User Authentication for the Internet of Things*

Frank Stajano[1] and Mark Lomas[2]

[1] University of Cambridge
[2] Capgemini

Abstract. The Internet of Things is coming to fruition, but current commercial offerings are dramatically insecure. The problem is not that many individual devices are vulnerable, but that there are billions of such devices and yet no concerted plan to make them secure. Since the IoT is here to stay, and will pervade the fabric of our society in a way that will make it impossible for any individual to opt out without retiring to a cave as a hermit, we must address the problem structurally, rather than with local band-aid fixes. This short position paper presents the basic requirements for a scalable user authentication solution for the Internet of Things. We hope it will stimulate a discussion leading to a coherent user authentication architecture for IoT. Our vision is that even the lowliest and most inexpensive of IoT devices ought to offer such basic security properties, but this will only happen if they are agreed upon and designed in from the start.

1 Introduction

Having been talked about under a variety of names for two or three decades, the Internet of Things is finally coming to fruition. What is still missing, though, is a proper security architecture for it. That currently deployed IoT devices are insecure is testified by the plethora of vulnerabilities that are discovered and exploited daily[3]: clearly "features" are higher priority than "security" in the eyes of the purchasers—and therefore of the manufacturers. But we are talking here of a more structural problem: not "this device is insecure" but "there is no strategic plan and no accepted blueprint to make IoT devices secure". We should also bear in mind that if purchasers do not understand security vulnerabilities, or cannot articulate their understanding, then manufacturers are unlikely to address them.

There is some role for government regulation. Indeed, the currently ongoing "Secure by Design" initiative in the UK[4], and corresponding ones in other countries, aims to establish a certification and labelling scheme that would assure to consumers that a certain IoT product is free from basic vulnerabilities.

[*] Revision 16 of 2018-06-18 12:02:18 +0100 (Mon, 18 Jun 2018).

[3] The Mirai botnet, which attacks IoT Linux-based IP cameras and home routers, is the one that most people remember, at the time of writing, but it is by no means an isolated incident.

[4] https://www.gov.uk/government/publications/secure-by-design

In this position paper we do not address IoT security in general: instead we focus specifically on the problem of *user authentication*, addressing which is a pre-requisite of any security architecture insofar as the three crucial security properties of Confidentiality, Integrity and Availability can only be defined in terms of the distinction between authorized and unauthorized users of the system[5]. However, we should not be misled by the word “authorized”; authorized users may misbehave.

2 The problem

Traditionally, user authentication has been addressed with usernames and passwords. This technology is strongly entrenched and difficult to replace, but it is clearly showing its structural limitations in today’s computing context, where even non-computer-experts have to wrestle with dozens of distinct accounts. Although passwords continue to dominate on the web, they would be pathetically inadequate for user authentication to the Internet of Things.

We define an IoT device as:

- an Internet-connected computing device,
- often (but not necessarily) embedded in an everyday object, that
- does not offer a traditional keyboard / screen / mouse UI, and
- connects to the network directly (rather than as a peripheral of a computer).

Examples of such devices include a pet monitoring IP camera, a smartwatch, an IP-controlled central heating system controller, an IP lightbulb, the mythical Internet-connected Refrigerator and, with a tip of the hat to Stuxnet, a remotely controllable[6] uranium enrichment turbine.

Each IoT device can be modelled as an object with methods. Security for IoT is primarily about ensuring that only authorized principals can invoke the methods offered by the object: only I and designated family members, but not a would-be burglar or a stalker, should be able to see through the lens and listen to the microphone of the cat-camera. This clearly requires a definition of “authorized principals”, and the ability for the IoT device to distinguish authorized from unauthorized principals. It may also be useful to consider limitations on authorization, especially in the context of delegation: I might permit a neighbour to keep an eye on my cat while I am on holiday but don’t want them to watch me while I am at home.

Passwords are inadequate for this job, both because each of us will have many more IoT devices than computers (and therefore the already unworkable

[5] Confidentiality is the property of a system in which certain information may only be read by authorized users. Integrity is the property of a system in which certain information may only be altered by authorized users, and in compliance with designated constraints. Availability is the property of a system to which authorized users have access, with designated guarantees, regardless of attempts by unauthorized users to deny such access.

[6] Don’t believe in airgaps.

proposition of a different password per account will never be able to scale) and because IoT devices tend not to have a UI suited to password input.

As first steps towards a solution we seek to define the requirements of a valid IoT user authentication strategy. What are the limitations of existing systems that we would like to address?

3 IoT user authentication

A valid strategy for IoT user authentication must:

1. scale to thousands of controlled devices (in particular without burdening the user's memory);
2. be suitable for computer-illiterate people (certainly as far as the frequent "login" action is concerned; but ideally also for the comparatively less frequent "account setup" and "device registration" actions);
3. protect, within reason, against impersonation;
4. protect user privacy;
5. ensure that cracking one device does not imply cracking my other devices—and that sharing key material with a device does not assist the manufacturer of that device in cracking my other devices;
6. work even when I lose Internet connectivity.

Desirable additional features include the following.

7. A usable and expressive way of defining who is allowed to use what methods and in what ways (hard and still unsolved research problem).
8. Revocable delegation of a subset of the user's rights.
9. A domain should be able to accept credentials from another, for example when granting my neighbour access to my cat camera.

It is well known that users cannot remember large numbers of secrets. If forced to supply a password for each device they will share the same password with many devices; more sophisticated users may use a password manager to store passwords. However, it is not convenient to use a password manager to authenticate to my fridge or my watch. A physical device holding my credentials and capable of dispensing them wirelessly to the desired verifier when needed, like Pico[7], may be a more useful proposition.

Needham and Schroeder persuaded many of us to adopt authentication protocols and Kerberos popularised that suggestion. Quite correctly they observed that, just because you share resources with other people, that does not mean that you completely trust them. Passwords used to pass across networks in clear. Authentication protocols can protect authentication information against eavesdroppers while in transit.

Although Needham and Schroeder picked on a loose thread they did not continue to unravel the torn jumper. Even when Kerberos is used to protect

[7] SPW 2011, LNCS 7114.

network traffic, an eavesdropper is still able to determine which devices are communicating and when. Or, returning to our main theme, an eavesdropper may be able to recognise that I am controlling my heating system from my office and whether or not my motion-sensitive cat camera is sending data from my sitting room. Either piece of information may be useful to potential burglars. While confidentiality is all about preventing unauthorized access to the data, privacy is also concerned with the metadata.

Mobile devices are often vulnerable to a similar problem. While it may be convenient to unlock my computer using my phone or watch, I would prefer that those devices didn't act as small beacons signalling that I am not at home while typing this paper. How nice things would be if I were able to limit reception of such signals to just the computer I was unlocking—but that's not the way things usually work.

A proximity token like a "modern" wireless car key is not resistant to impersonation: whoever finds it can use it. Some devices compensate for this by requiring a PIN when you first put them on (e.g. the Apple Watch). While biometrics may help with this, some manufacturers compensate for the false accept rate by insisting on a PIN or password at intervals.

The kind of system we envisage as useful is based on the following design principles:

1. Environments (homes, offices, nuclear power plants, etc) are conveniently divided into domains, similar to Active Directory. Each domain will contain at least one authentication server.
2. We anticipate that many people will not want the complication of a local authentication server at home, so we imagine these might be provided remotely, perhaps as a cloud-based service. However, larger environments may prefer to control their own server so it should also be possible to deploy this locally. We also like to imagine using our devices on a cruise ship which may have plentiful local network capacity but woeful Internet access. We suggest that cruise ships ought to carry authentication servers.
3. To preserve privacy of communicating parties, authentication transactions should not identify participants in clear, although we are willing to permit partial identification as a compromise. For example, it would be very difficult to hide all of the activity in my house while devices there are communicating over the Internet, but it may be sufficient to combine the data from many devices and to transmit spurious data so it is not clear when my cat camera is active (Ron Rivest suggested Chaffing and Winnowing for this purpose).
4. There are facilities for deploying large batches of IoT devices in one go. The symmetric key version is what SecureID does (here is a box of 50 tokens, each with a serial number written on it, and here is a table mapping these serial numbers to the secret keys embedded in the tokens). There is an equivalent but somewhat more secure way of doing it with public key crypto (here is a box of 50 tokens, each with a private key embedded in it that was generated on the device and that no-one else has ever seen, and the serial number is a hash of the public key; here is a table of the public keys and their hashes).

We believe it is relatively easy to identify well-known existing technical security mechanisms that solve each of the above requirements in isolation; that it is somewhat more challenging to combine them so as to solve all the above requirements simultaneously; and that the really serious problem of authentication for IoT is how to do that in a way that remains easy to use for ordinary people.

Acknowledgements

Frank Stajano is grateful to the European Research Council for funding the past five years of his research on user authentication through grant StG 307224 (Pico).