A Brief Survey of Imprinting Options for Constrained Devices

Eric Rescorla
RTFM, Inc.
ekr@rtfm.com

March 19, 2012

1 Introduction

Constrained devices such as thermostats, light bulbs, etc. provide a number of communications security challenges. First, because they have minimal computing power, many cryptographic operations which are normal in more sophisticated devices are expensive, sometimes prohibitively so. Second, because the devices have constrained user interfaces it can be challenging to introduce them into a new network. This document focuses on the second problem, which is often called "imprinting".

2 Problem Overview and Threat Model

The setting for this problem is that we have a sophisticated "base station", i.e., a general purpose computer with a full user interface which we can access securely, e.g., by a web-based console. We want to add a new constrained node such as a sensor, a light switch, etc. For concreteness, we will refer to that element as a "device" for the rest of this paper. Our objective is to establish secure communications between the devices. More concretely:

- The device knows that it is talking to the right base station
- The base station knows it is talking to the right device
- Communications between the base station and the device are protected against viewing and tampering by third parties.

These are of course well-known COMSEC problems and are trivially solved as long as we can arrange that either:

- The device and base station share a secret.
- The device and the base station each have an asymmetric key pair and the other side can verify the peer's public key.
- The device and the base station can verify that they have computed the same value in more or less real time.

Provided that either of these two conditions apply, then we can bootstrap ourselves up to a secure channel using a variety of well-known security techniques. Indeed, there are well-known techniques for achieving the first condition given the second or third, and indeed most COMSEC protocols first attempt to compute a high entropy shared secret and start from there. Note, however, that we can start with a low entropy shared secret and bootstrap it up to a high entropy shared secret using a *Passowd Authenticated Key Agreement* (PAKE) protocol (though these involve more computational power). Moreover, as long as the device and base station can establish such a secret once, then they can use it to authenticate all future operations. Hence imprinting consists of establishing that secret and need only be done once.

We assume that the base station and device communicate over an insecure wireless network (note that introducing the device to the network is itself an imprinting problem, so just reduces to our existing problem), and so the usual Internet threat model [RK03] in which the attacker has complete control of the network applies here. We also assume that the attacker can obtain their own base station and devices which would in principle be able to pair with the victim's network elements.

Probably the hardest variant of this problem is one in which the device has no user interface at all: it cannot display any values and the user cannot enter any beyond an activation signal such as inserting a battery or plugging it in. It can, however, have a sticker attached to it with a secret or a serial number.

3 Solution Space

This section provides an overview of the major known approaches.

3.1 Leap-of-Faith/Pushbutton Configuration

Probably the best known approach is "push-button configuration". Effectively, the administrator adds the device to the network and pushes some button on the base station at roughly the same time as he activates the device. The device and the base station then do an unauthenticated asymmetric key exchange (e.g., Diffie-Hellman) and assume that this initial exchange is not under attack. From that point forward, key continuity is used to avoid an attacker inserting himself into the middle.

The major advantage of this design is that it is simple to implement and as long as the initial exchange is not attacked, then it is secure. The near-simultaneous activation of the devices prevents an attacker from creating his own associations except during the time period while a legitimate device is being imprinted. However, an attacker with good control of the network can still mount a MITM attack at the time of first association. (See Section 3.5 for a related approach which does not have this drawback but depends on tight control of the physical layer.)

3.2 Central Provisioning

At the other end of the spectrum, we can centrally provision both the device and the base station. For instance, the base station and the device can come from the same manufacturer and whenever a device is purchased the manufacturer can automatically populate a database entry with the pairing of device to base station. (There are a number of techniques available to do this without privacy leakage).

The major drawback of this system, however, is that it requires a lot of coordination between the point of sale and the manufacturer in order to populate the database. This works for some kinds of devices (e.g., power meters) but not for most consumer facing devices such as light switches. Home Depot is not going to have their sales system report every sale back to the manufacturer together with the base station id of the user.

3.3 Devices with Displays

If the device has a display—and by assumption the base station has one—then we can use techniques based on matching values between the device and the base station. Roughly speaking, the workflow here is the same as in Section 3.1, except that the base station and the device both display some value (e.g., a hex string) and the user must compare them and verify that they are the same. If they are not, he aborts the pairing process (obviously the device must have some mechanism for doing this, which is not always straightforward).

Somewhat surprisingly, it is not necessary that the value to compare be large. For instance, Vaudenay [Vau05] describes a protocol that allow the attacker one guess per pairing action and so are secure with very short strings.

This design is not without drawbacks. First, it requires that the device have some sort of display (though perhaps blinking LED would be sufficient if the user were willing to decode flashes). Many devices will have no such display. In addition, it must be possible to abort a candidate pairing if attack is detected, and therefore it is only convenient if the user has more or less simultaneous access to both device and base station

in order to check the authentication string. Finally, the mechanism is insecure if the administrator does not check the string and the device, at least, has no real way of knowing whether he did. If the administrator does not check, this mechanism reduces to PBC/leap-of-faith as described in Section 3.1. This can be viewed as either a bug or a feature.

3.4 Devices With Labels

Clearly, many devices will not have a display of any kind. However, in many cases it is possible to affix a label with some fixed value. Generally, this value will be one of two things:

- A secret value.
- A public value that is tied to some secret value known the device (e.g., it's an asymmetric key), such as a key fingerprint.

In either case, there needs to be some mechanism to load the value into the base station (the device already has it.) Potential mechanisms include:

- A character string that users must type in.
- A bar code, QR code, or some other machine readable/visual indicator.
- An RFID tag.

Obviously, typing is only viable when the value is short.

3.4.1 Labels with Public Keys

One approach is to print a value that corresponds to the device's public key on the device. This value could be a hash of the public key or a serial number which allows lookup of the public key in a master database. Note that the hash must be long to prevent exhaustive search attacks for a hash preimage, whereas the serial number may be relatively short since the database guarantees preimage freeness.

In either case, this mechanism allows the base station to authenticate the device but does not provide for authentication of the base station. In cases where there is a risk of base station substitution, then, this approach is not entirely satisfactory.

3.4.2 Labels with Secrets

Alternatively, one can print a secret value on the device. This secret value can be used to mutually authenticate the device and the base station. If a PAKE protocol is used, the value can be short enough to type, e.g., as little as 4-8 characters (20-40 bits) if some sort of rate limiting is used to prevent repeated reauthentication attempts (which are the only mechanism of searching the key space.)[1] Note that the use of a PAKE system means that public key technology must be used, which can be a problem in some CPU-constrained devices.

The primary concern with a secret value system is that the value must be kept secret or an attacker can impersonate the base station to the device and vice versa. This makes mechanisms where the label is on the outside of the box rather than the device problematic, and RFID tag approaches doubly so. One might imagine mechanisms to prevent masss scanning on store shelves such as having the label concealed within a sticker that must be peeled back. (A similar approach with metallizeds stickers might work for RFID.) This obviously would not provide perfect security but might be sufficient in many applications.

[1] Note that you cannot use a public database here, and a private database must have lookup keys at least as long as the secrets.

3.5 Physical-Layer Approaches

The discussion above has been couched in terms of standard network models that are agnostic to the underlying technology. Recently, however Gollakota et al. [GAZK11] have described an approach that takes advantage of the properties of 802.11 wireless networks to prevent MITM attacks. Briefly, the system uses Diffie-Hellman key exchange but protects it by wrapping the DH share in a jam-detectable frame. Thus, a MITM attack can be detected by seeing that there are two candidate DH shares and thus something is wrong. Obviously, this is of use only in specific network environments and it is too soon to tell whether it is secure. Here too, we have to worry about the computational cost of the public key operations.

4 Summary

As should be clear from this brief discussion, there is no ideal solution to the problem of imprinting a constrained device/base-station pair in an insecure network. Rather, we have a family of solutions each of which is plausible—though not without flaws—in some environments and nonviable in others. Selecting the best solution is therefore an exercise in understanding your threat model and making the appropriate tradeoffs.

References

[GAZK11] Shyamnath Gollakota, Nabeel Ahmed, Nickolai Zeldovich, and Dina Katabi. Secure In-Band Wireless Pairing. In *20-th USENIX Security Symposium*, 2011.

[RK03] E. Rescorla and B. Korver. Guidelines for Writing RFC Text on Security Considerations. RFC 3552, July 2003.

[Vau05] Serge Vaudenay. Secure communications over insecure channels based on short authenticated strings. In Victor Shoup, editor, *CRYPTO*, volume 3621 of *Lecture Notes in Computer Science*, pages 309–326. Springer, 2005.