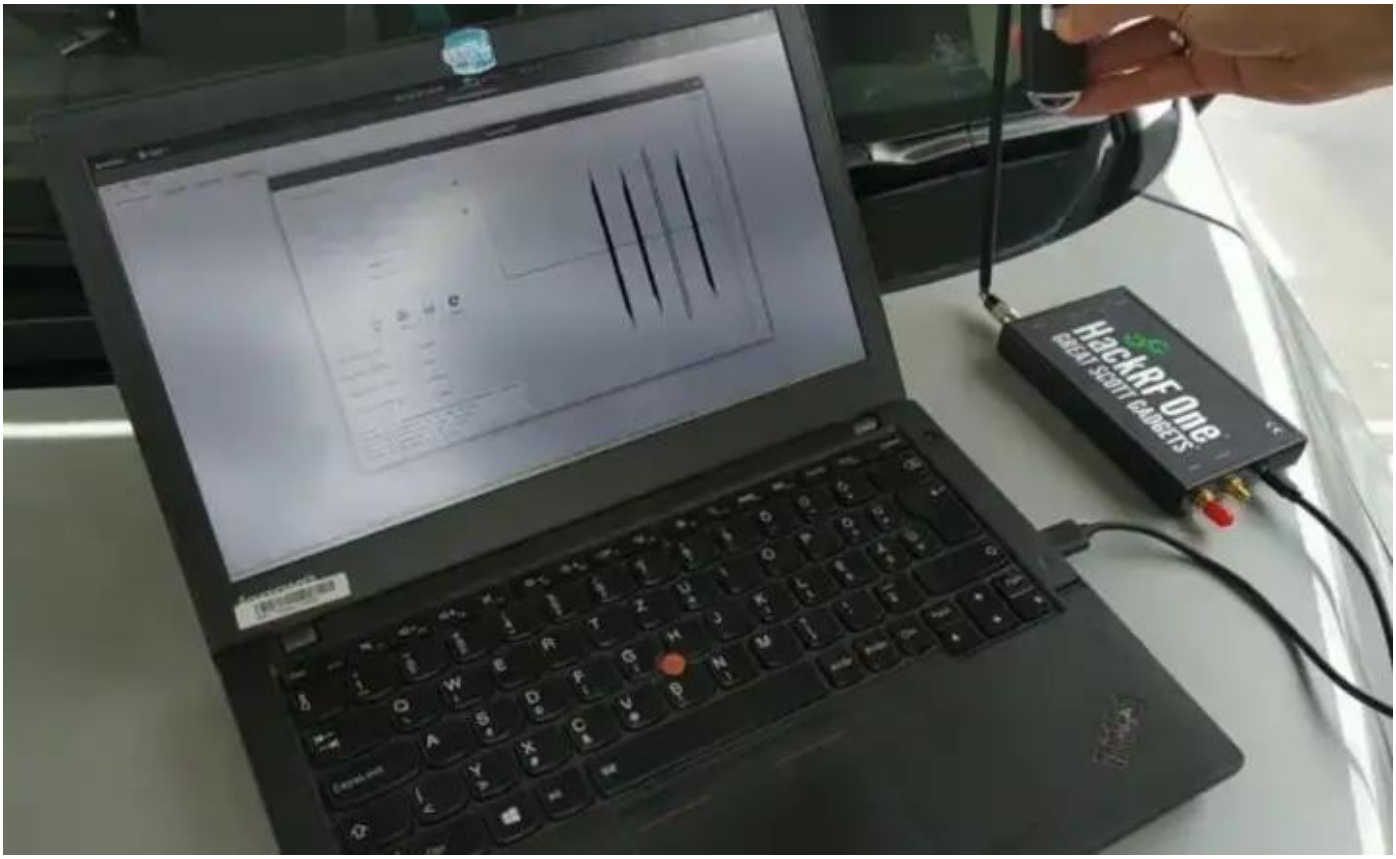
Published in CodeX

cs.lev Follow

Aug 11 · 13 min read · ✨ · Listen

RollBack- A New Time-Agnostic Replay Attack Against the Automotive Remote Keyless Entry Systems

Cyber security researchers from Singapore showed at BlackHat USA 2022 that they can unlock your new car with the simplest key fob signal replay attack you might have ever imagined



Update 23/11/2022: Our talk is available online

Car thefts are still (and probably will always be) on the rise, just as malicious actors being steps ahead of the manufacturers. Last year, there was a considerable spike in auto thefts in Oakville, where 42 luxury cars were stolen within four weeks.

In March, security researchers found a bug in Honda vehicles that allow an attacker to unlock and start your car.

Earlier this year (in April 2022), police became on high alert in the city of Mid Ulster (the UK), where the sixth car theft happened of the year — and it was only April.

A month ago (in July), another new bug targeting Honda vehicles was revealed, called Rolling-PWN attack. Researchers claim that while they tested only on the ten most popular Honda models of the last decade, all Honda vehicles are believed to be affected.

What is the common thing in all these recent attacks? They all fool the vehicles' Remote Keyless Entry (RKE) systems — one of the earliest comfort-enhancing inventions introduced to the industry to make our, and apparently the attackers' lives easier.

Background

The story continues with the scope of this article: RollBack — A New Time-Agnostic Replay Attack Against the Automotive Remote Keyless Entry Systems, or should we call it *RollBack — a yet-another attack targeting our vehicles' remote keyless entry systems?*

As one can observe from the title, RollBack is a replay attack. The most intriguing question is that how can a replay attack work today when rolling codes have been


press unique, effectively preventing simple replay attacks. Put differently, every button click on the key fob triggers a counter in it (and in the vehicle upon reception) to roll, making it valid for subsequent use in the future. Accordingly, sent codes that are used once or unused (i.e., sent but never received by the vehicle) are invalidated by the next code, effectively preventing replay attacks. The "ultimate" protection of rolling code-based systems was believed to be unbreakable until 2015, when Samy Kamkar proposed RollJam at Def Con 2015, a sophisticated attack technique that converts a safety provisioning feature of the rolling code technique into an exploit.

The safety provisioning feature of the rolling code-based systems is that the key fob can be out-of-sync compared to the vehicle's counter. However, being out-of-sync means that the counter in the key fob can *only* be steps ahead of what the car "believes" it will receive next. If you think about it, this is a quite neat (and necessary) feature because it basically allows us to press the key fob button (accidentally) outside the vicinity of the car. When your kid plays with your key fob or your pocket comes to life, these things can quickly happen. And the worst thing you want is to get locked out of your vehicle.

From a technical point of view, the vehicle or more precisely, the key fob signal receiving unit in the vehicle, does not only have one "future code" it awaits for. It has a set of future codes (pre-generated by a Pseudorandom Number Generator using the same seed in both the key fob and the receiving unit) that it will accept and reacts according to the instruction embedded in the signal (e.g., it unlocks). Obviously, if someone would have captured your key fob signals emitted due to accidental button presses but outside of the vicinity of the vehicle, those can be replayed and access to the vehicle is straightforward. However, obtaining such a *"future code"* (unseen by the vehicle) is extremely difficult — you might prank your mom by capturing signals at home while the car is in the garage and cannot receive signals, but getting those signals from a random person seems to be unrealistic. This is (probably the reason) why the system is designed in this way, and this property is *not* considered an exploit/vulnerability — it's a **feature**!

replaying, the victims can be lured into a situation, where these *"future codes"* can be obtained easily (see the figure below).

RollJam is particularly sensitive to timing; it has to be aware of the next valid unused code.

There are two main drawbacks of RollJam:

1. An attacker has to be precise, i.e., the timing is crucial. If the attacker misses her "future codes," i.e., the owner unlocks the vehicle again, the attacker has to restart from scratch.
2. The attack can be launched once. If the attacker wants to open the same car again

researchers in Singapore), when we focused on automotive keyless entry vulnerabilities and tried to experiment with known techniques (including RollJam) and see whether they still apply to our vehicles. We quickly bumped into a Nissan car and discovered that even if it uses rolling codes, **capturing and replaying two consecutive unlock signals** (irrespective of whether the vehicle received them) would unlock the car. See the operation of RollBack in the figure below.

The way RollBack works and how it differs from RollJam.

the name **RollBack** — Rollback is a database management process involving canceling a (set of) transaction(s) to bring the database to its previous state before those particular transactions would have been performed.

Unlike RollJam, **RollBack is time-agnostic.** As you might have read between the lines above, it does not matter whether the vehicle receives those signals. In short, RollBack has the following **two crucial advantages** (compared to RollJam)

1. Once an attacker captures the signals, RollBack can be launched **at any time in the future**, no matter how many times the owner uses the key fob and the car.
2. Even if RollBack was successfully launched, it could be relaunched **as many times as desired** without redoing anything from scratch. In short, you capture two signals once and have indefinite access to that vehicle.

However, while RollJam practically breaks all of today's rolling code-based systems, RollBack is only successful for 70% of them (so far).

RollBack was presented at BlackHat USA 2022.

Different RollBack variants

When we first discovered the vulnerability, we tested a pretty outdated vehicle. In this case, RollBack had the following properties.

First, we identified *how many signals* we needed to replay; this number turned out to be **only two**; however, as we will show, other vulnerable systems might require more than


The second observation we had is that the attacker strictly has to replay the signals in the same strictly consecutive sequence, i.e., she cannot miss any signals in between. On the other hand, we have found vulnerable vehicles that do not even care about this. Any two signals captured in the past and replayed work. Hence, we call the second property **SEQUENCE** and it can be *Strict* (like in case of the Nissan mentioned before), or *Loose* if it is not required, i.e., when replaying signals in the capturing (i.e., ascending) order but there were further valid and forfeited signals in between.

The *third property* is called **TIMEFRAME** and indicates the maximum number of seconds that can elapse between (two) signals when replayed. Some vehicles impose restrictions regarding this; some do not care. Once the signals are captured, replaying them within a given time frame (i.e., trimming "empty noises" in between) is straightforward using applications like Universal Radio Hacker.

Different variants of RollBack derived from our analysis.

RollBack is instruction-agnostic

If you thought that RollBack is already particularly alarming and launching it is a piece of cake, we have something more to share. *Do you still recall why jamming was necessary in the RollBack's explanation figure above?* It is required because this is how an attacker can easily capture consecutive unlock signals by forcing the victim to press the unlock button once more. Although, if the victim is not alone or s/he has to put something on the backseat or in the trunk, s/he has to press the unlock button twice — otherwise,


the instruction in the signals is “lock” or “unlock.” We have seen that the vulnerable vehicles (identified so far) are agnostic to the instruction. In particular, replaying any (sequential) old signals can be used to rollback the rolling code system, and the instruction in the last signal will take place.

Accordingly, if an attacker captures your “lock” signal when you leave your car in the parking lot, then waits for you to come back and capture your “unlock” signal, then you might be already in a trouble.

Even if the vehicle is vulnerable to a RollBack variant that would need three (or more) signals, obtaining them is still easy due to a typical human factor. This human factor is that we usually “lock the vehicle twice” to confirm it. The first lock is usually a “silent lock”, while pressing the “lock” button for the second time triggers the vehicle to honk and flash the emergency lights to notify you (from distance) about the adequate locking — the unlocking part of the story remains the same (and you might press the unlock button twice as well to unlock other than the driver’s door) and the attacker can have three (or more) consecutive signals easily.

Furthermore, some vehicle owners (due to the lack of any “find my car” feature) uses the same thing to locate the vehicle *per se*. Does it sound familiar? When you don’t remember where exactly you left your vehicle in a huge parking lot? You just press the lock button many times, and look for the honking/flashing vehicle from distance :) And, you are still “safe” (not from RollBack obviously) because you don’t unlock your vehicle (for someone who is closer than you) — you just make it easier for yourself to find your own vehicle.

Is my car vulnerable?

Before jumping into the most interesting questions of “Am I affected? Is my vehicle vulnerable”, let us make a disclaimer.


> *close vicinity. All of the captured signals (for the tests) had been stored temporarily only; after capturing the signals and replaying them, the data had been removed permanently immediately. Except two cases when we evaluated the time-agnostic feature of RollBack — we permanently deleted those signals afterward.*
>
> *Note, furthermore, replaying key fob signals do not cause any harm to the vehicle, the key fob, and the whole electronic ecosystem irrespective of being vulnerable to RollBack. Thus, the tested vehicles continue to work and behave as usual. You might temporarily get your key fob to be banned (due to some anti-theft measure), but such key fob resyncronization can be easily done by following a couple of steps, like when you replaced the battery in the key fob.*
>
> *The talk at BlackHat USA 2022 is* ***the first publicly disseminated information about our findings and about RollBack*** *in general. And, this blog post is probably the second one, released only after Black Hat USA 2022. We used our presentation's shorter and more condensed versions during our attempts in initiating disclosure processes with RKE chip manufacturers and Auto-ISAC members.*

We could examine a limited number of vehicles. In particular, for the time being, we could examine several popular Asian vehicle makes and models available in Singapore. The vehicles examined and their relevant data are detailed in the table below. *Model* means the actual model (intentionally obfuscated for now), while the *Mfg. date* denotes the actual manufacturing date of the vehicle we tested. Such information were obtained by using the vehicles' identifier, i.e., their VIN numbers, and publicly available services.

Furthermore, we could also obtain the exact RKE manufacturer and chip version and serial number most of the times by manually disassembling the key fobs, however, for now, they are also purposely obfuscated. When disassembling the key fob was either infeasible or the the chips on the PCB were obscured (e.g., via black paint), we tried to gather manufacturer information by keying in its FCC ID or looking for spare key fobs on different retailers' sites. If we could not obtain the RKE manufacturer by any of the above-mentioned ways, we left the corresponding cells intentionally blank.

Vehicles' details used for our in-house experiments. For the Toyota vehicles, the VIN numbers were not available, hence we left those cells intentionally blank. Moreover, for some vehicles, we could also not identify the RKE system manufacturer; hence, corresponding cells were also left intentionally blank.

From our experiment, which we continuously update, we can conclude the following.

1. The **age of the vehicle does not matter.** Newer cars can be just as vulnerable as older ones.
2. **Drive-train (hybrid vs. petrol) does not matter**. Even though hybrid, or full electric cars use more software (and hence, counter measures might be easier to be introduced), the RKE system itself seems to be independent.
3. Most of the **popular Asian cars tested are vulnerable**. For instance, all tested **Mazda, Honda, and Kia vehicles found vulnerable**. However, all tested **Toyota vehicles are not susceptible to RollBack.**
4. All vehicles using a **key fob made by Mfr.2. and Mfr.3. are affected** and they both **need two signals only**.


6. Vehicles having **key fobs from Mfr.4. seems to be safe.**

Although not the key fobs have the flaw but (probably) the receiving unit (typically manufactured by other OEMs), we observe a correlation; hence, we emphasized/categorized our results accordingly.

RollBack “in the wild”

We have recorded several videos to showcase our attack, RollBack. If you are a Hungarian speaker, you might find a subtitle as well ;)

In the first video, we show *how it works* in general.

Below, we show (for two different vehicles being vulnerable to two different RollBack variants) the *time-agnostic feature of RollBack*. Put differently, we show that the captured

The same *time-agnostic* feature shown on a different vehicle.

Responsible disclosure, root cause, mitigation

We reached out to key fob manufacturers in April 2022 and Auto-ISAC in May 2022. While the vulnerability (and the talk at BlackHat) was acknowledged, we could not advance yet towards finding the root cause and, accordingly, provide mitigation. We could also not provide any comments we received so far from some of the affected car manufacturers. We are looking forward to doing collaborations in the future for the same reason.

More alarming is that while precautionary measures can be taken against attacks relying on jamming (e.g., paying attention to signal reception and the car's reaction to the intentional key fob presses), RollBack is a passive listener during the reconnaissance phase. You cannot observe/confirm whether you are the victim of RollBack. And since it is time-agnostic, looking into your mirror to see if someone


Any CVE?

Sure, we did submit three CVEs, mostly based on the number of signals required. The CVEs are as follows:

CVE-2022–36945
CVE-2022–37305
CVE-2022–37418

TL;DR — 3 takeaways

1. Cyber security researchers from Singapore have shown that by *capturing and replaying a couple of key fob signals* (irrespective of their *instructions*) re-synchronizes the rolling codes and *unlocks most of today's modern (Asian) vehicles* (they tested).
2. RollBack attack *does not require jamming* at all. Furthermore, *RollBack is time-agnostic*; signals need be captured once and can replayed *at any time* and *as many times as desired*.
3. So far, *root cause is not confirmed* and *no explicit mitigation exists*. Adding timestamps to the signals (and checking them on reception) might help.

Stayed tuned, because in the next episode we hopefully disclose more information about the vehicles tested, the key fobs used, and compare our attack to a recent discovery, called Rolling-PWN. A whitepaper, summarizing everything you have to know about RollBack, will also be released soon.


effectiveness of RollJam. It is the base of our research and it probably still breaks most of today's rolling code-based systems. If Samy Kamkar did not show this in 2015 and we did not try it six years later, RollBack might not even exist. We appreciate the cyber security researchers' findings and disclosures, and we believe this is the only way how we can evolve and prevent any future threat before it goes viral.

Reach out to us

Levente Csikor (Institute for Infocomm Research (I2R), A*STAR, Singapore — I was with NCS Group, Singapore when this work started): levente.csikor[at]gmail[dot]com

Hoon Wei Lim (NCS Group, Singapore): hoonwei.lim[at]ncs[dot]com[dot]sg

Jun Wen Wong (DSBJ Pte. Ltd. — was with NCS Group, Singapore during this work)

Soundarya Ramesh (National University of Singapore)

Rohini Poolat Parameswarath (National University of Singapore)

Chan Mun Choon (National University of Singapore)

UPDATE

Online articles sharing the findings of RollBack:

SecurityWeek — Black Hat 2022: Ten Presentations Worth Your Time and Attention (RollBack is #1)

PCMag — Is Your Car Key Fob Vulnerable to This Simple Replay Attack?

PCMag — The 14 Scariest Things We Saw at Black Hat 2022 (RollBack is #11)

TechNewsCrypt — RollBack Breaks Into Your Car

InverseZone — Is Your Car Key Fob Vulnerable to This Simple Replay Attack?

SECRSS (Chinese)— BlackHat 2022|The most interesting topic at BlackHat 2022 (RollBack is #1)

IT MediaNews (Japanese) — 車のキーフォブを1回ハッキングすると、“いつでも何度でも”ロック解除できる技術 70%以上の車両で成功

What’s Now (French) — RollBack s’introduit dans votre voiture