Wanda: securely introducing mobile devices

Timothy J. Pierson, Xiaohui Liang, Ronald Peterson, David Kotz
Department of Computer Science, Dartmouth College

***Abstract*—Nearly every setting is increasingly populated with wireless and mobile devices – whether appliances in a home, medical devices in a health clinic, sensors in an industrial setting, or devices in an office or school. There are three fundamental operations when bringing a new device into any of these settings: (1) to configure the device to join the wireless local-area network, (2) to partner the device with other nearby devices so they can work together, and (3) to configure the device so it connects to the relevant individual or organizational account in the cloud. The challenge is to accomplish all three goals *simply*, securely, and consistent with user intent. We present a novel approach we call Wanda – a 'magic wand' that accomplishes all three of the above goals – and evaluate a prototype implementation.**

I. Introduction

Lately we have seen predictions of how the Internet of Things (IoT) is poised to make billions of everyday objects "smart" by adding wireless communication capabilities. The dream is that networks of these newly connection-enabled devices will give us greater insight into the behavior of complex systems than previously possible. The reality, however, is that configuring and managing billions of devices will be extremely difficult.

As an illustration in the healthcare domain, imagine that a general-practice physician tells a patient that he'd like the patient to take home a wireless blood-pressure monitor and use it every day so that the physician can remotely monitor the patient's health. The intention is that the blood-pressure measurements taken by the patient will end up stored in the patient's Electronic Health Record (EHR) at the physician's clinic. The physician can then see the patient's blood pressure on a daily basis and get automated alarms if any abnormal readings are recorded.

At least three problems arise in making scenarios such as at-home blood-pressure monitoring a reality. The first problem is that blood-pressure monitors, like many IoT sensors, do not normally come with long-range communication connections; they have short-range radios such as Wi-Fi, Bluetooth, or Zigbee. The blood-pressure monitor must somehow get connected with other devices in the home such as a Wi-Fi access point (AP) in order to transmit its medical data to the physician's EHR system. Making those connections is difficult for many people, especially considering that different types of devices from different manufacturers often have different methods of making a connection and that the devices themselves often have very limited user interfaces.

A second problem with this blood-pressure scenario is that once a connection is made between the blood-pressure monitor and a device capable of transmitting data long distances, the blood-pressure readings must get to the right patient record in the right physician's EHR system. This implies that the blood-pressure readings must be augmented with additional credentials (e.g., patient ID, password) and destination information (e.g., a Restful API URL).

A third problem arises when devices partner with other nearby devices so they can work together in a peer-to-peer fashion, such as a blood-glucose monitor working with an insulin pump. In these peer-to-peer cases the devices may maintain a connection with a long-range communication device, but may also need a connection with neighboring devices using encryption based on a unique key for a specific pair of devices, rather than a common key shared by all devices. Establishing the encryption can be difficult if the devices have never met before and have never shared a secret key.

To overcome these three and other difficulties inherent in configuring wireless devices, we present a system called **Wanda**. Wanda introduces a small hardware device called the 'Wand' that has two antennas separated by one-half wavelength and uses radio strength as a communication channel to simply, securely, and consistent with user intent, impart information onto devices. In this paper we focus on connecting devices, but the Wand could be used to impart *any* type of information onto a nearby device. Wanda is more than just a solution for pairing devices or connecting to access points.

Wanda builds on pioneering work done by Cai et al. in Good Neighbor [1] in that the Wand determines when it is in close proximity to another transmitting device by measuring the difference in received signal strength on the Wand's two antennas. Wanda then expands upon Good Neighbor by exploiting wireless signal reciprocity to securely impart information in-band from the Wand onto the nearby target device.

Unlike many other approaches, Wanda does not require any specialized hardware (or any hardware changes) in the new devices, does not require any pre-shared secrets, and does not require complex algorithms or complicated cryptography libraries. Furthermore, Wanda does not require the devices to be adjacent, or even movable – useful for large appliances as well as small mobile devices.

Using Wanda could hardly be easier: a person simply points the Wand at a nearby device that requires connectivity and the Wand almost magically imparts connectivity parameters onto the target device. This happens one time and afterward the Wand is not involved in future communications – the Wand itself disappears from the picture.

A. Assumptions

Throughout this paper we make the following assumptions about the "target device", which is the device receiving information from the Wand: (1) it has at least one radio antenna that it can use to transmit and receive wireless data, (2) it can measure the signal strength of wireless communication packets, (3) it may be limited computationally, but can run a small piece of software that implements the Wanda protocol, (4) it cannot be relied upon to have additional sensors such as cameras, microphones or accelerometers, and (5) it cannot be altered to add new hardware.

We make the following assumptions about the Wand: (1) it can be trusted to generate a secret key, (2) it has a radio compatible with that of the target devices, and two antennas located approximately one half wavelength apart, (3) it is easily portable and can be brought next to and pointed at the target device, and (4) it can run the Wanda protocol.

B. Contributions

Wanda is a novel approach for imparting information onto a target device, even though the target device has never been seen before, nor have any secrets been pre-shared. We make four **contributions** in this paper:

1) a consistent, fast, easy, and secure method to impart any kind of information onto commodity wireless devices, regardless of device type or manufacturer, without hardware modifications to the device;
2) protocols for imparting information onto new devices (such as a Wi-Fi SSID and password), introducing two devices so they can establish a secure and user-intended connection, and imparting cloud identity and credentials into a new device;
3) a prototype implementation and experimental evaluation; and
4) a security analysis of the system.

II. Radio signal strength primer

Wanda uses radio signal strength to impart information onto devices; in this section we briefly review some basic theoretical concepts that are key to Wanda's operation. The material in this section provide the theoretical foundations for why Wanda *should* work, while Section VI shows that Wanda *does* work.

The strength of a wireless signal can be predicted using the well known log-normal shadow model [2]:

$$P_r = P_0 - 10\alpha \log_{10}\left(\frac{d}{d_0}\right) + \chi_\sigma \tag{1}$$

where P_r is the power in dBm received at a distance d from a transmitter, P_0 is the power in dBm received at a distance d_0 from a transmitter, α is the path-loss exponent representing the reduction in power as the signal travels, and χ_σ represents noise. In free space α is 2, but it in real-world environments α can be much higher.

The χ_σ representing noise in Equation (1) can change rapidly, making actual measurements of received signal strength highly variable. Real world obstacles, moving and fixed, can attenuate a signal or cause reflections that create multiple paths between a transmitter and a receiver. The result is that multiple copies of the transmitted signal, each with a different attenuation, delay, and phase shift, arrive at the receiver superimposed upon each other. This superposition can result in either constructive interference where multiple copies of the signal add to each other, or destructive interference where multiple copies of the signal cancel each other. The changes in signal strength caused by these environmental factors is often called *fading*. More details can be found in our technical report [3].

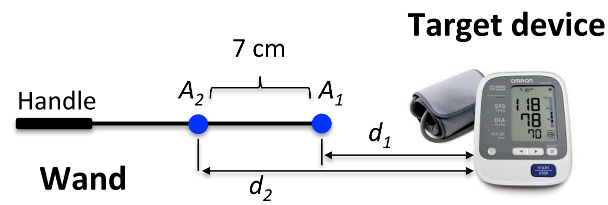

Fig. 1. Wand with two antennas, A_1 and A_2, separated by 7 cm in our prototype. The distance between antenna A_1 and the target device is d_1. The distance between antenna A_2 and the target device is d_2. The Wand is intended to be pointed directly at the target device, so that $d_2 = d_1 + 7$ cm.

Additionally, the signal strength captured by real equipment is also subject to manufacturing variability as well as thermal noise in the antenna [4]. Wanda exploits the variability from manufacturing and thermal noise, together with variability in the environment, to make it difficult for an adversary to eavesdrop on communications between Wanda devices (see Section VII).

III. Approach

Wanda builds on two insights that can be gleaned from the concepts highlighted in Section II. The first insight is that if a device has two antennas, it can determine when it is in close proximity to another device that is transmitting radio signals. The second insight, our major technical contribution, is that when a device with two antennas determines it is in close proximity to another device, it can use its two antennas to securely impart information onto the other device.

In Wanda, the Wand is the device with two antennas (see Figure 1) and it uses those antennas to implement two primitive operations: *detect* and *impart*. This section explains these primitives in detail.

A. Detect primitive

Wanda uses a two-antenna Wand to determine if it is in close proximity to another device that is transmitting a radio signal. Each antenna in the Wand is capable of independently measuring the power received and providing a Received Signal Strength Indicator (RSSI). Building on Equation (1), the power received on the two antennas of the Wand will be:

$$\begin{aligned} P_1 &= P_0 - 10\alpha \log_{10}(\frac{d_1}{d_0}) + \chi_\sigma \\ P_2 &= P_0 - 10\alpha \log_{10}(\frac{d_2}{d_0}) + \chi_\sigma \end{aligned} \tag{2}$$

where P_0 is the power in dBm measured at a distance of d_0 from the transmitter, P_i is the power in dBm measured at receiving antenna A_i, and d_i is the distance between the transmitter and receiving antenna i.

Armed with the equations in (2), we can now calculate the difference in signal strength between the two antennas A_1 and A_2 as follows:

$$\begin{aligned} P_1 - P_2 &= P_0 - 10\alpha \log_{10}(\frac{d_1}{d_0}) + \chi_\sigma \\ &- (P_0 - 10\alpha \log_{10}(\frac{d_2}{d_0}) + \chi_\sigma) \\ &= -10\alpha \log_{10}(\frac{d_1}{d_2}) \end{aligned} \quad (3)$$

The antennas on the Wand are physically close together; in our prototype they are 7 cm apart (roughly 1/2 wavelength). Because they are close together, the environmental factors represented by χ_σ in Equation (3) are likely to be similar on each antenna. By taking the difference in signal strength observed on two antennas, sometimes called the RSSI Ratio [5], the environmental factors tend to cancel out.

When the Wand and the target device are far apart, the distance between antennas A_1 and A_2 is small relative to the distance to the far transmitter. In that case the RSSI will be approximately, although not precisely, equal on each receiving antenna [1]. For example, suppose antennas A_1 and A_2 on the Wand are 7 cm apart and are aligned with the transmitting antenna so that A_2 is 7 cm farther away from the transmitting antenna than A_1 (see Figure 1). In this case $d_2 = d_1 + 7$ cm. Further suppose the distance between A_1 and the transmitting antenna, d_1 is 30 cm (i.e., more than 4 times the distance between the two antennas). In that case, using Equation (3) and assuming $\alpha = 2$ yields a difference, Δ, of:

$$\begin{aligned} d_1 &= 30 \text{ cm} \\ d_2 &= 30 \text{ cm} + 7 \text{ cm} = 37 \text{ cm} \\ \Delta &= -10\alpha \log_{10}(30/37) \approx 1.8 \text{ dBm}. \end{aligned} \quad (4)$$

When the Wand is close to the target device, the distance distance between antennas A_1 and A_2 is large relative to the distance to the transmitter. In that case the difference between received power on the two antennas on the Wand will be large. For example, assume the transmitter in Figure 1 is located 1 cm from A_1. In that case $d_1 = 1$ cm and $d_2 = 8$ cm and the expected RSSI difference is approximately 18.1 dBm.

When the Wand is in close proximity to a transmitting device, the difference in power readings between the Wand's two antennas will be significantly larger than the difference in power readings when the device is far away. In this example there is an expected 10-fold increase in the RSSI Ratio when the Wand moves from 30 cm to 1 cm between the transmitter and A_1. Figure 2 shows how the expected power changes as the distance between the device and transmitter changes.

Wanda determines whether the Wand and device are in close proximity by examining the average RSSI Ratio according to the following procedure:

$$\bar{\delta} = \frac{1}{\omega} \sum_{i=1}^{\omega} r_1(i) - r_2(i) \quad (5)$$

where i is the i^{th} packet transmitted and $r_1(i)$ is the RSSI for packet i measured on antenna A_1, $r_2(i)$ is the RSSI for the same packet measured on antenna A_2, and ω is a window size of the number of RSSI packets received. If the average difference $\bar{\delta}$ rises above a predetermined threshold τ, then the Wand declares it is in close proximity to the transmitting device. The Wand waits to check for proximity until it has received at least ω packets, and re-checks for proximity every $\omega/2$ packets afterward using the last ω RSSI values until it detects it is close to the device or times out.

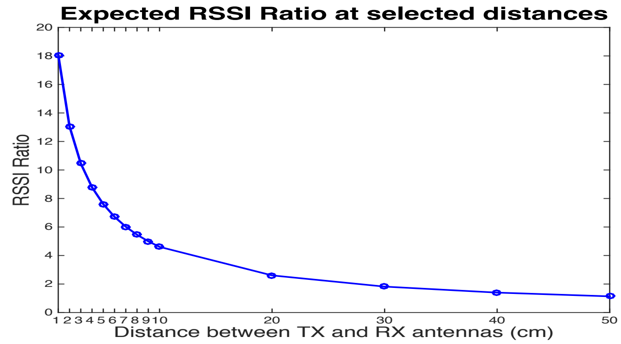

Fig. 2. Expected difference in RSSI with d_1 ranging from 1 to 50 cm.

In this way, the Wand can determine when it is in close proximity to a transmitting device even if the device has only a single antenna. If the device has multiple antennas, Wanda assumes it will transmit packets using only one of its antennas and will not change transmitting antennas while executing the *detect* primitive.

To execute *detect*, the user expresses the intent to start the process by taking an action such as pressing a button on the target device. The target device then begins broadcasting an *AssocReq* packet every 50 ms indicating that it is looking to connect with another device. The Wand uses those broadcast packets to determine whether it is in close proximity to the device using Equation (5).

The Wand can provide its user visual or audio feedback to encourage the user to move the Wand closer if needed. The Wand can change a row of LED lights or increase (decrease) the frequency of an audio tone if the spread between RSSI readings on the two antennas is becoming larger (smaller) to indicate if the Wand is getting closer to (farther from) the target device. Additionally, a visual indicator such as a sticker bearing a Wanda logo could be affixed on top of the antenna location on the target device to make *detect* easier. The user would then simply move the Wand close to the sticker and initiate the *detect* process.

B. *Impart primitive*

Once the Wand has determined it is close proximity to another device, it can exploit a property of radio wave propagation called *reciprocity* to impart information onto another device. Reciprocity says that a signal will experience the same multipath properties (e.g., attenuation phase shifts, delays) in both directions of the link [2]. This means that transmitting from the target device to the Wand has the same fading characteristics as transmitting from the Wand to the target device. As we saw above, the Wand should see a large RSSI Ratio when a transmitting device is close to the Wand.

Similarly, due to reciprocity, the device should see a large difference in RSSI when the Wand transmits from antenna A_1 vs. when it transmits from antenna A_2.

Wanda exploits the expected difference in RSSI on the target device to impart information. The Wand first converts the data to impart onto the device into a binary string m and then sends m one bit at a time. To send a 1, the Wand sends a packet using the closest antenna, A_1. To send a 0, it sends a packet using the farthest antenna, A_2. If the Wand and device are physically close together, the device will see a large difference in RSSI depending on which antenna the Wand used.

To decode the message m sent by the Wand, the target device simply calculates the average RSSI over all packets received and then compares the RSSI value for each packet with the average RSSI over all received packets. If the RSSI for an individual packet is above the average, the target device declares the packet to be a 1. If the RSSI is below the average, the target device declares the packet to be a 0. More formally:

$$\bar{r} = \frac{1}{n} \sum_{i=0}^{n} r(i) \tag{6}$$

$$\hat{m}(i) = \begin{cases} 1 & \text{if } r(i) \geq \bar{r} \\ 0 & \text{if } r(i) < \bar{r} \end{cases}$$

where $r(i)$ is the RSSI measured on the single antenna of the target device for packet i and $\hat{m}(i)$ is the i^{th} bit in the message received. Once this process is complete the device will have a string $\hat{m}$ representing the string m sent by the Wand.

To ensure the target device is not missing any bits in message m due to dropped packets, each packet sent by the Wand carries an increasing sequence number in the payload. The target device uses the sequence number of each packet to determine whether it missed any packets. If any packets are missing the device requests a resend of only those missing packets; otherwise it sends an empty list to the Wand.

To be clear, the information is transferred using the RSSI alone – the packets themselves sent do not contain portions of the message m. The payload only contains a sequence number so the target device can identify any missing bits.

The Wand sends the entire message without waiting for acknowledgement from the target device. When all message bits have been transmitted, the Wand sends a *Done* packet. The *Done* packet is similar to a *Message* packet, but it also includes a hash of m in the payload. Once the target device receives the *Done* message, it computes the value for each bit, creating message $\hat{m}$ on the target.

Finally, the target device hashes $\hat{m}$ and compares it with the hash of m included in the *Done* packet. If the hashes match, the device received all of the packets correctly. If the hashes do not match, the target device tries flipping each bit in $\hat{m}$, one at a time, re-hashes, and compares with the hash sent by the Wand. If a match is still not found, the target device follows a similar pattern but tries flipping two bits. If a match is still not found, the target device signals the Wand to restart by sending a *Restart* packet. If a match is found, the device transmits a *Success* packet to the Wand.

If the message to be imparted is long, it could be sent in chunks to enable the target device to efficiently flip bits. On the other hand, if messages are short they may be susceptible to an adversary discovering the message by brute-force flipping bits and hashing. To protect against these potential exploits Wanda can chunk long messages and pad short messages into 128-bit messages.

IV. Protocols

Wanda uses the primitive operations *detect* and *impart* described above to build protocols for configuring new devices. In this section we define three higher-level protocol operations.

A. Common Key protocol

The *Common Key* protocol is used when a new device must be configured with information common to all devices in a local-area network such as the blood-pressure monitor described above. The blood-pressure monitor must learn the SSID and password of a Wi-Fi AP. In this case we expect the Wand has earlier learned the SSID and password from the Wi-Fi AP over a wired USB connection. One can imagine the Wand being a 7 cm stick that lives in the USB port of the AP, keeping its batteries charged so it is ready when needed, and using the USB to securely obtain the connectivity parameters from the AP.

The Wand and target device then implement the *Common Key* protocol as follows: the Wand and target device run the *detect* primitive to determine if they are close together. Once the Wand determines it is in close proximity to the target device it runs the *impart* primitive to send the SSID and password to the target device. After the target device has confirmed it has properly received the message, flipping bits if necessary as described in the *impart* primitive, the target device connects to the Wi-Fi AP using the SSID and password it received, and the Wand is then not required for future communications.

B. Unique Key protocol

A slightly more complicated scenario arises when a user wants two devices to establish a connection using a key that is unique to those two devices. In this case the Wand can facilitate the introduction of the devices. The *Unique Key* protocol starts with the Wand generating a random key R. The Wand and Device 1 run *detect* and *impart* to send R to Device 1. The Device 1 includes its IP address (if it has one) in the payload of the *Success* message at the end of *impart* and the Wand notes the IP address as well as the MAC address of the target device from the packet headers. The user then carries the Wand close to Device 2 and the Wand then imparts R plus the MAC and IP address of Device 1 to Device 2 using *detect* and *impart*. Device 2 can now open direct communications with Device 1 by sending a hash of R to Device 1 at the MAC or IP address obtained from the Wand. Device 1 receives the hash from Device 2 and hashes its own copy of R. If the hashes match, then Device 1 bootstraps a MAC or IP layer connection with Device 2 using R as an initial key. If the hashes do not match, Device 1 does not attempt the connection.

C. Copy and Paste protocol

A third Wanda protocol is *Copy and Paste*. In *Copy and Paste* one device has information that the user would like imparted onto another device, although there may be no need for the devices to form a lasting pair as in the *Common Key* or *Unique Key* protocols. An example of where *Copy and Paste* could be useful is the blood-pressure monitor scenario described above. As shown above, the patient can use the *Common Key* protocol to link the blood-pressure monitor to a Wi-Fi AP, and while that solves the problem of getting a long-range communication connection for the short-range blood-pressure monitor, it does not solve the problem of getting the data stored in the patient's EHR. For data storage to happen the blood-pressure monitor (or perhaps the Wi-Fi AP) must know *where* and *how* to send the data. The blood-pressure monitor must know things such as a Restful API URL to send the medical readings, as well as the patient's credentials such as ID and password so the data can be stored in the correct patient record in the EHR.

Copy and Paste is designed to solve this problem. Continuing with the medical example, the patient brings the Wand to the doctor's office and performs the *Copy* phase by using *detect* and *impart* to send a random key R onto a device in the doctor's office. The doctor's office device encrypts the patient's credentials using R as a key and sends the resulting cypher text c to the Wand. The Wand stores the cypher text until the patient returns home. The patient then performs the *Paste* phase by using *detect* and *impart* to send random key R and cypher text c to the blood-pressure monitor. The blood-pressure monitor then decrypts the data and begins sending data to the doctor while the Wand deletes the cypher text. In this way, the *Copy and Paste* protocol copies the data from one device and pastes it onto another device, even though the devices may be physically far apart.

V. Implementation

We implemented a Wand prototype using a Raspberry Pi 2 Model B computer [6] connected to two external Panda Ultra Wireless N USB Wi-Fi adapters [7]. Figure 3 shows a photo of the prototype Wand and medical device. A production version would benefit by using one Wi-Fi card that has multiple antennas (commonly found on 802.11n or 802.11ac Wi-Fi devices). This single-radio, dual-antenna approach would ensure consistent energy is transmitted by the two antennas and could help reduce the potential for fingerprinting attacks [8], [9] by generating the radio frequency energy from the same source.

We used an FDA approved A&D Medical UA-767PC blood-pressure monitor [10] as the target device. Because we were unable to modify the software on FDA approved medical devices, we added an external Raspberry Pi with a single Alfa Networks AWUS036H Wi-Fi antenna [11] and connected to the blood-pressure monitor using a RS-232 over USB connection. This gave us the ability to extract the blood-pressure readings from the blood-pressure monitor using the RS-232 connection and the ability to communicate with the Wand over the single Wi-Fi antenna. Of course the manufacturer of the medical device would be able to alter their software to include the Wanda protocols (Wanda does not require hardware modification as long as the device has wireless connectivity), but our prototype demonstrates that even an existing device without a radio can be easily retrofitted to the conform to Wanda. We imagine the retrofit device to be a small dongle instead of our prototype Raspberry Pi-based system.

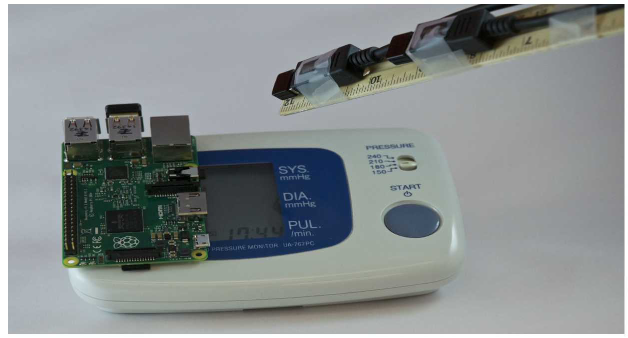

Fig. 3. Prototype Wand and A&D Medical blood-pressure monitor as target device (some cables removed for clarity).

We then used the prototype Wand to impart two types of information onto the retrofit blood-pressure monitor. First we imparted the SSID and password of a local Wi-Fi AP so the device could establish a connection and get to the internet. Second, we imparted the URL and a username and password for a Restful API representing a web service end point into a medical Electronic Health Record (EHR) in the cloud. The result is that now when someone measures their systolic, diastolic, and pulse, the Raspberry Pi reads those measurements and securely passes them to the simulated EHR.

We used Python and Scapy to create Wi-Fi data packets in our prototype and packets were sent at Layer 2. While our prototype used Wi-Fi, the technique could also be adapted for other protocols such as Bluetooth or Zigbee.

VI. Evaluation

We evaluated both the *detect* and *impart* phases of Wanda. For the evaluation we used the same software as our prototype, but for easier control and monitoring of our experiments we used a MacBook Pro instead of a Raspberry Pi.

A. Detect tests

We conducted 1,000 trials of the *detect* primitive where the distance d_1 between the Wand's A_1 antenna and the device's antenna ranged between 1 and 50 cm. Trials were conducted at 1 cm intervals from 1 to 10 cm, then at 10 cm intervals from 10 to 50 cm for a total of 14 distances with 1,000 trials each. The percentage of trials where the Wand detected it was in close proximity to the device is shown in Table I using a window size $\omega = 20$ and a threshold value $\tau = 6.2$. We chose this value for τ because the equations in Section III estimate that *detect* will declare the devices in close proximity when d_1 is less than 6 cm. We found that at distances less than 5 cm, proximity was detected 100% of the time. At 5 cm proximity was detected 87% of the time, and at 6 cm proximity was detected 38% of the time. At distances longer than 6 cm proximity was not detected. These results suggest that *detect*

Distance	*Detected close*
<5 cm	100%
5 cm	87%
6 cm	38%
>6 cm	0%

TABLE I
Percentage where *detect* primitive detected close proximity.

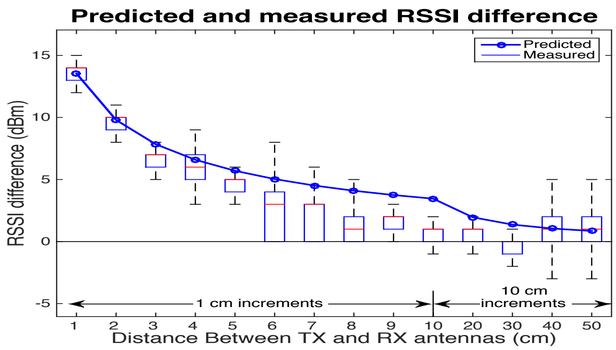

Fig. 4. Observed RSSI differences on a single-antenna device from 1,000 pairs of packets sent by the Wand alternating between antennas. The box represents the 75th and 25th percentiles of the observed RSSI differences, the red line is the median, and the whiskers represent the range of differences. The predicted RSSI difference according to Equation (3) is shown with $\alpha = 1.6$.

was able to correctly determine when it is in close proximity to the device with high probability.

B. Impart tests

We tested Wanda's ability to correctly impart data by first confirming the RSSI differences behaved as expected, then sent 1,000 messages from the Wand to the target device at various distances and counted bit errors to determine the Wand's effective range. Finally we measured how fast the Wand could impart information on target devices.

1) RSSI differences: To confirm that a single-antenna device is able to correctly receive a message when using the *impart* primitive, we tested whether it would measure a significant difference in RSSI based on the Wand's transmitting antenna (A_1 or A_2) as predicted by the equations in Section III. In these tests, the Wand sent 1,000 Wi-Fi data packets from each of its two antennas, alternating between antenna A_1 and A_2, where the distance d_1 between antenna A_1 and the device ranged from 1 to 50 cm and the distance d_2 was 7 cm larger than d_1. For this experiment, each *Message* packet contained a sequence number as specified in the *impart* primitive, as well as an indication of which antenna sent the packet to avoid confusion over which antenna actually sent the packet.

The target device recorded the RSSI of each packet and calculated an RSSI difference for each of the 1,000 pairs of packets it received. The results are shown in Figure 4 along with the RSSI difference predicted by Equation (3). The plot shows that the values observed mirror the predicted values when $\alpha = 1.6$.

2) Bit errors: Next we measured how well the Wand was able to impart information on another device. We ran 1,000 trials where the Wand sent a 128-bit random message to a single-antenna target device, and then counted the number

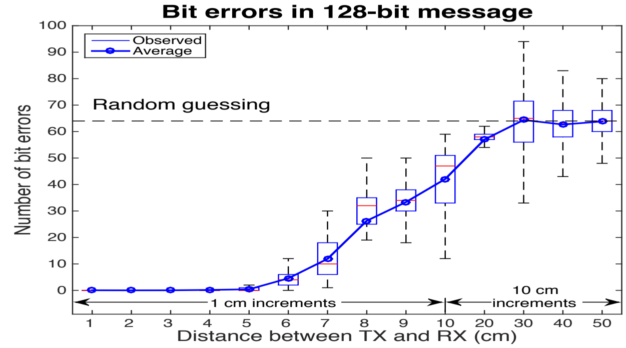

Fig. 5. Bit errors decoding a 128 bit message. The box represents the 75th and 25th percentile, the red line is the median, and the whiskers represent the range of bit errors.

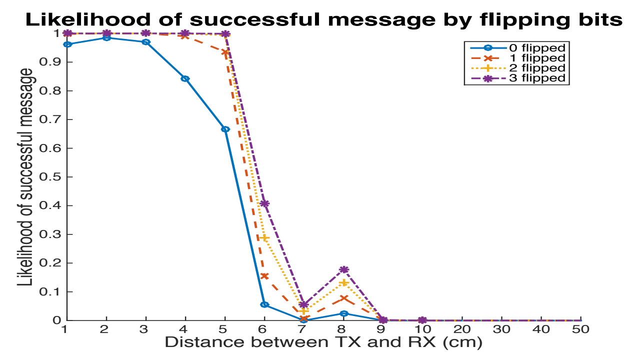

Fig. 6. Likelihood of successful message by flipping up to three bits.

of mismatched bits. Figure 5 shows that very few bit errors occurred at close range, but the number of errors increased significantly as distance between the Wand and the receiver, d_1, increased. Because each message contained 128 bits, random guessing should yield 64 correct bits. In our experiments this began to happen at a distance of about 30 cm.

Some of these errors can be corrected with the bit-flip technique described above where the target device flips bits in its derived message $\hat{m}$ and re-hashes. Figure 6 shows the percentage of successful message transfers at distance from 1 to 50 cm, correcting bits when needed, by flipping zero to three bits. From this graph we see that messages were transferred with a high probability of success when the Wand was less than 6 cm from the device.

3) Timing: We also measured the speed at which Wand was able to impart a message. We found the average time to send 128 bits was 0.454 seconds. This translates to just over 280 bits per second. We note that our implementation was written in Python. An implementation in C might have seen even faster throughput, although for many applications transferring a message in under half a second is acceptable. Long messages can be sent by imparting a key and then using that key to encrypt normal packets carrying data in their payload.

VII. Security

In prior sections we show that Wanda works well; in this section we evaluate its security against passive adversaries attempting to eavesdrop on communications between the Wand and the target device, and active adversaries attempting to inject malicious information onto the target device or Wand. We assume an adversary has complete knowledge of the Wanda protocol and can use that knowledge to try to exploit the system.

We assume the adversary:

- is able to receive, tamper with, or inject packets into the communications between the Wand and target device,
- is able to modulate its transmit power,
- may have multiple antennas and be positioned at multiple locations,
- does not try to jam the communications channel, creating a denial of service,
- does not have physical access to tamper with the Wand or target device, and
- is located more than 30 cm away from the target device and Wand while they are communicating.

A. Eavesdropping

Because the bits in the message m sent by the Wand are encoded only in the Wand's choice of transmitting antenna, an adversary must determine which antenna sent a packet in order to decode the information transferred. There are three main ways this could be done by an adversary: (1) receive packets from only one Wand antenna, (2) use the environment to differentiate between antennas, and (3) analyze the RSSI to differentiate between antennas.

Receive packets from only one Wand antenna:

If it were possible for an adversary to receive packets sent by only one of the Wand's antennas – not both – the adversary would be able to determine which antenna sent all of the bits in a message. The adversary would simply list the packet sequence numbers it receives and infer those packets represent a bit with a value of 1. For the sequence numbers the adversary does not receive, it can assume those packets came from the other antenna on the Wand and infer those represent a bit value of 0. After all the packets are sent, if the adversary does not drop any packets, the adversary will either be correct on all bits (the monitored antenna was actually sending 1s), or wrong on all bits (the monitored antenna was actually sending 0s) in which case the adversary simply flips all bits.

The adversary's dilemma is that both antennas on the Wand are close together and radiate energy that travels outward in a spherical shape. This makes receiving signals from only one antenna very difficult. An adversary could try to use a highly directional antenna and attempt to create a narrow main lobe pointed precisely at one of the antennas on the Wand. Given that the antennas on the Wand are only 7 cm apart, this is unlikely to work if the attacker is located a reasonable distance away because the main lobe expands with distance and should encompass both of the Wand's antennas.

Use the environment to differentiate between antennas:

An attacker might also attempt to determine which antenna sent a packet by detecting differences in the signal due to environmental effects. Because the characteristics of the received signal depend on the specific paths taken as the signal travels from the transmitter to the receiver, and signals from different transmit antennas might take different paths to an adversary, the adversary might be able to determine which antenna sent each packet. The chances of this attack succeeding, however, are vanishingly small. Cai et al. calculated the odds of an attacker succeeding with this type of attack from a random location to be 10^{-15} [1]. They go on to suggest that, in theory, an attacker might choose an ideal location by carefully measuring locations, geometries, and surface properties of all objects in the environment. While this precise measurement is practically impossible, nevertheless even that attack could be mitigated by incorporating a frequency-hopping scheme where each packet is sent on a different Wi-Fi frequency.

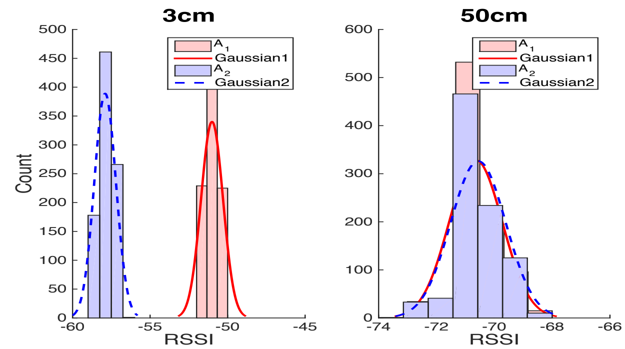

Fig. 7. RSSI distribution of 1,000 packets sent where d_1 = 3 cm and 50 cm.

Analyze the RSSI to differentiate between antennas:

Wanda uses a simple algorithm on the target device to determine which antenna sent a packet based on the RSSI, but we assume an adversary can use more sophisticated techniques. While we cannot anticipate every possible technique, we expect from Equation (3) that the difference in RSSI when the Wand uses antenna A_1 vs. when it uses antenna A_2 will be small when the Wand is not close to the adversary. Additionally, the environmental noise described in Section II increases as distance increases. Figure 7 illustrates these differences for 1,000 packets sent by antenna A_1 and 1,000 packets sent by antenna A_2 at $d_1 = 3$ cm and $d_1 = 50$ cm. As expected, the RSSIs of packets from the same transmit antenna form a Gaussian with a distinct mean (due to distance) and standard deviation (due to noise).

If an adversary were somehow armed with knowledge of the Gaussians of each antenna on the Wand, they might be able to determine which antenna sent a packet. When a packet arrives, the adversary could measure the RSSI and determine from which distribution that sample is drawn, that is, which antenna is most likely responsible for sending the packet. The distributions are constantly changing due to changing environmental factors, however, making this assumption of a priori knowledge of the Gaussians unrealistic.

Even if an attacker somehow did have perfect knowledge of the Gaussian distributions that characterize packets sent by each antenna on the Wand, the adversary will still suffer from a large number of errors when observing from long distances. Figure 8 shows that, even if armed with perfect knowledge of the packet distributions, an adversary only a short distance away would still make nearly 50% bit errors predicting which

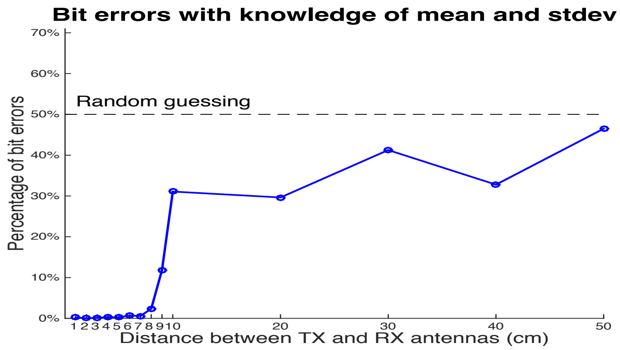

Fig. 8. Percentage of bit errors if adversary had perfect knowledge of RSSI distributions by antenna.

antenna sent a packet using the Gaussian distributions. We conducted those experiments with a prototype built with two radios (rather than one radio), cheap antennas (not specifically selected for a spherical radio dispersion pattern), and without precise antenna alignment (see Figure 3); a commercial Wand (with a single radio and two antennas selected and aligned carefully) would be even harder to attack in this manner.

B. *Malicious packets*

An active adversary may attempt to inject information onto the target device by tricking the target device into believing it is communicating with the Wand while the Wand is not actually present. Wanda defends against the attack by asking the user to declare the intention to start the protocol on the target device by taking an action such as pushing a button on the target device. This ensures that when the Wand is not present, the target device will not begin running the Wanda protocols. In that case, if an adversary were to try to communicate with the target device, the target device would not respond.

Alternatively, an adversary could try to override the information sent by the Wand while the Wand is communicating with the target device. To override the Wand, an adversary might modulate its transmission power; increasing power to send a 1 and decreasing power to send a 0. The target device, which may have only a single antenna, has no way of knowing if these modulated signals are coming from a nearby Wand or from a distant adversary because the RSSI of the packets would appear to the target device in the same way packets appear from the Wand. To prevent this attack, the Wand can monitor for rogue *Message* packets that it did not send. If it detects rogue packets, the Wand can send a *Stop* packet to the target device to halt the process.

The Wand can protect *itself* from storing malicious data (as in the *Copy and Paste* protocol), by ensuring any received packets have a large RSSI ratio. This test would ensure the data came from a nearby target device, and not a distant attacker attempting to exploit the Wand.

VIII. Related work

Researchers have proposed many solutions to the problem of securely configuring new devices. While the proposed approaches vary widely, they can be categorized into two main groups: out-of-band (OOB) and in-band communications. In OOB solutions a secret key is exchanged between devices over a secondary communication channel that is impervious to observation and interference by an adversary; the devices then bootstrap a secure connection over the primary channel using the information exchanged over the secondary channel. In-band approaches differ in that they only use the primary communication channel to establish a secure connection. In this section we examine some of the proposed solutions and highlight some of their differences with Wanda.

A. *Out-Of-Band*

Systems employing an OOB approach use a secondary channel to exchange secret information (e.g., a cryptographic key) that is used to secure the primary channel's communication. While many methods have been proposed, they often use the wired [12], visual [13]–[15], audio [16], [17], gesture [18], [19] or secondary radios such as RFID or NFC [20] channels to convey secret information. These approaches, however, assume the presence of hardware that may not be present on some devices and may also require complex processing that exceeds the capabilities of embedded devices.

Wanda differs significantly from these all of these approaches in that it does not assume the presence of specialized hardware other than the existing wireless radio, nor does it require advanced processing power. Furthermore, Wanda requires little human effort and the Wand's mobility allows it to be used when devices that are not physically adjacent or would be inconvenient to move (such as a treadmill and a Wi-Fi AP).

B. *In-Band*

Researchers have also suggested techniques that do not require an OOB channel, but instead exploit characteristics of the in-band radio channel. These techniques are typically more closely aligned with Wanda than OOB techniques.

Although Gollakota et al. developed an in-band method to defend against Man-In-The-Middle attacks [21], their approach alters the Wi-Fi protocol. Most in-band approaches, however, use characteristics of the radio channel to develop a secret key independently on two devices. To develop the secret key, each device typically goes through several phases. The first phase is bit extraction where each device monitors a common radio channel simultaneously and extracts bits from extreme signal fluctuations to form a string of bits. The next phase, reconciliation, ensures both devices have extracted the same bit string. Reconciliation normally involves several rounds exchanging information about portions of the bit string, such as checksums, in the clear. Finally, a privacy amplification phase reduces the size of the bit string to form a secret key that is known to the participating devices and unknown with high probability by an adversary [22]. Several works use a variant of this extraction-reconciliation-amplification approach [23]–[25].

The extraction-reconciliation-amplification approach has several shortcomings. First, it is quite slow, often taking 30 seconds or more to make connections. Wanda is fast, taking less than half a second on average to send a 128-bit message.

Another problem is that Wi-Fi, in many practical environments, lacks the necessary entropy to extract a secure bit string [4]. Wanda does not rely on random environmental fluctuations to generate common bits on two devices; it imparts the bits onto a target device based on the antenna chosen by the Wand.

Wanda does share common elements with two papers. In Good Neighbor [1] the authors use the equations in Section III of this paper to determine whether a sending device with a single antenna is in close proximity to a receiving device with two antennas. Good Neighbor, however, runs 8 times slower on average than Wanda and only protects the two-antenna receiver – it does not protect the single-antenna sender. For example, using the Good Neighbor final protocol, if an adversary sends its public key to the sender before the receiver does (as in a Man-In-The-Middle attack), the adversary can pair with the device for 11.64 seconds on average before the receiver determines its pairing failed and alerts the user. During that time the sending device has no idea it is connected to an attacker. Furthermore, when the user discovers the intended receiver is not connected, the user will likely suspect the pairing simply failed and may re-start the connection process, leaving the attacker with an ongoing valid connection. As noted in Section VII, however, Wanda protects both devices while they communicate. Also, with Good Neighbor at least one of the devices must be mobile so two devices can be placed in close proximity. If both devices are difficult or impossible to move, then Good Neighbor will not work. With Wanda, however, the Wand easily can move close to multiple non-mobile devices.

Another recent approach called SeAK [26] uses two antennas to develop a secret key, but in that paper each device independently develops a key based on the RSSI of exchanged packets. In Wanda, the Wand knows the secret information and imparts it onto the other device without the need for the Wand to develop the same key as the target device.

IX. Conclusion

In this paper we introduce a system called **Wanda**. Wanda is able to simply, securely, and consistent with user intent, impart data onto devices. Among other uses, this data can be used for three fundamental operations when bringing a device into a new setting: (1) configure new devices to join a wireless local-area network (using *Common Key*), (2) partner devices with other nearby devices so they can work together (using *Unique Key*), and (3) configure devices so they can connect to accounts in the cloud (using *Copy and Paste*). Wanda does this by implementing two primitive operations, *detect* and *impart*, which allow a new piece of hardware called the Wand to detect when it is physically near another device, then impart information onto that nearby device using a novel radio signal strength method of communication. Experiments with our prototype implementation show that Wanda is fast and effective, and our security analysis demonstrates that it should be resistant to passive and active adversaries. Indeed, we expect Wanda is faster, easier, more flexible, and more secure than existing alternatives for device pairing and for intentional interaction with wireless devices.

X. Acknowledgements

This research program is supported by National Science Foundation award number CNS-1329686. The views and conclusions in this document are those of the authors and may not necessarily represent the official policies of NSF.

References

[1] L. Cai, K. Zeng, H. Chen, and P. Mohapatra, “Good neighbor: Ad hoc pairing of nearby wireless devices by multiple antennas.” in *NDSS*, 2011.
[2] T. S. Rappaport, “Wireless communications: principles and practice,” *Prentice-Hall*, 2002.
[3] T. J. Pierson, X. Liang, R. Peterson, and D. Kotz, “Wanda: securely introducing mobile devices. Extended version,” Dartmouth College, Technical Report TR2016-789, 2016.
[4] S. Jana, S. N. Premnath, M. Clark, S. K. Kasera, N. Patwari, and S. V. Krishnamurthy, “On the effectiveness of secret key extraction from wireless signal strength in real environments,” in *MobiCom*, 2009, pp. 321–332.
[5] W. Cheng, K. Tan, V. Omwando, J. Zhu, and P. Mohapatra, “RSS-Ratio for enhancing performance of RSS-based applications,” in *InfoCom*, 2013, pp. 3075–3083.
[6] Raspberry Pi Foundation. [Online]. Available: http://www.raspberrypi.org
[7] Panda wireless. [Online]. Available: http://www.pandawireless.com/
[8] B. Danev, D. Zanetti, and S. Capkun, “On physical-layer identification of wireless devices,” *CSUR*, vol. 45, no. 1, p. 6, 2012.
[9] I. R. Jenkins, R. Shapiro, S. Bratus, T. Goodspeed, R. Speers, and D. Dowd, “Speaking the local dialect: exploiting differences between IEEE 802.15.4 receivers with commodity radios for fingerprinting, targeted attacks, and WIDS evasion,” in *WiSec*, 2014, pp. 63–68.
[10] A & D Medical UA-767PC blood pressure monitor. [Online]. Available: http://www.andonline.com/medical/products/details.php?catname=&product_num=UA-767PC
[11] Alfa networks. [Online]. Available: http://www.alfa.com.tw
[12] F. Stajano, “The resurrecting duckling,” in *Security Protocols*. Springer, 2000, pp. 183–194.
[13] A. Brown, R. Mortier, and T. Rodden, “Multinet: Reducing interaction overhead in domestic wireless networks,” in *ACM CHI*, 2013, pp. 1569–1578.
[14] N. Saxena, J.-E. Ekberg, K. Kostiainen, and N. Asokan, “Secure device pairing based on a visual channel: Design and usability study,” *WIFS*, vol. 6, no. 1, pp. 28–38, March 2011.
[15] M. Sethi, E. Oat, M. Di Francesco, and T. Aura, “Secure bootstrapping of cloud-managed ubiquitous displays,” in *UbiComp*, 2014, pp. 739–750.
[16] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun, “Loud and clear: Human-verifiable authentication based on audio,” in *ICDCS*, 2006, pp. 10–10.
[17] C. Soriente, G. Tsudik, and E. Uzun, “HAPADEP: human-assisted pure audio device pairing,” in *Information Security*, 2008, pp. 385–400.
[18] R. Mayrhofer and H. Gellersen, “Shake well before use: Intuitive and secure pairing of mobile devices,” *IEEE TMC*, vol. 8, no. 6, pp. 792–806, 2009.
[19] C. Soriente, G. Tsudik, and E. Uzun, “BEDA: Button-enabled device association,” *IWSSI*, 2007.
[20] NFC Forum. [Online]. Available: http://nfc-forum.org
[21] S. Gollakota, N. Ahmed, N. Zeldovich, and D. Katabi, “Secure in-band wireless pairing.” in *USENIX Security*, 2011.
[22] C. H. Bennett, G. Brassard, and J.-M. Robert, “Privacy amplification by public discussion,” *SIAM Computing*, vol. 17, no. 2, pp. 210–229, 1988.
[23] S. Mathur, R. Miller, A. Varshavsky, W. Trappe, and N. Mandayam, “ProxiMate: Proximity-based secure pairing using ambient wireless signals,” in *MobiSys*, 2011, pp. 211–224.
[24] L. Shi, M. Li, S. Yu, and J. Yuan, “BANA: Body area network authentication exploiting channel characteristics,” *J-SAC*, vol. 31, no. 9, pp. 1803–1816, 2013.
[25] K. Zeng, D. Wu, A. Chan, and P. Mohapatra, “Exploiting multiple-antenna diversity for shared secret key generation in wireless networks,” in *InfoCom*, 2010, pp. 1–9.
[26] C. Javali, G. Revadigar, L. Libman, and S. Jha, “SeAK: Secure authentication and key generation protocol based on dual antennas for wireless body area networks,” *RFID Security*, 2014.

