Approaches for Ensuring Security and Privacy in Unplanned Ubiquitous Computing Interactions

V. Ramakrishna, Kevin Eustice and Matthew Schnaider
Laboratory for Advanced Systems Research
Computer Science Department
University of California, Los Angeles, CA 90095
{vrama,kfe,matt}@cs.ucla.edu

Abstract

Modern technology and omnipresent computing and communication facilities are leading us closer to the ubiquitous computing vision. However, the very nature of ubicomp infrastructure, the openness of the environments and the characteristics of the interactions pose unique security and privacy challenges. We anticipate that the vast number of interactions will be unplanned and will occur among mutually unknown and untrusted systems. Mobile components will often find themselves in unfamiliar surroundings, forced to work with infrastructure whose trustworthiness cannot be determined. We must identify and address the security issues inherent in these types of interactions before a large-scale deployment of vulnerable infrastructure begins to pose a serious threat. Current security solutions for mobile computing and wireless communication are not sufficiently scalable or flexible to protect the heterogeneous and highly dynamic systems of the future; they do not even satisfactorily solve current mobile computing security issues.

In this paper we address the problems inherent in the infrastructure and in the interacting devices themselves. We also identify device theft as a problem exacerbated by mobile and ubiquitous computing. We emphasize device-based approaches towards handling security and privacy, broadly classifying them into three categories which, when taken collectively, form a three-layer defense for devices. These categories are: 1) resource and content protection mechanisms, 2) secure protocols for service discovery and assignment of resource access, and 3) trust frameworks. These categories are neither mutually exclusive nor exhaustive, yet they collectively address challenges inherent in a wide range of ubicomp scenarios. We emphasize protocol-based solutions and, to a lesser extent, trust frameworks. These aproaches are being investigated in the context of the QED and policy-guided negotiation work currently underway as part of our *Panoply* ubiquitous computing project.

1. Introduction

Ubiquitous computing promises a vision of computing capabilities at any place and at any time, supporting all kinds of human activities, including even the most mundane. A transition from mobile computing to ubiquitous computing is well underway thanks to both academic research efforts and commercial enterprises. Three important technological factors are contributing to this transition: 1) rapid growth and proliferation of wireless networking facilities, 2) computing and sensing components embedded in our surrounding environments, and 3) availability of smaller portable devices that can run most applications required by a mobile user. Mark Weiser envisioned a future in which computers would fade into the background [Weiser1991]. A more realistic vision, and one that is currently attainable, still involves devices that are recognizable to users as computers. This model of computing is typically distinguished from ubiquitous computing (*ubicomp*) as *pervasive computing*. In the pervasive computing paradigm, devices and networks communicate with each other and deal with each other in a more aware and intelligent fashion, without involving a human unless absolutely necessary. Most of these interactions occur in a mobile context and in an unplanned fashion. The onus is upon the devices and the applications to ensure that tasks proceed smoothly, hiding details from users. The challenges in pervasive and ubiquitous computing are similar to mobile computing, but with a higher scale of mobility, dynamism, and heterogeneity.

Primary networking challenges have more or less been addressed. These include the ability to discover networks and associate with them, and the addressing issues that are necessary to establish and maintain network connections. Efforts at the application layer have been made, and are still ongoing, to achieve seamless mobility of networked applications. As a result, the networking infrastructure can now handle complex tasks that were formerly relegated to the user.

Even as we design technology with new and better functionality, we must explore potential pitfalls. What happens when one or more of the participants in a mobile interaction do not play by the rules the designers of the mechanisms envisioned? Attackers could use their anonymity and the nature of network-based protocols to breach the security of trusting devices or obtain sensitive information. The networking infrastructure that makes mobile computing possible could also be subverted for illegitimate purposes. We will further explore the vulnerabilities inherent in these unplanned interactions and discuss how a complex balancing act is required to make ubiquitous computing usable, as well as secure.

1.1 Characteristics of Ubiquitous Computing Interactions

Ubiquitous interactions rely primarily on wireless network connectivity between numerous classes of devices. In this context, wired portable computing is significantly less interesting, and the networking and addressing issues have, for the most part, been dealt with; additionally, there is a much higher level of trust and accountability.

Interactions among mobile devices and ubiquitous infrastructure components are directed towards the discovery and access of external resources and information that are required for local applications. These include services provided by the immediate environment—typically wireless connectivity, connections to remote computers through the Internet, and sensory output. Most current applications of mobile computing involve access of web-based services. This requires that devices be able to associate with networks and configure Internet connections; the remaining application tasks are explicitly performed by the users. The transformation to a pervasive computing environment will increase the demands on the devices and the networks to which they connect. A much wider variety of tasks will be supported, and the devices must be more intelligent and aware in order to minimize the work that users must do. Users will expect less intrusiveness, seamless communication, and better performance.

Devices and networks will become more autonomic, specifically more self-configuring, self-adjusting, and self-healing. In the simplest form of mobile computing, where users explicitly handle applications and provide other input, the networking issues have relatively fewer security implications. When devices and applications are expected to perform tasks that satisfy user desires, without low-level user input, and sense and adapt to context changes, the security problems are magnified. Workable solutions must be provided so that users can trust their devices to run in an automated fashion and handle private data.

Ad hoc or unplanned interactions, which we believe will be very common in the emerging computing landscape, will present situations where there is a lack of familiarity or trust among the interacting entities. We cannot guarantee that different mobile devices and networks will have the same security or data privacy standards, and one challenge is to determine the opposite party's standards. Even in cases where interactions occur between known entities or entities with verifiable security relationships, the lack of trustworthiness of the wireless communication medium calls for precautions. This medium enables anonymity of entities; if such entities turn out to be malicious or compromised, they could provide fake services and obtain sensitive information. It is conceivable that the problem could be mitigated somewhat through the imposition of strict security standards and a universal trust framework, but such a worldwide standard would be impractical and impossible to enforce. It would also limit the options for each independent domain to determine its security policies. It also does not solve the problem of adaptation with context, since all possible situations cannot be planned for in advance.

1.2 Trading off Security, Privacy and Usability

Security has proven to be a challenge when it conflicts with user convenience and ease of use. Users dislike entering passwords repeatedly in order to perform tasks that require extra privilege. If the system provides an option of storing the password for subsequent use, many users would make use of it. Likewise, when a sensitive transaction requires the release of identity information and secret keys, privacy is often sacrificed with little thought. These examples and others indicate that there is a three-way tradeoff in security, privacy and usability that every system designer must address. In this context, we define usability as the ease of handling devices and applications, with minimal input and feedback required from the user for successful operation.

This complex tradeoff acquires a new dimension in mobile and ubiquitous computing due to the wireless medium, the open environments, the unplanned nature of interactions, and the anonymity of computing entities. In a static context, there is an added degree of trust, which is absent in a mobile wireless context. When communicating with strangers, the more knowledge a device gains about the other party, the better it can assess the appropriate level of trust to place in that party. Intrusive procedures for assessing trust could be used, indirectly leading to more security. This would make an entity more confident about allowing access to a local resource or giving up some private information in the hope that this might result in some benefit without the cost of misuse. Trust-based security therefore inevitably results in a loss of privacy. Conversely, a conservative policy could result in more privacy but a lower probability of a successful interaction because neither entity will be able to gain sufficient trust in the other. Also, in order to be absolutely secure, many security decisions will have to be made explicitly by the user, which is contrary to the ubiquitous computing goal of reducing human intervention. Many applications will also require the free exchange of privileged information such as location, local capabilities, and constraints. Applications could run in an automated fashion if free exchanges were allowed, but privacy constraints could force a more conservative approach. Various service discovery and access mechanisms could also result in inadvertent exposure of private content and resources, owing to careless design or a lax policy. Submitting to privacy demands could detract from the user experience by restricting the performance of tasks. Alternatively, if the system cannot reconcile privacy demands with the task requirements, user intervention may be required. Privacy, therefore, will often be at cross-purposes with usability.

This three-way tradeoff severely impacts and potentially restricts security and privacy choices in ubiquitous computing, where usability and performance are key. Most research efforts in wireless networking and ubiquitous computing have emphasized the usability aspect at the cost of security and privacy [Brooks1997] [Román2002]. Though this results in a richer set of applications and functionality, a retrofitted security solution usually employs fairly rigid policies which interfere with many of the features that make the system usable. The approach we take is to analyze ubicomp interactions as a whole, rather than on a per-application basis. In this paper we attempt to identify the unique security threats and privacy and access control issues that are posed by device mobility and mutual anonymity of interacting devices and networks. In Section 2 we outline the threats posed by insecure infrastructure and malicious entities, and observe how mobility impacts systems in a negative way. In Section 3 we describe currently used and proposed approaches for maintaining security and privacy. We classify device-based security solutions into three categories, each providing security at a different level; this helps us to better understand and analyze these solutions.

2. Challenges of Unplanned Interactions

In the traditional computing paradigm, devices operate in a few established environments. Ubicomp necessitates a break from this pattern. Traveling from well-known and presumably safe environments to unfamiliar and potentially hostile ones poses many security challenges in mobile and pervasive computing. Likewise, the computing elements embedded in the infrastructure will encounter new and possibly unsafe devices all the time. Though a certain amount of paranoia is both healthy and necessary, it should not prevent devices from running essential tasks for users. Both users and their devices must take precautions. Devices should be able to verify the authenticity of the networking infrastructure, and the machines with which they communicate. Additionally, they must be able to assess the security risks in carrying out such interactions. Similar caution must be exercised by infrastructural components when interacting with unknown mobile devices that have entered communication range. Even if the external environment does not pose a threat, it may hardly be friendly. In these circumstances, protecting the integrity of system resources and data, as well as maintaining a necessary amount of privacy, is difficult. Challenges arise primarily due to communication with strangers, but in the absence of a trustworthy networking infrastructure, similar problems may afflict communication with known entities too. We address security and privacy issues both from an infrastructural and a device point of view; these issues include device and service provider authentication, the risks of habitual mobility, intelligent failure modes, and software agents. Challenges in each area must be addressed by researchers in order to achieve a complete security solution.

2.1 Infrastructure Security and Privacy

With traditional 802.3 Ethernet-based networking, when one plugs a device into a wall jack, it is typically assumed that the device receives connectivity from the local infrastructure. Clearly, there are possible attacks in this space, but in general this is a reasonable assumption since a physical wire acts as a physical metaphor tying the device to the physical environment. Wireless communications lacks this metaphor; absent policy, our mobile wireless devices can and will receive connectivity from any accessible service providers. This poses potential problems in that traditionally we have trusted our infrastructure to provide network services such as routing and name lookup. Malicious service providers can capture wireless clients and reroute requests to malicious services; such services are intended to duplicate legitimate services and capture personal identification information such as logins, passwords, credit card information, and so on. This type of session hijacking can be performed at the routing layer or by subverting DNS.

There are several security problems here—one is the assumption that the networking infrastructure should provide routing and naming services in a secure and trusted manner; another is that one's device will associate with a given infrastructural component. These problems are related, especially if we seek to use trust relationships to deal with the former. The latter challenge is a problem of device authentication—i.e., how do we make sure we connect to the café's access point and not the malicious access point in a patron's backpack? This is a subset of the general device authentication problem—how do two mutually unknown devices authenticate one another?

Apart from ensuring the authenticity of the service provider whose network a mobile device is using, we must also deal with issues of data confidentiality and location privacy. These problems are exacerbated by the broadcast nature of the wireless medium, where eavesdropping is trivial for any device with a wireless card. Data confidentiality can be handled through encryption, and much research has gone into developing standards for 802.11 networks, which are mentioned in Section 3.1. But even if the communicated data cannot be interpreted, an eavesdropper can still infer the location of the communicating device and the entities it is talking to, which is information mobile users might want to keep private.

2.2 Device Security and Privacy

A number of security and access control problems lie within devices (or the end points of network connections) themselves. The problems arise due to misconfiguration, ineffective or bad security policies, vulnerable applications and insecure processes for remote discovery, access, and use of resources. Similar problems occur even in static desktop-based computing when communicating over the web, but the nature of devices in pervasive computing, mobility, and the frequency of contact with strangers worsens existing problems, as described below.

The Risks of Mobility

Mobility tends to exacerbate existing security and privacy challenges, such as system vulnerabilities and information leaks in network protocols. A mobile device moves in and out of environments with many unknown and potentially hostile devices, without the protection of infrastructure-based firewalls. This behavior exposes the device to more potential attackers, magnifying the risk of software vulnerabilities. When the mobile device is eventually taken home or to work, it passes behind traditional firewalls, possibly carrying an infection or an intruder.

A next-generation security system needs to be aware of these peripatetic devices that operate within its purview. The knowledge that a device is mobile and transient may allow the infrastructure to provide better support. Steps need to be taken to ensure the integrity of mobile devices and protect the rest of the local network from potential abuse. Challenges here include developing techniques to protect the network from mobile nodes while not overly inhibiting functionality.

Intelligent Failure Modes for Pervasive Security

Failure is an unfortunate fact of life. Mobile devices will be compromised, either over the network or by theft. It is incredibly important that the failure modes of such devices be engineered to minimize the impact of compromise. To that end, we need to focus on theft mitigation, reducing the ability to use or harvest data from a stolen device, as well as application limitations that restrict the powers of a compromised application, thereby protecting system integrity.

Theft Mitigation

Expensive and highly-portable mobile devices present tempting targets to thieves. In a time when identification theft is becoming all too common, these devices also represent a treasure trove of personal information. An important challenge thus is to mitigate the impact of theft—that is, reduce the utility of a stolen device, both in terms of actual functionality and in terms of extractable information. Additionally, recovery mechanisms including "phone home" features and secure remote localization capabilities would be valuable in the mobile device feature set.

Restricting Capabilities and Information Leaks

Mobility-oriented applications must be designed to limit the impact of compromise through segregation of functionality and by adopting the *least privilege* paradigm, limiting the application's privileges and data to those necessary to accomplish its tasks. This helps reduce the impact of malicious or compromised applications. Applications may deal with sensitive user data, including authentication information and financial data, as well as sensitive user context such as location or social relationships. A related challenge here is to limit the exposure of this data to the minimum necessary. Context can be made accessible at multiple fidelity levels, and only the necessary level of context should be exposed to the application. For example, location context can have levels such as "UCLA," "Boelter Hall," and "3564 Boelter Hall." The level of context exported to the application may depend on user policy, application needs, or the security characteristics of the local environment.

Similarly, the least privilege paradigm must be applied to information that is being transmitted. Remote computers should not be allowed to see more than is necessary for immediate purposes. Otherwise, information such as system or user identification information, system behavior patterns, etc., may be leaked to potentially hostile users. This information could be used by thieves to better target victims—i.e., the thief knows that one bus passenger has an expensive laptop and can determine which passenger, without even seeing the laptop. Similarly, if the presence of a given laptop in one's home is highly correlated with user presence, then radio emissions can be used to determine when someone is at home. In general, we need to be more careful about the radio emissions of our devices, as they do leak substantial information.

Software Agents and Mobile Code

Software agents and mobile code are frequently used in ubiquitous computing contexts to enable interoperability, application segmentation and migration, as well as customized handling of system operation. This raises serious security challenges. Mobile code may potentially harm the hosting device, or behave in unpredictable ways. The issuer of the mobile agent wishes to trust the result of the mobile code's execution, but the hosting device has control over the code. This poses a problem. Although this problem exists in the wired Internet, future pervasive environments may depend hugely on mobile agents to perform tasks, including the discovery of networks and services when devices are mobile. Such agents will be especially valuable in handling unplanned interactions.

Today's users already run a great deal of mobile code in the form of Java, JavaScript, Shockwave/Flash, and ActiveX controls. In many cases, mobile code intentionally or unintentionally has access to sensitive user data, often much more data than it strictly requires. We need reliable methods for protecting user data from disclosure and tampering while still permitting the execution of mobile code that is beneficial to the user. Accepting and running mobile code will require enhanced approaches for verification of code properties and establishment of trust.

3. Approaches

The concerns raised in the previous section can be summarized as: 1) protecting the integrity of the devices and networks, 2) preventing unnecessary data exposure, and 3) granting unknown entities permission to access private resources. As discussed in Section 1, enabling open interactions among mobile and infrastructure-based devices is a primary ubicomp goal. An impenetrable security system, though desirable in principle, would restrict access to many types of ubiquitous computing services. Instead, an effective system must be flexible in its approach to ensure both security and usability.

We can and must try to secure the networking infrastructure from malicious entities and eavesdroppers. Approaches to address this are discussed in Section 3.1. These will not solve the complete problem; traditional end-to-end security is still necessary. For the purposes of this discussion, we have chosen to define three subclasses within the solution space. While these subclasses are not exhaustive, we believe these are areas where further research could substantially address security and privacy challenges faced by most ubicomp scenarios.

The first class of approaches (Section 3.2.1) attempts to secure resources and content directly at the time of access. Such approaches also include situations where the device in question falls under the control of external entities, directly through theft or indirectly using mobile code. The second class of approaches (Section 3.2.2) comprises secure processes and protocols for interactions between devices, resulting in discovery of external resources and assignment of permissions to access those resources. The security and privacy solutions are managed by the device and are not tied to individual resources; the devices here are containers and controllers for a set of resources and services. The third class of approaches (Section 3.2.3) consists of cross-domain security frameworks that impose security solutions in a top-down manner. Any two entities that come across each other in a pervasive computing world can determine the nature of their relationship and the scope of their interactions through such a shared framework. All trust frameworks, certificate hierarchies, and access control solutions for open systems fall under this category.

From one perspective, these three classes of solutions could form three layers of defense for any kind of interaction that takes place in a ubiquitous environment [Eustice2003a]. The trust approaches could help to determine the security basis for interaction among computing entities. Protocols could be used by such entities to discover each other's resources, securely configure permissions for access, and perform security-sensitive actions. At the innermost layer, once devices get to know each other's resource capabilities, they could directly access those resources which are guarded by low-level protection mechanisms. These three sets of approaches are neither mutually exclusive nor exhaustive. Furthermore, it is unlikely that a complete security solution can be drawn from any one of them alone. Trust frameworks are usually coupled with secure protocols for determining trust in external entities before permitting discovery and access. Resource protection mechanisms can be used in a scalable way in this context only if they are accompanied by a dynamic process of discovery and reconfiguration of local security state. An ideal security solution would combine appropriate features from all three classes of approaches that prove well suited to deployment in dynamic environments. Before we look at examples of different approaches from each of the categories defined above, we consider some mechanisms for securing network infrastructure.

3.1 Networking Infrastructure Security and Privacy Approaches

The most obvious technique used to maintain data confidentiality over any network link is encryption. As mentioned in Section 2, the broadcast nature of wireless communication makes this problem harder. Despite this, cryptographers and security engineers have developed workable security solutions for data confidentiality at the wireless MAC layer. Given the initial failure of the 802.11 WEP standard, [Borisov2001], WPA was developed to overcome WEP's problems with stronger authentication schemes and a key management system. At higher layers in the network stack, devices have even more choices, and we can select from a variety of cryptographic schemes and key exchange protocols.

Preventing an eavesdropper from inferring the location of a device and the identity of the devices it is communicating with is still hard, mainly because of the broadcast nature of the communication medium. Also of interest is research in secure network discovery and connection to authentic service providers. This handles simultaneous discovery and authentication of a wireless network through automated means, which is complementary to the problem of private communication after connection establishment. Secure enrollment of a device to a network promises to mitigate the security problems associated with service provider selection and authentication, as described in Section 2.1.

Device Enrollment

The general problem of secure network enrollment within pervasive computing environments has been considered by several other projects. The canonical reference is Stajano and Anderson's *Resurrecting Duckling* [Stajano1999] where the authors presented a model for imprinting wireless devices with network membership information through brief physical contact. In the model, physical contact is required to create a logical connection between two otherwise wireless devices. The *mother duck* controlling device would maintain absolute control over a set of duckling devices and their respective policies.

The duckling model has been further extended by PARC [Balfanz2002] and applied to home and enterprise-wide wireless LAN setup [Balfanz2004]. PARC removes the requirement for a secure side-band channel through the use of public key cryptography—this increases the baseline requirements for member devices, but allows more open side-band channels such as infrared. Recently, other approaches have investigated the use of embedded cameras to capture visual authentication information embedded in barcodes attached to devices [McCune2005], as well as the use of audio cues [Goodrich2005] coupled with displayed textual information.

3.2 Device-Based Security and Privacy Approaches

In this section we discuss approaches for maintaining security and privacy that are executed locally on devices. In general, these solutions assume the presence of a trusted communication infrastructure, though some trust-based solutions circumvent the networking problem altogether by enforcing stringent authentication schemes at the end points.

3.2.1 Resource/Content Protection and Access Control

In the world of pervasive and ubiquitous computing, data is often at risk for disclosure or tampering. Data lives on mobile and portable devices and may be subject to theft. One approach to protecting the privacy of user data is to integrate the protection mechanisms with the resources themselves.

Secure File Systems

Cryptographically secure file systems have been available for more than ten years [Blaze1993] [Wright2003]. In practice, though, such file systems are not widely in use. Furthermore, even when such systems are used, it is common for users to store sensitive key material on the same device that is being protected. As a result, when devices are lost or stolen, it is likely that the information on those devices can be easily accessed by even modestly skilled attackers.

Additionally, when a device is taken over by malicious code, that code normally has full access to data on the device, including any encrypted data that the user may access. Typically, users rely on one master key or password to access their encrypted file systems. Thus, if the user accesses any encrypted data item, it is likely that all encrypted data items within that data-store are exposed to any malicious code that may be running on the device.

In order to protect data in this scenario, portable devices should not be the custodians of the key(s) to the sensitive data they hold. Rather, keys should be stored elsewhere and provided to applications on demand, based upon context and policy. If this were the case, certain data would be completely inaccessible to even the most determined attacker if the device was lost or stolen. Even in the case of device infection, much, if not all, sensitive data would be protected, ideally until the malicious code was discovered and purged.

Zero-Interaction Authentication

One system that possesses many of the properties mentioned above is Zero-Interaction Authentication (ZIA) [Corner2002]. In ZIA, each file is encrypted under a symmetric key, and that key is then encrypted with a key-encrypting key. A small security token, separate from the device itself, is the only entity that can decrypt file keys. The device must be in the presence of the token in order to access its own encrypted files. Thus, in our loss or theft scenario, ZIA cryptographically protects user data from disclosure from even the most determined adversary.

In addition to ZIA, other novel uses of cryptographic file systems and key management could greatly reduce the risk of disclosure of sensitive data through device loss or theft, or even device infection. Such systems should be informed by context and policy to provide more fine-grained and flexible control over encrypted data and associated keys than is currently provided by ZIA and other encrypted file systems.

Proof-Carrying Code

Although we can mitigate the dangers of device loss and theft, and we can to some extent limit the amount of sensitive data that is exposed in any particular context, it may be desirable or useful to run foreign code in various ubiquitous computing scenarios. Though many mobile code systems employ some facility for sand-boxing, much mobile code still has far more access than necessary, and often far more access than is safe. One possible approach to alleviating this problem is to use proof-carrying code [Necula1997]. In the ubiquitous world, devices will likely be offered mobile code from a variety of trusted and untrusted parties. In many cases, the user will explicitly run such code. In other instances, the device will be asked to run the code on behalf of the user. Proof-carrying code would maintain the usability we want, while preserving the safety and security of sensitive resources.

Proof-carrying code can provide proof of programmatic side-effects and invariants that can be reconciled with local policy. Depending on the level of trust (if any) ascribed to the provider of the code, the device can make safe and informed decisions without having to involve the user every time the question of executing mobile code is raised. Not only can proof-carrying code protect against malicious code that steals or tampers with sensitive user data, it can also preserve the overall integrity of the device, and may also have the added benefit of increasing the reliability of the device as a whole.

Proof-carrying code has addressed a very important problem, but we feel its complete potential has yet to be explored. A large number of ubicomp applications will depend on mobile code, and quick verification of security policy compliance would be very valuable. Application of proof-carrying code to ubicomp warrants further research.

3.2.2 Secure Interaction Protocols

Various situations will occur in ubiquitous computing where devices will need to discover each other's services and establish access permissions. The processes and protocols for managing secure discovery and assignment of access permissions comprise a different set of approaches, complementary to the resource protection mechanisms described above.

Trust Management

Trust management is a process that unifies security policies, credentials, authorization, and access control. This concept was introduced in PolicyMaker [Blaze1998] and refined in KeyNote [Blaze1999]. The process involves a request to perform a security-impacting action or to access private information or resources. The requestee runs a compliance checker taking as input the request, associated credentials from the requestor, and its local policies. If no conflict is detected, the request is granted; otherwise it is refused. This security or trust management solution requires a common trust framework, including a credential vocabulary, in order to be effective. In the mobile computing context, this solution maintains security and access control to the degree specified by the policies. One drawback is that the policies are static and are not sensitive to context changes. Although this process maintains the privacy and security of the requestee, it is not sensitive to the privacy considerations of the requester, who must provide all information and credentials demanded if the interaction is to succeed. Though both PolicyMaker and KeyNote were designed with traditional computing in mind, the technique could as well be used in pervasive computing when combined with a suitable process for discovery of networks and services.

Quarantine and Examination for Mobile Computing

We have explored a new paradigm for mobile and ubiquitous security called QED [Eustice2003b], or Quarantine, Examination, and Decontamination. In this paradigm, before mobile devices are allowed to join a wireless network, they are inserted into a quarantine zone.

This is done to protect other local network participants from potential malware carried by the mobile device. While in quarantine, the device is subjected to an examination process that can include a variety of techniques such as external port scans and service identification, as well as internal tests that require cooperation of the device, such as virus scans and service patch determination. If problems such as vulnerabilities, undesirable services, or compromised software are found, the device may go through a decontamination phase in which the problems are, if possible, rectified. Once the infrastructure is confident that the device poses no threat, it is allowed to fully participate in the local network.

A system like QED demonstrates how security and privacy requirements may be at odds in a pervasive computing scenario. Security is enhanced if mobile devices run foreign code as instructed and report results truthfully. But this results in a loss of privacy for the device. Also, running arbitrary code itself requires a high measure of trust in the code provider. These are extremely important issues that require further research. The use of proof-carrying code techniques to verify policy compliance of examination modules deserves serious investigation. Also, verification of authenticity of returned examination results is an interesting problem; this could also have implications for digital rights management.

The Cisco Network Admission Control (NAC) system [Cisco2003], a commercial product that is part of the Cisco Self-Defending Network Initiative, enforces access control in a domain through quarantine and examination. Access control decisions are based on a domain's security policies and involve checking incoming devices for vulnerabilities and infections. NAC suffers from certain drawbacks compared to QED; notably, it does not provide support for decontamination. Also, QED is completely software-based and open source, whereas NAC is integrated with Cisco hardware products. Using QED, security policies could be enforced in a flexible manner with access limits varying with degree of compliance. Also, the relationship between the mobile device and the network is more symmetric; this allows both the network and the mobile device to consider the privacy implications of running foreign code or releasing sensitive information. The primary goal of NAC is to enable domains to enforce security policies, and the relationship is inherently asymmetric. This solution will only work when a device interacts with familiar networks, and it is not flexible or scalable enough for ubicomp interactions.

Solutions performing QED functions are very valuable to mobile users who would be more tolerant of the added overhead. In the ubiquitous computing vision, applications must run smoothly in the face of frequent context changes. Scaling QED to work in those types of environments is well worth exploration.

Automated Peer Negotiation

We are exploring automated and flexible negotiation techniques among peers to enable interoperation among heterogeneous devices with diverse security and privacy policies [Eustice2003a]. Services can be discovered and resource access agreements can be reached via negotiation, while maintaining local security and privacy policies. Negotiation itself is not a new security mechanism, but rather ensures as much security as can be obtained through existing enforcement mechanisms. The policies, which are private to a system, describe the various constraints and inter-dependencies among system objects, and also describe the state of the system and the properties of its resources and mechanisms. The high level constructs are described in a common semantic language; we are leveraging Semantic Web frameworks like RDF and XML for this purpose.

Negotiation is a flexible way for two entities in a ubicomp context to access each other's resources up to the maximum allowable risk and within the resource usage policies local to each.

Most other approaches usually fall under extremes. At one end of the spectrum, some approaches for interaction obey rigid protocol semantics and are usually not applicable outside a particular domain. At the other end, open environments allow free and easy access without regard to security, such as early versions of Jini [Waldo1999]. Negotiation offers a way to balance the risk of resource access or exposure of private information and the utility of permitting that operation. The crucial aspects are: 1) a trust/risk model that allows assessment of the risk associated with an operation or the trust gained in the other party, 2) a utility model that allows assessment of the benefits of gaining certain resources, and 3) a set of heuristic functions that allows an entity to determine when utility outweighs risk. Of course, there will be situations where the other party could be determined to be malicious, or mobile code found to contain a virus, in which case utility will rarely balance risk. The functions can be computed using the policies local to a system, which include user preferences as well as knowledge of security properties; e.g., risk of opening up a network port, how much trust does possession of certificate X inspire, and so on. The negotiation protocol proceeds through a strategy whereby the parties can trade information, propose alternatives, and compromise within the limits of their policy constraints and the derived heuristic values. The policy language itself is backed by logical semantics and has a reasoning engine that enables query processing, knowledge chaining, and determination of conflicts. This is promising research, both from the security and privacy viewpoint and from the viewpoint of matching heterogeneous systems with available resources in a context-sensitive manner.

Negotiation as described above enhances the scope of prior work in automated trust negotiation [Winslett2003], best illustrated by the TrustBuilder [Winslett2002] and PeerTrust [Gavriloaie2004] [Nejdl2004] projects. Automated trust negotiation is a way of controlling access to a private resource over the web through a gradual process of trust building. In a typical instance of the protocol, requests for resource access generate counter-requests for credentials or other information, which in turn generate similar counter-requests. The process continues until a point of trust is reached or until failure occurs due to a conflict of privacy policies. Though trust negotiation was designed for the web, it can be adapted to the mobile and wireless context, though it would have to be augmented with secure discovery protocols. Through this process, resource access can be requested and obtained with minimum privacy loss for either party.

Zhu et al. [Zhu2005] outline a service discovery protocol for pervasive computing which preserves privacy without third party mediation. The service provider and client expose partial sensitive information in a progressive approach. The protocol terminates when both parties reach an agreement about the extent of exposure of the service and authentication information. Upon a mismatch or an unsatisfied request, the protocol can be terminated without loss of privacy. This protocol is meant to handle fake service providers as well as unauthorized clients. Since entities are assumed to share low-level security information, which is the basis on which they negotiate, the scalability of this approach is debatable. Still, protocols of this type provide novel ways to maintain security and access control constraints in a decentralized manner without sacrificing openness.

3.2.3 Cross-Domain Security Frameworks

In a utopian world, all devices, networks, and enterprise domains would be completely open to any other entity that wished to interact with them. This is not practical, since every device cannot and does not trust every other device in mobile environments. Certain device properties, such as identity and relationships, reflect the amount of confidence that different humans have in each other, and by implication, affect device interactions. With perfect trust in the other party and in the communication channel, the process of interaction and the mechanisms used for resource and data access cease to matter. In practice, perfect trust is not feasible, especially when interacting entities are mutually anonymous. For example, a user could take his laptop to his office and immediately obtain access to the local network, as well as a range of other resources, given his role as a trusted member of that organization. Apart from basic authentication mechanisms that allow his laptop to connect and be admitted to the network, and similar authentication by the laptop to verify the network access point, strict security is generally not required for discovering the available resources or accessing privileged information. If the authentication framework and the process for handing out authentication information are foolproof, this will work. If a device is compromised or the owner turns malicious, there are serious consequences. If we put aside the issue of trusted entities turned malicious, having an overarching trust framework could enable free interoperation among any set of devices and networks. Such trust-based security solutions are commonly in use within limited domains, but an enterprise-based framework does not scale globally, and bottom-up growth of infrastructure also poses an obstacle to deployment. Below, we examine solutions that help in assessment of trust and discuss their advantages and drawbacks.

Centralized, Monolithic Security

A globally centralized security solution is a potential approach. Currently, efforts are being made to deploy single-provider, city-wide 802.11 network connectivity in a variety of metropolitan areas [Google2005]. In theory, access to these services could be dependent on accepting a universal security policy. Every mobile device and network would be confident that all other entities would be constrained by that policy. This is conceptually a legitimate approach if it can be achieved at a worldwide scale, except for the fact that it would be undesirable to invest so much trust and power in one organization. This model creates a single point of failure which threatens user privacy as well as system reliability.

In the absence of a global security framework and policy, as well as an enforcement scheme, we need to devise frameworks for the dynamic establishment and assessment of trust in order to verify communication channels and enroll securely into foreign environments. These approaches are discussed below.

Certificate Hierarchies

The traditional distributed computing trust solution involves certificates. A certificate, in its simplest form, is a public key signed by certificate authorities. Gaining or verifying trust using certificates requires a hierarchy of certificate authorities. An ad hoc interaction could involve the presentation of a certificate; if the recipient shares a common parent with the certificate owner at some level in the hierarchy, a trust relationship can be established. Though this approach provides a certain degree of trust in mobile and ubiquitous computing, it has serious drawbacks which limit its use. First, given the bottom-up growth of ubicomp infrastructure, it is difficult to force everyone to accept one particular certificate hierarchy, and the higher up the common authority lies, the lower the value of trust becomes. Second, with a huge and unwieldy infrastructure, revocation and updates will be very inefficient. Third, this does not handle cases where strangers meet in a virtual bubble, possibly having no connection with a common trust authority. Last, and most important, certificates in their basic forms (or the way they are currently used in web transactions) are identity-based, and do not say anything more; every mobile device or network has different concerns and priorities, and simply verifying that a particular authority has certified the opposite party may not mean anything.

Peer-to-Peer Trust

Delegation has been proposed and used by various researchers to make the certificate distribution and verification scheme less strictly hierarchical and more suited to dynamic mobile environments. For example: entity A could delegate to entity B the right to issue certificates in A's name. Therefore, a delegated certificate issued by B could be trusted if A is a trusted source. This scheme has the property of creating chains and webs of trust [Zimmermann1994], which effectively form a peer-to-peer security framework that could be used as a basis for interaction. Though more dynamic, decentralized, and more resilient to network partitions, this kind of framework suffers from the same problems that afflict certificate hierarchies; it is difficult to assess the value of a credential issued by any particular peer. What makes the issuer of the credential trust a particular entity is not clear, especially if the distance along the chain between the certificate owner and the examiner is long. Clearly these delegated credentials need to provide more information than just identities. In this respect, we are building a voucher mechanism in which a voucher can be provided by one entity to another, certifying certain properties such as rights, group affiliation, and state. The use of a rights-delegating voucher is similar to SPKI [RFC2693].

Closely associated with webs and chains of trust is the notion of reputation, which in theory adds some more weight to the trust or confidence level in another party. Reputation is a way of assessing the trustworthiness of entities based on what other known and trusted entities say about them [Xiong2004]. If this were to work, it would be a strictly more reliable framework than one based on identity. Reputation models have not seen much success due to the impact of lying or colluding parties, and the huge number of variables involved in trust assessment [Sen2002]. Still, this is one way of establishing an overarching web of trust that could potentially cover most unplanned ubicomp interactions, and research in this area should be watched closely.

Role-based access control is a popular security framework adopted by open systems, where privileges are tied to a defined role. In its simplest form, this kind of access control works in the mobile context only if familiar entities interact. If strangers must interact securely, the system must be augmented by some process of role determination. Given a common credential vocabulary, a web of trust, and delegation permissions, privileges can be determined through a recursive process of proof-building, as demonstrated in the dynamic RBAC model [Freudenthal2002]. Combining role-based access control with delegation and trust chains has been employed in ubicomp middleware like Centaurus [Kagal2001a] and Vigil [Kagal2001b] [Kagal2002].

Quantitative Trust Models

Newer approaches have argued for a more dynamic notion of trust, and one that reproduces the way humans interact among themselves, such as the Secure project [English2002] [Cahill2003]. The dynamic nature of trust can be reproduced through the processes of trust formation and trust evolution, both of which use the history of past interactions in the trust evaluation functions. This project, as its basis, advocates making personal observations of an entity's behavior a part of the trust assessment function. A system for monitoring applications and reacting to events [English2004] is based on such dynamic trust models. This is a promising approach for managing dynamic environments, as it has the best potential for allowing secure interactions among strangers. Apart from identifying the important features of a trust framework, we need quantitative models to generate and make use of trust relationships. One approach could be a unified model that uses both identity and contextual properties and which expresses trust as a continuum [Shankar2002]. A different model attempts to model trust using probabilities, and in addition proposes ways to interpret the information during the actual process of performing a security-sensitive action [Jøsang1999].

We feel that dynamic trust models of the type discussed above hold great promise, and indeed are some of the few trust frameworks that scale to ubicomp environments. We cannot of course abandon identity and possession of certificates as a means of assessing trust; these are and will be key mechanisms for trust building. Therefore, research must concentrate on producing trust frameworks that make use of identity, properties, and observed results of actions. These kinds of trust frameworks also form the basis of automated peer negotiation, which was discussed earlier, and this is a promising research area that we are actively investigating.

4. Conclusion

We have discussed a wide spectrum of security and privacy issues that must be addressed before we can trust our devices to perform automated tasks on our behalf in a mobile context. Trustworthy and secure communication infrastructure is a prerequisite for secure mobile computing. Our own mobile devices and the other devices they interact with in the environment must have security and privacy solutions built in so that they can discover and access each other's resources even when connections are established in an ad hoc manner. In a ubiquitous computing world, usability is of primary importance, and security and privacy solutions must be designed in such a way that they preserve this property.

We have classified device-based solutions into three categories, roughly corresponding to three layers of defense for a mobile or infrastructure-based device interacting in dynamic circumstances with entities that may or may not be familiar. Each class of solutions has drawbacks if employed in isolation. Resource or content protection mechanisms employed without secure protocols for discovery and a trust basis either provides weak security (for interactions with strangers) or does not scale and would require some amount of manual configuration. Similarly, a secure negotiation protocol for sharing of resources without the enforcement mechanisms at the resource access level or a trust basis is not a comprehensive security solution. Trust frameworks without secure means of trust inference and enforcement at lower levels do not provide much value. A hybrid of the three classes of approaches is required for a scalable security solution, and for mobile devices to trust their surrounding environment and service providers when interactions are required.

We have also identified a number of promising approaches that address security and privacy challenges faced by mutually unknown entities interacting in an unplanned manner. We envision secure enrollment schemes growing in importance. More applications inevitably lead to more software vulnerabilities, and QED-like integrity analysis will be indispensable for halting the spread of malware. Some flavor of negotiation will inevitably come into play when interacting with strangers, since this promises to address the subtle balance required between security, privacy, and usability. Trust frameworks that are not purely identity-based are the weak point in today's research, and further investigation in this area would be very welcome.

We can assume that decentralized operation and numerous unplanned interactions will be predominant features of emerging ubiquitous computing systems. Dealing with unknown entities and unplanned events will pose numerous challenges. By limiting the risks of exposure and compromise at multiple levels, systems may remain secure, despite the dangerous and hostile intent of others. Taking lessons from the approaches discussed in this paper, future security framework designs must focus on risk minimization as a primary goal.

References

[Balfanz2002] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong, “Talking to Strangers: Authentication in Ad-Hoc Wireless Networks.” *NDSS 2004*.

[Balfanz2004] D. Balfanz, G. Durfee, R. Grinter, D. K. Smetters, and P. Stewart, “Network-in-a-Box: How to Set Up a Secure Wireless Network in Under a Minute,” *USENIX Security 2004*.

[Blaze1993] M. Blaze, “A cryptographic file system for UNIX,” *1st ACM Conference on Computer and Communications Security*, pages 9-16, November 1993.

[Blaze1998] M. Blaze, J. Feigenbaum, and M. Strauss, “Compliance Checking in the PolicyMaker Trust Management System,” *Proceedings of the Financial Cryptography Conference, Lecture Notes in Computer Science*, vol. 1465, pages 254-274, Springer, 1998.

[Blaze1999] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis, “The KeyNote Trust Management System Version 2,” *RFC 2704,* September 1999.

[Borisov2001] Nikita Borisov, Ian Goldberg, and David Wagner, “Intercepting Mobile Communications: the Insecurity of 802.11,” *Proceedings of the 7th annual International Conference on Mobile computing and networking*, pages 180-189, July 2001, Rome, Italy.

[Brooks1997] R. Brooks, “The Intelligent Room Project,” *Proceedings of the 2nd International Cognitive Technology Conference,* 1997, Aizu, Japan.

[Cahill2003] V. Cahill, E. Gray, J. Seigneur, C. D. Jensen, Y. Chen, B. Shand, N. Dimmock, A. Twigg, J. Bacon, C. English, W. Wagealla, S. Terzis, P. Nixon, G. di Marzo Serugendo, C. Bryce, M. Carbone, K. Krukow, and M. Nielsen, “Using Trust for Secure Collaboration in Uncertain Environments,” *IEEE Pervasive Computing*, vol. 02, no. 3, pages 52-61, July-September, 2003.

[Cisco2003] White paper—“Network Admission Control Executive Positioning Document,” http://www.cisco.com/en/US/netsol/ns466/networking_solutions_white_paper0900aecd800fdd66.shtml.

[Corner2002] M. Corner and B. Noble, “Zero-Interaction Authentication,” *Conference on Mobile Computing and Networking (MobiCom)*, September 2002.

[English2002] C. English, P. Nixon, S. Terzis, A. McGettrick, and H. Lowe, “Dynamic Trust Models for Ubiquitous Computing Environments,” *Proceedings of Workshop on Security in Ubiquitous Computing, Ubicomp* 2002.

[English2004] C. English, S. Terzis, and P. Nixon, “Towards Self-Protecting Ubiquitous Systems: Monitoring Trust-based Interactions,” *Journal of Personal and Ubiquitous Computing,* Volume 10, Issue 1, December 2005, pages 50-54.

[Eustice2003a] K. Eustice, L. Kleinrock, S. Markstrum, G. Popek, V. Ramakrishna, and P. Reiher, “Enabling Secure Ubiquitous Interactions,” *Proceedings of the 1st International Workshop on Middleware for Pervasive and Ad-Hoc Computing (in conjunction with Middleware 2003),* 17 June 2003, Rio de Janeiro, Brazil.

[Eustice2003b] K. Eustice, L. Kleinrock, S. Markstrum, G. Popek, V. Ramakrishna, and P. Reiher, “Securing WiFi Nomads: The Case for Quarantine, Examination, and Decontamination,” *Proceedings of the New Security Paradigms Workshop (NSPW) 2003*.

[Freudenthal2002] E. Freudenthal, T. Pesin, L. Port, E. Keenan, and V. Karamcheti, “dRBAC: Distributed Role-Based Access Control for Dynamic Coalition Environments,” *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02)*, IEEE Computer Society, July 2002.

[Gavriloaie2004] R. Gavriloaie, W. Nejdl, D. Olmedilla, K. Seamons, and M. Winslett, “No Registration Needed: How to Use Declarative Policies and Negotiation to Access Sensitive Resources on the Semantic Web,” *Proceedings of the 1st First European Semantic Web Symposium,* Heraklion, Greece, May 2004.

[Goodrich2005] M. Goodrich, M. Sirivianos, J. Solis, G. Tsudik, and E. Uzun, “Loud and Clear: Human-Verifiable Authentication Based on Audio,” *WISE 2005*.

[Google2005] V. Kopytoff and R. Kim, “Google offers S.F. Wi-Fi—for free / Company’s bid is one of many in response to mayor's call for universal online access,” http://www.sfgate.com/cgi-bin/article.cgi?file=/c/a/2005/10/01/MNGG9F16KG1.DTL.

[Jøsang1999] A. Jøsang, “Trust-Based Decision Making for Electronic Transactions,” *Proceedings of the Fourth Nordic Workshop on Secure IT Systems (NORDSEC'99)*, Stockholm, Sweden (Stockholm University Report, pages 99-105, 1999.)

[Kagal2001a] L. Kagal, V. Korolev, H. Chen, A. Joshi, and T. Finin, “Centaurus: A Framework for Intelligent Services in a Mobile Environment,” *21st International Conference on Distributed Computing Systems Workshops (ICDCSW '01),* April 16 - 19, 2001, Mesa, Arizona.

[Kagal2001b] L. Kagal, T. Finin, and A. Joshi, “Moving from Security to Distributed Trust in Ubiquitous Computing Environments”, *IEEE Computer*, December 2001.

[Kagal2002] L. Kagal, J. Undercoffer, F. Perich, A. Joshi, and T. Finin, “A Security Architecture Based on Trust Management for Pervasive Computing Systems,” *Proceedings of Grace Hopper Celebration of Women in Computing*, 2002.

[McCune2005] J. M. McCune, A. Perrig, and M. K. Reiter, “Seeing is Believing: Using Camera Phones for Human-Verifiable Authentication,” *IEEE Symposium on Security and Privacy*, 2005.

[Necula1997] G. Necula, “Proof-Carrying Code,” *Proceedings of the 24th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Langauges (POPL '97),* January 1997.

[Nejdl2004] W. Nejdl, D. Olmedilla, and M. Winslett, “PeerTrust: Automated Trust Negotiation for Peers on the Semantic Web,” *Secure Data Management 2004,* pages 118-132.

[RFC2693] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, “SPKI Certificate Theory.”

[Román2002] M. Román, C. Hess, R. Cerqueira, A. Ranganathan, R. Campbell, and K. Nahrstedt, “Gaia: A Middleware Infrastructure to Enable Active Spaces,” *IEEE Pervasive Computing*, pages 74-83, Oct-Dec 2002.

[Sen2002] S. Sen and N. Sajja, “Robustness of Reputation-Based Trust: Boolean Case,” *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: part 1*, July 15-19, 2002, Bologna, Italy.

[Shankar2002] N. Shankar and W. A. Arbaugh, “On Trust for Ubiquitous Computing,” *Invited paper in Workshop on Security for Ubiquitous Computing, UBICOMP*, October 2002.

[Stajano1999] F. Stajano and R. Anderson, “The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks,” *7th International Workshop on Security Protocols*, Cambridge UK, 1999.

[Waldo1999] J. Waldo, “The Jini Architecture for Network-Centric Computing,” *Communications of the ACM*, Vol. 42, No. 7, pages 76-82, 1999.

[Weiser1991] M. Weiser, “The Computer for the 21st Century,” *Scientific American* 265(30), pp. 94-104, 1991.

[Winslett2002] M. Winslett, T. Yu, K. E. Seamons, A. Hess, J. Jacobson, R. Jarvis, B. Smith, and L. Yu, “Negotiating Trust on the Web,” *IEEE Internet Computing*, Nov-Dec 2002.

[Winslett2003] M. Winslett, “An Introduction to Trust Negotiation,” *1st International Conference on Trust Management*, Crete, Greece, May 2003.

[Wright2003] C. P. Wright, M. Martino, and E. Zadok, “NCryptfs: A Secure and Convenient Cryptographic File System,” *Proceedings of the Annual USENIX Technical Conference*, pages 197-210, June 2003.

[Xiong2004] L. Xiong and L. Liu, “PeerTrust: Supporting Reputation-Based Trust in Peer-to-Peer Electronic Communities,” *IEEE Transactions on Knowledge and Data Engineering (TKDE), Special Issue on Peer-to-Peer Based Data Management*, 2004.

[Zimmermann1994] P. Zimmermann, “PGP User’s Guide,” *MIT*, October 1994.

[Zhu2005] F. Zhu, W. Zhu, M. W. Mutka, and L. M. Ni, “Expose or Not? A Progressive Exposure Approach for Service Discovery in Pervasive Computing Environments,” *PerCom 2005*, pages 225-234.