Remote Keyless Entry Security

By Jake Hellman, ECE '16

Convenience is a driving force that led to the initial adoption of Remote Keyless Entry (RKE) systems for accessing garages, vehicles, and other secure locations. Initial iterations were subject to several issues, but later developments have led to increased security. The Orange Team's project adds a layer of security on top of current RKE systems.

Motivations for RKE Systems

Convenience has always been paramount for consumers. The automobile market is continually evolving to provide more features to improve the user experience. One notable feature which has been in use for several decades is Remote Keyless Entry (RKE). With a RKE system, users do not need to use a physical key to access their vehicles. As these systems have evolved, they have introduced features such as vehicle start functionality and increased security. However, security remains a concern as nearly all wireless systems can be hacked. These vulnerabilities have led to novel methods for accessing vehicles but the problem has not been entirely solved. As current vehicle design increasingly integrates electronics and connectivity to the Internet of Things (IoT), both engineers and consumers must consider the potential security risks that these technologies produce.

Early RKE Technologies

Wireless systems used to control access to secure locations were in development as early as the 1930s ("Widely Separated," 1931). The earliest iterations of these systems were simple in operation and were not security conscious. RKE systems cover a wide range of applications including not only vehicle access, but also garage entry and home security. Early RKE systems were meant to open garage doors and were built upon a fixed code technique. These designs make use of a predefined access message that is known to both the transmitter and the receiver (Alrabady, 2005). If the receiver is within range of the transmitter and the received signal matches the access code, the desired action is performed.

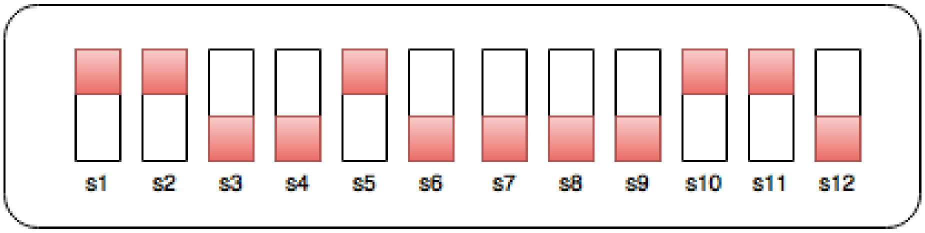

Figure 1. DIP switches used to set fixed code

Vulnerabilities

As these systems became more common, several security vulnerabilities became apparent and needed to be addressed. The first was that you and your neighbor might by chance have the same access code. Designs for fixed code systems often make use of Dual In-line Package (DIP) switches which can be used to set the specific access code (Figure 1). These designs are useful because the user can change his/her access code if it matches with a neighbor's. However, a larger security flaw is that the access code is always static. Because of the static message, an eavesdropper can record the message when the system is in use and replay it later to gain access to the system.

Development of Rolling Code

To resolve the vulnerability of the simple replay attack on a static access code, rolling code systems were developed. These systems keep a sequence counter which increments upon every attempt. The sequence counter is encrypted with an encryption key and the message is then transmitted. Because the sequence counter is always changed, the transmitted code is not

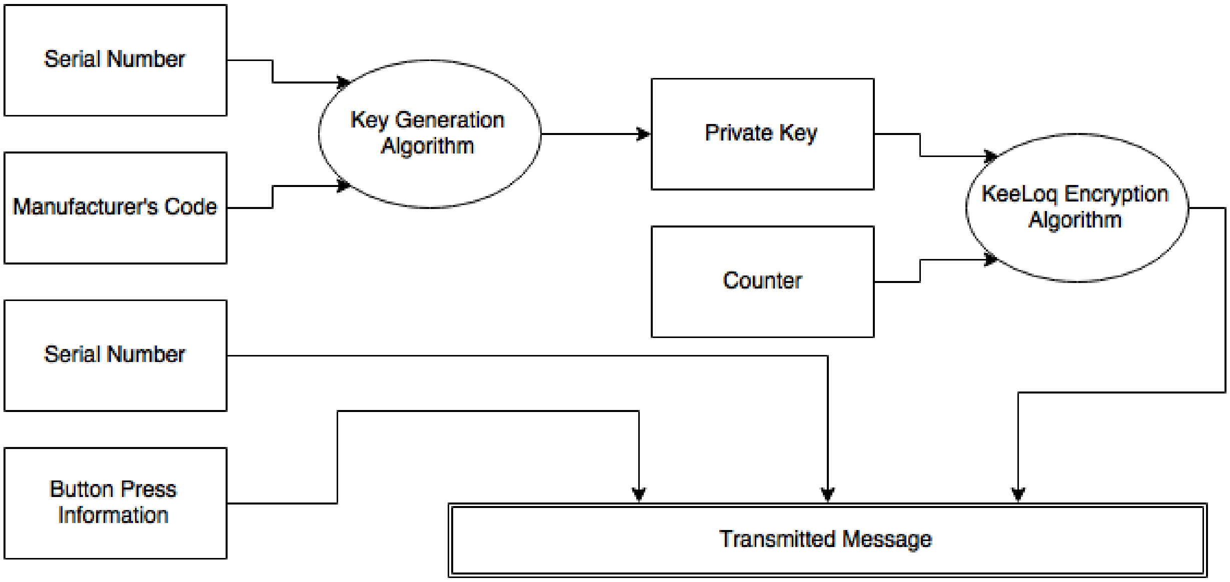

Figure 2. Generation of an encrypted key with the KeeLoq® Algorithm on the Microchip HCS301

static. Rolling code algorithms make use of an encryption scheme such that the messages sent back to back are uncorrelated. It is called rolling code because of the nature of the access code to change with every use. Several different encryption algorithms have been developed, but all are subject to several hacks which compromise the security of the system.

KeeLoq® Algorithm

KeeLoq® is the most common algorithm used to handle rolling code communication. It is used by many different automobile manufacturers including Chrysler, Daewoo, Fiat, GM, Honda, Jaguar, Toyota, Volvo, Volkswagen, and more (Coutois, 2008). A common chip used to implement the KeeLoq algorithm is the Microchip HCS301. Figure 2 shows the generation of the encrypted key to be transmitted. The chip uses the KeeLoq algorithm to encrypt the private key and counter. The transmitted message is then assembled so that the receiver will be able to decipher the access code. If the serial number matches one of the known transmitters then the counter value is extracted from the received encrypted message using the private key and is checked with the counter in the receiver. If the two counter values are within 16, then the desired action is performed (Microchip, 2001).

Vulnerabilities to Attacks

Brute Force Attack

The most obvious method of attack is to try and guess the code. Doing so is challenging however as many different possible bit combinations exist and no correlation exists between one code and the next. To break a rolling code system via a brute force attack requires trying different codes until one works successfully (Hu, 2009). While this attack is the simplest to carry out, it may take a very long time to work correctly and there is no promise that it ever will. Considering an encrypted transmission length of 24 bits and assuming that the car only allows 10 tries per second, it would take on average about 20 days to guess the code correctly (Alrabady, 2002).

Jam, Intercept, and Replay Attack

Another attack on rolling code systems known as jam, intercept, and replay can be used to gain access to the vehicle (Figure 3). In this kind of attack, the thief waits near the vehicle while the owner locks or unlocks it. When the owner uses his/her key the thief broadcasts noise so as to stop the transmission from making it to the vehicle. At the same time, the thief collects the message (by removing the noise that was introduced) and stores the message for later use. At this point, the owner of the vehicle notices that the key has not worked and he/she presses the button once more.

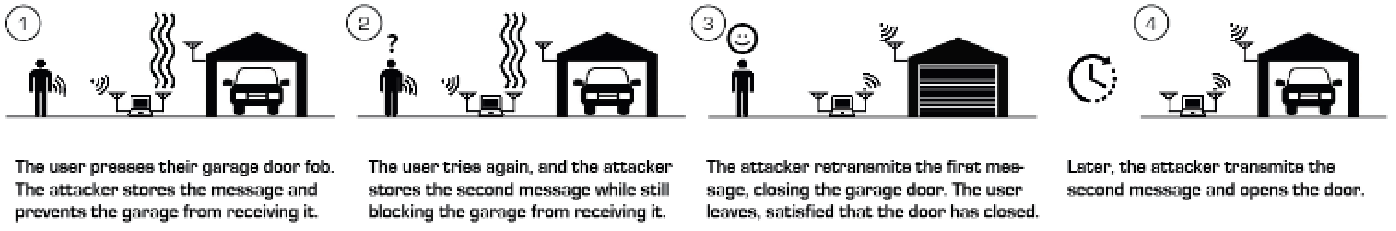
1
The user presses their garage door fob. The attacker stores the message and prevents the garage from receiving it.
2
?
The user tries again, and the attacker stores the second message while still blocking the garage from receiving it.
3
The attacker retransmits the first message, closing the garage door. The user leaves, satisfied that the door has closed.
4
Later, the attacker transmits the second message and opens the door.

Figure 2. Generation of an encrypted key with the KeeLoq® Algorithm on the Microchip HCS301

Again, the thief broadcasts noise and collects the second signal broadcast by the key. Simultaneously, the thief rebroadcasts the first message that was stored. The owner then observes the vehicle perform the desired action and thinks nothing of the failure. However, the thief now has another encrypted message that can be transmitted later and used to gain access to the vehicle. However, the thief now has another encrypted message that can be transmitted (Kamkar, 2015).

Conclusion

The Orange Team considered several RKE systems, which control access to vehicle entry, as well as their security issues. Although the security concerns are limited to accessing a vehicle, Kirk (2015) considers the potential to hack a vehicle and control every aspect of it. It is reasonable to consider a hacker gaining control of a vehicle and driving it while the owner is inside.

The Orange Team's project addresses the jam, intercept, and replay vulnerability and prevents the attack by adding a layer of security. As technology advances, it is important to consider the potential security flaws that are opened up by RKE systems that are implemented in vehicles for convenience. Although the security measures taken to prevent attacks have evolved over time, so too will the methods used to hack these systems. As our vehicles join the growing IoT, it is essential to address the security flaws which are created so that vehicles' safety and security remain a top priority.

References

"Widely Separated Inventors Develop Systems to Open Garage Doors with Radio Impulses." (1931, February). *Popular Science Monthly*, pp.32. Retrieved from https://books.google.com/books?id=3ScDAAAAMBAJ

Alrabady, A.I. (2002). *Security of Passive Access Vehicle* (Doctoral dissertation). Retrieved from http://www.ece.eng.wayne.edu/~smahmud/MyStudents/Dissertation_Ansaf.pdf

Alrabady, A.I., & Mahmud, S.M. (2005) Analysis of attacks against the security of keyless-entry systems for vehicles and suggestions for improved designs. *IEEE Transactions on Vehicular Technology, 52*(1), 41-50. doi: 10.1109/TVT.2004.838829

Coutois, N.T., & Bard, G.V., & Wagner, D. (2008) Algebraic and slide attacks on Keeloq. *Lecture Notes in Computer Science, 5086,* 97-115. doi: 10.1007/978-3-540-71039-4_6

Hu, Y., & Zhang, Y. & Sun, B. (2009) Design of RKE System Based on KEELOQ Encryption Technology. *Proceedings of the 2009 International Conference on Artificial Intelligence and Computational Intelligence,* 324-7. doi: 10.1109/AICI.2009.99

Kamkar, S. (2015). *Drive It Like You Hacked It.* Presentation, DEFCON 23.

Kirk, R. (2015) Cars of the future: the Internet of Things in the automotive industry. *Network Security, 2015*(9), 16-18. doi: 10.1016/S1353-4858(15)30081-7

Microchip. (2001). *KeeLoq Code Hopping Encoder.* Retrieved from http://ww1.microchip.com/downloads/en/devicedoc/21143b.pdf