Future Internet **2010**, *2*, 190-211; doi:10.3390/fi2030190

OPEN ACCESS

future internet

ISSN 1999-5903

www.mdpi.com/journal/futureinternet

Article

Applications and Security of Next-Generation, User-Centric Wireless Systems

Jerry Rick Ramstetter [1], Yaling Yang [2] and Danfeng Yao [3,⋆]

[1] Department of Computer Science, Rutgers University, Piscataway, NJ, 08854, USA;
E-Mail: rick.ramstetter@gmail.com

[2] Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA, 24061, USA;
E-Mail: yyang8@vt.edu

[3] Department of Computer Science, Virginia Tech, Blacksburg, VA, 24061, USA

⋆ Author to whom correspondence should be addressed; E-Mail: danfeng@cs.vt.edu.

Received: 6 March 2010; in revised form: 3 June 2010 / Accepted: 16 July 2010 / Published: 28 July 2010

Abstract: Pervasive wireless systems have significantly improved end-users' quality of life. As manufacturing costs decrease, communications bandwidth increases, and contextual information is made more readily available, the role of next generation wireless systems in facilitating users' daily activities will grow. Unique security and privacy issues exist in these wireless, context-aware, often decentralized systems. For example, the pervasive nature of such systems allows adversaries to launch stealthy attacks against them. In this review paper, we survey several emergent personal wireless systems and their applications. These systems include mobile social networks, active implantable medical devices, and consumer products. We explore each system's usage of contextual information and provide insight into its security vulnerabilities. Where possible, we describe existing solutions for defending against these vulnerabilities. Finally, we point out promising future research directions for improving these systems' robustness and security.

Keywords: wireless; pervasive; security; location based; context-aware; implantable medical devices

1. Introduction

The number of wireless devices for personal use (including mobile phones, wireless keyboards, and wireless networks) has increased dramatically in recent years. This change is in part due to decreased manufacturing costs and increased ease of deployment. Researchers are working on applying wireless technologies to emergent, user-centric applications (e.g. mobile social networking). These researchers believe that continued widespread adoption of wireless technologies will further improve end-users' quality of life.

We categorize wireless systems into three classes based on their communication and organizational infrastructure (or lack thereof). These categories are *centralized*, *decentralized*, and *hybrid* systems. In centralized wireless systems (e.g. wireless LANs), a base station or system administrator is responsible for managing the participating devices, including assigning security credentials and system resources to them. On the other hand, decentralized systems (e.g. peer-to-peer devices) lack pre-existing infrastructure and must organize themselves spontaneously.

We further categorize wireless systems as either *contextual* or *traditional* (e.g. non-contextual). Context-aware systems monitor their environment for changes and adapt to these changes. Context-aware systems can provide a more robust end-user experience, for example, by enabling the delivery of contextually relevant media and advertisements to end-users.

In this paper, we focus on the security and contextual aspects of pervasive, personal, often decentralized wireless systems. Decentralization creates technical challenges in the security and privacy of wireless systems due to a lack of trusted authorities therein. The personal nature of such systems dictates that successful attacks upon them may reveal highly sensitive information; this is especially the case for context-aware systems. Finally, the pervasive aspect of these systems gives attackers an ample number of devices to strike upon and to hide behind. As the adoption of user-centric wireless systems continues to grow, security vulnerabilities will create increasingly serious consequences for individuals. Part of the purpose of our review is to illustrate the importance of ensuring security in *all* user-centric wireless systems.

Our paper is organized as follows. We look at location or proximity based, user-centric wireless applications. In particular, we look at mobile social networks (section 2). We survey other types of user-centric devices, including consumer devices and implantable medical devices(section 3). We then provide a brief overview of cryptographic techniques, especially as it applies to wireless communications (section 4). As a general direction for improving location based services, we provide some discussion on the problem of location verification (section 5). In our conclusion, we discuss common trends and point out overarching directions for future work (section 6).

2. Mobile Social Networking

To begin our discussion of location-based services (LBS) for personal use, we turn to mobile social networking and the impact that contextual awareness might have on it. The phrase *mobile social networking* (MSN) is a broad category inclusive of all mobilely accessible social networking services; we include both context-aware and traditional social networking services here. In general, *context-aware services* are those that adapt according to environmental changes, including variations in physical location of use and changes in the end-user's intended application of use [1]. We consider *traditional services* to be services which do not fit the definition or intent of contextual awareness. Examples of traditional MSN services are Facebook, Myspace, and Twitter [2–4]. Though these services offer mobile access (e.g. via a WAP gateway and modern cellular phone), and can correlate end-user access with time, such services are incapable of correlating user access with intended use or physical location. System knowledge of detailed context information enables a more robust end-user experience. For example, services can recognize an end-user's repetitive use patterns, using those patterns to improve that user's socialization with other users. Unfortunately, highly available context information also allows for complex and intrusive attacks (such as violating a user's location privacy). We focus on context-aware mobile social networking (CAMSN) services in the following section.

Although service designers can apply context information to mobile social networking services in many ways, of the CAMSN services we have encountered, the primary goal has been to allow digital interaction between users concerned with the same physical location. For example, users might be able to create or edit content related to their shared physical location (e.g. reviews of the restaurant they are dining in). As another example, attendees at a social event might receive automatic notifications about persons in their vicinity with similar interests (the goal being to foster interaction between strangers).

2.1. Service Designs for Context-Aware Mobile Social Networking

We highlight some of the choices that context-aware mobile social networking service's designers make below.

Network architecture type: We find three architectural categories for CAMSN services: client-server, peer-to-peer, and hybrid. *Client-server* CAMSN services are centralized: their end-user platforms provide content by utilizing communications with fixed and authoritative application servers. *Peer-to-peer* CAMSN services are decentralized: they seek to avoid the usage of fixed servers or other authoritative entities. Peer-to-peer CAMSN services therefore need to spontaneously provide (1) localization or proximity functionalities, (2) network setup, and (3) application level functionalities. *Hybrid architecture* CAMSN services are a mixture of client-server and peer-to-peer architectures. In subsection 2.2, we describe in more details the network architecture types used by CAMSN services, and also provide examples of services using each.

Source of context and user information: A majority of CAMSN services derive application layer contextuality from either location information or proximity information. *Location information* refers to the knowledge of physical locations for all users (e.g. GPS coordinates) of a service, allowing the service to correlate and match user locations. On the other hand, *proximity information* for a user refers to knowledge of nearby users as determined via short range communications (e.g. Bluetooth). In subsection 2.3, we describe in more detail the methods with which CAMSN services can derive location or proximity information, and provide examples of services using each.

We find a further distinction among CAMSN services regarding where their user profile information is taken from. Most services require end-users to create a new, specific profile for usage on the service; these are *non-augmenting* services. *Augmenting* CAMSN services, however, allow users to augment existing, traditional social networking profiles (e.g. Facebook) with externally provided contextual information. An example of an augmenting CAMSN is WhozThat [5]. The data flow in WhozThat is bidirectional – information flows from end-user platforms to the WhozThat service (the IDs of proximate peers) and likewise flows in the opposite direction (the profile data of proximate peers). An augmenting CAMSN need not have this bidirectional data flow; CenceMe is one such example [6]. Client devices in this service push context information to their appropriate traditional social networking profiles. However, these same client devices do not receive context information from the service, they cannot answer the question "what devices are near me?".

2.2. Network Architectures for CAMSN Services

In this section, we explain in further details the network architectures used in context-aware mobile social networking services.

In client-server based CAMSN architecture, a client device might ask its application servers "what devices are near location X?" This same device may announce its location to its application servers. Most CAMSN services to date have client-server architectural design. These include FourSquare, BrightKite, Loopt, Google Latitude, and CenceMe [6–10].

CAMSN services in peer-to-peer architecture poll or discover physically neighboring devices without discovering location information for those devices. MobiSN, which defines an ad-hoc routing protocol and profile matching functionality, is one example of a completely peer-to-peer CAMSN service [11].

Hybrid architecture CAMSN services make use of some combination of client-server and peer-to-peer communications. A device on a hybrid service might discover proximity information (e.g. neighboring devices) via short-range communications, but use that information to look up user profiles on an authoritative application server. An advantage here is that the application server need not be concerned with location information. That is, an end-user device x will not ask the application server "what devices are near my location, which is L?" Rather, x would ask "what profile data is available for device y," where y is another device in the proximity of x (discovered via short-range communications). Examples of hybrid CAMSN services include WhozThat, Serendipity, and Aka-aki [5,12,13], all of which utilize both client-server (e.g. for user profile data retrieval) and peer-to-peer (e.g. for discovering proximity relations) network architectures.

2.3. Methods for Deriving Location or Proximity Information

We highlight the primary methods that context-aware mobile social networking services use to derive location or proximity information. Proximity information for a device refers to the knowledge of neighboring devices as locally discovered through short range communications.

Self reporting: CAMSN services of the self reporting type allow end-users to report their physical location to the service, for example by typing it in on their cell phone's keypad. This often cumbersome method of deriving location information relies on users to accurately (e.g. non-maliciously) report their location. We find that this method of deriving location information is rarely used, possibly due to the prevalence of GPS hardware on end-user platforms (e.g. cell phones). BrightKite is notable in that it allows users lacking a GPS enabled cell phone to self report their location [8].

Self-derived location: CAMSN services in this category use trustworthy, externally provided information to localize themselves. If the hardware platform used by such a CAMSN service is a modern cellular phone, the service might use GPS or Cellular Assisted GPS (A-GPS) to determine its location in real time (or at specified time intervals). Loopt and Google Latitude, for example, update A-GPS derived location information in real time, automatically dispersing that information to chosen friends [9,10]. FourSquare uses GPS derived location to suggest locations which users might *check in* to, thereby allowing them access to that location's content [7]. Where GPS coverage is not available, most of the A-GPS based services fall back to using cell tower triangulation to provide location information. We find that self derived location was not used in peer-to-peer CAMSN services, as gossiping GPS location data between devices associated with such a service would include neighbor discovery – effectively polling (see below).

Network-derived location: For CAMSN services in this category, the infrastructure or network is responsible for calculating a device's location and reporting this location back to the device. We did not encounter any CAMSN services that explicitly utilize network derived localization. This observation is likely because most CAMSN services to date have utilized cellular phone platforms, and such platforms generally include GPS receiver hardware. On the other hand, such platforms often use cellular-assisted GPS (A-GPS). Some implementations of A-GPS involve the wireless cellular network *pushing* additional information to receivers (e.g. cell phones) to aid in the localization process.

Peer polling and broadcast: CAMSN services utilizing polling continually look for neighbors over short range communication channels while simultaneously announcing their own presence over those channels. This method does not find location information for those peers but rather identifies which peers are nearby. Serendipity and WhozThat both utilize polling for neighbor discovery. Serendipity nodes broadcast and looks for Bluetooth IDs (BTIDs), whereas WhozThat (an augmenting CAMSN) utilizes user names from traditional social networks in place of BTIDs [5,12]. Other polling based CAMSN services are Jambo, which utilizes 802.11 WiFi rather than Bluetooth, and MobiSN [11,14]. Polling has the advantage of being inherently compatible with CAMSN services utilizing a peer-to-peer architecture, as polling does not require fixed infrastructure to work.

Hybrids of self reported location, self derived location, and proximity based mobile social networking services are possible. "Hybrid" as used here does not refer to a hybrid network architecture (see subsection 2.1), but rather to how a service derives location or proximity information. As mentioned, BrightKite is able to switch between self reporting and GPS derived location information. MobiLuck allows users to either *check in* to their locations or enable real-time GPS based updates, depending on their hardware platform and personal preferences [15]. Aka-aki uses a hybrid of peer proximity information (derived from Bluetooth) and A-GPS location information [16].

2.4. Attacks and Defenses in CAMSN services

In all types of mobile social networks, issues of security and privacy exist. Attacks on mobile social networking services (both contextual and traditional) include Sybil attacks, wormhole attacks, anonymity attacks, as well as social-engineering based phishing attacks. Furthermore, information leak and privacy issues are possible. Many of these problems can exist in a mobile social network without regard to the level of contextual awareness introduced by it. Below, we describe some of these problems in further detail and, where possible, provide mitigation strategies for them.

In a **Sybil attack**, a single entity takes on the roles of multiple entities (known as *Sybil identities*) for illicit purposes. The Sybil attack is possible in any service that utilizes reputation information assigned by service clients (e.g. peers of a device on a service), and is in many ways similar to the real-world concept of ballot stuffing [17–19].

Sybil attacks are possible in CAMSN. For example, an attacker can hijack a social networking site's highest ranked profiles list by creating many bogus accounts (or taking control of many legitimate accounts) and using those accounts to vote for a target profile. The bogus or stolen accounts need not cast their votes at the same time. As another example, an attacker using a CAMSN service can illicitly advertise a physical location (e.g. a bar or pub) by converging a number of bogus (or stolen) identities upon that location; the location will thereby appear popular to legitimate users of the CAMSN. In this case, the bogus or stolen accounts must converge on that physical location at the same time in order to make it appear popular.

Service designers and engineers can prevent a Sybil attack altogether only via careful certification of identities by a trusted authority [17], which must ensure every entity has only one identity. Zhang *et al.* designed a novel solution based on location-based cryptographic keys to prevent Sybil attacks, in particular, location-spoofing based Sybil attacks in wireless sensor networks [20]. The proposed location-based keys are generated using pairing-based identity-based cryptography by a trusted authority. Similarly, Trusted Computing Group (TCG) modules could allow for the attestation of code running on a MSN device [21], thereby helping to prevent the hijacking of legitimate identities. Power consumption of an on-board TCG module is an issue in the latter of these solutions, while the requirement of a trusted, external authority is unreasonable in the former (especially in the case of peer-to-peer systems). Methods of attesting memory contents in wireless sensor networks have been proposed [22,23]; these methods could potentially be applied to mobile social networks and the hardware thereof.

Steps a service's designers can take to limit a Sybil attack's effectiveness include limiting the privileges (or relative weight) of newly created identities and making the creation of new identities difficult. The latter can be accomplished, for example, by requiring that a CAPTCHA be solved [24]. Verifying the physical locations of identities to ensure that they move in a realistic manner has been proposed [19]. For example, the position of Sybil identities might be expected to shift locations rapidly. In the realm of CAMSN services, such location based methodologies for detecting Sybil attacks are of particular use when location information is already known – the cost of implementing them might be comparatively low (see this paper's discussion of Secure Location Verification in section 5). Lastly, a method of examining the connectivity characteristics of social graphs to prevent Sybil attacks has been proposed [25]. An overview of further solutions to the Sybil attack is found in [18].

Attacks against anonymity: CAMSN services are susceptible to attacks which leverage the fact that a user's identity is often intrinsically associated with a social networking profile and location information [26]. Such attacks attempt to violate the privacy of an end-user's location history by associating that history with a unique, application layer identifier. In peer-to-peer systems, such attacks are made possible by the exchange of unique identifiers (e.g. social networking service user names) and profile data between nodes. Malicious nodes can log these unique identifiers, allowing reconstruction of a target device's (or user's) location history.

In client-server systems, where unique identifiers are not swapped between nodes but rather from clients to application servers, direct anonymity attacks are possible by inserting malicious devices into the network. These malicious devices can continuously query an application server for the unique identifiers (or user names) of nearby devices and record the times at which such unique identifiers were seen. In both cases, attackers can use the observed identifiers to find corresponding social networking profiles. Importantly, anonymity attacks do not require a malicious node to receive information beyond what it would normally receive. That is, malicious entities can carry out such a direct anonymity attack by recording *only* a victim's unique identifier, that victim's physical locations, and the times at which the victim was seen (and correlating that information with the victim's profile).

Services can prevent or mitigate anonymity attacks via the use of encryption, though encryption is hard in the peer-to-peer case due the challenge of establishing shared keys (see section 4). A system using anonymous identifiers at the application level has been proposed [26]. This system is applicable to hybrid mobile social networking service architectures. It presumes that devices 1) utilize a short range communication technology for discovering proximity information (e.g. Bluetooth) and 2) have communication with a trusted *identity server* (e.g. via the Internet). In this system, no communications use unique identifiers. Rather, all communications use anonymous identifiers, which are assigned on a per-message basis by the trusted identity server. Only the identity server can map anonymous identifiers to specific devices, and the server will perform this mapping only after verifying that two potentially communicating devices are truly within each other's physical proximity. This system prevents malicious parties from mapping messages or profile information to a specific user.

Privacy leakage: While phishing attacks involve a malicious entity attempting to trick users, privacy leaks ostensibly occur with the user's permission. Both malicious and non-malicious entities can leverage this phenomenon. For example, a health insurance company might notice that a potential insuree's social networking profile is public, and look at that profile for information regarding the potential insuree's medical state. Though possible in any social network, the increased amount of personal information available to CAMSN services (and social networking services in general) dictates that such privacy leaks might have far greater consequences. For example, a website leveraging privacy leakage in CAMSN services has been released [27]. This website examines users' locations on CAMSN services and identifies users who are not in their home. By identifying users who are not at home, the site potentially aids malicious persons in finding empty homes to burglarize. The site *only* makes use of publicly available information; users who appear on the site have *chosen* to make their location publicly available (though, likely, without sufficient consideration). Sufficient documentation must be provided by mobile social networking services to ensure users are able to alter privacy settings. Default service privacy settings are of critical importance in battling privacy leakage, as even with documentation users might not understand how to alter privacy settings.

Network-based attacks: Mobile social networking services are susceptible to attacks on their infrastructure. Like many networked systems, mobile social networking services are subject to denial of service, eavesdropping, spoofing, and replay attacks. A *denial of service attack* consists of an attacker consuming excessive resources on a mobile social networking service's infrastructure. Twitter, for example, was brought down for days in August 2009 by a denial of service attack against its web servers [28]. An *eavesdropping attack* consists of a malicious user listening in on wireless communications, and is preventable via end-to-end encryption. A *replay attack* involves eavesdropping, recording, and replaying some critical pieces of information, such as a session authentication or other identifying message. *Spoofing attacks* occur when an attacker pretends to be an identity (or authority) which he is not. Replay attacks are often used in carrying out a spoofing attack. The link layer Address Resolution Protocol (ARP) is notoriously susceptible to spoofing attacks which allow an attacker to take control of the IP address of a legitimate network device [29].

The anonymous identifiers system proposed in [26] prevents eavesdropping, replay, and spoofing attacks. Services can use location verification (see section 5) to aid in the prevention of replay and spoofing attacks. Other methods of preventing replay attacks are summarized in [30].

A *wormhole attack* is a variant of the replay attack in which recorded data from a physical location A is replayed at a different physical location B, with the intent of making a user or device at location A appear to be at location B. Malicious entities can use wormhole attacks in mobile social networking services to create confusion amongst users or to alter network topology with the goal of maliciously intercepting (and replaying) traffic. Differing network speeds make wormhole attacks possible: in a peer-to-peer MSN, if device X sends a message m to device Y, m will follow an expected route (e.g. the fastest route) across peer devices. However, if a wormhole is introduced as the shortest path between X and Y, the message m might flow along the wormhole, thus allowing the wormhole's controller to eavesdrop on and manipulate the message m (assuming m is not encrypted). One solution to wormhole attacks involves the use of Packet Leashes, or limitations on a packet's lifetime and traveled physical distance [31]. A routing protocol which attempts to identify the effects of a persistent wormhole has been proposed [32]. Algorithms utilizing nodes dedicated to the task of monitoring network topology have been proposed [33]. Many other strategies to aid in the prevention and detection of wormhole attacks have been proposed and we refer the reader to the following literature for further reading [34,35].

3. Consumer Products and Active Implantable Medical Devices

Emergent user-centric and wireless systems consumer products may include personal media devices (e.g. an iPod paired with Bluetooth headphones), wireless computer mice, wireless keyboards, and off-the-shelf 802.11 WiFi hardware. As the cost of wireless transceivers continues to decrease, countless more devices and technologies will make use of them. The pervasive and often personal nature of these devices requires that designers pay attention to their security. We also discuss security issues associated with personal medical devices in this section.

3.1. Security and attacks in Consumer Products

We interpret the phrase *consumer-product security* in two ways. The first, which we adopt primarily, is that of *protecting users from malicious or insecure devices*. The second (and generally less common) view is that of *protecting devices themselves from consumer attacks or "hacks"* (for example, in order to access manufacturer disabled functionalities). A primary reason manufacturers intentionally disable functionalities is to enable product price tiering (e.g. low and high end products) while maintaining highly cohesive manufacturing processes: lower end devices will have excess functionalities disabled rather than removed altogether. These viewpoints of the phrase "consumer product security" are distinct but not mutually exclusive; many of the techniques proposed in dealing with the second viewpoint are directly applicable to the first. For example, the use of secure boot processes, robust code reviews, and component isolation in consumer devices has been advocated in order to prevent the introduction of unauthorized, feature-enabling new code [36]. These measures have direct end-user security benefits in that they aid in preventing the introduction of *all* new code, whether that code be user intended or malicious.

In the remainder of this section, we discuss security vulnerabilities in consumer products, describing instances of successful attacks upon those vulnerabilities where possible.

Consumer products are susceptible to attacks against anonymity, similar to CAMSN. One example of this is the Nike+iPod Sport Kit [37]. The product utilizes two paired devices, a sensor and a receiver. The sensor is placed in an end-user's shoes, while the receiver is attached to the user's pre-existing iPod. The users' footsteps are monitored by the sensor and reported to the receiver, allowing the user to monitor her exercise. The product utilizes a non-encrypted unique ID for every sensor. This allows attackers to sniff for the system's presence and, in turn, monitor the end-user's presence at various physical locations. Because the system is physically based (e.g. users wear it), attackers can easily associate overheard unique IDs with individual persons. In this case, the manufacturers of the device failed to recognize that communications between two statically paired devices (sensor and receiver) could easily be encrypted with a symmetric cryptography system.

Side channel attacks allow an attacker to gain useful information by means other than message contents; these attacks are a type of *information leak*. Examples of side channels are inter-packet timing, packet header contents, packet size, and network flow size. An attacker can observe all of these without knowledge of message contents or protocol encryption schemes. Side channel attacks are possible in any networked system including mobile social networks, vehicular ad-hoc networks, and medical devices. Certain features of consumer devices make them particularly favorable targets for side channel attacks. Specifically, side channel attacks are most easily carried out when an estimate of the type of network traffic to be eavesdropped is available.

The SlingBox Pro, a wireless consumer media device utilizing Variable Bit Rate (VBR) compression, was found to suffer from information leak problems [37]. The SlingBox Pro's goal is to encode live media signals (e.g. television or movies), apply VBR compression to the encoded media, and stream the compressed media over a network to the Internet. In testing the SlingBox Pro, researchers used an encrypted 802.11 WiFi network for communications. Researchers discovered that they did not need to examine message contents (nor break 802.11 WiFi security measures) in order to discover the movies being viewed by SlingBox Pro users. Rather, the SlingBox's use of VBR compression allowed investigators to correlate network flows with individual media items. For an observed network flow, researchers specifically correlated packet size, inter-packet timing, and packet rate with individual media items – all without knowledge of packet contents. We note that the highly specific purpose of the SlingBox Pro eased the task at hand for two reasons. First, researchers knew apriori the type of traffic they'd be concerned with (e.g. encoded and compressed movies). Secondly, the device's specific purpose ensured that researchers did not need to filter out extraneous traffic (e.g. web downloads) during their sniffing and analysis.

Obvious solutions for preventing side channels include padding and attenuating signals. *Padding* emitted signals with random noise would prevent that signal from being identified; however, issues of energy consumption remain. For example, the aforementioned attack against the SlingBox Pro could have been prevented by padding the VBR compressed media streams with random data. The problem here is that the goal of VBR compression is to reduce bandwidth usage, whereas padding media streams with random data would increase bandwidth usage. *Signal attenuation* would use metal shielding to prevent wireless signals from crossing physical boundaries. The economics of massive scale attenuation are hard to justify, as it might require wrapping whole buildings in shielding.

Device fingerprinting attacks occur when side channels are used to *fingerprint* (or uniquely identify) a wireless device. Such unique identification can be used to track a device even if the device takes proactive steps to ensure its anonymity. By simple measurement of a device's physical layer signal characteristics (such as the signal strength, angle of arrival, and time of arrival), invasive localization techniques might be able to track a wireless device's location without appropriate end-user consent, thereby violating the user's location privacy. Because such device fingerprinting is independent of MAC and IP addresses, malicious user tracking may be possible even if a device takes proactive steps (such as frequently changing its MAC and IP addresses) to ensure its anonymity. Fingerprinting attacks can be mitigated by introduction of random communication delays, use of periodic transmissions, introduction of signal attenuation, and use of randomly varying resistors in transceiver hardware (e.g. to randomly alter that signal's physical layer characteristics) [38,39].

Other common methods for protecting location privacy in the face of such device fingerprinting attacks are based on the concepts of *mixed zones* and *anonymous identifiers* [40–42]. In these methods, neighboring wireless devices sometimes enter a completely *silent* state wherein nothing is transmitted by any device. This silent state can be entered whenever a specified time period elapses or whenever a device enters a certain geographic area. At the conclusion of the silent state, all wireless devices adopt new identities (e.g. new MAC addresses and IP addresses). Since during the silent period the localization system cannot track mobile devices, it may be hard for a localization system to associate the new device identities and locations with the old (e.g. before the silent period) device identities and locations. In this way, the silent state creates a mixed zone that breaks any attempt by a localization system to continuously track a user's location, and hence protects the location privacy of users. Unfortunately, such methods have severe implications in that they limit the ability of *all* localization systems to function– even trusted ones. Similarly, the long disruptions on communication introduced by such methods may be unacceptable.

Many further attacks are possible on consumer products. These attacks include spoofing, phishing, eavesdropping, and wormholes, all of which are described in detail in subsection 2.4. Some manufacturers have proposed strategies for low cost improvements to consumer device security. For example, the usage of a low cost IPSEC proxy to encrypt communications from consumer devices has been proposed [43].

3.2. Security Trade-offs in Active Implantable Medical Device Designs

In this section, we focus on active implantable medical devices (AIMDs), a category inclusive of pacemakers, implantable cardiac defibrillators (ICDs), and implanted drug delivery systems. Such devices are becoming increasingly technologically advanced; current generation devices already support remote monitoring by health professionals via 802.11 WiFi [44]. Although external personal medical devices (e.g. glucose meters) and passive implantable medical devices (e.g. replacement joints) are likewise advancing technologically, we do not consider them. Because AIMDs can be repaired or replaced only via highly invasive surgical procedures, and because technologically advanced AIMDs might be susceptible to network attacks, interesting choices must be made and often competing goals must be balanced in designing them.

Designers of AIMDs must ensure that device security and patient privacy are maintained during attacks. However, design techniques aimed at addressing the goals of security and privacy are often at odds with other design goals. We highlight some of these trade-offs below, and, where possible, highlight some promising research in addressing them. In the following paragraphs, we use the term *security* to refer to the information security principles of confidentiality, integrity, and availability. We do not concentrate on physical safety, except in the context of AIMDs carrying out their primary, possibly life preserving functionalities.

Security *vs.* energy usage: Designers must keep the power usage of AIMDs to a minimum, especially considering the physical inaccessibility of AIMDs [45]. This places constraints on the computational power and communications abilities of AIMDs. For example, asymmetric cryptography may not be suitable for use in AIMDs due to its computational requirements.

Current-generation AIMD bearers commonly carry bracelets or other forms of identification to alert medical professionals of the AIMD's existence (e.g. in an emergency situation). Making these identifying bracelets or cards computationally powerful, and pairing them on a one-to-one basis with their AIMD, has been proposed [46]. An AIMD can offload long range communication and intensive processing onto an externally paired device. The one-to-one pairing between an AIMD and its external device allows the use of power-saving symmetric cryptography between the two. The lack of severe energy constraints on the external device allows for the use of computationally more-expensive asymmetric cryptography between it and other external devices (e.g. programmers).

Certainly other power-saving measures are possible. Medical practitioners and AIMD designers must take care to ensure that communications with an AIMD have sufficient benefit to warrant energy usage. For example, medical practitioners might allow non-critical telemetry data to queue on an AIMD rather than fetching it in near-real time, thus avoiding unnecessary AIMD energy usage.

Security *vs.* device lifetime: AIMDs cannot be replaced or physically repaired without surgery. Similarly, they cannot be updated (e.g. software patching) without a patient visit to a trusted device programmer (e.g. physician). Because of these facts, AIMDs must be designed with longevity in mind. This requirement is in contrast to the security paradigm of, for example, desktop computers, which require frequent rollouts of software updates to correct security flaws.

Protocols used to communicate with AIMDs must be designed for a very long lifespan. A single AIMD might serve a patient for 15 or more years, and this single AIMD might communicate with many different types and generations of external hardware throughout its lifespan. Each of these pieces of external hardware must support communication with the possibly obsolete AIMD; communication with a functioning AIMD cannot be hindered in the name of technological obsolescence [47].

Security *vs.* accessibility: An AIMD must be secure against an attacker on a day-to-day basis, but in an emergency situation that same device must allow for accessibility by previously unauthorized personnel (e.g. emergency room physicians or paramedics) [45]. An AIMD cannot easily differentiate between these scenarios in order to vary security settings, this due to a lack of physical access to the AIMD and aforementioned power constraints on it. Designers of an AIMD, then, might consider criticality-aware access policies infeasible (see [48] for reading on criticality-aware access policies).

Solutions for secure AIMD accessibility which are completely internal to an AIMD (that is, without a paired external device) have been proposed. One such solution proposes that AIMDs emit an audible warning during communication sessions, thereby enabling a patient to physically leave the communication range of unauthorized communicators [46]. It has been proposed that AIMDs and their programmers can mutually authenticate one another via observed physiological signals (for example, heart rate variability). These mutually observed signals would be used as the basis for a biometric encryption and authentication scheme between programmer and AIMD [49,50].

Security *vs.* functionality: As computational and medical hardware continue making strides in power efficiency, it is possible to put more functionality on-board an AIMD. Designers should analyze these functionalities to ensure they do not introduce excessive security or patient safety risks. For example, the ability to reprogram or otherwise communicate with an AIMD from a long distance is of great benefit medically, but unfortunately also allows attackers to carry out malicious activities from a long distance. Similarly, strong authentication between AIMDs and communicating devices is desired from a security standpoint, but latencies introduced by such a system may be medically unacceptable [46].

Security *vs.* data retention: Though a medical practitioner might desire an AIMD to contain a patient's entire medical history, this would allow a successful attacker access to the patient's history. Designers should evaluate the trade-offs between inclusion of data, the desire to keep that data out of an attacker's hands, and the effectiveness of device security measures.

Beyond the trade-offs of AIMD design described above, specific attacks on AIMDs are possible, including attacks against anonymity. Any communication system deployed in an AIMD must guard against denial of service attacks. Such attacks might include blocking of computational resources, depletion of energy reserves, or jamming of communications channels [45]. An AIMD utilizing a challenge-response authentication protocol might be susceptible to all three of these defects: generating a response utilizes computational resources, wireless communication utilizes significant energy reserves, and that same wireless communication further consumes communications channel bandwidth. An AIMD can mitigate these attacks by filtering communication from devices which are known, or reasonably suspected, to be malicious [51,52]. Efficiently and accurately categorizing third party devices as malicious remains a challenge.

4. Cryptography and Wireless Security

Cryptographic techniques form the first line of defense for securing wireless communications. Although this paper is not particularly focused on applied cryptography in wireless systems, we give a brief overview of the basic concepts. The three basic security requirements specified by the IEEE for wireless local area network (WLAN) environments are authentication, confidentiality, and integrity. *Authenticity* is to ensure the identity of the sender of a message. *Confidentiality* guarantees that a message is kept secret and can only be read by the intended receiver. Finally, *integrity* ensures that any tampering or distortion of a message by adversaries is identifiable. Many cryptographic protocols have been proposed to secure communications in wireless networks.

There are two main cryptographic paradigms: symmetric and asymmetric. *Symmetric cryptographic systems* (such as the Advanced Encryption Standard, AES [53]) use the same secret key for message encryption and decryption. They require the sender and receiver to agree on a shared key before communicating. Symmetric encryption is fast, relatively simple, and widely used for encrypting large data flows. *Asymmetric encryption schemes* (such as the RSA public-key encryption scheme [54]) require the receiver to have a key pair—a public key known by others and a private key known only to the receiver. The sender encrypts a message with the public key of the receiver. The receiver, upon obtaining the ciphertext, decrypts it with her private key. In public-key signature scheme, a signer uses her private key to sign a message for message integrity purpose and produces a digital signature. A verifier, upon obtaining the signature and the message, uses the signer's public key to verify the digital signature and make sure that the message has not been tampered with.

Asymmetric or public-key encryption schemes require the sender to know the correct public key of the receiver, similarly public-key signature scheme requires the verifier to know the public key of the signer. The authenticity of a public key is extremely important—using the wrong public key in decryption may result in the receiver erroneously accepting an adversary's message, possibly leading to attacks. Asymmetric cryptography is computationally more expensive than symmetric ones, making its usage in embedded or power constrained environments difficult.

In an environment where there are central trust authorities, services can use digital certificates to bind a public key to an identity and verify that a public key belongs to an individual. For example, VeriSign is one of the Internet's major certificate authorities. The digital certificates used in SSL and HTTPS utilize *public key cryptographic schemes*. Specifically, they use a public key digital signature scheme for initial trust establishment. Once trust is established between the client and the server (e.g. a Web server and a client's browser in the HTTPS case), a shared key is negotiated and used for encrypting subsequent communications.

In decentralized wireless networks (such as wireless ad-hoc networks) the infrastructure is fluidic—there is no pre-defined trust authority and the rate at which nodes join and leave the network may be high. The task of establishing trust between nodes is therefore difficult. For example, in Mobile Ad-Hoc Networks (MANETs) each node voluntarily forwards data to other nodes, but the determination of which nodes forward what data is made dynamically based on network connectivity. The mobile nature of MANETs leads to frequent network topology changes. Securing the communication in such a decentralized wireless system is challenging and may require advanced cryptographic tools such as threshold cryptography [55,56]. In *threshold cryptosystems*, a secret (e.g. certificate or other valuable data) is split into shares, and any k nodes can combine their shares to recover the master secret, which may be used for signing certificates or other critical operations. The advantage of using threshold cryptosystems in MANET is that no single node needs to take the responsibility of storing the master secret and being available and reachable all the time. The distribution of secrets among a number of nodes effectively *decentralized* the trust management without compromising the security. We refer readers to the literature for more discussion on applying threshold cryptography to securing decentralized wireless systems [57–60],

5. Increasing Robustness with Location Verification and Computation

The use of localization services in pervasive devices is increasing. Many technologies have been built upon an underlying assumption of available location information. These applications may have a need to securely verify the locations reported by pervasive devices, as such pervasive devices may be malicious or simply incorrect in reporting their location. Inaccuracies, whether malicious or accidental, are possible without regard to who or what is performing the localization. When a device is responsible for calculating its own location using trusted, externally provided information (as in *self-derived* localization), the device may calculate and report a malicious or incorrect location. Likewise, when a device has its location calculated by trusted infrastructure, and this location is then reported back to the device (as in *network-derived* localization), the device might maliciously alter information used by the network in calculating its location. For example, a malicious device can alter its signal characteristics.

The goal of *location verification* is to verify that location information reported by potentially malicious devices is accurate to some degree; service designers should view location verification as an additional, security-enhancing step in the process of localization. Location verification is a distinct problem from *secure localization*; the latter's goal is to allow devices to localize themselves in a hostile environment [61]. We do not consider the problem of secure localization in this paper.

5.1. Distance Ranging

Distance ranging is used to enable location verification via distance bounding, *i.e.*, distance ranging is the foundation of the location verification. In a distance bounding location verification system, one device (a verifier) asserts that another device (a prover) is within a preset distance from it. Below, we provide some insight into the methodologies used to carry out distance ranging.

Time of Flight (TOF) ranging methods entail precise measurement of the travel time t of data between two devices, A and B [62]. A ranging system can calculate the maximum distance from A to B by knowing the communication medium's transmission speed and an estimate h of hardware induced delays. In the case of radio transmissions, where the transmission speed is c (the speed of light), the maximum distance would be $c(t - h)/2$.

Time Difference of Arrival (TDOA) ranging systems use multiple TOF verifiers to triangulate the location of a verifier in a process known as *multilateration* [63,64]. Ranging via TOF does not require bidirectional communication; a unidirectional data transmission is sufficient for the receiver (e.g. verifier) to calculate its distance from the sender (e.g. prover) provided packets are appended with their send time and clocks are synchronized.

Malicious parties can abuse TOF ranging in both the unidirectional and bidirectional cases. In the unidirectional message case, the receiver cannot guarantee that the send time appended to the received packet is accurate. In the bidirectional message case (a message from A to B, then from B to A), A cannot guarantee that B did not delay its return message unnecessarily. As B can only increase the range estimates by delaying the return message, verification systems can still function [65,66]. Even operating under the assumption that both parties are trusted, TOF ranging requires highly precise timing measurements, and is thus subject to large discretization error [67]. A clock accurate to millisecond precision will allow for up to 300 kilometers of error in distance calculated during a round trip transmission (assuming radio transmissions are used). It is possible, though costly, to use alternative communication mediums with a slower propagation speed in order to allow for lesser clock accuracy. The use of ultrasound in one direction of communication lowers the error term by about the order of 10^6. Major problems with the usage of such slower mediums include their high cost and their susceptibility to *wormhole attacks*.

Received Signal Strength (RSS) ranging utilizes the fact that radio or ultrasound signals decay with distance; receipt of a weak signal indicates either a weak transmitter or a large distance from the sender. Knowing the transmitter's output power in advance allows for ranging by mapping the received signal power through a function to distance. Many technologies exist to utilize existing 802.11 WiFi or Bluetooth infrastructure to carry out RSS ranging. Unfortunately, a malicious node can defeat RSS ranging by varying its transmit power in different directions (e.g. by using directional antennas), or by changing its transmit power with time. In addition, off-the-shelf, commodity hardware often does not make available an accurate measure of received signals' power. 802.11 WiFi device drivers are not required to provide access to this information through device drivers. Rather, the drivers must specify a RSSI value, which need not linearly map to received signal power in decibels [68].

5.2. *Methods for Location Verification*

One of the earliest solutions to the problem of secure location verification is *distance bounding*, which attempts to prove that a pervasive device, a *prover*, is within a bounded distance of a trusted *verifier* [65,66]. The physical space in which a verifier will attest to a prover's location is known as that verifier's *Region of Acceptance*, or ROA. The distance bounding that a verifier performs on a prover involves distance ranging. The ability of a verifier to perform its job is based on the assumption that either 1) the prover has only limited capability to distort the distance ranging process (e.g. the prover can increase but not decrease his calculated range from a verifier) or 2) the prover does not know the location of verifiers.

Probabilistic methods for location verification have been proposed [69,70]. These methods utilize the fact that it is possible to probabilistically relate the hop count (or number of devices through which a message passes) between nodes of a randomly dispersed Wireless Sensor Network (WSN) to Euclidean distance [71]. Namely, they work by bounding the Euclidean distance between two nodes according to the number of network hops between them. Unfortunately, these methods impose sufficiency limitations on network density; malicious or malfunctioning nodes are more difficult to detect in sparse networks.

Attacks against such probabilistic methods of location verification are possible. As the methods rely on reputation information communicated between peers, an attacker can use a variant of the Sybil attack to disrepute a victim node and cause blacklisting of that node. Similarly, a malicious node can alter the per-packet hop count information on which probabilistic location verification methods are dependent. Methods of securing verification systems against these and other attacks have been proposed. Inferring hop count from packet length might help in securing networks against nodes which maliciously alter the hop count of packets [69].

Two algorithms for secure location verification have been proposed in [72]. These algorithms attempt to verify that estimations of the same parameters (e.g. node location) are sufficiently close. The first algorithm looks for inconsistencies in four derived matrices (representative of network layout information), while the second is an iterative algorithm which utilizes an indicator value based on node neighbor consistency.

Neighboring sensors may be utilized for location verification. A method of position verification tailored to Vehicular Ad Hoc networks (VANETs) has been proposed [73]. This method utilizes a weighting of multiple *sensors*. Importantly, in the work's context, sensors refer to parameters that are used to heuristically determine whether or not a node is reporting the wrong location. As an example of the work's defined sensors, the *acceptance range* sensor utilizes the fact that VANET transceiver hardware typically has a well defined range; communications received from nodes known to be outside this range are flagged as likely to be lying about their position. As another example, the *proactive exchange of neighbor tables* sensor communicates neighboring node information across nodes, using it to find and flag inconsistencies in the reported network topology. Another location-based neighborhood authentication scheme was proposed by Zhang *et al.* that takes advantage of a group of special mobile anchors [74]. The anchors may be mobile robots and can perform coordinated group movement for collecting and measuring sensor data for location verification.

5.3. Robust Location Computation for Attack Source Positioning

This section concerns methods for locating attack sources and attackers. With the pervasive deployment of wireless systems, it is quite easy for an attacker to launch network attacks from a wireless terminal against remote critical network infrastructure—including national, financial, energy, transportation and military systems. Traditional trace-back techniques for wired networks can only identify the wireless network access point used by an attacker. Such rough identification is far from sufficient in wireless networks, as the attack source can be any node in the coverage range of the access point. The highly mobile, anonymous, and stealthy nature of wireless communication demands that finding the true location of an attacker be done using wireless localization schemes.

Traditional localization methods [75,76] rely on passive observation of the attacker's signal characteristics, such as its connectivity with neighboring access points, signal strength, *etc.*, to make a position estimation. An intelligent attacker equipped with advanced radio technologies, like directional antennas and software defined radios (SDRs), can change its beam (or directional radio) direction and radio parameters to distort these signal features as well as significantly limit the measurements of its signal. In this way, it becomes very difficult for a localization system to compute the accurate location of an attacker and hence such an attacker can hide his/her position with great ease and anonymity.

A method for enabling the robust location computation of an intelligent attack source is proposed in [77]. In this method, the network proactively coordinates the wireless access points around the attacker to lure the attacker to change its beamforming (or directional signal filtering) direction. In this way, the attacker unintentionally allows more access points to measure its signal features, such as angle of arrival, Received Signal Strength Indicator (RSSI), and connectivity to its neighboring nodes. With enough measurement of network topology and wireless transmission features, the attacker can be correctly positioned.

6. Conclusions and Open Problems

We have described the challenges faced in securing the communication and services of user-centric wireless systems. Such systems are often decentralized. On one hand, this decentralization enables users to conveniently share, access, and control unprecedented amounts of information, resources, and services. On the other hand, the broadcast and ad-hoc nature of wireless communication creates security and privacy challenges. We started our discussion by looking at location or proximity based, user-centric wireless applications. In particular, we explored the security ramifications of decentralization and contextualization on mobile social networks. We discussed location verification techniques as a method of improving these and other location based services. We looked at security issues that must be dealt with when using wireless technologies in consumer products and active implantable medical devices.

At present, many methods for protecting location privacy against a malicious localization or tracking system depend on clients and networks co-operating in a security protocol. Network trust cannot always be established, however (e.g. at a public access point). It will be important to develop solutions that allow users to control their location privacy as well as their own communication history without establishing network trust. Furthermore, current methods for protecting location privacy might make the localization process difficult for all localizers—even trusted ones. Contextually supporting pervasive devices with location information while simultaneously preserving location privacy remains an issue.

Preventing information leaks via side channels remains an issue. Many wireless devices are susceptible to side channel attacks, for example by fingerprinting of a wireless transceiver's physical layer characteristics. Current methods of preventing against information leaks are either expensive or obtrusive. As wireless technologies become more pervasive, more opportunities for observing leaked information will be made available to attackers. It will be necessary to prevent information leakage at all communication layers, including the hardware level.

Many current cryptographic solutions (e.g. public key cryptosystems and threshold cryptography) consume a great deal of energy, and as such are unsuitable for resource constrained devices. Although a device's resource constraints can often be worked-around (e.g. via the use of a paired, non-resource-constrained device), security measures completely internal to that device would have significant benefits, including lower overall energy usage and fewer system components.

Acknowledgment

This work has been supported in part by NSF grant CNS-0831186.

References

1. Schilit, B.; Adams, N.; Want, R. Context-Aware Computing Applications. In Proceedings of WMCSA '94: The 1994 First Workshop on Mobile Computing Systems and Applications, 8–9 December 1994, Santa Cruz, CA, USA; IEEE Computer Society: Washington, DC, USA, 1994; pp. 85–90.
2. Facebook. Available online: http://facebook.com (accessed on 22 January 2010).
3. MySpace. Available online: http://myspace.com (accessed on 19 January 2010).
4. Twitter. Available online: http://twitter.com. (accessed on 19 January 2010).

5. Beach, A.; Gartrell, M.; Akkala, S.; Elston, J.; Kelley, J.; Nishimoto, K.; Ray, B.; Razgulin, S.; Sundaresan, K.; Surendar, B.; Terada, M.; Han, R. WhozThat? evolving an ecosystem for context-aware mobile social networks. *IEEE Network* **2008**, *22*, 50–55.
6. CenceMe. Available online: http://cenceme.org (accessed on 22 January 2010).
7. Foursquare: Learn More. Available online: http://foursquare.com/learn_more (accessed on 22 January 2010).
8. BrightKite: Learn More. Available online: http://brightkite.com/learn_more (accessed on 22 January 2010).
9. About Loopt. Available online: http://loopt.com/about (accessed on 22 January 2010).
10. Google Latitude. Available online: http://latitude.google.com (accessed on 22 January 2010).
11. Li, J.; Khan, S.U. MobiSN: Semantics-based Mobile *ad hoc* Social Network Framework. In Proceedings of 52nd IEEE Global Communications Conference (Globecom), Honolulu, HI, USA, December 2009.
12. Eagle, N.; Pentland, A. Social Serendipity: Mobilizing Social Software. *IEEE Pervasive Comput.* **2005**, *4*, 28–34.
13. Aka-Aki. Available online: http://aka-aki.com (accessed on 22 January 2010).
14. Jambo. Available online: http://jambo.net (accessed on 22 January 2010).
15. Mobiluck. Available online: http://mobiluck.com (accessed on 22 January 2010).
16. Aka-Aki Blog. Available online: http://blog.aka-aki.com/?p=237 (accessed on 22 January 2010).
17. Douceur, J.R. The Sybil Attack. In *IPTPS '01: Revised Papers from the First International Workshop on Peer-to-Peer Systems*; Springer-Verlag: London, UK, 2002; pp. 251–260.
18. Levine, B.N.; Shields, C.; Margolin, N.B. *A Survey of Solutions to the Sybil Attack*; Techical Report 2006-052; University of Massachusetts Amherst: Amherst, MA, USA, 2006.
19. Newsome, J.; Shi, E.; Song, D.; Perrig, A. The sybil attack in sensor networks: analysis & defenses. In Proceedings of IPSN '04: The 3rd International Symposium on Information Processing in Sensor Networks, Berkeley, CA, USA, 26–27 April 2004; ACM: New York, NY, USA, 2004; pp. 259–268.
20. Zhang, Y.; Liu, W.; Lou, W.; Fang, Y. Location-based compromise-tolerant security mechanisms for wireless sensor networks. *IEEE J. Sel. Area. Commun.* **2006**, *24*, 247–260.
21. Trusted Computing Group. Available online: http://trustedcomputinggroup.net (accessed on 22 January 2010).
22. AbuHmed, T.; Nyamaa, N.; Nyang, D. Software-Based Remote Code Attestation in Wireless Sensor Network. In Proceedings of 52nd IEEE Global Communications Conference (Globecom), Honolulu, HI, USA, December 2009.
23. Shaneck, M.; Mahadevan, K.; Kher, V.; Kim, Y. Remote software-based attestation for wireless sensors. In Proceedings of The 2nd European Workshop on Security and Privacy in *ad hoc* and Sensor Networks, Visegrad, Hungary 13–14, July 2005.
24. von Ahn, L.; Maurer, B.; McMillen, C.; Abraham, D.; Blum, M. reCAPTCHA: Human-Based Character Recognition via Web Security Measures. *Science* **2008**, *321*, 1465–1468.
25. Yu, H.; Kaminsky, M.; Gibbons, P.B.; Flaxman, A. SybilGuard: defending against sybil attacks via social networks. *SIGCOMM Comput. Commun. Rev.* **2006**, *36*, 267–278.

26. Beach, A.; Gartrell, M.; Han, R. Solutions to Security and Privacy Issues in Mobile Social Networking. In Proceedings of 2009 International Conference on Computational Science and Engineering (CSE), Vancouver, Canada, 29–31 August 2009; IEEE: Los Alamitos, CA, USA, 2009; Volume 4, pp. 1036–1042.
27. Please Rob Me. Available online: http://pleaserobme.com (accessed on 18 February 2010).
28. Twitter: ongoing DoS. Available online: http://status.twitter.com/post/157191978/ongoing-denial-of-service-attack (accessed on 19 January 2010).
29. Whalen, S. An Introduction to ARP Spoofing. Available online: http://packetstorm.securify.com/papers/protocols/intro_to_arp_spoofing.pdf (accessed on 19 January 2010).
30. Malladi, S.; Alves-Foss, J.; Heckendorn, R.B. On Preventing Replay Attacks on Security Protocols. In Proceedings of the International Conference on Security and Management, Las Vegas, NV, USA, June 2002; CSREA Press: Las Vegas, NV, USA, 2002; pp. 77–83.
31. Moss, C.; Evans, D.W. Securing wireless communication against wormhole attacks. In Proceedings of EHAC'05: The 4th WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communications, Salzburg, Austria, 13–15 February 2005; World Scientific and Engineering Academy and Society: Stevens Point, WI, USA, 2005; pp. 1–6.
32. Harbin, J.; Mitchell, P.; Pearce, D. Wireless sensor network wormhole avoidance using disturbance-based routing schemes. In Proceedings of the IEEE 6th International Symposium on Wireless Communication Systems (ISWCS), Siena, Italy, 7–10 September 2009; pp. 76–80.
33. Triki, B.; Rekhis, S.; Boudriga, N. Digital Investigation of Wormhole Attacks in Wireless Sensor Networks. In Proceedings of NCA '09: The 2009 Eighth IEEE International Symposium on Network Computing and Applications, Cambridge, MA, USA, 9–11 July 2009; IEEE Computer Society: Washington, DC, USA, 2009; pp. 179–186.
34. Xu, Y.; Chen, G.; Ford, J.; Makedon, F. Detecting Wormhole Attacks in Wireless Sensor Networks. In *Critical Infrastructure Protection*; Springer: New York, NY, USA, 2007.
35. Poovendran, R.; Lazos, L. A graph theoretic framework for preventing the wormhole attack in wireless *ad hoc* networks. *Wireless Networking* **2007**, *13*, 27–59.
36. Kan, T.; Kerins, T.; Kursawe, K. Security in Next Generation Consumer Electronic Devices. In Proceedings of Securing Electronic Business Processes–Highlights of the Information Security Solutions Europe 2006 Conference, Rome, Italy, 10–12 October 2006; Vieweg: Wiesbaden, Germany, 2006; pp. 45–53.
37. Saponas, T.S.; Lester, J.; Hartung, C.; Agarwal, S.; Kohno, T. Devices that tell on you: privacy trends in consumer ubiquitous computing. In Proceedings of SS '07: 16th USENIX Security Symposium on USENIX Security Symposium, Boston, MA, USA, 6–10 August 2007; USENIX Association: Berkeley, CA, USA, 2007; pp. 1–16.
38. Dilparic, L.; Arvind, D.K. Design and Evaluation of a Network-Based Asynchronous Architecture for Cryptographic Devices. In Proceedings of ASAP '04: The 15th IEEE International Conference on Application-Specific Systems, Architectures and Processors, Galveston, TX, USA, 27–29 September 2004; IEEE Computer Society: Washington, DC, USA, 2004; pp. 191–201.
39. Srinivasan, V.; Stankovic, J.; Whitehouse, K. Protecting your daily in-home activity information from a wireless snooping attack. In Proceedings of UbiComp '08: The 10th international

conference on Ubiquitous computing, Seoul, South Korea, 21–24 September 2008; ACM: New York, NY, USA, 2008; pp. 202–211.

40. Jiang, T.; Wang, H.J.; Hu, Y.C. Preserving location privacy in wireless LANs. In Proceedings of MobiSys '07: The 5th international conference on Mobile systems, applications and services, San Juan, Puerto Rico, 11–14 June 2007; ACM Press: New York, NY, USA, 2007; pp. 246–257.
41. Grlach, A.; Heinemann, A.; Terpstra, W.W. Survey on Location Privacy in Pervasive Computing. In *Privacy, Security and Trust within the Context of Pervasive Computing, the Kluwer International Series in Engineering and Computer Science*; Kluwer (Springer) Academic Publishers: New York, NY, USA, 2004; pp. 23–34.
42. Beresford, A.R.; Stajano, F. Location Privacy in Pervasive Computing. *IEEE Pervasive Comput.* **2003**, *2*, 46–55.
43. Karasawa, K.; Kira, Y.; Tsuchiya, Y.; Yamada, K.; Takahashi, K. A Detachable IPsec Device for Secure Consumer Communication Platform. In Proceedings of IEEE Consumer Communications and Networking Conference Poster Session, Las Vegas, NV, USA, 3–6 January 2005.
44. St. Jude's WiFi Pacemaker Wins Approval. Available online: http://dotmed.com/news/story/9878 (accessed on 28 January 2010).
45. Halperin, D.; Heydt-Benjamin, T.S.; Fu, K.; Kohno, T.; Maisel, W.H. Security and Privacy for Implantable Medical Devices. *IEEE Pervasive Comput.* **2008**, *7*, 30–39.
46. Denning, T.; Fu, K.; Kohno, T. Absence makes the heart grow fonder: new directions for implantable medical device security. In Proceedings of HOTSEC '08: The 3rd Conference on Hot Topics in Security, Boston, MA, USA, 28 July–1 August 2008; USENIX Association: Berkeley, CA, USA, 2008; pp. 1–7.
47. Drew, T.; Gini, M. Implantable medical devices as agents and part of multiagent systems. In Proceedings of AAMAS '06: The Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, Hakodate, Japan, 8–12 May 2006; ACM: New York, NY, USA, 2006; pp. 1534–1541.
48. Gupta, S.K.S.; Mukherjee, T.; Venkatasubramanian, K. Criticality Aware Access Control Model for Pervasive Applications. In Proceedings of PERCOM '06: The Fourth Annual IEEE International Conference on Pervasive Computing and Communications, Pisa, Italy, 13–17 March 2006; IEEE Computer Society: Washington, DC, USA, 2006; pp. 251–257.
49. Zhang, Y.T.; Bao, S.D.; Shen, L.F. Physiological Signal Based Entity Authentication for Body Area Sensor Networks and Mobile Healthcare Systems. In Proceedings of 27th IEEE Conference on Engineering in Medicine and Biology, Shanghai, China, 1–4 September 2005; pp. 2455–2458.
50. Cherukuri, S.; Venkatasubramanian, K.; Gupta, S. Biosec: A biometric based approach for securing communication in wireless networks of biosensors implanted in the human body. In Proceedings of International Conference on Parallel Processing Workshop, Kaohsiung, Taiwan, 6–9 October 2003; pp. 432–439.
51. Mahimkar, A.; Shmatikov, V. Game-Based Analysis of Denial-of-Service Prevention Protocols. In Proceedings of CSFW '05: The 18th IEEE Workshop on Computer Security Foundations, Aix-en-Provence, France, 20–22 June 2005; IEEE Computer Society: Washington, DC, USA, 2005; pp. 287–301.

52. Andersen, D.G. Mayday: distributed filtering for internet services. In Proceedings of USITS '03: The 4th conference on USENIX Symposium on Internet Technologies and Systems, Seattle, WA, USA, 26–28 March 2003; USENIX Association: Berkeley, CA, USA, 2003; pp. 3–3.
53. Barker, W.C. *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*; NIST Special Publication 800-67; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2008.
54. Rivest, R.; Shamir, A.; Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Commun. ACM* **1978**, *21*, 120–126.
55. Desmedt, Y.G.; Frankel, Y. Threshold cryptosystems. In Proceedings of CRYPTO '89: The 9th Annual International Cryptology Conference, Santa Barbara, CA, USA, 20–24 August 1989; Springer-Verlag: New York, NY, USA, 1989; pp. 307–315.
56. Gemmel, P. An introduction to threshold cryptography. *CryptoBytes* **1997**, *2*, 7–12.
57. Feng, Y.; Liu, Z.; Li, J. Securing Membership Control in Mobile *ad hoc* Networks. In Proceedings of ICIT '06: The 9th International Conference on Information Technology, Mumbai, India, 18–21 December 2006; IEEE Computer Society: Washington, DC, USA, 2006; pp. 160–163.
58. Khalili, A.; Katz, J.; Arbaugh, W.A. Toward Secure Key Distribution in Truly Ad-Hoc Networks. In Proceedings of SAINT-W '03: The 2003 Symposium on Applications and the Internet Workshops, Orlando, Florida, USA, 27–31 January 2003; IEEE Computer Society: Washington, DC, USA, 2003; p. 342.
59. Kong, J.; Zerfos, P.; Luo, H.; Lu, S.; Zhang, L. Providing Robust and Ubiquitous Security Support for Mobile *ad hoc* Networks. In Proceedings of ICNP '01: The Ninth International Conference on Network Protocols, Riverside, CA, USA, 11–14 November 2001; IEEE Computer Society: Washington, DC, USA, 2001; p. 251.
60. Luo, H.; Lu, S. *Ubiquitous and Robust Authentication Services for ad hoc Wireless Networks*; Technical Report UCLA-CSD-TR-200030; University of California, Los Angeles: Los Angeles, CA, USA, 2000.
61. Lazos, L.; Poovendran, R. SeRLoc: Robust localization for wireless sensor networks. *ACM TOSN* **2005**, *1*, 73–100.
62. Lanzisera, S.; Lin, D.T.; Pister, K.S.J. RF time of flight ranging for wireless sensor network localization. Presented at Workshop on Intelligent Solutions in Embedded Systems (WISES), Vienna, Austria, 30 June 2006; pp. 1–12.
63. Chiang, J.T.; Haas, J.J.; Hu, Y.C. Secure and precise location verification using distance bounding and simultaneous multilateration. In Proceedings of WiSec '09: The Second ACM Conference on Wireless Network Security, Zurich, Switzerland, 16–18 March 2009; ACM: New York, NY, USA, 2009; pp. 181–192.
64. Capkun, S.; Hubaux, J.P. Secure positioning of wireless devices with application to sensor networks. In Proceedings of INFOCOM 2005: 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Miami, FL, USA, 13–17 March 2005; Volume 3, pp. 1917–1928.
65. Sastry, N.; Shankar, U.; Wagner, D. Secure Verification of Location Claims. Presented at ACM Workshop on Wireless Security (WiSe 2003), San Diego, CA, USA, 19 September 2003; pp. 1–10.

66. Brands, S.; Chaum, D. Distance-bounding protocols. In Proceedings of EUROCRYPT '93: Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology, Lofthus, Norway, 23–27 May 1993; Springer-Verlag: Secaucus, NJ, USA, 1994; pp. 344–359.
67. Kolodziej, K.W.; Hjelm, J. *Local Positioning Systems: LBS Applications and Services*; CRC Press: Boca Raton, FL, USA, 2006.
68. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*; IEEE 802.11-2007; IEEE Standards Association: Piscataway, NJ, USA, 2007.
69. Ekici, E.; Vural, S.; McNair, J.; Al-Abri, D. Secure probabilistic location verification in randomly deployed wireless sensor networks. *Ad Hoc Networks* **2008**, *6*, 195–209.
70. Liu, Y.; Zhou, H.; Zhao, B. Secure Location Verification Using Hop-Distance Relationship in Wireless Sensor Networks. In Proceedings of APSCC '07: The 2nd IEEE Asia-Pacific Service Computing Conference, Tsukuba Science City, Japan, 12–14 December 2007; IEEE Computer Society: Washington, DC, USA, 2007; pp. 62–68.
71. Vural, S.; Ekici, E. Analysis of hop-distance relationship in spatially random sensor networks. In Proceedings of MobiHoc '05: The 6th ACM International Symposium on Mobile *ad hoc* Networking and Computing, Urbana-Champaign, IL, USA; ACM: New York, NY, USA, 2005; pp. 320–331.
72. Wei, Y.; Yu, Z.; Guan, Y. Location Verification Algorithms for Wireless Sensor Networks. In Proceedings of the 27th International Conference on Distributed Computing Systems (ICDCS), Toronto, Ontario, Canada, 25–29 June 2007; pp. 70–70.
73. Leinmuller, T.; Schoch, E.; Kargl, F. Position Verification Approaches for Vehicular *ad hoc* Networks. *IEEE Wirel. Commun.* **2006**, *13*, 16–21.
74. Zhang, Y.; Liu, W.; Fang, Y.; Wu, D. Secure localization and authentication in ultra-wideband sensor networks. *IEEE J. Sel. Area. Commun.* **2006**, *24*, 829–835.
75. Hightower, J.; Borriello, G. *A Survey and Taxonomy of Location Sensing Systems for Ubiquitous Computing*; UW CSE 01-08-03; University of Washington, Seattle, WA, USA, 2001.
76. Srinivasan, A.; Wu, J. A Survey on Secure Localization in Wireless Sensor Networks. In *Encyclopedia of Wireless and Mobile Communications*; CRC Press, Taylor and Francis Group: Boca Raton, FL, USA, 2008.
77. Han, C.; Zhan, S.; Yang, Y. Proactive attacker localization in wireless LAN. *SIGCOMM Comput. Commun. Rev.* **2009**, *39*, 27–33.