeloq code hopping encoder data sheet - microchip technology," Oct 2001. [Online]. Available: https://ww1.microchip.com/downloads/en/devicedoc/21143b.pdf