EagleVision: A Pervasive Mobile Device Protection System

Ka Yang, Nalin Subramanian, Daji Qiao, and Wensheng Zhang
Iowa State University, Ames, Iowa - 50011

***Abstract*— Mobile devices like laptops, iPhones and PDAs are highly susceptible to theft in public places like airport terminal, library and cafe. Moreover, the exposure of sensitive data stored in the mobile device could be more damaging than the loss of device itself. In this work, we propose and implement a pervasive mobile device protection system, named EagleVision, based on sensing and wireless networking technologies. Comparing with existing solutions, EagleVision is unique in providing an integrated protection to both device and data. It is a context-aware system which adjusts the protection level to the mobile device dynamically according to the context information such as the user proximity to the mobile device, which is collected via the interactions between the sensors carried by the user, embedded with the mobile device and deployed in the surrounding environment. Furthermore, it does not require explicit user intervention to utilize the system and hence avoid adding extra distractions to the user. Prototype implementation and extensive field test results demonstrate the effectiveness of EagleVision.**

I. Introduction

Mobile devices, such as laptops, smart phones and PDAs, have become an essential part of our daily life. They are small and easy to carry but also powerful in computational and storage capabilities. Unfortunately, these merits also put them at risk. For example, because mobile devices are small, they usually are highly susceptible to theft, especially at public places like airport terminal, library and cafe. Recently, as mobile devices get slimmer and more powerful, the number of mobile device thefts surges. According to the FBI's National Crime Information Center, the number of reported laptop thefts in 2008 rose with a 48 percent increase over the previous two years, from 73,700 to almost 109,000 [1].

On the other hand, keeping data secure in a mobile device is not just a daunting challenge, but a critical requirement. Unfortunately, a majority of the mobile device users do not take necessary actions to protect the data stored in their mobile devices. Therefore, the loss of a mobile device could mean the loss and exposure of sensitive information stored in the lost device, which may be much more valuable than the device itself. According to CNN [2], a laptop theft case in 2006 related to Veterans Affairs Department resulted in the exposure of millions of veterans' personally identifiable information and costed the department 20 million dollars to settle the lawsuit.

Currently, there are a few mobile device or data protection products available in the market [3]–[8]. Unfortunately, each of them has its own limitations, which will be discussed in detail in Section II. To address the limitations of existing mobile device protection systems, in this paper, we propose a pervasive mobile device protection system, named EagleVision, with the help from sensing and wireless networking technologies. With EagleVision, we deploy low-cost wireless devices at public places of our interest to form a wireless network infrastructure. Users and mobile devices carry special-purpose wireless sensing devices which work with the wireless network infrastructure to provide protection to the mobile device and the data stored in it. Specifically, EagleVision has the following unique features:

- *Context Awareness:* Sensors carried by the user and the mobile device interact with each other as well as with the wireless network infrastructure to collect context information (e.g., proximity of the user to the mobile device) and then the system adapts its behavior properly and promptly to the context change.
  - *Anti-theft Protection for Mobile Device*: When the user is away from the mobile device, system monitors the mobile device. When a potential theft is detected, system quickly alerts the user as well as the central authority.
  - *Data Protection*: System adapts the protection level for data stored in the mobile device and incorporates a carefully-designed authentication mechanism to eliminate possible security attacks.
- *Transparency:* System adapts its behavior autonomously without requiring explicit user intervention or causing extra distractions to the user.
- *Low-cost and Light-weight:* System utilizes low-cost sensors and networking devices. The software implementation is light-weight and may be adapted for mobile devices of various kinds.

In this paper, we present a prototype of EagleVision using sensor motes as the wireless and sensing devices for both users and mobile devices. In practice, the wireless and sensing devices can also be implemented using Wi-Fi devices, like cell phone with Wi-Fi capability for user and Wi-Fi access point for wireless network infrastructure.

The rest of the paper is organized as follows. Section II briefly reviews the related work. Section III gives a system overview and Section IV presents the design details of EagleVision. Section V analyzes the security properties of the system. Section VI describes the prototype implementation and Section VII presents the performance evaluation and field validation results. Section VIII discusses some related issues of the system. Section IX concludes the paper.

II. Related Work

A. Mobile Device Protection

Various protection systems have been designed and implemented for different types of mobile devices. In general, they can be classified into the following two categories: *recovery/tracking-oriented systems* and *prevention-oriented systems*. Most of the well-known mobile device protection systems are recovery/tracking-oriented, such as Absolute Software's ComputraceComplete [3], Lojack for laptop [4] and Gadget for PDA [5]. In these systems, a back-end software process runs on the device, which can send "help" messages across the Internet to the tracking service provider in case the device is lost or stolen. The service provider can pinpoint the location of the lost device based on the "help" messages. Some systems also provide GPS options to help track the lost devices [6].

In general, recovery/tracking-oriented systems are ineffective in preventing mobile device thefts since they aim at recovering the devices after theft. In addition, most of the recovery/tracking-oriented systems have the following issues: (i) they may be disabled easily by either forcing the running

process to terminate or replacing the hardware; (ii) they rely on the Internet or the cellular network to track the lost devices, which means that the adversary may bypass the protection system by keeping the devices off-line; (iii) the systems which rely on GPS may fail if the lost devices are kept indoor.

In comparison, prevention-oriented systems aim at deterring the adversary from compromising the mobile devices, such as Kensington security lock and permanent bar code tags. Some products, e.g., Caveo's Anti-Theft PCMCIA card [7] and Musatcha for IBM laptops [8], utilize motion sensors to detect potential thefts. When a potential theft is detected, the system raises an audible alarm to deter the adversary from completing the theft. Nevertheless, laptop locks and permanent tags cannot stop a determined and skilled thief. Though motion sensor-based systems may be more effective in deterring the adversary, they require the users to start and stop the service manually. This may be an unpleasant distraction to the user. For example, a user working with a laptop in a library may walk around often to check reference books. In this case, the user needs to start or stop the service every time when the user leaves or returns to the laptop, respectively. Frequent explicit user interventions required for using a system may discourage users from using the system and thus limit its practical applications.

B. Data Protection

Data protection systems also can be classified into two categories: *cryptographic file systems* and *remote security systems*. Cryptographic file systems encrypt sensitive data stored in the mobile devices all the time. Whenever there is a need to access the data, the user feeds in a decryption key to decrypt them. Remote security systems usually work together with recovery/tracking-oriented mobile device protection systems. Once the recovery/tracking-oriented system locates the lost mobile device, the remote security system can remotely encrypt or remove the sensitive data from the device.

Though many cryptographic file systems such as Blaze's CFS [9] and Microsoft's EFS [10] provide elegant cryptographic methods for data protection, they require explicit user intervention to re-authenticate the user manually. As for remote security systems, they suffer the same weaknesses as tracking/recovery-oriented mobile device protection systems because they rely on tracking/recovery-oriented systems to function properly.

Some mobile devices [11], [12] utilize biometric sensing methods, such as fingerprint recognition, face recognition, etc, to manage the access to the mobile devices. Compared with traditional password-based methods, though they provide a more convenient way for the users to authenticate themselves, the biometric authentication methods still require explicit user intervention which could be a burden for the users.

In [13], the authors proposed a pervasive data protection system called ZIA. In ZIA, a user wears a small authentication token that communicates with a laptop over a short-range, wireless link. Whenever the laptop needs decryption authority, it acquires it from the token. ZIA relieves the user from the burden of frequent manual re-authentication. However, ZIA only provides data protection but cannot defend against physical compromise of the mobile device.

Comparing with the protection systems discussed above, EagleVision provides an integrated protection to both the mobile device and the data stored in it. Moreover, EagleVision achieves the desired protection in an autonomous manner without requiring explicit user intervention.

III. System Overview

In this section, we present the system model of EagleVision and its trust and threat models. We also briefly explain how EagleVision works through an example scenario.

A. System Model

EagleVision consists of the following four components: *Mobile Device Sensor (MDS)*, *User Sensor (US)*, *Infrastructure Sensor (IS)* and *Central Server (CS)*. Each mobile device carries an MDS which has several embedded sensors (e.g., an accelerometer) and can communicate wirelessly with other system components. User of the mobile device carries a US, which interacts with other system components.

Multiple ISs are deployed to form a wireless network infrastructure to cover an area of our interest (e.g., library, cafe or airport terminal). The wireless network infrastructure monitors the activities of USs and MDSs in the covered area continuously, and is connected to the CS which keeps the information about users and their mobile devices. ISs are loosely time-synchronized and preloaded with coarse location information (e.g., the ID of the room where an IS is installed). USs and MDSs receive location and time information from ISs. For each covered area, there is a central authority which maintains the CS and the wireless network infrastructure. Fig. 1 shows an example EagleVision deployment in a library, where the ISs are implemented using sensor motes.

In practice, the ISs may be implemented using Wi-Fi Access Points (APs) which have a much larger coverage than sensor motes, thus making the system more suitable for large-scale environments such as campus, airport terminals, etc.

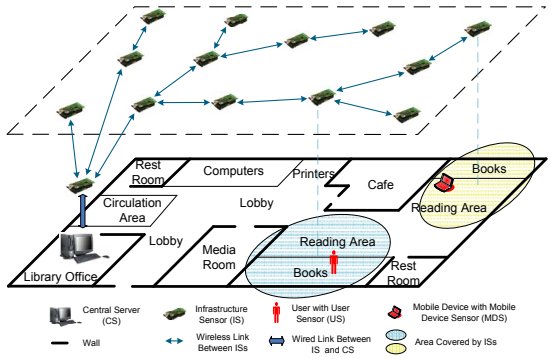


Fig. 1. An example deployment of EagleVision in a library

B. Trust and Threat Model

In EagleVision, the CS and ISs are considered trustable and non-compromisable. All the communications between the CS and ISs are secured by a shared secret key. A US is assumed to be secure as long as it is in the user's possession. An MDS is assumed to be secure when the user is nearby but may be tampered by the adversary if the user is away.

The aim of the adversary is to compromise the mobile device (and the sensitive data stored in it) without being detected. We consider the following practical attacks which can be launched easily by the adversary in real-world scenarios: (i) eavesdropping, (ii) fabricating/modifying messages, (iii) replaying messages, and (iv) relaying messages. The adversary also may try to compromise user privacy such as user's location. Privacy related issues are out of the scope of this work and hence not discussed in the paper.

C. An Example Scenario

The following example scenario explains how EagleVision works. Suppose Alice enters a library reading room with her laptop. Alice's US and her laptop's MDS register with the wireless network infrastructure via sending authenticated "probing" messages to nearby ISs. After registration, suppose Alice wants to leave the reading room to get some coffee from the cafe. Upon detection of Alice's absence (via proximity sensing), the laptop's MDS switches its safety mode and the laptop locks automatically with sensitive data encrypted. Meanwhile, the laptop's MDS starts to (i) sample its accelerometer to detect any movement of the laptop, and (ii) broadcast authenticated "alive" messages so that nearby ISs can monitor the laptop. If a sudden movement is detected, the laptop's MDS triggers an alarm with alert messages sent to Alice automatically (directly, via the wireless network infrastructure, or via email/text messaging). If the adversary destroys the MDS in the laptop, ISs will not be able to receive authenticated "alive" messages from the MDS, and thus the incident may be detected. As a result, ISs will report the incident to the CS, which in turn sends alert messages to Alice and the central authority. On the other hand, if no anomalies are detected during Alice's absence, when she comes back to her laptop, the laptop resumes normal operation by unlocking automatically with sensitive data decrypted.

In this example, EagleVision exhibits the following operational capabilities:

- *Transparency:* The system adapts its behavior autonomously without requiring explicit user intervention or causing extra distractions to the user.
- *Context Awareness:* Sensors carried by the user and the mobile device interact with each other as well as with the wireless network infrastructure to collect context information (e.g., proximity of the user to the mobile device) and the system adapts its behavior properly and promptly to the context change.

IV. System Design

Before proceeding to the design details of EagleVision, we first introduce the concept of *safety modes* of a mobile device. In EagleVision, depending on the proximity of the user to a mobile device, the mobile device may operate in one of the following three safety modes:

- *Strong Safe:* in this mode, the user is in close proximity to the mobile device (e.g., user sits next to the mobile device) and hence the possibility that the mobile device may be compromised is extremely low.
- *Weak Safe*: in this mode, the user is not in close proximity to the mobile device but not very far away either (e.g., user is a few feet from mobile device but still in the same room); therefore, the mobile device may be susceptible to compromise but the probability is low.
- *Unsafe*: in this mode, the user is far away from the mobile device and hence the mobile device is highly susceptible to compromise.

Fig. 2 illustrates the different functional modules executed in EagleVision at different safety modes and during transitions between safety modes. Details of these modules will be discussed in the following sections.

A. Context-Aware Switching of Safety Modes

In EagleVision, the MDS interacts with the US periodically to obtain the user proximity information. Usually, the user proximity may be inferred by the received signal strength of the US signals or may be estimated by using specific proximity sensors such as infrared sensors or ultrasonic sensors. EagleVision adopts the former method due mainly to its simplicity: *the received signal strength is measured and reported by most wireless devices and no additional hardware is required for such measurement.* Based on the measured signal strength of the received US signals, the MDS determines an approximate distance between the user and the mobile device and then switches to the corresponding safety mode automatically. The MDS also periodically informs the US it's current safety mode.

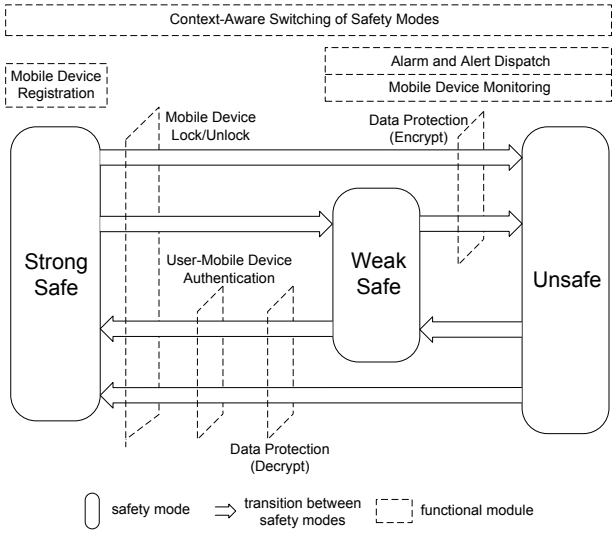


Fig. 2. At different safety modes and during transitions between safety modes, different EagleVision functional modules are executed.

B. Mobile Device Registration with the Infrastructure

Before describing the registration process in detail, we first explain the information that is preloaded to different components in EagleVision. As shown in Fig. 3, the CS and ISs are preloaded with a public one-way hash function $H(\cdot)$ and a secret key K_{I}. Each pair of US and MDS are preloaded with a unique identifier $ID_{\mathrm{U/M}}$, the hash function $H(\cdot)$, and two secret keys K_{auth} and $H(K_{\mathrm{I}}|ID_{\mathrm{U/M}})$ (where " | " represents concatenation). K_{auth} is used for authentication between the US and the MDS, which will be explained in Section IV-C, while $H(K_{\mathrm{I}}|ID_{\mathrm{U/M}})$ is used for authentication with the infrastructure. Each pair of US and MDS share an initial secret key $K_{\mathrm{U/M}}^{(0)}$, and the user preloads the login password for the mobile device (*PWD*) to the US.

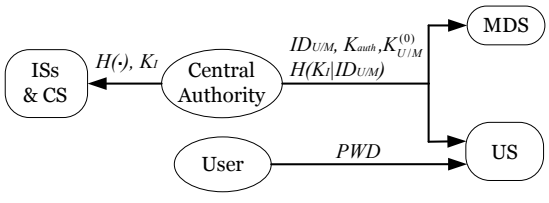


Fig. 3. Information preloaded to different EagleVision components

When the user enters an area covered by the wireless network infrastructure, his/her US and MDS start the registration process with the infrastructure through nearby ISs. The registration process consists of the following steps:

- The US broadcasts periodically "probing" messages in the following initial format:

$$\langle ID_{\mathrm{U/M}}, ID_{\mathrm{room}}, t, \mathrm{MAC}_{\mathrm{U/I}}, \mathrm{MAC}_{\mathrm{U/M}} \rangle,$$

where ID_{room} is zero, t is the US's local time before registration, $\text{MAC}_{\text{U/I}}$ and $\text{MAC}_{\text{U/M}}$ are message authentication codes used between US and IS, and between US and MDS, respectively. Here, $\text{MAC}_{\text{U/I}} = H\left(H(K_I|ID_{\text{U/M}})|ID_{\text{U/M}}|ID_{\text{room}}|t\right)$ and $\text{MAC}_{\text{U/M}} = H\left(K_{\text{auth}}|ID_{\text{U/M}}|ID_{\text{room}}|t\right)$.

- On hearing a "probing" message, the ISs get $ID_{\text{U/M}}$, generate the user's secret key by calculating $H\left(K_{\text{I}}|ID_{\text{U/M}}\right)$, and use it to verify the authenticity of the received message. Then they reply to the US with the location (i.e., room ID) and time information, authenticated also with a MAC generated by using $H(K_I|ID_{\text{U/M}})$.
- On receiving the ISs' replies, the US and MDS set their local time and set their ID_{room} as the room ID of the IS that has the strongest received signal strength. This completes the association and future "probing" messages become "associated probing" messages.
- On receiving the first "associated probing" message, the associated IS sends a "confirming" message to the US and forwards the mobile device's information to the CS.
- On receiving the mobile device's information from the IS, the CS records the mobile device's information.

Note that during the registration process, only the US communicates with the infrastructure. Since the mobile device's information is also contained in the US's "probing" messages, the mobile device gets registered with the infrastructure as well. After the initial registration, the user and mobile device may move to a different location. Various handover techniques [14] may be used to make sure that the US and MDS always maintain association with the IS that is closest to them. The newly associated IS then informs the CS of the user and mobile device's latest location. Moreover, to defend against possible security attacks to the registration process, all the messages exchanged between the US and ISs are attached with a MAC generated using the one-way hash function $H(\cdot)$ and the secret key $H(K_{\text{I}}|ID_{\text{U/M}})$.

C. User-Mobile Device Authentication

EagleVision allows the user to authenticate and log into the mobile device autonomously without requiring explicit user-mobile device interaction. The user authentication at the mobile device consists of the following two parts:

- The "probing" message, which is broadcast periodically by the US and heard by the MDS, needs to be authenticated so that the mobile device switches to the *Strong Safe* mode only when the user is truly nearby.
- Upon entering the *Strong Safe* mode, the US provides *PWD* to the MDS to log into the mobile device.

The authentication of the "probing" message is achieved by verifying $\text{MAC}_{\text{U/M}}$, which is generated by calculating $H\left(K_{\text{auth}}|ID_{\text{U/M}}|ID_{\text{room}}|t\right)$. The pairwise secret key K_{auth} is preloaded to both US and MDS (see Fig. 3). In order to deal with relay attacks, which will be discussed in Section V-D, the US's location is included in the "probing" message (see Section IV-B). Then the validity of the "probing" message is also verified at the MDS by comparing the locations of the US and MDS. If they are different, the "probing" message is deemed invalid since the MDS should not hear a "probing" message sent by the US from a different room.

Upon entering the *Strong Safe* mode, the MDS requests *PWD* from the US to log into the mobile device. Note that the "password request" is also authenticated with timestamp and a MAC and the transmitted *PWD* is encrypted using the password encrypting key $K_{\text{U/M}}^{(i)}$. To eliminate potential threats to *PWD*, the US records the mobile device's last known location, which is the location when the system enters the *Weak Safe* mode. $K_{\text{U/M}}^{(i)}$ is updated after each password request. Fig. 4 illustrates the login and key updating protocol in EagleVision.

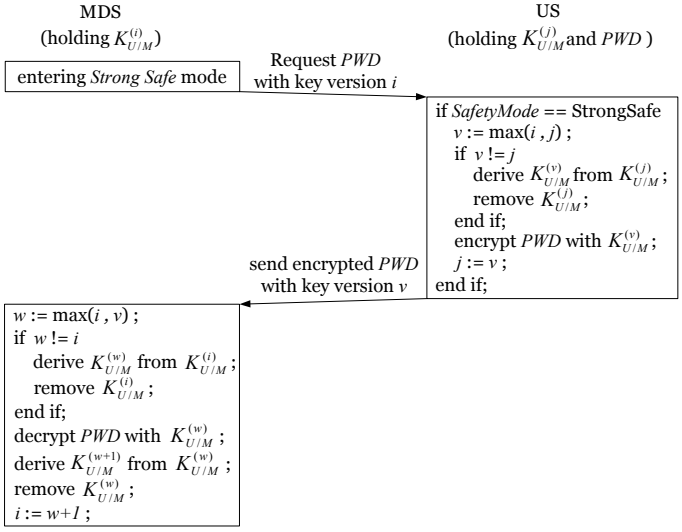


Fig. 4. The login and key updating protocol

The login and key updating process occurs only when the MDS enters the *Strong Safe* mode, i.e., when the US is very close to the MDS; so the probability that the "password request" or the "password reply" is lost is very small. However, if any of them is lost, the MDS will receive no "password reply" from the US; the MDS will then retransmit the request until the reply is received.

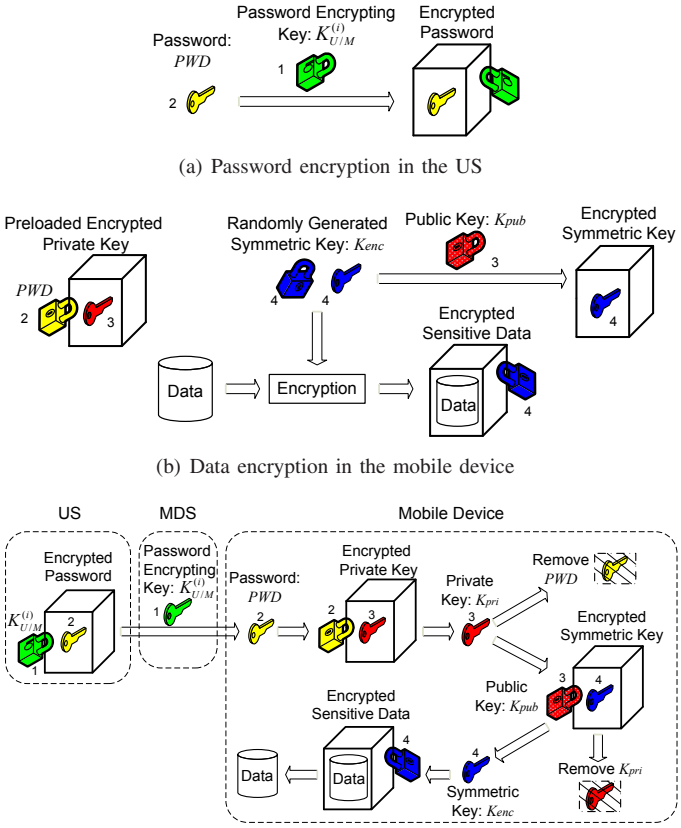
D. Mobile Device Lock/Unlock

When the user moves away from the mobile device and the mobile device enters the *Weak Safe* mode, the mobile device is secured by locking the device's operating system thus blocking unauthorized access. However, data stored in the mobile device is not encrypted in this situation. The reason for such design is due to the consideration that the user is not far away from the mobile device at the *Weak Safe* mode and hence the probability for the mobile device to be compromised is low. Moreover, locking the system usually is much more energy-efficient than performing data encryption. On the other hand, when the user returns and the mobile device enters the *Strong Safe* mode, the MDS acquires the password from the US to unlock the mobile device automatically, as discussed in Section IV-C.

E. Data Protection on Mobile Device

OS-level authentication may prevent unauthorized access to the mobile device's operating system. However, if the mobile device is physically compromised, data stored in the mobile device may still be accessed by other means such as plugging the hard drive into an alien mobile device. To provide protection to the data stored in the mobile device, EagleVision adopts a scheme that combines the usage of symmetric key, PKI and password, as shown in Fig. 5.

In EagleVision, the file system on the mobile device is encrypted using a symmetric key K_{enc}, which allows lower encryption and decryption latency. K_{enc} is protected with a PKI public key K_{pub} and the encrypted symmetric key $\{K_{\text{enc}}\}_{K_{\text{pub}}}$ is stored on the mobile device, as show in Fig. 5(b). Fig. 5(c)



(a) Password encryption in the US

(b) Data encryption in the mobile device

(c) Data decryption in the mobile device

Fig. 5. Data protection in the mobile device: (a) password encryption, (b) data encryption, and (c) data decryption procedures. In the figures, keys are color-coded: 1 – green, 2 – yellow, 3 – red, and 4 – blue.

shows the decryption procedure of the file system based on $\{K_{\text{enc}}\}_{K_{\text{pub}}}$, which is decrypted using two pieces of information: (i) the PKI private key K_{pri} which is encrypted with a password and stored on the mobile device, and (ii) the password which is stored on the US and passed to the mobile device through a secure channel upon the user's return to the mobile device that switches the mobile device's safety mode to *Strong Safe*. The password used in data protection may or may not be the same as the user login password for the mobile device.

F. Mobile Device Monitoring

When the user is away from the mobile device, i.e., when the mobile device is at the *Weak Safe* or *Unsafe* mode, EagleVision monitors the mobile device and provides theft detection in the following ways:

- *via motion detection:* the accelerometer embedded in the MDS helps detect sudden movement to the mobile device and trigger the response system if needed.
- *via absence of "alive" messages:* recall that the US and the MDS are always associated with the IS that is closest to them and the CS always maintains the latest location information about the mobile device. When the mobile device is at the *Weak Safe* or *Unsafe* mode, the MDS sends "alive" messages periodically to its associated IS, which is responsible for monitoring the mobile device and taking actions if needed. For example, if the adversary disables the MDS, the associated IS would be able to detect the incident via absence of "alive" messages and then trigger the response system. Similar to "probing" messages described in Section IV-B, the "alive" messages are also authenticated with timestamp and MAC, generated by calculating $H\left(H(K_I|ID_{\text{U/M}})|ID_{\text{U/M}}|ID_{\text{room}}|t\right)$, to defend against potential attacks which will be discussed in Section V.

G. Alarm and Alert Dispatch

When a potential theft is detected, an audible alarm will be triggered to deter the adversary from completing the theft. At the same time, alert messages will also be sent to the user. Specifically, when a theft is detected via motion detection, the MDS initiates the alert messages and sends them either directly to the user if the user is within direct communication range to the mobile device, or via the wireless network infrastructure. On the other hand, if a theft is detected via absence of "alive" messages, the associated IS initiates the alert messages and sends them to the user. In practice, it is possible that the user may go outside the area covered by the wireless network infrastructure. In this case, alert messages are dispatched to the user via text messaging to the user's cell phone and/or email to a user-specified email account.

V. Security Analysis

In this section, we analyze the security performance of EagleVision against the attacks specified in the threat model in Section III-B.

A. Resilience to Eavesdropping

The adversary may eavesdrop the wireless communications between different components in the system to capture or infer the secret keys that are used to generate the MACs (as discussed in Section IV-B and Section IV-C). If such attacks succeed, the adversary can impersonate the US and compromise the mobile device without being noticed, or impersonate the IS or CS to perform other attacks such as the denial of service attacks. However, the success probability of such attacks is negligibly low because it is extremely hard for the adversary to infer the secret keys from the overheard MACs due to the large size of secret keys and the application of secure one-way hash function in producing the MACs.

B. Resilience to Message Fabrication/Modification

The adversary may inject fabricated messages into the system or intercept and modify messages transmitted in the system. If such messages are not filtered out, the operations of the system will be disrupted. However, the success probability of such attacks is also negligibly low because it is extremely hard for the adversary to find out the secret keys used to compute the MACs (as analyzed above); without knowing the secret keys, the probability to produce a correct MAC for a fabricated or modified message is only $2^{-\ell}$ where ℓ is the number of bits in a MAC.

C. Resilience to Replaying Attacks

The adversary may replay some existing messages to disrupt the operations of the system. For example, the adversary can replay old "probing" messages of the US to the MDS, so as to cheat the mobile device to unlock itself. Such attacks cannot succeed because all the valid components in the system are time-synchronized and every message is attached with a timestamp; as a result, any replayed message can be detected and filtered out at the MDS. In practice, there could be a small time difference (about 10ms) between system components; however, the difference is much smaller than the interval (about 250ms) between consecutive "probing" messages, which means that the

time-stamp mechanism is adequate in dealing with replaying attacks.

D. Resilience to Relay Attacks

The adversary may intercept valid messages and relay them to a remote receiver to cheat the receiver into believing that the messages are sent from some nodes in its neighborhood. There are three possible relay attack scenarios:

- The US is away from the MDS but they can still communicate with each other directly, and hence the mobile device is at the *Weak Safe* mode. The adversary may relay the messages between the US and the MDS so that the US's RSSI at the MDS is as high as when the US is nearby. If the MDS is unaware of the attack, it would enter the *Strong Safe* mode. Consequently, it would unlock itself and stop theft detection.
  Remarks: EagleVision is resilient to such attacks for the following reasons. Since the MDS is within the communication range of the US, it will receive more than one US messages with the same time-stamp (i.e., one is directly from the US and the other is relayed by the adversary). Therefore, the MDS can detect the anomaly and alert the user of the threat. In case the communication between MDS and US is shielded and the packets are tunneled to MDS, IS can overhear more than one US messages. Therefore, IS can detect the anomaly and alert the user of the threat.
- The US is far away from the MDS (e.g., in two different rooms) and the mobile device is at the *Unsafe* mode. The adversary may launch the same relay attack as the one mentioned in the previous scenario.
  Remarks: EagleVision is resilient to such attacks because the MDS can filter out the relayed messages since the location of the user, which is included in the message, is different from the location of the mobile device (see Section IV-C).
- The adversary relays the messages of a remote IS to the US and MDS. The purpose is to associate the US and MDS to the remote IS. If this succeeds, when the user is away from the mobile device to a location near the remote IS, the adversary can launch the same relay attack as described above.
  Remarks: EagleVision is resilient to such attacks due to the static deployment of the wireless network infrastructure. Specifically, each IS is static and preloaded with the IDs of its one-hop neighbor ISs. So, if the messages sent by an IS are relayed and overheard by other ISs that are not one-hop neighbors to the source IS, the anomaly can be detected.

E. Effectiveness in Data Protection on Mobile Device

If the adversary has captured a mobile device, it can access all un-encrypted data stored in the mobile device and the attached MDS. However, as explained in Section IV-E, sensitive user data have already been encrypted immediately after a user moves away from his/her mobile device. Therefore, the adversary can access the data only if it can obtain the key used to encrypt the data. Furthermore, since the encryption key is encrypted with a password, the adversary needs to obtain the password. The password is stored in the US, not in the mobile device or the MDS attached to it. To obtain the password, the adversary can only make use of the partial information stored in the compromised mobile device and MDS, or the overheard information related to the password, or cheat the US to send back the password. Next, we analyze the effectiveness of these approaches:

- The adversary holds the current password encrypting key, $K_{\text{U/M}}^{(j)}$, which is stored in the compromised MDS. It may also have overheard the previous encrypted password, which was encrypted using an older version of the password encrypting key, $K_{\text{U/M}}^{(i)}$. As explained in Section IV-C, the version of $K_{\text{U/M}}^{(j)}$ is always newer than that of $K_{\text{U/M}}^{(i)}$, i.e., $i < j$, and $K_{\text{U/M}}^{(j)}$ is computed via applying a secure one-way hash function on $K_{\text{U/M}}^{(i)}$. Therefore, it is extremely hard for the adversary to obtain $K_{\text{U/M}}^{(i)}$ from $K_{\text{U/M}}^{(j)}$ to decrypt the password.
- The adversary, holding $K_{\text{U/M}}^{(j)}$, may attempt to let the US send the password encrypted by $K_{\text{U/M}}^{(j)}$. To execute this attack, the adversary can fabricate password requests using the secrets obtained from the compromised MDS and send the requests to the US. A US switching to unsafe mode is designed to respond the password request if and only if it returns to the MDS's last known location and switches back to weak or strong safe mode (see Section IV-C). However, as EagleVision notifies the user within a few seconds after the theft occurs, likely the user would have already been notified of the theft; thus the user can simply turn off the US to deny the adversary's attempt to retrieve the password from the US.

Besides, the adversary who has compromised some USs and/or MDSs may attempt to break other USs' and or MDSs' secret keys. Recall that a US's secret key is in the form of $H(K_{\text{I}}|ID_{\text{U/M}})$. Hence, without knowing K_{I}, which is a secret held by the infrastructure, it is infeasible for the adversary to compute other USs' and MDSs' secret keys.

VI. Prototype Implementation

We have implemented a prototype EagleVision and tested it in our department building. Details of the prototype implementation are given in the following sections.

A. Hardware Components

In the prototype, we choose to use laptops as the mobile devices to be protected. Hardware choices for other components in the prototype are as follows:

- *Mobile Device Sensor (MDS)* – MICAz mote with an attached MTS310 sensor board, which has a 2-axis accelerometer. MICAz mote is connected to the laptop via an MIB520 programming board (rather than embedded as part of the laptop).
- *User Sensor (US)* – TelosB mote which is very small in size and can be carried in the user's pocket.
- *Infrastructure Sensor (IS)* – TelosB mote.
- *Central Server (CS)* – Dell desktop connected to a MICAz mote that serves as the communication interface between the CS and ISs.

We deploy a wireless network infrastructure on the third floor of our department building, as shown in Fig. 6. All the ISs are placed inside the ceiling. Specifically, IS_1 is deployed to cover an office, while the other ISs are deployed to cover the corridor. The CS is located in the same room as IS_1. The CS and ISs form a tree structure which is rooted at the CS and has two branches: from IS_1 to IS_6 and from IS_7 to IS_8.

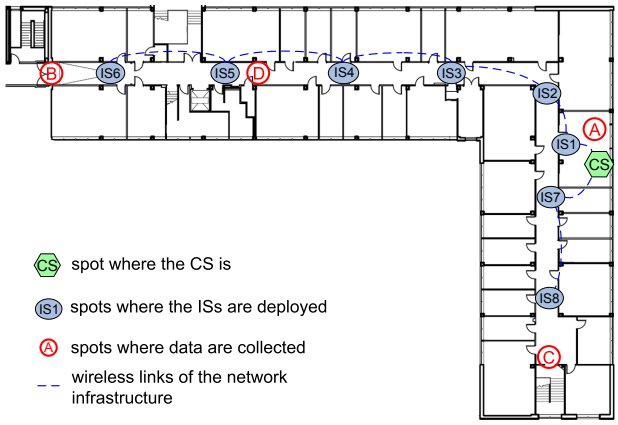


Fig. 6. Deployment of the prototype EagleVision

In the prototype implementation, in order to loosely time-synchronize all the ISs, the CS broadcasts synchronization beacons to all the ISs every 10 minutes. Each IS keeps a local clock with a precision of 5 seconds and the clock is represented by a 32-bit counter (32 bits enable it to run more than 500 years without repeating). Upon receiving a synchronization beacon, the IS adjusts its time according to the time carried in the beacon. Besides, each IS calibrates its clock periodically based on the measurement of actual interval between consecutively-received beacons. For example, suppose the interval of two beacons received consecutively by an IS is 620 seconds measured in the IS's local time, which means that the clock drift for this IS is +20 seconds every 10 minutes comparing to the CS's clock. As a result, the IS will decrease its local time by 2 seconds every 62 seconds.

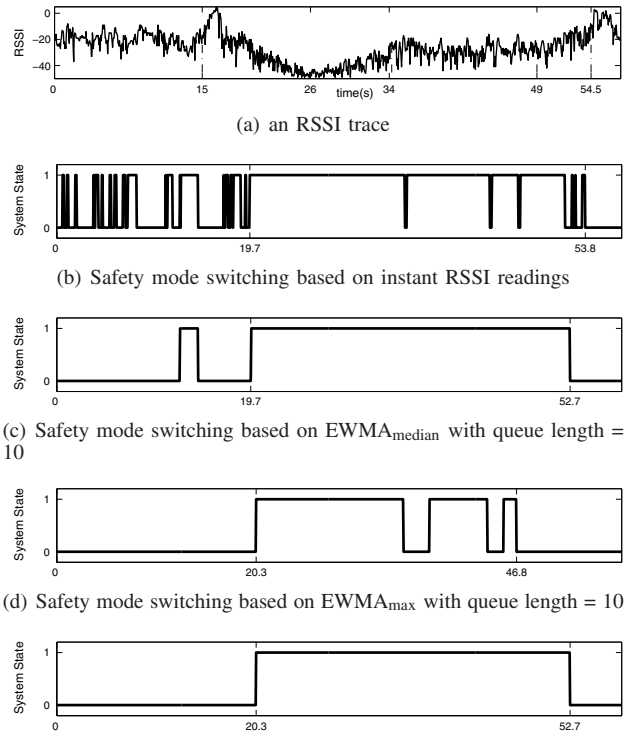
B. Context-Aware Switching of Safety Modes

The *"Context-Aware Switching of Safety Modes"* module is the kernel software module in EagleVision. Next we present its implementation details in the prototype.

1) Using RSSI to Infer User Proximity: In EagleVision, context refers to the user proximity to the mobile station, which is inferred by the RSSI (Received Signal Strength Indicator) reading of the US "probing" messages reported by the MDS. With MICAz motes, RSSI readings range from -50 to 50 (no unit) with an offset of approximately -45 [15]. For example, if an RSSI reading is -20, the measured signal strength is approximately -65 dBm. Fig. 7(a) plots an RSSI trace of US "probing" messages collected in a 60-second experiment. During the experiment, the laptop is placed on a desk while the user changes his relative position to the laptop over time. We notice that:

- RSSI values vary drastically over time due to the interference and environmental changes between the US and the MDS (e.g., human body, furniture, movement, etc).
- Even when the user sits next to the laptop, the RSSI values can be very low at times (e.g., shortly before 15s) because the signal may be blocked by the desk and/or the human body. In comparison, when the user stands up, the RSSI values improve (e.g., right after 15s).
- When the user sits next to the laptop, the occurrence of very low RSSI values is transient: usually a single or a short burst of very low readings.

Based on these observations, we conclude that instant RSSI readings should not be used to infer the user proximity and decide the switching of safety modes, since it may cause too



(a) an RSSI trace

(b) Safety mode switching based on instant RSSI readings

(c) Safety mode switching based on $\text{EWMA}_{\text{median}}$ with queue length = 10

(d) Safety mode switching based on EWMA_{max} with queue length = 10

(e) Safety mode switching in EagleVision

Fig. 7. (a) An example RSSI trace. (b)–(f) show the safety mode switching patterns when different schemes are used, where "0" respresents the *Strong Safe* mode (laptop's screen unlocked) and "1" represents the *Weak Safe* or *Unsafe* mode (laptop's screen locked). User's action over time: (i) 0–15s, sits next to the laptop; (ii) 15–26s, stands up and walks away from the laptop; (iii) 26–34s, walks back towards the laptop; (iv) 34–49s, stops at about 4 meters from the laptop; (v) at 49s, continues walking back to the laptop; (vi) at 54.5s, comes back to the laptop.

many false actions. This can be clearly observed from Fig. 7(b). Instead, an effective outlier filtering scheme shall be designed to reduce false actions when using RSSI readings to infer user proximity.

There is a well-known outlier filtering scheme used by the the B-MAC protocol [16] for wireless sensor networks to deal with high RSSI variation. The basic idea is to enter the RSSI readings into a FIFO queue and the *median* RSSI reading of the queue is added to an exponentially weighted moving average (EWMA) with a decay factor of 0.2. The output of the EWMA is then used to compare with decision thresholds. We refer to this scheme as $\text{EWMA}_{\text{median}}$. Fig. 7(c) plots the results when $\text{EWMA}_{\text{median}}$ is used. It can be seen that the number of false actions is reduced significantly: only a single false action is observed. Alternatively, we may use the *max* RSSI reading of the queue as the input to EWMA. We refer to this scheme as EWMA_{max} and results are shown in Fig. 7(d). Comparing with $\text{EWMA}_{\text{median}}$, EWMA_{max} is more resilient to short bursts of very low RSSI readings during the *Strong Safe* mode. As a result, false actions are eliminated completely when the user sits next to the laptop. Unfortunately, more false actions can be observed when the user is away from the mobile device. So there is a tradeoff. The safety mode switching algorithm used in the prototype EagleVision is a hybrid scheme that combines $\text{EWMA}_{\text{median}}$ and EWMA_{max}, which reduces the false actions effectively at all safety modes, as shown in Fig. 7(e). Details of this algorithm will be discussed in the next section.

2) Mode Switching in EagleVision: In general, two decision thresholds, TH_{S} and TH_{W}, are needed to decide the switching of

safety modes between *Strong Safe* and *Weak Safe*, and between *Weak Safe* and *Unsafe*, respectively. In the prototype, instead of using a single TH_S, we use two thresholds, TH_{SW} and TH_{WS}, which correspond to the switching from *Strong Safe* to *Weak Safe* and from *Weak Safe* to *Strong Safe*, respectively. The reason for such design is to reduce the frequent switching between *Strong Safe* and *Weak Safe*, which could be particularly annoying to the user because the laptop may toggle between lock screen and unlock screen frequently. In the prototype, the laptop maintains outputs of both $EWMA_{median}$ and $EWMA_{max}$. $EWMA_{max}$ is applied when the safety mode is *Strong Safe*; otherwise $EWMA_{median}$ is applied. The switching conditions between different safety modes are shown in Fig. 8. Based on the empirical results, we set the FIFO queue length to 10, the EWMA decay factor to 0.2, TH_{SW} = -25, TH_{WS} = -18 and TH_W = -42.

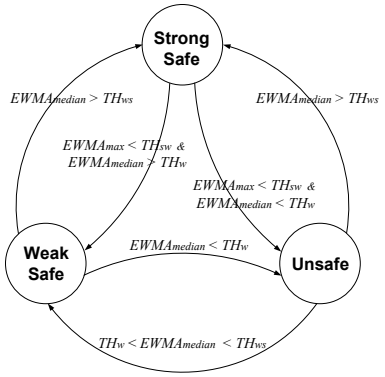


Fig. 8. Switching of safety modes in EagleVision

A side effect for using such a queue-and-EWMA-based mode switching scheme is the reduced system responsiveness to the context change. Comparing Fig. 7(e) with Fig. 7(a), we can see that EagleVision locks the laptop at 20.3s, which is 5.3 seconds after the user starts to walk away from the laptop. This delay is slightly larger than that when instant RSSI readings or $EWMA_{median}$ are used (4.7 seconds), but the difference is insignificant.

C. Other Software Modules

Other functional modules in the prototype EagleVision are implemented as follows:

- *Mobile Device Registration:* All the ISs that overheard the "probing" messages from a newly joining US reply with its ID and a time-stamp. The US chooses to associates with the IS that has the strongest RSSI.
- *User-Mobile Device Authentication:* We use TinySec to encrypt the password and generate the MACs.
- *Mobile Device Lock/Unlock:* A Java API is used to lock and unlock the laptop screen.
- *Data Protection on Mobile Device:* We use OpenSSL to encrypt and decrypt the user data. Specifically, we use AES to protect the user data and RSA to protect the symmetric key that is used to protect the user data.
- *Mobile Device Monitoring:* The MDS samples its accelerometer every second if it is not in the *Strong Safe* mode. Meanwhile, for each monitored mobile device, the associated IS will report alert if 5 consecutive "alive" messages are missing.
- *Alert Dispatch:* Alert messages are dispatched to the user via a unicast routing protocol through the wireless network infrastructure. Meanwhile, a Java API is used to send text message and email to the user.

We measure the ROM and RAM consumptions for MDS, US and IS. Results are shown in Table I.

TABLE I
STORAGE OVERHEAD FOR SOFTWARE MODULES IN SENSORS (IN KB)

	MDS	US	IS
ROM	27.928	21.702	28.170
RAM	1.557	1.282	3.098

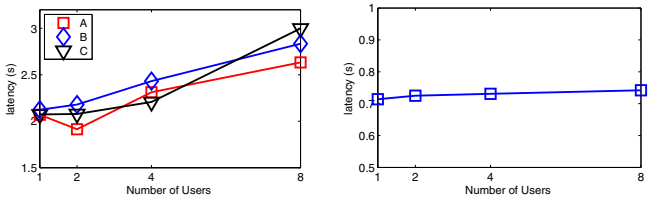
VII. PERFORMANCE EVALUATION AND FIELD TEST

We conduct various experiments to evaluate the performance of the prototype EagleVision.

A. Performance Evaluation

1) Mobile Device Registration Delay: The registration delay should be kept small so the mobile device can be under protection as soon as the user enters a area covered by the wireless network infrastructure. The registration delay is defined as the time interval from when the user enters the area to when the CS receives the user's information. We place the laptop at different locations: A, B and C in Fig. 6, and measure the registration delay.

Experimental results are plotted in Fig. 9(a). It can be seen that all registrations can be done quickly within 3 seconds, and results do not vary much with the laptop location. The slightly increased registration delay when the number of users increases is due to collisions of messages from multiple users.



(a) registration delay (b) alert dispatch delay

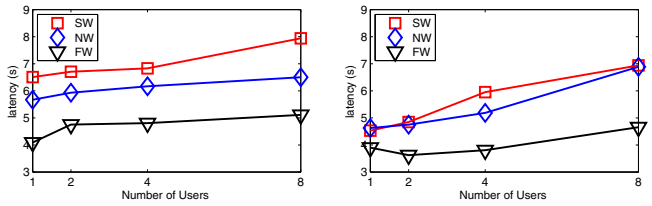
Fig. 9. Registration delay and alert dispatch delay (results are averaged over 10 experimental runs)

2) Mobile Device Lock/Unlock Delay: We measure the delay of laptop screen lock/unlock delay, which is also the delay of switching between *Strong Safe* and *Weak Safe/Unsafe* modes. Specifically, these delays are defined as:

- *Delay of lock screen* – time interval between when the user starts to walk away from the laptop and when the screen is locked.
- *Delay of unlock screen* – time interval between when the user comes back to the laptop and when the the screen is unlocked, i.e., the time during which the user waits for the screen to unlock.

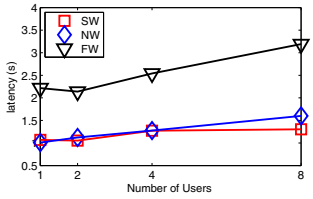
We measure these delays in two different testing scenarios: office and lobby. The size of the office is approximately 12 $\times$ 30 feet and the lobby is about 40 $\times$ 40 feet. We test three different user movement patterns: slow walking (SW), normal walking (NW) and fast walking (FW), which correspond to the moving speed (approximately) of 0.5m/s, 1.5m/s and 2.5m/s respectively. Results are shown in Fig. 10.

In general, EagleVision reacts quickly to the user's proximity change and there is a mild performance degradation when the

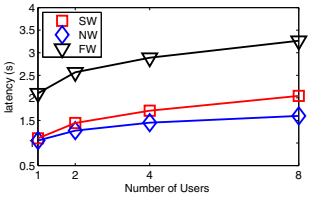


(a) lock delay in office

(b) lock delay in lobby



(c) unlock delay in office



(d) unlock delay in lobby

Fig. 10. Mobile device lock/unlock delay (results are averaged over 10 experimental runs)

number of users gets large. In the worst situation for locking screen (i.e., office scenario, 8 users, slow walking), the laptop takes about 8 seconds to lock the screen, or equivalently, the laptop locks the screen when the user is approximately 4 meters away. In real-world scenarios, this is likely sufficient to prevent any unauthorized access. In the worst situation for unlocking screen (i.e., lobby scenario, 8 users, fast walking), the user needs to wait for about 3 seconds for the screen to unlock. This is because that $\text{EWMA}_{\text{median}}$ is slow in reacting to the suddenly increasing RSSI readings, thus causing the MDS to be less responsive. Although a delay of 3 seconds may be tolerable to users in real-world scenarios, we plan to study, as part of the future work, how to improve EagleVision to be more responsive while still maintaining small number of false actions.

3) Theft Detection Delay: The theft detection delay is defined as the time interval from when the adversary takes the laptop to when the theft is detected, either via motion detection or via absence of alive messages. In our prototype implementation, the theft detection delay via absence of alive messages is always 1.25 seconds (i.e., 5 alive message intervals), as explained in Section VI-C. Therefore, we only measure the theft detection delay via motion detection.

We ask some students to help emulate adversary behaviors. Three adversary behaviors are studied: fast walking, normal walking and slow walking, which correspond to the moving speed of 2.5m/s, 1.5m/s and 0.5m/s, respectively. A slow walking adversary tries to move the laptop as cautiously as possible without triggering the accelerometer.

Each experiment is repeated for 10 times and Table II lists the results. EagleVision is able to detect all the anomalies in our experiments. Notice that for a slow-walking adversary, it may takes about 8 seconds to detect the theft; however, in this situation, the adversary is only about 4 meters away and a quick follow-up action will likely prevent the eventual mobile device loss. In practice, an extremely cautious and slow-walking adversary may be able to bypass the theft detection via motion detection. However, once the laptop leaves the coverage of the associated IS, the infrastructure can detect the anomaly in 1.25 seconds via absence of alive messages and then trigger the response system. Such a double protection mechanism makes EagleVision particularly effective in protecting mobile devices.

TABLE II

THEFT DETECTION DELAY VIA MOTION DETECTION (IN SECONDS)

Adversary Behavior	Average Delay
fast walking	1.025
normal walking	3.464
slow walking	8.278

4) Alert Dispatch Delay: The alert dispatch delay is defined as the time interval from when the theft is detected to when the alert is received by the user. If the user is within the direct communication range of the MDS, it can receive the MDS's alert almost immediately. So we only measure the delay when the alert is dispatched through the wireless network infrastructure or email/text messaging systems.

We measure the alert dispatch delay by placing the laptop at location D in Fig. 6. The user is at location B and the MDS cannot communicate with the US directly. This is considered the worst-case scenario for alert dispatch in our prototype implementation because alert messages have to go to the CS first and then from the CS to the US.

Fig. 9(b) and Table III show the results when alert messages are dispatched through the wireless network infrastructure and text messaging/email systems, respectively. We can see that the delay for alert messages to route through the infrastructure is very small (less than 1s). With text messaging or email, the delay is longer (about 10s) due mainly to the delay incurred at the cellular network or email servers.

TABLE III

ALERT DISPATCH LATENCY VIA TEXT MESSAGING/EMAIL (IN SECONDS)

	Average Delay
text messaging	8.72
email	10.50

B. Field Test

To evaluate the effectiveness of EagleVision in real-world conditions, we recruit four students (referred to as P1, P2, P3 and P4) to participate in a field test. All the students are given a laptop with EagleVision installed and they use the laptop for a long period of time (e.g., 8 to 11 hours). Their cubicles are in the same room as location A in Fig. 6. We record the numbers of false positives and false negatives during the test, which are defined in Table IV. Results are given in Table V.

From Table V, we can see that FP1 occurs rarely, about **once** per hour on average. Based on participants' feedback, we learn that most of FP1 events occur when the participant changes the sitting posture, which usually causes the RSSI reading to vary significantly. However, our system quickly unlocks the

TABLE IV

DEFINITIONS OF FALSE POSITIVES AND FALSE NEGATIVES

False Positives	*FP1*: screen is locked when the user is using the laptop
	FP2: screen is unlocked when the user is away
	FP3: theft is reported but no theft happens
False Negatives	*FN1*: screen remains unlocked when the user walks away
	FN2: screen remains locked when the user comes back

TABLE V

FIELD TEST RESULTS

	T_{total}	# of FP1	# of FP2	# of FP3	# of FN1	# of FN2
P1	8 hours	8	0	0	0	0
P2	10 hours	9	0	0	0	0
P3	9 hours	11	0	0	0	0
P4	11 hours	11	0	0	0	0

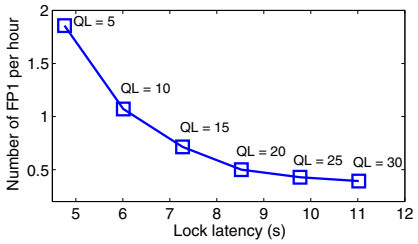


Fig. 11. Lock latency vs. FP1 event frequency when different queue lengths (QL) are used. The lock latency is measured in the scenario where four users are present in the office and results are averaged over ten tests. The FP1 event frequency is calculated based on the RSSI traces collected from the four participants.

laptop once the participant returns to a stable sitting posture. It is possible to reduce the number of FP1 events by increasing the length of the FIFO queue, which however would trade off the responsiveness of the system. Fig. 11 shows the tradeoff between the responsiveness of the system measured by lock latency and the frequency of FP1 events on average when different FIFO queue lengths are used. We can see when longer FIFO queue is used, fewer FP1 events are encountered per hour, while the latency to lock the laptop increases. Our prototype sets the FIFO queue length to be 10; in such case both metrics are considered acceptable. Other types of false positives or false negatives are not observed in the test. These results suggest that EagleVision accomplishes well the design goal of *transparency*: it introduces rare/no extra distractions to the user.

VIII. Discussion and Future Work

A. Vulnerability of US Loss

The US enables the pervasive protection to the mobile device. The loss of a US could be a very serious threat. In EagleVision, once a US is lost, the user would be denied pervasive access (e.g., autonomous log into the system) to the mobile device; therefore, the user can notice the loss of US in a timely manner. Then, the user can manually log into the mobile device and change the password to prevent further potential threats.

B. Protection of Inactive Mobile Devices

As long as the MDS is active, EagleVision is able to protect the mobile device (such as theft detection, alert dispatch, etc) even if the mobile device is inactive (e.g., hibernated, suspended, etc). To provide protection of data on an inactive mobile device, sensitive data could be encrypted before the mobile device enters the inactive state and decrypted after the mobile device returns to normal operation.

C. Trustworthiness of Infrastructure Sensors

In our system, ISs are assumed to be deployed and maintained by the administrator of the protected building. The administrator can make ISs safe by installing them in protected places like locked boxes. In addition, ISs themselves may be protected by motion detection sensors. Once excessive motion of an IS is detected, its neighboring ISs and CS will be notified and then this IS will be inspected by the authorities for any physical tampering.

D. Other Issues

In this paper, we focus on practical attacks that can be launched easily in real-world scenarios. Technically, there are other attacks that may be launched to shut down or break our system. For example, when a user is not near the mobile device, an adversary might shield the communication between US and MDS as well as nearby IS, and tunnel the packets from US to MDS, thereby unlocking the mobile device. However, it may be practically impossible for the adversary to shield the US from the MDS and all the nearby ISs of a user's locality, particularly when the user is on the move. Such attacks, hence, are not considered in this paper.

Other design choices like the deployment density of ISs may affect the performance of our system. In case the IS density is low, the registration and the alert dispatch delay may vary based on the user density. On the other hand, lower IS density implies larger coverage of each IS. As a result, US and MDS that are far away from each other (e.g., in different rooms) may still associate with the same IS. This may affect the security performance of our system such as defense against relay attacks. One possible solution is to use RSSI values of the messages from US and MDS to differentiate their locations. However, due to rapid fluctuation of RSSI values in practice, this issue needs further investigation.

IX. Conclusions

In this paper, we propose EagleVision, a pervasive mobile device protection system. It is a context-aware system and varies the protection level to the mobile device and the data stored in the mobile device in an adaptive manner to the context change, such as the user proximity to the mobile device. We implement a prototype of EagleVision using MICAz and TelosB motes and deploy it on the third floor of our department building. Various experiments are conducted to evaluate its effectiveness. Results show that EagleVision responds promptly to the context change and provides adequate protection to both device and data, while not requiring explicit user intervention or causing extra distractions to the user. Future work includes further improvement of the system responsiveness and the application of EagleVision to other types of mobile devices.

Acknowledgments

The research reported in this paper was supported in part by the Information Infrastructure Institute (iCube) of Iowa State University and the National Science Foundation under Grants CNS 0831874 and CNS 0716744.

References

[1] Laptop Theft Resource, "2009 Laptop Theft Expected To Skyrocket," http://www.laptoptheft.org/2009/01/02/2009-laptop-theft-expected-to-skyrocket/.
[2] CNNPolitics.com, "VA will pay $20 million to settle lawsuit over stolen laptop's data," http://www.cnn.com/2009/POLITICS/01/27/va.data.theft/index.html.
[3] "AbsoluteSoftware," http://www.absolute.com/.
[4] "Lojack for Laptops," http://www.lojackforlaptops.com/.
[5] "GadgetTrak," http://www.gadgettrak.com/.
[6] "MyLaptopGPS," http://www.mylaptopgps.com/.
[7] "Caveo," http://www.caveo.com/.
[8] "Musatcha.com," http://www.musatcha.com/.
[9] M. Blaze, "A cryptographic file system for unix," 1993.
[10] Microsoft Corporation, "Encrypting file system in windows xp and windows server 2003," http://technet.microsoft.com/en-us/library/bb457065.aspx.
[11] "ThinkPad R/T/X/W Series Laptops," http://www.lenovo.com/.
[12] "Sony VAIO Laptops," http://www.sonystyle.com/.
[13] M. D. Corner and B. D. Noble, "Zero-interaction authentication," in *Proceedings of the 8th annual international conference on Mobile computing and networking*, 2002.
[14] N. Ekiz, T. Salih, S. Kucukoner, and K. Fidanboylu, "An Overview of Handoff Techniques in Cellular Networks," *International Journal of Information Technology (IJIT)*, Jul 2005.
[15] "CC2420 Data Sheet," http://inst.eecs.berkeley.edu/∼cs150/Documents/CC2420.pdf.
[16] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*. New York, NY, USA: ACM, 2004, pp. 95–107.