# CyMinded Security Report

## **Subject Matter:** CVE-2022-37418

**Note:** this report's structure represents my thought process, going from one example to the main subject.

As you probably notice. The blue fronted sentences are a representation of my thought process while researching and discovering new information… 😊

# CVE-2022-37418

The RKE receiving unit on certain Nissan, Kia, and Hyundai vehicles through 2017 allows remote attackers to perform unlock operations and force a resynchronization after capturing two consecutive valid key fob signals over the radio, aka a **RollBack** attack. The attacker retains the ability to unlock indefinitely.

**Publishing Date:** 24.8.2022

**Main Publishers:** Levente Csikor & Hoon Wei Lim – NCS Group

**Venerable tool:** RKE-Vehicles' Remote Keyless Entry receiving unit

**Affected Products:** Nissan, Kia, Honda

**Versions**: up to, including (<=) 2017

**Attack Vector:** Rollback

**CVSS scores:**

**Base score:** 6.4, **severity**: MEDIUM

Stepping in:

## RKE receiving unit

The **Remote Keyless Entry (RKE) Receiving Unit** is an electronic module in a car responsible for receiving and processing signals from the key fob to lock or unlock the doors remotely.

Over 70% of vehicles today come with an RKE system, either standard or optional. RKE systems consist of a key fob transmitter and a receiver inside the vehicle.

Most used frequencies: 315MHz in the U.S. and Japan, and 433.92MHz in Europe.

### RKE Functionality:

When you press a button on your key fob, the RKE receiving unit "listens" for the signal, decodes the message, and triggers the corresponding function, such as unlocking the doors or opening the trunk.

RKE systems may also include additional functionalities, such as engine start, activating the car's alarm, or other vehicle functions.

**How does it work?**

The key fob sends an encoded radio frequency (RF) signal with a unique binary code that carries the command (e.g., lock or unlock).

*Radio frequency (RF)*

*Range: 20 kHz - 300 GHz*

The RKE receiving unit decodes the signal. It then verifies the binary code sent by the key fob to confirm the legitimacy of the command. This verification often includes matching rolling codes to prevent replay attacks.

If the code is correct and verified, the RKE sends a command to the car's central locking system to perform the requested action, such as unlocking the doors.

**Attack on key management and cryptographic algorithms:**

Key enrollment, Key replacement and Key extraction.

**Manipulation of Key fob signals:**

Signal jamming, Relay and Replay attacks

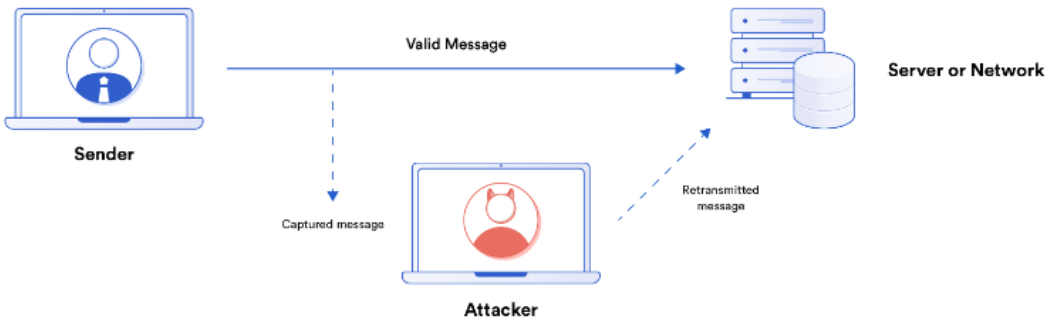
For further introduction info, watch this fun YouTube video:

https://youtu.be/5CsD8I396wo

## Replay attack:

*"A replay attack is when a malicious actor captures and retransmits valid data to achieve fraudulent authentication or execute unauthorized actions in a network."*

*(ChainLink)*



From the definition and image, we can understand the direction the attack aims for:

Capturing the validated "unlocking" action and use it maliciously,

In our case – the ability to capture the car fob's RF signal which communicates with the RKE receiver, therefore gaining control of the vehicle.

## Defending the RKE systems:

## Common Security Measures:

- **Rolling Codes**: To prevent unauthorized access, the RKE uses rolling or hopping codes, changing the code with each button press. Every code has an index that increases each time a signal is received. This ensures that intercepted codes can't be reused to gain entry.

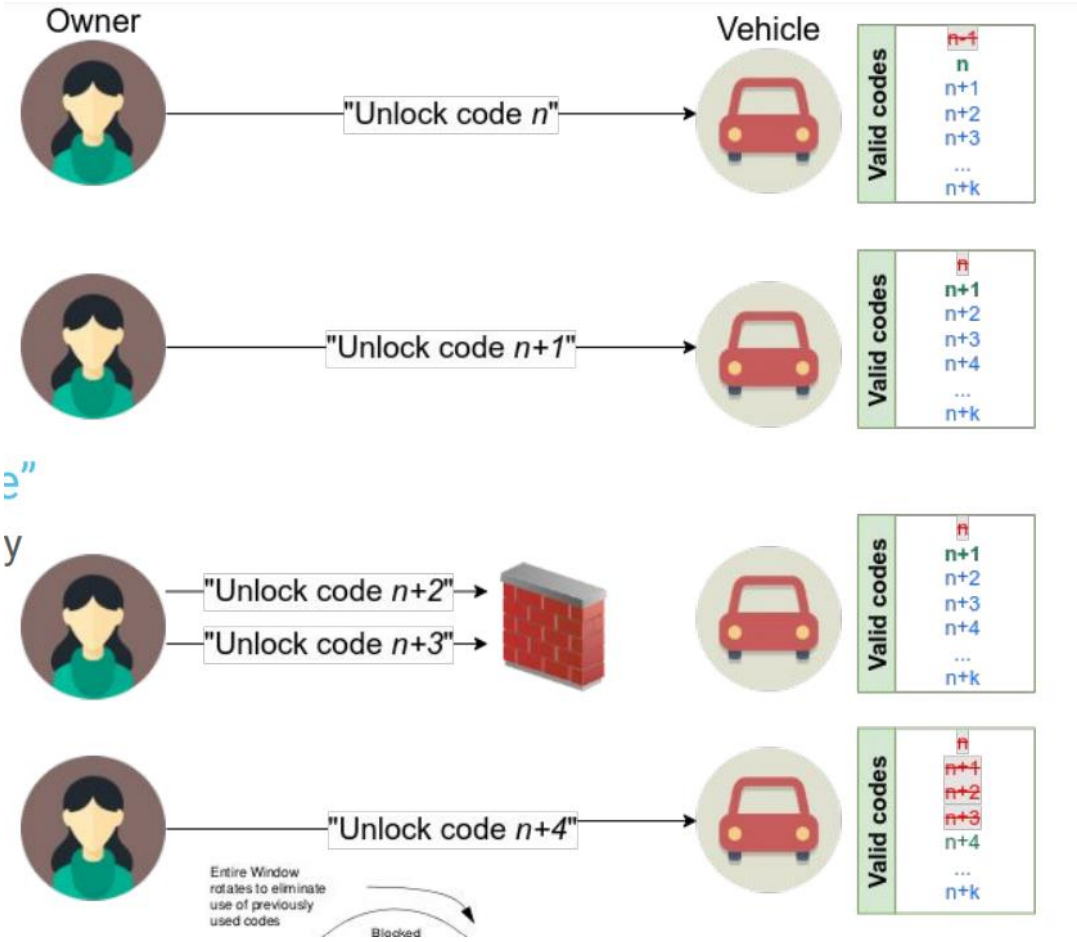
    - Only if counters are in sync upon reception → vehicle acts as instructed/expected.
    - **Note:** no two unlock signals are the same.

- **Counter Resynchronization Issue:** If the car fob is pressed while out of range, or if there are unintentional presses, the fob's code index can get ahead of the car's RKE index, causing a mismatch and preventing the car from unlocking.

  **Solution:** RKE addresses this by resynchronizing the counters: if the system detects that the fob's counter is ahead of the car's, it resets to the car's lower counter value, allowing the fob to continue functioning without lockout.

- **Encryption**: many RKE employ encryption algorithms, such as AES (Advanced Encryption Standard) or HMAC, to protect the code from being easily duplicated or hacked.

- **Signal Timeouts**: the RKE may only accept signals within a limited time window, reducing the chance of interference or tampering.




## Rollback – the new replay attack

And so, we think we solved the replay attack, and that we can rest our case here.

**But here is the next issue:**

Even after an index reset, any missed code from before the reset remains valid and could still be used to unlock the car, introducing the **"RollJam attack"**:

## RollJam attack

**Rolling code + Jamming + Replaying =**

Attackers can use a device to jam and capture 2 key fob signals (unlock \ unlock), allowing them to replay missed signals later to unlock the car.

However, **Levente Csikor**, a researcher at I2R, (*aka publishing researcher*) noted that jamming may not be necessary for some vehicles.

In certain cases, an attacker could simply replay previously captured signals to **'rollback'** the car's counter, causing it to accept old codes again. This vulnerability means the car's RKE's index 'rolls back' and recognizes past signals as valid, even if it already registered them.

**Csikor** also proved that this attack can still succeed months later, with signals captured over 100 days earlier being successfully used to unlock the car.

*See: Mazda RollBack - 100 days later*

In the video he used a standard Lenovo ThinkPad attached to a HackRF software defined radio unit.

On screen, he captured five button-presses of a Kia key fob. The car can be seen responding to all of them. He then played back the first two, which were ignored, but the following three were accepted by the vehicle.

After submitting their research to Black Hat, Csikor's team discovered an additional flaw: any **sequence of button presses**—even if they're just a mix of **lock and unlock signals**—can trigger the "rollback" vulnerability.

In other words, an attacker doesn't need to capture a complex series of codes or only specific signals (like just "lock" or just "unlock"). They only need a sequence of any button presses in order, like **lock-unlock-lock**, to cause the car to roll back its counter and accept these older codes again.

## Affected Cars:

Initial discovery in Aug 2021, the research team teste many cars.

Out of the 20 cars in Csikor's presentation, only six were immune to the RollBack attack.

**Brand-specific results**:

- Mazda, Honda, Kia – all tested were vulnerable
- **Toyota**: all tested were safe.

**Button-press requirements** for successful RollBack varied by brand:

- **Nissan, Kia, Hyundai**: Vulnerable models required **only 2 button-presses**.
- **Honda**: Vulnerable models required **5 button-presses**.

| Car Make | Model | Mfg. date | RKE manufacturer | RollBack (variant) |
|---|---|---|---|---|
| Honda | Model 1 (hybrid) | 2016 | Mfr. 1 - chip 1 | $RollBack_{\otimes}^{\text{Strict}}(5)$ |
| | Model 1 | 2018 | Mfr. 1 - chip 2 | $RollBack_{\otimes}^{\text{Strict}}(5)$ |
| | Model 2 | 2017 | Mfr. 1 - chip 1 | $RollBack_{\otimes}^{\text{Strict}}(5)$ |
| | Model 3 | 2017 | Mrf. 1 - chip 1 | $RollBack_{\otimes}^{\text{Strict}}(5)$ |
| Hyundai | Model 1 | 2015 | Mfr. 2 - chip 1 | $RollBack_{\otimes}^{\text{Loose}}(2)$ |
| | Model 1 | 2012 | Mfr. 1 - chip 3 | NO |
| | Model 2 | 2020 | | NO |
| Kia | Model 1 | 2017 | Mfr. 2 - chip 2 | $RollBack_{\otimes}^{\text{Loose}}(2)$ |
| | Model 1 | 2015 | Mfr. 2 - chip 2 | $RollBack_{\otimes}^{\text{Loose}}(2)$ |
| Mazda | Model 1 | 2018 | Mfr. 1 - chip 4 | $RollBack_{\otimes}^{\text{Strict}}(3)$ |
| | Model 2 | 2018 | Mfr. 1 - chip 5 | $RollBack_{\otimes}^{\text{Strict}}(3)$ |
| | Model 3 | 2020 | Mfr. 1 - chip 4 | $RollBack_{\otimes}^{\text{Strict}}(3)$ |
| | Model 4 | 2019 | Mfr. 1 - chip 4 | $RollBack_{\otimes}^{\text{Strict}}(3)$ |
| | Model 5 | 2018 | Mfr. 1 - chip 5 | $RollBack_{\otimes}^{\text{Strict}}(3)$ |
| Nissan | Model 1 | 2014 | Mfr. 1 - chip 6 | NO |
| | Model 2 | 2009 | Mfr. 3 - chip 1 | $RollBack_{5}^{\text{Strict}}(2)$ |
| | Model 3 | | Mfr. 1 - chip 7 | $RollBack_{8}^{\text{Strict}}(2)$ |
| Toyota | Model 1 | | | NO |
| | Model 2 | | Mfr. 4 - chip 1 | NO |
| | Model 3 | | Mfr. 4 - chip 2 | NO |

## Conclusions of the cars model and RKE chip:

- car type (Petrol vs Hybrid) didn't matter
- Age doesn't matter
- Most of Asian cars tested were affected
- 4 types of RKE manufacturers:
    - All Mfr. 2 & 3 are affected: needing 2 signals only.
    - Most Mfr.1 RKE were affected: Mazda needed 3 signals and Honda 5.
    - Vehicles using Mfr.4 RKE were not affected.

## Conclusion:

In 2022, the root cause remained unknown but appeared to be related to the RKE chip design rather than the car fob itself. Therefore, no explicit mitigation was available.

With that being said, the researchers recommended the use of "timestamps" with the signal's application (explained above).

**Further Recommendation:**

**If you wish to continue digging deeper:**

**I recommend starting with:**

- **RKE chip design**
- **Car fob RF signals**
- **Keeping up with known CVE's websites and Cybersecurity channels.**

*****

## Sources:

CVEdetails.com: https://www.cvedetails.com/cve/CVE-2022-37418/

NIST: https://nvd.nist.gov/vuln/detail/CVE-2022-37418

Blackhat 2022 NCS Group: https://i.blackhat.com/USA-22/Thursday/US-22-Csikor-RollBack-A-New-Time-Agnostic-Replay-Attack.pdf

YouTube, Rollback-Black Hat conference speech:

https://www.youtube.com/watch?v=zihLJbmDG3Q

Analog.com: https://www.analog.com/en/resources/technical-articles/designing-remote-keyless-entry-rke-systems.html

YouTube, Steve Mould: https://youtu.be/5CsD8I396wo

ChainLink: https://chain.link/education-hub/replay-attack

PCMAG: https://www.pcmag.com/news/is-your-car-key-fob-vulnerable-to-this-simple-replay-attack

Levente Csikor: *Mazda RollBack - 100 days later*