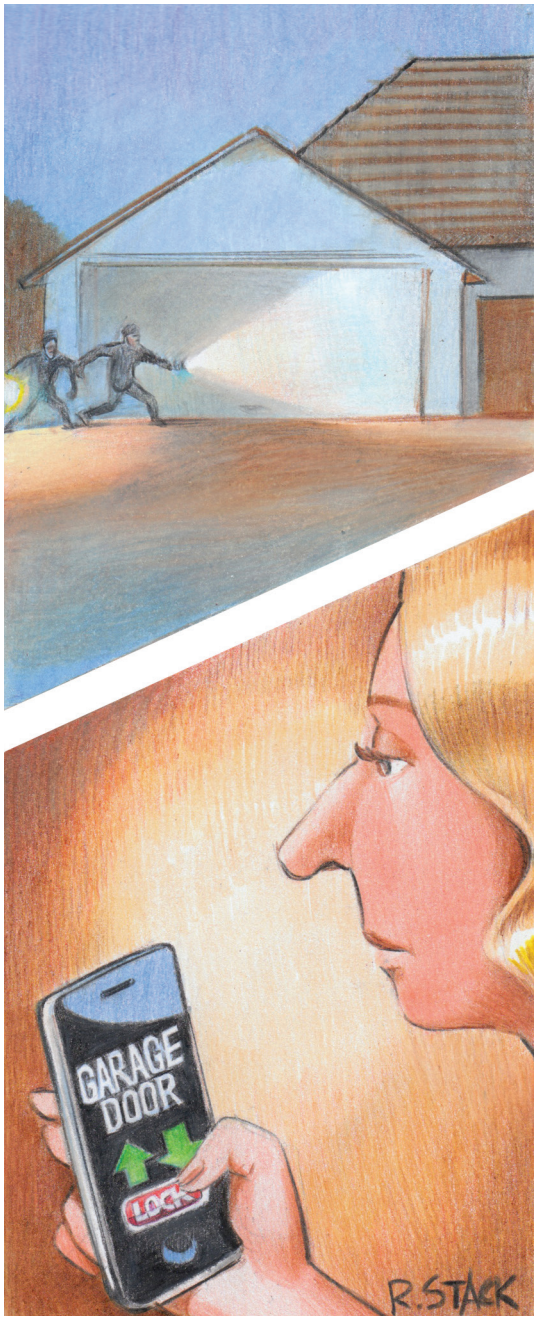
Editor: Jonathan Margulies, jonathan@qmulos.com

Garage Door Openers:
An Internet of Things Case Study

Jonathan Margulies | Qmulos



Early last year, my garage door opener's motor died. While researching potential replacement units, I focused on Chamberlain's products because they had a reputation for high quality. Once I settled on a model, I noticed another option: for a little more money, Chamberlain would include the MyQ Internet Gateway, its new system for monitoring and controlling the opener via the Internet. Curiosity got the best of me, so I went for it.

After installing the opener, the MyQ languished in my closet for months. I loved the idea of getting an alert if I left my garage door open—I can't count the number of times I've turned around five minutes after leaving the house to double-check that it was closed—but I felt sure there would be a security flaw in the MyQ that would make me worse off. The emergence of the Internet of Things (IoT) has turned trusted, long-standing companies into unwitting network attack vectors.[1]

But the MyQ is different from—and more impactful than—other IoT devices: it controls access to my house. That got me thinking: What if I wanted to solve this problem from the ground up? How would I design an Internet-connected garage door opener ("IoT opener") to be adequately secure? Is it possible?

The Ground Rules

First, I define "adequately secure" to mean no less secure than the traditional rolling-code garage door opener. Because that's the system the MyQ is replacing, it seems like the right standard.

Second, due to space constraints, I focus only on design. The usual implementation caveats—for example, the need for well-written code and correct use of encryption libraries—still apply, but I won't address them in detail.

Third, I assume the same basic set of features the MyQ offers: a user can open or close a garage door via the Internet from a smartphone or computer and receive emails or push notifications when the door's status changes.

Standard Garage Door Openers

If the standard is the security of rolling-code openers, we first need to understand how those openers work.

The most popular rolling-code implementation is a product called KeeLoq, a lightweight block cipher that generates codes based on a cryptographic key and a counter (www.webcitation.org/6ZZYZpH2n). When a user syncs a remote control with a garage door opener, the remote control begins to generate the same codes, in the same order, as that opener. Thereafter, when a user pushes the "open" button, the remote control increments its counter, generates a new code, and broadcasts that code wirelessly. When the opener receives a code, it checks the code against the next 256 codes in its

Copublished by the IEEE Computer and Reliability Societies

queue. (Checking against so many possible codes helps ensure that the remote control and the opener don't lose sync when a user presses the button outside the opener's receiving range.) If the code is a match, the opener increments its counter to just above the matching code and opens the door. In addition to using remote controls, some users mount keypads in front of their garages that similarly sync with the openers; these keypads broadcast a code when a user correctly enters a numeric password.

The simplest way for attackers to open a rolling-code garage door opener is to sync it to a new remote control. Replacement remote controls are available at just about any hardware store, and syncing them requires only a few minutes alone in the garage. A similarly easy option is to go after the keypad by spying on the user, or deduce or brute-force the code. A third option is a physical attack. Most openers include an emergency release rope just inside the door. If an attacker can slip a wire hanger above the door and latch onto that rope, a skilled tug can unlock the door. The final option for attacking traditional openers is to go after the rolling-code mechanism itself. Over the past decade, several researchers have developed methods to derive a KeeLoq key given access to a working, synced remote control.[2–4] A simpler but less effective approach is to sniff a code over the air from a remote control by pushing the "open" button outside the opener's range, and then using that code before the owner comes home (at which point, that code will expire).

All these attacks require close proximity to either the garage or the remote control and are sufficiently difficult that virtually all intruders prefer to break a window, force a door open, or pick a lock. But where are those intruders? Why aren't they taking advantage of the universally weak security of modern suburban homes? As podcaster Roman Mars eloquently observed, "locks have become a social construct as much as they are a mechanical construct" (http://99percentinvisible.org/episode/perfect-security). Garage door openers only need to be secure enough to let passersby know we don't want them to come in.

Openers and the Internet of Things

Exposing garage door openers to the Internet might make them such easy targets as to pose a real risk. What if an attacker could indiscriminately send open commands to any opener? What if every time an email account is hacked, the hacker is given a clear path to find the user's home address and credentials to open that user's garage? What if an attacker managed to download a whole database of user credentials for IoT openers? Any of these possibilities would make home intrusions so easy as to be inevitable. We're starting to see this evolution with cars that use Bluetooth keys; it's become so easy and cheap to break into some of them that insurers have started demanding additional security measures.[5]

Exposing garage door openers to the Internet might make them such easy targets as to pose a real risk.

But before I can delve deeper into how those types of attacks might happen on IoT openers, I must first address a key architectural question: Will the opener authenticate and authorize the user, or will a cloud service do so on the opener's behalf? When I bought the MyQ, I hoped it would be the former, as I had grand plans for putting the opener on its own private virtual LAN and having my wife use a virtual private network to reach it, because who would trust a garage door opener company with network access control? But I was naive.

The IoT industry has clearly decided that having a central service act as a clearinghouse for authentication, authorization, and commands is a must, and it's easy to see why: it frees them from having to worry about configuring home routers, setting up dynamic DNS for when customers' IP addresses change, or having access to all of a user's relevant data when those inevitable tech support calls come in. The problem is that the cloud service opens another attack surface, and a big one: instead of having to hack a single IoT opener at a time, attackers can try to hack them all through the cloud service. It's a single point of failure for authentication, integrity, and availability. Indeed, MyQ experienced an unplanned four-hour disruption in late April that affected all users, and I doubt it will be the last.

Cloud Service Authentication

Using a cloud service as a central hub isn't the security decision I would have made, but it seems like such a foregone conclusion that I'll treat it as an assumption for the remainder of this article. Authentication at both the cloud service and the opener is of paramount concern. The obvious way for the cloud service to authenticate to the opener is with a certificate, but what happens when attackers compromise the private key behind that certificate? This could allow attackers to send arbitrary commands to any opener, or something much worse: if the certificate can be used to force over-the-air software updates, attackers

Chamberlain Statement

The following statement was sent to the author on 12 June 2015:

> Chamberlain built its MyQ technology—as we have all of our products for more than 40 years—on a foundation of safety and security. We have an aggressive product roadmap that includes continuous security updates, including the feature recommendations noted in the IEEE [*Security & Privacy*] story.
>
> Specifically, Chamberlain will upgrade password requirements in the second half of this year [2015], and is looking at the best ways to implement two-factor authentication based on our user needs and their usage scenarios. We also plan to introduce multi-user access so that account administration details and log-in credentials are secure to only one account owner, while allowing garage door access to other users. This will include assigning different levels of permissions based on user access level, times of day, days of week, etc.
>
> In addition to the specific updates noted above, we combine our own team's expertise in security technologies with reputable third-party security firms to audit our systems on an ongoing basis. Our continuous security updates and processes include using industry standard encryption, applying the latest security techniques, and periodic security testing with respected outside services. We also recommend IoT technology leaders continuously advise their customers on how to maximize the security of their home Wi-Fi network, which are critical gateways to device security for consumers. Chamberlain's brand and reputation are built on a heritage of delivering safe and secure products to consumers; we take the safety and security of the smart home very seriously.

could gain control of the whole system. For this reason, software updates should be user initiated and openers should regularly check for certificate revocation.

Authentication at the cloud service is more complicated. Perhaps the worst-case scenario is when attackers download the password database, as in the famous attacks on Sony's PlayStation Network (www.cnet.com/news/playstation-network-still-offline-after-suspected-attack) and LinkedIn (http://money.cnn.com/2012/06/06/technology/linkedin-password-hack). If the passwords aren't sufficiently complex to stymie brute-force attacks or aren't encoded by an adequate key derivation function (such as bcrypt), user account takeover becomes trivial.

What about password reset, the issue behind attacks on countless celebrities' email and Apple iCloud accounts? Google recently published research suggesting that security questions are insufficient to protect accounts.[6] As of this writing, MyQ uses email for password reset, which, in this case, seems like a terrible idea: any time attackers hijack an email account, they can search for emails containing the term "MyQ" to determine whether the user has a MyQ account, and then search for shipping information to determine the user's home address. Ironically, just about the only personal information the MyQ website asked for was my home address, which is the information they should least want to have on file. This combination of information allows attackers to build databases of locations of vulnerable openers.

An opener is actually an interesting case from a password reset perspective in that it has an unusual security feature: it never moves. That means a user can't lose it, and it would therefore be reasonable for Chamberlain to require users to electronically prove they have possession of it (for example, the opener could display a code that rolls every few seconds). Such a feature would also help new homebuyers prove transfer of opener ownership.

Potential Security Improvements

The other side of this discussion is the security improvements networked openers might offer. One improvement is *two-factor authentication* (2FA). Many of the problems outlined in this article can be mitigated by 2FA and, in a system that's already so reliant on smartphones, users are already carrying the obvious second factor in their pockets. A second improvement is policy-based access control. This can be useful in several ways:

- allowing multiple user accounts to control the door, but only one to administer it;
- allowing administration from specific devices only;
- restricting certain accounts (such as caregiver or contractor) to operating the door only during business hours; and
- creating time-limited guest accounts.

These policies could be easy to administer through a Web interface.

A third improvement is a more granular alerting system. I'd be much more interested in knowing that a new user or remote control was given access to my opener, or that the door opened in the middle of the night, than in knowing that my wife opened the door at 5 pm.

Convenience will continue to drive companies that lack information security expertise to build IoT devices, and consumers to buy them. Ideally, this new IoT world would be built on a few competing platforms by people who understand and can address the security risks—Apple, Google, and Facebook have all recently launched the beginnings of IoT platforms—rather than inexperienced companies rolling their own authentication, authorization, and communication code. Although the risks and functions change from one IoT device to another, they all need the same basic security infrastructure: a way for users to authenticate, two-way mapping between users and devices, access policy creation and enforcement, logging and alerting capability, and secure communication. These problems are largely solved; they just need to be made easy for IoT developers to use. When that happens, and the right security features are in place, I think IoT openers could become difficult enough to attack that no one would bother with them, just like the openers most of us have today. In the meantime, my MyQ is going back in the closet. ■

Author's Note

I contacted Chamberlain after writing this article, and a representative responded with an outline of plans for addressing some of the concerns in this article. See the sidebar for the response.

References

1. J. Fontana, "Belkin Patches Vulnerabilities in WeMo Devices," ZDNet, 19 Feb. 2014; www.zdnet.com/article/belkin-patches-vulnerabilities-in-wemo-devices.
2. S. Indesteege et al., "A Practical Attack on KeeLoq," *Proc. 27th Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques* (EUROCRYPT 08), 2008, pp. 1–18.
3. A. Bogdanov, "Cryptanalysis of the KeeLoq Block Cipher," Cryptology ePrint Archive, report 2007/55; https://eprint.iacr.org/2007/055.pdf.
4. I. Sheetrit and A. Wool, "Cryptanalysis of KeeLoq Code-Hopping Using a Single FPGA," Cryptology ePrint Archive, report 2011/242; https://eprint.iacr.org/2011/242.pdf.
5. H. Osborne, "Thieves Target Luxury Range Rovers with Keyless Locking Systems," *Guardian*, 27 Oct. 2014; www.theguardian.com/money/2014/oct/27/thieves-range-rover-keyless-locking.
6. J. Bonneau et al., "Secrets, Lies, and Account Recovery: Lessons from the Use of Personal Knowledge Questions at Google," Google Research Archive, report 43783; http://research.google.com/pubs/pub43783.html.

Jonathan Margulies is the chief technology officer at Qmulos. Contact him at jonathan@qmulos.com or follow him on Twitter @unsaltedhash.