Personal PKI for the smart device era.

LYLE, J., PAVERD, A., KING-LACROIX, J., ATZENI, A., VIRJI, H., FLÉCHAIS, I. and FAILY, S.

2013

Personal PKI for the smart device era

John Lyle[1], Andrew Paverd[1], Justin King-Lacroix[1], Andrea Atzeni[2], Habib Virji[3], Ivan Flechais[1], and Shamal Faily[1]

[1] Department of Computer Science, University of Oxford, UK
first.last@cs.ox.ac.uk
[2] Dip. di Automatica e Informatica, Politecnico di Torino, Torino, Italy
andrea.atzeni@polito.it
[3] Samsung Electronics Research Institute, Ashford, Surrey, UK
habib.virji@samsung.com

Abstract. As people use an increasing number of smart devices for their everyday computing, it is surprising that these powerful, internet-enabled devices are rarely connected together to create *personal networks*. The *webinos* project is an attempt to make this possible so that resources can easily be shared between devices, regardless of the operating system or network they are using. However, increased connectivity raises a number of security and privacy issues, and in this paper we introduce a public key infrastructure designed to be suitable for *personal* computing across multiple devices. We recognize the need for our PKI to work on both mobile and home networks, use existing online user identities and take into consideration the different interaction styles found on smart devices in different form factors. We propose a set of principles for *personal* key infrastructures, describe our implementation and outline how it mitigates common threats and issues.

1 Introduction

We have slowly seen a move away from personal *computers* towards personal *computing*: while PCs are still widely used, nowadays people tend also to have a multitude of other form factors, such as tablets, smartphones, set-top boxes and in-car computers. Likewise, personal computing now involves multiple devices, both private and shared, as well as a growing number of personal internet services.

Home and *personal area* networks have been researched and developed to manage the personal usage of multiple computers [1,2,3,4]. These typically focus on creating secure networks of devices that are co-located either physically or logically, such as on a home wireless network. This has simplified administration and the sharing of resources, however the *home* networking concept has been slow to adapt to mobile devices and online services. For example, a home network tends to use only one medium, such as WiFi, and therefore cannot encompass mobile devices which often connect from remote locations via mobile data. Therefore, we argue for the creation of *personal networks* [5] rather than home networks to better suit the needs of personal computing.

Lacking a suitable personal network, users frequently rely on multiple, disparate cloud services and applications for interoperability; these are accessible from any device with an internet connection, regardless of geographic location or network medium [6]. However, this increases reliance on cloud storage, web applications and internet connectivity, as well as a myriad of different policies, usage patterns and trust relationships. A personal network, which we define as the set of all devices and resources used by a particular person or group, could provide a viable alternative which doesn't rely on multiple third party cloud providers, and provides a consistent model for interoperability and security.

Security and privacy issues are rife in personal networking. Smart devices store data with both personal and monetary value. The impact of device theft or compromise is substantial, and the increasing number and mobility of personal devices only makes it more likely. The trend towards cloud services does not necessarily help: web browsers may store passwords, and are thus pre-authenticated to these cloud services. Furthermore, cloud-based data may be cached locally for efficiency and connectivity reasons, leaving the user open to numerous attacks. The rise of mobile applications has also highlighted privacy issues, particularly in personal and home contexts. Access to location, camera, and other sensors can result in both accidental and deliberate privacy violations. Recent high-profile breaches have garnered concern from users regarding privacy issues, and the problem is gaining traction in the research community [7,8].

Access control can solve many security and privacy problems, in particular when combined with strong authentication of users and devices, which could be provided through a public key infrastructure. However, no suitable PKI exists: the web PKI model is adequate for authenticating websites, but inappropriate for identifying individual devices and users. Home PKIs (such as UPnP Device Protection [9]) might work for authenticating devices, but cannot interoperate with those on mobile networks. User-based PKIs suffer from usability problems [10], and fail to leverage user identities established by social networks and websites. This motivates the need for a suitable PKI for *personal networks* capable of protecting user security and privacy.

In this paper we take a fresh look at PKI for personal networks, based on these observations as well as the state of the art in home networking. In section 2 we discuss related work on PKI in a personal and home context and use these and our own experiences in section 3 to elicit the key threats to personal networks. We then give design principles for a personal PKI in section 4, proposing the use of internet protocols, web technologies and existing authentication and discovery mechanisms. Section 5 presents the *webinos* PKI system, a realization of these principles, and describes security features including certificate management, network discovery and revocation. In section 6 we evaluate *webinos* and its initial implementation against the identified threats, discuss how a personal PKI could be migrated and compare with an alternative capability-based approach. Finally, in section 7 we conclude.

2 Related work

ITU-T Recommendation X.1112 [11] presents two models for home network device certification. In one model, all home device certificates are issued by an external CA essentially resulting in a global PKI. In the other model, an internal CA is used to generate keys and certificates for home devices. A secure home gateway can be used for this purpose; this can have a device certificate issued by an external CA to facilitate authentication between the home network and external service providers. Kinkelin et al. [1] make a similar distinction between the two models; they suggest that trust relationships can be built between internal CAs based on the social relationships between home networks owners. We extend this concept to devices on mobile networks, and propose a certificate exchange process based on online identities.

The recently standardized *UPnP Device Protection* (UPnP-DP) service [9] uses PKI in a home network context. This service allows a device to restrict access to specific UPnP operations or resources. In UPnP-DP, device identity is provided by means of self-signed X.509 certificates on each device. Before accessing restricted services or resources, devices must first be paired with each other. As explained by Baugher and Lortz [12], device pairing uses the same mechanisms as WiFi Protected Setup (WPS) and avoids exposing the underlying PKI to the user. In order to enforce the principle of least privilege, UPnP-DP uses three roles: *Public*, *Basic*, and *Admin*. The *webinos* system described in section 5 uses a similar combination of X.509 certificates and device pairing but defines slightly different levels of privilege and certificate hierarchies.

The use of public key infrastructures in personal area networks (PANs) has been investigated previously as part of the SHAMAN project [4] and in related work by Mitchell and Schaffelhofer [3]. To adapt conventional PKI for use in a PAN, one of the devices in the PAN acts as a 'Personal CA', issuing certificates to other devices. The final SHAMAN technical reports discuss the operation of the personal CA as well as mechanisms for device initialization, proof of possession of key pairs and revocation in PAN PKIs. The limited availability of the personal CA and the risk of this device being lost or stolen have been identified as potential issues. The proposed mitigation involves the use of multiple CAs, but requires additional synchronization steps. The alternative proposed in this paper involves making the CA a cloud-based web service.

Ford et al. [13] propose the 'Unmanaged Internet Architecture' (UIA) in order to provide connectivity between mobile devices independent of network location. All devices are given a *personal name* and belong in a *personal group*. Devices in the same group can access one another. No central infrastructure is required, and users only deal with convenient names rather than device addresses. Parts of this approach are a generalization of the *webinos personal zone* model discussed in section 5. However, we suspect that UIA would rapidly end up resembling a *webinos*-like network due to the prevalence of NAT and mobile networks, forcing at least one device to become a permanently-addressable router. Furthermore, user authentication is not fully addressed, and the proposed approach does not take advantage of existing user identities.

Kubota and Miyake [14] suggest that secure DNS domains can be created for personal networks through DNSSEC. In their scheme, the hash of the domain public key is itself a pseudo domain name that anyone can verify. This has the advantage of requiring no change to client applications, only changes to the DNS resolver mechanism. Secure collaborations between domains can be created purely through knowledge of the pseudo domain name. However, challenges such as discovery of the pseudo domain name and key management issues such as revocation remain unresolved.

3 Personal network definition and threats

We define a personal network as a set of communicating devices belonging to (or are used by) a particular individual. Personal networks may include devices in different form factors and operating on different physical networks. Both inter- and intra-network communication are supported, so that each device *within* the personal network can communicate with others, but also devices *outside* the personal network can connect in order to facilitate sharing of data or services between users. Personal networks may include online services and even virtual cloud-based devices. Finally, personal networks may also provide synchronization of data and settings in order to maintain a seamless user experience between devices.

Home and personal networks of devices face a number of threats to user security and privacy. The key assets requiring protection in a personal network are the devices themselves, each of which may hold user credentials and may have access to valuable services such as pay-per-view television. The network may also provide internet connectivity through potentially expensive mobile networks. Personal data, such as user location history and browsing preferences, must be protected for privacy reasons. Personal devices may be used to access work data and so attacks may put valuable intellectual property at risk.

The following five threat categories have been identified. We use data from existing analyses [15,12,16] as well as threats discovered as part of the *webinos* project [17] which are relevant to home or personal scenarios. We ignore issues commonly dealt with at lower layers in the communication stack, such as communication jamming, as well as potential weaknesses in standard cryptography. We also focus primarily on the end user as a stakeholder, rather than third party software or service providers.

Unauthorized physical access to personal networks. This includes the theft of an authorized device, or the misuse of a device residing in a shared context. Vulnerabilities may be due to weak or stolen authentication credentials (e.g. an easily-guessed or shoulder-surfed password). Shoulder-surfing can also result in disclosure of other private data, particularly for shared devices such as a communal television or tablet PC. Sharing a physical network (such as home WiFi) can implicitly authorize access to a home router or administration console, which facilitates attacks such as adding rogue devices to the

trusted network. Baugher and Lortz [12] suggest that three tiers of access are therefore required: guest, user, and administrator.

Unauthorized remote access to personal networks. This may be the result of weak credentials on a personal device accessible from the internet, or the compromise of a cloud-based network service. Weak access control policies or flawed protocols might allow an attacker to send requests to a personal network device over the internet and extract valuable data. For instance, an attacker may use port scanning to find a personal network device which exposes the local file system through a well-known API, or a vulnerable remote login service may be publicly-exposed.

Malicious software. Malware may collect private data for sale or identity theft, or misuse access to valuable resources such as online bank accounts. It may misuse a device's internet access by participating in botnets for distributed denial of service attacks. As well as its impact on the attack's target, this might affect the device's performance and cost money if bandwidth is expensive. Malware can be invoked through some action on the part of the user – perhaps a phishing attempt, or after an attack on an app store [18] – or through a runtime exploit of the web browser or operating system.

Intercepting and modifying network communication. Insecure networks could allow session hijacking, misuse of user authentication at a remote service or on the device, or simply eavesdropping on any unencrypted traffic. While some network link types may mitigate these attacks, the number of these (WiFi, Bluetooth, mobile data, and so on) means such measures are not always assured. Furthermore, the security perimeter of the personal network likely differs from that of the network link currently in use.

Misuse of device interoperability. The interoperation of multiple devices creates a new class of threat [19]. Security is a weakest-link problem: the weakest personal device can potentially be used as a gateway to the rest of the network. Furthermore, any synchronization of data or settings between devices can be used as an attack vector; if any device can be made to poison the synchronized data, it could plant attacks against vulnerabilities in another. Personal networks may also result in greater replication of data, increasing the impact of any compromise.

4 Principles for *Personal* Key Infrastructures

Existing work on personal area networks [4] and personal certificate authorities has not taken into account the variety of smart devices or online identities. In this section we propose a set of design principles for a personal network.

Leverage existing identities. People already have identities on the web: their social networks, email accounts, and homepages. These should be reused in public key infrastructures. We suggest that a mapping from a social network identity to a public key or certificate should be created, and that users should

be able to find each other through this web-based identity. This takes advantage of existing relationships and therefore avoids the discovery and bootstrapping problems often associated with PKI [20]. Furthermore, as these identities are web-based, they do not rely on the user having a particular piece of hardware with them at any time.

Assume devices are mobile. Tablets, smartphones, laptops, and cars are all designed to be mobile. This significantly increases the risk of a device being lost or stolen. As a result, revocation must be primarily concerned with removing a lost and potentially rogue device from the personal network, and must also not rely on the user having another enrolled device to hand.

Avoid using PKI metaphors. End users should not be expected to understand PKI terminology. This implies that all keys and certificates should be generated automatically, and there should never be a prompt or question asked to users referring to these things. Instead, friendly names and existing identities should be used. This is in line with suggestions by Balfanz et al. [20], and we agree with Ford et al. [13] that a combination of user and device identities should be used.

Use web technologies to make networks interoperable. Many existing home PKIs expect to operate within a single local area network, and so do not scale to cover devices which may be on mobile networks with frequently changing IP addresses. The interoperability of web applications – which provide a common, accessible web server for communication – should be re-used to make personal networks available to any device capable of making outgoing connections to web servers.

Delegate key storage to operating systems. The best way to protect private keys is likely to be device-specific. For example, some devices support secure hardware which may provide a high level of protection. Furthermore, devices in different contexts will have different authentication requirements: e.g., a shared PC might only unlock private keys after authenticating the end user, whereas a mobile device may be assumed to belong to one person only. Similarly, each platform has different application security infrastructures, so protection from malware is hard to achieve in a truly cross-platform manner. We suggest, therefore, that key storage be delegated to the operating system, which already has to deal with many of these issues.

Device keys are not always a factor of user authentication. It is tempting to treat a device-held private key as a factor of user authentication. However, this ignores the fact that personal devices are designed to be mobile. A device key should be used to identify the device only, and only as a second factor when the device is appropriate: e.g., a laptop or mobile phone with a single user and a login prompt. We propose that another layer of authorization should be used on top of the key infrastructure. This is in contrast to Balfanz et al.'s 'instant PKI' which identifies that certificates could be treated like capabilities [20].

5 The *webinos* PKI

In this section we describe the *webinos* application platform which implements a novel form of personal network called a *personal zone*. Personal zones define the devices used by (and often belonging to) a particular person and have a public key infrastructure for device authentication. Personal zones have a master device called a *personal zone hub* which acts as a certificate authority and can be implemented as an online web service.

webinos is a cross-platform runtime environment for web applications. Extending the capabilities provided by a web browser, it provides a set of standard JavaScript APIs for web applications to use in order to access local device functionality – such as a camera or address book – and to communicate with other devices both in and out of the users' personal zone. For example, a web application running on a PC would be able to use *webinos* APIs to access the camera on a smartphone. *webinos* also provides an access control policy system for mediating access to these APIs in order to protect user security and privacy.

In the following subsections we describe the architecture of *webinos* and how it implements a personal network PKI system by following the requirements outlined in the previous section.

5.1 Components and communication

webinos consists of several software components on multiple devices, as shown in Table 1. Each device has a web runtime and a personal zone proxy (PZP) running on it. Web applications are displayed and executed in a web runtime (such as a browser) and communicate with the local personal zone proxy (PZP), which implements APIs as well as communicating with the personal zone hub (PZH). The hub is a web-based service which passes messages between proxies, synchronizes access control policies and settings, and provides administrative functions. PZPs can communicate with each other and with the PZH over mutually-authenticated TLS sessions. The PZH acts as a certificate authority, issuing certificates to each proxy.

Component	Key features and capabilities
Personal Zone Hub (PZH)	Web-based, constantly available and addressable, routes messages, acts as a certificate authority.
Personal Zone Proxy (PZP)	Runs on a device and implements APIs. Responsible for policy enforcement and communicating with the WRT, PZH and other PZPs.
Web Runtime (WRT)	User interface to web applications.

Table 1. Personal zone components

5.2 Certificate hierarchy

The personal zone hub is the certificate authority for a personal zone. The hub issues a certificate for itself as well as certificates for all proxies in the zone. These certificates are used to create mutually-authenticated TLS sessions. The hub also has a web interface, which uses a separate certificate. The CA certificate can be self-signed, or signed by the service provider who owns the infrastructure the hub is running on. In the case that one service provider hosts multiple hubs, the web interface may be authenticated through a different certificate owned by the service provider. If a user has a reliable home internet connection, then the hub can optionally be hosted by a home router or server. In this case, either the CA certificate would be self-signed or signed by the manufacturer. An example certificate hierarchy is shown in Figure 1.

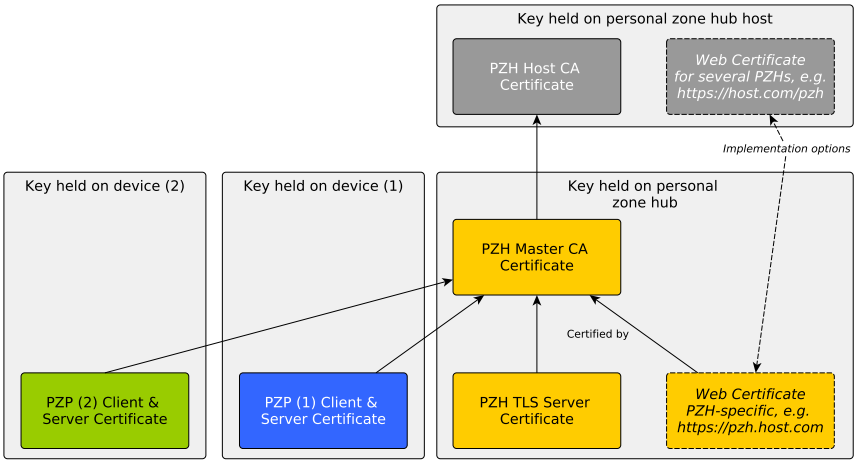

Fig. 1. Certificate hierarchy in *webinos*

5.3 User discovery

The discovery process in *webinos* links an existing personal identity - such as an OpenID account - to a personal zone hub address. We suggest that the WebFinger protocol be employed [21]. This process requires an identity provider to host an eXtensible Resource Descriptor [22] (XRD) describing *how* to look up user identity information, and then an XRD information record per user describing their links to other resources. This information should be hosted and served by a trustworthy party over a secure connection. A record for the user 'Joe Smith' would contain an entry such as:

```
<Link rel='http://webinos.org/spec/1.0'
      ref='http://phz.example.org/profile/joe.smith'/>
```

5.4 Certificate exchange

CA certificates must be exchanged when two devices in different personal zones communicate for the first time. The following two scenarios describe how certificate exchange occurs when users are in proximity (e.g. connecting over the same WiFi network) or when users are connecting remotely over the internet.

Peer-to-peer offline exchange. Exchanging certificates between two users who are in physical proximity is the device pairing problem [23]. Typical solutions involve users visually comparing a short code to authenticate a secure channel created between their devices. We intend to do the same using the SHCBK key agreement protocol [24] with either a *word-matching and number-typing* or *repeated numeric comparison* scheme [25] for code comparison.

Online exchange. We imagine a scenario where Alice wants to access one of Bob's device APIs over the internet, but neither of them have connected to each other before. We assume that Alice and Bob know each other's email or social network identity, but are not in physical proximity.

The first step is discovery. Alice either already knows Bob's personal zone hub URL (he may have shared it through email or his address book entry) or she uses an identifier for Bob (such as his email address) to discover it. She visits Bob's PZH URL and requests access to his resources. The request redirects to Alice's PZH, which then sends a signed request to Bob's PZH containing her CA certificate. Bob's certificate is automatically added to her zone's list of known users. Because Alice had discovered Bob based on an already-known identity or URL, we assume that no further identity assertion from Bob is required. However, Bob does not know who Alice is, and needs to connect the request she makes with her identity as claimed by a mutually-trusted social network. There are two options. Either Bob's personal zone hub insists that Alice must identify herself, using her OpenID credentials to 'log in', or Bob performs a discovery of Alice's hub URL through the discovery process based on her claimed identity. Bob's PZP then downloads Alice's certificates and he approves or rejects her request to access resources. The discovery approach has the advantage of symmetry, and involves no additional work by the personal zone hub. However, it depends on the popularity and security of the discovery process.

5.5 Enrolment

There are several ways in which new devices can be enrolled into the personal zone, with the following currently implemented: The process starts with the *webinos* PZP software being installed on the new device. The user then visits the hub's website and logs in with his or her OpenID credentials. Having logged

in, the user selects 'add new device'. This tells the hub to temporarily accept unauthenticated TLS connections. The website also presents the user with a short string (also encoded as a QR tag) which is used as an authentication token. This token is entered into the new PZP, along with the URL of the hub. The PZP then attempts to connect to the hub and is challenged to present a valid token. After a fixed number of attempts, the hub either issues the PZP with a signed certificate or rejects the connection.

For future work, we would like to allow a trusted web application to enrol the user's device after initial OpenID login. This would simplify the process, but requires further modification to the web runtime to allow a privileged action to take place. Another improvement would be for the hub to temporarily support password-authenticated key exchange (PAKE) protocol for new PZP, such as TLS-SRP [26], using the authentication token as a password. This would remove the need to accept unauthenticated connections for enrolment.

5.6 Revocation of personal zone devices

Within a personal zone, revocation is implemented through a synchronized certificate revocation list (CRL) issued by the personal zone hub. This allows the user to selectively revoke a device by visiting the hub's web interface and clicking a 'revoke' button. The CRL is synchronized between devices whenever they connect to the hub. The choice of CRLs rather than the Online Certificate Status Protocol (OCSP) was motivated by a desire for simplicity, as well as the fact that CRLs for a zone should be small, and that hubs already implement synchronization for policy enforcement.

Devices which remain offline for significant periods run the risk of having an out-of-date CRL. A stolen device or compromised key could successfully connect in a peer-to-peer manner with such a device. There are several ways to mitigate this threat although a perfect solution is impossible, as offline, peer-to-peer communication between devices is a requirement of *webinos*. However, a method analogous to OCSP stapling [27] could be used. Each device could be required to present an up-to-date CRL for the zone with every new connection. These CRLs could be issued by the hub on a well-known schedule, such as daily or weekly. This would limit the window for a malicious device. An alternative would be to require any device which has not connected to the hub for a certain time period to reconnect briefly when a peer-to-peer connection is made. Neither of these solutions is likely to be easily understood by users, and we therefore do not address this threat at present. For most people who have devices stolen purely for their intrinsic monetary value rather than as part of a bigger attack, this seems a reasonable trade-off.

Between personal zones, revocation can be implemented by personal zone hubs sharing CRLs regularly. This should be reasonable as CRLs ought to remain small. Revocation of personal zone hubs and migration from one service provider to another is discussed in section 6.2.

5.7 Access control and authentication

When an API access request is made, the proxy will first check to see whether the application, device and user are authorized to make the request. This is implemented through an XACML policy architecture described in [28]. Importantly, access control policies are synchronized between devices so that general user-based policies such as 'Bob may access Alice's location' are possible, as well as specific device and application-based policies such as 'Application X on device D can access Alice's camera on device E'.

The *webinos* PKI does not assume that device identities automatically map to individual user identities. For some devices this will be the case, but for other devices a further level of authentication is required. This means that a policy applying to Bob might apply to access requests originating from Bob's smartphone, but from Bob's TV it would require Bob to re-authenticate. User authentication is implemented through Bob logging in with his OpenID account to the current device (in the case of local access control) or to the personal zone hub (in the case of remote access control). User authentication is *always* required for privileged actions such as modifying policies and adding or removing personal devices from the zone.

5.8 Key storage

Both TLS keys and the hub's CA signing key require storing and protecting. On personal zone proxies, the private key is stored using OS-provided mechanisms. In particular, we have implemented support for the Mac OS X Keychain and Gnome Keyring. This delegates the task of releasing TLS keys to the operating system (or a trusted platform-specific application) which ought to be configured to understand the context in which the device is used. For example, it might require a username and password for some shared devices, or simply unlock keys as soon as the machine has been logged into for the first time. CA keys are more sensitive – the potential for misuse is greater – and as such must always be protected by the personal zone hub service provider. Use of the key requires authentication using OpenID credentials, and the key itself should never be accessible to any user. This makes it a good candidate for protection using secure, tamper-proof hardware [1].

5.9 Backup and recovery

The *webinos* personal zone system does not require an explicit recovery system. The OpenID provider will manage loss of passwords or user credentials. For PZPs, in the instance that a key is compromised or lost the device can be revoked and then re-added. This requires UI modification but no new underlying functionality. With this in mind, all access control policies and references to devices in *webinos* should be to a *friendly name* rather than public key identity or certificate serial number. This means that any key can be mapped onto a particular device name, making the re-issue of keys for a device straightforward.

An advantage of this approach is that keys do not need to be backed-up onto another device, reducing maintenance overhead.

6 Evaluation and discussion

We have built an open source implementation [4] of *webinos* for several platforms including Windows, Linux, Mac, Android and Pandaboard. The prototype is not complete, but provides a proof-of-concept for the underlying PKI. Although more analysis is needed, our current experience suggest that it successfully provides mutual authentication for the combination of users and devices in personal networks. In the rest of this section we evaluate the *webinos* PKI with respect to the threats identified in section 3 and then discuss two issues: how to migrate a personal zone hub to another service provider, and whether the proposed PKI compares favourably to a more distributed design using capabilities.

6.1 Mitigating identified threats

Unauthorized *physical* access to the personal network is an impossible threat to mitigate purely through software. However, the *webinos* PKI supports revocation when a device is lost, and the access control policies limit the authority of any one device. Shared personal devices are not assumed to have the full authority of the end user, limiting the impact of this potential vulnerability. Indeed, further user authentication for privileged actions is required. Because all functionality is implemented at the application layer, war driving attacks are not possible unless other vulnerabilities exist. We do not implement the three tiers of access defined by Baugher and Lortz, but instead identify users and devices independently.

Unauthorized *remote* access is only possible if someone steals a copy of a device key, or knows the OpenID credentials of the user. All access to the personal zone depends on either one of these things. As the OpenID authentication method is dependent on the identity provider, we assume this can be configured to be as secure as necessary. Because this authentication process will be well known to users, it should also limit the likelihood of credential loss and need for recovery. Theft of a device key is a problem, and might be achieved through either physical theft or malware. We rely on the local operating system's key storage to protect private keys from both of these attacks. In essence, we rely entirely on existing mechanisms to solve this problem.

Malware is a key threat to *webinos* considering that it is an application platform. We control all applications within *webinos* through a policy framework described in previous work [28]. The policy framework also potentially limits the impact of one malicious device on the rest of the zone. The design of *webinos* separates private keys and the proxy software from the web browser using OS process isolation, which reduces the potential for exploit of the browser. Of course, malware is still a problem, and future work will consider how *attestation* [29] could be employed to eliminate the threat of rootkits.

[4] `https://developer.webinos.org`

The use of mutually-authenticated TLS sessions between all components mitigates the threat of interception or modification of network communication. However, the interoperability of devices does create new attacks. We believe that by implementing the same security controls on all platforms, we raise the bar of the weakest device and therefore mitigate at least some of these threats.

6.2 Migration of personal zone hubs

Users may wish to move their personal zone hub to a different service provider for a number of reasons, such as cost or quality of service. They may also be concerned that their hub provider has been compromised or is behaving maliciously. One of the goals of *webinos* is to enhance user control over personal data, so it is necessary to define how a personal zone hub can be revoked and migrated. To our knowledge, relatively little work exists on the migration of personal PKI infrastructures. In this subsection we discuss how this may be achieved in terms of revocation and certificate management.

One potential scenario is that Alice would like to move her personal zone hub from provider P to provider Q but neither are assumed to be fundamentally untrustworthy or insecure. As such, the main challenge is to move her hub data from P to Q with minimal disruption to her devices and any other users who connect to her personal zone. Availability is assumed not to be as important as consistency, as Alice's personal devices are designed to deal with temporary loss of connectivity anyway. The process for migrating in this scenario is a relatively straightforward export and import of data, followed by updating the user's WebFinger record. It is also likely that keys generated and stored at P will need to be recreated at Q, as either the user or Q may not trust them. To do this, existing PZPs would need to make sure the change was authorized by Alice, import the new PZH keys and have new certificates issued to it. The new PZH would need to temporarily accept PZPs with old certificates before re-issuing new ones. Finally, all old certificates would need to be added to the new PZH's CRL. External users would update their records by first failing to connect to P and then re-discovering the user's new address at Q through WebFinger. We note that much of this process can be simplified if the personal zone hub is running on its own virtual machine.

A different solution is required if the old host is not assumed to be trustworthy. Users must assume that their PZH keys were compromised, and that the service provider could impersonate them by accepting requests from remote parties. Solving this problem requires disconnecting and updating each personal zone device in turn as well as notifying all external contacts to re-discover the personal zone hub URL from the user's WebFinger record. Because this process may be slow and error-prone, we suggest that users should regularly re-discover their contacts.

During our risk analysis, we identified migration as a process potentially vulnerable to attacks in both scenarios. It is necessary to protect the migration process such that data import and export can only be performed by authorized users, migration cannot be used as a denial of service against a user, and the

updating of certificates does not allow a rogue device to join the personal zone. Finally, much of the security of the personal zone hub comes down to reliance on the OpenID provider and WebFinger account. We leave protecting these as future work, but note that OpenID is flexible in it's own authentication methods and can support multiple factors to enhance security.

6.3 Comparison with a capabilities-based approach

The *webinos* PKI relies heavily on the personal zone hub to identify and authenticate users and devices. An alternative, distributed approach would be to use capability-based security [30]. In such a system, each security principal – which could be a device, user, or application – can possess its own 'keyring' of *capabilities*, each of which represents access to a single resource. These can then be delegated to other parties in a controlled manner. For example, a smartphone application could have access to the local camera API, and the laptop's secure storage facility, but not the laptop's webcam; it would possess capabilities to the first two resources, but not the third. This removes the need for a central identity authority, PKI infrastructure and policy-based access control.

Capabilities, therefore, simplify and reduce the trust required in the personal zone hub, which would only need to provide routing and communication. In addition, performance-intensive XML policy processing is no longer required on each device, and no synchronization of policies is necessary. Peer-to-peer operation is more flexible: since capability issuance decisions are inherently made locally, rights to a feature on one device can easily be issued to another, within or outside the personal zone, removing the need for certificate exchange. Similarly, delegation and revocation of rights is straightforward. Such a model would also fit with the use of web token-based authentication approaches such as OAuth.

However, there are some disadvantages. In particular, the existence of a central security authority is convenient for auditing privileged actions, such as adding devices to the zone. In contrast, a capability-based model for *webinos* would need to include auditing of every capability issuance event to a global log. Furthermore, as communicating with devices on mobile networks requires a central access point, such as a personal zone hub, for routing and communication purposes anyway, introducing capabilities can never completely eliminate centralization. Other issues with auditing delegation and creating secure channels would also need to be solved.

7 Conclusion

We have introduced a PKI infrastructure to support *personal networks* based on the new requirements and usage patterns created by the popularity of smart devices and applications. We have developed an initial implementation on multiple platforms, including Windows, Android and Linux as part of the *webinos* project. Our system helps mitigate many identified threats of home and personal networks, and we have defined methods for discovery, certificate exchange,

enrolment, revocation and key storage. *webinos* does not require any additional usernames or passwords, and almost exclusively uses existing protocols. We have provided a theoretical comparison with an alternative access control approach and evaluated our system against identified threats.

8 Acknowledgements

The research described in this paper was funded by EU FP7 *webinos* Project (FP7-ICT-2009-5 Objective 1.2).

References

1. Kinkelin, H., Holz, R., Niedermayer, H., Mittelberger, S., Carle, G.: On Using TPM for Secure Identities in Future Home Networks. Future Internet **3**(1) (January 2011) 1–13
2. Müller, A., Kinkelin, H., Ghai, S.K., Carle, G.: An assisted device registration and service access system for future home networks. In: 2nd IFIP Wireless Days, IEEE (December 2009) 1–5
3. Mitchell, C.J., Schaffelhofer, R.: Chapter 3 - The Personal PKI. In: Security for Mobility. Institution of Engineering and Technology (2004) 35–61
4. SHAMAN Project: Deliverable 13, work package 3. `http://www.isrc.rhul.ac.uk/shaman/docs/d13a3v1.pdf` (November 2002)
5. Niemegeers, I., Heemstra de Groot, S.: From Personal Area Networks to Personal Networks: A User Oriented Approach. Wireless Personal Communications **22** (2002) 175–186
6. Jehangir, A., Heemstra de Groot, S.M.: Securing Personal Network Clusters. In: Proceedings of the third international conference on security and privacy in communication networks. SecureComm (2007) 320 –329
7. Egele, M., Kruegel, C., Kirda, E., Vigna, G.: PiOS: Detecting Privacy Leaks in iOS Applications. In: Proceedings of the 18th Annual Network and Distributed System Security Symposium,. NDSS, The Internet Society (February 2011)
8. Hornyack, P., Han, S., Jung, J., Schechter, S., Wetherall, D.: These aren't the droids you're looking for: retrofitting android to protect data from imperious applications. In: Proceedings of the 18th ACM conference on Computer and communications security. CCS '11, ACM (2011) 639–652
9. UPnP Forum: UPnP Device Protection Service. Technical report, UPnP Forum (2011)
10. Whitten, A., Tygar, J.D.: Why Johnny can't encrypt: a usability evaluation of pgp 5.0. In: Proceedings of the 8th USENIX Security Symposium. SSYM'99, Berkeley, CA, USA, USENIX Association (1999) 14–14
11. International Telecommunication Union: ITU-T Recommendation X.1112 — Device certificate profile for the home network. Technical report, ITU (2007)
12. Baugher, M., Lortz, V.: Home-Network Threats and Access Controls. In: Trust and Trustworthy Computing. Volume 6740 of LNCS. Springer Berlin / Heidelberg (2011) 217–230
13. Ford, B., Strauss, J., Lesniewski-Laas, C., Rhea, S., Kaashoek, F., Morris, R.: Persistent Personal Names for Globally Connected Mobile Devices. In: Proceedings of the 7th symposium on Operating systems design and implementation. OSDI '06, Berkeley, CA, USA, USENIX Association (2006) 233–248

14. Kubota, A., Miyake, Y.: Autonomous DNSSEC: Secured Pseudo DNS Domains for Personal Networks. In: GLOBECOM Workshops (GC Wkshps), 2010 IEEE. (dec. 2010) 1576 –1580
15. International Telecommunication Union: ITU-T Recommendation X.1111 — Framework of security technologies for home network. Technical report, ITU (2007)
16. International Telecommunication Union: ITU-T Recommendation X.1121 — Framework of security technologies for mobile end-to-end data communications. Technical report, ITU (2004)
17. The webinos consortium: User expectations of security and privacy phase 2. `http://webinos.org/blog/2011/11/01/webinos-repot-user-expectations-of-security-and-privacy-phase-2/` (September 2011)
18. Chia, P.H., Yamamoto, Y., Asokan, N.: Is this App Safe? A Large Scale Study on Application Permissions and Risk Signals. In: In proceedings of WWW2012: The World Wide Web Conference. (April 2012)
19. Lyle, J., Faily, S., Flechais, I., Paul, A., Goker, A., Myrhaug, H., Desruelle, H., Martin, A.: On the design and development of webinos: a distributed mobile application middleware. In: Proceedings of the 12th IFIP International Conference on Distributed Applications and Interoperable Systems. DAIS (2012)
20. Balfanz, D., Durfee, G., Smetters, D.: Making the Impossible Easy: Usable PKI. In: Security and Usability: Designing Secure Systems that People Can Use. O'Reilly, Sebastopol, CA (2005) 319–334
21. Jones, P.E., Salgueiro, G., Smarr, J.: WebFinger: IETF Network Working Group Internet Draft. `http://tools.ietf.org/html/draft-jones-appsawg-webfinger-04` (May 2012)
22. OASIS: Extensible resource descriptor (xrd) version 1.0. `http://docs.oasis-open.org/xri/xrd/v1.0/xrd-1.0.html` (November 2010)
23. Saxena, N., Ekberg, J.E., Kostiainen, K., Asokan, N.: Secure device pairing based on a visual channel. In: IEEE Symposium on Security and Privacy. (may 2006) 6 pp. –313
24. Nguyen, L.H., Roscoe, A.W.: Authenticating ad hoc networks by comparison of short digests. Information and Computation (an international journal) **206, Issues 2-4** (Feb-Apr 2008) 250–271
25. Kainda, R., Flechais, I., Roscoe, A.W.: Secure and usable out-of-band channels for ad hoc mobile device interactions. In: Proceedings of the 4th IFIP WG 11.2 international conference on Information Security Theory and Practices. WISTP'10, Springer (2010) 308–315
26. Taylor, D., Wu, T., Mavrogiannopoulos, N., Perrin, T.: Using the Secure Remote Password (SRP) Protocol for TLS Authentication. RFC 5054 (Informational) (November 2007)
27. Eastlake 3rd, D.: Transport Layer Security (TLS) Extensions: Extension Definitions. RFC 6066 (Proposed Standard) (January 2011)
28. Lyle, J., Monteleone, S., Faily, S., Patti, D., Ricciato, F.: Cross-platform access control for mobile web applications. In: Proceedings of the IEEE International Symposium on Policies for Distributed Systems & Networks, IEEE (July 2012)
29. Coker et al.: Attestation: Evidence and Trust. In: Proceedings of the 10th International Conference on Information and Communications Security. Volume 5308 of LNCS., Springer (2008) 1–18
30. Levy, H.M.: Capability-Based Computer Systems. Butterworth-Heinemann, Newton, MA, USA (1984)