Connecting the Things to the Internet: An Evaluation of Four Configuration Strategies for Wi-Fi Devices with Minimal User Interfaces

Michael O. Jewell, Enrico Costanza, Jacob Kittley-Davies
Agents, Interaction and Complexity Group
School of Electronics and Computer Science
University of Southampton, UK
{moj,ec,jkd3g11}@ecs.soton.ac.uk

ABSTRACT
The availability of low-power Wi-Fi radio modules opens up opportunities to leverage the existing prevalent Wi-Fi infrastructure for large-scale trials and deployments of Ubicomp technology. In this paper we address the challenge of supporting end-users, especially when they are not technical experts, in connecting new low-power, low-cost Wi-Fi devices with very minimal UIs to an existing, secure Wi-Fi infrastructure. We report two usability studies through which 30 participants, with no formal technical training, compared 4 alternative configuration techniques, selected based on cost and consumption constraints, and on adoption in off-the-shelf products. Through an analysis of success rate and causes of failure, our results indicate that two techniques are noticeably more usable than others. These are a web-based configuration mechanism, where users connect to an access point on the Wi-Fi device, and one that makes use of a standard audio cable to connect a smartphone to the device to be configured.

Author Keywords
Internet-of-things; configuration; 802.11; deployment; user study.

ACM Classification Keywords
H.5.2. Information Interfaces and Presentation: User Interfaces

INTRODUCTION
Networking and computational devices, sensors, and actuators are becoming ever-increasingly available, miniaturized and inexpensive. At the same time, Internet connectivity and the Web are integrated into everyday life in most of the developed world, and increasingly so in developing countries. This combination promotes a growing adoption of Ubiquitous Computing outside controlled research environments. This technology is now seeing a new wave of popularity in the mass media and consumer market under the phrase "Internet of Things" (IoT). In particular, the introduction onto the market of low-power Wi-Fi (IEEE 802.11) radio modules[1] makes it possible to leverage the existing and prevalent Wi-Fi infrastructure to connect low-consumption, and even battery-powered, Wi-Fi devices to the Internet. Indeed, statistics from 2012 indicate Wi-Fi availability in a quarter of households worldwide[2]. This low-power technology combined with such a high level of coverage opens up opportunities for large-scale trials and actual deployment.

Our work is motivated by scenarios where devices may be installed by end-users directly, or by a "facilitator" who is not a technical expert. Such a facilitator may for example be an energy poverty advisor installing environmental sensors to better understand their clients' routines, as proposed by Fischer et al. [9], or a caregiver setting up an instrumented pill box to help improve their clients' medication compliance, as envisioned by de Oliveira et al. [7]. The facilitator would visit households of clients and install wireless devices, connecting them to an existing Wi-Fi infrastructure.

We are specifically interested in small devices, such as battery-powered wireless sensors, displays or actuators. These devices, hereafter referred to as 'Wi-Fi devices' for conciseness, include very minimal UIs (LEDs rather than graphic displays, single buttons rather than keyboards) making it difficult for a person to input information directly into them. Instead this task can be supported by an external device with a more complete UI such as a smartphone, a ubiquitous, always at-hand object. In practical terms, for a new Wi-Fi device to join an existing (secured) Wi-Fi network, the network name ("SSID") and passphrase must be somehow entered. The question we address, then, is: what is the best method to allow the user to transfer the Wi-Fi network name and passphrase from a smartphone to a Wi-Fi device with little or no UI? We constrained the design space to only use low cost hardware, to reflect the observation that our potential users would often have "to manage on a tight budget" [9].

To date, the Ubicomp community has contributed considerably to the design and implementation of systems and appli-

[1]E.g. the Microchip RN family or the GainSpan Wi-Fi can reportedly run off 2 AA batteries for over a year.

[2]`http://tinyurl.com/7d2h3b5`

UbiComp '15, September 7–11, 2015, Osaka, Japan.

ACM 978-1-4503-3574-4/15/09...$15.00.
http://dx.doi.org/10.1145/2750858.2807535

cations, as well as studies of their adoption and usage, with prototypes and field trials set up by researchers (e.g. [10, 7, 15, 34, 9]). In contrast, limited work has been done to address how to set up the necessary infrastructure (e.g. [6, 3, 31]). We believe that the configuration of Ubicomp systems by end users (especially those without technical training) poses a considerable interaction challenge not yet fully addressed.

To address this challenge, we report two usability studies through which 30 participants, with no formal technical training, compared a number of alternative techniques. The choice of configuration techniques was informed by technical constraints that would allow them to be realistically adopted on a Wi-Fi device, as well as by the techniques already integrated in off-the-shelf products. The results indicate, through an analysis of success rate and causes of failure, that some techniques are noticeably more usable than others. These are a web-based configuration mechanism, where a user connects to an access point on the Wi-Fi device, and one that makes use of a standard audio cable to connect the smartphone to the device to be configured. The participants' subjective preferences, elicited through a questionnaire, suggest that different configuration procedures can also affect the user experience.

RELATED WORK

Our research aims to evaluate how end users handle the initial network configuration of small Wi-Fi devices with minimal UI. As such, we are building upon prior research that has studied how householders create and maintain their home networks; approaches to simplify network configuration; HCI for Wireless Sensor Networks (WSNs); and existing technical work in configuration technologies, especially regarding device pairing.

Home Network Configuration

Prior research has investigated how householders maintain and interact with their home networks, especially with many purpose-specific devices now being added to them. The complexity of these networks is often hidden: Grinter [12] describes how cables are hidden behind and underneath furniture; equipment is hidden in cupboards and under couches; antennae are hidden in plants. This leads to difficulty in identifying sources of problems when troubleshooting. Some networked devices are integrated into established infrastructure and routines, or to reflect practical concerns in the home [30], and in-the-wild studies have found that there is typically a 'home technology driver' who takes responsibility for this [20]. It is therefore vital that consumers, even 'passive users', feel comfortable bringing and installing devices into their home network and routines [22]. While this existing research examines the positioning of off-the-shelf new hardware in the home, and how it is maintained following its installation, our work instead evaluates different interaction design options for incorporating a new device with a minimal UI into an existing network.

Infrastructure Simplification

Several studies have evaluated novel approaches to home network configuration by making modifications at an infrastructure level. MultiNet [3] uses a modified home router that can create virtual access points on demand. A usability experiment demonstrates that its setup took significantly less time than the solutions currently on the market, and users found the approach simple to learn. A similar study approach was used to evaluate Network-In-A-Box (NiaB) [1] and ICEbox [33]. Both systems use IR communication to exchange credentials between a laptop and a Wi-Fi router. For both of them, usability studies indicate that they outperform commercial alternatives in terms of task completion time (NiaB only), success rate and subjective preferences. Derthick et al. [8] evaluate the opportunities for appliance UIs, where appliances are augmented to run their own Wi-Fi hotspots, and serve web pages. This is a different issue than having a new device joining an existing network: indeed the authors state, 'getting an appliance onto a shared network in the first place is a problem, since credentials need to be provided to the appliance somehow.' This is the problem that we address. Similar to the approach by Derthick et al. we also make use of a smartphone to provide a 'more powerful UI than [the device's] physical controls provide.' In summary, these approaches address a different problem than the one we focus on as they modify the infrastructure to simplify its management. Instead our aim is to find the best way to take advantage of the existing infrastructure, as in many cases it may be impossible to change it.

Wireless Sensor Networks

In contrast to devices typically deployed on home networks, our Wi-Fi devices are low-cost, low-power, and have minimal UIs, so are akin to nodes deployed on Wireless Sensor Networks (WSNs). As such, we review the limited prior HCI research in this area.

SensorTune [6] offers an approach to the placement of wireless nodes in WSNs, with non-speech audio used to inform users as to the connectivity of the network that they are setting up. This method is specific to multi-hop mesh networks, with a specialised radio, and hence is quite different from our focus on Wi-Fi.

With Siftables, Merrill et al [21] investigate WSNs as user interfaces, applying principles from sensor network technologies to TUI research. However, their work is focussed on the opportunities of using an established WSN as a UI, rather than on interaction strategies for setting up wireless nodes.

Device Pairing

The concept of *device pairing* can be considered as a specific kind of wireless configuration. Pairing is the process of creating a secure bidirectional link between two wireless devices, so that data can subsequently be exchanged between them. The pairing process normally uses an out-of-band (OOB) channel, and it takes advantage of human sensory capabilities, such as audio, visual and tactile, to authenticate the wireless channel [16]. For example, when pairing a Bluetooth mouse with a computer, users may be required to push a button on the mouse when they see a prompt on the screen.

In our case we want to associate a Wi-Fi device with a Wi-Fi router, so that it can join a wireless network. This could be interpreted as pairing the Wi-Fi device with the router, but we make use of a third, external, device to handle the association. This extra device provides the UI that is not present or sufficient on the Wi-Fi device or the router. In contrast, prior work on pairing tends to assume that at least one of the devices will have a relatively rich UI (e.g. able to display or input text). However, in our work the interaction between the phone and the Wi-Fi device is similar to pairing, so the research in this area is reviewed by way of background.

While a cable provides no ambiguity about the device that is being paired [27], the majority of pairing approaches are wireless. These include infrared [29, 2] and NFC, used by Bluetooth OOB pairing [17]; audio techniques where melodies [26] or synthesized speech are compared [11]; and visual approaches, where one of the two devices has a camera and it decodes either a 2D barcode [19] or flashing LED [23] on the other device. Others are more physical, involving synchronized button pressing [25] or shaking of the devices [14, 18]. Chong et al. [4] report an overview of how users interact with devices when pairing low-fidelity prototypes. Some of these techniques are well suited for transmitting configuration details between the devices (e.g. infrared, bluetooth, flashing LED), while others are only appropriate for pairing (e.g. coordinated button pressing and shaking).

Recent work has also investigated the use of a smartphone as a mediator for pairing 'interface restricted devices' [28] with, in the simplest case, the phone generating a secret key and passing it to both devices. This allows for the pairing of devices that may not share the same pairing technology (e.g. one may have NFC, the other Bluetooth). In our work, it is possible that the Wi-Fi router will not have any built-in pairing functionality, so the smartphone only communicates with the Wi-Fi device.

Comprehensive evaluations of pairing techniques have been undertaken: Chong et al. [5] report a conceptual analysis and literature survey about "device association" and highlight that "more studies and new methodologies for understanding users are needed", which our work extends by comparing 4 configuration techniques. Kumar et al. [16] report a usability study of a variety of pairing techniques, but these are aimed at mobile phones with displays and input mechanisms, whereas our work focuses on devices with minimal UIs. The results of their study do suggest, however, that audio techniques are a good choice when displays are not available. HAPADEP [26] is an example of such an audio pairing approach. One device plays an audio sequence that represents the encoded version of its public key, while the other records and decodes it. The user then verifies that a 'slow' version of the recorded sequence matches that played by the other device. Compared to a text-to-speech approach with automatically-generated sentences, it was found to be an easy to use alternative to prior techniques, though unsuitable for noisy environments and requiring a microphone on the receiving device, and a speaker on both. These hardware requirements are not suitable for our low-cost and low-power Wi-Fi devices, and instead we use an audio transfer approach that uses a headphone jack connection.

Other researchers have investigated pairing approaches that use flashing lights. Saxena [23] found that a flashing LED on one device, paired with a camera on the other, would be a performant alternative to approaches that required screens showing barcode representations of public keys [19]. Systems for data transmission subsequent to pairing have also been developed to make use of flashing lights: LedTX [24] provides data transfer from a UbiLighter instrumented lighter, with a flashing LED on the lighter decoded by a Javascript program accessing a laptop's webcam; FlashLight [13] creates a bidirectional communication channel between a tabletop and mobile phone, using the built-in cameras and displays of the devices. The latter in fact uses this as both a means for pairing and data transmission, as the transmission cannot be hijacked without being visible to the user. While these systems have not been evaluated with users, the authors' proof of concept implementations suggest that a flashing light approach could be an option for the configuration of a Wi-Fi device. However, while we can assume that a smartphone has a suitable display for the purposes of transmission, a camera is not a feasible addition to the device. Instead, we evaluate an approach that uses a light sensor in its place.

CONFIGURING WI-FI DEVICES WITH MINIMAL UI

As described in the introduction, we are interested in taking advantage of existing, password protected, Wi-Fi networks, which are already available in many buildings, to connect new, low-power Ubicomp devices to the Internet. In practice, this means that users will need to input the Wi-Fi network name ("SSID") and passphrase – two alphanumeric sequences – into the devices. As stated above, our interest is on devices that have only minimal UI, and for which it would be impractical to add UI capabilities just for the task of initial configuration. It would not be cost-effective (or even sustainable) to include a screen on a networked sensor device, just for the purpose of configuration – an activity that is supposed to take place sporadically.

Some wireless routers [32] include a "Protected Setup Push Button Configuration" (WPS-PBC) feature, which is designed to simplify the connection of new devices to the network. While this technique would in principle be suitable to configure the type of devices we are interested in, in practice WPS-PBC is not available on all existing routers, and it has known security issues[3]. Moreover, even in situations where it is available, physical access to the router may be impractical, such as when the routers are installed out of reach (e.g. near the ceiling).

Our chosen approach, instead, is to take advantage of the keyboard and screen available on another device, such as a computer, phone or tablet. In our evaluation, detailed below, we use a smartphone rather than a PC or tablet for two reasons. Firstly, phones are generally more limited in terms of processing power, available ports and UI, so we expect any configuration approach that works on the phone will also easily transfer to a personal computer. Secondly, mobile phones are normally more convenient or practical to carry around than a computer, especially if a user is installing devices in multiple

[3] `http://tinyurl.com/7ztevun`

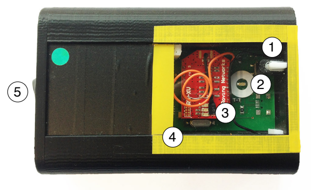

Figure 1: A prototype Wi-Fi Device with the button (1), status light (2), WiFly module (3), window (4), and power button (5) labelled.

Figure 2: Phone placed on Wi-Fi device, covering the window.

locations. To avoid introducing multiple novel factors into the study design, we used input methods (touchscreen keyboard for text entry) commonly found on smart phone interfaces.

We are particularly interested in applications where the budget may be tight, such as energy poverty advice and health care (as described in the introduction), so we further constrained the design space to only use low cost hardware, and therefore ruled out cameras, screens, speakers, or Bluetooth radios, assuming they will only be used for configuration. We also ruled out technology that is not yet prevalent on mobile phones, such as NFC. As a result, we narrowed the options to compare down to three initial approaches. These were chosen because they mirror options used in industry: connecting the phone to the device through a USB cable, transmitting information visually by flashing the screen of the phone, or creating a temporary ad-hoc network used only for the initial configuration. Each is described in detail below.

Prototypes

In order to test the various configuration methods, 3 prototype Wi-Fi devices were developed. These have a number of features in common to ensure that they are comparable in every aspect except for their configuration mechanism, and these are described before focussing on the individual differences. Each device has a power switch, a configuration button, and a status light, as illustrated in Figure 1. They are enclosed in identical 3D printed boxes, which include a plastic window, to allow users to see the status light. When first turned on the devices indicate that they need to be configured by flashing the status light. On pressing the configuration button, the status light stops blinking and remains lit until the configuration is either successful (the status light turns off), or unsuccessful (the status light resumes blinking). While in commercial products the device would boot into configuration mode at first, we took into account the process of *reconfiguration* to more easily allow performance measurements.

Each device contains the same hardware at its core: a custom board with an AVR microcontroller. The board also includes the status LED and button, a light sensor, USB connector, and programming pins. Finally, an RN-XV WiFly module is mounted on the board. This is a low power Wi-Fi module that allows for the board to connect to a Wi-Fi network. At just over £25, this is the most expensive of all the components in the device: the bill of materials being approximately £35.

A custom firmware was developed for each individual prototype, allowing them to be configured using the three different approaches described in the following subsections.

USB

This approach uses a USB cable to transmit the network name and passphrase from the phone to the Wi-Fi device. The device's firmware makes use of an open source USB library[4] to communicate with an Android application running on the phone. A similar technique is used to configure many Wi-Fi printers already on the market, such as those by HP and Brother, using a laptop or desktop computer. With these products, users connect a USB cable from their computer to the printer, and software installed on the computer lets the user set up the device to connect to an existing Wi-Fi network.

With our prototype, users need to physically connect the smartphone to the device and press the button on the device to put it into configuration mode. At this point the phone detects the device's unique identifier and automatically launches the configuration application. This application consists of a simple input form through which users enter the network information (network name and passphrase), which is then transmitted to the device via the USB connection. As soon as the transmission is over, the application automatically closes. The device is programmed to configure its wireless connection accordingly. If incorrect details are given, or it is otherwise unable to set up the connection, the device returns to an unconfigured state so the user can try again.

It should be noted that, while USB is a common standard, it is only supported by some mobile phones, and a special USB "On-The-Go" cable is required. This method would, however, be compatible with the majority of laptop and desktop computers, provided suitable software drivers were installed.

Flashing

Flashing configuration makes use of a light sensor on the Wi-Fi device to detect changes of brightness of the phone screen. The phone encodes the information to be sent to the device into a sequence of brightness levels, and this is then *flashed* to the sensor. This technique is used in some existing commercial products, including those that use the ElectricImp[5] platform, such as the Aros air conditioner[6].

[4] `http://www.obdev.at/vusb/`

[5] `http://electricimp.com`

[6] `https://www.quirky.com/shop/752`

With this approach, users need to enter their network details on the phone, press the configuration button on the device, and then put the phone face-down over the window of the device[7], as shown in Figure 2. After a 5 second countdown from when they enter the information on the phone, the screen flashes to transmit the credentials over to the device. To indicate the completion of the transmission, the phone vibrates and a 'chime' sound is played. At this point the user can see if the configuration process was successful or not. This use of sound is essential, as the display of the phone is covered once it is placed onto the device. There is also a timeout on the device: if it takes longer than 35 seconds, the device returns to its unconfigured mode.

From a technical point of view, our implementation of flashing configuration is platform agnostic. It runs entirely within the web browser using HTML and Javascript: only a screen, keyboard, and speaker are required, whether on a smartphone or on a computer.

A modified Manchester encoding is applied to a binary representation of the network information data: 1 is encoded as a `LOW,HIGH` sequence, and 0 as a `HIGH,LOW` sequence, where `LOW` and `HIGH` refer to the screen brightness levels: black and white. This type of encoding was chosen to allow the clock recovery and synchronization from the received signal, as well as for the equal distribution of `LOW` and `HIGH` symbols, to facilitate calibration. The standard Manchester encoding, however, relies on an accurate clock on the transmitter, which cannot be guaranteed with a Javascript implementation running in a web browser. To address this, we modified the encoding strategy: if two `LOW`s or `HIGH`s are adjacent, we replace the second value with a medium brightness level (grey), or `MID`. For example, given `LOW,HIGH,HIGH,LOW`, the result will be `LOW,HIGH,MID,LOW`: no 2 adjacent values are the same.

The payload consists of 4 sections: a calibration section, so that the device can ascertain the light levels for black, white, and grey; a header; the length of the message; and then the encoded message. Each byte of the message, and the length, has a parity bit appended, so the device can detect any corruption. A signalling interval of 65 milliseconds per bit was found to guarantee an acceptable level of performance for our prototype and phone model. With this setting, the total time required to transmit the network name and passphrase used in the study is 33 seconds, including 7 seconds of calibration.

Web

This approach supports the configuration process entirely over Wi-Fi, with the smartphone connecting directly to the Wi-Fi device, which initially acts as an access point. Some commercial products (e.g. the Nest Smoke Alarm[8], Belkin Wemo devices[9], and FitBit Aria Scales[10]) use this technique, often taking advantage of computer or phone applications.

[7] as the device enclosure is made of plastic there is no risk to damage the phone screen

[8] **https://nest.com/**

[9] **http://www.wemothat.com/**

[10] **http://www.fitbit.com/uk/aria**

When the user puts our device into configuration mode, by pressing its button, the Wi-Fi module enters a setup state: it acts as an access point, it creates an "ad-hoc" network, and it launches a web server running on the embedded device. By connecting to the network and navigating to a specific web page, the network name and passphrase can be specified, as well as more advanced details.

We aimed to use web configuration as a baseline for our experiment, and thus wanted to make it as simple to set up as possible, hiding the complex configuration options exposed by the specific Wi-Fi module we use. So, a native Android application was developed that automatically connects to the wireless network at launch, to minimize interaction with the phone's settings interface, and exposes the same UI for network SSID and passphrase as our USB application, described above. These details are then sent to the webserver from the application. This approach would not work with iPhones or iPads, as it is not possible to change the wireless network connection from an application in iOS.

STUDY 1

A lab usability study was designed and conducted to comparatively evaluate the IoT configuration methods described above. This was performed in the lab to easily allow precise performance measurements, which would be difficult or impossible in a field study. Moreover, the comparison of different configuration strategies would in any case conflict with a naturalistic observation in the field.

Participants

Fifteen volunteers were recruited via mailing lists. A total of 12 females and 3 males, aged between 24 and 65 (mean: 40, std. dev. 13). All subjects expressed interest in participating via email, showing familiarity with computers, and received a £10 shopping voucher for their time. Anyone who expressed interest and was above 18 years of age was included in the study, so long as they did not have a technical background (i.e. no degree in computing or engineering and not in technical employment) and did not suffer from epilepsy (due to the flashing lights used in the flashing configuration).

All of the participants in the study stated that they used a computer for work on a daily basis. 10 (67%) use a phone for work on a daily basis; 4 (27%) occasionally use a mobile phone; and 1 (6%) rarely uses a mobile phone. All but one participant had connected their computer or phone to a wireless network; 9 (60%) had set up a wireless network; and 9 (60%) had set up a device, that was not a computer, to connect to a wireless network.

Method

The user trial consisted of three tasks to configure a Wi-Fi device to connect to an existing wireless network: one using the flashing approach; one using the USB approach; and one using the web approach. The ordering of these tasks was permuted to reduce carry-over effects: 5 participants started with flashing, 5 with USB, 5 with web, and the remaining 2 conditions were alternated.

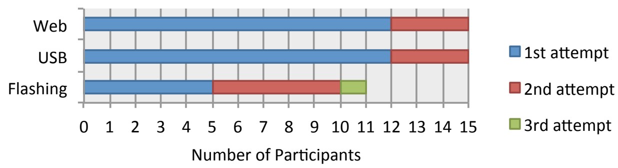

Figure 3: Overview of participants' performance in Study 1, showing the number of participants who successfully completed the task within 3 attempts.

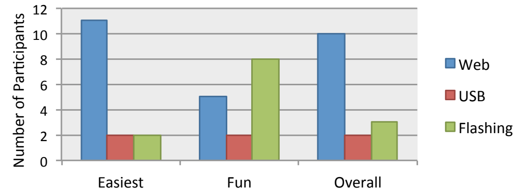

Figure 4: Participant preferences for configuration approaches in Study 1.

At the beginning of the experiment, participants received a general instruction sheet. This described the task and the device, and it listed the network name and password to be used to configure the device. The sheet also stated that participants should read the instructions for each condition fully before starting. Prior to each task, the participants were given task-specific instructions, the device, a Samsung Galaxy SIII smartphone, and, in the case of the USB approach, a USB cable. They were then asked to follow the instructions to configure the device, and to notify the researcher when they felt they had finished.

Following the completion of the tasks, participants were asked to complete a questionnaire on a laptop. This included questions about demographics and general background, as well as subjective evaluation of the approaches used and prior knowledge and experience with wireless devices and wireless device configuration. In particular, subjects were asked which approaches they found easiest to use and why; which approach they found most enjoyable and why; and which approach they preferred overall and why. Note that this is based on subjective perception of one's own performance, which may or may not correspond to factual performance.

The specific background questions asked whether users were familiar with computers and mobile phones; whether they had connected their computer or phone to a new wireless network; whether they had ever set up a wireless network; and whether they had set up a device, other than a computer, to connect to a wireless network. If they answered the last question positively, they were asked for more specifics as to the device they had configured.

The experiment was filmed using two video cameras: one giving a wide angle view of their interactions with the devices and one facing the participant. The researchers also took notes during the task. The entire experiment took approximately 30 minutes, including the time to read and sign a consent form, read the instructions, complete the tasks and fill out the questionnaire.

Results

The *USB* and *web* configuration strategies were the most successful, with every participant managing to successfully complete the task, i.e. configuring the device. In both cases, 3 participants were unsuccessful on their first attempt, but configured the device on their second attempt. *Flashing* configuration led to the most failures: 4 of the 15 participants failed to configure the device entirely; 5 managed on their first attempt; 5 on their second attempt; and 1 on their third attempt. The results are summarized in Figure 3.

Task completion times, measured from the pressing of the configuration button to the completion of successful (or unsuccessful) configuration, were also recorded and analyzed. It took participants 102 seconds on average (SD: 52.6) to complete each task, ranging from 40 seconds to 245 seconds. A two-way ANOVA on configuration method and task order revealed no statistical significance across methods, but a significant effect of order ($P<0.05$), with no effect of interaction between the two. A post-hoc Tukey test revealed that successfully completing the 1st task took significantly longer (average of 149 seconds, SD: 53.5) than the 2nd (average of 69 seconds, SD: 26.7) and 3rd (average of 73 seconds, SD: 15.9). No correlation was found between the participants' experience level and their performance in the study.

Results from the subjective ratings are reported in Figure 4. Web configuration was rated as the easiest and overall preferable option, while flashing was rated as most enjoyable.

Error Analysis

To understand the issues encountered by our participants during the study, we performed a video analysis of the recorded footage. Two researchers individually coded the footage for errors that occurred during configuration, and then harmonized them through consensus, deriving 8 initial codes. The codes were then grouped into 5 categories, described below. The distribution of error categories for each of the 3 configuration strategies is reported in Figure 5.

Timing Issue

Some participants encountered timing issues when configuring the device using the flashing approach. These typically occurred when the participant was too slow in placing the phone onto the device, and made it unlikely that it would calibrate correctly.

Skipped Step

This describes a case where participants missed a step in the configuration process. For example, some participants did not press the configuration button on the device before undertaking the configuration process: as such, it could not be configured. Similarly, one participant did not press the configuration button on the phone interface during flashing configuration, so the device returned to an unconfigured state.

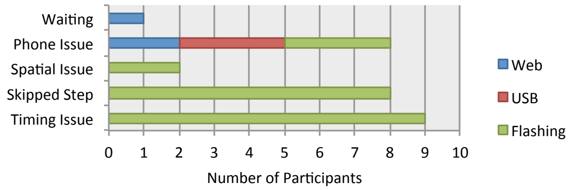

Figure 5: Categories of errors participants made in Study 1.

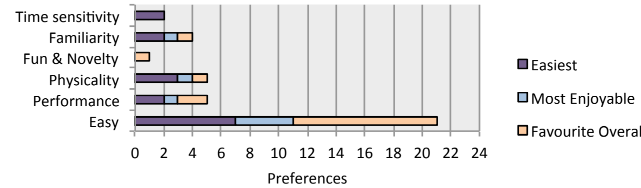

Figure 6: Reasons expressed by participants for preferring web configuration in Study 1.

Spatial Issue

The instructions for flashing configuration required that participants positioned the phone as shown in Figure 2 during configuration. Some participants placed the phone incorrectly on the device, resulting in insufficient light reaching the light sensor. In addition, one participant could not find the window on the device, on which the phone was supposed to be placed.

Phone Issue

This code encompasses errors caused by smartphone usage. Some participants accidentally touched the screen and launched an application while turning the phone upside-down to place it onto the device during flashing configuration; some entered the network name or passphrase incorrectly.

Waiting

On one occasion it was not clear to the participant that they should advance to the next step of the instructions. The participant launched the configuration application (the first step in the instructions) but did not then press the device configuration button (the second step in the instructions), as the phone screen stated that it was "Searching for wireless node". The participant assumed that they should wait for this to complete, and thus the researcher intervened after a few minutes.

Questionnaire Analysis

The responses from the questionnaires were coded by two researchers, and then grouped into 8 broad categories: easy, time-sensitive, performance, physicality, feedback, fun & novelty, familiar, miscellaneous. Figure 6 illustrates the reasons expressed by participants for preferring web configuration (the favourite approach).

Examples in the *easy* category included "Seemed like the most straightforward!". The *time-sensitive* category included "[...] didn't feel like there was a time element involved". Examples in the *performance* category included "No problems with the process" and "because it worked". Comments categorized as *physicality* included "it is better not to have wires" and "I think a physical connection between devices has a psychological connection for the lay user!". The *feedback* category included "It's easier to see it's doing something". Comments in the *fun & novelty* category included "It was the newest method to me so interesting to try out" and "Good fun with sounds and felt a bit different!!" Finally, the *familiar* category included "Most like my phone" and "Most consistent with other methods I'm used to using."

Discussion

Web and USB configuration performed equally well. All participants managed to configure the Wi-Fi device, and both approaches had 3 errors – considerably fewer than the 22 encountered with flashing. This result suggests that both strategies could be employed successfully to configure IoT devices. Given that we originally expected that all of the configuration approaches would perform similarly due to their adoption in industry, it is surprising that the number of errors were so different. It is worth remembering, though, that this implementation of web configuration was made as simple as possible, at the expense of generality, to act as a baseline. It requires a specific phone OS and a native application in order to manipulate the phone's Wi-Fi settings. Most existing approaches require that the user joins an ad hoc network manually before launching the configuration application.

The video analysis revealed a variety of issues that our participants encountered during the experiment, which prevented them from successfully completing the configuration task. Some of these seem to be specifically related to the configuration strategies being tested, while others seem to be more generic. In particular, "phone issues" appear to stem from the participants' inexperience with the phone being used in the experiment; the "waiting" seems instead to be related to flaws in our task instructions, possibly simple to amend through a simple edit. Similarly, "skipped steps" seem to be due to our participants being overwhelmed by the number of steps in the instructions. In contrast, the "spatial" and "timing" issues appear to be more directly related to the specific configuration procedure that was being tested, namely flashing.

It is worth pointing out, then, that all the failures which occurred with the USB and web strategies appear to be related to general issues (phone issues and waiting) rather than being specific to them. Instead, the flashing configuration was the only one affected by issues specific to that procedure. Of the 22 errors identified during flashing configuration, 9 were timing issues, 2 spatial issues, 8 skipped steps and 3 phone issues. On one hand, this analysis confirms the flashing configuration to be the least preferable. On the other, it identifies specific amendments that could be applied to the device, procedure and instructions to improve their performance. In terms of subjective preferences, our participants seem to have valued the simplicity of the web option, but also appreciate the novelty and perhaps even the challenge involved with flashing. While the USB and web approaches had exactly the same measured performance, it was perhaps the simplicity of the web approach that led to it being the preferred option in terms of ease of use, and also overall.

DESIGN ITERATION

The first study revealed opportunities to iterate the design of the configuration strategies, including both the instructions and the procedure itself. In particular, we decided to update the flashing approach to try to address the timing and spatial issues.

Originally we considered the web option as just a baseline comparison, and so provided an implementation that was as simple to use as possible, but very platform specific. Results from the study suggested instead that web may provide a concrete alternative, and so we decided to test it under more realistic conditions: let the user alter the phone settings to connect to the wireless network generated by the Wi-Fi device.

Finally, USB and web were equally successful in the study, with the key differences to web configuration being the use of a cable and familiarity, and some of our participants mentioned this in the questionnaire. As such, we wanted to compare how the new web approach would compare to a cable-based solution. As USB is platform dependent, and access from phone applications is often restricted (e.g. on iOS), we implemented an audio configuration approach that makes use of the headphone socket on the phone to transmit the configuration details.

Web v2

In order to make web configuration more realistic, the approach was altered such that users had to connect to the Wi-Fi device's access point by updating the phone settings, and then launch the application to configure the device. Step by step printed instructions were provided to guide users through the process, targeting the specific phone model we used.

Again a native phone app was used to simplify the Wi-Fi module's configuration dialog. However, the approach does not rely on platform-specific features, so it would be straightforward to develop apps for other platforms. In principle, it would even be possible to modify the firmware on the Wi-Fi module itself to make its configuration UI more usable.

Flashing v2

The flashing configuration procedure was modified with the aim of reducing the issues that affected its earlier version. In particular, to more closely guide participants step-by-step through the instructions (and hence reduce step skipping), we decided to display them on the smartphone screen, one step at a time (following the "wizard" UI pattern). Also, having instructions on screen allowed us to include some animated illustrations to direct users as to where the Wi-Fi device should be placed, in the hope of reducing spatial issues.

However, having the instructions just on the phone screen conflicted with the phone being face-down on some of the steps. To solve this issue, the approach was changed such that the user would place the Wi-Fi device onto the phone screen, leaving a small portion of the display uncovered and still visible with the device in place. In order to facilitate the precise positioning of the device on the phone, a half-circle was added to provide a means for orientation. The user is asked to complete a full circle on the phone screen using this half-circle, thus ensuring that the device is oriented correctly (see Figure 7).

Finally, to try and reduce timing issues, the configuration button on the Wi-Fi device was moved to the end of the casing, to allow for the button to be pushed after the device is placed on the phone.

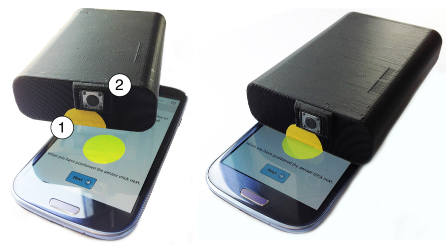

Figure 7: The Wi-Fi device placed on the phone's screen, oriented to complete a circle (1). The relocated configuration button can also be seen (2).

Audio

The USB approach from the first study was successful (only 3 failed attempts, all due to typing issues) but, as mentioned earlier, it suffers from platform dependency: not all phone operating systems permit applications to access USB devices (even if the port is physically available). As such, we developed and tested audio configuration in its place.

Audio configuration builds upon the USB approach, though in this instance the phone is connected via an audio cable that plugs into the phone's headphone socket and into a similar socket on the Wi-Fi device. The Mimo Baby Monitor makes use of this technique to configure its Wi-Fi base station[11]. While a microphone in the device was considered as an option, this would require extra complexity to function in noisy environments, and would potentially place more demand on the device's processor and battery.

For efficiency, our technical implementation is based on the flashing configuration approach, with amplitude readings from the audio cable corresponding directly to intensity readings from the light sensor. The light sensor was replaced by a basic amplitude demodulator (capacitor, diode and resistor). In contrast to USB configuration, no extra drivers or access rights are required on the phone, as we take advantage of its existing audio playback functionality. The experience on the phone also differs: it is not possible to launch a native application when the audio cable is connected, unlike with USB, and so the device is configured via a web page that is launched by the user. This is similar to flashing configuration, with the user pressing a button in the application, and the transmission taking some time to complete.

The data is transmitted in exactly the same way, but 0, 1, and 2 correspond to the amplitude of a 5KHz wave. As the sensor input is 0 when no sound is playing, a countdown is not required: the device instead uses a signal threshold to determine when transmission starts. The phone screen is visible

[11] `http://mimobaby.com/program`

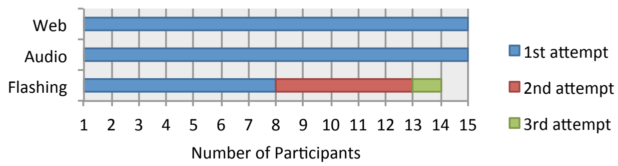

Figure 8: Overview of participants' performance in Study 2, showing the number of participants who successfully completed the task within 3 attempts.

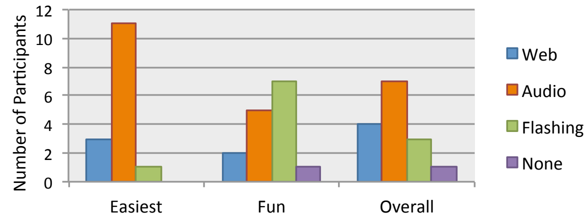

Figure 9: Participant preferences for configuration approaches in Study 2.

with this approach, so a progress indicator is displayed during the transmission. As with the new version of flashing configuration, the majority of the instructions were moved onto the phone screen.

The network name and passphrase are encoded into audio using a simple web application implemented in PHP. As the majority of mobile phones can play audio files from the web, without needing any extra applications to be installed, this configuration approach has much better support than the USB approach described earlier.

STUDY 2

A second study was designed and carried out to compare the 3 new configuration approaches, closely following the design of the previous study.

Participants

15 volunteers were recruited, using the same approach as the first study. A total of 10 females and 5 males, aged between 23 and 54 (mean: 37, std. dev. 8.5) took part.

13 (87%) of the participants in the study stated that they used a computer for work on a daily basis, with the remaining 2 (13%) stating that they developed software using a computer (even though their formal background is not technical and their job title did not suggest it). 14 (93%) of the participants use a phone for work on a daily basis, and 1 (7%) uses a phone occasionally. 13 (87%) had connected their computer or phone to a wireless network; 6 (40%) had set up a wireless network; and 5 (33%) had set up a device, that was not a computer, to connect to a wireless network.

Method

The user trial consisted of three tasks to configure a Wi-Fi device to connect to an existing wireless network, using the three new approaches. As in Study 1, the ordering of these conditions was permuted to carry-over effects. The study was filmed and task completion time recorded. A questionnaire similar to the one from Study 1 was also administered.

Results

The *audio* and *web* configuration strategies were the most successful, with every participant configuring the Wi-Fi device on their first attempt. *Flashing* configuration led to the most failures: 1 of the 15 participants failed to configure the device entirely; 8 managed on their first attempt; 5 on their second attempt; and 1 on their third attempt. The results are summarized in Figure 8.

Task completion times were also recorded and analyzed. It took participants 110 seconds on average (SD: 38.8) to complete each task, ranging from 60 seconds to 226 seconds. A two-way ANOVA on configuration method and task order revealed no statistical significance across methods, but a significant effect of order ($P<0.05$), with no effect of interaction between the two. A post-hoc Tukey test revealed that successfully completing the 1st task took significantly longer (average of 124 seconds, SD: 48.8) than the 2nd (average of 111 seconds, SD: 29.8) and 3rd (average of 98 seconds, SD: 34.0). Again, no correlation was found between the participants' experience level and their performance in the study.

Results from the subjective ratings are reported in Figure 9. Audio configuration was rated as the easiest and overall preferable option, while flashing was rated as most enjoyable.

Error Analysis

As with the first study, two researchers analysed the footage recorded during the tasks, deriving initial codes for errors that occurred during configuration. The codes were then grouped in 5 categories. Of these, 3 were in common with those of Study 1: *spatial*, *phone issue*, and *skipped step*. Where in the old study users would skip steps in the printed instructions, in the new study they would advance to the next step in the onscreen instructions too soon. The 2 new categories are described in the following subsections. The distribution of error categories for each of the 3 configuration strategies is reported in Figure 10.

Network Details

One participant did not notice the network name and passphrase on the router, and assumed that they were not necessary. As such, they left the fields blank when configuring the device.

Turned Device Off

In one instance, the participant mistook the power button on the Wi-Fi device for the configuration button, and turned it off. Notably, the participant assumed the device had been configured correctly, as the status light had turned off.

Questionnaire Analysis

The responses from the questionnaires were coded by two researchers, and these were split into 9 broad categories: easy, less steps, performance, physicality, feedback, fun & novelty, familiar, instructions, miscellaneous. The additional category in this study was *instructions*, which included comments such as "I found this to have the simplest instructions" and "Having all the information on the phone screen made things much more convenient when following the steps." Figure 11 illustrates the reasons expressed by participants for preferring audio configuration (the favourite approach).

Discussion

Both audio and the web approach performed well, with all users configuring the device on their first attempt. Audio does seem a viable substitute for USB: it is completely platform independent – it can run in a browser, without requiring any drivers. It should also be noted that the audio option requires fewer electronic components than the USB one, making it simpler and cheaper to build.

It should also be underlined that the implementation of the audio configuration method that we evaluated was a rapidly developed prototype, so there is plenty of room for improvement. In particular, two issues have already been identified. Firstly, the waveform currently used is quite unpleasant to hear if the audio cable is unplugged by mistake during the transmission (this happened during our technical development, but never in the study). So, this waveform could be replaced by one that is more pleasant to listen to. Secondly, it should be easily possible to increase the transmission rate by fine tuning the analog demodulator.

The two new error categories identified by the video analysis of Study 2 are both general, i.e. not specific to any tested configuration. We believe that both of them indicate issues that can be easily solved. The device's power button should be labelled more clearly to avoid users confusing it with the configuration button. The web application requesting users to enter the network name and password can easily be amended to make at least the network name required to progress with the interaction.

The updated version of flashing configuration performed considerably better than the version evaluated in the first study, but still less well than the alternatives. With flashing, one participant still could not successfully complete the configuration task within the 3 attempts. Five other participants required 2 attempts and one 3 attempts, giving a total of 10 failed attempts. Even excluding the 4 failed attempts due to issues that can be considered general or that can be easily avoided (2 "network details", 1 "device being turned off", and 1 phone issue), there were still 6 failed attempts due to issues specific to this configuration, despite the design iteration. In terms of subjective preferences, most participants in Study 2 found the flashing option to be the most enjoyable. The qualitative analysis of the answers suggests that this is mostly due to a novelty factor and because it involved physically putting the two devices in contact.

Our comparative evaluation of the configuration methods raises some questions about the suitability of flashing for device configuration. It is perhaps surprising that this method performed worse than the other methods, and yet is adopted in industry. Even though the implementation that we tested cannot be guaranteed to be the same as ones commercially available (as they are proprietary), online public product reviews also yield negative comments about the malfunctioning of the flashing configuration process[12].

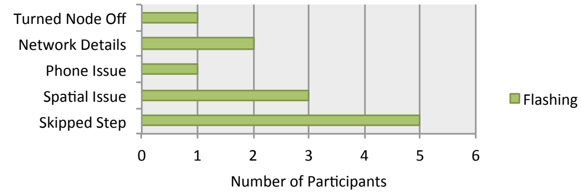

Figure 10: Categories of errors participants made in Study 2.

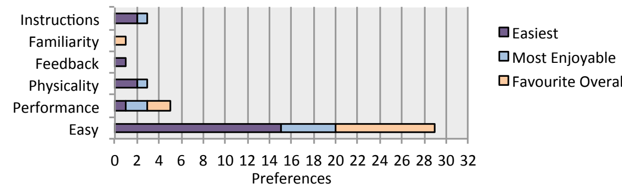

Figure 11: Reasons expressed by participants for preferring audio configuration in Study 2.

CONCLUSIONS

In this paper, we have presented two usability studies, with a total of 30 participants, none of whom had formal technical training. In each study, the participants configured Wi-Fi devices with minimal UIs using a variety of techniques, inspired by methods already adopted by products on the market. Through a combination of failure analysis and subjective questionnaires, our results indicate that two configuration approaches are more usable and would be best suited to set up Wi-Fi devices. These are web configuration, where a user connects to an access point on the Wi-Fi device, and audio configuration, where the network details are sent over an audio cable that connects the smartphone to the device.

Each of them presents inherent benefits and limitations in terms of technical requirements. Web configuration requires little by way of extra hardware, but needs more platform-specific instructions and software integration. Conversely, audio configuration is more platform-agnostic, but requires a cable and a few extra electronic components.

Configuration approaches need to be intuitive and perform well if end users are to be able to incorporate, and rely on, these devices in their daily lives. Therefore, we hope that the results presented in this paper will stimulate discussion and further work related to actual deployments of Ubicomp systems that take advantage of the prevalent Wi-Fi infrastructure.

ACKNOWLEDGEMENTS

Work partially supported by EPSRC CharIoT and CTech projects (ref: EP/L02392X/1, EP/K002589/1). Data URL: http://doi.org/557. Study approved by U.Southampton FPSE Ethics Committee (ref: 9186).

[12] `http://tinyurl.com/mae4ded`

REFERENCES

1. Balfanz, D., Durfee, G., Grinter, R. E., Smetters, D. K., and Stewart, P. Network-in-a-box: How to set up a secure wireless network in under a minute. In *USENIX Security Symposium* (2004).
2. Balfanz, D., Smetters, D. K., Stewart, P., and Wong, H. C. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proc. NDSS'02* (2002).
3. Brown, A., Mortier, R., and Rodden, T. MultiNet: Reducing interaction overhead in domestic wireless networks. In *Proc. CHI'13* (2013), 1569–1578.
4. Chong, M. K., and Gellersen, H. How users associate wireless devices. In *Proc. CHI'11* (2011), 1909–1918.
5. Chong, M. K., Mayrhofer, R., and Gellersen, H. A survey of user interaction for spontaneous device association. *ACM Comput. Surv. 47*, 1 (May 2014), 8:1–8:40.
6. Costanza, E., Panchard, J., Zufferey, G., Nembrini, J., Freudiger, J., Huang, J., and Hubaux, J.-P. SensorTune: a mobile auditory interface for DIY wireless sensor networks. In *Proc. CHI'10* (2010).
7. de Oliveira, R., Cherubini, M., and Oliver, N. Movipill: Improving medication compliance for elders using a mobile persuasive social game. In *Proc. UbiComp'10* (2010).
8. Derthick, K., Scott, J., Villar, N., and Winkler, C. Exploring smartphone-based web user interfaces for appliances. In *Proc. MobileHCI'13*, ACM (New York, NY, USA, 2013), 227–236.
9. Fischer, J. E., Costanza, E., Ramchurn, S., Colley, J., and Rodden, T. Energy advisors at work: charity work practices to support people in fuel poverty. In *Proc. UbiComp'14* (2014).
10. Gaver, W., Bowers, J., Kerridge, T., Boucher, A., and Jarvis, N. Anatomy of a failure: how we knew when our design went wrong, and what we learned from it. In *Proc. CHI'09* (2009).
11. Goodrich, M., Sirivianos, M., Solis, J., Tsudik, G., and Uzun, E. Loud and clear: Human-verifiable authentication based on audio. In *Proc. ICDCS'06* (2006), 10.
12. Grinter, R. E., Edwards, W. K., Newman, M. W., and Ducheneaut, N. The work to make a home network work. In *Proc. ECSCW'05* (2005), 469–488.
13. Hesselmann, T., Henze, N., and Boll, S. Flashlight: Optical communication between mobile phones and interactive tabletops. In *Proc. ITS'10*, ACM (New York, NY, USA, 2010), 135–138.
14. Holmquist, L. E., Mattern, F., Schiele, B., Alahuhta, P., Beigl, M., and Gellersen, H.-W. Smart-its friends: A technique for users to easily establish connections between smart artefacts. In *Proc. UbiComp'01* (2001), 116–122.
15. Kalnikaite, V., Rogers, Y., Bird, J., Villar, N., Bachour, K., Payne, S., Todd, P. M., Schöning, J., Krüger, A., and Kreitmayer, S. How to nudge in situ: designing lambent devices to deliver salient information in supermarkets. In *Proc. UbiComp'11* (2011).
16. Kumar, A., Saxena, N., Tsudik, G., and Uzun, E. Caveat eptor: A comparative study of secure device pairing methods. In *Proc. PERCOM'09* (2009), 1–10.
17. Kuo, C., Walker, J., and Perrig, A. Low-cost manufacturing, usability, and security: An analysis of bluetooth simple pairing and wi-fi protected setup. In *Proc. FC'07/USEC'07*, Springer-Verlag (Berlin, Heidelberg, 2007), 325–340.
18. Mayrhofer, R., and Gellersen, H. Shake well before use: Authentication based on accelerometer data. In *Proc. PERVASIVE'07* (2007), 144–161.
19. McCune, J. M., Perrig, A., and Reiter, M. K. Seeing-is-believing; using camera phones for human-verifiable authentication. *Int. J. Secur. Netw. 4*, 1/2 (Feb. 2009), 43–56.
20. Mennicken, S., and Huang, E. Hacking the natural habitat: An in-the-wild study of smart homes, their development, and the people who live in them. In *Pervasive Computing*, J. Kay, P. Lukowicz, H. Tokuda, P. Olivier, and A. Krger, Eds., vol. 7319 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, 143–160.
21. Merrill, D. Siftables: towards sensor network user interfaces. In *Proc. TEI'07*, ACM Press (2007), 75–78.
22. Poole, E. S., Chetty, M., Grinter, R. E., and Edwards, W. K. More than meets the eye: Transforming the user experience of home network management. In *Proc. DIS'08*, ACM (New York, NY, USA, 2008), 455–464.
23. Saxena, N., Ekberg, J.-E., Kostiainen, K., and Asokan, N. Secure device pairing based on a visual channel: Design and usability study. *IEEE Transactions on Information Forensics and Security 6*, 1 (2011), 28–38.
24. Scholl, P. M., Kckyildiz, N., and Van Laerhoven, K. Bridging the last gap: LedTX - optical data transmission of sensor data for web-services. In *Proc. UbiComp'13*, ACM (New York, NY, USA, 2013), 47–50.
25. Soriente, C., Tsudik, G., and Uzun, E. Beda: Button-enabled device pairing. *Proc. IWSSI'07 2007* (2007), 246.
26. Soriente, C., Tsudik, G., and Uzun, E. HAPADEP: Human-assisted pure audio device pairing. In *Proc. ISC'08*, Springer-Verlag (Berlin, Heidelberg, 2008), 385–400.
27. Stajano, F., and Anderson, R. J. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Proceedings of the 7th International Workshop on Security Protocols*, Springer-Verlag (London, UK, 2000), 172–194.

28. Suomalainen, J. Smartphone assisted security pairings for the internet of things. In *Wireless Communications, Vehicular Technology, Information Theory and Aerospace Electronic Systems (VITAE)* (May 2014), 1–5.

29. Swindells, C., Inkpen, K. M., Dill, J. C., and Tory, M. That one there! pointing to establish device identity. In *Proc. UIST'02* (2002), 151–160.

30. Tolmie, P., Crabtree, A., Rodden, T., Greenhalgh, C., and Benford, S. Making the home network at home: Digital housekeeping. In *Proc. ECSCW'07* (2007).

31. Ur, B., Jung, J., and Schechter, S. The current state of access control for smart devices in homes. In *Workshop on Home Usable Privacy and Security (HUPS)* (July 2013).

32. Wi-fi Alliance. Wi-fi CERTIFIED wi-fi protected setup: Easing the user experience for home and small office wi-fi networks. Tech. rep., Wi-fi Alliance, 2010.

33. Yang, J., and Edwards, W. K. Icebox: Toward easy-to-use home networking. In *Proc. INTERACT'07*, Springer-Verlag (Berlin, Heidelberg, 2007), 197–210.

34. Yang, R., Newman, M. W., and Forlizzi, J. Making sustainability sustainable: Challenges in the design of eco-interaction technologies. In *Proc. CHI'14* (2014).