# Lock it and still lose it

Garcia, Flavio; Oswald, David; Kasper, Timo; Pavlides, Pierre

*DOI:*
10.5555/3241094.3241166

*Document Version*
Peer reviewed version

*Citation for published version (Harvard):*
Garcia, F, Oswald, D, Kasper, T & Pavlides, P 2016, Lock it and still lose it: on the (in)security of automotive remote keyless entry systems. in *Proceedings of the 25th USENIX Security Symposium* . USENIX Association, pp. 929-944, 25th USENIX Security Symposium , Austin, Texas, United States, 10/08/16. https://doi.org/10.5555/3241094.3241166

# Lock It and Still Lose It – On the (In)Security of Automotive Remote Keyless Entry Systems

Flavio D. Garcia[1]
*School of Computer Science, University of Birmingham, UK.*
*f.garcia@bham.ac.uk*

David Oswald[2]
*School of Computer Science, University of Birmingham, UK.*
*d.f.oswald@bham.ac.uk*

Timo Kasper[2]
*Kasper & Oswald GmbH, Germany.*
*info@kasper-oswald.de*

Pierre Pavlidès[1]
*School of Computer Science, University of Birmingham, UK.*
*pierre@pavlides.fr*

## Abstract

While most automotive immobilizer systems have been shown to be insecure in the last few years, the security of remote keyless entry systems (to lock and unlock a car) based on rolling codes has received less attention. In this paper, we close this gap and present vulnerabilities in keyless entry schemes used by major manufacturers. In our first case study, we show that the security of the keyless entry systems of most VW Group vehicles manufactured between 1995 and today relies on a few, global master keys. We show that by recovering the cryptographic algorithms and keys from electronic control units, an adversary is able to clone a VW Group remote control and gain unauthorized access to a vehicle by eavesdropping a single signal sent by the original remote. Secondly, we describe the Hitag2 rolling code scheme (used in vehicles made by Alfa Romeo, Chevrolet, Peugeot, Lancia, Opel, Renault, and Ford among others) in full detail. We present a novel correlation-based attack on Hitag2, which allows recovery of the cryptographic key and thus cloning of the remote control with four to eight rolling codes and a few minutes of computation on a laptop. Our findings affect millions of vehicles worldwide and could explain unsolved insurance cases of theft from allegedly locked vehicles.

## 1 Car Keys

For several decades, car keys have been used to physically secure vehicles. Initially, simple mechanical keys were introduced to open the doors, unlock the steering, and operate the ignition lock to start the engine. Given physical access to a mechanical key, or at hand of a detailed photograph, it is possible to create a duplicate. In addition, mechanical tumbler locks and disc locks are known to be vulnerable to techniques such as lock-picking and bumping that allow to operate a lock without the respective key. Finally, for most types of car locks, locksmith tools exist that allow to decode the lock and create a matching key.

### 1.1 Electronics in a Car Key

With electronic accessories becoming available, additional features were integrated into the locking and starting systems of cars: some of them to improve the comfort, others to increase security. On the side of the car key, this implies some electronic circuitry integrated in its plastic shell, as illustrated in Figure 1. Note that the link between Remote Keyless Entry (RKE) and immobilizer is optional. In the Hitag2 system (Section 4), the immobilizer interface can be used to re-synchronize the counter used for RKE, while VW Group vehicles (Section 3) completely separate RKE and immobilizer. In vehicles with Passive Keyless Entry and Start (PKES) (Section 1.1.2), the low-frequency immobilizer link is used to trigger the transmission of a door opening signal over the high-frequency RKE interface.

#### 1.1.1 Immobilizer Transponders

One of the most notable events in the history of car security was the introduction of the immobilizer, which significantly reduced the number of stolen cars and so-called joyrides conducted by teenagers. An electronic immobilizer improves the security of the car key with respect to starting the engine. Technically, most immobilizers rely on Radio Frequency IDentification (RFID) technology: An RFID transponder is embedded in the plastic shell of the car key and contains a secret that is required to

[1]These authors contributed the research on Hitag2.
[2]These authors contributed the research on VW Group.

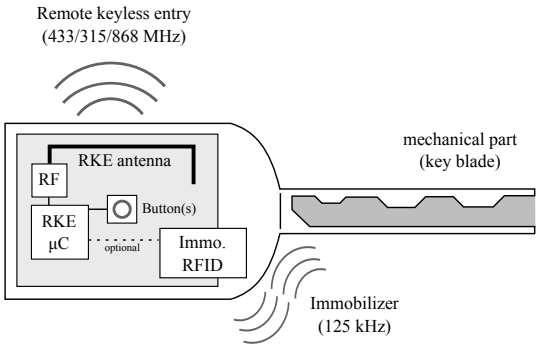


Figure 1: Main components of a car key: RKE and immobilizer systems are separate and use different RF frequencies.

switch on the ignition and start the engine. An antenna coil around the ignition lock establishes a bidirectional communication link and provides the energy for the transponder in order to verify its authenticity with a range of a few centimeters. All modern immobilizers use cryptography for authentication between transponder and vehicle, typically based on a challenge-response protocol.

For many years, only weak, proprietary cryptography was implemented in immobilizer transponders worldwide. This may have been caused by the limited energy available on RFID-powered devices, technological limitations, and cost considerations. The first type of immobilizer transponder to be broken was the widespread DST40 cipher used in Texas Instrument's Digital Signature Transponder (DST), which was reverse-engineered and broken at Usenix Security 2005 [8]: The 40-bit secret key of the cipher can be revealed in a short time by means of exhaustive search. This paper was at the same time one of the first published attacks on a commercial device in the literature. A few years later, at Usenix Security 2012, researchers published several cryptanalytic attacks on NXP's Hitag2 transponders [30, 32], the most widely used car immobilizer at that time. The authors showed that an attacker can obtain the 48-bit secret key required to bypass the electronic protection in less than 360 seconds. One year later, in a paper submitted to Usenix Security 2013 (and finally published in 2015), the security mechanism of the Megamos Crypto transponder were found to be vulnerable to cryptanalytic attacks [31, 33]. The 96-bit secret key of the cipher is mapped into a 57-bit state of a stream cipher that can be rolled back. A flawed key generation (multiple bits of the secret key are set to zero) additionally found in various transponders decreases the attack time from the order of days to a few seconds using a Time-Memory Tradeoff (TMTO).

As a result, the majority of RFID immobilizers used in today's vehicles can be cloned: the secret of the transponder can be obtained by an adversary to circumvent the added security provided by the immobilizer. The cryptography of these immobilizers has to be considered broken as their added protection to prevent criminals from starting the engine of a car is very weak.

### 1.1.2 Passive Keyless Entry and Start

Today, certain modern cars (especially made by luxury brands) are equipped with PKES systems that rely on a bidirectional challenge-response scheme, with a small operating range of about one meter: When in proximity of the vehicle, the car key generates a cryptographic response to a challenge transmitted by the car. A valid response unlocks the doors, deactivates the alarm system, and enables the engine to start. As a consequence, the only remaining mechanical part in some cars is a door lock for emergencies (usually found behind a plastic cover on the driver's side), to be used when the battery is depleted.

Unfortunately, PKES does not require user interaction (such as a button press) on the side of the car key to initiate the cryptographic computations and signal transmission. The lack of user interaction makes PKES systems prone to relay attacks, in which the challenge and response signals are relayed via a separate wireless channel: The car key (e.g., in the pocket of the victim) and vehicle (e.g., parked hundreds of meters away) will assume their mutual proximity and successfully authenticate. Since the initial publication of these relay attacks in 2011 [14], tools that automatically perform relay attacks on PKES systems are available on the black market and are potentially used by criminals to open, start, and steal vehicles.

### 1.1.3 Remote Keyless Entry Systems

RKE systems rely on a unidirectional data transmission from the remote control, which is embedded in the car key, to the vehicle. Upon pressing a button, an active Radio Frequency (RF) transmitter in the remote control usually generates signals in a freely usable frequency band. These include the 315 MHz band in North America and the 433 MHz or 868 MHz band in Europe, with a typical range of several tens to hundreds of meters. Note that a few old cars have been using infrared technology instead of RF. RKE systems enable the user to comfortably lock and un-

lock the vehicle from a distance, and can be used to switch on and off the anti-theft alarm, when present.

The first remote controls for cars used no cryptography at all: The car was unlocked after the successful reception of a constant "fix code" signal. Replay attacks on these systems are straightforward. We encountered a Mercedes Benz vehicle manufactured around 2000 that still relies on such fix code RKE systems.

The next generation of RKE systems are so-called rolling code systems, which employ cryptography and a counter value that is increased on each button press. The counter value (and other inputs) form the plaintext for generating a new, encrypted (or otherwise authenticated) rolling code signal. After decryption/verification on the side of the vehicle, the counter value is checked by comparing it to the last stored counter value that was recognized as valid: An increased counter value is considered new and thus accepted. A rolling code with an old counter value is rejected. This mechanism constitutes an effective protection against replay attacks, since a rolling code is invalidated once it has been received by the vehicle. The cryptographic mechanisms behind rolling code systems are further described in Section 2.

In principle, such unidirectional rolling code schemes can provide a suitable security level for access control. However, as researchers have shown in the case of KEELOQ in 2008, the security guarantees are invalidated if they rely on flawed cryptographic schemes: KEELOQ was broken both by cryptanalysis [7, 15] and, in a more realistic setting, by side-channel attacks on the key derivation scheme executed by the receiver unit [12, 17]. Although it is frequently mentioned that KEELOQ is widely used for for vehicle RKE systems, our research indicates that this system is prevalently employed for garage door openers.

Another attack, targeting an outdated automotive RKE scheme of an unspecified vehicle (built between 2000 and 2005), was demonstrated by Cesare in 2014 [9]: An adversary has to eavesdrop three subsequent rolling codes. Then, using phase-space analysis, the next rolling code can be predicted with a high probability. However, apart from this attack the cryptographic security of automotive RKE systems has not been investigated to our knowledge. In particular, a large-scale survey and security analysis of very wide-spread rolling code systems has not been carried out.

A different, simple but effective method used by criminals to break into cars is to jam the RF communication when the victim presses the remote control to lock the car. The victim may not notice the attack and thus leave the car open. A variant of the attack is "selective jamming", i.e., a combined eavesdropping-and-jamming approach: The transmitted rolling code signal is monitored and at the same time jammed, with the effect that the car is not locked and the attacker possesses a temporarily valid (one-time) rolling code. Consequently, a car could be found appropriately locked after a burglary. This approach was first mentioned in [17] and later practically demonstrated by [16, 27]. Note that one successful transmission of a new rolling code from the original remote to the car usually invalidates all previously eavesdropped rolling codes, i.e., the time window for the attack is relatively small. Furthermore, it is usually not possible to change the signal contents, for example, convert a "lock" command into an "unlock". This limitation is often overlooked (e.g. in [16, 27]) and severely limits the practical threat posed by this type of attack.

## 1.2 Contribution and Outline

In this paper, we study several extremely widespread RKE systems and reveal severe vulnerabilities, affecting millions of vehicles worldwide. Our research was in part motivated by reports of unexplained burglaries of locked vehicles (for example [1, 2]), as well as scientific curiosity regarding the security of our own, personal vehicles.

The remainder of this paper is structured as follows: In Section 2, we briefly summarize the results of our preliminary analysis of different RKE systems solely by analyzing the transmitted RF signals. The main contributions presented subsequently are:

1. In Section 3, we analyze the RKE schemes employed in most VW Group group vehicles between 1995 and today. By reverse-engineering the firmware of the respective Electronic Control Units (ECUs), we discovered that VW Group RKE systems rely on cryptographic schemes with a single, worldwide master key, which allows an adversary to gain unauthorized access to an affected vehicle after eavesdropping a single rolling code.

2. In Section 4, we study an RKE scheme based on the Hitag2 cipher, as used by many different manufacturers. We have reverse-engineered the protocol in a black-box fashion and present a novel, fast correlation attack on Hitag2 applicable in an RKE context. By eavesdropping four to eight rolling codes, an adversary can re-

cover the cryptographic key within minutes and afterwards clone the original remote control.

## 2 Preliminary Analysis of RKE Systems

To address the research question of this paper: "how secure are modern automotive RKE systems?", we captured RF signals from the remote controls of a variety of vehicles, including our own cars (VW Passat 3B, Škoda Fabia, Alfa Romeo Giulietta). Today, the required hardware for receiving (and sending) RKE signals is widely available. For our analyses, we used various devices, including Software-Defined Radios (SDRs) (HackRF, USRP, `rtl-sdr` DVB-T USB sticks) and inexpensive RF modules. Figure 2 shows our simple setup which costs ≈ $40, is battery-powered, can eavesdrop and record rolling codes, emulate a key, and perform reactive jamming.

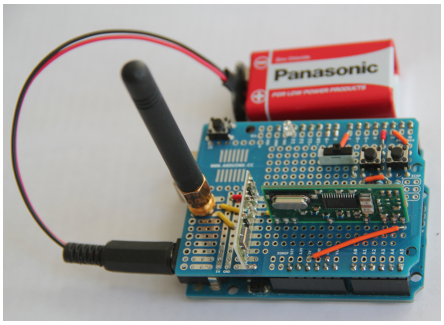


Figure 2: Arduino-based RF transceiver

Studying the raw received signals and guessing the respective modulation and encoding schemes turned out to be straightforward: The majority of the studied RKE systems uses simple Amplitude-Shift Keying (ASK) as modulation scheme, while a smaller percentage employs Frequency Shift Keying (FSK). For the encoding of the actual data bits, the most prevalent methods are Manchester encoding and pulse-width encoding. The utilized bit rates range from less than 1 kBit/s (for older remotes) to 20 kBit/s (for newer remotes).

A typical rolling code packet consists of a preamble (i.e., a regular sequence of 0 and 1), a fixed start pattern (a sequence of one or a few fixed bytes), the actual, cryptographic data payload, and a final checksum, cf. Figure 3. Note that many schemes slightly deviate from this general structure. Also, in virtually all cases, the same packet is sent multiple times, presumably to increase the reliability in presence of environmental disturbances.

The data payload normally contains the Unique


Figure 3: General packet structure of a rolling code. Gray background indicates that the part is either encrypted or authenticated.

Identifier (UID) of the remote control, the rolling counter value, and the pressed button (i.e., "unlock", "lock", "open trunk", in the US also "panic" or "alarm"). Obviously, the data sent by the remote control has to be cryptographically authenticated in some way. There appear to be two major routes that were taken by designers of RKE systems:

**Implicit authentication:** The complete payload (or part of it) is symmetrically encrypted. The receiver then decrypts the packet, and checks if the content is valid, i.e., if the UID is known to the vehicle and the counter is in its validity window. Examples for this approach can be found in Section 3.
**Explicit authentication:** Some form of Message Authentication Code (MAC) is computed over the data payload and then appended to the packet. An example of this approach is the Hitag2 scheme described in Section 4.

As a next step, we tried to determine the utilized encryption algorithms. However, a search for publicly available documentation or data sheets yielded little results. For example, the systems employed in VW Group vehicles (VW, Seat, Škoda, and Audi) appear to be a complete black box without any publicly documented security analysis. Since VW Group vehicles are extremely wide-spread, we selected this manufacturer as the target of our first case study (Section 3). Our second case study focuses on the Hitag2 scheme, for which abridged (one-page) data sheets can be found on the Internet [26]. We found Hitag2-based remote controls in vehicles made by a variety of manufacturers, hence, we opted to recover the exact functionality and further analyze the security of this RKE scheme (Section 4).

## 3 Case Study 1: The VW System

With over 23% market share in Europe (September 2015) and 11.1% worldwide (August 2014), the VW Group is amongst the leading global automotive manufacturers [13]. We had access to a wide variety of VW Group vehicles for our security analysis, from vehicles manufactured in the early 2000s to ones for the model year 2016. In total, the VW Group has sold almost 100 million cars from 2002 until 2015. While not all of these vehicles use the

RKE schemes covered in this section, we have strong indications that the vast majority is vulnerable to the attacks presented in the following. Note that the VW Group also includes certain luxury brands (e.g., Porsche, Bentley, Lamborghini, Bugatti) that we did not analyze in detail. Instead, we focused on more wide-spread vehicles manufactured by VW, Seat, Škoda, and Audi. For a list of cars that we validated our findings with, refer to Section 3.5.1. Eavesdropping and analyzing the signals transmitted by numerous remote controls, we identified at least 7 different RKE schemes, referred to as VW-x (x = 1…7) in the following. Out of these systems, we selected the four schemes covering the largest amount of vehicles:

**VW-1:** The oldest system, used in model years until approximately 2005. The remote control transmits On-Off-Keying (OOK) modulated signals at 433.92 MHz, using pulse-width coding at a bitrate of 0.667 kBit/s.

**VW-2:** Used from approximately 2004 onwards. The operating frequency is 434.4 MHz using OOK (same as for VW-3 and VW-4), transmitting Manchester-encoded data at a bitrate of 1 kBit/s.

**VW-3:** Employed for models from approximately 2006 onwards, using a frequency of 434.4 MHz and Manchester encoding at a bitrate of 1.667 kBit/s. The packet format differs considerably from VW-2.

**VW-4:** The most recent scheme we analyzed, found in vehicles between approximately 2009 and 2016. The system shares frequency, encoding, and packet format with VW-3, but uses a different encryption algorithm (see below).

The remaining three schemes are used in Audi vehicles from approximately 2005 until 2011 (VW-5), the VW Passat since 2005 (model B6/type 3C and newer, VW-6) and new VW vehicles like the Golf 7 (VW-7). We have not further investigated the security of these systems, but at least for older vehicles, it seems likely that similar design choices as for VW-1–VW-4 were made.

For our initial analyses, we implemented the most likely demodulation and decoding procedure for all of the above systems. We then collected rolling codes of multiple remote controls for each scheme and compared the resulting data. For all schemes VW-1–VW-4, we found that most of the packet content appeared to be encrypted, except for a fixed start pattern and the value of the pressed button sent in plain. We hence assumed that all systems use implicit authentication, i.e., check the correctness of a rolling code after decryption. Demodulation routines for VW-3 and VW-4 were independently published in 2015 [6] after we had carried our preliminary analysis. Note that this does not cover any of the cryptographic algorithms presented here.

## 3.1 Analysis of Remote Control and ECU

We obtained various VW Group remote controls and extracted the Printed Circuit Boards (PCBs) for further analysis of the hardware. A typical PCB for a VW Group RKE remote includes a Microcontroller (µC), an RF transmitter, an antenna (integrated on the PCB) and a coin cell battery as the main components. On many remote control PCBs (e.g., implementing VW-2), we found a µC marked with Temic/Atmel M44C890E, cf. Figure 4. According to the datasheet available online [3], this µC is a 4-bit processor, the so-called MARC4. The µC is mask-programmed, i.e., the program code is placed in Read Only Memory (ROM) and hence fixed at manufacturing. According to Laurie [21], it is possible to re-construct the program code of MARC4 processors by taking microscopic photographs of the ROM memory and applying further image processing to extract the value of each individual bit. However, we did not follow this approach because we did not have access to suitable microscopic equipment.

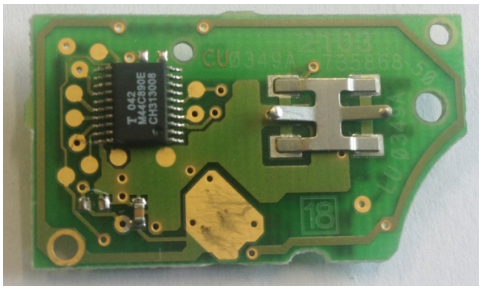


Figure 4: PCB of an older VW Group remote control using a MARC4 µC

When studying remote controls of newer vehicles, we found different, not easily identifiable µCs on the PCB. An example of this is shown in Figure 5: We could not identify the type of µC from the markings on the main IC (top, towards the right), which complicates the reverse engineering.

It seemed conceivable that some form of key derivation could be present, which would have to be implemented on the receiving ECU's side. Thus, we opted to analyze the RKE ECUs in the vehicle that receive and process the remote control signals. We therefore bought a number of ECUs implementing the respective RKE functionality, and attemp-

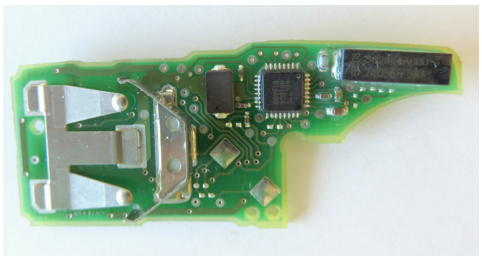

Figure 5: PCB of a newer VW Group remote control using an unidentified µC

ted to extract the firmware of the µCs present on the PCB of the ECU. Note that in contrast to the low-power 4-bit or 8-bit processors usually employed in the remote control, the RKE ECUs often handles numerous additional features of the vehicle and thus utilizes a more powerful, Flash-programmable 16-bit or 32-bit µC (with documented debug and programming interfaces).

Using widely available, standard programming tools for automotive processors, we were able to obtain firmware dumps for all studied ECUs. We then located and recovered the cryptographic algorithms by performing static analysis of the firmware image, searching amongst others for constants used in common symmetric ciphers and common patterns of such ciphers (e.g., table lookups, sequences of bitwise operations). The results of this process are described in more detail for each scheme VW-1–VW-4 in the following. Note that as part of our negotiations with VW Group, and to protect VW Group customers, we agreed to not fully disclose the part numbers of the analyzed ECUs and the employed µCs at this point. We furthermore agreed to omit certain details of the reverse-engineering process, as well as the values of cryptographic keys.

## 3.2 The VW-1 Scheme

The VW-1 system is the only VW Group scheme discussed in this paper that operates at 433.92 MHz (all newer systems use a frequency of 434.4 MHz). In contrast to newer RKE schemes, the start of a packet is not indicated by a long preamble, but by a single 1-0 pattern (500 µ$s$ high level, 500 µ$s$ low level). After this, the data bits are transmitted LSB-first in pulse-width encoded form: A zero is indicated by a short high level followed by a longer low level, while a one is represented with the opposite pattern (long high, short low). We discovered that the first four bytes hold the UID of the remote in an obfuscated form (several bytes of the packet are `XOR`ed). The following three bytes *lfsr* hold the (byte-permuted) state of a Linear Feedback Shift Register (LFSR) that is clocked a fixed number of ticks for each new rolling code (i.e., the LFSR state has the role of a counter). For reasons of responsible disclosure, we do not provide the full details of the obfuscation function and the LFSR feedback in this paper. One bit of the final nibble *btn* indicates the pressed button. The overall structure of a VW-1 rolling code packet is shown in Figure 6:

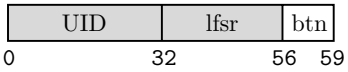


Figure 6: Packet structure of a rolling code for VW-1. Gray background indicates that the part is obfuscated or holds the LFSR state. The start pulse is not shown.

In conclusion, the security of the VW-1 is *solely based on obscurity*. Neither is there a cryptographic key involved in the computation of the rolling code, nor are there any vehicle or remote control specific elements for some form of key diversification. With the knowledge of the details of the obfuscation function and the LFSR, an adversary can generate valid rolling codes to open and close a VW-1 vehicle based on a single eavesdropped signal (to obtain the UID and the current state of the LFSR). Note that we observed similarly insecure LFSR-based schemes in older Audi vehicles built before 2004.

## 3.3 The VW-2 and VW-3 Schemes

Starting with VW-2, a rolling code packet has the following structure: A preamble (regular 0-1 pattern) is followed by a fixed start sequence *start* (individual per scheme), an encrypted 8-byte payload, and finally a byte *btn* indicating the button that was pressed. The packet structure (not showing the preamble) is depicted in Figure 7.

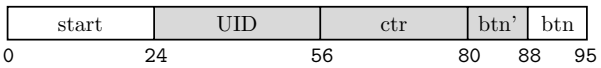


Figure 7: Packet structure of a rolling code for VW-2–4. Gray background indicates that the part is encrypted. Note that the fixed start pattern is shorter for VW-2.

The 8-byte payload is generated from the following plaintext: a 4-byte $UID$, a 3-byte counter $ctr$, and one byte $btn'$ again indicating the pressed button. This payload is then encrypted using a proprietary block cipher that we recovered from the ECU

firmware as described in Section 3.1. We later found that this cipher appears to be the so-called AUT64 cipher employed in certain immobilizer transponders as well [4]. Hence, we will use the name AUT64 in the following and follow the notation given in the public datasheet.

AUT64 is an iterated cipher, operating on 8-byte blocks. It uses a round structure as depicted in Figure 8: In each round $i$ the state (represented as bytes $a_0 \ldots a_7$) is first byte-permuted, using a *key-dependent* permutation $\sigma$. This permutation is fully described by a $3 \cdot 2^3 = 24$ bit string. Then, bytes $a_0 \ldots a_6$ are left unchanged, while byte $a_7$ is updated using the round function $g(a_0, \ldots, a_7, key_i)$, where $key_i$ is a 32-bit round key. In the case of AUT64 in the VW Group system, the cipher has 12 rounds, while the datasheet [4] only specifies a possible number of rounds between 8 and 24. The

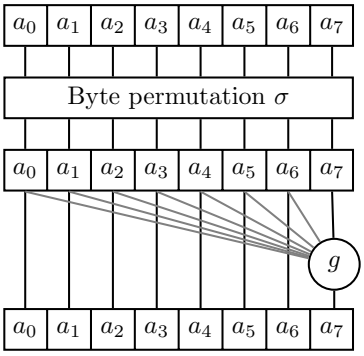


Figure 8: One round $i$ of the AUT64 block cipher as used in VW-2 and VW-3. $a_0, \ldots, a_7$ is the 8-byte state of the cipher, $g(a_0, \ldots, a_7, key_i)$ the round function.

internal structure of $g$ is shown in Figure 9: The input bytes $a_0, \ldots, a_7$ are first combined with the 32-bit round key $key_i$ using a sequence of concatenations, table look-ups, and XOR operations denoted as $f$. Note that the round key is derived from a part (denoted as $k_f$ in the following) of the main key $k$ by a fixed, nibble-wise permutation per round. Each nibble of the 8-bit output of $f$ is then passed through the same 4-to-4 S-Box $\tau$, bit-permuted using the same permutation $\sigma$ used for the state (but applied on a bit-level), and again passed through a second instance of $\tau$. Note that both $\sigma$ and $\tau$ are *key-dependent* in addition to $key_i$. Hence, the full key of the AUT64 cipher is the tuple $k = (k_f, \sigma, \tau)$ with an overall key size of $32 + 3 \cdot 2^3 + 4 \cdot 2^4 = 120$ bit.

However, not all choices for $\tau$ and $\sigma$ are permissible in order to have a bijective S-Box and a valid permutation—in total, there are 16! bijective

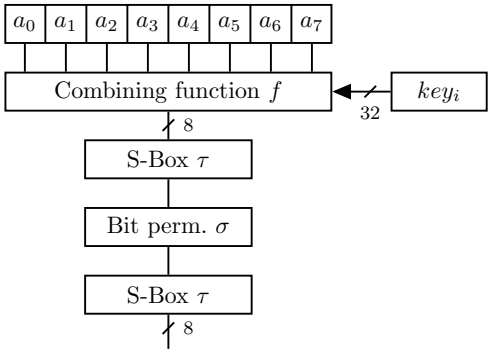


Figure 9: One round function $g$ of the AUT64 block cipher as used in VW-2 and VW-3. $a_0, \ldots, a_7$ is the 8-byte state of the cipher, $key_i$ the round key.

4-to-4 S-Boxes and 8! permutations. This results in an effective key size of $32 + \log_2(8!) + \log_2(16!) = 91.55$ bit. Finding an AUT64 key by exhaustive search is therefore beyond current computational capabilities, where a security level of 80 bit is usually deemed acceptable for lightweight ciphers.

We have not further analyzed the mathematical security of the cipher, but believe this to be an interesting research problem, especially due to the unconventional design with several key-dependent operations. For the analysis of the VW-2 and VW-3 RKE systems, however, it turned out that no further cryptanalysis is necessary: Both schemes use a fixed, *global master key* independent of vehicle or remote control. In other words, this means that the same AUT64 key is stored in millions of ECUs and RKE remotes, *without any key diversification* being employed at all. The sole means by which the vehicle determines if a rolling code is valid is hence by whitelisting certain UIDs and checking if the counter is within the validity window. Incidentally, this also implies that a VW Group vehicle using a particular scheme receives and decrypts all rolling codes for that scheme transmitted in the vicinity.

Note that the global AUT64 master keys for VW-2 and VW-3 are different, but both can be extracted from the ECU firmware and possibly from the µC in the remote control as well (e.g. with invasive attacks like micro-probing or side-channel analysis).

## 3.4 The VW-4 Scheme

In newer VW Group vehicles from approximately 2009 onwards, we found an RKE system that has the same encoding and packet structure as VW-3 (although with a different start pattern), but does not employ the AUT64 cipher. For this system VW-4,

the analysis of the respective ECU firmware revealed that the XTEA cipher [24] is used to encrypt a rolling code packet with a format otherwise identical to VW-3 (cf. Figure 7).

XTEA is a block cipher based on a 64-round Feistel structure with 64-bit block size and 128-bit key. Due to the structure of the round function based on Addition, Rotate, XOR (ARX) operations, it is well suited for lightweight software implementations required for low-end and low-power devices like RKE remotes. The best known cryptanalytical attack on XTEA [22] is of theoretical nature (related-key rectangle attack on 36 rounds with $2^{63.83}$ byte of data and $2^{104.33}$ steps) and hence not relevant in the context of RKE systems.

However, again we found that a *single, worldwide key* is used for all vehicles employing the VW-4 system. The same single point of failure of the older systems VW-1–VW-3 is hence also present in recently manufactured vehicles. For example, we found this scheme implemented in an Audi Q3, model year 2016, and could decrypt and generate new valid rolling codes to open and close this vehicle (and numerous other VW Group vehicles, cf. Section 3.5.1).

## 3.5 Implications and Observations

As the main result of this section, we discovered that the RKE systems of the majority of VW Group vehicles have been secured with only a few cryptographic keys that have been used worldwide over a period of almost 20 years. With the knowledge of these keys, an adversary only has to eavesdrop a single signal from a target remote control. Afterwards, he can decrypt this signal, obtain the current UID and counter value, and create a clone of the original remote control to lock or unlock any door of the target vehicle an arbitrary number of times.

We observed that (mostly) VW-4 vehicles blocked the original remote control if a valid rolling code with a counter more than 2 behind is received. In other words, if *ctr* is the value expected by the vehicle, any rolling code with $ctr-2$ or less leads to the blocking. If an adversary sends at least two valid signals with increased counter values (e.g., "unlock" and "lock"), the original remote control of the owner will stop working in the moment when the car receives an outdated signal. In this case, usually automatic re-synchronization procedures described in the respective vehicle's manual help technically experienced car owners to re-synchronize the remote control to the car. In contrast, if the adversary only sends a single valid signal, the original remote will not be blocked, but only operate on the second button press, because the counter in vehicle and remote are in sync afterwards. Note that the blocking behaviour could be used for an automatized Denial-of-Service (DoS) attack (aiming to lock out the legitimate car owners of affected vehicles) by intentionally sending an old signal (with a counter value of $ctr-2$ or less).

In conclusion, while the cryptographic algorithms have improved over the years (from LFSR over AUT64 to XTEA), the crucial problem of key distribution has not been properly solved in the studied schemes VW-1–4. However, according to VW Group, this problem has been addressed in the latest generation of vehicles, where individual cryptographic keys are used. We discuss the consequences and general implications of a successful attack on a RKE system in more detail in Section 5.

### 3.5.1 Vulnerable Vehicles

Our findings affect amongst others the following VW Group vehicles manufactured between 1995 and 2016. Cars that we have practically tested are highlighted in bold. Note that this list is not exhaustive, as we did not have access to all types and model years of cars, and that it is unfortunately not clear if and when a car model has been upgraded to a newer scheme.

**Audi:** **A1**, **Q3**, R8, S3, TT, various other types of Audi cars (e.g. remote control part number 4D0 837 231)

**VW:** **Amarok**, (New) Beetle, Bora, **Caddy**, **Crafter**, **e-Up**, **Eos**, Fox, **Golf 4**, Golf 5, **Golf 6**, Golf Plus, **Jetta**, Lupo, **Passat**, **Polo**, **T4**, **T5**, Scirocco, **Sharan**, **Tiguan**, Touran, **Up**

**Seat:** **Alhambra**, Altea, Arosa, Cordoba, **Ibiza**, **Leon**, MII, Toledo

**Škoda:** City Go, Roomster, **Fabia 1**, **Fabia 2**, **Octavia**, **SuperB**, **Yeti**

It is conceivable that all VW Group (except for some Audi) cars manufactured in the past and partially today rely on a "constant-key" scheme and are thus vulnerable to the attacks described in this paper, except for those cars that rely on the latest platform, e.g., the Golf 7 for VW.

Note that identical VW Group cars are sold under different names in other countries, e.g., some Golf versions were sold as "Rabbit" in North America. We have tested some remote controls operating at 315 MHz, e.g., for the US market, and found them to be vulnerable to our attacks as well, i.e., the only difference to their European counterparts is the operating frequency. Furthermore, cars of different brands

may share the same basic technology, e.g., we found some model years of Ford Galaxy that have the same flawed RKE system as their VW Group derivatives VW Sharan and Seat Alhambra.

#### 3.5.2 Temporary Countermeasures

Completely solving the described security problems would require a firmware update or exchange of both the respective ECU and (worse) the vehicle key containing the remote control. Due to the strict testing and certification requirements in the automotive industry and the high cost of replacing or upgrading all affected car keys in the field, it is unlikely that VW Group can roll out such an update in the short term. Hence, we give recommendations for users of affected vehicles in the following.

The well-known advice (see e.g. [25]) to verify that a vehicle was properly locked with the remote control (blinking direction lights, sound) is no longer sufficient. An adversary may have eavesdropped the "lock" signal from a distance of up to 100 m and generate a new, valid "unlock" rolling code any time later. Preventing or detecting the eavesdropping of RF signals is impractical. Hence, the only remaining (yet impractical) countermeasure is to fully deactivate or at least not use the RKE functionality and resort to the mechanical lock of the vehicle. Note that in addition, for many cars, the alarm will trigger after a while if the car doors or the trunk are mechanically opened, unless the immobilizer is disarmed with the original key.

With respect to forensics, there are several potential indicators (due to the nature of rolling code schemes) that the remote control may have been cloned: If the vehicle does not unlock on the first button press, this could imply that an adversary has sent valid rolling codes with counter values greater than the one stored in the original remote control. Note that no traces of the attack are left once the counter in the original remote control has caught up with the increased value stored in the car. Further, a complete blocking of the remote control (see above) may be an indicator (e.g., for insurance-related court cases) that the RKE system was attacked. It should however be taken into account that, according to our practical tests, the remote control will also be blocked if the car receives a counter that is increased by more than 250 compared to the last stored value—this could for example happen if the remote control buttons are pushed many times while not in the range of the vehicle.

## 4 Case Study 2: The Hitag2 System

The Hitag2 rolling code system is an example of a RKE scheme that is not specific to a single vehicle brand. Instead, it is implemented on the PCF7946 and PCF7947 ICs manufactured by NXP. While these ICs contain an 8-bit general-purpose µC that (in theory) allows to realize a fully proprietary scheme [26], it appears that numerous vehicle manufacturers have used a similar (though not identical) RKE system, potentially following NXP's reference implementation. In contrast to the VW Group system described in Section 3, it seems that manufacturers did not use a fixed, global cryptographic key. Hence, to break this system, we developed a novel attack to exploit the cryptographic weaknesses of Hitag2 in the RKE context.

We first describe the Hitag2 cipher, which was previously published in [35]. We have fully reverse-engineered the rolling code scheme used in the Hitag2 remote control ICs PCF7946/7947 as further described in Section 4.2. The analysis was done in a black-box fashion—we used a remote control for which we were able to obtain the Hitag2 key (since it was shared with the immobilizer in this particular case), guessed potential implementations (based on the immobilizer protocol) for the rolling code system, and finally recovered the complete scheme. In contrast to the analysis of the VW Group systems, no firmware extraction and reverse-engineering of program code was necessary.

To this date, the best known practical cryptanalysis of Hitag2 was proposed in [32] in the context of vehicle immobilizers. Their attack requires 136 authentication attempts and $2^{35}$ encryptions/lookups, which take 5 minutes on a laptop. In the context of RKE systems, gathering 136 rolling code traces is not practical in a realistic scenario, as it requires to wait for the victim to push a button on the remote that many times. We therefore propose a new attack that requires eavesdropping less authentication attempts (usually between 4 and 8) and one minute computation on a laptop. In Section 4.4, we present our novel correlation attack on Hitag2 in a RKE scenario.

We first need to introduce some notation. Let $\mathbb{F}_2 = \{0, 1\}$ the field of two elements (or the set of Booleans). The symbol $\oplus$ denotes exclusive-or (XOR) and $0^n$ denotes a bitstring of $n$ zero-bits. Given two bitstrings $x$ and $y$, $xy$ denotes their concatenation. $\overline{x}$ denotes the bitwise complement of $x$. We write $y_i$ to denote the $i$-th bit of $y$. For example, given the bitstring $y = \texttt{0x03}$, $y_0 = y_1 = 0$ and $y_6 = y_7 = 1$. We denote encryptions by $\{-\}$.

## 4.1 Hitag2 Cipher

The targeted RKE protocol uses the Hitag2 stream cipher. This cipher has been reverse engineered in [35]. The cipher consists of a 48-bit LFSR and a non-linear filter function $f$. Each clock cycle, twenty bits of the LFSR are put through the filter function, generating one bit of keystream. Then the LFSR is shifted one bit to the left, using the feedback polynomial to generate a new bit on the right. See Figure 10 for a schematic representation.

**Definition 4.1** *The feedback function* $L\colon \mathbb{F}_2^{48} \to \mathbb{F}_2$ *is defined by* $L(x_0 \dots x_{47}) := x_0 \oplus x_2 \oplus x_3 \oplus x_6 \oplus x_7 \oplus x_8 \oplus x_{16} \oplus x_{22} \oplus x_{23} \oplus x_{26} \oplus x_{30} \oplus x_{41} \oplus x_{42} \oplus x_{43} \oplus x_{46} \oplus x_{47}$.

The filter function $f$ consists of three different circuits $f_a, f_b$ and $f_c$, which output one bit each. The circuits $f_a$ and $f_b$ are employed more than once, using a total of twenty input bits from the LFSR. Their resulting bits are used as input for $f_c$. The circuits are represented by three Boolean tables that contain the resulting bit for each input.

**Definition 4.2 (Filter function)** *The filter function* $f\colon \mathbb{F}_2^{48} \to \mathbb{F}_2$ *is defined by*

$$\begin{aligned} f(x_0 \dots x_{47}) = f_c(&f_a(x_2 x_3 x_5 x_6), f_b(x_8 x_{12} x_{14} x_{15}),\\ &f_b(x_{17} x_{21} x_{23} x_{26}), f_b(x_{28} x_{29} x_{31} x_{33}),\\ &f_a(x_{34} x_{43} x_{44} x_{46})), \end{aligned}$$

*where* $f_a, f_b\colon \mathbb{F}_2^4 \to \mathbb{F}_2$ *and* $f_c\colon \mathbb{F}_2^5 \to \mathbb{F}_2$ *are*

$$\begin{aligned} f_a(i) &= (\texttt{0xA63C})_i\\ f_b(i) &= (\texttt{0xA770})_i\\ f_c(i) &= (\texttt{0xD949CBB0})_i. \end{aligned}$$

*Because* $f(x_0 \dots x_{47})$ *only depends on* $x_2, x_3, x_5 \dots x_{46}$ *we shall define* $f_{20}\colon \mathbb{F}_2^{20} \to \mathbb{F}_2$, *writing* $f(x_0 \dots x_{47})$ *as* $f_{20}(x_2, x_3, x_5 \dots x_{46})$.

**Remark 4.3 (Cipher schematic)** *Figure 10 is different from the schematic that was introduced by [35] and later used by [11, 28, 34]. The input bits of the filter function in Figure 10 are shifted by one with respect to those of [35]. The filter function in the old schematic represents a keystream bit at the previous state* $f(x_{i-1} \dots x_{i+46})$, *while the one in Figure 10 represents a keystream bit of the current state* $f(x_i \dots x_{i+47})$. *Furthermore, we have adapted the Boolean tables to be consistent with our notation.*

## 4.2 Rolling Code Scheme

This section describes the rolling code scheme used by remotes based on the chips PCF7946/7947. When a button on the remote control is pressed, it transmits a message of the form shown in Figure 11. $UID$ is a 32-bit identifier; $btn$ is a 4-bit button identifier; $lctr$ are the 10 least-significant bits of a 28-bit counter $ctr$; ks are 32-bits of keystream; and $chk$ is an 8-bit checksum. The checksum is computed by simply XORing each byte, i.e., computing a parity byte.

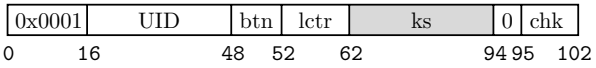


Figure 11: Packet structure of a rolling code for Hitag2. Gray background indicates the keystream part produced by the cipher.

During the authentication protocol, the internal state of the stream cipher is initialized. The initial state consists of the 32-bits UID concatenated with the first 16 bits of the key $k$. Next, the counter $ctr$ is incremented and then $iv = ctr||btn$ is XORed with the last 32 bits of the key and shifted into the LFSR. From this point, the next 32 bits of keystream, which are output by the cipher $ks$, are sent as proof of knowledge of the secret key $k$.

## 4.3 Cipher Initialization

The following precisely defines the initialization of the cipher and the generation of the LFSR stream $a_0 a_1 \dots$ and the keystream $ks$.

**Definition 4.4** *Given a key* $k = k_0 \dots k_{47} \in \mathbb{F}_2^{48}$, *an identifier* $id = id_0 \dots id_{31} \in \mathbb{F}_2^{32}$, *a counter* $ctr = ctr_0 \dots ctr_{27} \in \mathbb{F}_2^{28}$, *a button identifier* $btn_0 \dots btn_3 \in \mathbb{F}_2^4$ *and keystream* $ks = ks_0 \dots ks_{31} \in \mathbb{F}_2^{32}$, *we let the initialization vector* $iv \in \mathbb{F}_2^{32}$ *be defined as*

$$iv = ctr||btn.$$

*Furthermore, the internal state of the cipher at time* $i$ *is* $\alpha_i := a_i \dots a_{47+i} \in \mathbb{F}_2^{48}$. *Here the* $a_i \in \mathbb{F}_2$ *are given by*

$$\begin{aligned} a_i &:= id_i && \forall i \in [0,31] && (1)\\ a_{32+i} &:= k_i && \forall i \in [0,15] && (2)\\ a_{48+i} &:= k_{16+i} \oplus iv_i \oplus f(a_i \dots a_{i+47}) && \forall i \in [0,31] && (3)\\ a_{80+i} &:= L(a_{32+i} \dots a_{79+i}) && \forall i \in \mathbb{N}\,. && (4) \end{aligned}$$

*Furthermore, we define the keystream bit* $ks_i \in \mathbb{F}_2$ *by*

$$ks_i := f(a_{32+i} \dots a_{79+i}) \qquad \forall i \in [0,31]. \quad (5)$$

*Note that the* $a_i$, $\alpha_i$, *and* $ks_i$ *are formally functions of* $k$, $id$, *and* $iv$. *Instead of making this explicit by writing, e.g.,* $a_i(k, id, iv)$, *we just write* $a_i$ *where* $k$, $id$, *and* $iv$ *are clear from the context.*

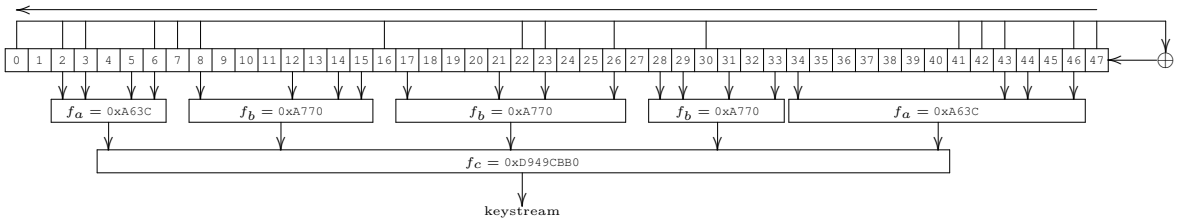


Figure 10: Structure of the Hitag2 stream cipher, based on [35]

## 4.4 A Fast Correlation Attack on Hitag2

This section describes a practical key-recovery correlation attack against Hitag2. This attack requires a minimum of four rolling codes ("traces"), but will be faster and have higher success probability if more are provided. The rolling codes can have an arbitrary counter value, i.e., do not have to be consecutive. In fact, the probability of success is higher when the traces are not consecutive, as consecutive traces often only differ in a few bits from each other, thus providing less correlation information. It also depends on whether the same button was pressed or not. The lower limit of four traces for the key recovery to work was determined experimentally. The number of consecutive traces needed is higher, usually eight. Let $\langle UID, iv^j, ks^j \rangle$, $j = 0 \ldots n-1$ be $n$ authentication traces for $n > 3$. Then, the attacker proceeds as follows:

1. The adversary first guesses a 16-bit window corresponding to LFSR stream bits $a_{32} \ldots a_{47}$. Observe that $a_{32} \ldots a_{47} = k_0 \ldots k_{15}$ and together with the UID, this gives the adversary LFSR bits $a_0 \ldots a_{47}$, see Definition 4.4. Also note that $a_0 \ldots a_{47}$ is constant over all traces. The adversary can now compute $b_0 = f(a_0 \ldots a_{47})$.
2. The adversary will then shift this 16-bit window to the left of the LFSR, until bits $a_{32} \ldots a_{47}$ are on the very left of the LFSR. This is the point when the cipher starts outputting $ks$, see Equation 5.
3. Next, the adversary will compute a correlation score for this guess. The window determines 8 input bits $x_0 \ldots x_7$ to the filter function $f_{20}$ (see Figure 10) while the remaining 12 inputs remain unknown. This correlation is taken as the ratio of those $2^{12}$ input values $x_8 \ldots x_{19}$ that produce the correct keystream bit ($ks_0$). Furthermore, shifting our window further to the left allows the adversary to perform tests on multiple keystream bits ($ks_0 \ldots ks_{15}$). Although, with every bit shift, the window becomes smaller as the leftmost bits will fall outside the LFSR, meaning that more input bits are unknown.

   **Definition 4.5** *We define the single-bit correlation score as:*

   $$bit\_score(x_0 \ldots x_{n-1}, b) = \frac{\#(b = f_{20}(y_0 \ldots y_{19}))}{2^{19-n}}$$

   *where* $y_0 \ldots y_{n-1} = x_0 \ldots x_{n-1}, n < 20$ *(at the first iteration of Step 3, n=8). We define the multiple-bit correlation score as:*

   $$score(x_0, ks_0) = bit\_score(x_0, ks_0)$$
   $$score(x_0 \ldots x_{n-1}, ks_0 \ldots ks_{n-1}) = bit\_score(x_0 \ldots x_{n-1}, ks_{n-1}) * score(x_0 \ldots x_{n-2}, ks_0 \ldots ks_{n-2})$$

   *for* $n < 20$.

   The adversary will assign this guess the average score over all traces. Note that, so far this scoring computation is independent of the value $iv$ as it happens before $iv$ gets to have any influence on it (i.e. it is only XORed with unknown bits).
4. The adversary will now sort all guesses according to their score and store them in a table of fixed size, discarding the guesses with lowest scores when needed. Experiments show that a table of size 400,000 guesses is usually sufficient.
5. For each guess in the table, the adversary goes back to Step (1) and proceeds as before, except that she will now extend the window size by one (to size $17, \ldots, 32$), guessing the next LFSR stream bit $(a_{48}, \ldots, a_{51})$. The bigger window allows the adversary to test on an additional bit of keystream, giving her more meaningful correlation information each time. Special care needs to be taken at Step (3) while scoring multiple traces, since $a_{48} = k_{16+i} \oplus iv_i \oplus b_0$ (see Eq. 3) and the $iv$ will be different in each trace. This is not a problem since in the previous Step (1) we had computed the corresponding keystream bit $b_i$, and $iv_i$ is sent in clear. Therefore key bits $k_{16+i}$ can be computed for $i \in [0, 31]$.

The power of this attack comes from using the window on the right of the LFSR to compute the necessary keystream bits to correct the internal state, while combining different traces and using the window on the left of the LFSR to get meaningful correlation information on multiple keystream bits.

### 4.5 Practical Results and Implications

We implemented the above correlation attack on a standard laptop. When executing this attack in practice, the first obstacle that an adversary faces is the fact that only the 10 Least Significant Bits (LSBs) of the counter *ctr* are sent over the air (see Figure 11), but the full 28-bit counter is used to initialize the cipher (both car and remote store the full counter). Therefore, the adversary needs to guess the remaining 18 bits. In practice, this is not a problem as it takes $2^{10} = 1024$ key pushes on the remote to have a carry to the Most Significant Bits (MSBs) and therefore this usually happens only a couple of times a year. In the worst case, the adversary has to repeat the above attack with increasing MSBs until she has the correct guess.

On average, our attack implementation recovers the cryptographic key in approximately 1 minute computation, requiring a few eavesdropped rolling codes (between 4 and 8). As mentioned, the adversary needs to repeat this computation for each guess of the 18 MSBs of the counter. For the vehicles we tested, the MSBs of the counter were usually between 0 and 10, which results in a total attack time of less than 10 min. Besides, there was a strong correlation between the vehicle's age and the counter value, so educated guesses are also possible.

We verified our findings in practice by building a key emulator and then unlocking and locking the vehicles with newly generated rolling codes:

| Manufacturer | Model | Year |
|---|---|---|
| Alfa Romeo | Giulietta | 2010 |
| Chevrolet | Cruze Hatchback | 2012 |
| Citroen | Nemo | 2009 |
| Dacia | Logan II | 2012 |
| Fiat | Punto | 2016 |
| Ford | Ka | 2009, 2016 |
| Lancia | Delta | 2009 |
| Mitsubishi | Colt | 2004 |
| Nissan | Micra | 2006 |
| Opel | Vectra | 2008 |
| Opel | Combo | 2016 |
| Peugeot | 207 | 2010 |
| Peugeot | Boxer | 2016 |
| Renault | Clio | 2011 |
| Renault | Master | 2011 |

The vehicles in the above list are our own and also from colleagues and friends who volunteered. We furthermore found the following list of supported vehicles for an after-market universal remote control [19] that is presumably implementing the Hitag2 RKE scheme: *Abarth* 500, Punto Evo; *Alfa Romeo* Giulietta, Mito; *Citroen* Jumper, Nemo; *Fiat* 500, Bravo, Doblo, Ducato, Fiorino, Grande Punto, Panda, Punto Evo, Qubo; *Dacia* Duster; *Ford* Ka; *Lancia* Delta, Musa; *Nissan* Pathfinder, Navara, Note, Qashqai, X-Trail; *Opel* Corsa, Meriva, Zafira, Astra; *Peugeot* Boxer, Expert; and *Renault* Clio, Modus, Trafic, Twingo. This list includes most of our tested vehicles. This would indicate that all vehicles mentioned in the list (although not practically tested by us) are vulnerable to the described attacks as well.

In contrast to the VW Group scheme, the vulnerabilities in the Hitag2 RKE system are caused by the cryptographically weak cipher, not a weak key distribution method. In consequence, even though it must be said that the correlation attack of Section 4.4 is devastating from a cryptographic point of view, the data complexity is slightly higher compared to the VW Group schemes, which can be broken with one single eavesdropped signal. The attack on Hitag2 requires at least four (not necessarily consecutive) rolling codes, i.e., the adversary has to be present for a longer period of time to capture signals for multiple key presses on the victim's remote control.

However, to quickly obtain the required rolling codes, the adversary could selectively jam the signal during the final checksum byte (which is predictable). In this case, the vehicle ignores the rolling code, but the adversary nevertheless obtains the keystream. The victim would hence notice that the vehicle does not respond, and instinctively press the button repeatedly. After having received the fourth signal, the adversary stops jamming and the remote control operates normally from the victim's point of view. However, the attacker has then collected the required amount of rolling codes to subsequently extract the cryptographic key. Hence, if the described behaviour is observed by a vehicle owner, it is an indication that an attack may be in progress.

## 5 Conclusion

Answering the original research question about the security of automotive RKE systems, the results of this paper show that major manufacturers have used insecure schemes over more than 20 years. Due to the widespread use of the analyzed systems, our findings have worldwide impact. Owners of affected vehicles should be aware that unlocking the doors of their car is much simpler than commonly assumed today. Both for the VW Group and the Hitag2 rolling code schemes, it is possible to clone the original remote control and gain unauthorized access to

the vehicle after eavesdropping one or a few rolling codes, respectively. The necessary equipment to receive and send rolling codes, for example SDRs like the USRP or HackRF and off-the-shelf RF modules like the TI Chronos smart watch, are widely available at low cost. The attacks are hence highly scalable and could be potentially carried out by an unskilled adversary. Since they are executed solely via the wireless interface, with at least the range of the original remote control (i.e., a few tens of meters), and leave no physical traces, they pose a severe threat in practice.

**Security and Safety Implications** The implications of our findings are manifold: Personal belongings left in a locked vehicle (as well as vehicle components like the infotainment system) could be stolen if a thief uses the vulnerabilities of the RKE system to unlock the vehicle after the owner has left. 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). Moreover, since a valid rolling code usually disables the alarm system, the theft is more likely to remain undetected for a longer period of time. Common recommendations like "lock it or lose it" [25] or "verify that the vehicle has been successfully locked and the transmission has not been jammed" (blinking direction lights, sound) are hence no longer sufficient to effectively prevent theft. A successful attack on the RKE and anti-theft system would also enable or facilitate other crimes:

– theft of the vehicle itself by circumventing the immobilizer system (e.g. [32, 33]) or by programming a new key into the car via the OBD port with a suitable tool

– compromising the board computer of a modern vehicle [10, 20], which may even affect personal safety, e.g., by deactivating the brakes while switching on the wiping system in a bend

– inconspicuously placing an object or a person inside the car. The car could be locked again after the act

– on-the-road robbery, affecting the personal safety of the driver or passengers if they (incorrectly) assume that the vehicle is securely locked

Note that due to the long range of RKE systems it is technically feasible to eavesdrop the signals of all cars on a parking lot or at a car dealer by placing an eavesdropping device there overnight. Afterwards, all vulnerable cars could be opened by the adversary. Practical experiments suggest that the receiving ranges can be substantially increased: The authors of [18] report eavesdropping of a 433 MHz RFID system, with technology comparable to RKE, from up to 1 km using low-cost equipment. Likewise, a large-scale DoS attack targeting VW Group cars would be possible with an automated approach—as a result, the RKE system of the vulnerable vehicle types would be deactivated for the respective remote control and VW Group would face increased demand for customer service, i.e., re-synchronizing remotes.

**Legal Implications, Forensics, and Countermeasures** It is unclear whether such attacks on the RKE scheme are currently carried out in the wild by criminals. However, there have been various media reports about unexplained theft from locked vehicles in the last years. The security issues described in this paper could explain such incidents. Note that we have analyzed further automotive RKE systems (with similar results regarding their (in)security), but due to the difficulty of responsible disclosure, cannot publish all results at this point.

As of today, even experts in car theft cases expressed the opinion that the alarm and electronic door locking systems of a car cannot be easily circumvented. From now on, they have to consider that special universal remote controls to bypass the security mechanisms might be used by criminals. In contrast to mechanical tools to open vehicles, such a device would leave no physical traces. Insurance companies may thus have to accept that certain car theft scenarios that have so far been regarded as insurance fraud (e.g. theft of personal belongings out of a locked car without physical traces) have, considering the results of this paper, a higher probability to be real. From a forensics point of view, the need to press the button of the remote control more than once in order to unlock the vehicle is an indicator that the car might have been accessed by a criminal. For VW Group vehicles, the "blocking" of a remote control should be regarded as suspicious as well. However, there are other causes for such behaviour, e.g., short range due to an empty battery of the remote control or environmental RF noise.

While the vulnerabilities of the VW Group system are due to worldwide master keys, Hitag2-based systems suffer from weaknesses in the cipher itself. Hence, in conclusion, for a "good" RKE system, both secure cryptographic algorithms (e.g., AES) and secure key distribution are necessary. Techniques to solve the security problems discovered in this paper are widely available [23]. Atmel has created an open RKE protocol design [5], which is pub-

lished in full detail. The security of their design was scrutinized by Tillich et al. in [29]. It is now up to vehicle manufacturers to securely implement such next-generation RKE schemes.

For owners of affected vehicles, as a temporary countermeasure in cases where valuable items are left in the vehicle, we can unfortunately only recommend to stop using or disable/remove the RKE part of the car key and fall back to the mechanical lock: **Lock It or Lose It? Remove It!**

## 6 Responsible Disclosure

Regarding the vulnerabilities of VW Group systems, we contacted VW Group first in November 2015. We discussed our findings in a meeting with VW Group and an affected sub-contractor in February 2016, before submitting the paper. VW Group received a draft version of this paper and the final version. VW Group acknowledged the vulnerabilities. As mentioned in the paper, we agreed to leave out amongst others the following details: cryptographic keys, part numbers of vulnerable ECUs, and the used programming devices and details about the reverse-engineering process.

For Hitag2, we notified NXP in January 2016. NXP received a version of this paper before submission. We would like to mention that the fact that Hitag2 is cryptographically broken has been publicly known for several years and NXP has already informed their customers back in 2012. We would further like to highlight that for several years, NXP offers newer, AES-based RKE ICs that are not affected by the vulnerabilities described in this paper. Furthermore, many car manufacturers have already started using the more secure chips for new designs.

## References


[1] ABC7NEWS. Key fob car thefts, 2013. http://abc7news.com/archive/9079852.

[2] ARSTECHNICA. After burglaries, mystery car unlocking device has police stumped, 2013. http://arstechnica.com/security/2013/06/after-burglaries-mystery-car-unlocking-device-has-police-stumped.

[3] ATMEL. M44C890 Low-Current Microcontroller for Wireless Communication , 2001. datasheet, available at http://pdf1.alldatasheet.com/datasheet-pdf/view/118247/ATMEL/M44C890.html.

[4] ATMEL. e5561 Standard Read/Write Crypto Identification IC, 2006. datasheet, available at http://www.usmartcards.com/media/downloads/366/Atmel%20e5561%20pdf-190.pdf.

[5] ATMEL. Embedded AVR Microcontroller Including RF Transmitter and Immobilizer LF Functionality for Remote Keyless Entry - ATA5795C. datasheet, available at http://www.atmel.com/images/Atmel-9182-Car-Access-ATA5795C_Datasheet.pdf, November 2014.

[6] BLOESSL, B. gr-keyfob. Github repository, 2015. https://github.com/bastibl/gr-keyfob.

[7] BOGDANOV, A. Attacks on the KeeLoq Block Cipher and Authentication Systems. In *Workshop on RFID Security (RFIDSec'08)* (2007). rfidsec07.etsit.uma.es/slides/papers/paper-22.pdf.

[8] BONO, S. C., GREEN, M., STUBBLEFIELD, A., JUELS, A., RUBIN, A. D., AND SZYDLO, M. Security analysis of a cryptographically-enabled RFID device. In *14th USENIX Security Symposium (USENIX Security 2005)* (2005), USENIX Association, pp. 1–16.

[9] CESARE, S. Breaking the security of physical devices. Presentation at Blackhat'14, August 2014.

[10] CHECKOWAY, S., MCCOY, D., KANTOR, B., ANDERSON, D., SHACHAM, H., SAVAGE, S., KOSCHER, K., CZESKIS, A., ROESNER, F., AND KOHNO, T. Comprehensive experimental analyses of automotive attack surfaces. In *20th USENIX Security Symposium (USENIX Security 2011)* (2011), USENIX Association, pp. 77–92.

[11] COURTOIS, N. T., O'NEIL, S., AND QUISQUATER, J.-J. Practical algebraic attacks on the Hitag2 stream cipher. In *12th Information Security Conference (ISC 2009)* (2009), vol. 5735 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 167–176.

[12] EISENBARTH, T., KASPER, T., MORADI, A., PAAR, C., SALMASIZADEH, M., AND SHALMANI, M. T. M. On the Power of Power Analysis in the Real World: A Complete Break of the KeeLoq Code Hopping Scheme. In *Advances in Cryptology – CRYPTO'08* (2008), vol. 5157 of *LNCS*, Springer, pp. 203–220.

[13] European Automobile Manufacturers Associaction. New passenger car registrations, 2015. available at http://www.acea.be/uploads/press_releases_files/20151016_PRPC_1509_FINAL.pdf.

[14] Francillon, A., Danev, B., and Capkun, S. Relay attacks on passive keyless entry and start systems in modern cars. In *Proceedings of the Network and Distributed System Security Symposium, NDSS 2011* (2011), The Internet Society.

[15] Indesteege, S., Keller, N., Dunkelmann, O., Biham, E., and Preneel, B. A practical attack on KeeLoq. In *27th International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT 2008)* (2008), vol. 4965 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 1–8.

[16] Kamkar, S. Drive It Like You Hacked It: New Attacks and Tools to Wirelessly Steal Cars. Presentation at DEFCON 23, August 2015.

[17] Kasper, M., Kasper, T., Moradi, A., and Paar, C. Breaking KeeLoq in a Flash: On Extracting Keys at Lightning Speed. In *Progress in Cryptology - AFRICACRYPT'09* (2009), B. Preneel, Ed., vol. 5580 of *LNCS*, Springer, pp. 403–420.

[18] Kasper, T., Oswald, D., and Paar, C. Wireless security threats: Eavesdropping and detecting of active RFIDs and remote controls in the wild. In *19th International Conference on Software, Telecommunications and Computer Networks – SoftCOM'11* (2011), pp. 1–6.

[19] Keyline S.p.A. RK60 guide, 2015. available at http://www.keyline.it/files/teste-elettroniche/electronic_heads_guide_13316.pdf.

[20] Koscher, K., Czeskis, A., Roesner, F., Patel, F., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., and Savage, S. Experimental security analysis of a modern automobile. In *31rd IEEE Symposium on Security and Privacy (S&P 2010)* (2010), IEEE Computer Society, pp. 447–462.

[21] Laurie, A. Fun with Masked ROMs — Atmel MARC4. Blog entry, 2013. http://adamsblog.aperturelabs.com/2013/01/fun-with-masked-roms.html.

[22] Lu, J. Related-key rectangle attack on 36 rounds of the XTEA block cipher. *International Journal of Information Security 8*, 1 (2008), 1–11.

[23] Moradi, A., and Kasper, T. A new remote keyless entry system resistant to power analysis attacks. In *Information, Communications and Signal Processing – ICICS 2009* (2009), IEEE, pp. 1–6.

[24] Needham, R. M., and Wheeler, D. J. TEA extensions. *Technical Report, Cambridge University, UK* (1997).

[25] Newport Beach PD. Lock It Or Lose It - Newport Beach Vehicle Crime, 2011. Video available at https://www.youtube.com/watch?v=Mmi2LRF7al8.

[26] Philips. PCF7946AT – Security Transponder Plus Remote Keyless Entry, 1999. datasheet, available at http://www.datasheet4u.com/pdf/PCF7946AT-pdf/609011.

[27] spencerwhyte. Jam Intercept and Replay Attack against Rolling Code Key Fob Entry Systems using RTL-SDR. Website, retrieved January 21, 2016, March 2014. http://spencerwhyte.blogspot.ca/2014/03/delay-attack-jam-intercept-and-replay.html.

[28] Sun, S., Hu, L., Xie, Y., and Zeng, X. Cube cryptanalysis of Hitag2 stream cipher. In *10th International Conference on Cryptology and Network Security (CANS 2011)* (2011), vol. 7092 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 15–25.

[29] Tillich, S., and Wójcik, M. Security analysis of an open car immobilizer protocol stack. In *10th International Conference on Applied Cryptograpy and Network Security (ACNS 2012)* (2012).

[30] Verdult, R. *The (in)security of proprietary cryptography*. PhD thesis, Radboud University, The Netherlands and KU Leuven, Belgium, April 2015.

[31] Verdult, R., and Garcia, F. D. Cryptanalysis of the Megamos Crypto automotive immobilizer. *USENIX ;login: 40*, 6 (2015), pp. 17–22.

[32] Verdult, R., Garcia, F. D., and Balasch, J. Gone in 360 seconds: Hijacking with Hitag2. In *USENIX Security Symposium* (August

2012), USENIX Association, pp. 237–252. https://www.usenix.org/system/files/conference/usenixsecurity12/sec12-final95.pdf.

[33] Verdult, R., Garcia, F. D., and Ege, B. Dismantling Megamos Crypto: Wirelessly Lockpicking a Vehicle Immobilizer. In *22nd USENIX Security Symposium (USENIX Security 2013)* (2015), USENIX Association, pp. 703–718.

[34] Štembera, P., and Novotný, M. Breaking Hitag2 with reconfigurable hardware. In *14th Euromicro Conference on Digital System Design (DSD 2011)* (2011), IEEE Computer Society, pp. 558–563.

[35] Wiener, I. Philips/NXP Hitag2 PCF7936/46/47/52 stream cipher reference implementation. http://cryptolib.com/ciphers/hitag2/, 2007.