Practical Security for Rural Internet Kiosks

Sumair Ur Rahman, Urs Hengartner, Usman Ismail and S. Keshav
David R. Cheriton School of Computer Science,
University of Waterloo, Waterloo ON N2L 3G1, Canada
{surrahman, uhengart, uismail, keshav}@cs.uwaterloo.ca

ABSTRACT

Rural Internet kiosks typically provide weak security guarantees and therefore cannot support secure web access or transaction-oriented applications such as banking and bill payment. We present a practical, unobtrusive and easy-to-use security architecture for rural Internet kiosks that uses a combination of physical and cryptographic mechanisms to protect user data and kiosk infrastructure. Our contributions include (a) a detailed threat analysis of rural Internet kiosks, (b) a security architecture for rural Internet kiosks that does not require any specialized hardware features in kiosks, and (c) an application-independent and backward-compatible security API for securely sending and receiving data between kiosks and the Internet that can operate over disconnection-tolerant links.

Categories and Subject Descriptors

C.2.0 [**General**]: Security and protection; C.2.1 [**Network Architecture and Design**]: Store and forward networks, Wire-less communication

General Terms

Design, Measurement, Security

1. INTRODUCTION

Internet kiosks are being deployed in developing regions around the world [13] providing low-cost access to the Internet. Our kiosk deployment model consists of one or more recycled commodity PCs that connect to the Internet either over a long-range wireless link (e.g., WiMAX) or over a purpose-built DTN (Disconnection-tolerant Network) [5, 10], as shown in Figure 1. (We describe the roles of Kiosk Controllers and Proxy Servers in Section 2.) Currently, most such kiosks provide weak security and therefore cannot support secure applications such as banking. For example, there is nothing to stop a kiosk administrator from installing malicious software or accessing user data stored on the kiosk.

NSDR'08, August 18, 2008, Seattle, Washington, USA.

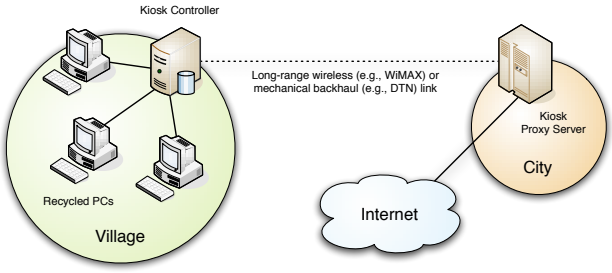

Figure 1: Rural Internet Kiosk

Securing rural Internet kiosks is more challenging than securing privately-owned PCs because users of these kiosks do not own the hardware that they use and cannot always trust kiosk owners/operators. In addition, the potential use of DTN links in remote regions to connect kiosks to the Internet precludes the use of traditional session-oriented technologies such as SSL to build secure applications for kiosks.

Previous work on securing public computing infrastructure like Internet kiosks has focused on the use of smartphones to establish trust in a computing platform [4] or to offload the processing of sensitive information [2, 8, 11], and the use of lightweight cryptosystems such as Hierarchical Identity-based Encryption (HIBE) to protect kiosk user data [1, 7, 9]. Unfortunately, challenges posed by rural kiosks, specifically (a) the absence of specialized hardware features such as Trusted Platform Modules (TPMs) [15] or a modifiable BIOS in older recycled PCs, (b) the potential use of DTN, (c) the absence of a production-ready implementation of HIBE and (d) the limited availability of smartphones in most developing regions make these approaches difficult, if not impossible, to implement in our scenario.

We propose a practical, unobtrusive and easy-to-use security architecture that uses a combination of physical and cryptographic mechanisms to protect user data and kiosk infrastructure. Our contributions include (a) a detailed threat analysis of rural Internet kiosks, (b) a security architecture for rural Internet kiosks that builds on off-the-shelf open-source software and does not require any specialized hardware features in kiosks or the use of mobile devices, and (c) an application-independent and backward-compatible security API for securely sending and receiving data which, unlike SSL, can operate over DTN links.

Due to space limitations, this paper provides a high-level overview of our security architecture. A sketch of our scheme

appeared in ICTD2007 [5]. This earlier paper does not include the detailed threat analysis, the security API for sending and receiving data, and a security analysis describing how our system addresses the identified threats. Further details of our implementation, a discussion of the issues we encountered and an evaluation of the system's performance can be found in the extended version of this paper [16]. Source code for our implementation is available on our website [14].

The rest of this paper is organized as follows. In Section 2, we present our system model, followed by a threat analysis in Section 3. In Section 4, we present our security architecture and in Section 5 we analyse the effectiveness of this scheme. We review related work in Section 6 and discuss potential directions for future work in Section 7.

2. SYSTEM MODEL

In this Section, we introduce our deployment model for rural Internet kiosks, identify the entities that operate, support or use these kiosks and outline a typical usage scenario.

2.1 Overview

Each rural Internet kiosk provides service to users in a specific geographic region and is independently administered by a *Franchiser* (see Section 2.2). We illustrate a typical deployment scenario in Figure 1. Components of a typical rural kiosk include:

- *Kiosk Controllers* – servers deployed at rural kiosks that have wired connections to terminals (see below), providing them with network boot, a network file system, user management, and Internet connectivity through kiosk proxy servers (also see below) via long-range wireless (e.g., WiMAX) or a DTN.
- *Kiosk Terminals* – inexpensive recycled PCs capable of running Linux that allow users to access the Internet through a wired connection to a single kiosk controller.
- *Kiosk Proxy Servers* – servers deployed at data centers that serve as a proxy between kiosk controllers and legacy servers on the Internet.

We define a *Kiosk* as a set of one or more *Terminals* and a single *Kiosk Controller*. For the purposes of this paper, Kiosk proxy servers and controllers are termed *Infrastructure Components*.

2.2 Entities

The following entities have an interest in the correct and reliable operation of a rural kiosk:

- *Franchisers* – franchisers are public or private organizations that own, operate and administer kiosk infrastructure components deployed in a particular geographic area.
- *Franchisees* – franchisees are private organizations or individuals licensed by a franchiser to operate terminals connected to a kiosk controller provided by the local franchiser.
- *Application Service Providers (ASPs)* – application service providers are public or private organizations (e.g., micro finance banks) that are licensed by franchisers to deploy their applications on a rural Internet kiosk.
- *Users* – users subscribe to kiosk services with a franchiser, usually through a franchisee, to access services, such as applications provided by ASPs, and the Internet using terminals owned/operated by franchisees.

Franchisers control all infrastructure components. This effectively creates a "closed universe" where franchisers have control over all franchisees and registered users in their region, as well as user data and the software running on terminals. We make use of this organizational structure in Section 4 when proposing the use of a public key infrastructure (PKI) in our security architecture.

2.3 Usage Scenario: Selling Produce

In the usage scenario below, we describe how Kiran, a rural Internet kiosk user, uses a kiosk in her village to negotiate the sale of her farm's produce using a hypothetical application, *SellProduce*, deployed at the kiosk by a produce wholesaler registered as an ASP.

1. Kiran logs into a recycled PC at her local kiosk, launches SellProduce and specifies the produce she has for sale along with her asking price. She then hits the submit button to send her offer to the wholesaler. SellProduce generates an offer message and transmits this to the kiosk controller.
2. The kiosk controller forwards the offer message to a kiosk proxy server which then forwards the message to a kiosk used by the wholesaler. (This may be connected to the proxy directly via the Internet.)
3. The produce wholesaler reviews Kiran's offer in SellProduce and responds with another offer message detailing desired purchase price, quantity and timeframe. When the wholesaler submits this offer through SellProduce, the message is sent to the kiosk proxy server.
4. The kiosk proxy server forwards the wholesaler's offer message to Kiran's kiosk controller, which stores it for her to review and respond the next time she logs in. Kiran would then either respond to further negotiate her terms of sale or confirm the sale as described above.

In the absence of security mechanisms that guarantee the authenticity, integrity and privacy of the offer messages exchanged between Kiran and the wholesaler, it would be difficult for either side to trust and act on the messages that they receive. As mentioned earlier, the potential use of a DTN link between kiosks and the kiosk proxy server prevents the use of traditional, session-oriented technologies such as SSL to secure kiosk applications such as SellProduce. Other applications with similar security requirements include rural banking, government record and telemedicine.

3. THREAT ANALYSIS

In this Section, we identify potential attackers, describe their capabilities, and list key attacks against rural kiosks.

3.1 Potential Attackers

In addition to outsiders, defined as being none of the entities introduced in Section 2.2, we assume attackers to be either franchisees or users. Franchisers do not appear in our list of attackers because as operators of the system, its correct and reliable functioning is in their best interest. We exclude ASPs under the premise that any software, data and configuration changes by these entities will be inspected and approved by the appropriate franchiser. Attackers may possess one or more of the following:

Wireless Communication Channel – the ability to eavesdrop on, inject messages into or jam the wireless communication channel between infrastructure components, given sufficient physical proximity to these systems.

Physical Access – unfettered physical access to infrastructure components in the absence of authorized franchiser personnel, without knowledge of their administrator passwords.

Technical Expertise – the experience/expertise required to modify the software and/or configuration of a Linux-based system, given network-based or physical access.

3.2 Recognized Threats

Threats against rural kiosks can be categorized as attacks against the confidentiality, integrity or availability of the system. In terms of confidentiality, we are concerned with the privacy of user data and any secret keys stored in their accounts (see Section 4.2 for details). For integrity, we are concerned with the integrity of this data plus the integrity of infrastructure components, recycled PCs used at kiosks and the impersonation of franchiser personnel and kiosk users. For availability, we consider the jamming of wireless links between infrastructure components. (Recycled PCs are connected to kiosk controllers by a wired link.) When combined with potential attackers, these threats give us the grid in Figure 2 below. We also classify each attack by its likelihood.

The classification of likely and unlikely threat-attacker combinations in Figure 2 is based on the capabilities of a particular attacker, the cost of mounting a particular attack, and the potential benefits. For example, a franchiser would be more likely to attempt to modify the configuration of a kiosk controller in order to disable its wireless interface than set up a jamming signal to achieve the same result, given the cost of setting up the jamming signal and the simplicity of disabling the device's wireless interface.

Threat-attacker combinations that are marked as ignored appear as such because either the cost of mounting the attack exceeds the benefit to the attacker or because a lower-cost attack that achieves the same result is available.

4. SECURITY ARCHITECTURE

In this Section, we highlight our security goals with respect to the concerned entities introduced in Section 2.2 and then describe how our scheme protects users, the recycled PCs they use and all infrastructure components.

4.1 Security Goals

Our overall security goals are to provide the best possible security for users, operators and infrastructure components given the need to minimize costs, the limited processing capabilities of infrastructure components and the recycled PCs used as terminals, as well as the absence of specialized hardware in recycled PCs, such as TPMs or a modifiable BIOS.

Specific security goals, in terms of the four entities that use or operate rural kiosks, are as follows: (a) franchisers are concerned with the security of their infrastructure and want to detect, if not prevent, the misuse of their infrastructure components, (b) franchisees are concerned with the security of their kiosks and want protection against the spread of viruses over and any attacks launched against their kiosks, (c) depending on the type of service that they provide, ASPs may want franchisers to guarantee the integrity of their software when deployed on rural kiosks, where examples of such software include tax payment and land registry systems operated by the government, and (d) users are concerned with the confidentiality and integrity of their data. We describe the mechanisms used to achieve these goals below.

4.2 User and Operator Security

Entities that use or operate rural kiosks each possess a unique set of *Entity Credentials*. Entity Credentials consist of an RSA key pair and a corresponding X.509 certificate that binds the holder's identity to the public part of its key pair. Rural kiosk users and operators obtain and use their Entity Credentials as described below.

Franchisers self-generate an RSA key pair and then use the public part of this key pair to obtain a certificate signed by a trusted CA such as VeriSign or Thawte. This key pair is then used to sign certificates issued to franchiser administrative personnel, licensed franchisees, and ASPs.

Franchisees obtain certificates in a similar fashion to franchisers, with the only difference being their certificates are signed by their franchiser. Franchisees use their keys to sign certificates issued to users registered at their kiosks.

ASPs obtain certificates from the local franchiser in an identical fashion to franchisees. They use their Entity Credentials to authenticate software deployed at kiosks on their behalf by franchisers and any subsequent updates to this software and to secure the transfer of data between ASPs and users, if necessary.

Users obtain their credentials when they register at a rural kiosk. Their certificates are signed by the local franchisee. The usage of certificates and key pairs is transparent to users, which is important because previous research has shown that users cannot be expected to manually deal with certificates [6]. Namely, a key pair and a certificate are automatically created upon registration and stored in the user's encrypted home directory (see Section 4.4.2). Furthermore, usage of the keys is simplified through the *Secure Directory API* (see Section 4.4.1), where incoming data is transparently decrypted and verified, and outgoing data is transparently encrypted and signed without user intervention.

Certificates for users are made available to other kiosk users, franchisers, ASPs, other franchisees and the Internet by means of a kiosk user database known as the *White Pages*. This database is maintained by each region's franchiser and updates to it are periodically sent out to all kiosks and licensed ASPs. The database is the only place that is consulted by a kiosk upon receipt of a signed message. Any certificate that no longer shows up in the database is considered revoked, which eliminates the need for a separate certificate revocation mechanism. For a user base of 10,000 with each certificate requiring about 2KB of storage, the entire White Pages database would be around 20MB in size.

All certificates described above are chained to a trusted root CA's certificate (e.g., VeriSign or Thawte) such that trusting this certificate alone is sufficient to verify the above entities' certificates. This way, an ASP can, but does not have to, delegate identity verification to a franchisee or even a franchiser, which can be important in rural environments.

4.3 Infrastructure Security

We now briefly describe how infrastructure components are protected against attacks using a combination of cryptographic and physical security mechanisms.

For physical security, we assume that kiosk controllers are equipped with sealed, tamper-evident enclosures. These

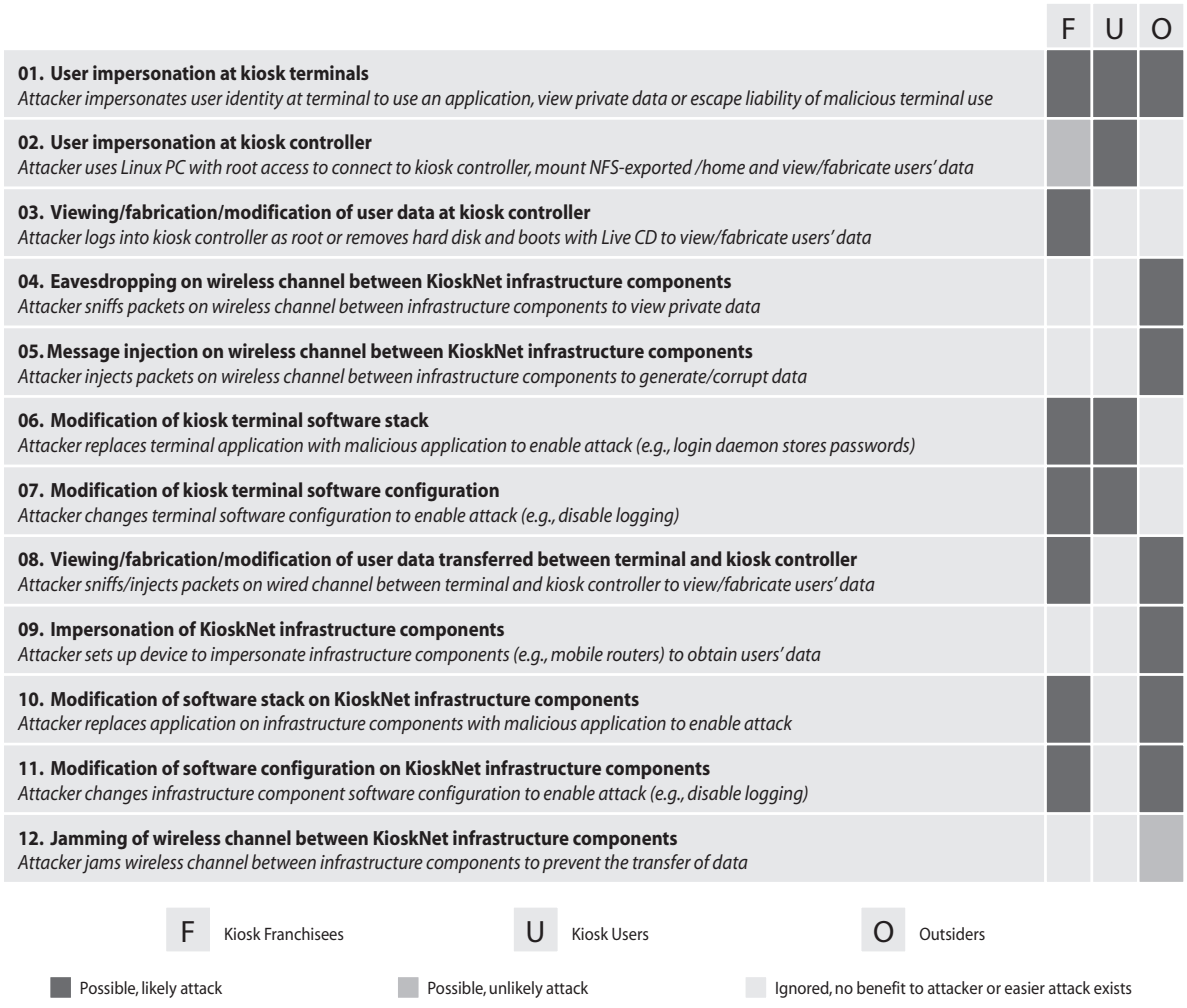

Figure 2(a): Recognized threats against rural Internet kiosks vs. potential attackers

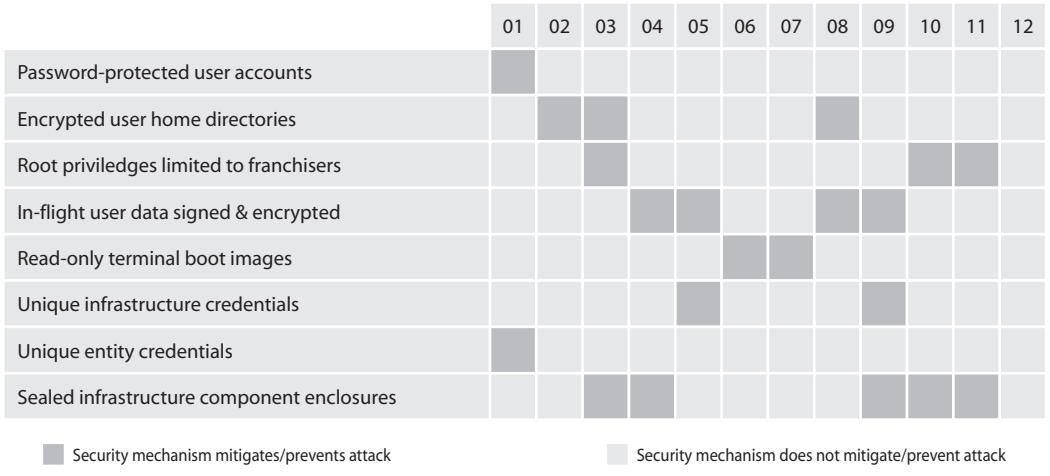

Figure 2(b): Security mechanisms vs. recognized threats

Figure 2: Threats against rural Internet kiosks and proposed security mechanisms.

enclosures would most likely utilize proprietary screws and locks, in addition to sticker seals over removable enclosure panels, similar to those used by vendors of commercial electronics to detect attempts to open the devices. The other physical security mechanism we rely on is the regular inspection of deployed infrastructure components by franchiser field technicians to check for tampering or damage.

In terms of cryptographic security, all infrastructure components are issued unique credentials called *Infrastructure Credentials* by the franchisers that operate them. Similar to Entity Credentials, Infrastructure Credentials consist of a key pair generated by the device and a certificate signed by the franchiser that binds the infrastructure component's identity to its public key. A device uses its credentials to authenticate to other infrastructure components, and to secure communication between infrastructure components. Administrator privileges on infrastructure components are limited to authorized franchiser personnel.

4.4 Terminal Security

Terminals are PCs, typically recycled, that network boot from a read-only image stored in a kiosk controller. These images contain the Linux kernel used by terminals, configuration files and applications. Because terminals may be disk-less, user data is stored in the kiosk controller. All applications launched by the user are run on the terminal for better performance. To prevent an attacker from impersonating a user, every user is assigned a password during registration and has to enter this password into the terminal when logging in. This password is also used for protecting a user's data, which we describe in the following subsections.

4.4.1 Secure Directory API

The *Secure Directory API* allows developers to easily produce applications that communicate securely between kiosks and the Internet. Applications can simply write outgoing data to a user's `~/application/supload` directory and read incoming data from `~/application/sdownload`, where `~/` corresponds to the mount point on the terminal for the user's encrypted home directory (see Section 4.4.2).

A daemon running on the terminal automatically decrypts and verifies signatures on incoming data using the user's private key (stored in his/her encrypted home directory) and other users' public keys available in the White Pages database (introduced in Section 4.2) and places it in the `sdownload` directory. Similarly, this daemon automatically encrypts and signs all outgoing data placed in `supload` after looking up the recipient's public key in the White Pages database. We note that data destined for a server reachable over the Internet is either encrypted on a specific ASP's public key or on the proxy's public key and then sent in plaintext over the Internet by the proxy.

4.4.2 Encrypted User Home Directories

All user data, such as a user's pictures and emails, are stored in kiosk controllers and exported over NFS for access via terminals connected to a kiosk controller. As highlighted in Section 3.2, this makes it possible for an attacker to connect to the kiosk controller using a Linux PC with administrator access to override filesystem permissions and access the NFS-exported user data or, in a more extreme scenario, to break into the kiosk controller, remove its hard disk, and boot it in a PC with a Live CD to achieve the same.

To protect user data stored in kiosk controllers, users' home directories are created in encrypted virtual volumes. Users' virtual volumes are exported in their encrypted form to terminals over NFS for automatic mounting and decryption when users logs in. The process is reversed when users log out. Our implementation is based on off-the-shelf, open-source software (see the extended version of this paper [16].)

In the event a user forgets his/her password, an encrypted backup copy of the key used to encrypt the user's virtual volume is maintained by the franchiser. This key can be used to decrypt the user's virtual volume, allowing the user to select a new password and then encrypt it again.

4.5 Security Architecture Usage Scenario

We now briefly describe how our security architecture works in the usage case outlined in Section 2.3. We assume the produce wholesaler Kiran wishes to sell her produce to is registered as an ASP with her local kiosk. When Kiran hits submit in SellProduce, the application writes her offer message to her home directory in `~/sellproduce/supload`. The Secure Directory API daemon then signs the message with Kiran's private key stored in her encrypted home directory, attaches her public key certificate and encrypts the message for the wholesaler, by looking up their public key in the White Pages database. The Secure Directory API daemon sends the secure offer message to the kiosk controller for forwarding to the kiosk proxy server. (We note that the link between the kiosk controller and proxy server may either be DTN or long-range wireless.) The proxy server then forwards the message to the wholesaler's kiosk controller. A Secure Directory API daemon running on the wholesaler's computer (this could be a special "terminal" only used by the wholesaler) decrypts the offer message, validates Kiran's public key certificate, verifies the signature on her message and then passes the validated offer message up to SellProduce running on the wholesaler's computer. Offer messages sent by the wholesaler to Kiran would be secured in the same fashion and stored in her encrypted home directory.

5. SECURITY ANALYSIS

In this Section we present a brief analysis of the security mechanisms used to protect rural Internet kiosks. Figure 2 above summarizes the security mechanisms we propose and shows how these are combined to guard against the attacks that were presented earlier in Section 3.

5.1 Users

To prevent users from impersonating other users, each user is assigned a password. The operating system running on the terminal ensures that a user enters this password before granting him/her access. For additional security, a terminal is shut down when a user logs out, killing any processes that the user might have left behind and that could use up CPU or memory resources. The next user must then boot the terminal himself/herself, making it harder for franchisees to launch the phishing attack for users' login passwords described in Section 5.2 below.

5.2 Franchisee

Attacks against terminals and kiosk controllers by franchisees would likely involve tampering with the devices' software stacks and credentials or the impersonation of kiosk terminals to launch phishing attacks for login passwords.

Physical security mechanisms, as described in Section 4.3, prevent franchisees from tampering with the software and data stored in kiosk controllers (e.g., adding malicious terminal software, replacing the certificate identifying a franchiser, or extracting the kiosk controller's private key).

Preventing franchisees from setting up fake login screens on kiosk terminals to obtain users' login passwords is a more challenging problem. The only robust solution is shutting down a kiosk terminal when a user logs out and training users to boot the kiosk terminal before logging in. Currently proposed techniques for verifying the integrity of public computing platforms such as kiosk terminals [4] require the use of trusted mobile computing devices and the presence of specialized hardware in kiosk terminals (e.g., TPMs), assumptions which will likely not be reasonable in developing regions for quite some time. The use of a scheme such as that proposed by Surie et al. [12], whereby users boot terminals from a trusted USB memory stick before logging in as a means of verifying the integrity of the terminal's software, is precluded by the use of recycled PCs as terminals. (Older PCs are typically unable to boot from a USB device.)

5.3 Outsiders

We also need to defend against attacks on a kiosk by outsiders. A terminal is connected to its controller by a wired connection, which makes interception or man-in-the-middle attacks by an outsider difficult. If these attacks are a concern, the boot process could be extended such that a terminal authenticates the controller and downloads the kernel image over a secure connection. However, current off-the-shelf network boot software does not support this feature.

6. RELATED WORK

Previous research has studied how users can access remote services via an untrusted proxy without revealing sensitive information, such as private keys or passwords, to the proxy. In some approaches [4, 12], a user queries the proxy about its state and verifies that this state is trustworthy. These approaches require that the user has a trusted device to execute a query. Other approaches [2, 8, 11] offload the processing of sensitive information from the untrusted proxy to a user's trusted device. These approaches are not applicable to our scenario because kiosk users may not have trusted devices.

There are several security architectures for delay-tolerant networks. Some architectures [1, 7, 9], two of them developed by co-authors of this paper, are based on IBE. Unfortunately, the only production-ready implementation of IBE is proprietary, and licensing fees to its vendor, Voltage Security, Inc., make its use in a low-cost setting unrealistic. We also note that this implementation does not support Hierarchical IBE, as required by Kate et al. [7] and Seth and Keshav [9], and that it is no longer available for public download as of May 2008. Instead, we opted for a PKI-based approach, building on open-source implementations of well-known cryptosystems such as RSA and AES. The DTN research group also favors a PKI-based approach [3]. The group explicitly considers key management an open issue, whereas we present a solution. Furthermore, the group concentrates on the secure exchange of messages between DTN nodes and the required format specifications, whereas our main concern is at the application and usability level; that is, how users can send and receive secure messages.

7. CONCLUSION AND FUTURE WORK

In this paper we have presented a threat analysis for rural Internet kiosks, identified security goals for the system given the needs of a diverse group of stake holders, and proposed a practical, unobtrusive security architecture that meets these requirements. We have also addressed all the challenges to securing these kiosks identified earlier in Section 1. Implementation details, a discussion of the issues we encountered and a performance evaluation of our scheme can be found in the extended version of this paper [16].

Potential directions for future work include the use of smartcards and biometric authentication systems to provide end users with simplified, password-free access to kiosks and support for the secure roaming of users between kiosks.

8. REFERENCES

[1] N. Asokan, K. Kostianinen, P. Ginzboorg, J. Ott, and C. Luo. Towards Securing Disruption-Tolerant Networking. Technical Report NRC-TR-2007-007, Nokia Research Center, March 2007.

[2] D. Clarke, B. Gassend, T. Kotwal, M. Burnside, M. van Dijk, S. Devadas, and R. Rivest. The Untrusted Computer Problem and Camera-Based Authentication. In *Proc. of Int'l Conference on Pervasive Computing (Pervasive 2002)*, pages 114–124, August 2002.

[3] S. Farrell, S. Symington, H. Weiss, and P. Lovell. Delay-Tolerant Networking Security Overview - draft-irtf-dtnrg-sec-overview-03. Internet Draft, July 2007.

[4] S. Garriss, R. Cáceres, S. Berger, R. Sailer, L. van Doorn, and Z. Zhang. Towards Trustworthy Kiosk Computing. In *Proc. of 8th IEEE Workshop on Mobile Computing Systems and Applications (HotMobile'07)*, pages 41–45, February 2007.

[5] S. Guo, M. H. Falaki, E. A. Oliver, S. Ur Rahman, A. Seth, M. A. Zaharia, U. Ismail, and S. Keshav. Design and Implementation of the KioskNet System. In *Proc. of IEEE/ACM International Conference on Information and Communication Technologies and Development (ICTD2007)*, December 2007.

[6] P. Gutmann. Plug-and-Play PKI: A PKI your Mother can Use. In *Proc. of 12th USENIX Security Symposium*, pages 45–58, August 2003.

[7] A. Kate, G. Zaverucha, and U. Hengartner. Anonymity and Security in Delay Tolerant Networks. In *Proc. of 3rd Int'l Conference on Security and Privacy in Communication Networks (SecureComm 2007)*, September 2007.

[8] A. Oprea, D. Balfanz, G. Durfee, and D. K. Smetters. Securing a Remote Terminal Application with a Mobile Trusted Device. In *Proc. of 20th Annual Computer Security Applications Conference (ACSAC 2004)*, pages 438–447, December 2004.

[9] A. Seth and S. Keshav. Practical Security for Disconnected Nodes. In *Proc. of 1st Workshop on Secure Network Protocols (NPSec 2005)*, pages 31–36, 2005.

[10] A. Seth, D. Kroeker, M. Zaharia, S. Guo, and S. Keshav. Low-cost Communication for Rural Internet Kiosks Using Mechanical Backhaul. In *Proc. of 12th Int'l Conference on Mobile Computing and Networking (MOBICOM 2006)*, pages 334–345, September 2006.

[11] R. Sharp, J. Scott, and A. R. Beresford. Secure Mobile Computing via Public Terminals. In *Proc. of 4th Int'l Conference on Pervasive Computing (Pervasive 2006)*, pages 238–253, May 2006.

[12] A. Surie, A. Perrig, M. Satyanarayanan, and D. J. Farber. Rapid Trust Establishment for Pervasive Personal Computing. *IEEE Pervasive Computing*, 6(4):24–30, October-December 2007.

[13] Telecentre.org. `http://www.telecentre.org`. Accessed May 2008.

[14] Tetherless Computing Group. `http://blizzard.cs.uwaterloo.ca/tetherless`. Accessed May 2008.

[15] Trusted Computing Group. `https://www.trustedcomputinggroup.org`. Accessed May 2008.

[16] S. Ur Rahman, U. Hengartner, U. Ismail, and S. Keshav. Securing KioskNet: A Systems Approach. Technical Report CS-2007-43, David R. Cheriton School of Computer Science, University of Waterloo, November 2007.