Enabling Users to Self-manage Networks: Collaborative Anomaly Detection in Wireless Personal Area Networks

Zheng Dong

School of Informatics and Computing, Indiana University, Bloomington, IN, USA
zhdong@indiana.edu

Abstract. Personal area networks such as home or small office LANs are usually more vulnerable to cyber-attacks than those with dedicated support staff and the ability to invest consistently in security defenses. In this paper I propose leveraging physical characteristics of these personal area networks in order to enable non-technical individuals to secure their networks or at least be aware that their devices have been compromised. This proposal leverages records of location for mobile devices, proximity authentication, and individual homophily. In this work, I summarize previous studies on securing personal networks, proximity authentication, and software attestation. I then present a preliminary design for the detection of and recovery from infection for personal area networks. Limitations and future work are also discussed.

1 Introduction

With the improvement in affordability of many electronic devices, small-scale networks are commonly constructed in home and small office environments. The term "personal area network" (or "PAN") usually refers to a local, connected group of personal devices. These devices may include laptop computers, personal digital assistants (PDA), palmtops, and cell phones [1]. The boundary of PAN is the area physically covered by the wireless network and/or the central server. Previous network members (e.g. laptops and phones) may leave and re-enter the network multiple times.

Unlike larger networks that are dominated by wired connections, devices in personal area networks are normally connected by wireless protocols, such as Wi-Fi (802.11) or Bluetooth (802.15). It has been documented that wireless networks tend to be more vulnerable than wired networks [2].

While significant research has been performed on inter-PAN network security, little attention has been paid to security issues inside a personal area network. PAN environments are unique for two reasons.

First, most personal area networks have clear physical boundaries and basic access controls. For instance, all participating personal devices are at some time in the home, so that only the residents (and possibly a few guests) have physical access to the devices.

Second, the ownership of personal area networks (including all devices) is unitary. As a consequence, the device owner has an incentive to protect the security of the

J. Blythe, S. Dietrich, and L.J. Camp (Eds.): FC 2012 Workshops, LNCS 7398, pp. 53–67, 2012.

network. At the same time, the owner is unlikely to be skilled in computer networking. Therefore techniques designed for personal area networks need to be highly automated with minimum human interference embedded in the interactions, and should fully leverage the geographical information inherently provided by a personal area network.

The purpose of this work is to design a security protocol specifically for personal area networks. This protocol incorporates proximity authentication, collaborative rating, and software attestation, but it is not a simple combination of the above techniques. Protocol phases are carefully designed and adjusted to meet the security needs of personal area network devices. This proposal builds on physical location, particularly co-location and proximity authentication of the devices. Following proximity authentication, the proposed design also uses Bluetooth, leveraging the inherent distance limitations of Bluetooth. Other proximity authentication methods are equally applicable.

The rest of the paper is organized as follows. Section 2 introduces related work on personal network, proximity authentication, and software attestation. Section 3 defines the threat model, enumerating the threats which the proposal is designed to mitigate. Section 4 discusses the assumptions underlying the protocol design. Section 5 provides an overview of the components of the design, including participants. Section 6 provides details of the proposed protocol with an example consisting of three devices. Section 7 summarizes the findings, and concludes the paper.

2 Related Work

2.1 Personal Networks

As electronic devices become even more affordable, it is common for homes to contain an increasing number and diversity of digital devices with a small-scale network shared amongst them. Bisdikian et al. [1] introduced the notion of "wireless personal area network (WPAN)", which includes various types of personal wearable or handheld devices such as laptop computers, personal digital assistants (PDA), palmtops, cell phones, etc. These authors also pointed out that WPAN differs from traditional wireless local networks (WLAN) in network size, implementation cost, usability, and power consumption. The definition of "personal area network" assumes that all networked devices are within a short distance, typically within 10 meters. Furthermore, IEEE 802.15 [3] defines the characteristics of WPAN.

To connect personal area networks (PANs) that are geographically distributed, a personal network (PN) can be established [4]. The definition of PN relies on the idea of pervasive computing. The design of PN, however, is not merely an extension of PAN. Mechanisms such as addressing, routing, and authentication need to be implemented. Extending PN, Hoebeke et al. proposed a "personal network federation" (or "PN-F"), which enables device linkage between different personal networks [5]. PN-F addresses secure communication needs within a common interest group, such as family members, classmates, and colleagues. In addition to the proposed scheme, the author also discussed several designed challenges, such as membership management, application support, and system maintenance.

Network security and user privacy concerns are also increasing with the proliferation of personal networks. These concerns generally focus on untrusted inter-PAN web traffic. Jacobsson et al. proposed a secure PN mechanism that ensures anonymity by encryption and MAC or IP address change after certain intervals [6]. Social activities, such as device lending or sharing, were also considered. Patrikakis et al. analyzed typical threats in personal networks and introduced a trust model over personal networks [7,8]. In their design information needed for authentication was treated differently than sensitive data like user preference. A central server was established in this scheme for device registration and group key distribution. In addition, networked device status needs to be reported so that malicious users and devices can be detected. This work is different from their design for a much smaller network range, and therefore different assumptions and techniques are proposed. For example, since all networking devices are within a certain physical range, I do not consider inter-PAN network traffic, which requires further encryptions. I also consider issues of usability, social context and social engineering.

2.2 Proximity Authentication

Significant research has been performed on authentication between network devices within a short distance. These approaches typically rely on inherent physical constraints. McCune et al. [9] proposed a mechanism that establishes a trusted channel between camera-phones by integrating public keys in 2D barcodes. Rasmussen et al. [10] introduced a proximity-based protocol to authenticate remote access for medical devices that are implanted in patients' body. This approach relies on the speed of sound, which is a constant. In addition, Cai et al. [11] proposed a mechanism that verifies when communication devices are co-located. This approach requires more than one antenna in the verifier, and is based on the relationship between signal parameters and distance.

The proximity-based authentication system, Amigo [12], relies on a mechanism that verifies co-located mobile devices by generating digital signatures from wireless radio strength, and then comparing the remote signature with the local one. Similar radio strengths indicate that two devices are within a short distance. Based on a similar idea, another proximity-based authentication system, Ensemble [13], relies on variation in radio signal strengths to determine physical proximity; trusted third parties (e.g. MP3 players, laptop computers) are included in this approach to monitor the security channel establishment and help verifiers prove authentication.

2.3 Software Attestation

Ensuring software execution on untrusted platforms is not the research contribution of this work. I recognize this as a distinct research challenge while building on the advances of others. There are two fundamental approaches to software attestation with the difference being the assumption of the (non) existence of a TCP. Seshadri et al. introduced Pioneer [14], a software attestation protocol that validates the execution of codes on an untrusted platform, even though malicious codes may be on

the machine. For embedded systems such as smart phones, SWATT [15] was proposed to detect malicious memory changes in embedded systems caused by viruses, Trojan horses, etc. The SWATT technique does not require prior authentication on the verified phone memory. Two types of attacks against these software-based attestation protocols were suggested [16]. To conquer these attacks, Jakobsson et al. [17] designed a new attestation protocol that evaluates both active applications in the memory and inactive programs that have been swapped out.

3 Threat Model

The primary threat to personal area network security is infection by malicious software (Malware). A number of malware types have been reported. Malware can be characterized by its payloads, targets, and mechanism for propagation.

Malware payloads refer to the primary actions taken [18] by active malware or the damage caused by malicious code [19]. Different types of malware may vary significantly in their payloads. For example, certain computer viruses (such as the well-known Melissa [20] and Iloveyou [21] viruses) were designed to tamper with users' files and/or operating systems. An example of the most destructive computer viruses would be CIH [22], which is capable of overwriting the BIOS on victims' computers. In addition to unauthorized modification on file systems, some malware steal data from the victims' computer. As an example, Schlegel et al. proposed Soundcomber [23], a context-aware sound Trojan that steals sensitive information from smartphones. Additionally, adware and spyware are often included as part of a software installation package [24]. Adware displays commercial advertisements and spyware monitors system surreptitiously, forwarding the collected information to third-parties. Botnet is an important type of malware. Instead of infecting a single machine, the botnet master can control thousands of bots. By directing a large number of infected machines, attacks originated from a botnet are often powerful. Typical malicious activities from botnets include DDoS attacks [25], email spams [26], etc.

Propagation mechanisms have also been used to categorize malware. Among all malware types, computer viruses and worms attract the most public attention. Generally, when a computer virus is executed, it replicates itself and spreads to uninfected files. Compared to viruses, worms are more active in propagation. In addition to self-replication, worms are capable of automatically detecting system vulnerabilities and infecting victim machines autonomously [27]. This characteristic leads to different propagation media for viruses and worms. According to recent studies, removable storage (such as CD, DVD, flash disks), emails and online downloads are the primary entry points [28] for viruses, while online transmission is a critical part in the propagation of worms.

It is much easier than many people would believe for malware infection. In fact, malware threats to mobile devices, especially smartphones, arise with the enhancement on device functionalities. It has been documented that the capabilities of web browsing, online messaging (e.g. send and receive multimedia emails or instant messages), reading flash-memory cards, or communicating by Bluetooth radios may all lead to vulnerabilities [29]. In other words, every machine faces a unique and wide range of possible malware attacks.

It has been reported that malware targeting on mobile devices has increased in recent years. According to the malicious mobile threat report published by Juniper Networks on May 2011 [30], the number of unique malware variants targeting the Android platform has increased by 400% since summer 2010. Malware detected on Nokia Symbian and Windows Mobile still dominate mobile malware according to the Jupiter sample database.

In addition to malware propagation, public recognition of malware threats remain insufficient. As shown in many forum posts, smartphone users do not realize that their phones need antivirus software just like computers. In fact, mobile devices are often more vulnerable to attackers than desktop computers. First, the mobile users are often considered more economically valuable targets. As the mobile applications and functionalities proliferate, more information is stored on the phones. Greater incentives are therefore created for malware development and distribution. Second, antivirus software is less well developed for mobile devices. Compared to antivirus programs developing for PCs, software functionality is preliminary or limited on phones. Furthermore, more malware is run in the background, which makes it difficult to detect, without the help of antivirus software.

In order to understand malware distribution, some researchers focused on the scale of machine subversion. In [31], Eeten et al. proved by a large-scale experiment and argued for Internet Service Providers (ISPs) as good control points for botnet mitigation. I agree that this is necessary but it is not sufficient. In this work, I propose that the malware mitigation could be augmented within personal area networks. I argue that this goal is achievable by incorporating collaborative rating and software attestation into the protocol design. The design problem is different for a PAN. I describe in Section 5 the technical heterogeneity and user homophily.

4 Assumptions

I made the following assumptions in this work.

- A1. Machines in a PAN are not infected simultaneously.

The proposed solution can only apply if the infection of a machine is not determined by the location of that machine. That is to say, in a home or personal network with x devices, the likelihood of subversion for these devices is independent. This is particularly the case in a typical home environment where there is significant heterogeneity in device models in the home. For example, individuals are less likely than firms or organizations to dispose of a machine simply because it has non-standard or dated capacities. PAN networks may include phones, laptops, desktops, eReaders, and a single router or server.

- A2. Power limits are not a concern in the home itself.

That is, when a mobile device is at its home, it is easy to plug in. Power consumption is a constraint in most cases when security protocols are designed for mobile networks. Because I am focusing on the home, the consumption of power is not such a limiting factor.

- A3. There is a transport layer that is shared to some degree. In other words, each device has the knowledge of other devices.

In the case there is not a shared transport layer, I assume the ability of a machine to sense the behavior of other participants through interactions during a re-introduction phase of the protocol. Devices are also required to share state.

- A4. There is a pattern or patterns of interaction generally on a daily or weekly basis.

Please note I assume that there are at least two devices in the constructed personal area network. The interactions among devices roughly follow certain patterns, particularly if proximity is considered an interaction. The interaction patterns between the devices can create or predict the context. For example, if an individual's daily schedule ends at 10pm, then a device login at 1am is particularly suspicious. If there are two devices, there is usually also a management device (router or bus). Notice that this depends on the colocation of the devices. For instance, when none of the mobile devices is present, the desktop should be inactive.

- A5. There is limited human capacity but there is the incentive enough to motivate set-up, interaction, and recovery.

An initial configuration is needed when constructing personal networks, while very little human interaction is expected afterwards. Each device in the personal network would be incorporated into the network with human interaction. I do not want authentications to run automatically when a new participant is added. Introductions are based on proximity authentication. Authentication is automatic when a known participant returns to the network. Humans engage in introduction and recovery only. Re-introductions and evaluations are handled by the machines.

- A6. Mobile devices are aware of their own locations and reintroduce themselves when returning to the home area network.

Considering the mobile nature of some devices in personal networks, it is necessary that mobile devices are given unique IDs so that linking authentication requests is possible. However, depending on time disconnected and probability of connection to external networks, the investment in authentication may change when a mobile device leaves and returns.

- A7. There is a limited period upon initial introduction during which devices are either trustworthy or can be made trustworthy with self-audit.

I argue that self-recovery is possible and can be automated once initiated by a human. The recovery task may be accomplished by a third-party recovery service. In this protocol, the self-recovery is executed from the central server which I will introduce later.

- A8. The central server is trusted.

Comparing to mobile devices, security measures on servers are more common. In addition, given the fact that the central server is responsible for proximity authentication, collaborative rating, and possible device recovery, the individual would have a strong incentive to protect the security of the central server. Note that I begin with a central server design and move to a distributed solution.

5 Protocol Design

5.1 Central Server Model: Participating Parties

Three parties are included in this protocol, the central server, the claimant and the verifier(s). Given that personal area networks are often implemented within relatively small ranges and with clear physical boundaries (e.g. home or office), I assume that communications among mobile devices and servers are trusted.

The central server is in charge of mobile device management. Specifically, a database is maintained on the server. It contains devices' physical addresses (for example, MAC addresses of WLAN or Bluetooth adapters), presence information of mobile devices (for example, records on entering and leaving the network), and collaborative rating results. Considering the importance of data transmission and storage to authentication, I recommend that the central server is located near the mobile devices. In addition, due to security concerns and data transmission rates, multiple personal networks should not share a central server.

The claimant is a mobile device that is being verified by other mobile participants. Each mobile device can be distinguished by its physical address, and it is also possible to include a secret message in the identification process. Note that being a claimant in a verification transaction does not exclude a device from being a verifier in another transaction. In this design, the data integrity of a mobile participant will be verified when the device enters a personal network, and on a pre-set frequency (for example, every two hours) afterwards.

In this design, the verifier(s) refer to one or more mobile participants that examine the identity and data integrity of the claimant. I require that at least one verifier presents in the network before the verification process starts. By observing the amount of the claimant's inbound and outbound data, deviation from historical patterns, the response to attestation challenges, each verifier submits a score to the central server, indicating the level of confidence that the claimant device has been subverted. No further action is taken until a final verification result is generated on the central server.

5.2 Protocol Phases

For the purpose of simplicity and clarity, I begin with a proposal that includes a central server. I then propose that authentications could also be accomplished without the central server. There are four components to the protocol: an introduction phase, a run phase, a reintroduction phase, and a recovery phase.

In the introduction phase I choose a proximity authentication. This authentication process ensures that the device is actually located within the house range. Specifically, a challenge is generated by the central server, and passed to the mobile device. The mobile device then responds to the central server. The mobile device will not be granted full network access unless it passes the test. Normally, these challenges rely on physical constraints and/or mathematical hardness. I therefore argue that it is infeasible for an outside attacker to pass this test.

In the reintroduction phase, mobile devices need to prove to the central server that they have been registered before. I base this phase on the design that each mobile device keeps historical keys for a period of time under a proper key management protocol. It is therefore possible to identify an old device by validating previous authentication information. Specifically, the central server generates a historical challenge such as a previous assigned key index. To pass this test, the mobile device searches for a previous key with the index, and sends the hash value of the key back to the server. After validation of the previous communication key, the server continues with a proximity authentication. After the device passes both history and proximity tests, a new communication key will be assigned by the central server, and the device database will be updated accordingly.

In the run phase the mobile devices audit each other in two ways. First, each device attests to the other that it has not changed state. Second, since historical activities have been recorded for each participating device, the transmissions of the devices could then be compared to past transmissions and states after reintroduction. If significant and sudden deviations are detected, then the recovery phase is entered. I base this phase on software attestation. Specifically, an application is installed on each mobile device, and performs scheduled verification tasks even if malicious programs are executed. The application is dedicated to check the memory status, as well as inbound and outbound network traffic. Results from the application will be shared with verifiers and the central server and be considered as a strong indication of whether a claimant has been subverted.

The recovery phase is focused on the repair of machine or malware infection. In the first implementation of this protocol I assume that this is a central server in the network to assist in recovery. In later instantiations recovery is addressed as a socio-technical challenge when the human is directed to implement recovery using a set of hard-wired external systems for that process. Majority voting by devices is required with per device risk assessment of other devices. Malicious reports on other devices' behaviors initiate automatic or human-driven recovery.

6 Example of Implementation

In this section, I propose a sample implementation of this security protocol. Please note that there are other possible technologies that may be utilized to achieve the security goal of the protocol. For example, in proximity authentication, the 2D challenge could be substituted by technologies such as Bluetooth pairing.

Two sets of cryptographic keys are utilized in this implementation: the public/private keys used to initialize symmetric keys and short-term symmetric keys. Each participant (verifier or claimant) holds a long-term public key. At the beginning of this protocol, the central server and the new mobile device need to exchange their public keys with the SSL or MQV protocol. While short-term symmetric keys are used in attestation message encryptions, long-term public keys are needed in both symmetric key generation and digital signatures.

In the introduction phase, the mobile device first submits its MAC address to the central server over the Wi-Fi connection. Upon receipt of the message, the server starts with a proximity authentication algorithm such as the so-called "Seeing-is-believing" algorithm, which was designed by McCune et al. [9]. In this example the server generates a 2D challenge (or displays an unchanged 2D bar code), and the mobile device uses its camera to capture the 2D code. Regarding the mechanism of proximity authentication, I propose that a nonce and the hash value of the server's public key should be included. For 2D the mobile device would then respond to the server with a message 'hiding' in the 2D bar code. To prevent fake responses from eavesdropping attackers, the new device also attaches the hash value from the response message, which can be generated by a message authentication code (MAC) algorithm with the long-term public key of the mobile device. After proximity authentication, the central server adds a new entry to the device database, and assigns the symmetric key to the device with a MAC result of that message to ensure information integrity. The entire process of this phase is illustrated in Figure 1. Notations that I use in the figures are summarized in Table 1.

Table 1. Notations in the Protocol Figures

Notation	Explanation	Primary Purpose
$K_{pub\text{-}serv}$/ $K_{pub\text{-}dev}$	Public key of the server/ a mobile device	Symmetric key allocation
$K_{prv\text{-}serv}$/ $K_{prv\text{-}dev}$	Private key of the server/ a mobile device	Symmetric key allocation
$Adrs_{dev}$	Physical address of a device	Device identification
$K_{dev,time}$	Symmetric key which allocates to a device at a certain time	Attestation message encryption
Hash	Hash function	Prevent message forgery
Nonce	Cryptographic nonce	Ensure message freshness
Timestamp	Current system time	Ensure message freshness

In the reintroduction phase, the central server first searches in the database for previous keys which have been assigned to the device in the past. The server then asks the device to send back the hash value of a key that was assigned at a particular point in time. The mobile device then looks up the particular symmetric key indicated in the challenge and attaches a hash value of the key in its response. The server compares the hash value with the previously stored information and decides if the mobile device has entered into the network before. In addition, similar to the introduction phase, the reintroduction phase then performs a proximity authentication

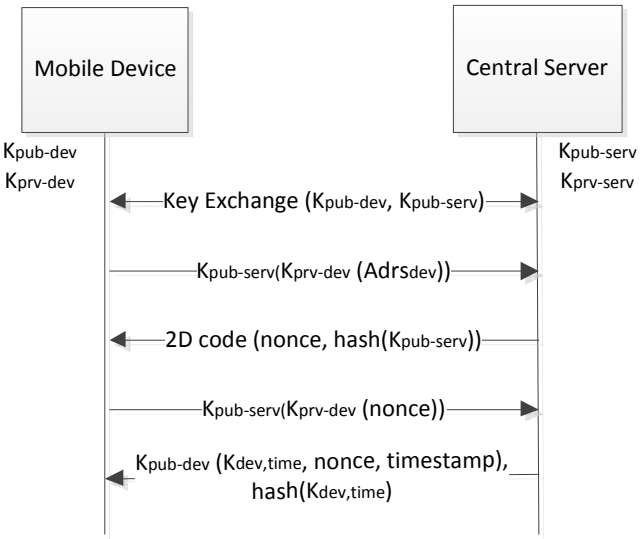

Fig. 1. Introduction Phase

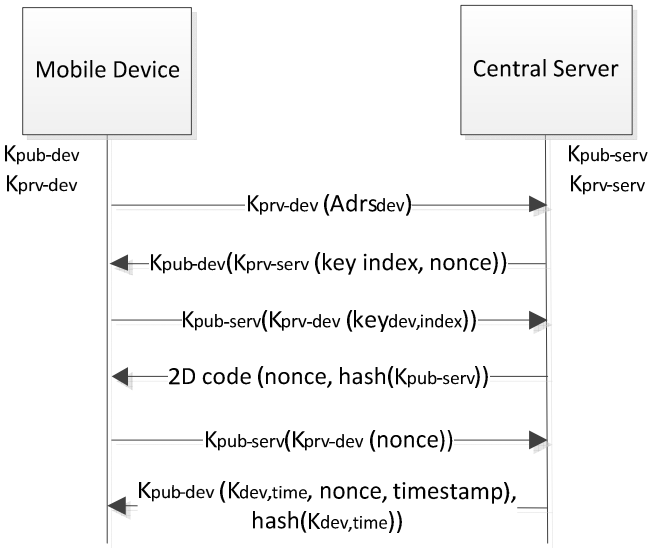

Fig. 2. Re-introduction Phase

and verifies the response regarding the 2D bar code. A new key is generated and assigned when both historical and proximity authentications have completed. The reintroduction phase is shown in Figure 2.

Figure 3 shows the run phase in this recommended implementation. Specifically, I apply a collaborative rating algorithm [32] on the run phase of the protocol. Each mobile device provides the attestation message, which may include recent application activities, inbound and outbound network traffic information to verifiers (by default, all other mobile devices in the network) and a timestamp. The device to be tested also needs to send a hash value of the attestation message generated by MAC with its symmetric key. The verifiers compare the provided information with previous 'experience' with the device. A score indicating the probability of a device being subverted is then sent to the central server. The hash value of the attestation message also needs to be forwarded to the central server to prevent masquerade and replay attacks. After evaluating all ratings from devices, the central server decides if a recovery phase is entered. In the recovery phase a special application is sent to the suspicious device to remove possible malware on the mobile device. This package should be transferred to the target machine by appropriate transport layer protocols, such as SSL. I argue that this self-recovery phase is feasible, and the execution of the recovery application could be guaranteed. While this application is protected by software attestation, and therefore can run without interference of malware, a self-recovery application can be sent to suspicious devices. This application should scan the suspicious device and determine the nature of repairs that needs to be undertaken before the actual work.

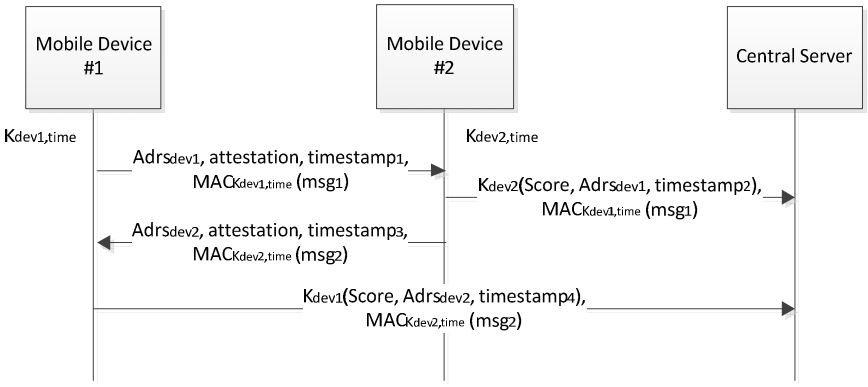

Fig. 3. Collaborative Rating (with server)

In addition, I propose that mobile device verification can also be performed without a central server, as shown in Figure 4. This process is based on the assumption that each device knows public keys for all other devices. This phase also starts with sending out attestation messages from one device to the rest of the PAN. Instead of sending scores to the central server, the verifier then sends the score with its digital signature on a hash value of the message to all other mobile participants. After this sharing process, scores may be generated by any one of the other devices.

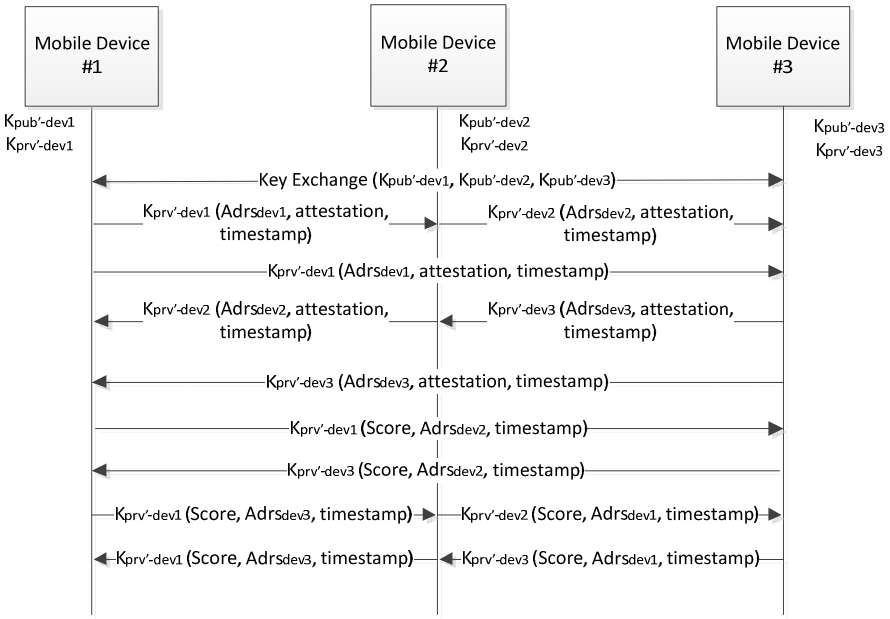

Fig. 4. Collaborative Rating (without server)

In this protocol anomaly detection relies on collaborative ratings from other mobile devices. I base this method on the observation that mobile devices owned by a single individual (or friends in a social network) tend to be similar in many ways (for example, the applications installed, the web browsing patterns, etc.). I analyzed the browsing history of over 1,000 college students that live in the same dormitory [33]. The subjects were selected for their homogeneity in order to mimic a social network. I finally showed that for a highly homogeneous network (with more than 5 participants), more than 95% of websites have been visited in the past. Therefore, if the previous activities of a claimant device are not available, the verifier may still generate a score by comparing its previous pattern with that of the claimant, while the predictability of human behavior has previously been seen as an obstacle. (e.g. in password generation [34]). In this case I leverage predictability and seek randomness or change as identifiers.

7 Conclusion

As the rapid development of portable electronic devices, it is common that people own more than one mobile device. While it is potentially vulnerable in small-scale computer networks, little research has focused on the security and privacy in this particular area. In this paper I proposed a preliminary security protocol for personal area networks. In this protocol, new mobile devices such as laptop computers, PDAs, cell phones are first introduced into the network after a proximity authentication; returning devices need to pass an additional history check before being added into the

network; each participating device performs collaborative anomaly detection with a pre-set frequency.

There are a few limitations in this work. First, this protocol only works for personal area networks. In other words, the assumptions and protocol design would be completely different if the mobile devices were geographically distributed. Additionally, I did not evaluate the performance of the protocol implementation, and I leave this as part of my future work. Certain types of attacks, such as denial-of-service, are not discussed in this work. Further, this protocol design relies on the assumption that the central server is always trusted. I realize that there are possible attacks against the central server during authentication and collaborative rating processes and plan to further the study in this general direction, with an eye towards DoS attacks on this protocol in particular.

Acknowledgements. The author would like to thank Professor L. Jean Camp for her valuable comments and suggestions and John McCurley for his editorial comments.

References

1. Bisdikian, C., Bhogwat, P., Golmie, N.: Wireless personal area networks. IEEE Network 15(5), 10–11 (2001)
2. Rogers, D.: Why Wireless Networks Are More Vulnerable Than Wired Networks, http://www.articlesbase.com/computers-articles/why-wireless-networks-are-more-vulnerable-than-wired-networks-886434.html (accessed 2009)
3. IEEE. IEEE 802.15 Working Group for WPAN, http://www.ieee802.org/15/
4. Niemegeers, I., Heemstra De Groot, S.: Research Issues in Ad-Hoc Distributed Personal Networking. Wireless Personal Communications 26(2-3), 149–167 (2003)
5. Hoebeke, J., Holderbeke, G., Moerman, I., Jacobsson, M., Prasad, V., Wangi, N., Niemegeers, I., Groot, S.: Personal Network Federations. In: Proceedings of the 15th IST Mobile and Wireless Communications Summit, Myconos, Greece (2006)
6. Jacobsson, M., Niemegeers, I.: Privacy and anonymity in personal networks. In: Pervasive Computing and Communications Workshops, pp. 130–135 (2005)
7. Patrikakis, C., Kyriazanos, D., Prasad, N.: Establishing Trust Through Anonymous and Private Information Exchange Over Personal Networks. Wireless Personal Communications 51(1), 121–135 (2009)
8. Patrikakis, C., Kyriazanos, D., Voulodimos, A., Nikolakopoulos, I.: Privacy and resource protection in Personal Network Federations. In: Proceedings of the 2nd International Conference on Pervasive Technologies Related to Assistive Environments, Corfu, Greece, pp. 29:1–29:5 (2009)
9. McCune, J., Perrig, A., Reiter, M.: Seeing-Is-Believing: using camera phones for human-verifiable authentication. In: IEEE Symposium on Security and Privacy, Oakland, CA, pp. 110–124 (2005)
10. Rasmussen, K., Castelluccia, C., Heydt-Benjamin, T., Capkun, S.: Proximity-based access control for implantable medical devices. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, Chicago, IL, pp. 410–419 (2009)

11. Cai, L., Zeng, K., Chen, H., Mohapatra, P.: Good Neighbor: Ad Hoc Pairing of Nearby Wireless Devices by Multiple Antennas. In: Proceedings of the 18th Annual Network & Distributed System Security Conference (NDSS 2011), San Diego, CA (2011)
12. Varshavsky, A., Scannell, A., LaMarca, A., de Lara, E.: Amigo: Proximity-Based Authentication of Mobile Devices. In: Krumm, J., Abowd, G.D., Seneviratne, A., Strang, T. (eds.) UbiComp 2007. LNCS, vol. 4717, pp. 253–270. Springer, Heidelberg (2007)
13. Kalamandeen, A., Scannell, A., de Lara, E., Sheth, A., LaMarca, A.: Ensemble: Cooperative Proximity-based Authentication. In: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, San Francisco, CA, pp. 331–344 (2010)
14. Seshadri, A., Luk, M., Shi, E., Perrig, A., van Doorn, L., Khosla, P.: Pioneer: Verifying Code Integrity and Enforcing Untampered Code Execution on Legacy Systems. In: Proceedings of the Twentieth ACM Symposium on Operating Systems Principles, Brighton, United Kingdom, pp. 1–16 (2005)
15. Seshadri, A., Perrig, A., Doorn, L., Khosla, P.: SWATT: SoftWare-based ATTestation for Embedded Devices. In: Proceedings of the IEEE Symposium on Security and Privacy, Oakland, CA, p. 272 (2004)
16. Castelluccia, C., Francillon, A., Perito, D., Soriente, C.: On the difficulty of software-based attestation of embedded devices. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, Chicago, IL, pp. 400–409 (2009)
17. Jakobsson, M., Johansson, K.-A.: Retroactive detection of malware with applications to mobile platforms. In: Proceedings of the 5th USENIX Conference on Hot Topics in Security, Washington, DC, pp. 1–13 (2010)
18. Kanellis, P. (ed.): Digital Crime And Forensic Science in Cyberspace. Idea Group Publishing, Hershey (2006)
19. Malware Wiki, http://malware.wikia.com/wiki/Payload (accessed 2011)
20. CNN. Clone of 'Melissa' virus infects the Internet, http://articles.cnn.com/2001-04-19/tech/virus.matcher_1_melissa-bug-windows-address-original-melissa-virus?_s=PM:TECH
21. CNN. Destructive ILOVEYOU computer virus strikes worldwide, http://articles.cnn.com/2000-05-04/tech/iloveyou.01_1_melissa-virus-antivirus-companies-iloveyou-virus?_s=PM:TECH
22. CNN. CIH virus may hit on Monday, http://www.cnn.com/TECH/computing/9904/23/cihvirus.idg/index.html?iref=allsearch
23. Schlegel, R., Zhang, K., Zhou, X., Intwala, M., Kapadia, A., Wang, X.: Soundcomber: A Stealthy and Context-Aware Sound Trojan for Smartphones. In: Proceedings of the 18th Annual Network & Distributed System Security Symposium (NDSS 2011), San Diego, CA, pp. 17–33 (2011)
24. Stafford, T., Urbaczewski, A.: Spyware: The Ghost in the Machine. Communications of The AIS (2004)
25. Mirkovic, J., Prier, G., Reiher, P.: Attacking DDoS at the Source. In: Proceedings of the 10th IEEE International Conference on Network Protocols, Washington, DC, pp. 312–321 (2002)
26. Levy, E.: The making of a spam zombie army. Dissecting the Sobig worms. In: Proceedings in IEEE Security & Privacy, Oakland, CA, pp. 58–59 (2003)
27. Pfleeger, C., Pfleeger, S.: Security in Computing, 4th edn. Pearson Education Inc., Boston (2006)
28. Skoudis, E., Zeltser, L.: Malware: fighting malicious code. Prentice Hall PTR, Upper Saddle River (2003)

29. Lawton, G.: Is It Finally Time to Worry about Mobile Malware? Computer 41(5), 12–14 (2008)
30. Juniper Networks Malicious Mobile Threats Report 2010/2011, http://www.juniper.net/us/en/local/pdf/whitepapers/2000415-en.pdf (accessed May 2011)
31. Eeten, M., Bauer, J., Asghari, H., Tabatabaie, S.: The Role of Internet Service Providers in Botnet Mitigation: An Empirical Analysis Based on Spam Data. In: Proceedings of The Ninth Workshop on the Economics of Information Security (WEIS 2010), Cambridge, MA (2010)
32. Kinateder, M., Rothermel, K.: Architecture and Algorithms for a Distributed Reputation System. In: Nixon, P., Terzis, S. (eds.) iTrust 2003. LNCS, vol. 2692, pp. 1–16. Springer, Heidelberg (2003)
33. Dong, Z., Camp, L.: The Decreasing Value of Weak Ties in Recommended Networks. ACM SIGCAS Computers and Society 41(1) (2011)
34. Burr, W., Dodson, D., Polk, W.: Electronic authentication guideline: Recommendations of the National Institute of Standards and Technology (2006)
35. Jansen, W., Gavrila, S., Korolev, V.: Proximity-based Authentication for Mobile Devices. In: Proceedings of the 2005 International Conference, Las Vegas, NV, pp. 398–404 (2005)