An Amulet for trustworthy wearable mHealth

Jacob Sorber, Minho Shin*, Ronald Petersen, Cory Cornelius, Shrirang Mare,
Aarathi Prasad, Zachary Marois, Emma Smithayer, David Kotz

Department of Computer Science, Dartmouth College, Hanover, USA
{firstname.lastname}@dartmouth.edu

*Computer Engineering, Myongji University, South Korea
shinminho@gmail.com

ABSTRACT

Mobile technology has significant potential to help revolutionize personal wellness and the delivery of healthcare. Mobile phones, wearable sensors, and home-based tele-medicine devices can help individuals (and their caregivers) to better monitor and manage their health. While the potential benefits of this "mHealth" technology include better health (and more effective healthcare) and reduced cost, this technology also poses significant security and privacy challenges. In this paper we propose *Amulet,* an mHealth architecture that provides strong security and privacy guarantees while remaining easy to use, and outline the research and engineering challenges required to realize the Amulet vision.

1. INTRODUCTION

Mobile technology has significant potential to help revolutionize personal wellness and the delivery of healthcare. Smart phones, coupled with wearable sensors (like pedometers or cardiac monitors), implanted medical devices (like insulin pumps or pacemakers) and home-based tele-medicine devices (such as bathroom scales or blood-pressure cuffs) can help individuals (and their caregivers) to better monitor and manage their health [16].

Products are already emerging to support long-term continuous medical monitoring for outpatients with chronic medical conditions (such as diabetes or cardiac rehab) [19,21], individuals seeking to change behavior (such as losing weight) [5], physicians needing to quantify and detect behavioral aberrations for early diagnosis (such as depression) [4, 15], or athletes wishing to monitor their condition and performance [**?**], to name a few examples. [xxx Shrirang, do we have room to add citations that provide an example of each of the above cases? [shri] I added some examples. We will remove them if we run out of space, which I think we might.] In this paper, we use the term "Patient" to describe the subject of sensing in all such use cases, using the capitalized form as a reminder of its broader meaning.

The data is typically stored in the Patient's mobile phone, or (increasingly) in a cloud-based health records system (HRS) operated by a healthcare provider or device vendor. The resulting data may be used directly by the Patient [1, 20] or may be shared with others: with a physician for treatment [17], with an insurance company for coverage, with a scientist for research [7, 14], with a coach for athletic training [3], or with family members and friends in social-networking communities targeted towards health and wellness [8, 13, for example]. [xxx Shrirang, could you coordinate an update to the above list of citations, resulting in at most 5 citations. The current list are outdated and not necessarily a good fit to each example. [shri] klasnja:pocket [11] is a recent (2011) jr paper, and has good citations and examples for the cases: 'data directly used by the Patient', patient sharing data with physicians and getting feedback, and patient sharing data with people in his social network; fenu:athlete09 [?] is an athlete example; didn't find any good citations for insurance example.]

While the potential benefits of this patient-centric form of "mHealth" technology include better health (and more effective healthcare) and reduced cost, this technology also poses significant security and privacy challenges: to be successful, mHealth technology must be (1) trusted by the Patient to ensure the privacy of the personal information collected, (2) trusted by both Patient and Provider to ensure the integrity of the data and the security of any actuators in the system, and (3) usable without technical expertise. Current approaches fail to provide the desired security, privacy, or usability goals, or are limited to a specific solution isolated to a particular product. In this paper we propose *Amulet,* an mHealth architecture, shown in Figure 1, that provides strong security and privacy guarantees while remaining easy to use, and outline the research and engineering challenges required to realize the Amulet vision.[xxx does Figure 1 still help here if we have all of the variants outlined in the table in the next section? —JACOB]

To enable trustworthy patient-centric mHealth we need to ensure several important properties. The system must provide data confidentiality (avoiding exposure of patient data to unauthorized parties), data integrity (to protect data from tampering, or replay of stale data), data authenticity (to ensure that the data comes from the correct sensor, on the correct patient), data availability (limiting data loss and latency), and command authenticity and integrity (ensuring that commands sent to actuators are not forged, tampered, or replayed). Furthermore, given the likely use of wireless body-area communications, the system should protect patient anonymity (preventing bystanders from learning the Patient's identity or inferring their medical condition). Most challenging, such systems must also support interoperability and modularity (to avoid device proliferation), and ease of use (regarding both functionality and security). In our earlier work, we provide a comprehensive survey of privacy in the context of mHealth [2], a body-area protocol with strong security and anonymity properties [12], and an overview of the challenges of assuring data quality in mHealth [18].

Existing approaches do not provide all of these properties, however. Although space is too limited to list all existing systems, all of them fit into one of four models, and all models have significant limitations, as follows. [xxx I liked the table but I think we need the words, to be able to explore the problems with each model. A more-compact list format would be good.]

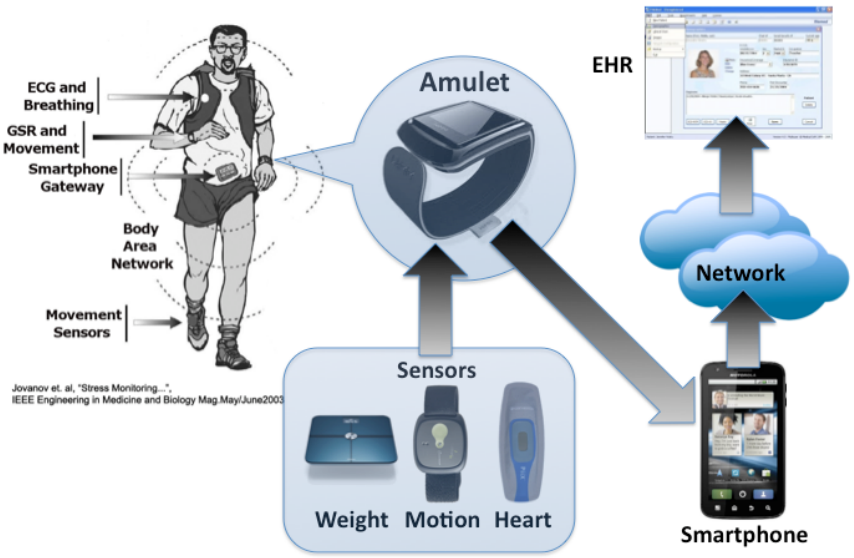


Figure 1: Patient-centric mHealth example, comprising a set of sensor nodes (SN), a mobile phone (MP), a wrist-worn Amulet, and a cloud-based health records system (HRS).

Smartphone only: many apps use the phone's internal sensors for health-related monitoring; but some health conditions require other sensors or require contact with a specific part of the body. And, what if the smartphone is compromised (smartphone malware is an increasing risk), or the phone is lost, set aside, or lent to someone else?

Smartphone + wearable or home device: an app may communicate with wearable devices, or nearby devices such as a blood-pressure cuff. Again, what if phone is absent, lost, or compromised? Does the wireless body-area network (WBAN) technology provide sufficient security and anonymity? Most phones are limited to Bluetooth, and cannot communicate with ANT or Zigbee devices. Are we sure the phone is on the same body as the sensor [6]? Are they on the *right* body?

Home base station + wearable or home device: in many current products, the device vendor provides both a wearable sensor node and a proprietary base station (or dongle for a PC); the station provides a gateway to the Internet and some limited processing or user interface. Do you need a base station for every device? The homeowner must configure every base station for their home network (inconvenient), or every base station must have an independent vendor-provided WWAN connection (expensive, and works only with cell coverage). What if the Patient is away from the home base for an extended period?

Standalone wearable or home device: some home devices require no smartphone or base-station for communication; they include an integrated cellular or Wi-Fi connection. Most of the above problems remain; and, for wearable sensors, a WLAN or WWAN network connection will require more space, cost, and power than may be feasible in a compact device. Furthermore, the sensor device may be more complex (and complexity leads to bugs), and an independent network connection increases exposure to network-based attackers.

Many interesting mHealth applications involve sensors that must be on the body, periodically or continuously, obviating the first approach; but providing the necessary computational and network infrastructure on every sensor node is too expensive, obviating the last approach. Hence the need for a smartphone or base station to provide computational and network support. A base station remains at home, and thus is not always present. A smartphone provides portability, and yet may be set aside, left behind, lost, or lent to another, thus it too may not be present; as general-purpose computing platforms, smartphones are difficult to secure [xxx cite]. Many critical apps (monitoring the heart for atrial fibrillation, or managing blood glucose through an insulin pump) requires continuous presence of a trusted device.

Our position. To reach their full potential in transforming healthcare delivery, wearable networks of sensors and actuators must be able to operate continuously and securely without relying on smartphones and other non-wearable personal computing devices. We need a personal device that stays with the user at all times, can authenticate its wearer, can be secured independently of other applications on the smartphone or home computer, can provide a trustworthy interface to the user, and can provide computational support and a network gateway for low-power body sensors and the smartphone or other Internet gateway.

This paper describes our research vision for such a wearable, trusted platform, called Amulet. An amulet is "an ornament or small piece of jewelry thought to give protection against evil, danger, or disease" [MacOS dictionary]; at its core, Amulet is a trustworthy wearable device in a wrist-watch form factor, and provides the properties described above. Unlike prior approaches, Amulet is designed to enable continuous sensing and actuation, requiring a wireless gateway (mobile phone or access point) only for occasional connectivity to back-end servers and other off-body network resources.

When complete, Amulet provides the following contributions:

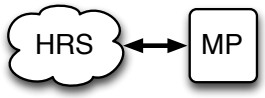
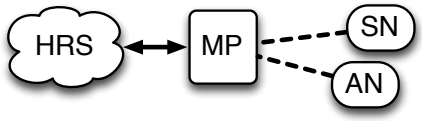
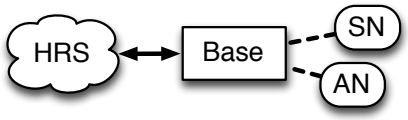
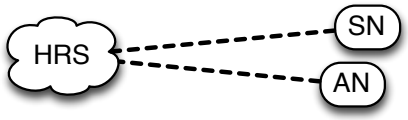
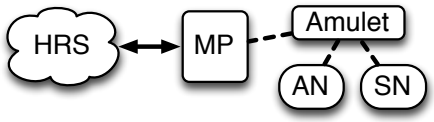
	Smartphone only (MP = mobile phone)
	Smartphone + wearable or home-based actuator(AN) or sensor(SN) device
	Base station + wearable or home-based AN or SN device
	Standalone device with direct connection to HRS
	Amulet model
	[xxx FIXME: Need to add text describing the disadvantages of these approaches. Then tell why Amulet is better.]

Table 1: Approaches to patient-centric mHealth.

1. Amulet is a trustworthy hub for patient-centric mHealth that is omnipresent, interoperable, usable, and secure.
2. Amulet provides a means to authenticate its wearer, and to determine which set of sensors are on the same body.
3. Amulet provides a trustworthy path for communicating with its wearer, the Patient.
4. Amulet is a robust security architecture based on a tamper-resistant physical platform, a clean separation between the health-related applications on Amulet and the Patient's other applications on a smartphone or PC, and supports a secure method for software distribution and management.
5. Amulet provides a safer programming model for safety-critical processes, automating data provenance and other security aspects of the system.

In the next section we describe Amulet, and discuss its many advantages over the existing models. In subsequent sections, we dig deeper in to some of the key properties, anticipating the remaining research and engineering challenges required to realize the Amulet vision. In Section **??**, we describe how this architecture could also be applied to domains other than healthcare. Section 6 describes related work, and Section 7 concludes.

2. AMULET VISION

[xxx what is it—diagram]

[xxx how does it work? Authentication, interface, programming provenance]

[xxx story-style presentation?]

3. AMULET ADVANTAGES

In answer to the disadvantages of existing smartphone-based, base station-based, or standalone approaches to mHealth, we propose Amulet, a wearable approach to managing and securing a mHealth patient's body area network. An Amulet is a wireless computing device worn by a patient that

[xxx A lot of this was said in the intro. What needs to be stated (or restated here?)][xxx This DESCRIBE WHAT AMULET IS and DOES: worn on the wrist, authenticates devices to patients, identifies the patient, secures communication, authorizes code updates to apps and sensors, runs processing applications (close-loop or otherwise), connects the BAN to the Internet].

As a small form-factor special-purpose device that is tightly coupled to a patient's body, Amulet has significant usability, security, and interoperability advantages over existing smart-phone and home base-station-based systems:

[xxx in the intro and/or somewhere here we should say something about keeping wearable sensors simple to reduce the likelihood of recalls and errors, while shrinking the attack surface]

Usability is a critically important consideration for any mobile device, and even more-so for wearable devices. In order to be successful, patient-centered mHealth devices must be simple, require minimal configuration, and blend into the Patient's daily life.[xxx Become invisible—maybe cite Mark Weiser here? Maybe talk about tension: it needs to be invisible—so easy you forget

about it, but hard to lose]

One usability challenge to using smartphones for continuous mHealth applications is that phones are easily lost, left at home, and temporarily lent to others. A device that provides support for safety-critical sensors and actuators must be ever-present. In contrast to a smartphone that may be kept in a pocket, purse, or handbag, a wrist-worn Amulet is physically attached to the body and can comfortably be worn at all times (even when sleeping). [xxx Need to make sure that it is clear at this point why we need omnipresence.] While not all Patients regularly wear a wrist watch, Amulet is in a socially-accepted form factor, functional jewelry for both genders and most cultures, unlike other wearable alternatives. [xxx Maybe add somewhere that such a common form factor means that the presence of an Amulet does not reveal anything about the Patient's condition, or even that she has a condition.]

[xxx Amulet serves as an identity token when interacting with shared devices at home or in clinic, such as scale, BP cuff, or displays. Phone is not suitable for this purpose because they can be left behind or loaned to another.]

[xxx fewer people wear a watch these days: but if the device is comfortable, stylish, and adds value, people will.]

[xxx Waterproof amulet could even be worn while swimming or bathing. Phones can't do that.]

Amulet's form factor also ensures that it is always in the same position on the body, while a phone in a pocket or purse tends to be loose and even a holster may change position from day to day. This tight coupling to the Patient's body provides a natural opportunity for improved accelerometer-based activity recognition [**?**] as well as more seamless gesture-based interfaces that would otherwise require the Patient to first locate their phone. In addition, a device that is constantly in contact with the skin can also measure many of the Patient's physiological parameters, like heart rate, blood oxygenation, galvanic skin response (GSM), and electrical impedance that are useful in a wide range of medical contexts and may also be useful for automatically authenticating the Patient to the Amulet, without the need to enter pins and passwords. These possibilities are described further in Section 5.2.

[xxx Maybe say something about small UI resources not really being a curse—forces us to replace traditional security interactions (PINs & passwords) and other *pointing and clicking* interactions that often have usability problems even on larger devices (phones, tablets, laptops), with interactions that are more natural in a wearable context.]

Security is a significant challenge when using mobile phones in medical-grade applications. Today's smart phones are complex multipurpose computing platforms that host a variety of applications provided by different sources, some of whom the Patient may not trust. Dual-persona phones [**?**] can isolate critical applications from others, but the large attack surface of a smart-phones remains susceptible to malware [**?**] and other software-based attacks. [xxx look in PnT paper for citations, also I seem to remember a recent CUL addition on this topic.] [xxx [shri] We have one citation for malware [?]; these two [**?**, 9] are additional citations about malware in smartphones, in case we want to use them.] If successful, these attacks can violate a Patient's privacy, tamper with data needed to make diagnoses or prescribe treatments.

In spite of tighter resource constraints, a limited-function device running only mHealth apps (and normal wrist watch time and date functions) can provide much tighter security than a general-purpose phone. Reducing the range of applications that Amulet can support, makes it possible to manage execution more strictly. [xxx It also allows us to rethink our OS design, and design security mechanisms (keys, crypto, provenance) in as a first-class primitive]. [xxx smaller devices are easier to make tamper resistant—is this true, Ron?]

[xxx Maybe mention that smaller, simpler code base makes it easier to test (maybe even formally verify) correctness. Safety from bugs.]

Amulet's security can be strengthened further by allowing trustworthy output to and input from the Patient, on a small screen, with audible tones, or with vibration. Wrist is a glance-able display, better than any pocket or holster device. Phone cannot provide the same level of trust, because it is a general-purpose device open to malware. Phone also may not provide the same degree of privacy, because you may loan or leave your phone, so alerts showing there may be visible to others.

Amulet can obtain trusted input from the user, with a few buttons, a touch screen (maybe), or gestures (based on accelerometer and gyroscope). Later, speech recognition. BCC may also be possible, enabling touch-to-communicate via BCC from amulet through fingers.

[xxx aspects of this fit with interop as well.] Amulet can include tamper-resistant secure storage for keys - private keys for authenticating the patient, or public keys for authenticating the provider and back-end servers, or session keys with various devices and services. Phone is not suitable for this purpose unless a major manufacturer decides to build in secure storage and make it available to developers. Quantity of storage is likely sufficient; even microSD cards have O(GB) storage now, and can be secured with encryption.

Interoperability. In addition to the challenges of security, privacy, and usability, providing connectivity between wearable devices, mobile devices, and cloud-based services poses significant challenges. [xxx Mobile phones have WiFi for providing high-speed rich communication with the Internet and BlueTooth for infrequent communication with peripherals (like headsets).] The drive for long operational lifetimes has inspired the development of a wide range of low-power wireless technologies (Zigbee, BlueTooth Low Energy, ANT, [xxx are there others we should mention here?]). [xxx argue that phones makers are not going to add all of these to mobile phones. I'm betting that BTLE will end up in phones before the others for backwards compatibility reasons.] An Amulet with multiple low-power radios can make it possible to build mHealth WBANs from devices with heterogeneous radio technologies, and can provide a gateway between a WBAN and a mobile phone.

Adding Near Field Communication (NFC) capabilities to Amulet can provide a ubicomp interface with the surrounding world. Touch and scan-based interactions between an Amulet and NFC-compatible tags and devices can enable simple intentional pairing and other interactions that require and understaning of the intent of the Patient.

4. AMULET DISADVANTAGES

[xxx identify limitations and debunk unfounded disadvantages]

[xxx not everyone wears a watch]

[xxx another device to buy, configure, wear: because you wear it, you're not likely to leave it behind; because it's wearable, it's not hard to bring along; however, cost may be a factor.]

5. RESEARCH CHALLENGES

5.1 Hardware Platform

[xxx What HW do we need? Can existing platforms work? WIMM, MetaWatch, etc?] It is desirable that an Amulet device be small and wearable, have long battery life, an efficient low power radio, and sufficient processing power to handle data from a variety of sensor types. The small form factor means that current technologies will impose limits on what could be built today, though we expect radio and cpu efficiency as well as battery chemistries to improve over time. We can examine some workload requirements and compare them to the capabilities of existing devices to explore the feasability of building an Amulet.

Two common microcontrollers used in low powered devices are the TI MSP430 and the ARM processor family, used respectively by the commercial MetaWatch [1] and WIMM Labs WIMM platforms [2], which we consider here as candidates for an Amulet platform. Texas Instruments has published a benchmark study [3] of these processors comparing their performance in computing an optimized Finite Infinite Response filter (input is 51 16-bit values, the order of the filter is 17), which is representative of some tasks an Amulet might carry out. The results show a cycle count of 33,114 for the ARM7 and a cycle count of 107,146 for the MSP430. The WIMM MCU runs at 667MHz while the MetaWatch MSP430 has a max clock of 25MHz. Thus, with nothing else running on each MCU they could potentially compute 20,142 FIR's per second (ARM) and 233 FIR's per second (MSP430). However, these platforms achieve long battery life by spending most of their time in sleep mode, so experiments will need to be carried out to more accurately characterize what workloads are achievable.

Both the WIMM and MetaWatch platforms use ordinary Bluetooth radios which are not energy optimized for a BAN. Using Bluetooth Low Energy (BT LE) radios would greatly reduce the energy used for wireless transfer of data. Although we have not yet run experiments using BT LE radios, we do have experience with the TI CC1101 low power radio used in the TI Chronos wristwatch development platform, which has some similarities. The Chronos also uses an MSP430 MCU, similar to the MetaWatch. Tests have shown that we can send a continuous stream of three byte X,Y,Z accelerometer readings at 33Hz for about a week with the watch powered by a CR2032 battery. The CR2032 is specified at 3V with 220mAh capacity (2,376 joules). The WIMM platform has a 267mAh 3.7V Li-Poly battery (3,556 joules). So we expect that using a low power radio the MetaWatch and WIMM platforms, with a low continuous sample/packet rate (tens of Hz), could last up to a week before needing recharging. The WIMM ARM MCU may reduce the amount of power available, but the WIMM also has a lower power MCU built in which can let the main MCU sleep until enough data has arrived for processing.

[1] http://www.metawatch.org/
[2] http://www.wimm.com
[3] http://www.ti.com/lit/an/slaa205c/slaa205c.pdf

The research challenge here is to balance the form factor, CPU and radio characteristics, workload allowance, and battery life to design an Amulet that is useful for a wide range of applications.

5.2 Usability Wearable Security

In many ways, Amulet requires a fundamentally different approach to security. Not only is the device's processing power and battery life limited, but its user interface is limited as well. It would be infeasible, for example, for a user to input a password because of the few button and small screen. In fact, even if the user interface were more full featured to allow the input of passwords, users would still find passwords difficult to manage.[xxx Needs a good citation.] Thus, the security of Amulet must be mindful of both the processing and energy overhead as well as usability.

There are several security-related advantages of Amulet over the traditional type of devices envisioned. For example, the number of locations a smartphone can live varies over the course of even a day. A smartphone owner might carry it in their pocket or purse, hold it in their hand, or lend it to another person. The heterogenous nature of a smartphone's location relative to its owner makes it difficult to provide guarantees about the authenticity of the data. If the owner lends their smartphone to another person, then any data the smartphone produces in that timeframe must be labeled as having come from that other person. Ideally, this authentication process would be automated, but it is not clear how this can be done with a smartphone.

In contrast to a smartphone, Amulet is physically strapped to a Patient's body much like a watch. Thus, unlike a smartphone, it is unlikely a Patient will share their Amulet. Furthermore, Amulet can always know when it is strapped to a Patient and perform authentication to determine which Patient is wearing it. An Amulet could perform gesture recognition using a built-in accelerometer as a means of active authentication, or it could use physical characteristics of the Patient's wrist to passively authenticate. The same cannot be said about a smartphone since both it's location changes frequently and there is no obvious and usable way for a smartphone to authenticate which Patient is currently using it.

Supposing a smartphone could perform Patient authentication as above, the case of wearable sensor nodes presents another problem. Because wearable sensor nodes typically communicate with a smartphone wirelessly, the smartphone now needs to verify that the sensor is collecting data about the same Patient the smartphone has authenticated. Not doing so would violate the authenticity of the data since Patient A could authenticate with the smartphone while Patient B could be wearing the sensor. The smartphone would then be labeling sensor data about Patient B as coming from Patient A. Thus, the smartphone needs to be able to perform some type of same-body authentication when other sensor nodes are present.

While there is some existing research on the problem described above, the location of a smartphone at any given time makes it difficult to realize these authentication schemes. [xxx Needs citations] However, because of the relatively static location of Amulet, such same-body authentication schemes would be feasible.

[xxx Minimal Trusted I/O for basic configuring the network. A watch face, and a few buttons. What can we do with this? What configurations can be done at this level?]

[xxx Later versions of the TP could include speaker for speech-

synthesis output and microphone for spoken commands, much like some cellphones today, or speech-based authentication.]

[xxx We might also want under some conditions to leverage another display device's display/I/O. Can we preserve the trusted nature of the system even though we may not trust the other device. Display automatically reduces privacy guarantees we can make.]

[xxx Sean and Scout did some work with mechanisms to trust public kiosks etc. I'll ask Sean. —DAVE]

5.3 Programming Model

In addition to enabling automatic authentication (and other security mechanisms described in the previous section), as a more focused device, Amulet also has the opportunity to provide a much safer and trustworthy computing environment than a smartphone or PC. Amulet runs only one type of applications, which interact with a mobile health sensor, collect health information about a patient and forward it to the patient's health record. Using the Amulet, we can now separate the BAN management roles from the phone and take advantage of a more focused target. All the applications running on the Amulet are operating on the streams of data coming from mHealth sensors. We describe below the programming model for Amulet, to support these applications.

Installation When a patient buys a sensor from the pharmacy, the pharmacist installs a code on the sensor with the patient's information on it. The sensor will then keep searching for an Amulet belonging to the patient. Once it comes within range of the patient's Amulet, it requests to be paired with the Amulet. The Amulet verifies whether the sensor's authenticity; has it been bought from the pharmacy recommended by the patient's doctor? If valid, the Amulet sends the sensor symmetric keys for exchanging messages and hashes.[xxx A]re the keys exchanged between the app and sensors?The sensor then sends the Amulet the url to the application that the Amulet needs to download and install to manage the sensor. The Amulet verifies whether the url has been verified by a trusted authority, and if valid, it downloads the application and installs it. Once installed, the app requests the sensor for information like its sensor id, manufacturer id, time of manufacturer, seller id etc. This method works even in scenario 1, when the patient is not present when the sensor is purchased from the pharmacy.

Collection When the patient wants to use a sensor, she starts the app in the Amulet. The app then opens a secure channel for communication with the sensor. The sensor collects information about the patient, encrypts using the symmetric key provided by the Amulet and sends it to the app, along with a hash so that the app can verify whether the data came from the right sensor.

Processing Every app installed on the Amulet is paired with a patient's sensor. The app collects the data from the sensor and sends it to the patient's health record, so that it can be shared with the patient's health providers, family and friends. How can the data recipients verify whether the data is coming from the right sensor, was used by the right patient and in the right manner? The Amulet can help by collecting metadata which can then be used to verify the provenance of the patient's health data. When the app is developed for the Amulet to interact with a sensor, the manufacturer specifies the metadata that will be required for the data recipient to verify the provenance of the data collected by the sensor. When the app receives data from the sensor, it looks for sources to get the required metadata. For eg talk about the scenario and how to get metadata in that case.

Access When an emergency responder connects to the Amulet, an app is loaded, which interacts with all other apps running on the Amulet and retrieves information useful for the emergency responder, like the last 24 hour data that was recorded by each sensor. This "break the glass" app also has access to information about what conditions the patient is or was diagnosed with and what medications she is taking or had taken recently. Wondering whether I should talk about how the emergency responder is authenticate.

Uninstallation Not sure how this could be done. Is it sufficient to do a manual delete on the Amulet?

Output facilities Display a small amount of information on the watch, ability to vibrate

[xxx Safer programming for safety-critical systems.]

[xxx Advantage of separating BAN management roles from the phone is we can now take advantage of a more focused target. All applications are operating on sensors and streams of sensor data. We want a wide range of processing options, but few output options (send to a back-end service, display a small amount of info on the watch, vibrate, etc).]

[xxx New programming models. Can we automate provenance, confidentiality? Keys can be a first-class computing resource in Amulet's "OS". Data can be automatically encrypted/decrypted as it moves from sensor to app and onto the network. Some metadata (time, the person carrying the sensor, etc) can be automatically bound in with the processed data. Other types of data might be requested by the application.]

5.4 Deploying Code Safely

[xxx SHOULD WE DROP THIS SECTION?]

[xxx how to safely get code on the wearables.]

[xxx prescription-based trust/deployment model.]

6. RELATED WORK

[xxx we'll need a lot of help gathering related work – eg, on use of nearby displays, ambient display of info, other WBAN approaches, personal services, etc. We should cite the]

[xxx describe different flavors of BAN. How to organize it? Maybe those that use some form of personal server, those with dedicated base stations, and sensor centric (all of the logic is distributed in the sensors)?]

AlarmNet paper from Wood et al. [**?**]; IMD shield paper [10].

[xxx The Personal Server - Want, Pering—similar concept, only they are thinking at a different level than we are (files and multimedia) and they don't really consider security. Definitely an early and related take.]

7. SUMMARY

8. SCRAPS TO INTEGRATE

[xxx These are little ideas and scraps of text that should be considered for integration elsewhere. It's my crude way of grabbing text from email threads.]

- Regarding the IO capabilities of the TP, in baseline model I expect a small screen and a few buttons (like an LCD watch). But in future it seems feasible to include speaker and microphone, so one might include speaker recognition and speech synthesis. And, gestures based on accelerometer or gyroscope.

8.1 Ron's thoughts

The software needed to interpret sensor data in our BAN is really a split processing architecture. Some data processing often has to occur on the sensor to reduce use of wireless bandwidth/battery. Some processing is better done on the TP for efficiency to take advantage of commonalities and for security and privacy. (And some might take place on a phone or at the backend but we'll ignore that for now.) Defining standards so that intermediate sensor data can easily be processed using any suitable TP software seems difficult, if not impossible (e.g., does -this- TP software have enough internal computation precision to work with -that- high resolution sensor.) Sensor makers want to differentiate their products, and want to keep the data processing proprietary. So it makes sense to have the sensor maker write both the sensor software and the associated TP software, with the output of that chain being a standardized physiological measurement set that is easily relatable to conditions a doctor wants to monitor. Sensor makers can add some additional, optional, output parameters to distinguish their products from others (as they do already.)

[xxx The above makes sense. It's natural for the sensor vendor to write the TP software. The challenging part will be to encourage them to prepare the software's output to some standard conventions, but efforts seem to be underway in that direction. —DAVE]

In practice, the doctor wants to monitor a condition. She knows that to monitor arrhythmia all she needs to do is write a prescription for an arrhythmia sensor (if there are specific types of arrhythmias she is interested in then she can use the technical terms she already knows to specify which kind: tachycardia, bradycardia, atrial fibrillation, atrial flutter, etc.) If the doctor likes some of the optional output parameters from a particular sensor maker she might also specify a brand, but generally I've seen doctors try to avoid getting in the middle of marketing wars so they avoid specifying brands. The sensor maker creates their sensor, its software, and its TP software and asks the FDA to certify the combination as an "arrhythmia, atrial flutter" sensor. The sensor maker could also write other software to get the same hardware certified for other kinds of sensing. The patient goes to the pharmacist with the prescription for an "arryhythmia, atrial flutter" sensor.

[xxx Or, possibly, some other certification body. There exist several non-governmental groups (e.g., Continua, CCHIT) that might be suitable, from a standards point of view. —DAVE]

[xxx Continua doesn't define blood pressure or pulse oximetry standards, so some standards need to exist to define what sensing "atrial flutter" and such means before an organization like Continua can define an interface standard. Does anyone know where standards for medical sensing might be found? For newer types of sensing the FDA may have to help define standards; for those parameters that have long been measured electronically there must be standards somewhere that a group like Continua could rely on, although the mobile and continuous aspect of some sensing may be unprecedented and require new standards. So there may be several levels of standards development required. —RON]

[xxx To some extent, this is an ontology problem. Also, I expect the details vary a lot across fields (cardiology, pulminology, etc) so there are probably different organizations that might define the terms, and the measurements related to those terms. Ultimately I wouldn't be surprised if there simply /are/ no precise (standard) definitions for many interesting conditions. One might look in various textbooks and the literature, then the rest is left to expert interpretation of data. —DAVE]

The pharmacist has several products available, some are more expensive sensors that it might benefit the patient to buy because they are certified for a wider variety of types of monitoring or are modular (which could be useful in the future), others are less expensive because they are only certified to do the one kind of sensing and are not expandable. Some offer a variety of colors and surface patterns. Some may allow more interaction via the TP. The pharmacist installs the TP software for the sensor the patient chooses. [xxx Do we expect the pharmacist to install the software? or the physician? or the patient securely downloads and self-installs? It seems easy for the pharmacist to do the install, but it requires the patient to visit the pharmacy in person. Is that needed for the trust properties we desire? —DAVE]

So the doctor doesn't have to learn much new to prescribe this sensor, she basically just needs to know the technical medical name for the condition she wants to monitor. The pharmacists don't need much special training to sell medical sensors, they just have to match the technical name on the prescription to the FDA certification on the products, and learn how to install TP software. The FDA and the sensor maker (and the TP makers) are the ones who have to make sure the sensor does what it should, which is exactly the way it works now.

So this would require little change in the way things currently work, and avoids having the doctors and pharmacists needing to learn the fine details of sensor and TP software integration.

[xxx We'd want to explore it with someone who actually knows this space a bit more, lest the above turn out to be oversimplified. —DAVE]

Also, note that there is an issue here with the TP that may concern manufacturers. If they have to recertify their entire system for every TP on the market then that could be a huge cost and maintenance problem for them. Is there a way to alleviate this? If there is a monopoly on TP's that's one solution. Maybe very strict standards for TP's would be another solution (i.e., TP's are guarenteed to be interoperable, like implementations of Java...maybe that's not such a good example.) I guess sensor makers could make their own TP's and test them only with their own sensors and software (the walled garden approach). We don't have to define a solution, however it might be important that we say there are ways to prevent this being a large cost sink. I'm not sure how that might be possible if we make the TP modular (e.g., it may have a variety of radios, run on several different CPU's, have a variety of interfaces, etc.)

Interoperability is always a tough problem. Standards would help but for medical sensing complete system testing may always be a requirement.

[xxx Yes, I think this is one of the major challenges the health-device field faces. We've heard several times from both the trade literature and from people in the biz that interoperability doesn't happen because, in part, manufacturers tend to build a closed system so they can "own" the whole and test it all. To interoperate with another system, or worse, a variety of other systems, means they are likely responsible for demonstrating correct behavior with all those interacting systems. I'm not sure how we can avoid this problem - but we'll need to have something to say. —DAVE]

Acknowledgments

This research results from a research program at the Institute for Security, Technology, and Society at Dartmouth College, supported by the National Science Foundation under award numbers 0910842 and 1143548 and by the Department of Health and Human Services (SHARP program) under award number 90TR0003-01. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsors.

9. REFERENCES

[1] University of Washington. Assisted Cognition project. Online at http://www.cs.washington.edu/assistcog, visited March 2008.

[2] Sasikanth Avancha, Amit Baxi, and David Kotz. Privacy in mobile technology for personal healthcare. *ACM Computing Surveys*, July 2011. Accepted for publication, to appear in 2013, Online at http://www.cs.dartmouth.edu/~dfk/papers/avancha-survey.pdf.

[3] Ryan Aylward and Joseph A. Paradiso. A compact, high-speed, wearable sensor network for biomotion capture and interactive media. In *Proceedings of the International Workshop on Information Processing in Sensor Networks (IPSN)*, pages 380–389. ACM Press, April 2007. DOI 10.1145/1236360.1236408.

[4] Ethan M. Berke, Tanzeem Choudhury, Shahid Ali, and Mashfiqui Rabbi. Objective measurement of sociability and activity: Mobile sensing in the community. *Annals of Family Medicine*, 9(4):344–350, July 2011. DOI 10.1370/afm.1266.

[5] F. Buttussi and L. Chittaro. Smarter phones for healthier lifestyles: An adaptive fitness game. *IEEE Pervasive Computing*, 9(4):51–57, October 2010. DOI 10.1109/MPRV.2010.52.

[6] Cory Cornelius and David Kotz. Recognizing whether sensors are on the same body. In *Proceedings of the International Conference on Pervasive Computing*, Lecture Notes in Computer Science, pages 332–349. Springer, June 2011. DOI 10.1007/978-3-642-21726-5_21.

[7] Intel Research. Digital Home project. Online at http://www.intel.com/research/exploratory/digitalhome.htm, visited March 2008.

[8] Daily Strength. Dailystrength.org. Online at http://www.dailystrength.org/, visited October 2009.

[9] Alan M. Dunn, Owen S. Hofmann, Brent Waters, and Emmett Witchel. Cloaking malware with the trusted platform module. In *Proceedings of the USENIX conference on Security (USENIX Security'11)*, 2011. Online at https://db.usenix.org/events/sec11/tech/full_papers/Dunn.pdf.

[10] Shyamnath Gollakota, Haitham Hassanieh, Benjamin Ransford, Dina Katabi, and Kevin Fu. They can hear your heartbeats: non-invasive security for implantable medical devices. In *Proceedings of ACM SIGCOMM'11*, SIGCOMM '11, pages 2–13. ACM, 2011. DOI 10.1145/2018436.2018438.

[11] Predrag Klasnja and Wanda Pratt. Healthcare in the pocket: Mapping the space of mobile-phone health interventions. *Journal of Biomedical Informatics*, September 2011. DOI 10.1016/j.jbi.2011.08.017.

[12] Shrirang Mare, Jacob Sorber, Minho Shin, Cory Cornelius, and David Kotz. Hide-n-Sense: Privacy-aware secure mhealth sensing. Technical Report TR2011-702, Dept. of Computer Science, Dartmouth College, September 2011. Online at http://www.cs.dartmouth.edu/~dfk/papers/mare-hns-tr.pdf.

[13] Organized Wisdom. Organizedwisdom.com. Online at http://organizedwisdom.com, visited October 2009.

[14] Intel Research. PlaceLab project. Online at http://www.placelab.org/, visited March 2008.

[15] A. B. Raij, P. Blitz, A. Ali, S. Fisk, B. French, S. Mitra, M. Nakajima, M. Nuyen, K. Plarre, M. Rahman, S. Shah, Y. Shi, N. Stohs, M. al'Absi, E. Ertin, T. Kamarck, S. Kumar, M. Scott, D. Siewiorek, and A. Smailagic. mStress: Supporting continuous collection of objective and subjective measures of psychosocial stress on mobile devices. Technical Report CS-10-004, Department of Computer Science, University of Memphis, 2010. Online at http://sites.google.com/site/autosenseproject/publications-and-presentations/mStress-TR-CS-10-004.pdf.

[16] L. A. Saxon, D. L. Hayes, F. R. Gilliam, P. A. Heidenreich, J. Day, M. Seth, T. E. Meyer, P. W. Jones, and J. P. Boehmer. Long-term outcome after ICD and CRT implantation and influence of remote device follow-up: The ALTITUDE survival study. *Circulation*, 122(23):2359–2367, December 2010. DOI 10.1161/CIRCULATIONAHA.110.960633.

[17] University of Rochester. Smart Home project at Center for Future Health. Online at http://www.futurehealth.rochester.edu/smart_home, visited March 2008.

[18] Janani Sriram, Minho Shin, David Kotz, Anand Rajan, Manoj Sastry, and Mark Yarvis. Challenges in data quality assurance in pervasive health monitoring systems. In David Gawrock, Helmut Reimer, Ahmad-Reza Sadeghi, and Claire Vishik, editors, *Future of Trust in Computing*, pages 129–142. Vieweg+Teubner Verlag, July 2009. DOI 10.1007/978-3-8348-9324-6_14.

[19] Darren Walters, Antti Sarela, Anita Fairfull, Kylie Neighbour, Cherie Cowen, Belinda Stephens, Tom Sellwood, Bernadette Sellwood, Marie Steer, Michelle Aust, Rebecca Francis, Chi K. Lee, Sheridan Hoffman, Gavin Brealey, and Mohan Karunanithi. A mobile phone-based care model for outpatient cardiac rehabilitation: the care assessment platform (cap). *BMC Cardiovascular Disorders*, 10(1):5+, January 2010. DOI 10.1186/1471-2261-10-5.

[20] Qixin Wang, Wook Shin, Xue Liu, Zheng Zeng, Cham Oh, Bedoor K. Alshebli, Marco Caccamo, Carl A. Gunter, Elsa Gunter, Jennifer Hou, Karrie Karahalios, and Lui Sha. I-Living: An open system architecture for assisted living. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC)*, volume 5, pages

4268–4275. IEEE press, October 2006. DOI 10.1109/ICSMC.2006.384805.

[21] Fuchao Zhou, Hen-I Yang, José Álamo, Johnny Wong, and Carl Chang. Mobile personal health care system for patients with diabetes. In Yeunsook Lee, Z. Bien, Mounir Mokhtari, Jeong Kim, Mignon Park, Jongbae Kim, Heyoung Lee, and Ismail Khalil, editors, *Aging Friendly Technology for Health and Independence*, volume 6159 of *Lecture Notes in Computer Science*, chapter 12, pages 94–101. Springer, 2010. DOI 10.1007/978-3-642-13778-5_12.