Grand Theft House: RF Lock Pick Tool to Unlock Smart Door Lock

Seungjoon Lee, Kwonyoup Kim

Agenda

Background

- Wireless Door Lock System
- Hardware Teardown
- Door Lock RF Signal Basic

Encryption & Authentication

- Door Lock RF Packet Encryption
- Key Generation
- Packet Confusing
- TxID Matching and Authentication

The Art of Lockpicking

- Sniffing and Unlock Door Locks
- Resynchronization
- Force Synchronization
- Lock Picking Time Analysis

Rolling Code and Replay

- Rolling code nutshell
- Roll Jam/ Roll Back Attack
- Loop Back Attack
- Evaluation

Gadget Preparation

- Direct Mode and Synchronization
- Proprietary Baseband Encoding
- Build Receiver and Transmitter

Background

"Replay Attack" Aired in 2019

, DDL RF Module- 4 wires (Power 2 /open 2)

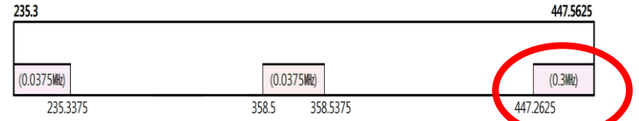
Model		
Transmission Method		Camera- 4 wires, CCTV- 2 wire Interphone- 4 wires , Sensor- 2
Power Source		100V-240V~, 50/60Hz
Consumption		Stand-by : 3.5 W Max : 15 W
Intercom system		HANDS FREE type (Voice swit
Monitor		7″ FULL TOUCH LCD
Video subscription time		Individual door camera monitor Call connection / Calling: 60 ±5
Wiring Distance	UTP CAT 5e. Standard (ø 0.5)	Camera / CCTV / Interphone <
		28 meters (MAX)
		Resistance under 10Ω per 100
Operating Temp.		0 ~ +40℃ (32°F ~ 104°F)

Registration Status (Korea National Radio Research Agency)

Application	'17	'18	19	tot.
447MHz Module		2	2	4
alarm detector	1	6	-	7
Digital Doorlock	**23**	**16**	**18**	**57**
Guidance for disabilities	10	8	14	32
Thermal Detector	5	3	-	8

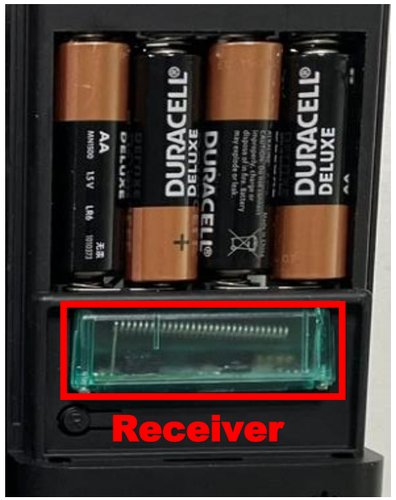
Korea Table of Frequency Allocations(Low-power wireless device for security & safety systems)

Application	Band	Power
Guidance equipment for the visually impaired	235.300MHz (Fixed)	< 10m W
	358.500MHz (Mobile)	
Transportation support system for disabilities	235.3125MHz	< 100m W
	…	
Security system, Anti-theft system, fire alarms, etc.	447.2625 MHz ~447.5625 MHz	< 10m W



Wireless Door Lock System

- It can be used in a variety of settings, including homes, offices, and other commercial or industrial.



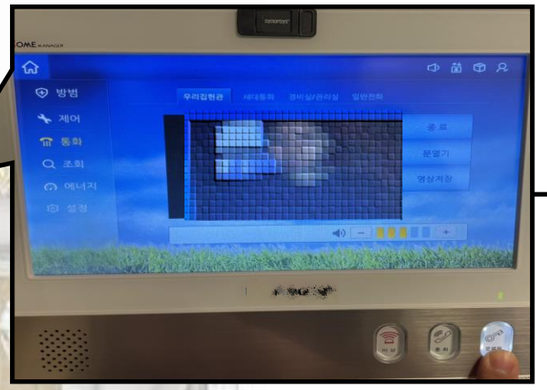
Doorlock Backside

1) Video Doorbell

3) Key Fob

2) NTSC/PAL & Serial

Wallpad



3) Wallpad Switch

4) RF Transmission

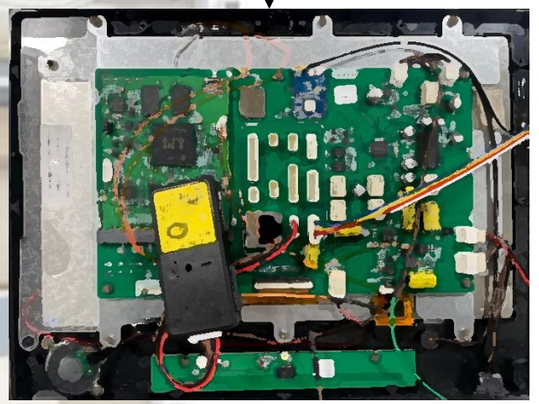
Wi-Fi

Door lock

Enterance

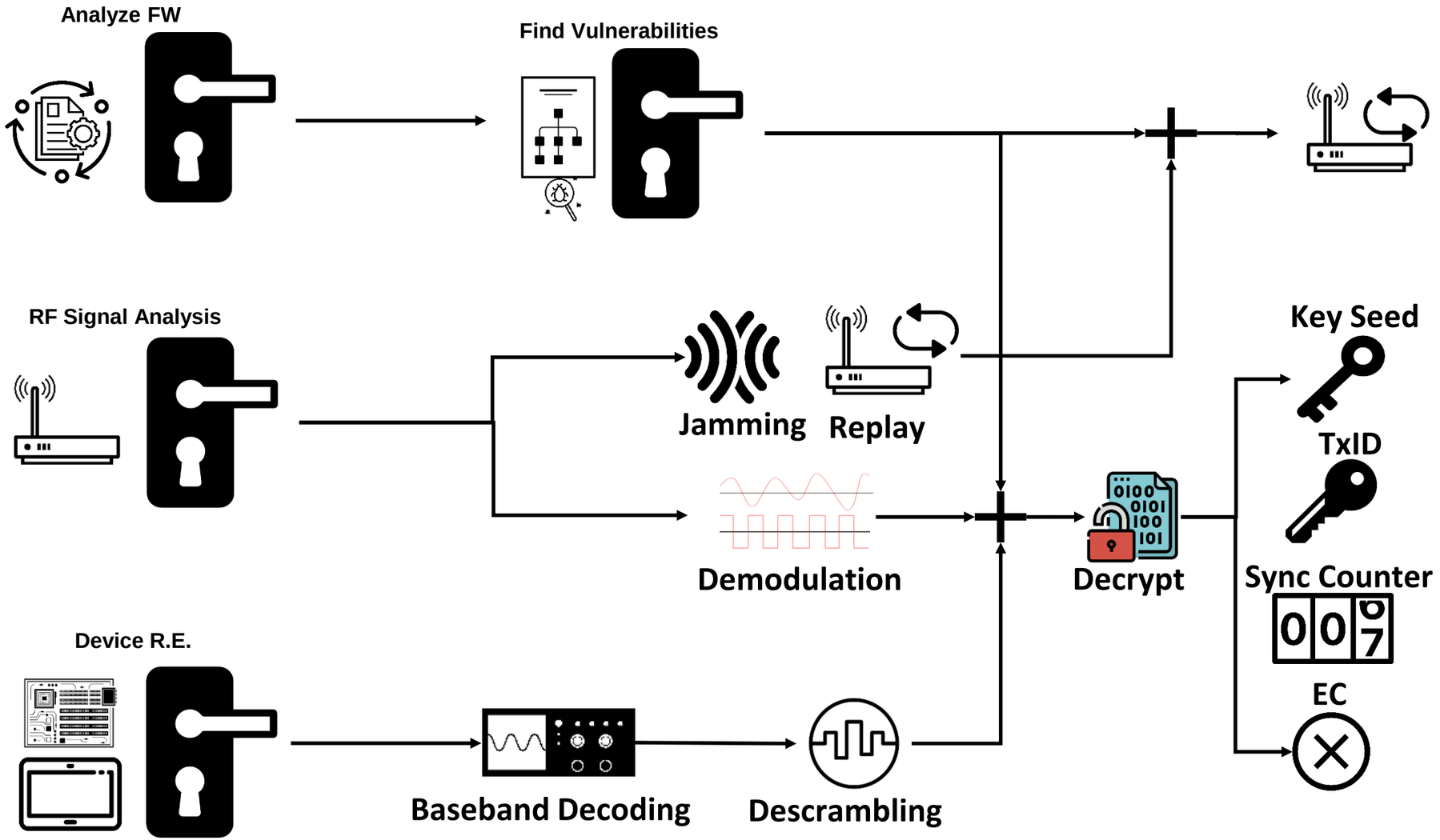
Wireless Router

3) Smartphone APP



Wallpad Backside

Summary



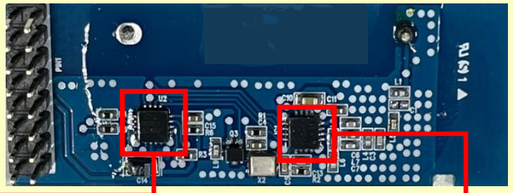
- Classic Replay Attack
- Rollback Attack
- Loop Playback Attack

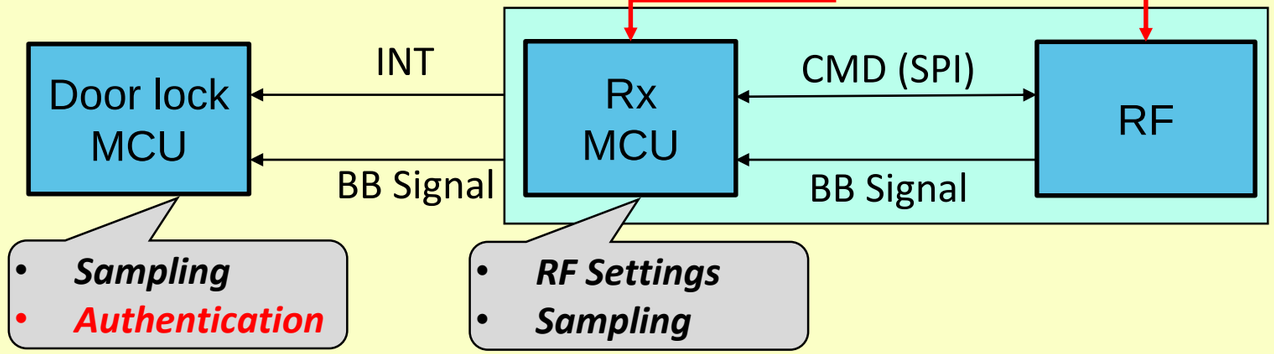
- Sniffing and Unlock Attack
- Lock Picking Attack

Types of Door Lock RX

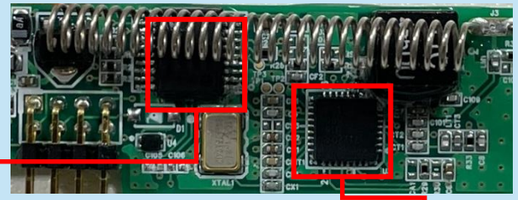
- Door Lock Mainboard MCU
  - Usually use one of “8bit STM” or “16Bit RL78” or “32Bit ARM”
  - usage : Ten-key pad, E-Mortise Control, Authentication
- RX MCU
  - Usually use one of “8bit 8051” or “8bit PIC” or “8bit STM” or “16bit PIC”
- RX MCU
  - Usually use transceiver, TH71120 /CMT 2219 / CMT 2300
  - Transceiver IC used but Simplex Communication

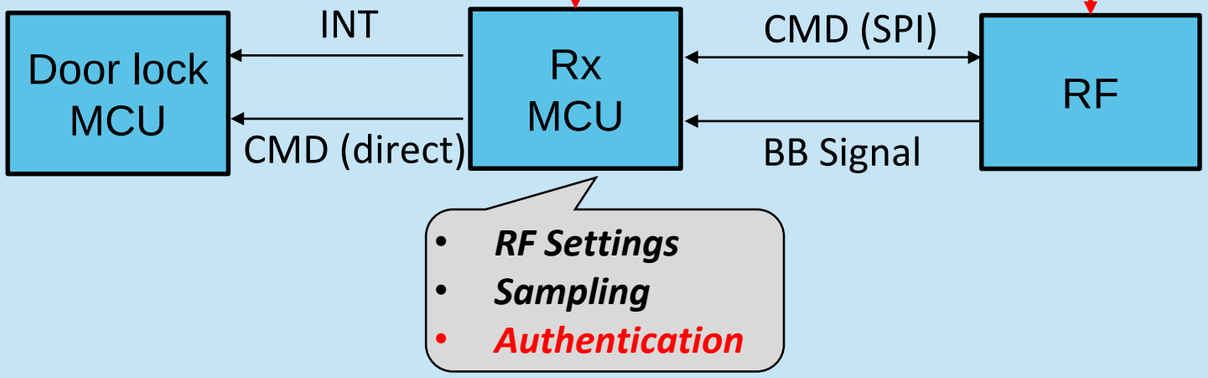
Mainboard-based Auth.
(RX MCU + RF IC Dongle)



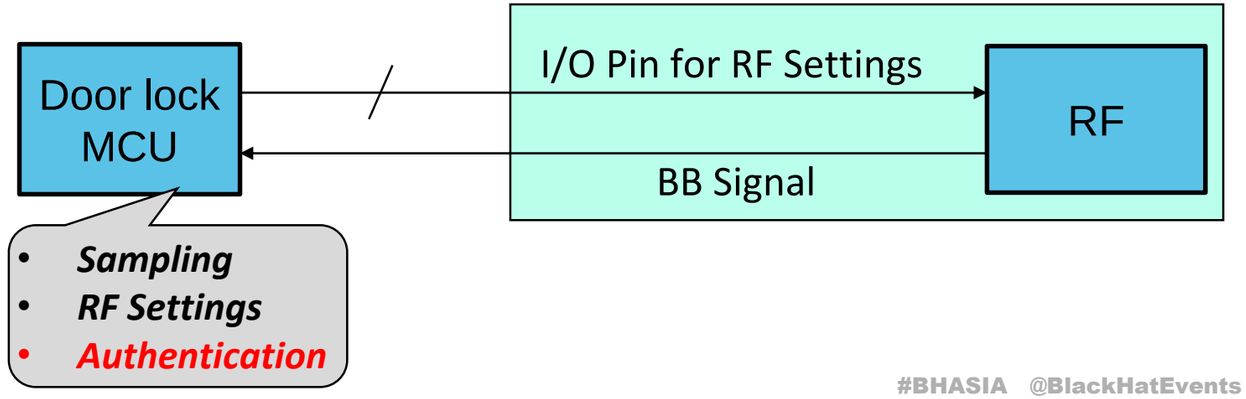


Receiver-based Auth.
(RX MCU + RF IC Dongle)



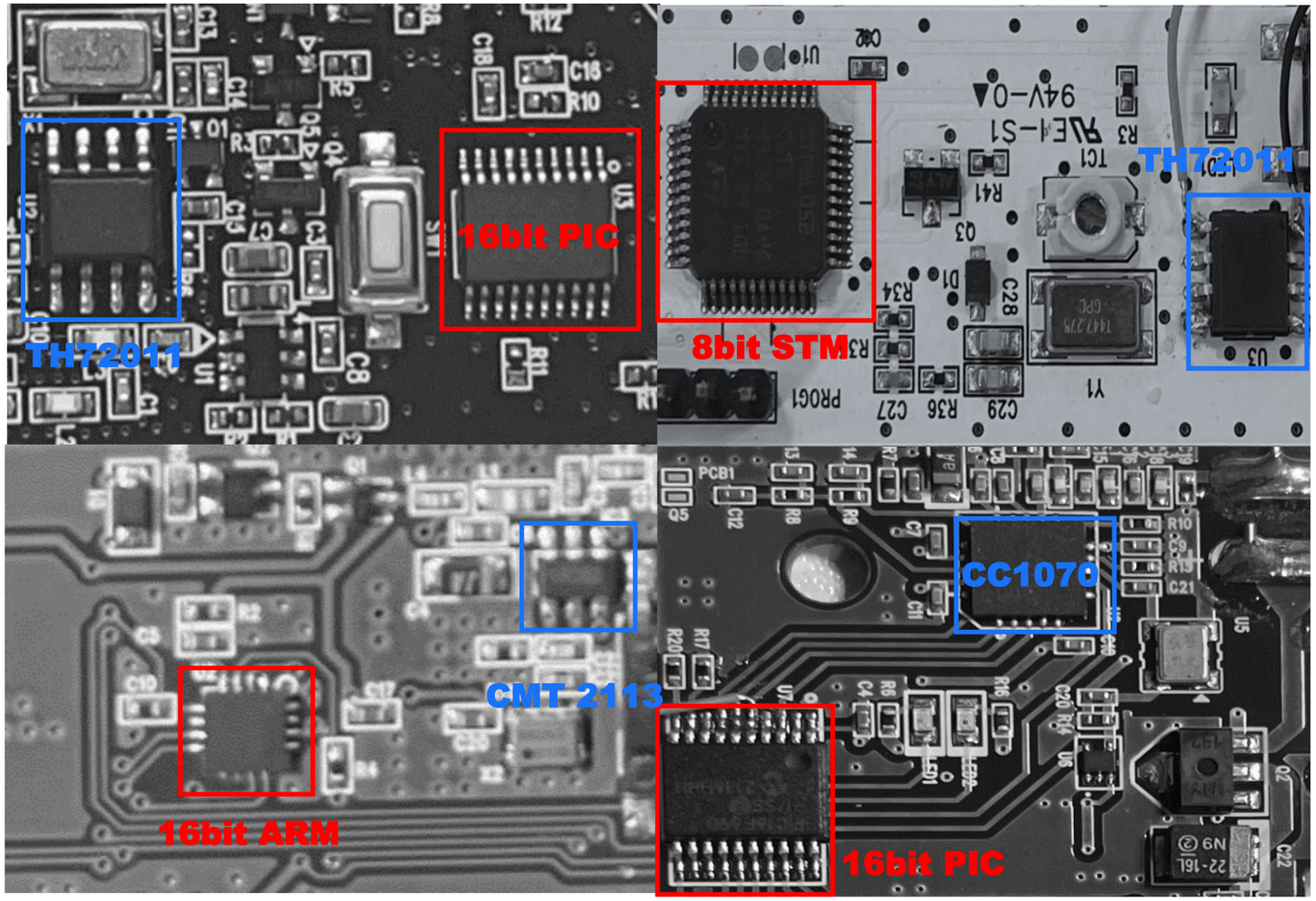


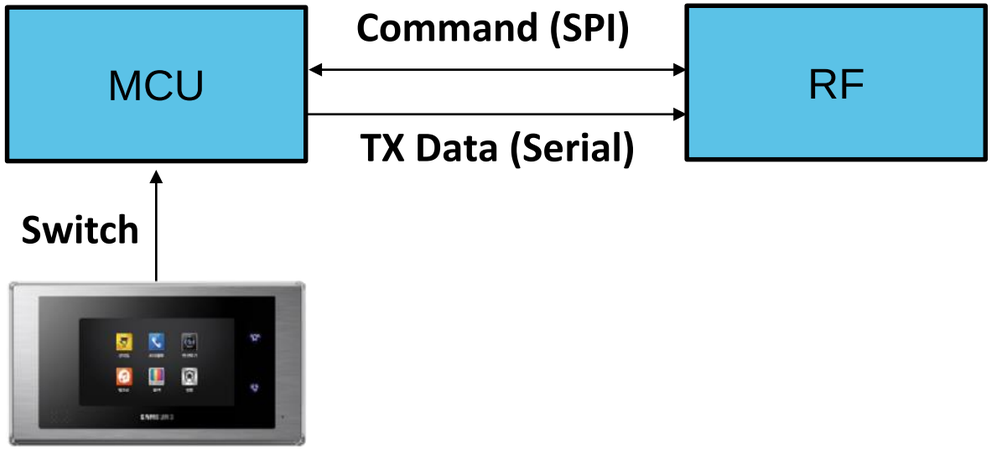
Mainboard-based Auth.
(RF IC Only Dongle)



Types of Door Lock TX

- Simplex Communication





Feature of Transmitter HW

- **MCU**
  - 16Bit ARM /16bit PIC /8bit PIC /8bit STM
  - Digital Encoding
  - Message Encryption
- **RF Chip**
  - Tx Only, FSK Modulation
  - Low-cost(<$3) Discontinuous Phase Type (CC1070 /TH72011 /CMT 21113)

Understanding Rolling Code and Variant Replay Attack

Concise summary and Applied to Door Lock

Principles of Secure Rolling Code

To ensure Secure Rolling Code transmission

1. No transmission is ever repeated

   - Each transmitted message should have **different contents**

   - Receiver should **ignore messages** that have already been sent

   - Keep track of the last used code **and never accept previously used counter**

2. The packet contents are virtually impossible to predict, even if previous messages are known

   - Protect the confidentiality of the rolling code (Encryption Algorithm)

     ❑ "TxID" and "rolling counter" are the information that needs to be kept confidential

     ❑ it can only be read by the intended recipient

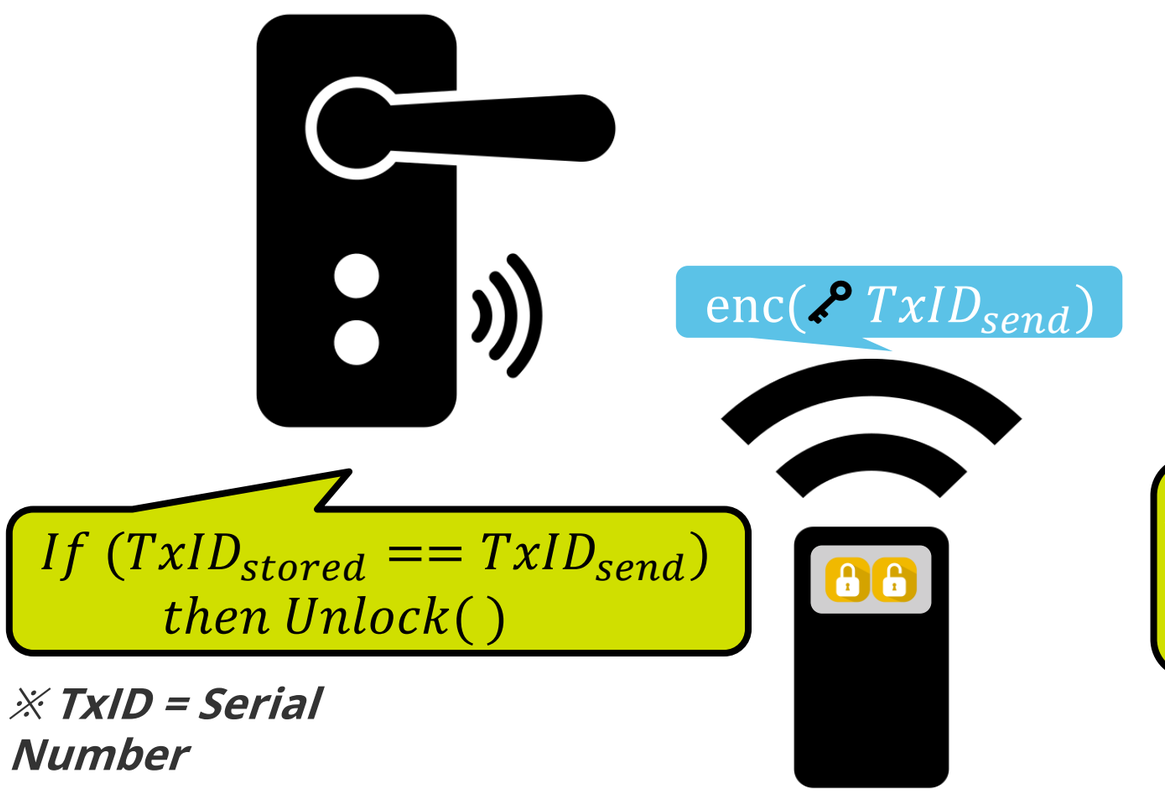
3. Prevent some unauthorized access

   - Filtering mechanism, a unique serial number(TxID) is used to achieve

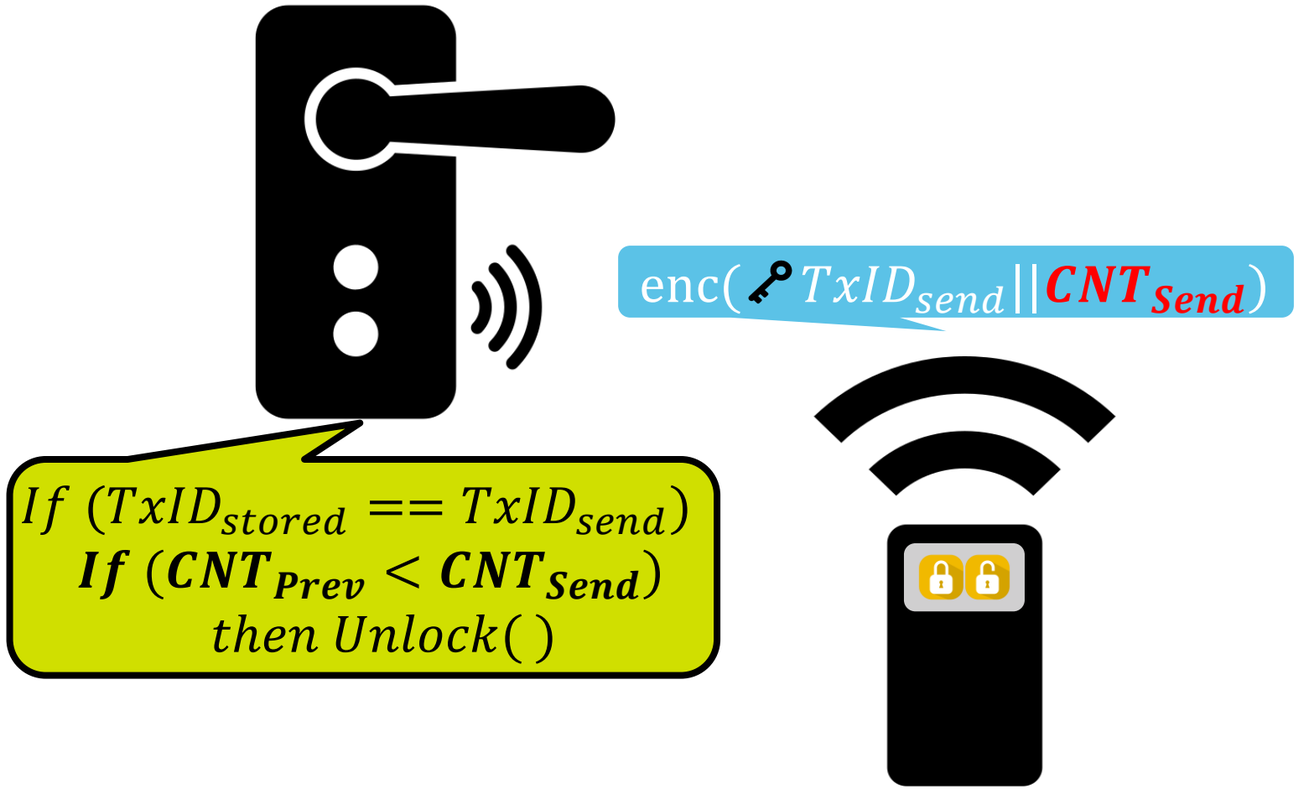
     ❑ TxID should not be guessable and should not appear in a sequential format

Rolling Code in essence

It is used to protect the packet from being replayed



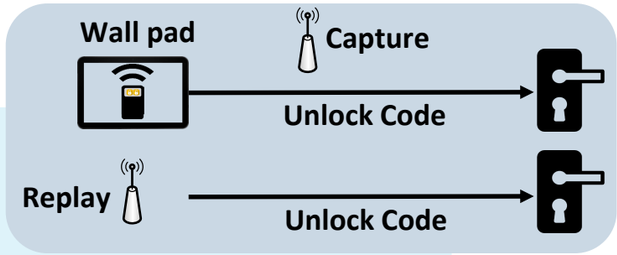
Fixed Code



Rolling Code

Fixed Code on Door locks

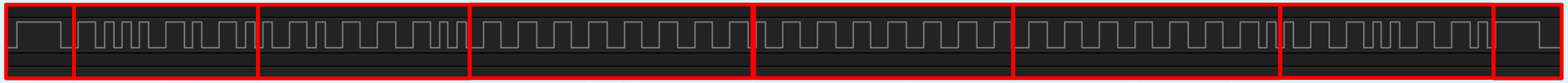
❑ Fixed code is very weak in “classic replay attack”



If pressing the button, produces 17 identical message for robustness

SYNC symbol

End SYM



0x75 | 0xA3 | 0x00 | 0x80 | 0x01(Spacer) | 0x99(CS)

Serial Number(=TxID)

Vendor E

Received 5-Byte in door lock mainboard(volatile)

```
00 00 00 00 00 01 00 02 00 50 00 00 00 02 00 00   .........P......
00 00 00 00 7A 22 C9 86 01 00 EB 00 40 01 08 C5   ....z"......@...
```

Registered 4-Byte TXID in door lock mainboard(non-volatile)

```
7A 22 C9 86 75 00 00 00 00 00 00 00 00 00 00 00   z"..u...........
FF 4F 52 4B FF FF FF FF FF FF 00 00 00 00 00 00   .ORK............
```

```
0x89    PUSHW X
0x97    LD    XL,A
0xA605  LD    A,#0x05
0x42    MUL   X,A
0x1F01  LDW   (0x01,SP),X
0x1C10A0 ADDW X,#0x10a0
0xCDFC91 CALL 0xfc91
0xB124  CP    A,0x24
0x260C  JRNE  0xd69f
```

Rest 3bytes comparison

2nd Byte CMP

```
0x1E01   LDW   X,(0x01,SP)
0x1C10A1 ADDW  X,#0x10a1
0xCDFC91 CALL  0xfc91
0xB125   CP    A,0x25
0x2703   JREQ  0xd6a2
```

3rd Byte CMP

```
0x1E01   LDW   X,(0x01,SP)
0x1C10A2 ADDW  X,#0x10a2
0xCDFC91 CALL  0xfc91
0xB126   CP    A,0x26
0x26F1   JRNE  0xd69f
```

4th Byte CMP

```
0x1E01   LDW   X,(0x01,SP)
0x1C10A3 ADDW  X,#0x10a3
0xCDFC91 CALL  0xfc91
0xB127   CP    A,0x27
0x26E5   JRNE  0xd69f
```

The comparison of the value of the first byte stored

Roll Jam Attack (DEFCON23)

- ❑ Variant of Replay Attack
- ❑ No time stamp, difficult to prevent unused code replay attack
- ❑ RollJam Concept

  Eve capture unused code(signal)

  Replay unused code later

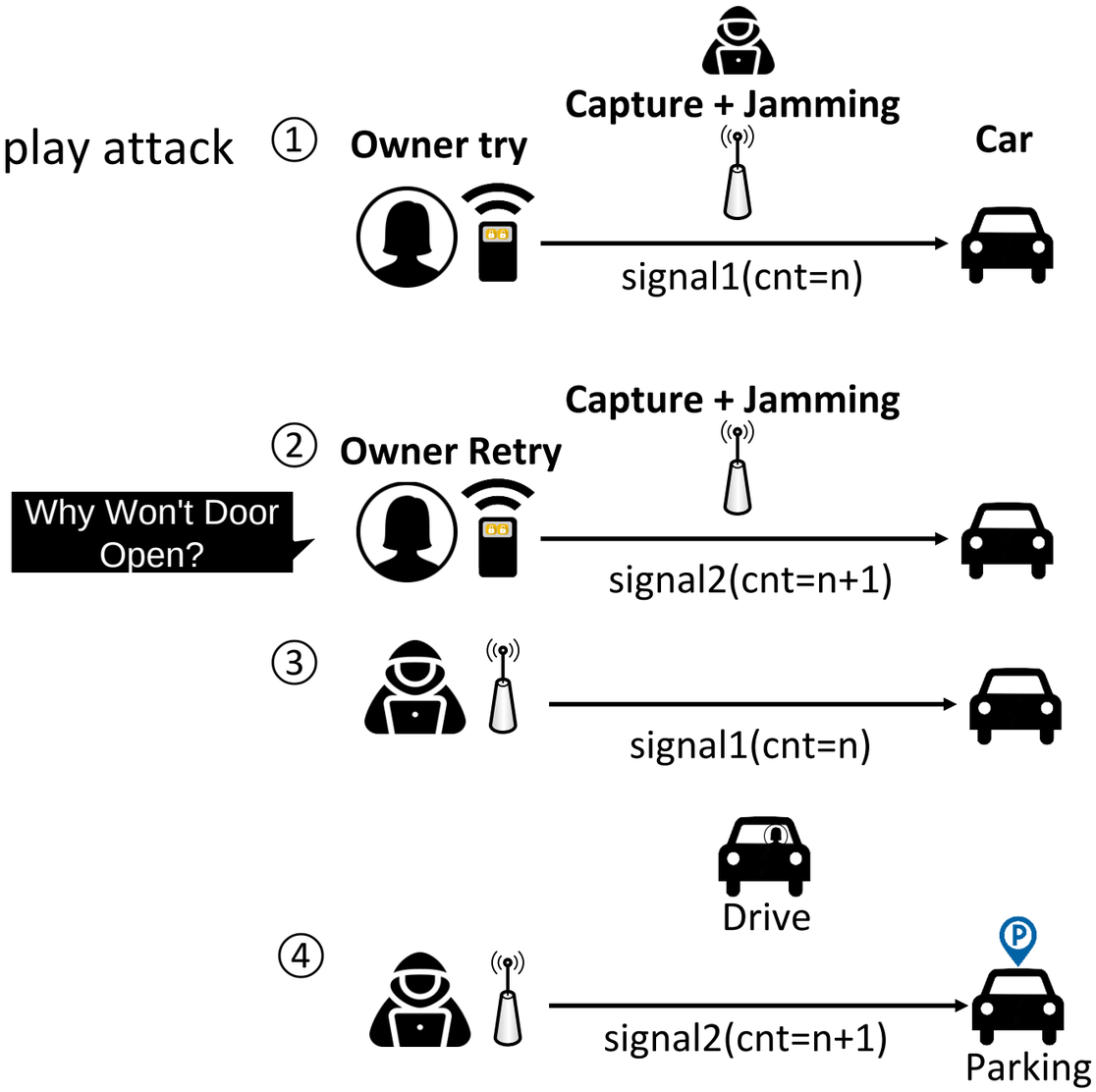
- ❑ Process

  ① Capture signal1 + Jamming

  ② Capture signal2 + Jamming

  ③ Capture Signal1 Replay (for her)

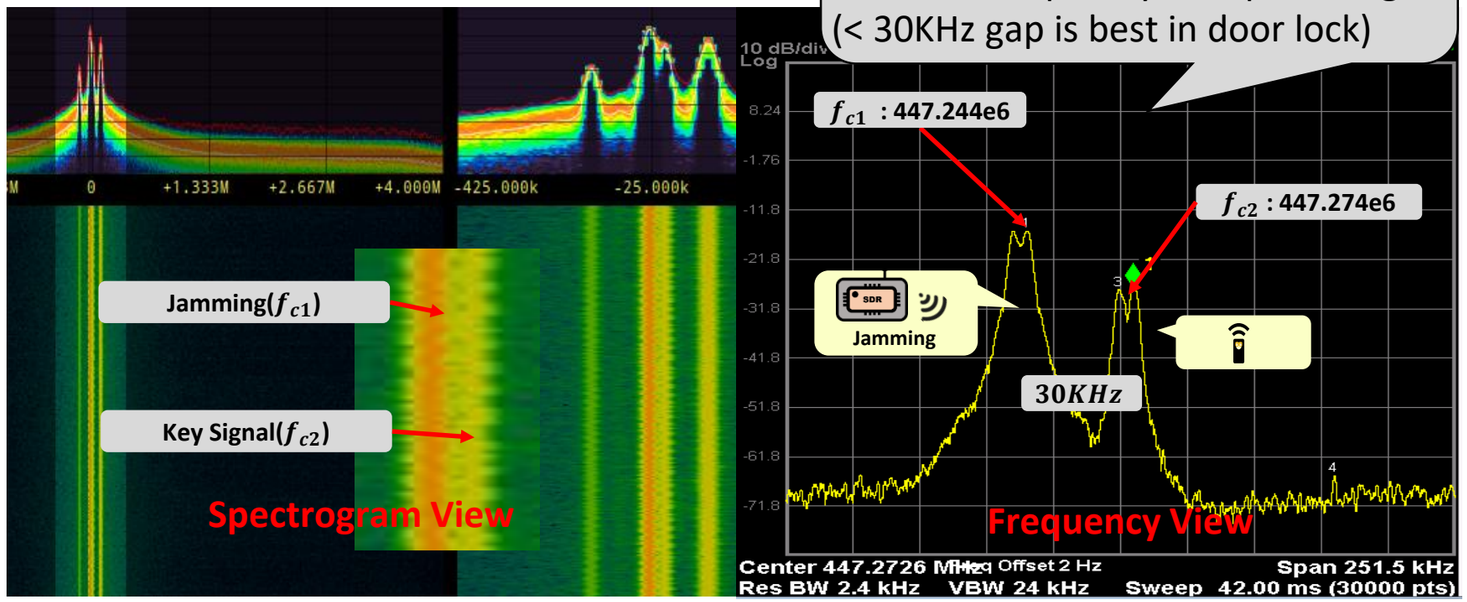
  ④ Capture Signal2 Replay (for carjack)



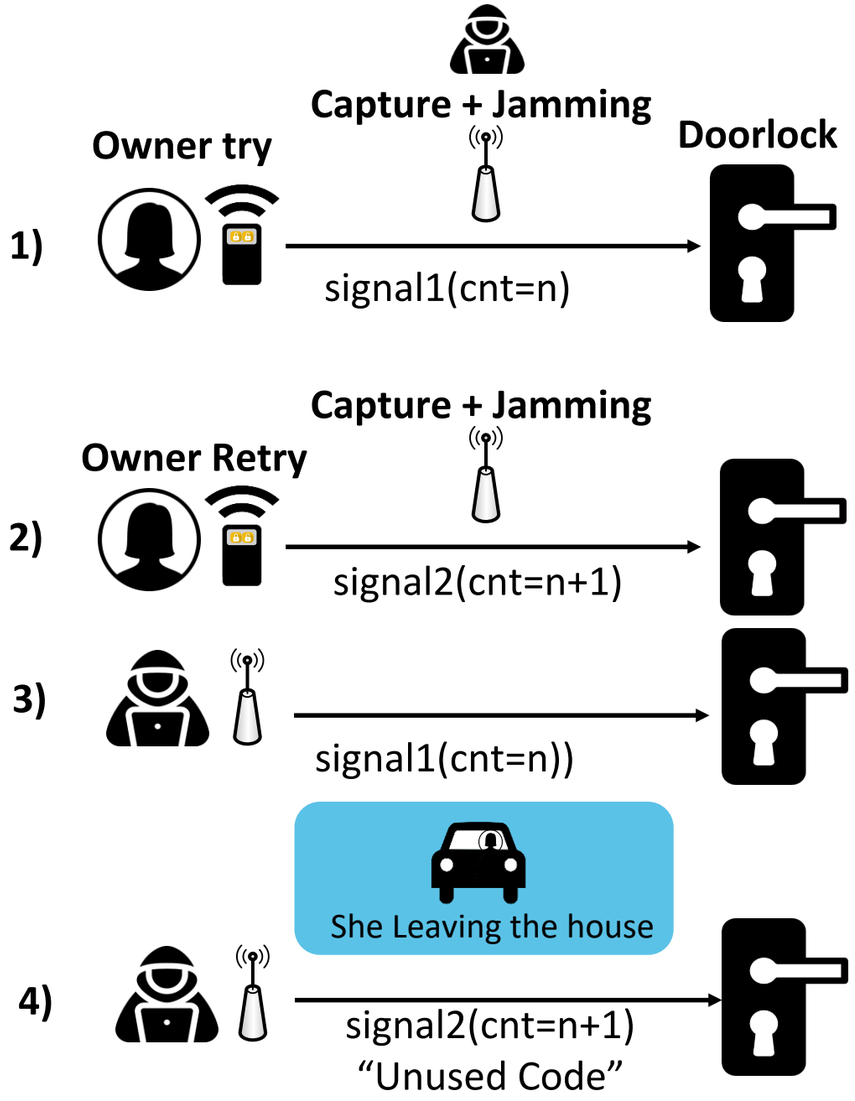
RollJam Attack on Door Lock

- Door Lock RF more vulnerable to jamming
  - narrow bandwidth(80KHz)
- Two drawbacks of RollJam
  - An attacker has to be precise
  - The attack can be launched once

The jamming signal should effectively interfere with the original signal without completely overpowering it. (< 30KHz gap is best in door lock)



Jamming & Capture on Door Lock



RollBack Attack (BHUSA2022)

❑ Rollback Attack Process

Attacker

Victim

1) Send Unlock signal

2) Capture + Jamming

3) Send Unlock signal

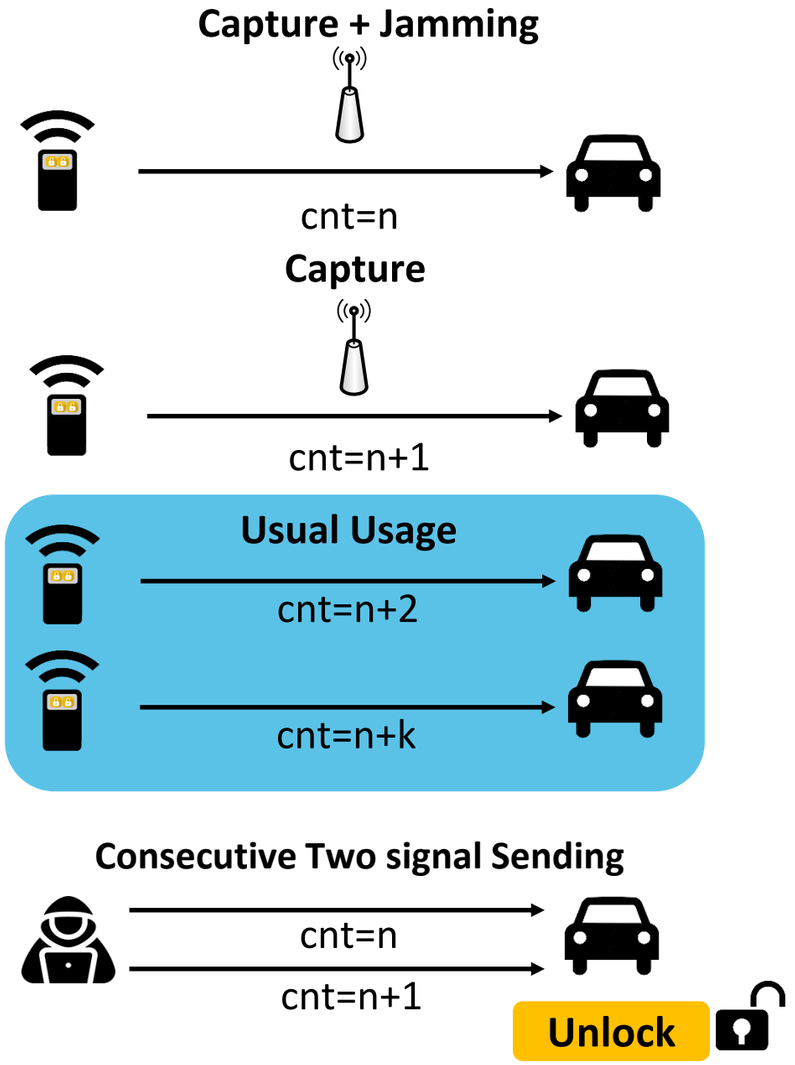
4) Capture

5) Owner uses the key fob as usual(many times)

6) replay the two consecutive signal

❑ Characteristic

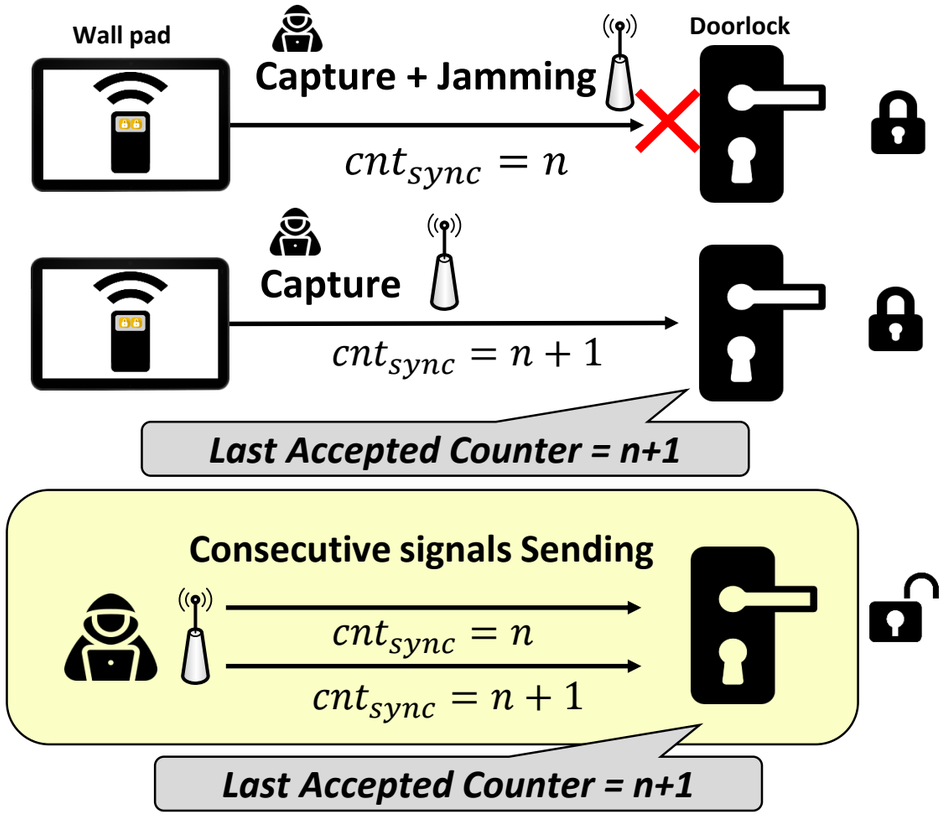
- ❑ At any time in the future
- ❑ As many times as desired



note: vehicle re-synchronize to a previous code

Root cause of RollBack on Door Lock

❑ Issue: If a received message is valid, then the counter value always be stored on memory

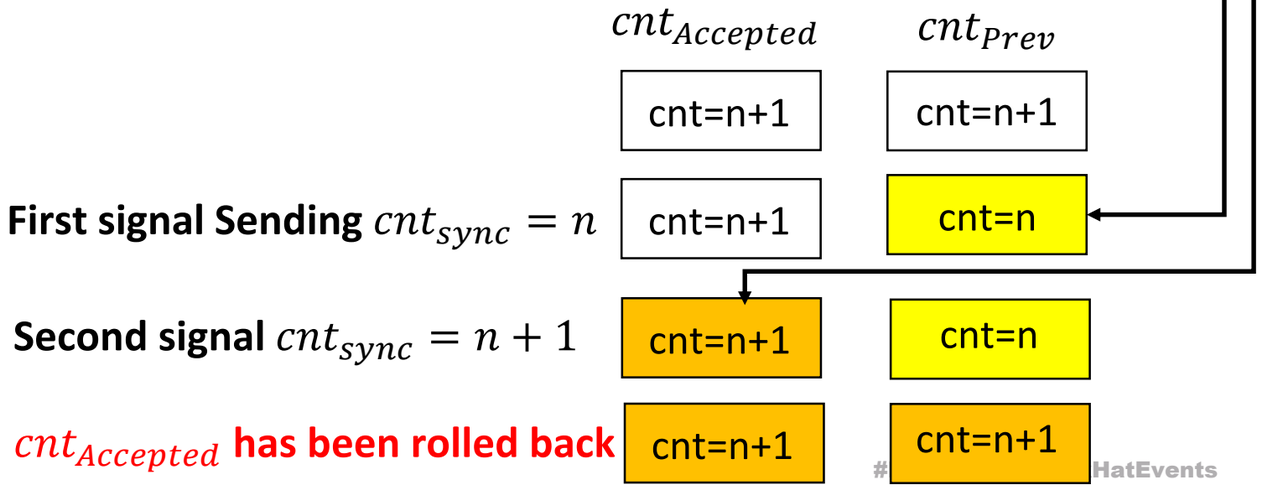



```
001060  FF FF FF 00 00 00 00 00 00 00 00 00 00 00 00 00
001070  00 00 00 0C                                  00
001080  95 A5 28 AD                                  00
001090  B6 B4 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0010A0  2C 08 9E AD                                  9F FF FF
0010B0  FF 00 00 00                                  00 00 00
```

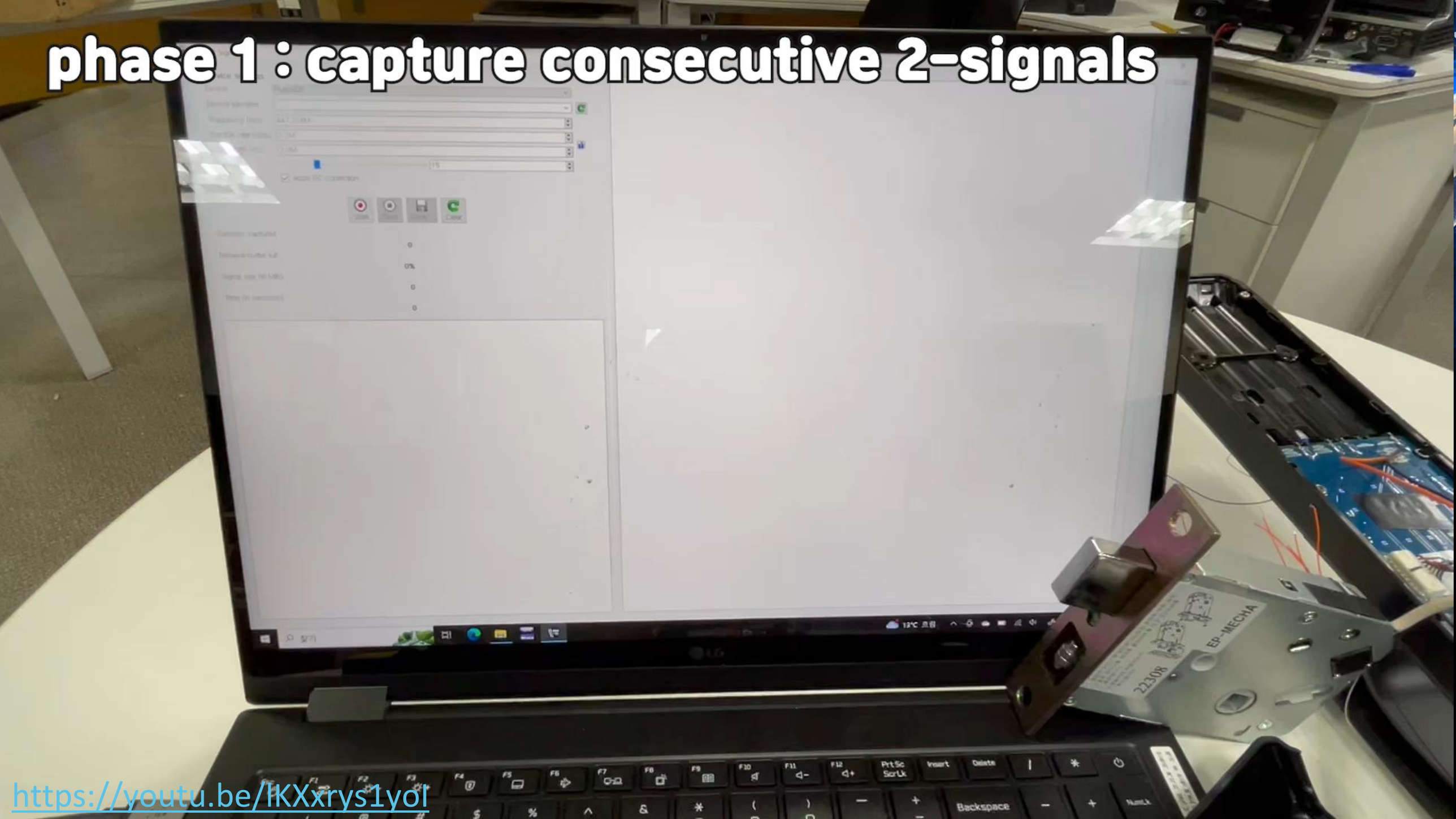
$cnt_{Accepted}$ (Last Accepted Counter)

cnt_{Prev} (Previous Counter)

Input $cnt_{Received}$ // the current counter
Variable cnt_{Prev} // the counter received just before current counter
Variable $cnt_{Accepted}$ // last counter that successfully opened the door
if *Message is valid* ***then***
  if $cnt_{Received} > cnt_{Prev}$ ***then*** // If valid counter
    $cnt_{Accepted} \leftarrow cnt_{received}$ // update Last Accepted Counter
    $cnt_{Prev} \leftarrow cnt_{received}$ // received counter stored
    Doorlock Open()
  else then
    $cnt_{Prev} \leftarrow cnt_{received}$ // received counter stored?? Why?

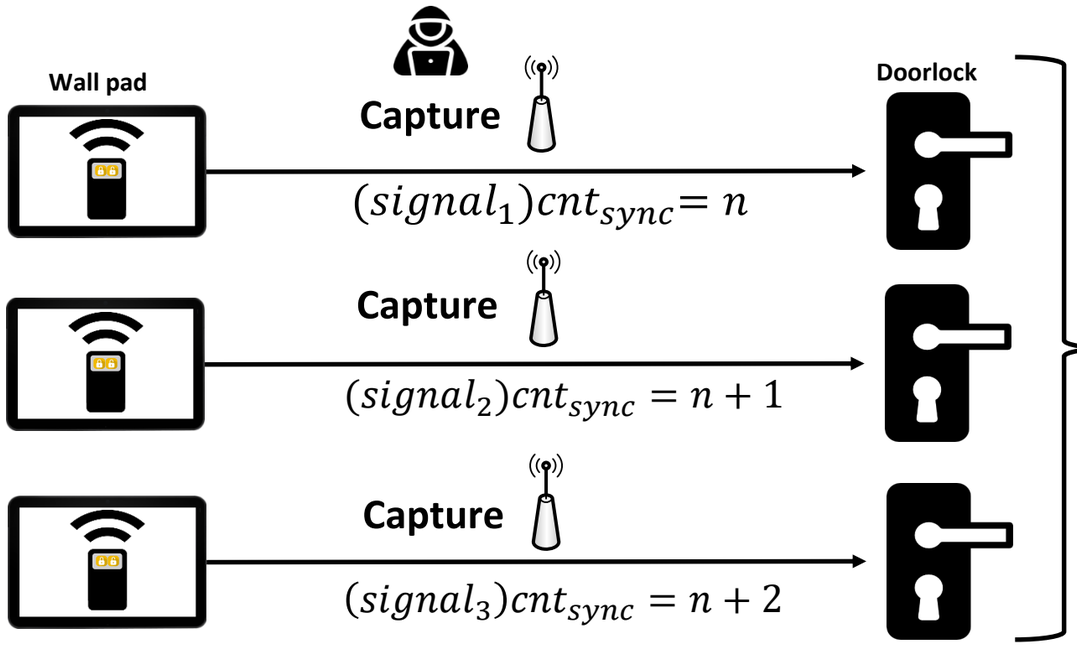


phase 1 : capture consecutive 2-signals



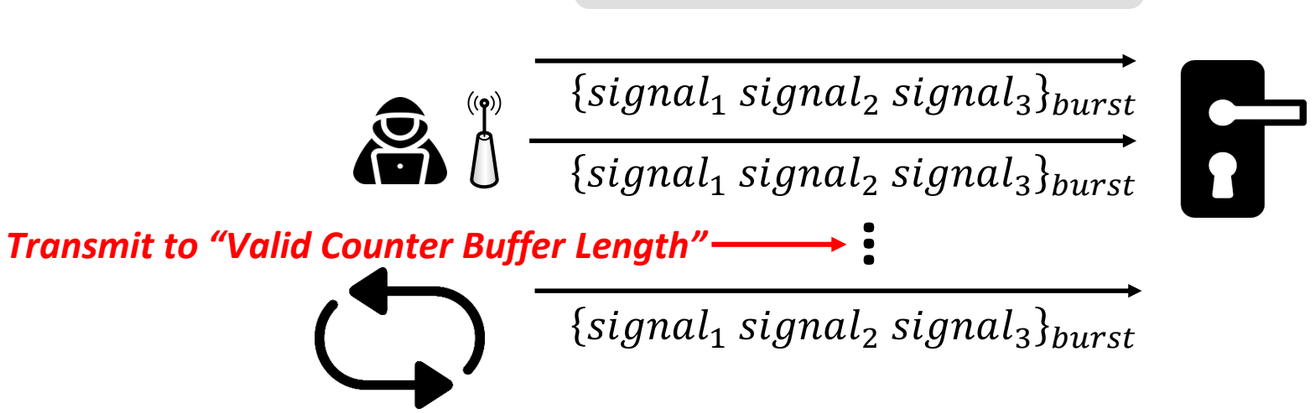
https://youtu.be/IKXxrys1yoI

Loop Playback Attack

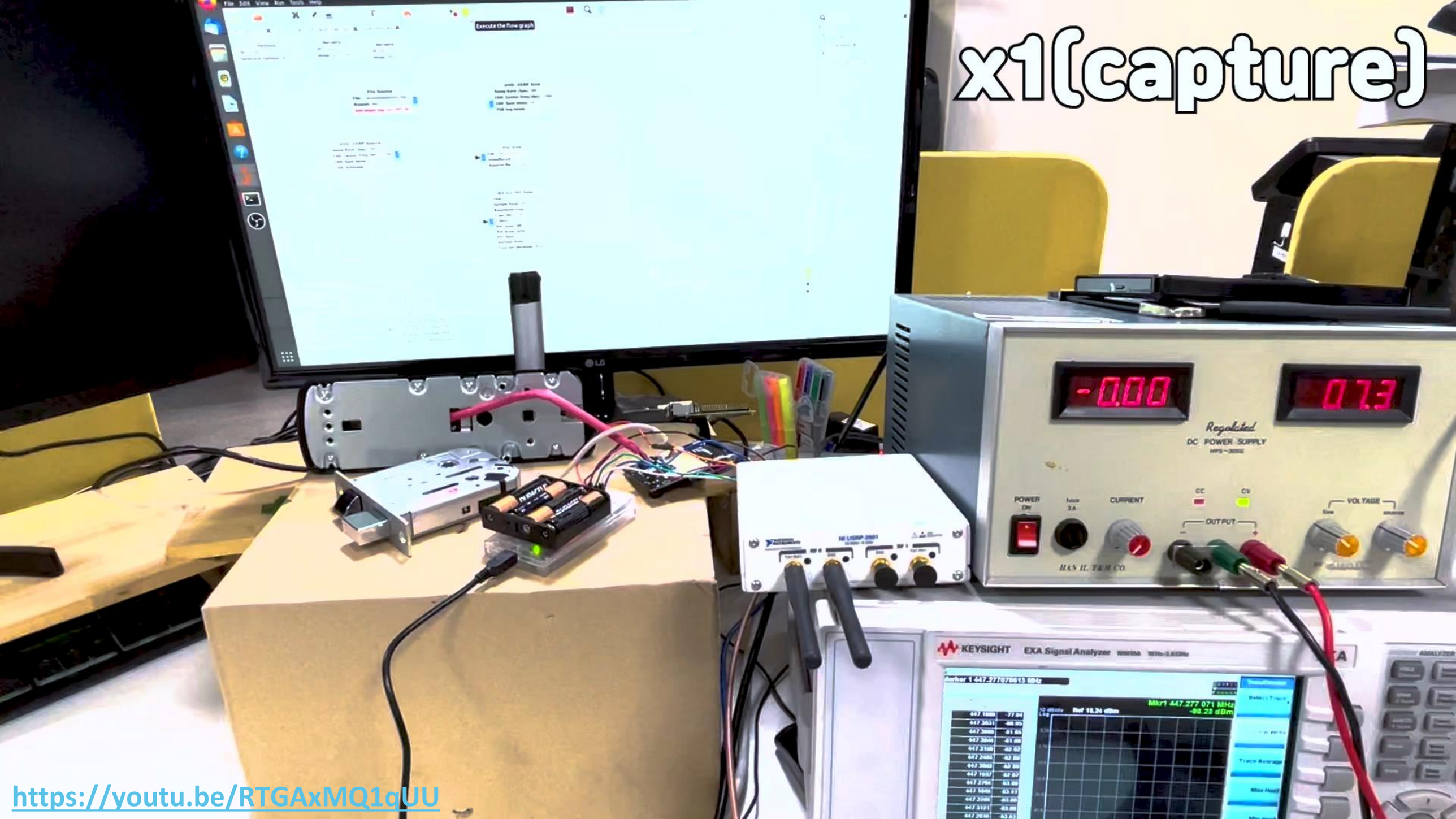


Phase 1: Capture Three Consecutive Signals
The follow signals does not have to be strictly consecutive

Usual Usage of user



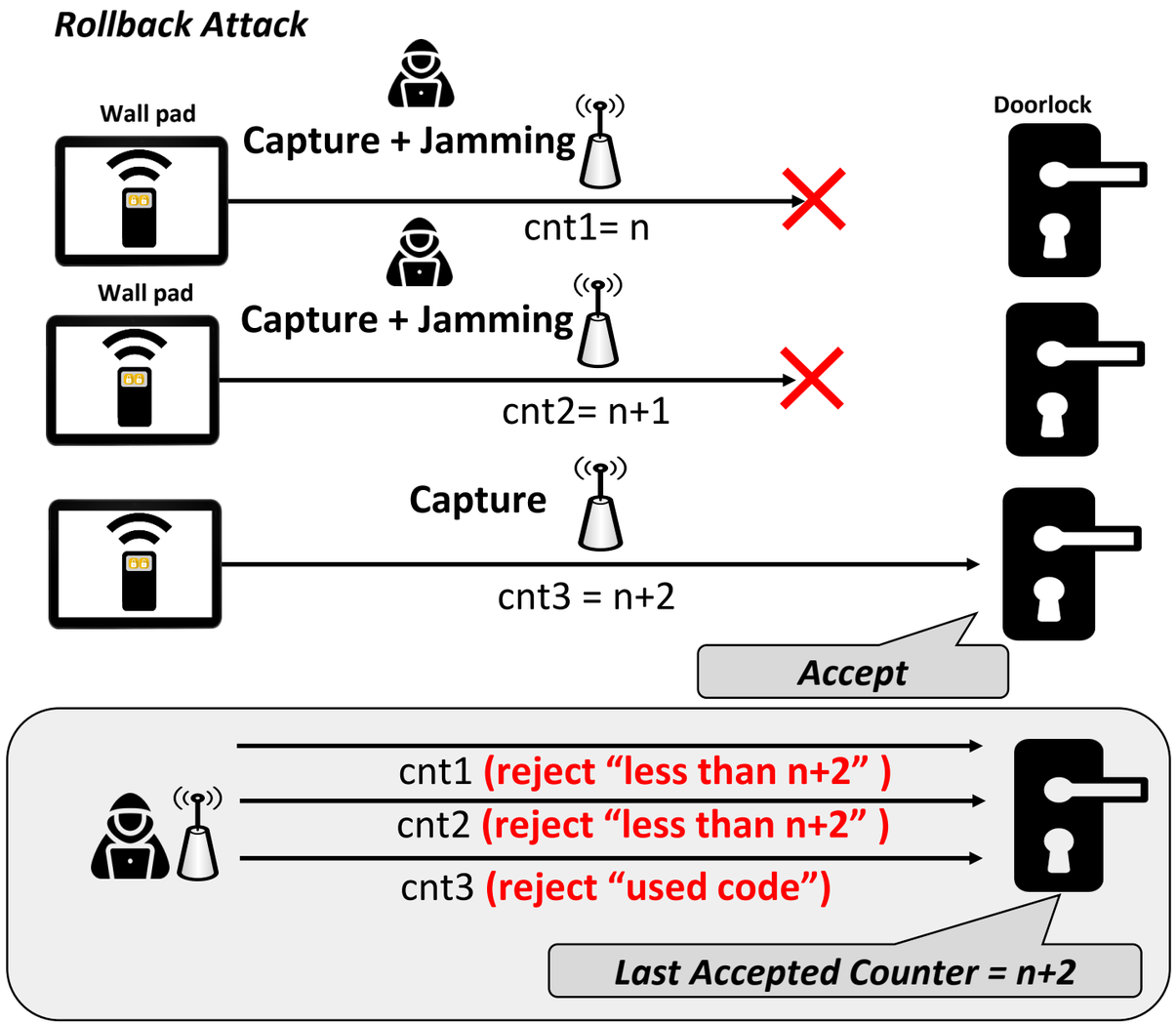
Phase 2: Repeating replay of consecutive three signal

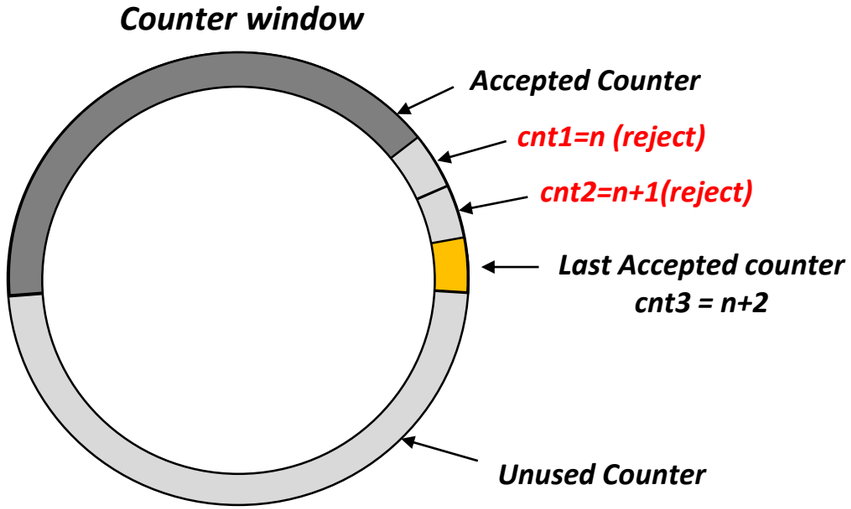


https://youtu.be/RTGAxMQ1qUU

Variant Replay Attacks should not be allowed

- Door locks should keep track of the last used code, and never accept previously used counter





Evaluation : Replay Attack

- Evaluation on a limited set of door locks(2021~2022)

Vendors	Models	Classic Replay	Rolljam	Rollback	Loop Playback	Remarks
A	A-1	√	✗	✗	✗	Consecutive two signals
B	B-1	√	✗	√	✗	Consecutive signal Loop Playboack
	B-2	√	✗	√	✗	
	B-3	√	✗	√	✗	
	B-4	√	✗	√	✗	
C	C-1	√	✗	√	√	-
	C-2	√	✗	√	√	
	C-3	√	✗	√	√	
D	D-1	√	✗	√	√	
	D-2	√	✗	√	√	
E	E-1	✗	-	-	-	one signal
F	F-1	✗				
G	G-1	✗				
H	H-1	✗				
I	I-1	√	✗	√	√	
J	J-1	√	✗	√	√	

✗ Very weak

✗ Weak

√ Moderate

√ Probably safe

Protect the confidentiality of the code

Unveiling the Vulnerabilities in Door Lock RF Encryption

Principles of Secure Rolling Code

To ensure Secure Rolling Code transmission (The three critical properties)

1. No transmission is ever repeated

   - Each transmitted message should have **different contents**

   - Receiver should **ignore messages** that have already been sent

   - Keep track of the last used code

     ❑ But, re-synchronization should be considered

2. The packet contents are virtually impossible to predict, even if previous messages are known

   - Ultimately, the system should be designed to make it difficult for an attacker to guess and replicate the message

     ❑ “Serial number(=TxID)" and “sync counter" are the information that needs to be kept confidential

     ❑ It can only be read by the intended recipient

3. Prevent unauthorized access

   - Filtering mechanism, a unique serial number(TxID) is used to achieve

     ❑ TxID should not be guessable and should not appear in a sequential format

Cipher Key management On 1- Way RF

- **Type1:** Pre-programmed cipher key

  Type1-1 Fixed Key

  - The implementation is simple and cost-effective
  - same cipher key is used across multiple transmitters, increased security risk

  Type1-2 Random Key

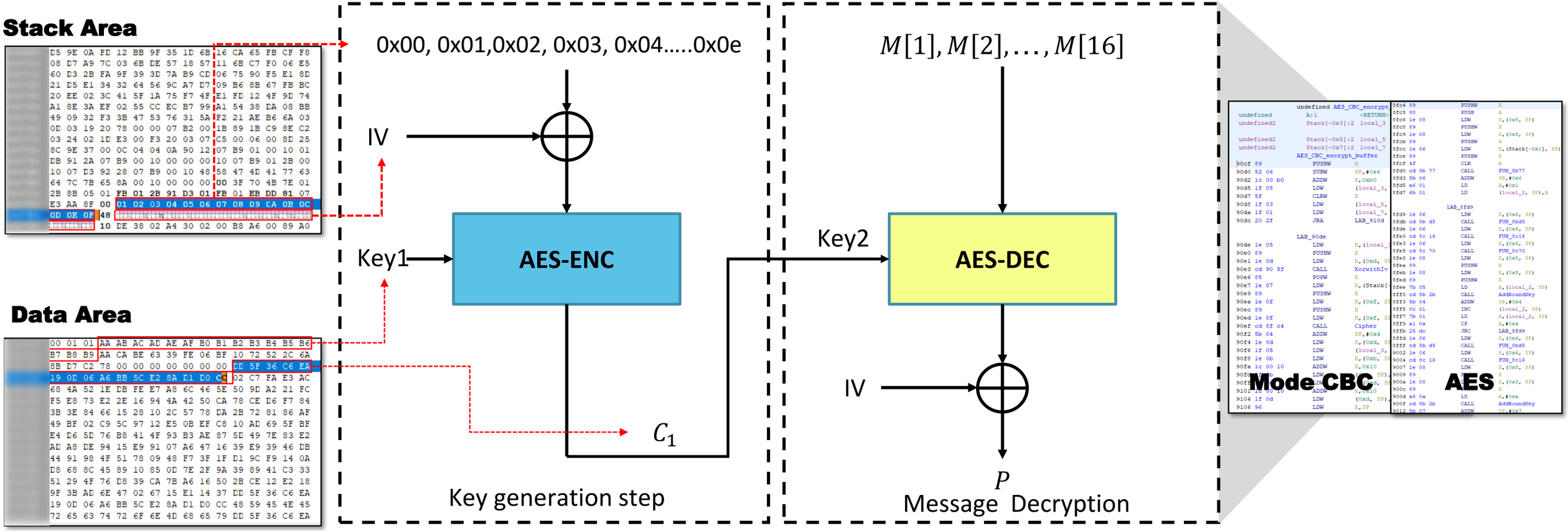
  - Random like generated cipher keys is used, better protection
  - **Lost or damaged, a new transmitter cannot be used with the receiver, if not have a learning mechanism**

- **Type2:** Derives the encryption key by using received data during normal operation
  - It is more secure than Fixed Key, and more flexible than random key
  - **This method requires additional security measures for enhancing**

- **Type3:** Transmit key generation seed value at learning time
  - The receiver uses this seed value to derive the same encryption key
  - During normal operation, attacker will not have any information about the encryption key used

AES-based Door Lock RF Encryption

- Using hardcoded values for the IV and plaintext in every key generation operation
  - The values are hard-coded in code area



The Flaws in Key Generation

- Derives the encryption key by using received data during normal operation
  - It can aid key synchronization, but it is important to note that security relies on encryption scheme, key derivation process

If M[0] ‘1’ is even

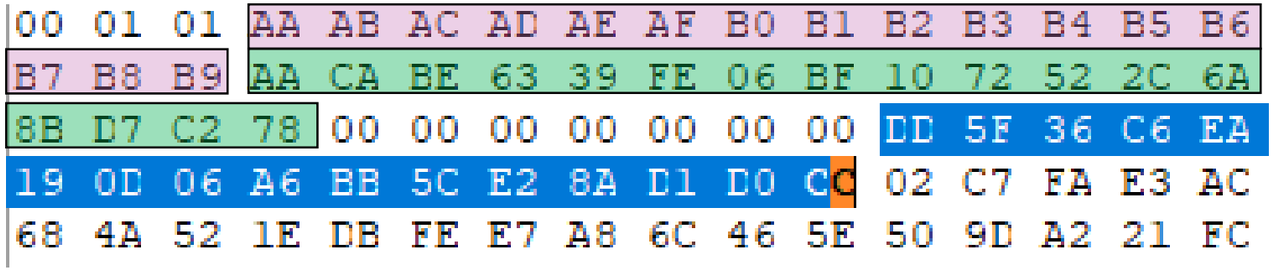
Received Packet 17-Bytes

AA CA BE 63 39 FE 06 BF 10 72 52 2C 6A 8B D7 C2 78

AA = b’1010_1010 (number of bit 1 is EVEN)

Generated Key1 16-Bytes

= AA AB AC AD AE AF B0 B1 B2 B3 B4 B5 B6 B7 B8 B9



If M[0] ‘1’ is odd

Received Packet 17-Bytes

BC 0E 2C 19 35 44 1B F7 52 1D 43 6D 0A 10 C6 20 DA

BC = b’1011_1100 (number of bit 1 is ODD)

~ 0xBC = 0x43 (Bit inversion)

= 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52

If M[0] ‘1’ is even
then $\text{key1} = M[0], M[0]+1, M[0]+2, \ldots M[0]+15\}$

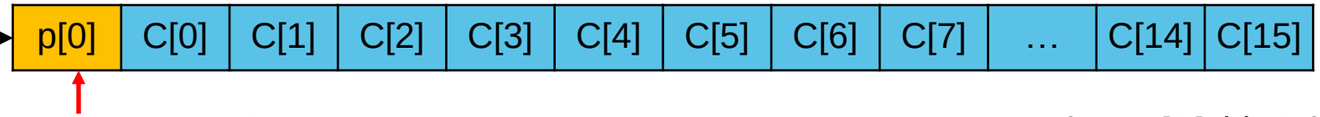
If M[0] ‘1’ is even
then $\text{key1} = \sim M[0], \sim M[0]+1, \sim M[0]+2, \ldots \sim M[0]+15\}$

Little trick, confusion to an attacker

Inadequate Security of LUT-Based Key Generation

❑ Key1 ← LUT[sync counter]

- The tables have fixed values across all product lines. not derived by secret.(pre-set and static)



Encryption Key Seed

$LUT[i]$

Baseband Data = {Key1[0] || C_2}

f2560	F8	EC	E4	ED	EE	F7	E8	FC	E7	FE	D8	1E	D1	18	DE	1D
f2570	D6	16	D3	DC	11	1C	D4	D2	1F	DF	17	DA	D5	1A	D7	19
f2580	DB	13	1B	12	D9	10	D0	14	DD	15	28	3E	21	38	2E	3D
f2590	26	36	23	2C	31	3C	24	22	3F	2F	37	2A	25	3A	27	39
f25a0	2B	33	3B	32	29	30	20	34	2D	35	48	5E	41	58	4E	5D
f25b0	46	56	43	4C	51	5C	44	42	5F	4F	57	4A	45	5A	47	59
f25c0	4B	53	5B	52	49	50	40	54	4D	55	68	7E	61	78	6E	7D
f25d0	66	76	63	6C	71	7C	64	62	7F	6F	77	6A	65	7A	67	79
f25e0	6B	73	7B	72	69	70	60	74	6D	75	88	9E	81	98	8E	9D
f25f0	86	96	83	8C	91	9C	84	82	9F	8F	97	8A	85	9A	87	99
f2600	8B	93	9B	92	89	90	80	94	8D	95	A8	BE	A1	B8	AE	BD
f2610	A6	B6	A3	AC	B1	BC	A4	A2	BF	AF	B7	AA	A5	BA	A7	B9
f2620	AB	B3	BB	B2	A9	B0	A0	B4	AD	B5	C8	DE	C1	D8	CE	DD
f2630	C6	D6	C3	CC	D1	DC	C4	C2	DF	CF	D7	CA	C5	DA	C7	D9
f2640	CB	D3	DB	D2	C9	D0	C0	D4	CD	D5	E8	FE	E1	F8	EE	FD
f2650	E6	F6	E3	EC	F1	FC	E4	E2	FF	EF	F7	EA	E5	FA	E7	F9
f2660	EB	F3	FB	F2	E9	F0	E0	F4	ED	F5	00	01	02	03	04	05
f2670	06	07	08	09	00	01	02	03	04	05	8A	3C	23	F0	02	E0

256 Byte Loockup Table

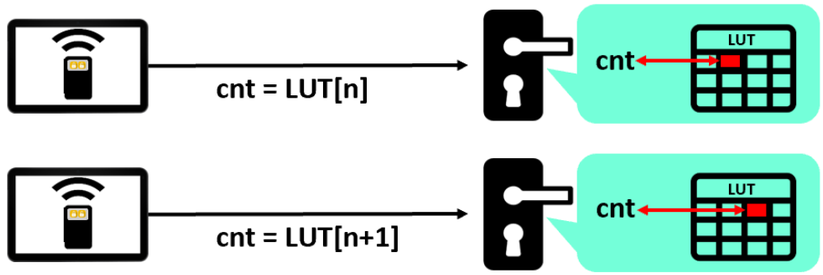
j	Upper_Nibble	Lower_Nibble	E.g., Table[j]
0x00	0	8	0x08
0x01	1	E	0x1E
0x02	0	1	0x02
0x03	1	8	0x19
0x04	0	E	0x0F
31	1	5	0x16

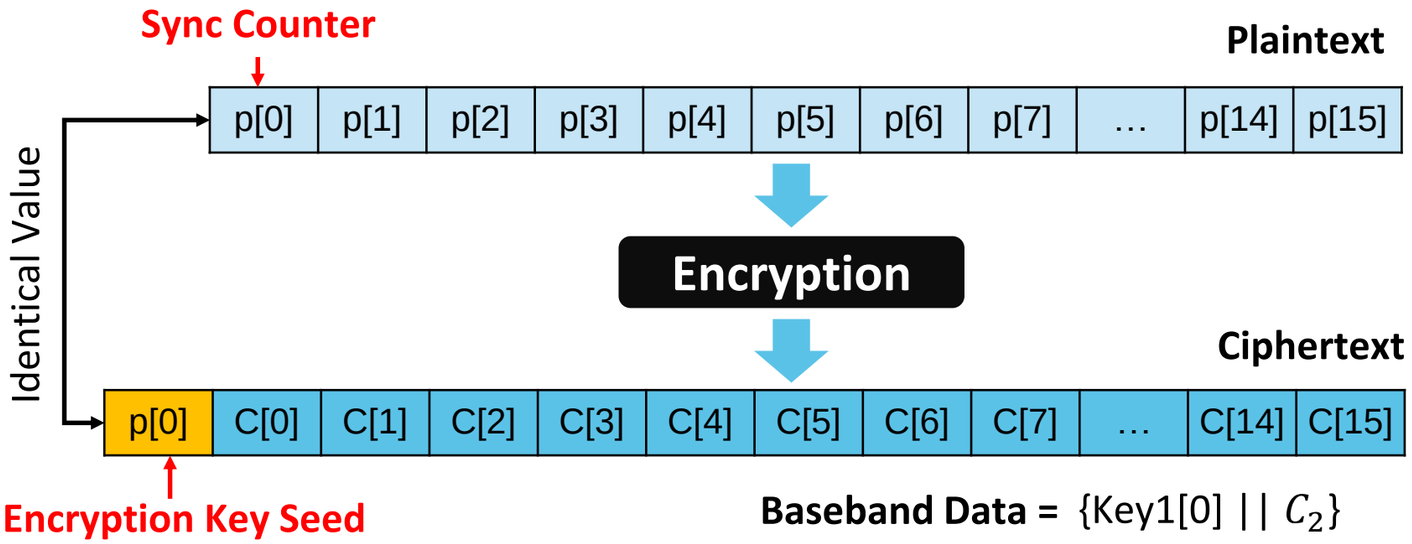
For *i ∈ 0, 2, 4, 6, 8, 10, 12, 14*
  For *(j = 0 : 31)*
    (Upper_Nibble[j] + i) << 4 | Lower_Nibble [j]

LUT Generation Method

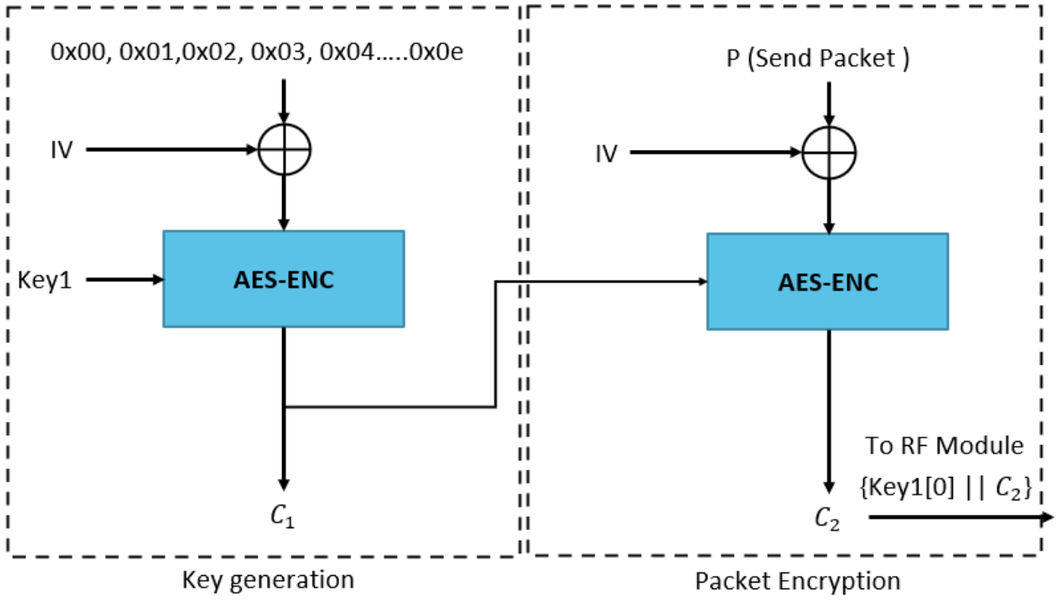
Counter as Key Generation Seed

- **The Encryption key seed is also used to sync counter**
- The key seed is exposed on is being transmitted in plaintext
  - An attacker could decrypt ciphertext at any time
- It rely on the secrecy of the encryption scheme and key generation mechanism
  - It may potentially allow an attacker to break other devices that use the similar implementation





Inferred Encryption scheme



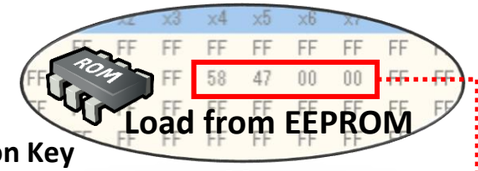
Proprietary Cipher on Door Lock

- Transmit cipher key at learning time
  - Preprogrammed 8-byte key for 8 rounds, with the 5th key being exposed in the packet for 2 rounds
  - This cipher is to use a combination of substitution and operator table to generate a rolling counter

Packet Structure

ID[0]	ID[1]	ID[2]	ID[3]	CNT[0]	CNT[1]	CNT[2]	CNT[3]	CNT[4]	CNT[5]	CNT[6]	CNT[7]	CMD	Key[4]

8~9 Round Encryption Key



Memory

BANK1	*Data*
29h	60
2Ah	EF
2Bh	DF
2Ch	13
2Dh	00
2Eh	00
2Fh	00
30h	00
31h	00
32h	00
33h	58
34h	47

Preprogramed 8Byte Key

8-Byte Key (1~8Round) and Key[4] (9~10Round)

Total 10 Round

Key[i] & 0x0F (Lower Nibble)

Key[i] & 0xF0 (Upper Nibble)

Lookup

Lookup

If Key[4] 0x2D

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
													C9		
													27		
													B7		
													05		

T1: Byte Substitution Table

If Key[4]0x2D

00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
		XOR													
		+1													
		SUB													
		NOT													

T2: Operator(Arithmetic+Logical) Table

Operand2

Operator

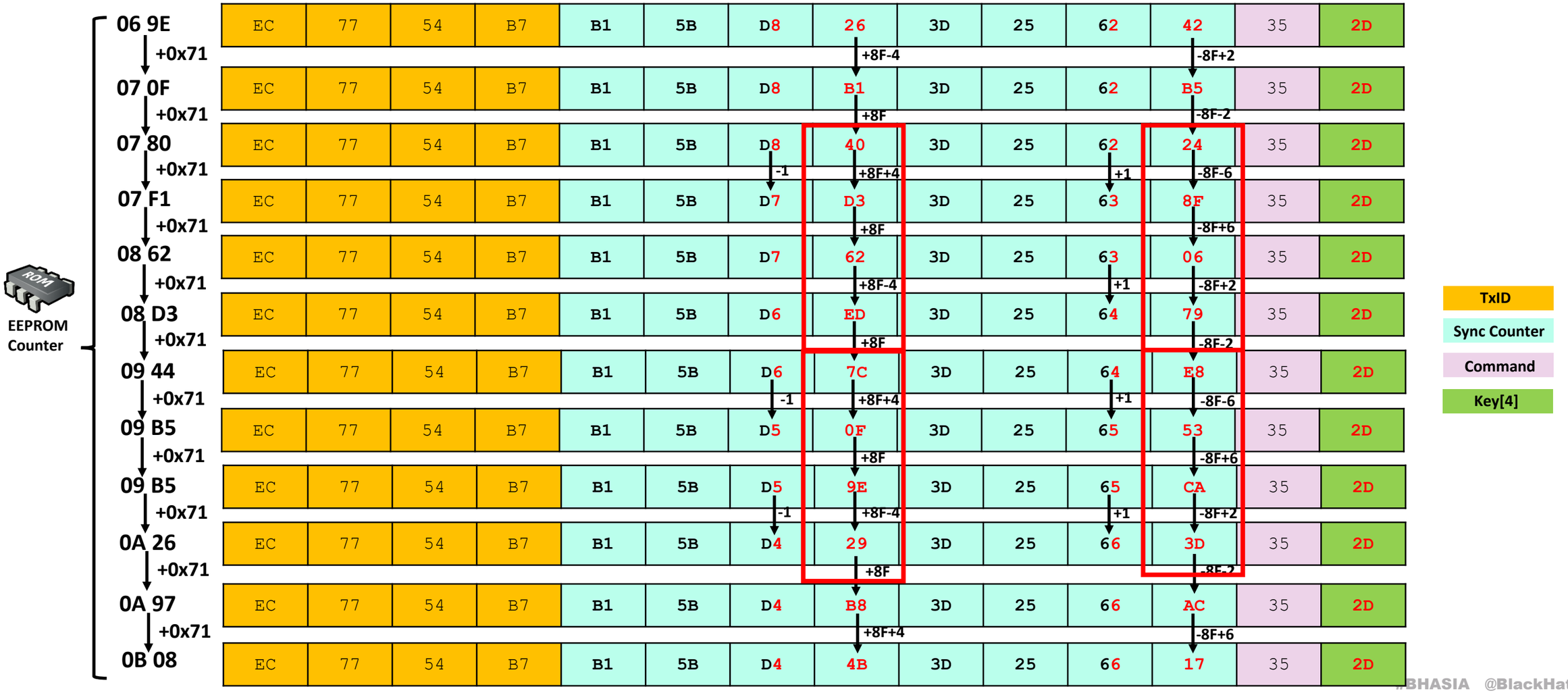
Operand1

Calculation

update

Monotonous Pattern of Code

- The consistent pattern code may be due to the absence of a permutation feature
  - The generated rolling counter has a somewhat monotonous pattern, making it easy for attackers to predict



Evaluation : Confidentiality

- Evaluation on a limited set of doorlocks
  - All models are vulnerable to picking the lock with sniffed RF packets

Vendors	Models	Algorithm	Confidentiality	Remarks
A	A-1	AES-128	✗	1) Deriving the encryption key from received data 2) Sync counter is leaked on Packet 3) Key Seed is leaked on Packet
B	B-1	AES-128	✗	
	B-2	AES-128	✗	
	B-3	AES-128	✗	
	B-4	AES-128	✗	
C	C-1	AES-128	✗	
	C-2	AES-128	✗	
	C-3	AES-128	✗	
D	D-1	AES-128	✗	
	D-2	AES-128	✗	
E	E-1	None	✗	-
F	F-1			
G	G-1			
H	H-1			
I	I-1	XTEA	✗	1) Pre-programmed Fixed Key(Hardcoded)
J	J-1	Proprietary Encryption	✗	1)Pre-programmed random key is transmitted at learning time 2) It is feasible to deduce the next code from the packet 3) Serial Number(=TxID) is leaked to plaintext
	J-2			

✗ Compromised

✗ Potentially compromise

√ Probably safe

Authentication in RF-based Door Locks

The crucial of ID in RF system

Principles of Secure Rolling Code

To ensure Secure Rolling Code transmission (The three critical properties)

1. No transmission is ever repeated

   - Each transmitted message should have **different contents**

   - Receiver should **ignore messages** that have already been sent

   - Keep track of the last used code

     ❑ But, re-synchronization should be considered

2. The packet contents are virtually impossible to predict, even if previous messages are known

   - Protect the confidentiality of the rolling code (Encryption Algorithm)

     ❑ "TxID" and "rolling counter" are the information that needs to be kept confidential

     ❑ It can only be read by the intended recipient

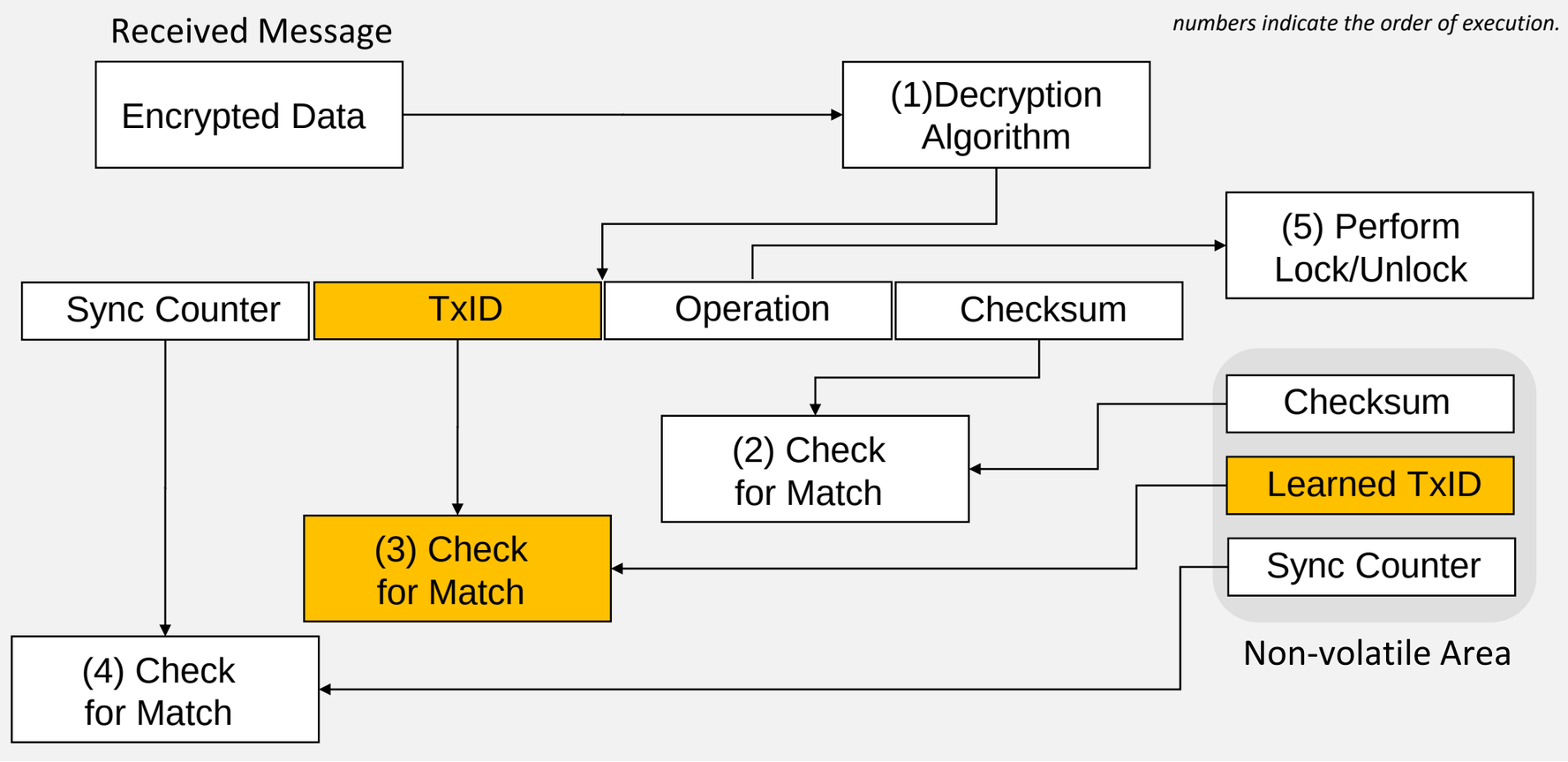
3. Prevent unauthorized access

   ❑ Serial Number is learning information in most of door lock, ID verification is a common method for authorizing

   ❑ Serial Number (=ID) should not be guessable and should not appear in a sequential format

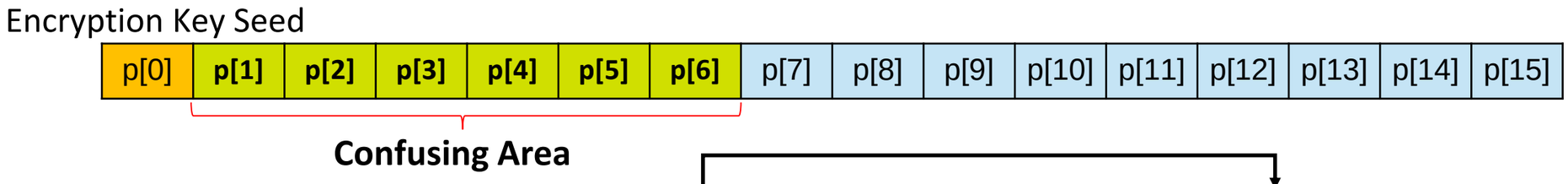
Authentication check

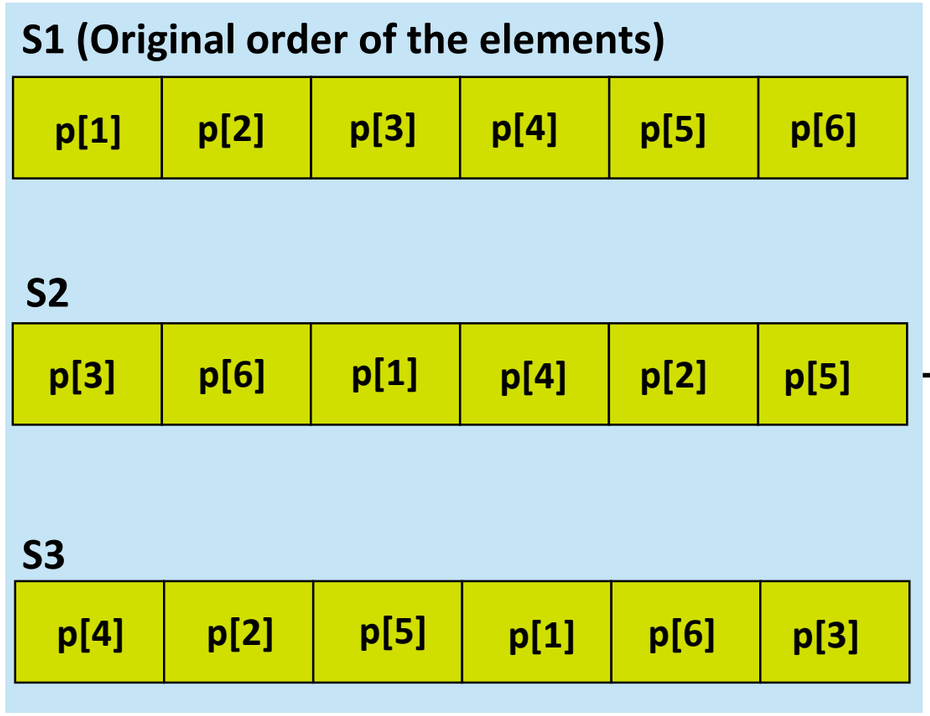
- Filtering mechanism, a unique serial number(TxID) is used to achieve
- When a valid message is received, the message is decrypted, and the serial number is used to determine if it is from a learned transmitter. If it is from a learned transmitter, the synchronization counter is verified



Packet Confusing before Encryption

- It may be difficult to keep track of the original order of the elements





3-States (vendor-specific)

```
function confusing(encryption_key_seed, s1, s2, s3, range1, range2)
    L_Nibble = encryption_key_seed AND 0x0F

    if L_Nibble < range1 then
        circular_shift(s1, L_Nibble)
        return s1
    else if L_Nibble >= range1 AND L_Nibble < range2 then
        circular_shift(s2, L_Nibble)
        return s2
    else
        circular_shift(s3, L_Nibble)
        return s3
    end if
end function
```

Vendor-specific

Circular Shift based confusing

Sequential ID values

- TxID(=Serial Number) should not be guessable and should not appear in a sequential format
  - If the TxID values are sequential or predictable, the attacker can easily predict the next door's value
- Actual TxID changed by only 2~3Bytes, the number of possible values is significantly reduced
  - It is important to use unique and non-sequential TxID values to prevent potential attacks.

$TxID_1$= {0xBA, 0xA4, 0x0A, 0xA6}
$TxID_2$= {0xBA, 0xA3, 0x0A, 0x43}
$TxID_3$= {0xBA, 0xA6, 0x08, 0x6C}
$TxID_4$= {0xBA, 0xA2, 0x09, 0x5E}
$TxID_5$= {0xBA, 0xA4, 0x09, 0xC4}
$TxID_6$= {0xBA, 0xA4, 0x06, 0xC4}

E.g., Vendor B

$TxID_1$= {0x95, 0xA5, 0x28, 0xAE}
$TxID_2$= {0x95, 0xA4, 0x26, 0xFC}
$TxID_3$= {0x95, 0xA4, 0x1B, 0xDB}
$TxID_4$= {0x95, 0xA5, 0x26, 0xAE}

E.g., Vendor A

Vendors	Models	Serial Number	Remarks
A	A-1	✗	≈ 2.5 ~ 3Bytes
B	B-1	✗	≈ 2 Bytes
	B-2	✗	
	B-3	✗	
	B-4	✗	
C	C-1	✗	
	C-2	✗	
	C-3	✗	
D	D-1	✗	
	D-2	✗	
E	E-1	✗	≈ 3 Bytes
F	F-1	✗	≈ 2.5 ~ 3Bytes
G	G-1	✗	
H	H-1	✗	
I	I-1	✗	
J	J-1	√	≈ 4 Bytes

✗ Very Weak

✗ Weak

√ Moderate

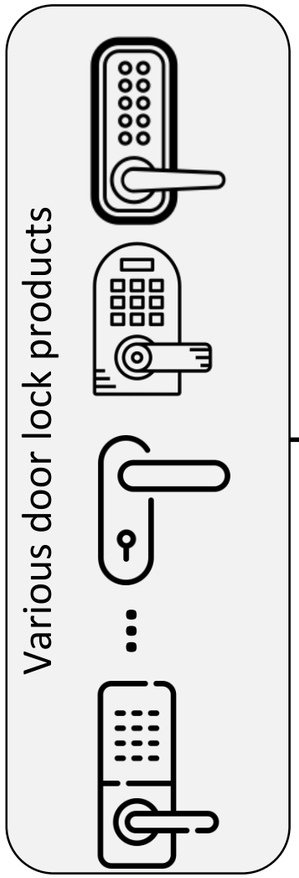
√ Strong

RF Lock Picking Tool Preparation

for RF Capture, Decoding, and Transmission

An overview of making a RF lock picking tool set

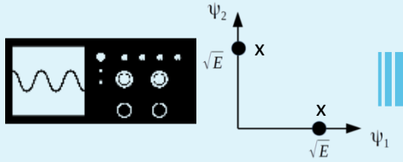
- The RF Lock Picking tool set : **CodeCatcher** + **CodeCrusher**
  - **CodeCatcher** : The sniffer could include for demodulation, decoding, descrambling, decrypt, digital data recording
  - **CodeCrusher** :The transmitter includes the reverse of the above functions, replaying signals or sending custom signals



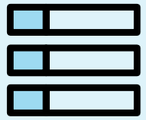
Essential Information gathering

1) RF parameter Measurement

2) Training sequence confirmation

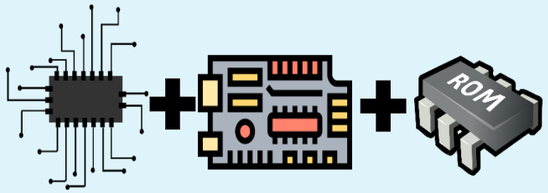


3) Baseband Decoding And Symbol Mapping



4) Examine Packet Structure

Two method for making RF Lock Picking tool set

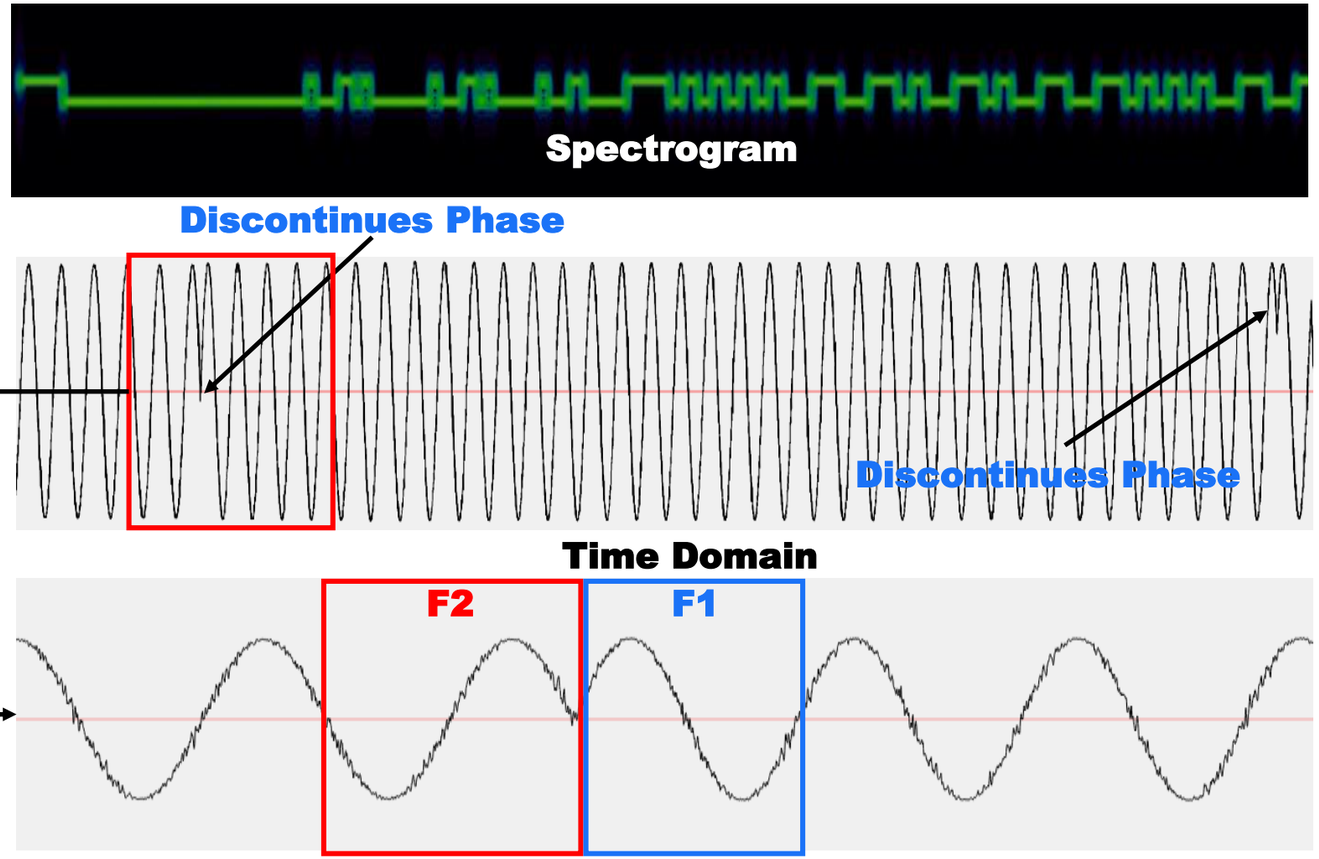


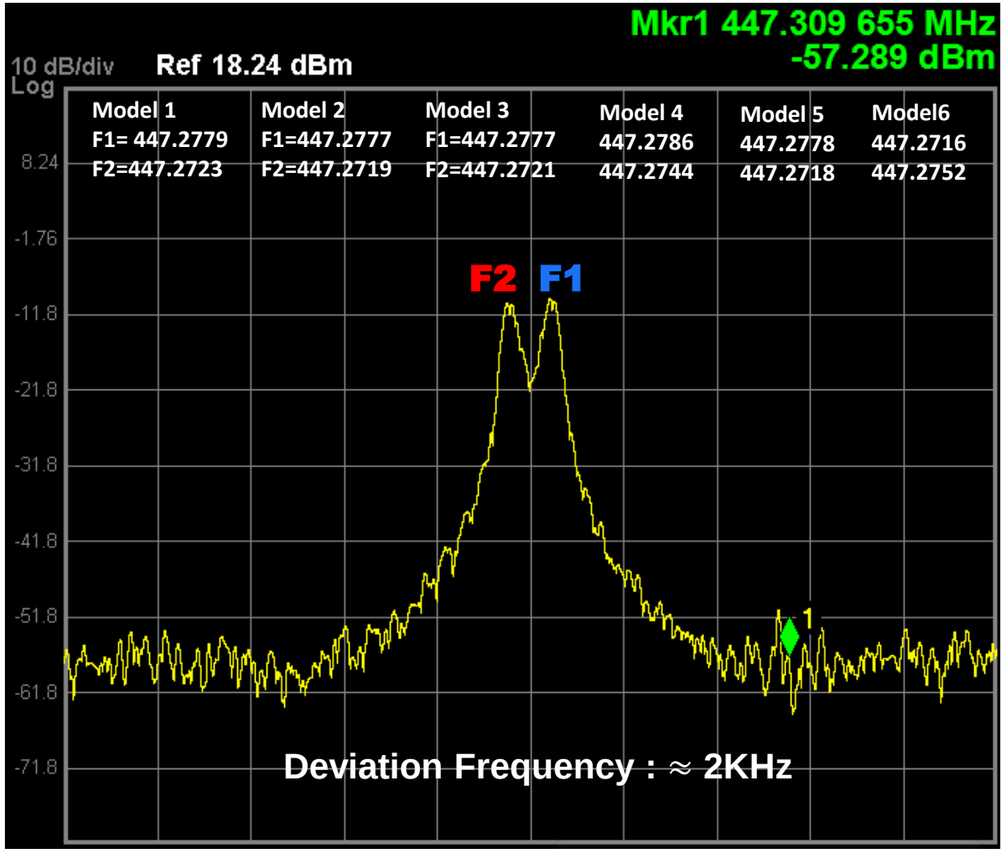
RF Chip + MCU
(Cost Effective & Easier to conceal & multiple deployment)

SDR + GNURadio
(Performance & Flexible)

Viewing Door Lock RF Signal

- BFSK Modulation
- Center frequency : 447.274 MHz or 447.261 MHz
- Modulation and Deviation : 2KHz Fix (Regulation)





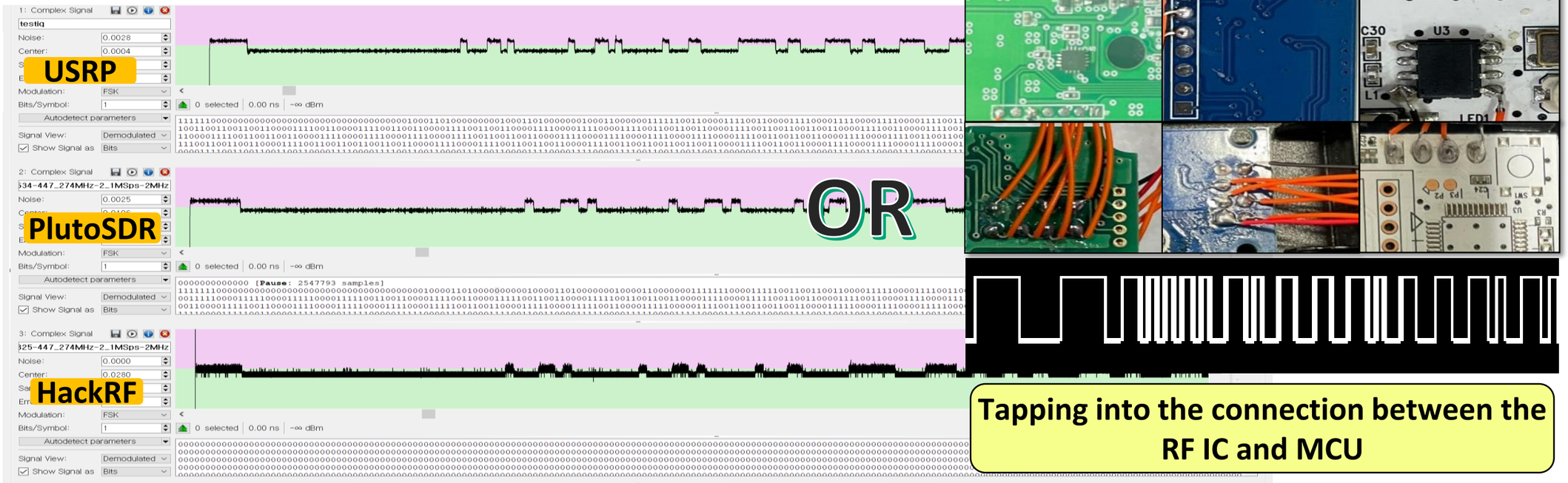
Viewing RF Baseband Signal

- URH can provide insights for base band analysis, without any RF knowledge
  - But, It may be impractical to continuously monitor and collect signals for our real-world attack
- Tapping into the connection between the RF IC and MCU is also best option
- PS. Small deviation frequency might cause interference in the IF signal
  - Note : Super heterodyne receiver, HackRF one, the optimal deviation frequency was found to be 100Khz or higher

USRP

PlutoSDR

HackRF

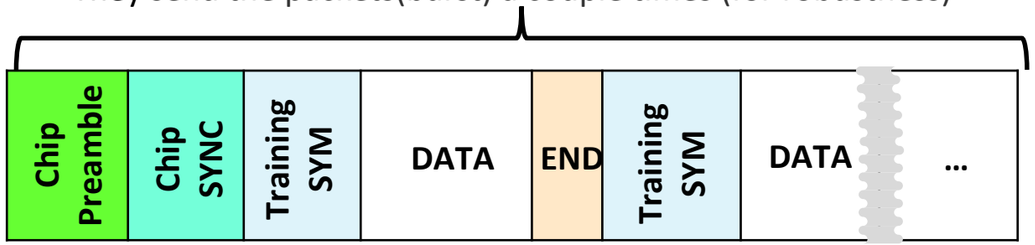


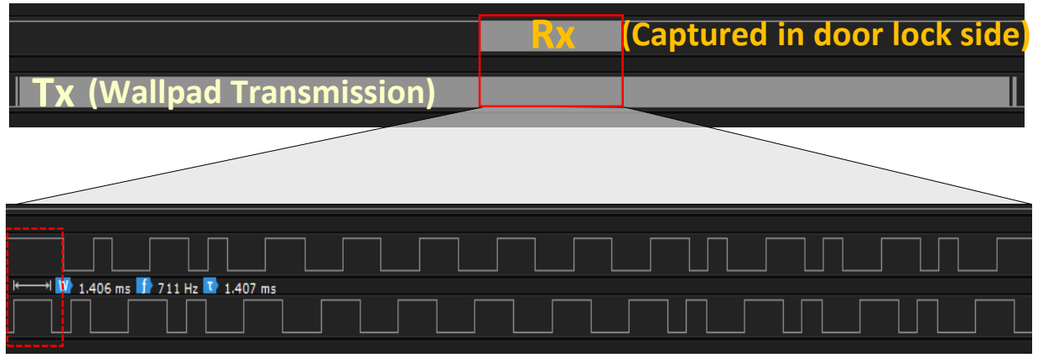
Tapping into the connection between the RF IC and MCU

Training Sequence for custom packet

- Vendor-specific training sequence for timing synchronization
  - The preamble and syncword generated by the RF chip handler are not used in door locks

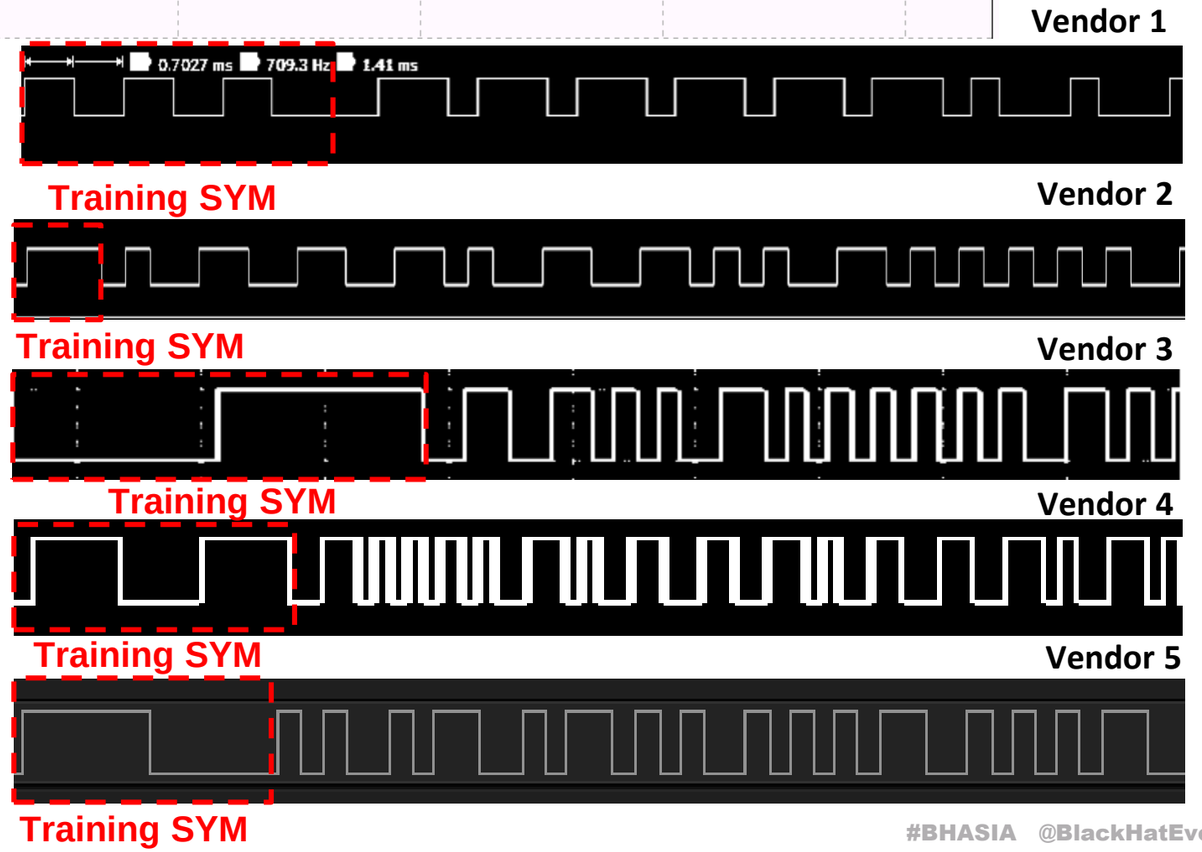
They send the packets(burst) a couple times (for robustness)





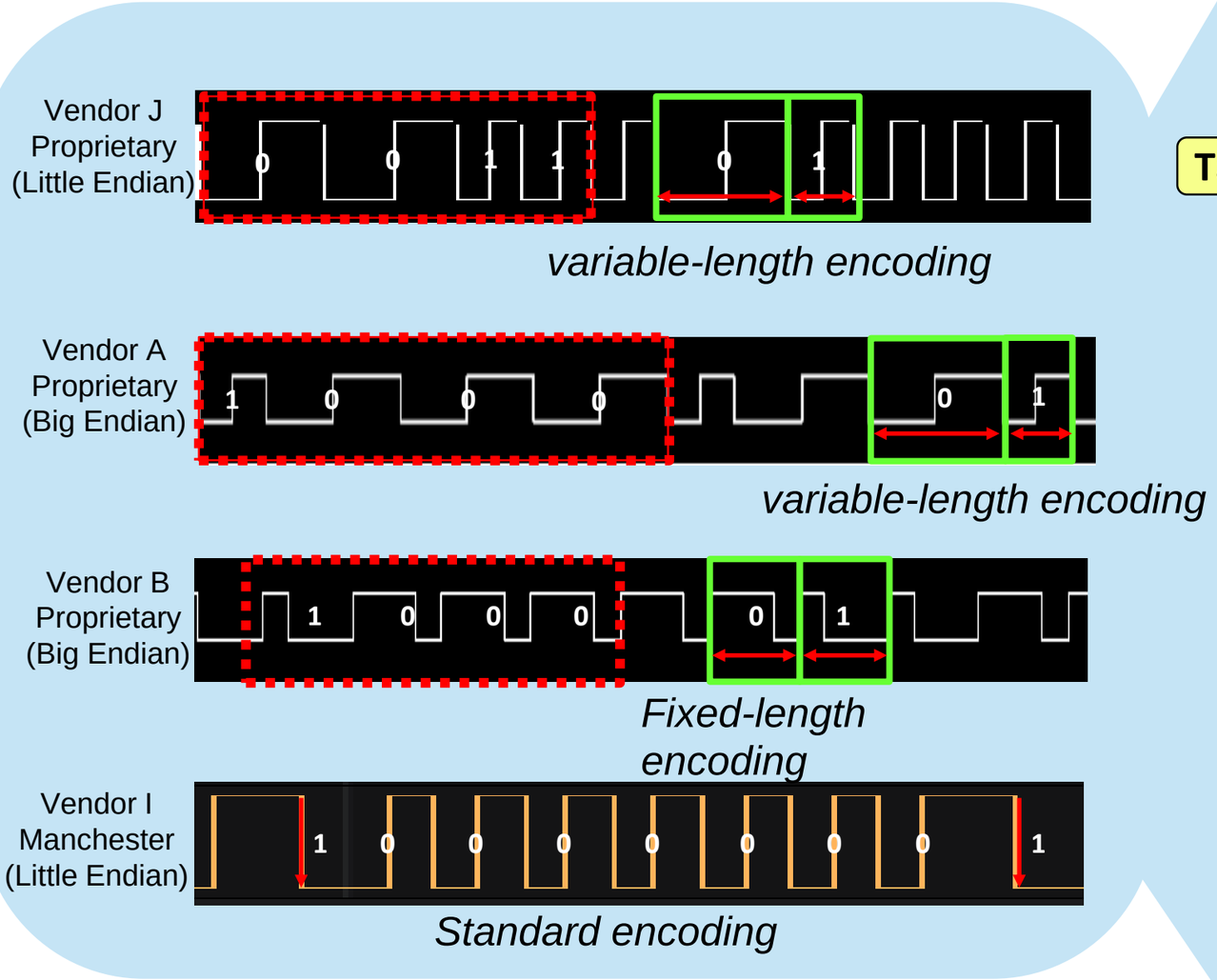
Soft SYNC Detection (Compare Pattern & Duration)

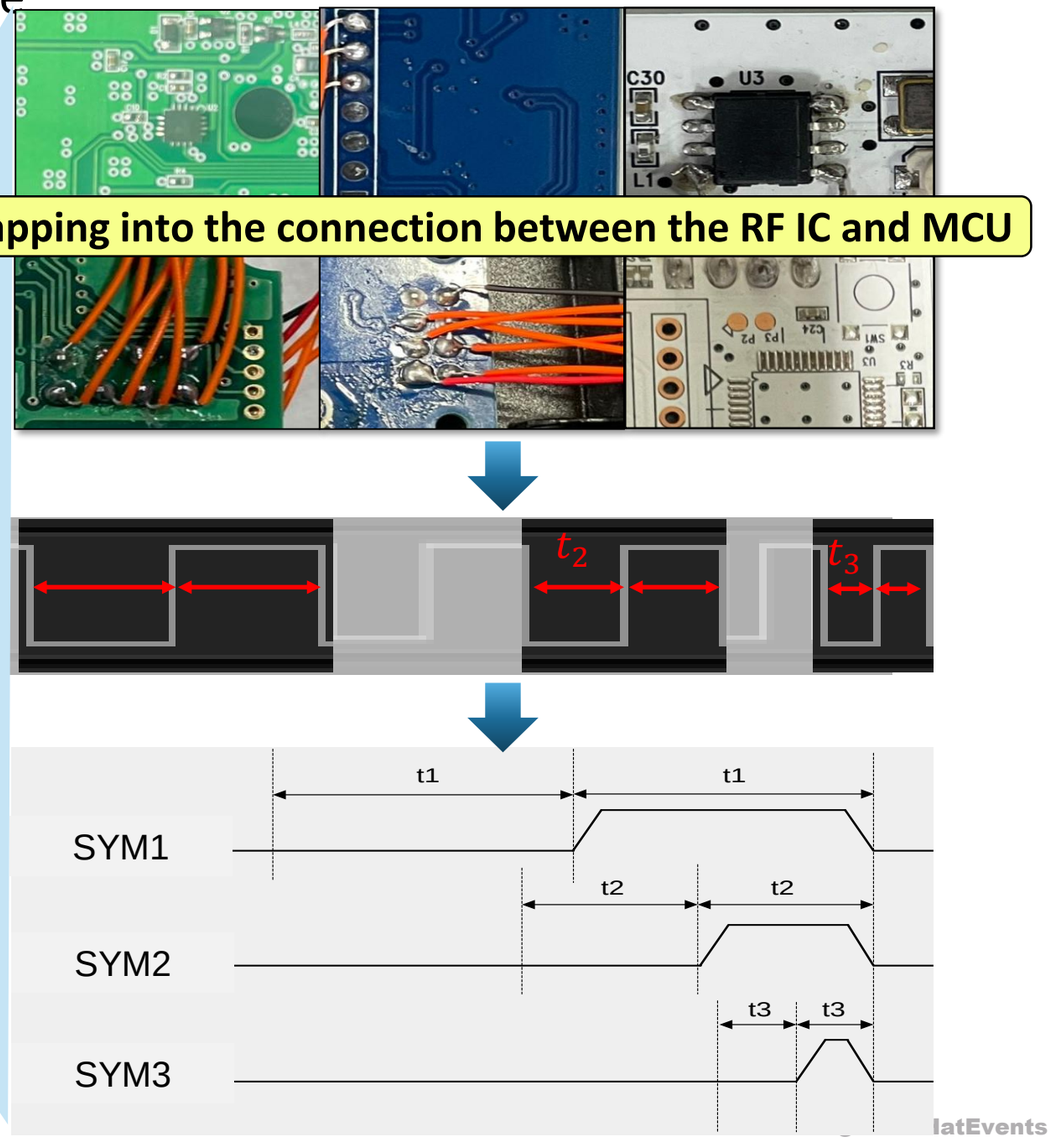
If there is a match, the data is accepted



Proprietary Baseband Encoding

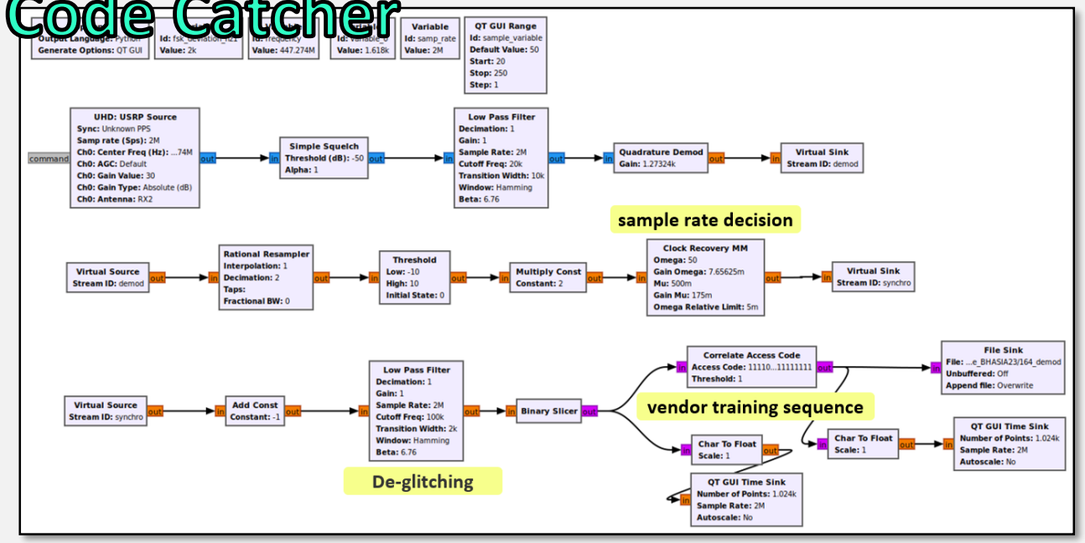
- Vendor-specific digital encodings in RAW transmission mode





SDR based Code Catcher & Crusher

Code Catcher




```
shaftmom@shaftmom-System-Product-Name:~/Desktop/demod_decode_BHASIA23$
./doorloc sniff and decrypt.exe
```

```
SNIFF&DEC
```

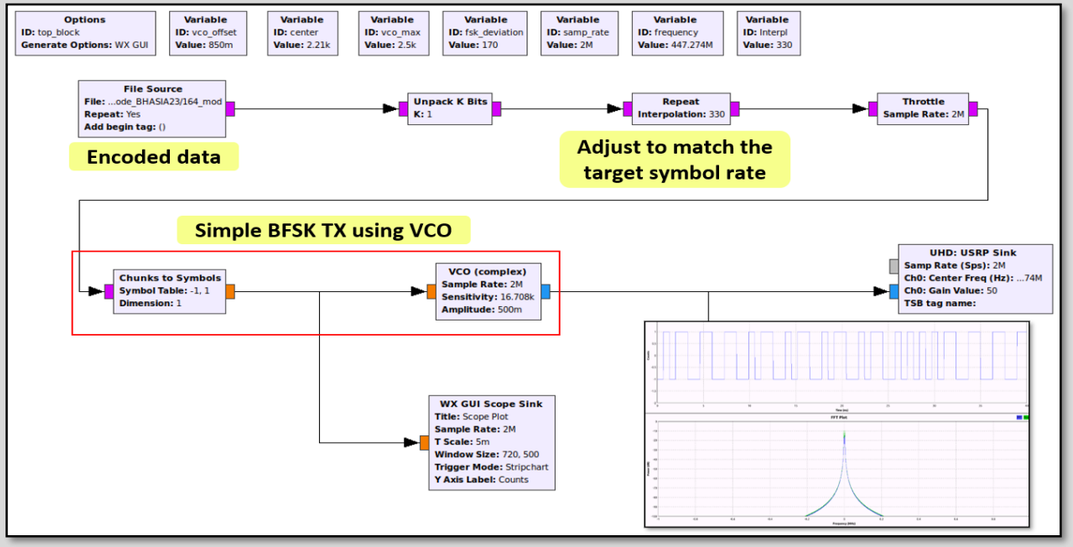
```
==================================================================
preamble detected 1
Recived Message :E 18 E0 87 1D E3 36 B5 1C D4 13 B5 BA BE F2 96 78
First Block Key :F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF 00
Second Block Key :D2 18 49 3F 28 87 D3 52 87 2C 21 66 BC AA 37 49

Encrypted Message :18 E0 87 1D E3 36 B5 1C D4 13 B5 BA BE F2 96 78
3::f195a523890100000000000000000d8
Decrypted Message :F1 95 A5 23 89 01 00 00 00 00 00 00 00 00 D8

==================================================================
preamble detected 2
Recived Message :E 18 E0 87 1D E3 36 B5 1C D4 13 B5 BA BE F2 96 78
First Block Key :F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF 00
Second Block Key :D2 18 49 3F 28 87 D3 52 87 2C 21 66 BC AA 37 49
```

Code Crusher

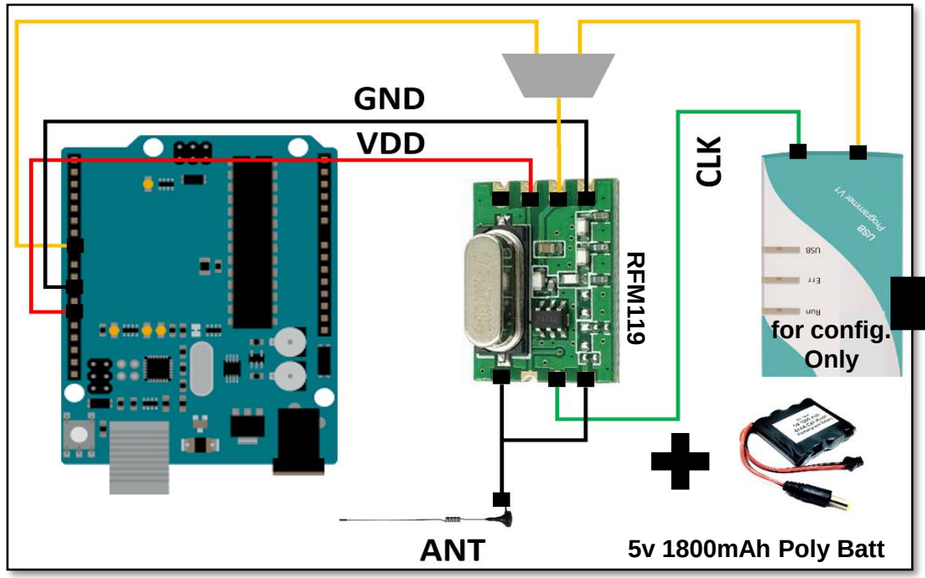
```
shaftmom@shaftmom-System-Product-Name: ~/Desktop/demod_decode_BHAS...
File  Edit  View  Search  Terminal  Help
shaftmom@shaftmom-System-Product-Name:~/Desktop/demod_decode_BHASIA23$
./bb_sig_gen.exe f1 95 a5 23 89
```

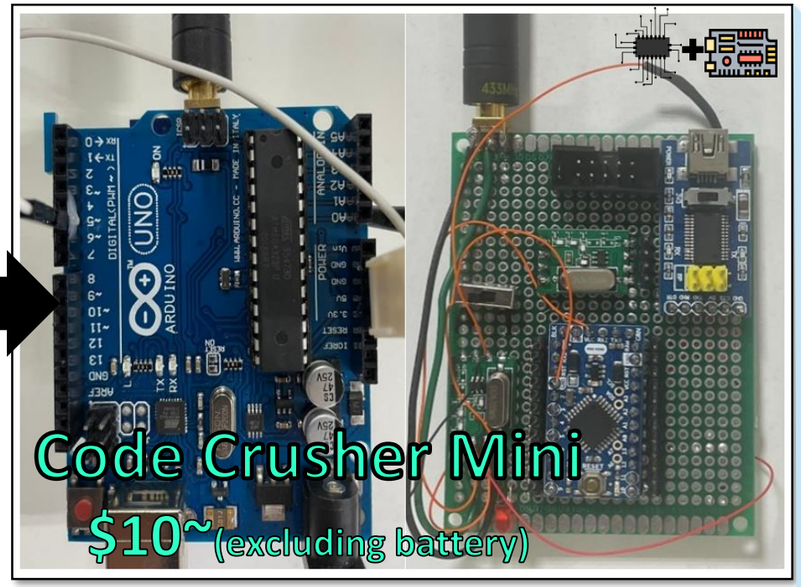
```
BB SIG GEN
```

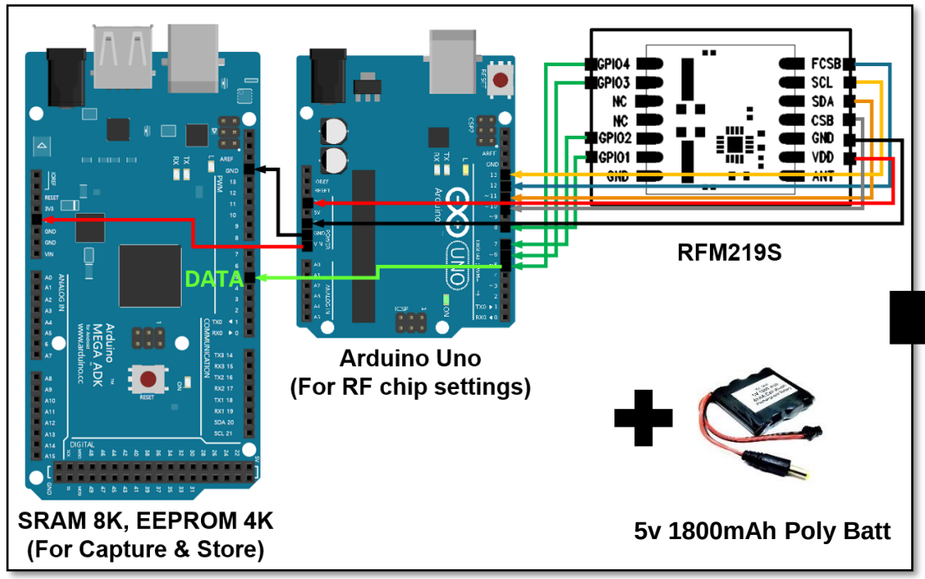
```
The input values are: 0xf1 0x95 0xa5 0x23 0x89
----------------------------Generated key-----------------------------
Key Encryption Key:f2 f3 f4 f5 f6 f7 f8 f9 fa fb fc fd fe ff 00 01
Message Encryption Key:22 f4 08 5a a8 c1 3f f4 ca ad 8a 6b 92 1e 4e 0a

-------------------------Encrypted Message--------------------------
plaintext :f2 95 a5 23 89 01 00 00 00 00 00 00 00 00 d9
ciphertext:87 aa 8a 3e 78 58 9f 1c 5c c2 80 25 f8 76 59 64
Self-Decryption :f2 95 a5 23 89 01 00 00 00 00 00 00 00 00 d9

--------------------------Created Packet----------------------------
created packet :0d 87 aa 8a 3e 78 58 9f 1c 5c c2 80 25 f8 76 59 64
Burst Generated..... PATH: ./164_mod
[INFO] [UHD] linux; GNU C++ version 7.5.0; Boost_106501; UHD_3.14.0.HEA
D-0-g6875d061
```

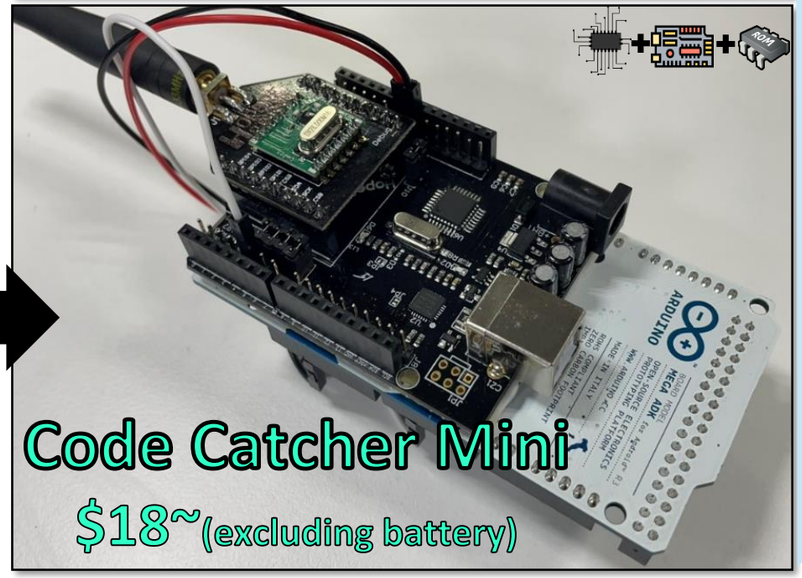


Code Catcher and Crusher Mini





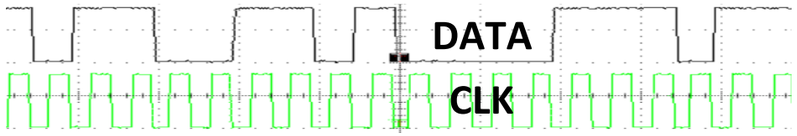




Suffixed at a hidden spot

Code Catcher Mini
Configuring RF Chip for Direct Mode

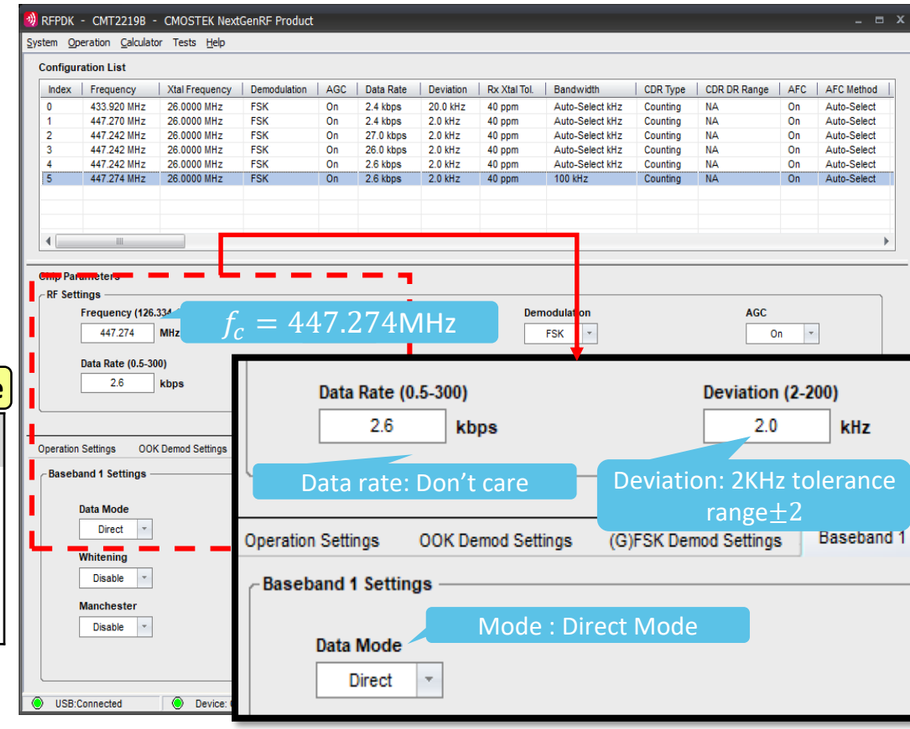
- Validate through signal debugging for RF parameter decision
- According to Transmission Mode..
  - Direct Mode(Aka. RAW transmission mode)
    - Received RXDATA is output on a physical output pin in real-time



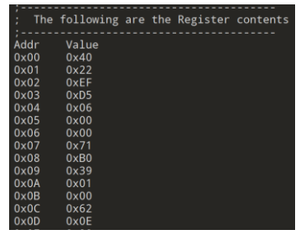
  - Packet Mode
    - The data is packaged into a specific format

Choosing Chips that Support Direct Mode

Direct / Packet (o)	Packet Mode Only (x)	Direct Mode Only (o)
CMOSTEK CMT2219B/CMT2300	Analog Device MAX4147	Melexis TH71101/71120
Silicon LABS Si4455x/443x	CMOSTEK CMT2217	Analog Device MAX7042
TI CC1000/1101	TI CC1125	-
TDA 5150	-	-



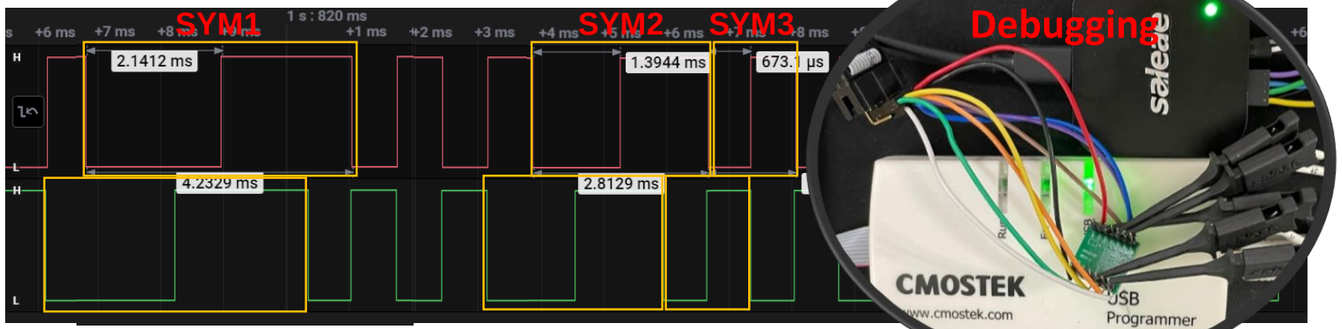
When analog debugging is complete, export the configuration file



RF Config. Value

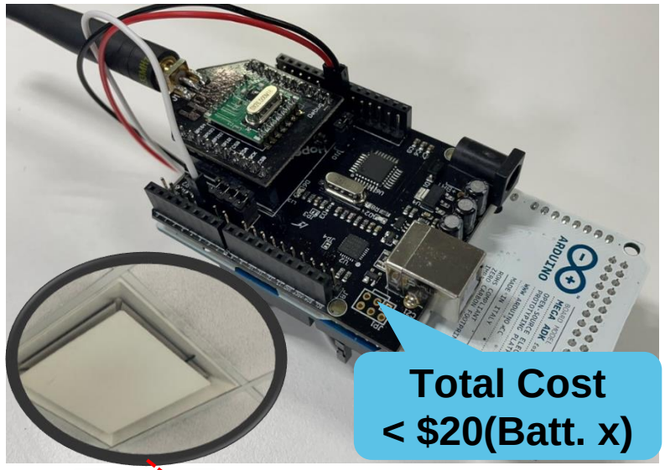
Chip Demod. Signal

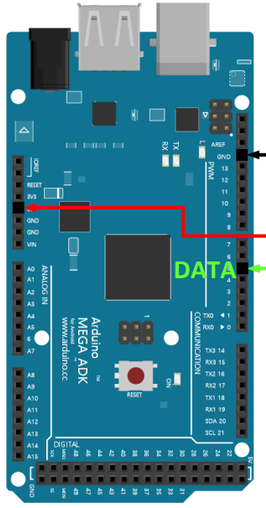
Transmitter B.B. Signal



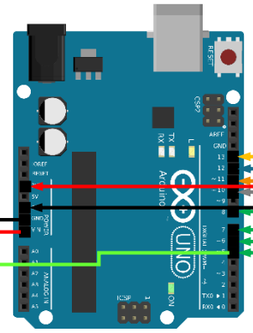
Code Catcher Mini
Portable Door Lock RF Sniffer

- Depending on the HW Spec and sampling rate, capture and store either four or more signals of a door lock
  - Using an 1800mAh battery, can run for approximately 90 hours(3.75Days)

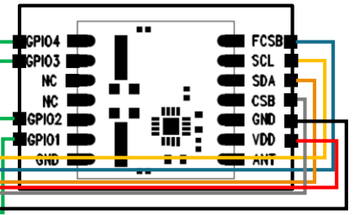




SRAM 8K, EEPROM 4K
(For Capture & Store)
$10(copy)



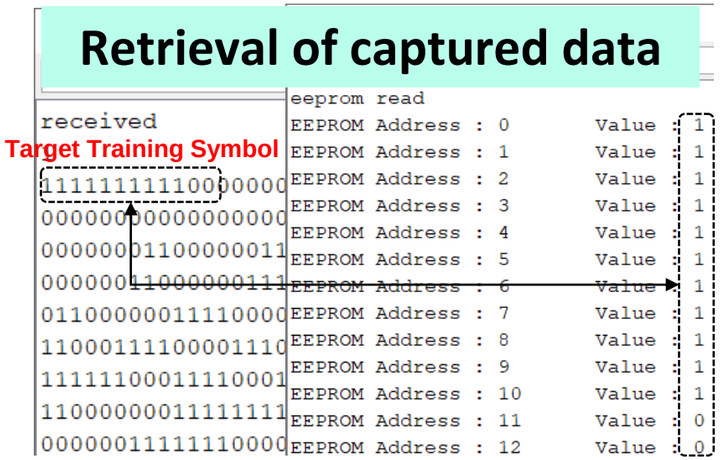
Arduino Uno
(For RF chip settings)
$5(copy)



RFM219S
Sub 1GHz Rx IC
$3

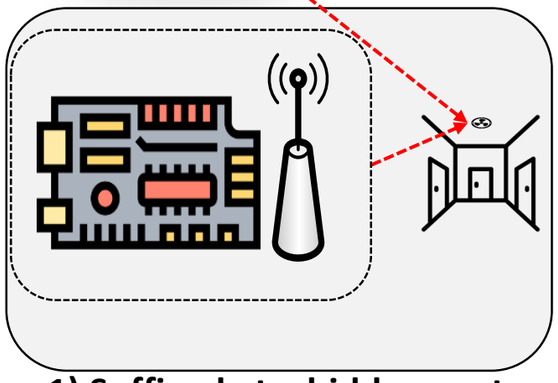
5v 1800mAh Poly

Retrieval of captured data

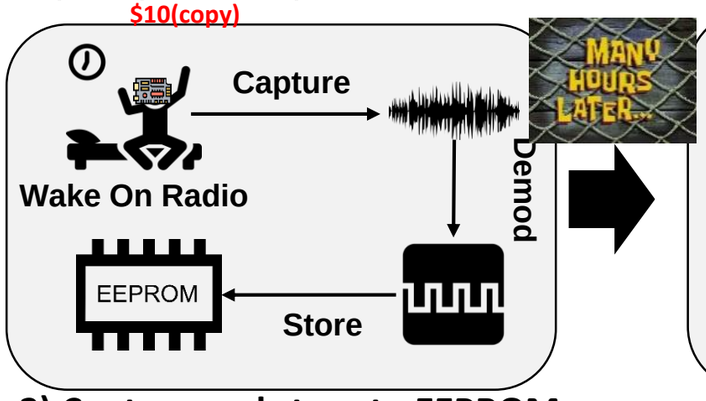


Write signal to EEPROM

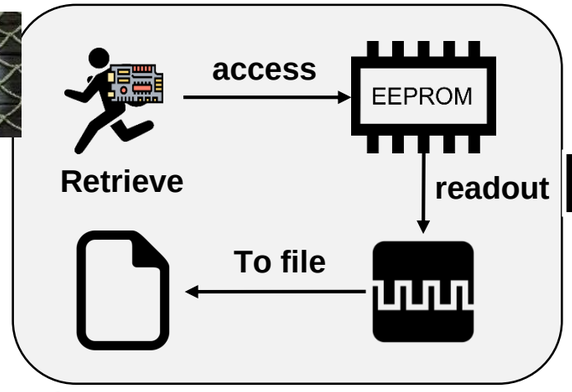
EEPROM Readout



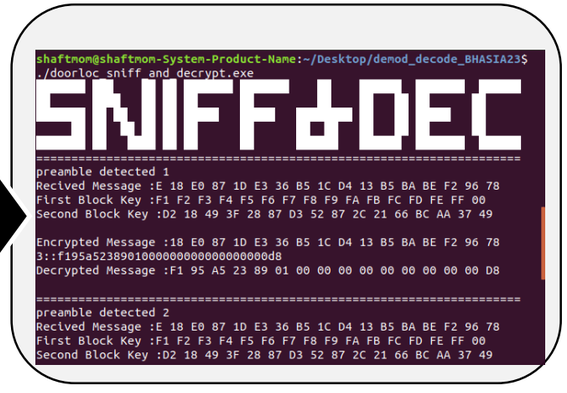
1) Suffixed at a hidden spot



2) Capture and store to EEPROM

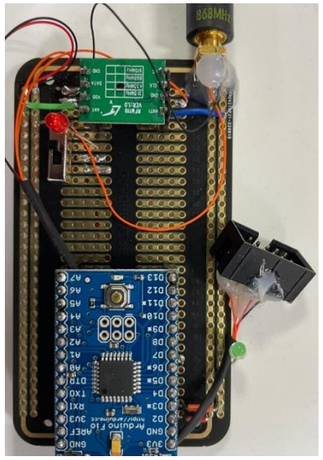


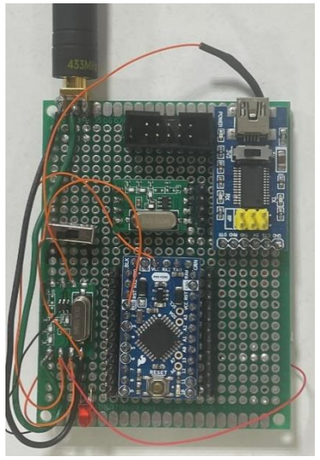
3) Retrieve and EEPROM Readout



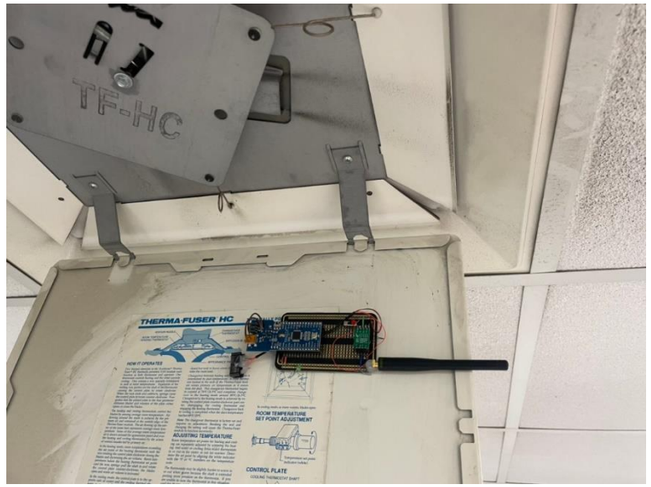
4) Decode and Decrypt

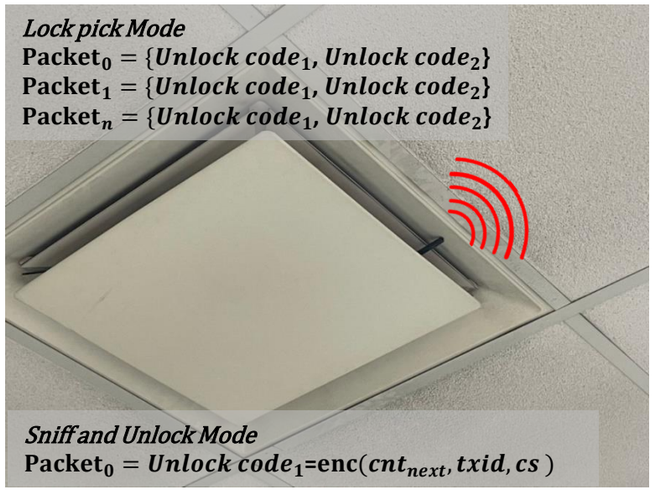
Code Crusher Mini : RF IC Based Tx



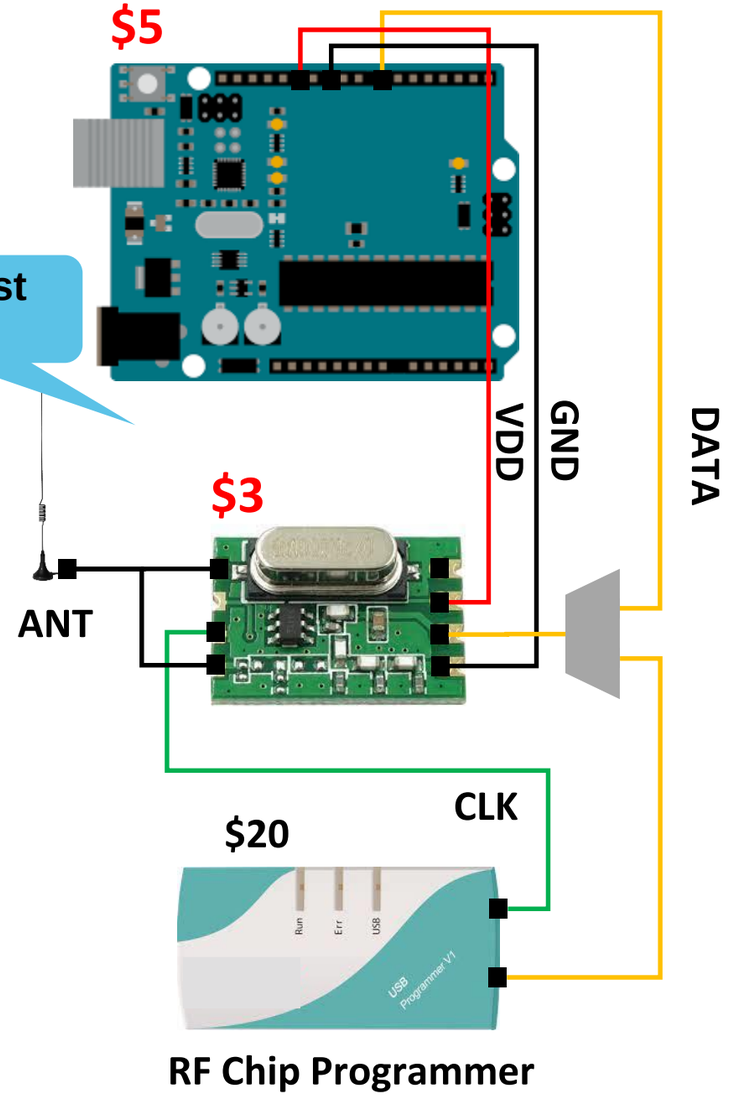


"It can be attached near by target, and the operating time can be extended as much as desired depending on the Lithium Polymer Battery that is installed."

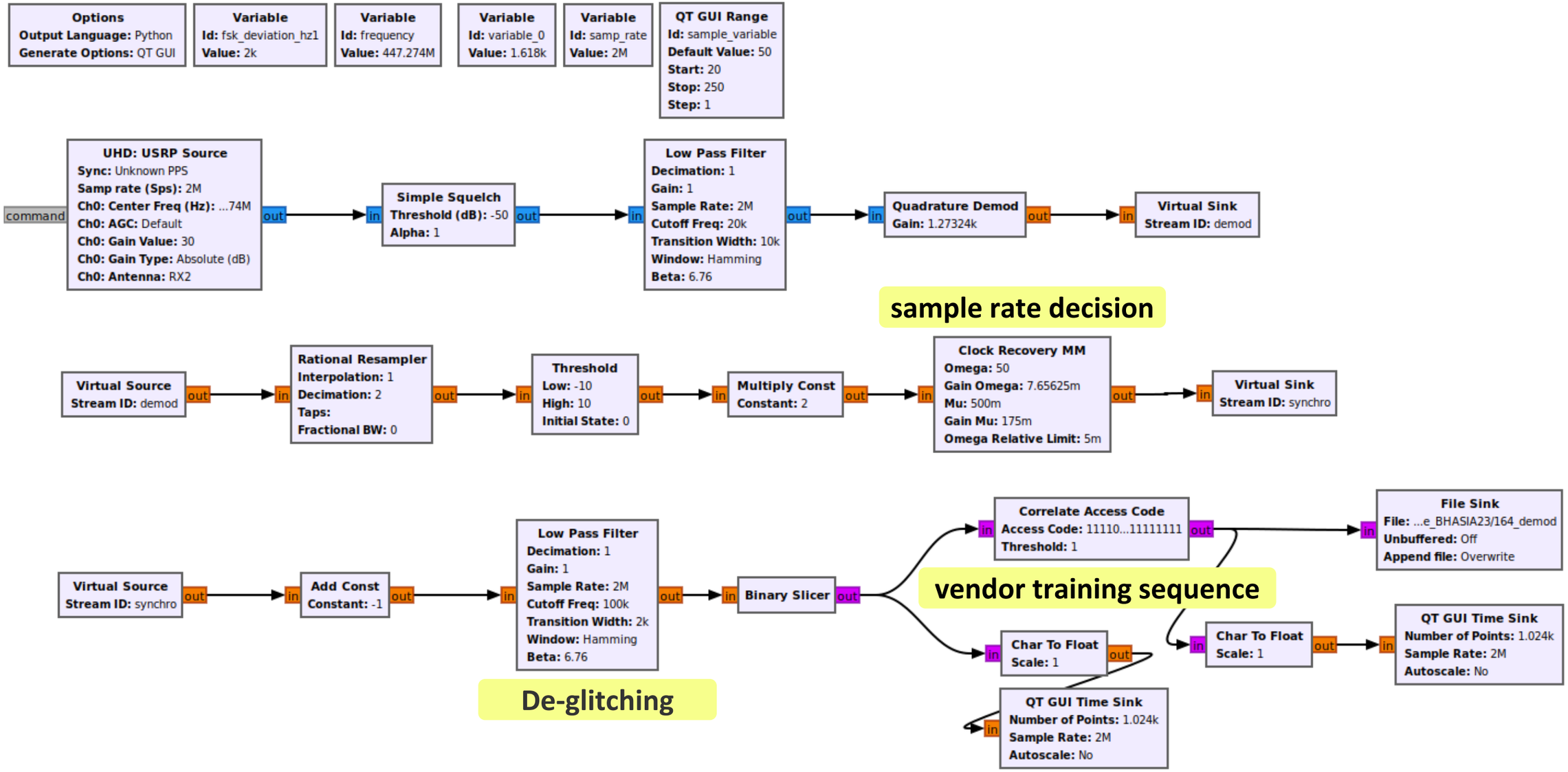




Suffixed at a hidden spot

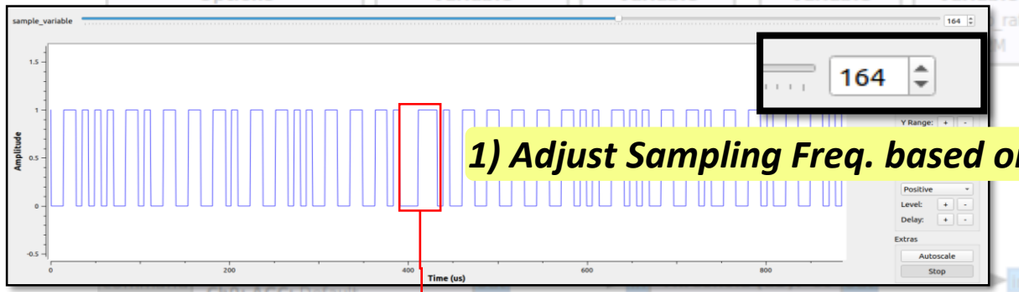


SDR based Code Catcher
gr-block for Door Lock RF Sniffing



SDR based Code Catcher configuration for Sniffing

numbers indicate the order of step.



1) Adjust Sampling Freq. based on the symbol rate

Parameter : 164

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1A10h	00	00	00	00	00	00	00	01	01	01	01	00	00	00	00	00
1A20h	00	00	00	00	00	00	00	00	01	01	01	01	01	01	01	01
1A30h	01	01	01	01	01	00	00	00	00	00	01	01	01	01	00	00
1A40h	00	00	00	00	00	00	01	01	01	01	01	01	01	01	01	00
1A50h	00	00	00	01	01	01	01	00	00	00	00	00	00	00	00	00
1A60h	01	01	01	01	01	01	01	01	00	00	00	00	01	01	01	01
1A70h	01	00	00	00	00	01	01	01	01	00	00	00	00	00	00	00
1A80h	00	00	01	01	01	01	01	01	01	01	00	00	00	00	01	01
1A90h	01	01	01	00	00	00	00	00	00	00	00	01	01	01	01	01

Syncword

Parameter : 50

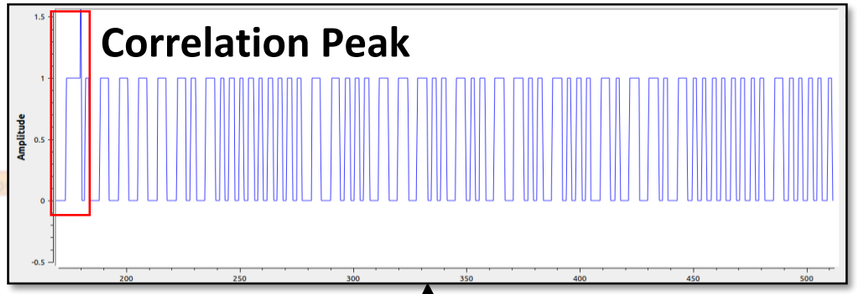
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
F7B0h	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
F7C0h	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
F7D0h	00	00	00	00	00	00	00	00	00	00	01	01	01	01	01	01
F7E0h	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
F7F0h	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01	01
F800h	01	01	01	01	00	00	00	00	00	00	00	00	00	00	00	00
F810h	00	00	01	01	01	01	01	01	01	01	01	01	01	01	01	01
F820h	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

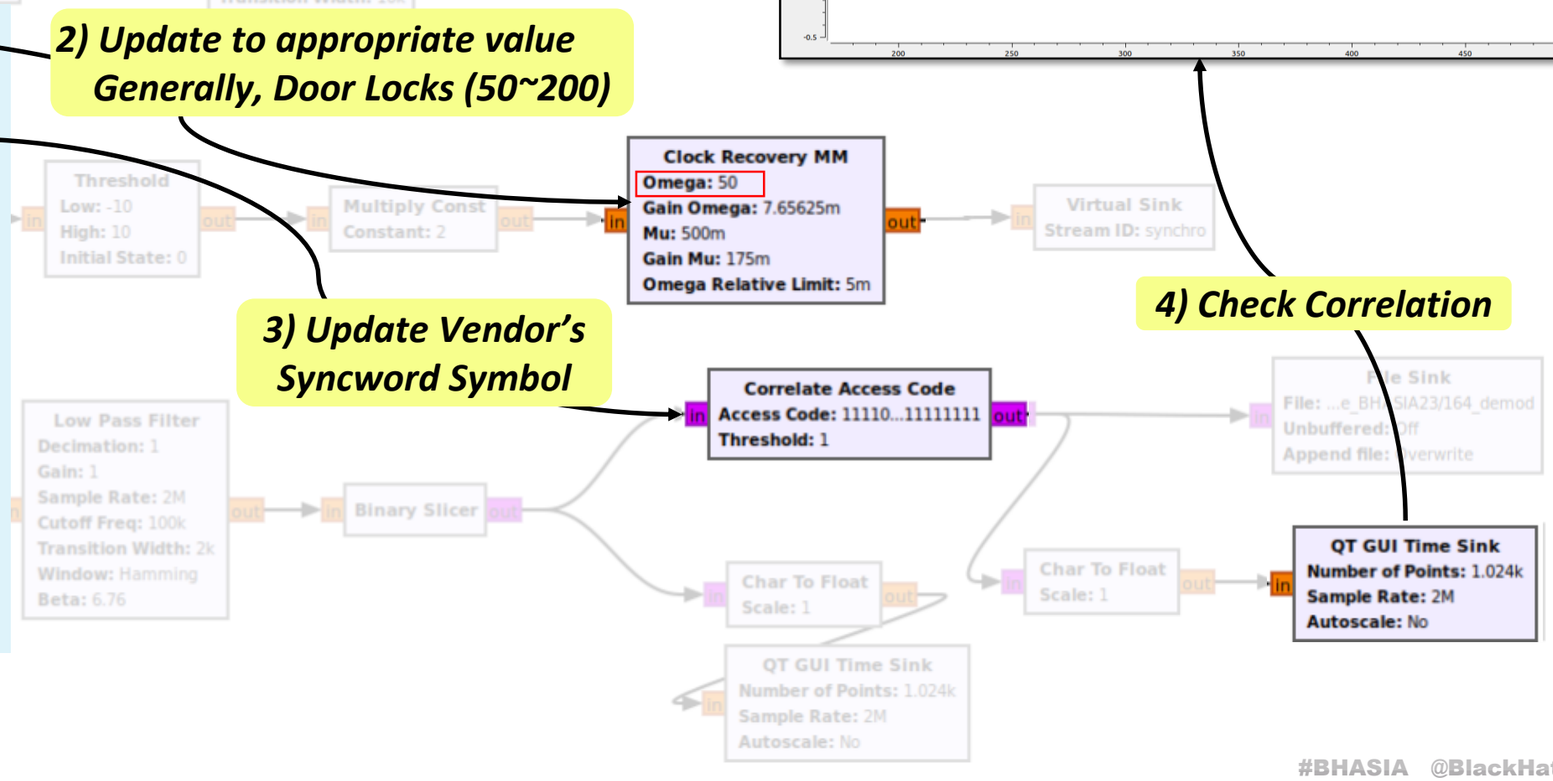
Syncword

2) Update to appropriate value
Generally, Door Locks (50~200)

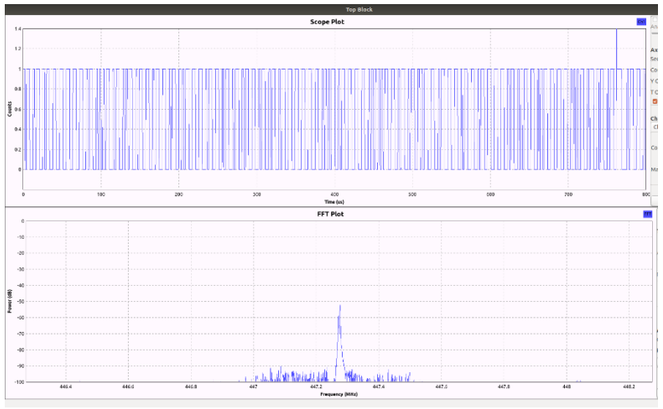
3) Update Vendor's Syncword Symbol

4) Check Correlation





SDR based Code Catcher
Decode and Decrypt



Capture and Demodulation

BIN

```
   1  02  03  04  05  06  07  08  09  0A  0B  0C  0D  0E  0F
00 00 00 00 01 01 01 01 01 01 01 00 00 00 00
00 00 00 00 00 01 01 01 01 01 01 01 01 00 00
00 00 00 00 00 00 01 01 01 01 01 01 01 01 00
00 00 00 01 01 01 01 00 00 00 00 00 00 00 00
00 00 00 00 01 01 01 01 01 01 01 01 01 01 01
01 02 00 00 00 00 00 00 00 00 01 01 01 01 01
01 01 00 00 00 00 00 00 00 00 00 01 01 01 01
01 01 01 00 00 00 00 00 00 00 00 00 01 01 01
01 01 01 01 01 00 00 00 00 00 00 00 00 01 01
01 01 01 01 01 01 00 00 00 00 01 01 01 01 00
00 00 00 01 01 01 01 00 00 00 00 01 01 01 01
00 00 00 00 00 00 00 00 01 01 01 01 01 01 01
01 00 00 00 00 00 00 00 00 01 01 01 01 01 01
01 01 00 00 00 00 00 00 00 00 01 01 01 01 01
```

Demodulated Signal

```
shaftmom@shaftmom-System-Product-Name:~/Desktop/demod_decode_BHASIA23$
./doorloc sniff and decrypt.exe
SNIFF&DEC
================================================================
preamble detected 1
Recived Message :E 18 E0 87 1D E3 36 B5 1C D4 13 B5 BA BE F2 96 78
First Block Key :F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF 00
Second Block Key :D2 18 49 3F 28 87 D3 52 87 2C 21 66 BC AA 37 49

Encrypted Message :18 E0 87 1D E3 36 B5 1C D4 13 B5 BA BE F2 96 78
3::f195a523890100000000000000000d8
Decrypted Message :F1 95 A5 23 89 01 00 00 00 00 00 00 00 00 D8

================================================================
preamble detected 2
Recived Message :E 18 E0 87 1D E3 36 B5 1C D4 13 B5 BA BE F2 96 78
First Block Key :F1 F2 F3 F4 F5 F6 F7 F8 F9 FA FB FC FD FE FF 00
Second Block Key :D2 18 49 3F 28 87 D3 52 87 2C 21 66 BC AA 37 49
```

Decode and Decrypt

Transmitter A

```
preamble detected 7
Recived Message :11 DE 9F E7 33 D6 D8 B2 83 A9 3C 90 B5 7B 71 A1 61
First Block Key :11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F 20
Second Block Key :8A E9 E1 36 4D 45 3E 51 CF 27 ED 82 FB EC 5F AE

Encrypted Message :DF 9F E7 33 D6 D8 B2 83 A9 3C 90 B5 7B 71 A1 61
Decrypted Message :11 95 A4 1B DB 01 00 00 00 00 00 00 00 00 41
```

Sync Counter TXID: 95 A4 1B DB Checksum

Transmitter B

```
preamble detected 16
Recived Message :44 1 1F 73 92 85 42 79 2 3C EB 13 7 79 C4 86 D8
First Block Key :44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51 52 53
Second Block Key :21 D8 FF 33 63 30 71 ED E3 6B 4E D3 29 67 0F 0D

Encrypted Message :01 1F 73 92 85 42 79 02 3C EB 13 07 79 C4 86 D8
Decrypted Message :44 95 A5 28 AE 01 00 00 00 00 00 00 00 00 55
```

Sync Counter TXID: 95 A5 28 AE Checksum

gr block based Tx : Encode and Transmit

Options
ID: top_block
Generate Options: WX GUI

Variable
ID: vco_offset
Value: 850m

Variable
ID: center
Value: 2.21k

Variable
ID: vco_max
Value: 2.5k

Variable
ID: fsk_deviation
Value: 170

Variable
ID: samp_rate
Value: 2M

Variable
ID: frequency
Value: 447.274M

Variable
ID: Interpl
Value: 330

File Source
File: ...ode_BHASIA23/164_mod
Repeat: Yes
Add begin tag: ()

Unpack K Bits
K: 1

Repeat
Interpolation: 330

Throttle
Sample Rate: 2M

Encoded data

Adjust to match the target symbol rate

Message
Encrypt and Encoding

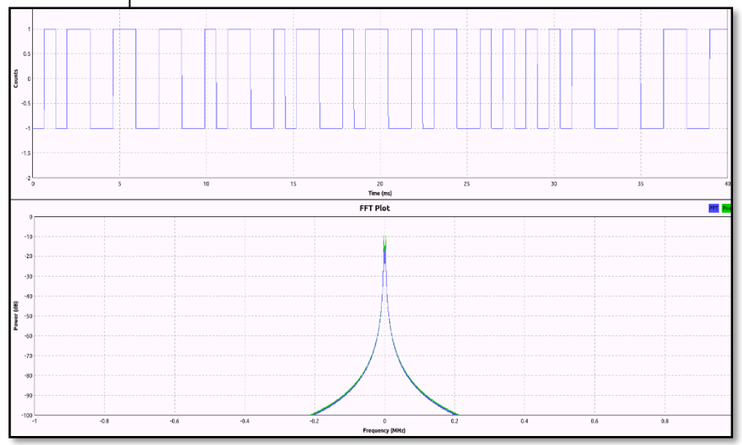
Simple BFSK TX using VCO

Chunks to Symbols
Symbol Table: -1, 1
Dimension: 1

VCO (complex)
Sample Rate: 2M
Sensitivity: 16.708k
Amplitude: 500m

UHD: USRP Sink
Samp Rate (Sps): 2M
Ch0: Center Freq (Hz): ...74M
Ch0: Gain Value: 50
TSB tag name:

WX GUI Scope Sink
Title: Scope Plot
Sample Rate: 2M
T Scale: 5m
Window Size: 720, 500
Trigger Mode: Stripchart
Y Axis Label: Counts



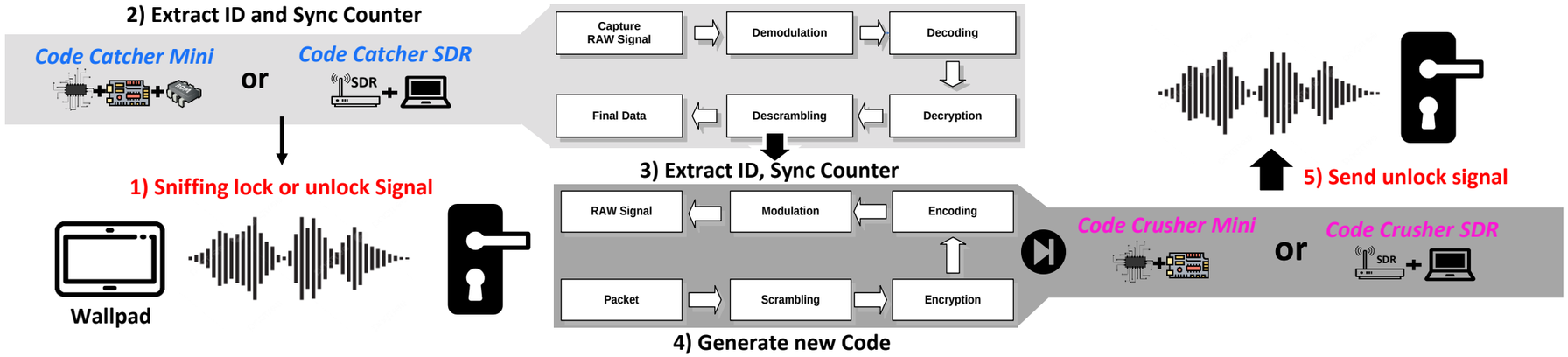
Practical Attack

The Art of RF Lock Picking

Two Types of Lock Picking

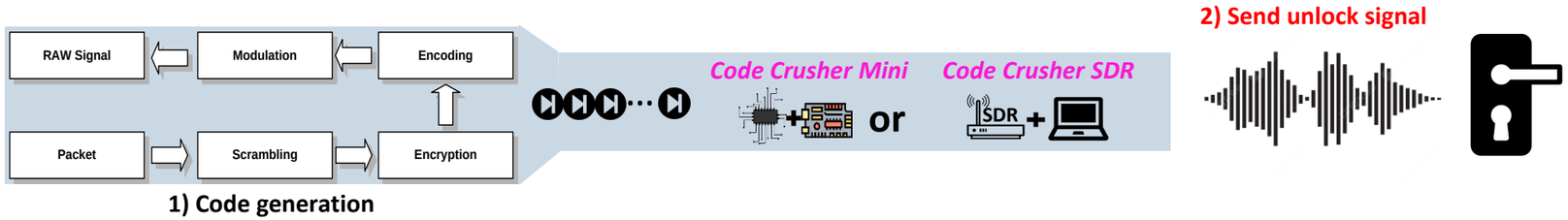
- Type1: Sniff and unlock
  - The attacker extracts the “ID” and “synchronous counter” and generates a new code

※ Attacker know “current synchronous counter” value, and is expected to know the next value

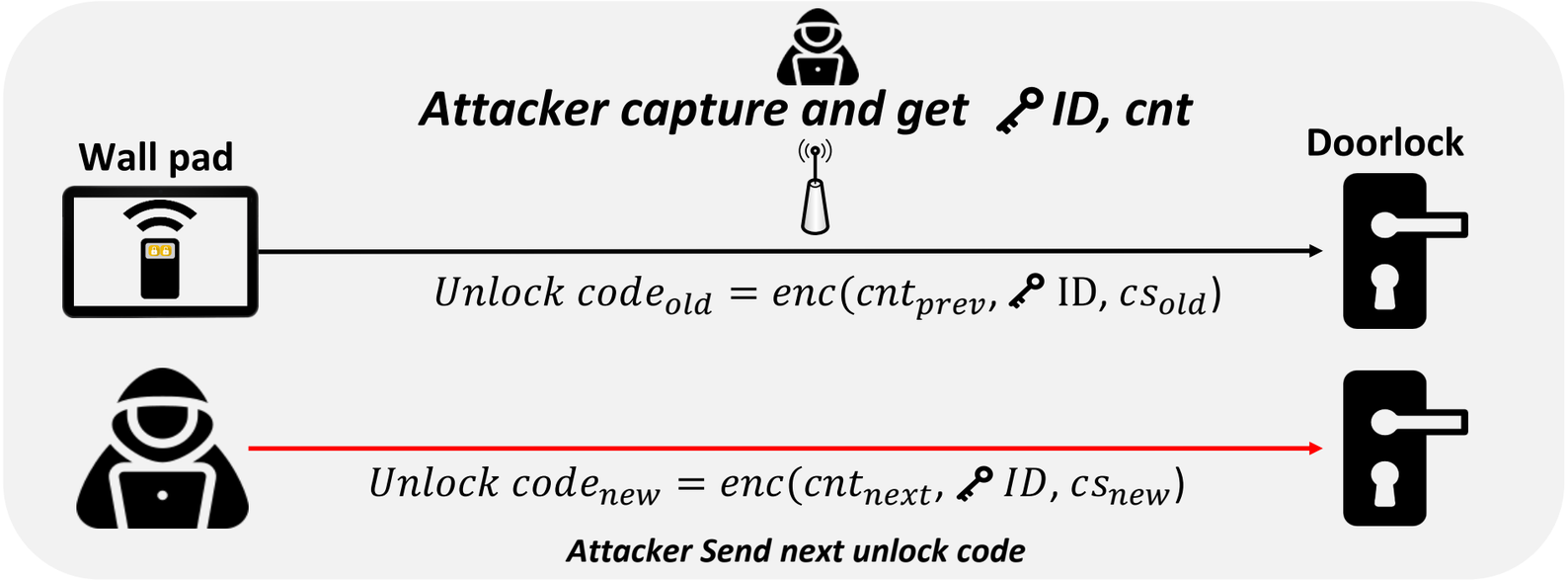


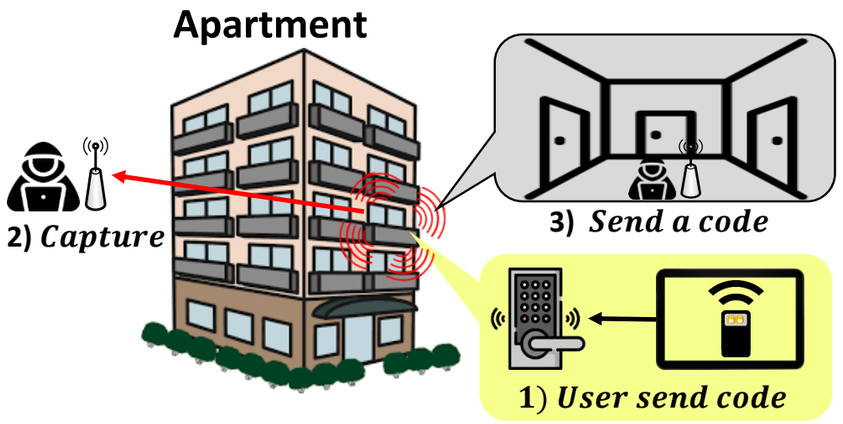
- Type2: Lock picking - Without Sniffing (=Brute Force Attack)
  - The attacker generates a new code by only changing the ID(=Serial Number) value

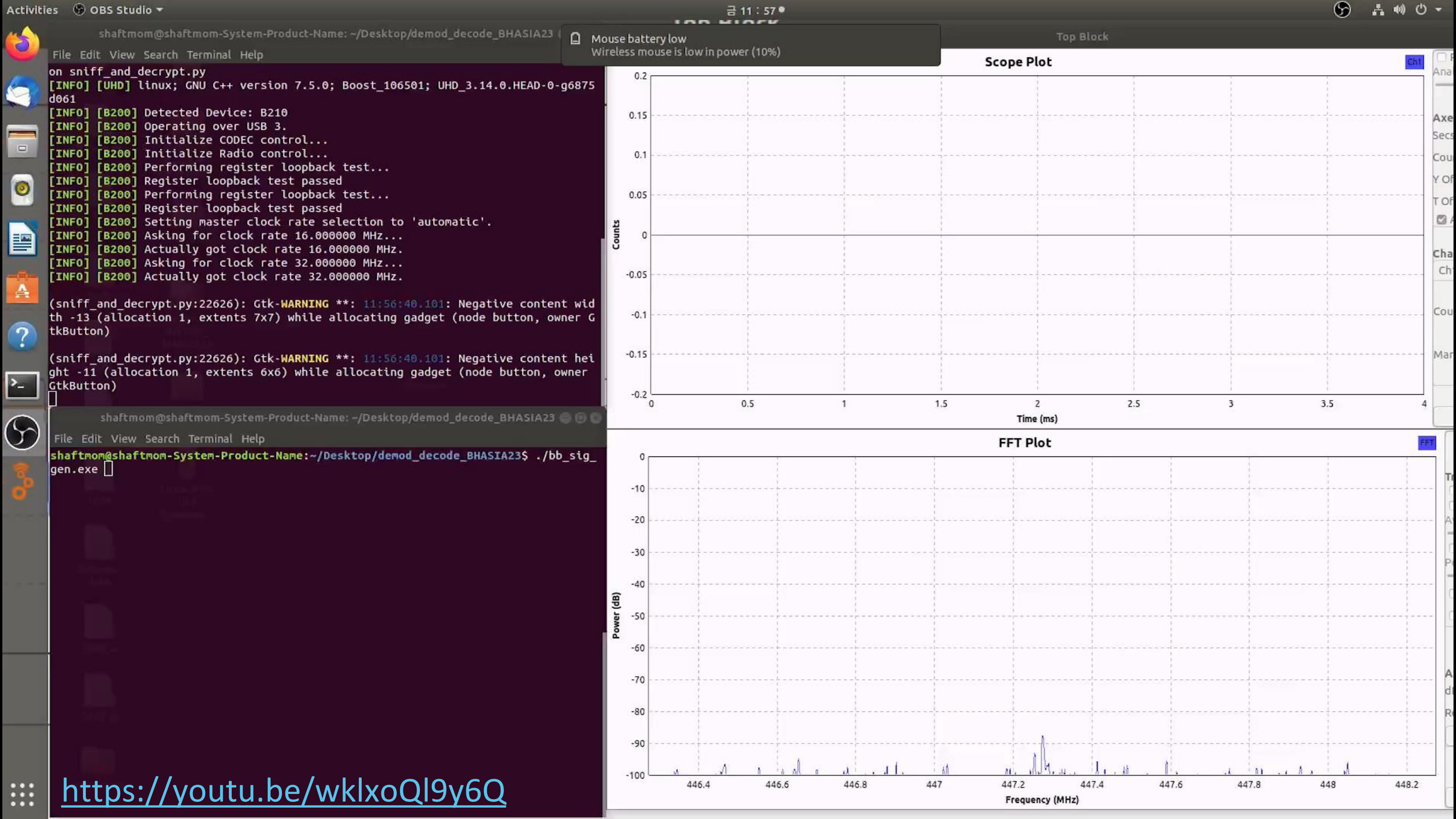
※ Attacker don’t know “current synchronous counter” value



Lock Picking : Sniff and Unlock

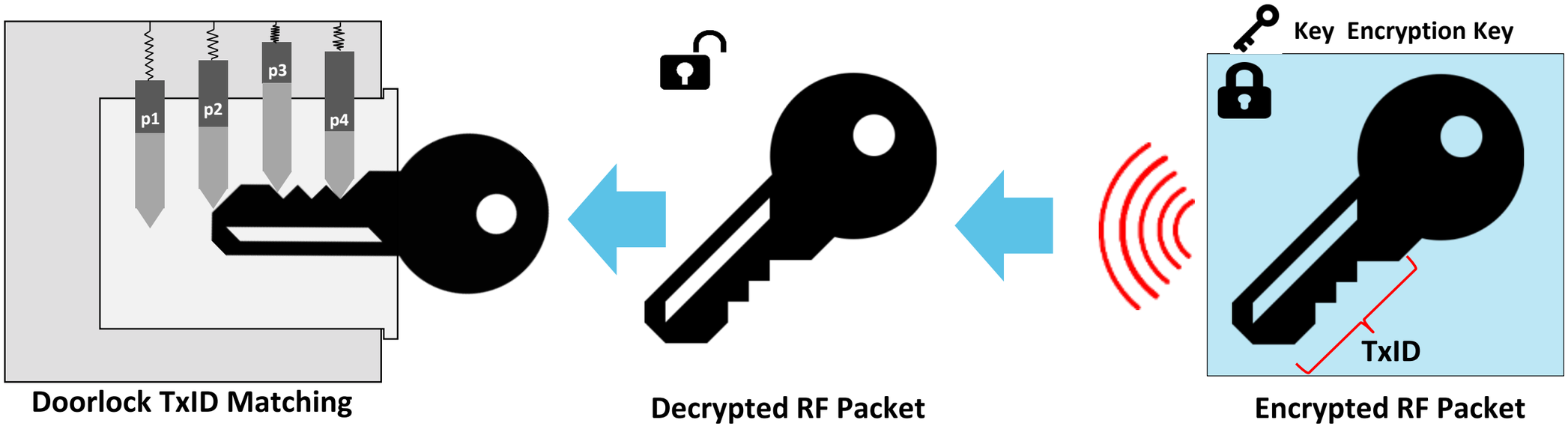




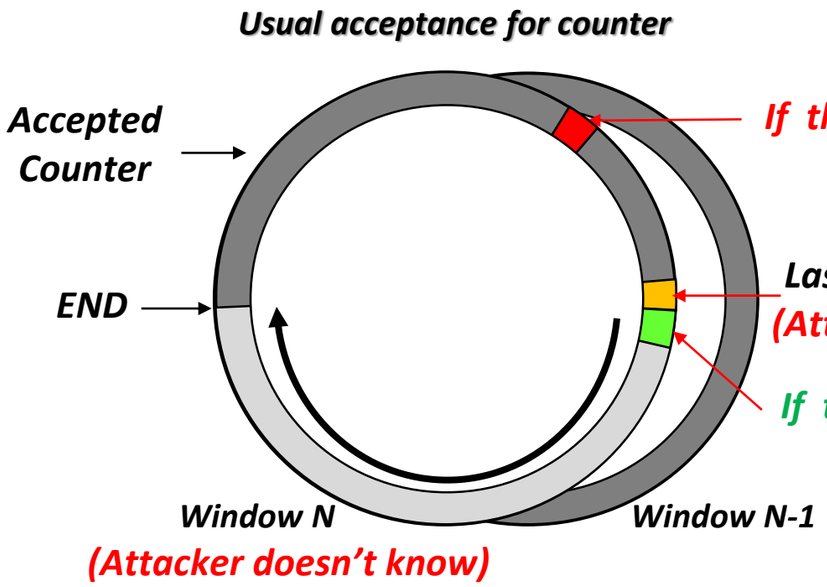

Activities
OBS Studio
금 11 : 57
Mouse battery low
Wireless mouse is low in power (10%)
shaftmom@shaftmom-System-Product-Name: ~/Desktop/demod_decode_BHASIA23
File Edit View Search Terminal Help
on sniff_and_decrypt.py
[INFO] [UHD] linux; GNU C++ version 7.5.0; Boost_106501; UHD_3.14.0.HEAD-0-g6875d061
[INFO] [B200] Detected Device: B210
[INFO] [B200] Operating over USB 3.
[INFO] [B200] Initialize CODEC control...
[INFO] [B200] Initialize Radio control...
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Performing register loopback test...
[INFO] [B200] Register loopback test passed
[INFO] [B200] Setting master clock rate selection to 'automatic'.
[INFO] [B200] Asking for clock rate 16.000000 MHz...
[INFO] [B200] Actually got clock rate 16.000000 MHz.
[INFO] [B200] Asking for clock rate 32.000000 MHz...
[INFO] [B200] Actually got clock rate 32.000000 MHz.
(sniff_and_decrypt.py:22626): Gtk-WARNING **: 11:56:40.101: Negative content width -13 (allocation 1, extents 7x7) while allocating gadget (node button, owner GtkButton)
(sniff_and_decrypt.py:22626): Gtk-WARNING **: 11:56:40.101: Negative content height -11 (allocation 1, extents 6x6) while allocating gadget (node button, owner GtkButton)
shaftmom@shaftmom-System-Product-Name: ~/Desktop/demod_decode_BHASIA23
File Edit View Search Terminal Help
shaftmom@shaftmom-System-Product-Name:~/Desktop/demod_decode_BHASIA23$ ./bb_sig_gen.exe
Top Block
Scope Plot
Ch1
Counts
Time (ms)
FFT Plot
FFT
Power (dB)
Frequency (MHz)
https://youtu.be/wklxoQl9y6Q

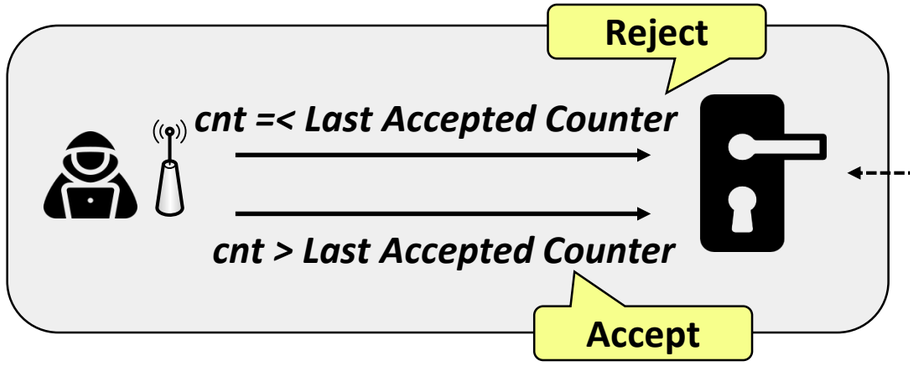
Lock Picking : without sniffing

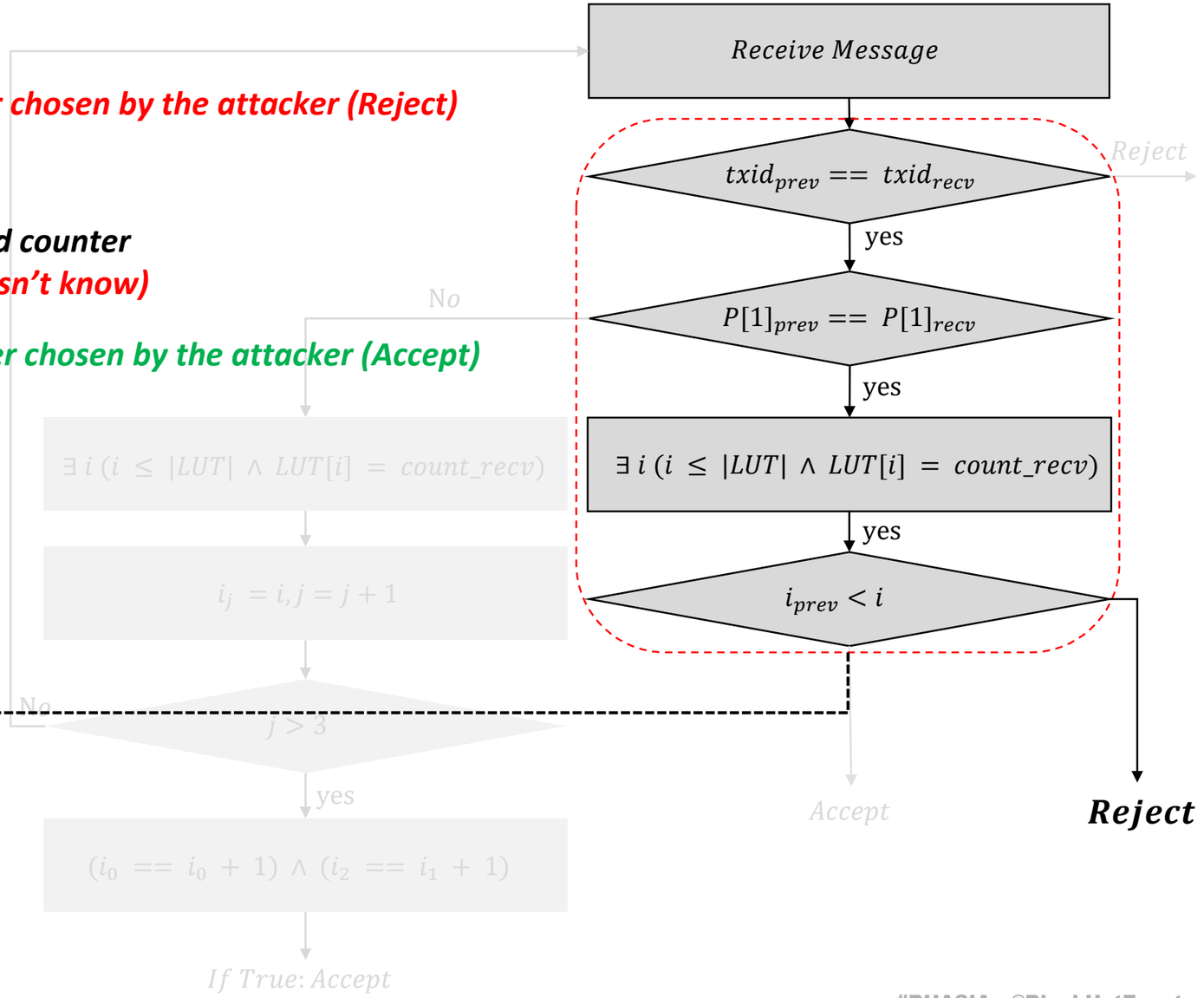
- This attack involves unlocking the victim's door without the need for any RF sniffing
- The attack is to find a Serial Number(=TxID) that matches the one used by the door lock.
- The attacker's message must always be accepted, even if the current counter value is unknown



Considering Attacker's Counter Value Selection

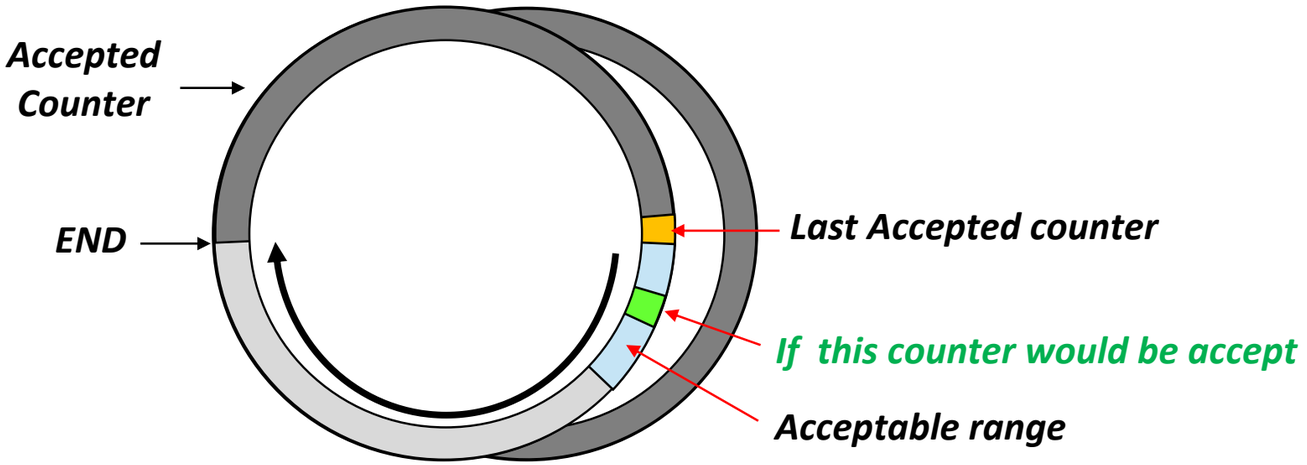


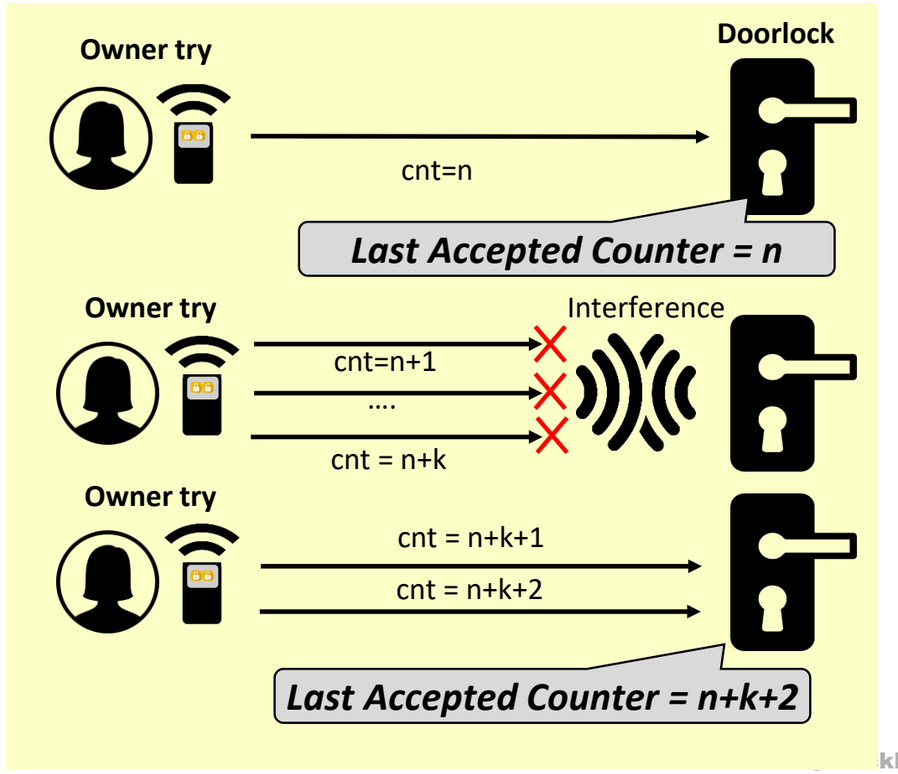




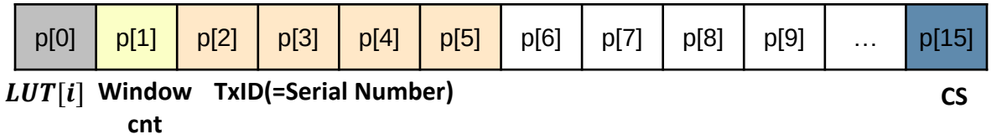
Re-Synchronization and Acceptable Counter Range

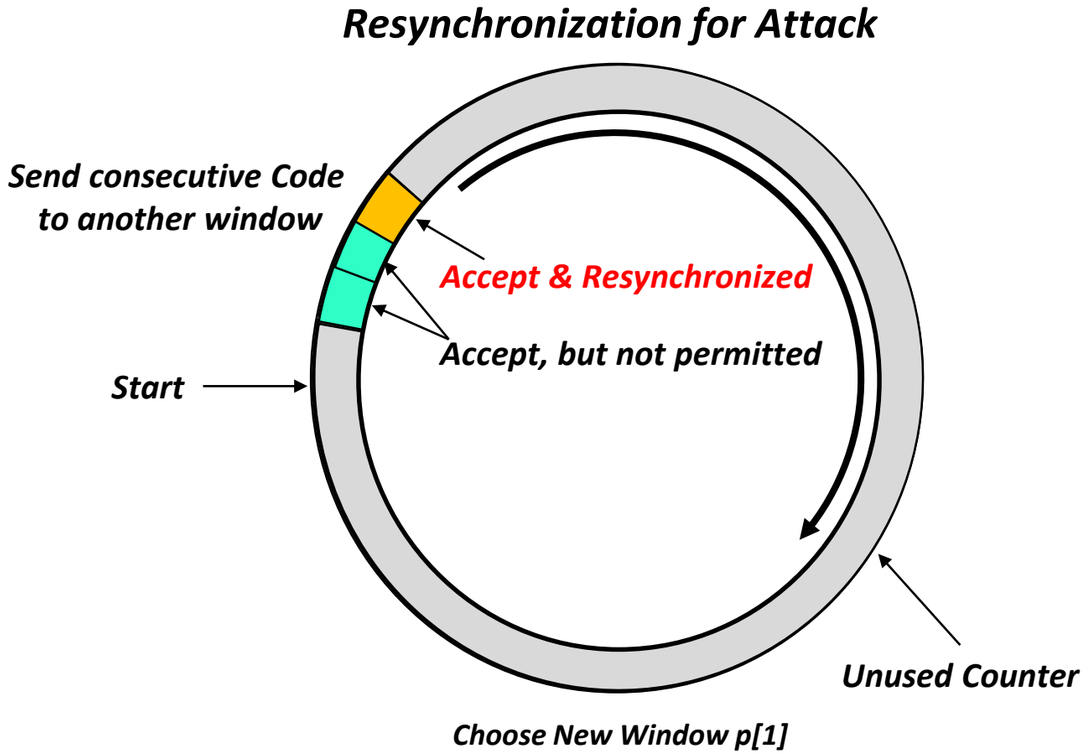
- There is always the possibility that the transmitter has been activated several times outside the receiver's range, the receiver must accept values
- To address this issue, many door locks have a synchronization function that allows them to accept in a specific range of counter values
  - Door locks will not accept a large counter value exceed to specific range



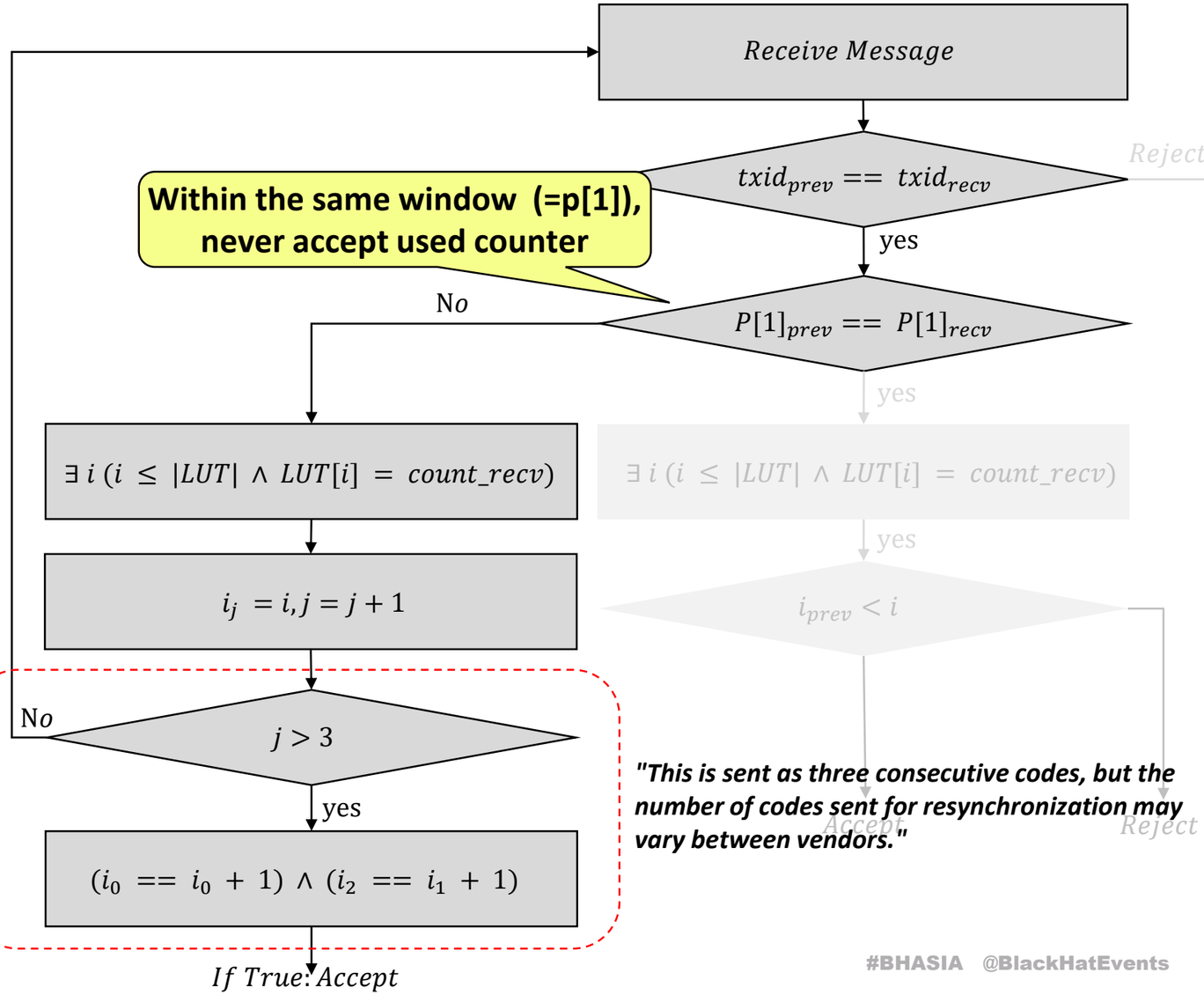


Force Synchronization





Minimizing the transmission packets necessary for resynchronization is crucial factor

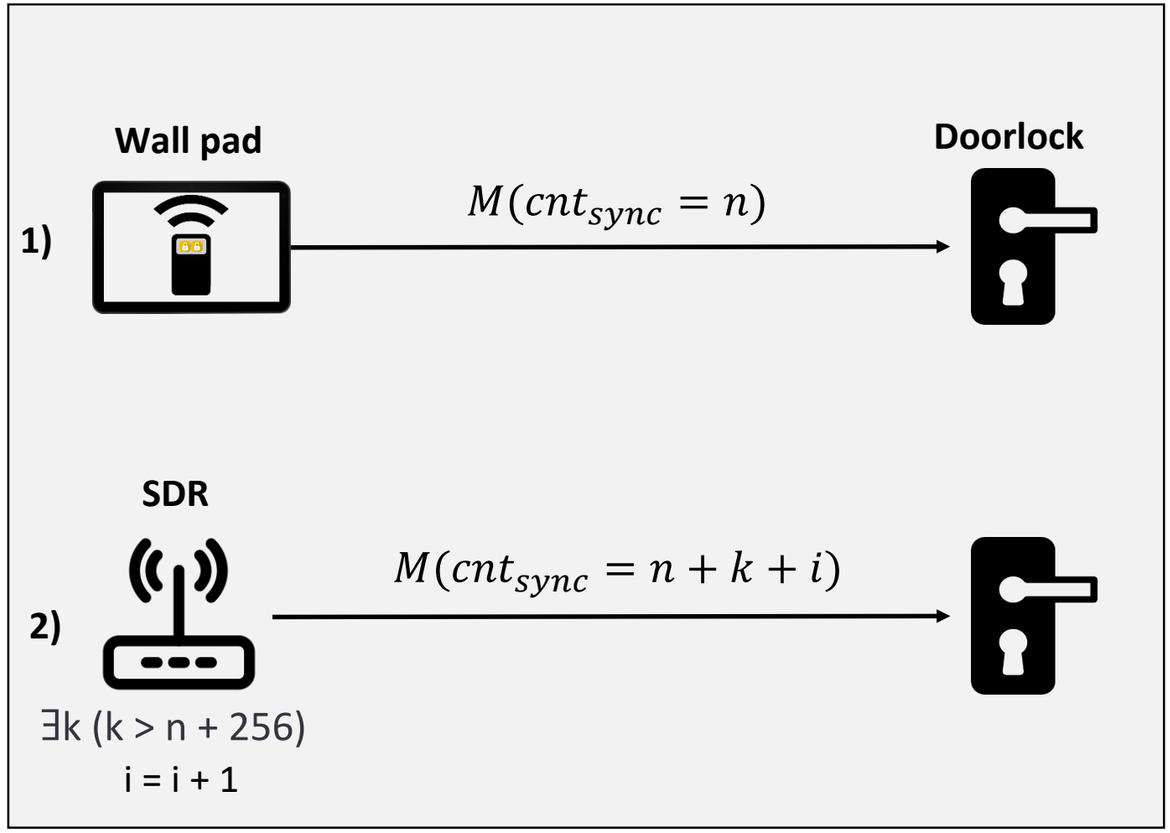


"This is sent as three consecutive codes, but the number of codes sent for resynchronization may vary between vendors."

Packet count for Force Synchronization

- The number of consecutive code transmissions required for force synchronization
  - It is an important factor for increasing overall attack time

Test Method



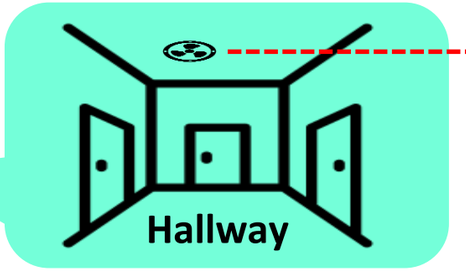
Code Counts for force resynchronization

Vendors	Models	code count for force resync. max(i)
A	A-1	1 or 2
B	B-1	3
	B-2	3
	B-3	3
	B-4	3
C	C-1	2
	C-2	2
	C-3	2
D	D-1	2
	D-2	2
I	I-1	1
J	J-1	1

Lock Picking Attack Scenario

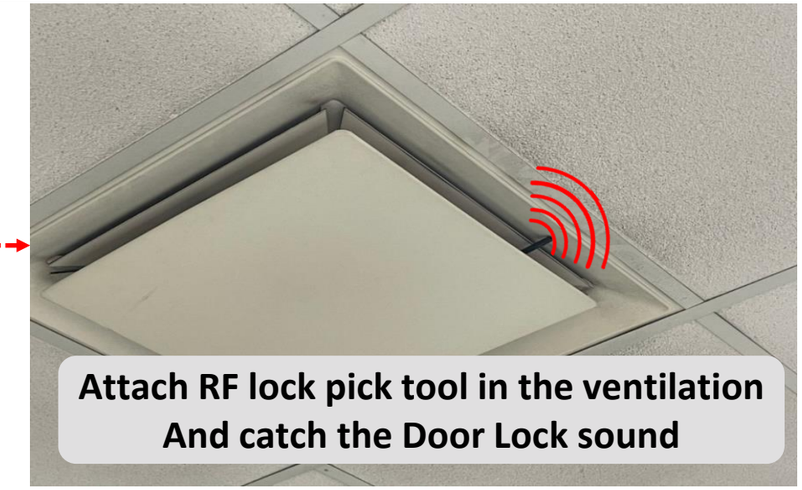
Case : Vendor B *(The products Send Three Consecutive Code for Re-synchronization)*

Attack Scenario



Suffixed at a hidden spot

Deploying



$Message$

$code_1 = enc(cnt_0, window_N, txid, cs)$

$code_2 = enc(cnt_1, window_N, txid, cs)$

$code_3 = enc(cnt_2, window_N, txid, cs)$

⋮

$code_1 = enc(cnt_0, window_N, txid + n, cs)$

$code_2 = enc(cnt_1, window_N, txid + n, cs)$

$code_3 = enc(cnt_2, window_N, txid + n, cs)$

Attacker

Vendor X

Vendor B serial number Range is
{0xBA, 0xA0, 0x00, 0x00}~ {0xBA, 0xAF, 0x0F, 0xFF}
$Number\ of\ Codes: 2\ Bytes = \ 2^{16} = 65536$

Total Tx time in the worst case
=(Number of Codes × Packet count for Force Sync)
/ Transmission Rate (MPS)
*= 65536 * 3 / 5*
= approximately 39321 sec = 10.9 Hours

Activities Terminal 5월 6 16 : 04

shaftmom@shaftmom-System-Product-Name: ~/Desktop/dem...

```
shaftmom@shaftmom-System-Product-Name:~/Desktop/demod_decode_BHASIA23$ ./bb_sig_g
en.exe
```

shaftmom@shaftmom-System-Product-Name: ~/Desktop/dem...

https://youtu.be/cGX5IVFmulo

OBS 25.0.3+dfsg1-2 (linux) - Profile: Untitled - Scenes: Untitled

File Edit View Profile Scene Collection Tools Help

Scenes	Sources	Audio Mixer	Scene Transitions	Controls
Scene	Screen Capture (XS	Desktop Audio 0.0 dB	Fade	Start Streaming
			Duration 300 ms	Stop Recording
				Studio Mode
				Settings
				Exit

LIVE: 00:00:00 REC: 00:00:02 CPU: 6.2%, 30.00 fps

Discussion and Conclusion

Takeaways

❑ Variant Replay Attack

- **“RollJam”** is inevitable without timestamps and **“RollBack”** is also feasible in door lock systems
- Our new variant attack called **"Loop Play Back"** has been confirmed as feasible in door lock systems
- At least in door lock system, the root cause of these attacks is confirmed

❑ Lock Picking Attack

- Easily exploitable by picking the lock w/ sniffing one signal
  - If signal archetype is known, it would be possible to recover the next code with one time
- It may still be vulnerable to open any door lock that's the same model w/o the use of sniffing
  - Depending on the properties of the ID(=Serial#), it can be more practical to carry out this attack
  - Re-synchronization process is also key factor for brute force attack

Lessons Learned

- RF security testing is essential
  - It is an important component of a comprehensive security strategy, especially for systems that rely on wireless
- Security through obscurity is not an answer
  - It's important to design systems that are resilient to attacks even if an attacker knows how they work
- The implementation should be based on elaborate principles and best practices
  - Mutually complementary and interdependent
- Assigning a unique key to each product is a better secure approach in door lock
  - Although it may not be the best ways, I believe that doing it this way is better
  - If an attacker steals a key from one product, it will not affect communication with other products
  - Of course, using secure encryption algorithms is essential for security.

black hat ASIA 2023

Thank you!

If you have any question, please send me email

Kwonyoup Kim
CEO/founder
kkyoup@sntworks.kr

Seungjoon Lee
Senior Researcher
sj.lee@sntworks.kr
shaftmom@gmail.com