Protecting Unattended Computers Without Software

Carl E. Landwehr
Naval Research Laboratory
Code 5542
Washington, DC 20375-5337
e-mail: landwehr@itd.nrl.navy.mil

Abstract

In many environments, users login to workstations and then leave them unattended. Rather than trying to stop users from doing what comes naturally, this paper suggests a simple, hardware-based system that can protect computers in such an environment from unauthorized use by those with physical access to the monitor and keyboard. Requirements for the system are described, some design issues are discussed, and a sketch of a design for an initial prototype is provided, together with an assurance argument for it. A prototype implementing many of the concepts described has been built; two dozen copies of a second prototype are soon to be installed in an office environment.

1. Introduction

Computers mediate more and more actions. Where computer-mediated actions have important consequences, the concept of authorization is frequently invoked: the computer should only permit authorized acts. To determine whether a proposed action is authorized or not, the computer needs to identify the human who has invoked the action and should be held accountable for it.

But people like to be recognized without inconvenience, and in many settings it is quite inconvenient to identify oneself to a computer system repeatedly during the course of a day. This is a principal reason users in some environments, ranging from military command centers to hospital units, resist systems that require them to log in (and out). Where systems require a user to present an identifier and a password, users routinely try to pick simple passwords that, by being easy to guess, defeat the intended purpose. Further, users may log in and walk away without logging out, unintentionally leaving workstations available to unauthorized users.

People want to secure their computers, but not if it costs too much (not more than five to ten percent of the system's cost, by some accounts [1]) and not if it makes the system inconvenient to use or incompatible with needed applications. The challenge is to find ways to make systems both more secure and easier to use, without making them expensive as well. Can we build a system that would allow an authorized user to gain immediate access to his computer when he walked up to it, but would prevent unauthorized users from doing so after he has walked away? Can such a system be built without installing new software or altering existing software on the computer?

This paper briefly reviews technology relevant to this problem, proposes a set of requirements that could define a family of useful systems, and describes an existing prototype that realizes one set of requirements.

2. Related Technologies

2.1 Biometrics

One avenue of approach to this problem is through biometrics. If the user can be identified by the system on the basis of his or her physical characteristics -- fingerprint [2], iris pattern [3], voice [4], hand geometry [5], or other feature or behavior [6] -- it may be possible to log the user in without requiring a password. Although the cost of these technologies is declining and their accuracy improving, they are not yet available at cost/accuracy combinations that make them attractive for use on typical computer workstations. Further, they do not generally simplify detecting that the user, once authenticated, remains in the vicinity of the workstation. Where such assurance is specifically needed today, it is typically gained by repeating the authentication process periodically.

2.2 Tokens

Tokens, in the form of magnetic stripe cards, smart cards, and PC cards can also be used to authenticate users. Each of these requires a reader of some sort, and the reader

Landwehr, C.E. Protecting Unattended Computers Without Software. *Proc. Thirteenth Ann. Computer Security Applications Conf*, San Diego, CA, Dec. 1997, pp.274-283

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE
1997

2. REPORT TYPE

3. DATES COVERED
00-00-1997 to 00-00-1997

4. TITLE AND SUBTITLE
Protecting Unattended Computers Without Software

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Research Laboratory,Code 5542,4555 Overlook Avenue, SW,Washington,DC,20375

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE
unclassified	**unclassified**	**unclassified**

17. LIMITATION OF ABSTRACT

18. NUMBER OF PAGES
11

19a. NAME OF RESPONSIBLE PERSON

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

can retain the token for the duration of the session. Retaining the token does not assure that the user associated with it is still present at the workstation. In each of these cases, the user must actively insert the card in a special purpose device, which is a nuisance and provides an incentive for the user to leave the card in the reader when he leaves the workstation on short errands. Also, the reader will in general require some cooperation from the workstation software to function.

Tokens that can be sensed without the need for a physical contact are also a possibility. For example, the Fastoll system [7] incorporates a windshield-mounted token (a transponder) that emits an identifying signal when illuminated by a radio frequency (RF) transmitter mounted on the toll booth. Mobil's Speedpass [8] uses a smaller transponder to support wireless transactions with gasoline pumps. Many facilities use badges that can be detected by readers incorporated into walls; the badge need only be brought within a few inches of the reader to be activated. These kinds of readers, however, are not generally suited for installation on individual workstations. Olivetti has developed an Active Badge system [9] that can track users' whereabouts. The badges emit infrared (IR) pulses every few seconds that are detected by a network of sensors. This system permits workstation-mounted sensors, but it requires tracking software to be run on the host workstation and the use of IR means that a badge that is not within the line of sight of a sensor is effectively turned off.

There are also tokens such as Security Dynamic's SecureID token [10] that are designed to be integrated with password systems. Such tokens can automate one-time password systems and require no external input device other than the keyboard, but they do not help detect the authenticated user's departure.

A potentially more promising avenue can be found in automotive remote key entry systems. These use RF identification techniques, are based on low cost components, permit several different signals to be sent (lock, unlock, panic), and in some cases are designed to defeat replay attacks. Present systems do, however, require the user to announce his intentions by pressing a button; they don't automatically detect the user's departure. They are also vulnerable to the theft of the token; stealing the owner's key permits you to steal the car.

2.3 Detectors

Suppose, then, that tokens represent a feasible component of a solution to the problem. How can they be used to secure a system without installing software on it? Consider the vulnerabilities of a computer that has no keyboard or monitor attached to it. It will be considerably harder for a person with physical access to such a machine to attack it than it would be to attack a machine that is displaying the results of the last operation on its monitor while actively waiting for the next user command to be entered through its keyboard or mouse. Suppose, then, that we introduce a simple Detector that senses the presence or absence of an authorized token and either connects or disconnects the keyboard/mouse and monitor from the processor if no authorized token is present. Such a Detector could simply be plugged into an existing workstation without altering the installed software at all. Such a Detector could, with appropriate connectors and a few minor electrical tricks, be used unaltered with a PC, Macintosh, Sun, or other workstation.

We next consider the requirements of a system for wireless user identification and system protection along these lines.

3. Wireless Identification System Requirements

A *wireless identification Agent* (WIA) is a device that a user can carry in his or her pocket that, with little or no outward action on the part of the user, can identify the user to the workstation in combination with a workstation-mounted *Detector*. It also permits the workstation to detect when the user has left the vicinity so that others cannot place requests on behalf of the previously identified party or read results that may have been left on the screen. We envision a *family* of such devices to meet different threats and to cooperate with different applications. Some family members might incorporate a small keypad, to allow the owner to authentic herself to the device, and a beeper to allow the device to alert its owner. Figures 1 and 2 show possible physical configurations for an Agent and a Detector.

The fundamental technologies to develop and market such a device at an acceptable price appear to be in place, although there are many details to be specified and tradeoffs to be addressed if a practical implementation is to be developed. The following two subsections provide sample requirements for the Agent and the Detector that might be appropriate for a hospital or military command center willing to invest up to $150 per workstation to ease user access and improve security.

3.1 Wireless Identification Agent (WIA) Requirements

Ease of use: The Agent must provide its owner's identity to the Detector with little or no action by the owner. It should be able to function without being removed from owner's pocket (some family members should operate from a front or back pocket, others may be visible and be able to hold a photo of the owner, for use as badge).

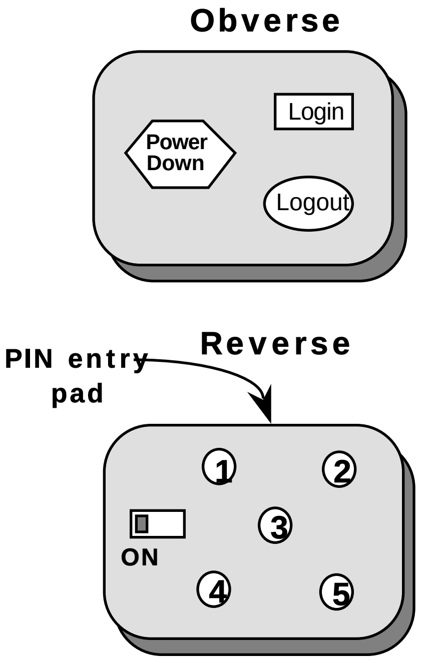


Figure 1. Wireless Identification Agent possible configuration. Size 2” x 1” x 1/2” or smaller.

Size: The Agent must be small. Preferably it could be attached to a key-ring; it could possibly be as big as PC Card

Low maintenance: The Agent should not require a new battery for three years, or it should at least be easy to change the battery with safe key storage.

Light weight: the Agent should not be heavier than the average PC Card.

Low cost: Agent plus Detector should retail for less than $150, in quantity.

Limited range: The Agent should have an active range up to *D* feet. (estimate *0<D<10*)

User-limited detectability: The Agent’s owner must be able to switch it off (to avoid use as continuous location Detector).

Theft-secure: Simple theft of the Agent should not permit the thief to spoof the Detector.

Replay-secure: The Agent/Detector combination should resist replay attacks.

Tamper-resistant: the Agent must resist tampering (to resist casual, but not determined, attacks).

Embeddable: (Some family members) The Agent must be embeddable in PDA, pager, cell phone.

3.2 Detector Requirements

Simple hardware integration: The Detector should require minimal modifications to host hardware (preferably, it should simply plug in to existing interfaces; maximum change might be to insert a card in a PC).

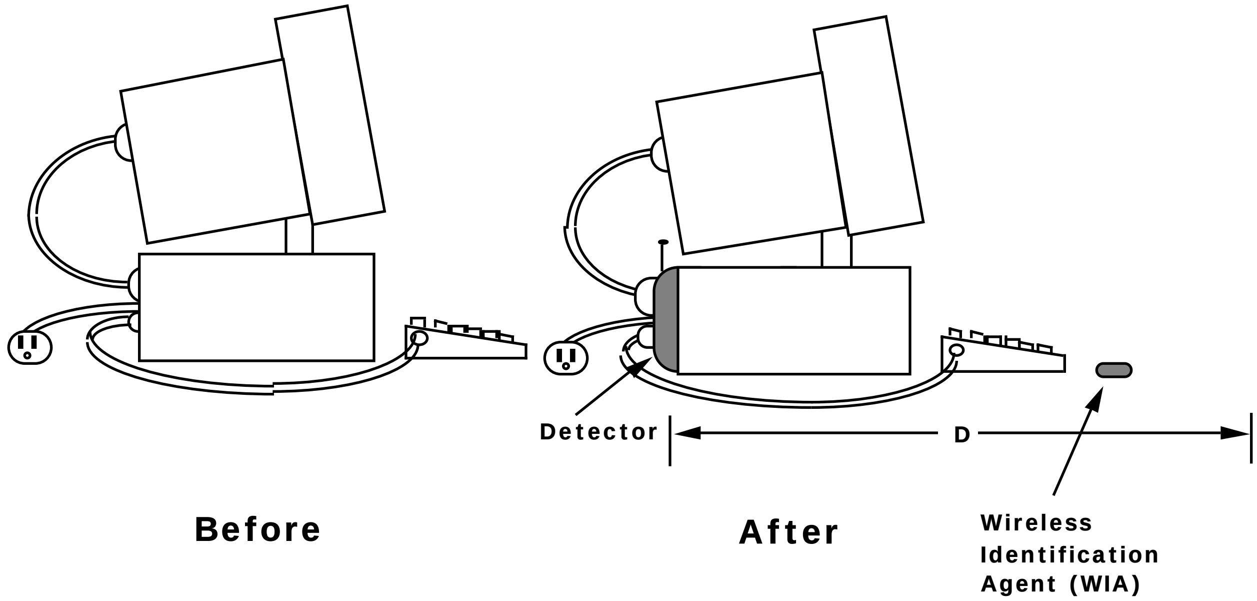


Figure 2. Workstation before and after installation of (conceptual) wireless identification system.

Landwehr, C.E. Protecting Unattended Computers Without Software. *Proc. Thirteenth Ann. Computer Security Applications Conf*, San Diego, CA, Dec. 1997, pp.274-283

Simple software integration: The Detector should require minimal changes to host software (preferably none, but for specialized applications such as a medical information system or a command and control system, some minor modifications might be tolerated).

Environmental constraints: The Detector must be able to receive signals in a typical office environment

Accuracy: The Detector should produce low rates both of false positives (accepting an unauthorized user) and false negatives (rejecting an authorized user).

Low cost . Agent plus Detector should retail under $150.

Capacity: A Detector should be able to enroll and distinguish at least two Agents.

Tamper resistance: A Detector should resist moderate tampering, and it should make tampering evident to its authorized user.

4. Design Issues

Choice of protocol. The protocol to be used over the wireless link must be carefully chosen. If the goal is to have a widely available, competitive set of devices to choose from, all will have to share this protocol. The number of bits that need to be transmitted in a challenge and a response will affect the identification time and will affect power consumption. If there is to be a family of devices, perhaps providing different levels of resistance to various threats and differing levels of authentication, the protocol will need to have some flexibility. For establishing feasibility of the entire approach, the protocol used in an initial prototype needs to be as simple as possible while meeting the specified requirements. If the requirement of resisting replay attacks is to be met, some form of challenge-response protocol seems essential.

Frequency range and signal format: Infrared (IR) would be feasible for Agents that are worn visibly, but is not suited for in-pocket operation. This constraint suggests radio frequencies (RF) will be most suitable. Different family members could use different frequencies and signal formats, but allowing this will limit interoperability of family members. The technology chosen should resist simple sorts of jamming attacks.

User interface: If simple theft is to be resisted, the user will need to authenticate herself to the device somehow. There could be a small keypad for entering a PIN, but the intended small size of the device could make this awkward. A possible alternative is a one-button interface for entering a PIN as a sequence of button-presses, analogous to a telegraph key. Some family members might be designed to use a plug-in facility, so that once a day (perhaps when the user arrives at the office) the user plugs the device into a system-provided port that initializes the device with appropriate algorithms and keys for the day. This would render lost or stolen tokens inoperative after at most 24 hours.

Power: a small lithium battery backed by solar cells seems a likely choice for the Agent, though an easily replaceable battery with backup key storage might suffice. The Detector can draw power from the host system through the signaling interface or from a wall socket, if it provides power to the workstation.

Detector-Host system interface: This is a key element, but difficult to specify because host systems will differ, both physically and functionally.

For an environment where a user normally works at a single, personal workstation for an extended period, but may occasionally leave during the day, it would suffice for the user to login once on arriving and then, each time the user walks away, the Detector could initiate locking of the screen and keyboard until the same user returns. This kind of operation could be supported through a component attached to the host's monitor output and keyboard input. When a user walks away, the interface could blank the screen and block input from the keyboard until the same user returns. The Detector would ignore other authorized users until the original user returned. If the original user disappeared for an extended period (and the Detector also controlled power to the workstation), the Detector could power cycle the workstation to force a re-boot and then be ready to accept a new authorized user.

In a hospital environment, a physician might like to walk up to any terminal and use it immediately. This behavior might be supported by requiring the physician to log in once in the morning to activate her account, which would permit her to access any workstation connected to the hospital's internal network during the day. This operation would clearly require support from the system software that would not be required in the former case. Also, in this case the user's walking away from a workstation should result in clearing the workstation's state in preparation for a new user.

Issues involving several users and several detectors close by:

How can a Detector resolve responses from different Agents (i.e., avoid or tolerate collisions)?

If two or more authorized users approach a workstation together, both of their WIAs might respond simultaneously to a challenge from the Detector. A unit number (different for each WIA) can be used to introduce a delay in the response of a Detector. Units with different unit numbers would introduce different delays before responding to a challenge, so collisions would be less likely. Alternatively, the WIA could randomize the delay based on unit number; this would avoid having (say) low numbered units always having priority in capturing an idle machine.

How can we avoid multiple responses to a currently active Detector?

If an active Detector broadcasts a challenge that is distinguishable from an idle Detector, Agents could respond differently in each case. One bit would suffice for this distinction, but it might be useful for an active Detector to broadcast the unit number of the Agent that has captured it. Then each Agent would listen for (and respond to) either its own ID number or a broadcast from an idle Detector. In this case, we might want the Agent to keep one bit of dynamic state information indicating whether it has captured a Detector or not. If not, it would respond to broadcasts from idle Agents; if so, it would only respond to challenges addressed to it specifically. Once set to respond only to addressed challenges, this bit could be reset to respond to broadcasts if no challenge addressed to the machine were received within some predefined window. Since the Agent already needs a clock to determine when reauthentication of the user is required, this new requirement should not affect the system hardware requirements.

If two or more users approach an idle machine, will it be apparent which one has captured it?

Unless there is visual feedback as to which device has captured the computer, this could indeed be a problem. A possible solution in this case would be to have the captured WIA beep, requiring the Agent to have sound output. If the WIA were integrated in a pager, other output modes (e.g. vibration) might be available.

Timing of signals

Figure 3 (below) shows a potential layout of signal timing for the WIA and the Detector, as follows:

$T_{re\text{-}ident}$ time interval for user to re-enter PIN to the WIA (estimated 24 hours).

T_p time between polls sent by Detector

T_d: detection interval for a Detector: no valid response from current active user within this interval causes Detector to blank screen and lock keyboard

$T_{re\text{-}init}$ time interval after which locked Detector resets itself to idle (potentially reboot workstation)

A response from the WIA between T_d and $T_{re\text{-}init}$ causes the Detector to unlock the keyboard and monitor and to start a new time interval.

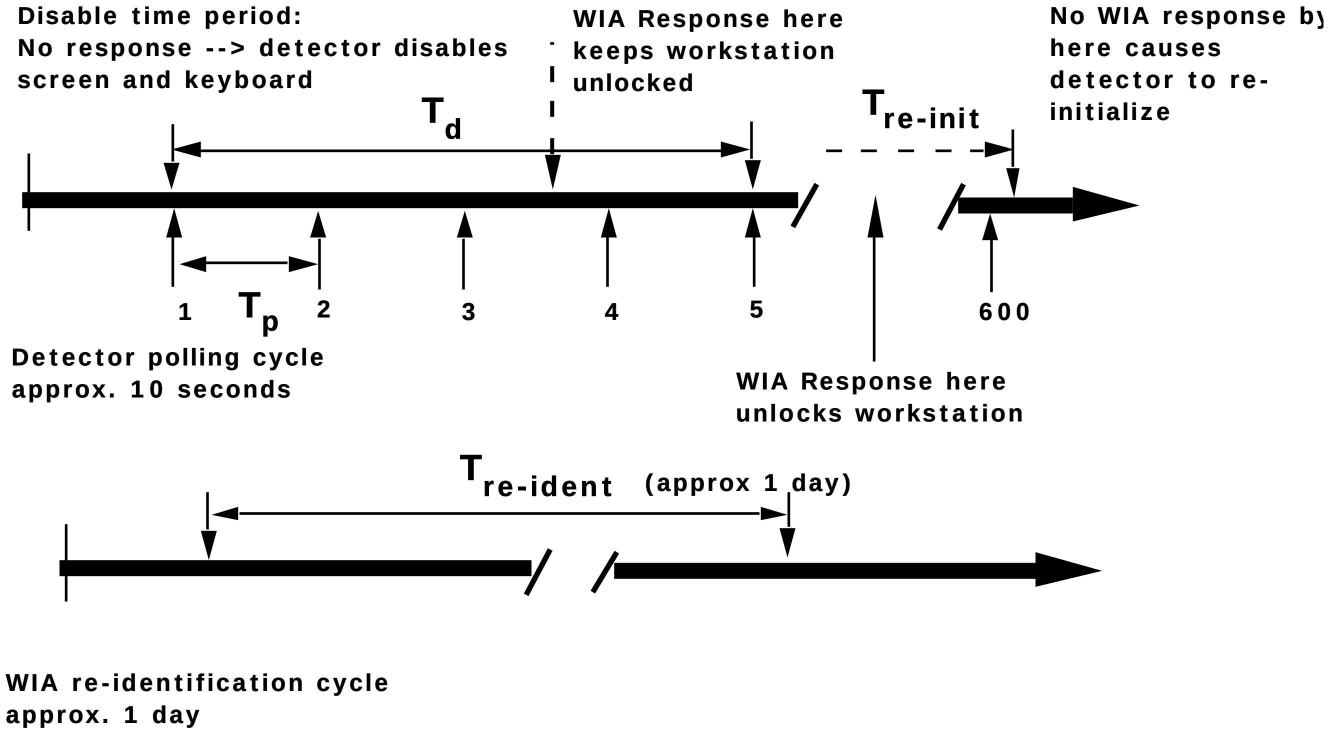

Disable time period:
No response --> detector disables screen and keyboard
WIA Response here keeps workstation unlocked
No WIA response by here causes detector to re-initialize
T_d
$T_{re\text{-}init}$
1
T_p
2
3
4
5
600
Detector polling cycle approx. 10 seconds
WIA Response here unlocks workstation
$T_{re\text{-}ident}$ (approx 1 day)
WIA re-identification cycle approx. 1 day

Figure 3. Possible Timelines for Detector and Agent.

Landwehr, C.E. Protecting Unattended Computers Without Software. *Proc. Thirteenth Ann. Computer Security Applications Conf*, San Diego, CA, Dec. 1997, pp.274-283

5. Prototype Specification

To test the viability of the concept, we develop a simple specification for a prototype Agent and Detector.

5.1 Protocol

Symmetric crypto version

Each Wireless Identification Agent (WIA) has a clock and stores

a PIN,
a system identifier,
a unit number,
its secret (e.g. DES) key,
an idle/capture bit, and
a saved clock value.

Each Detector has a clock and stores

a system identifier,
a list of unit numbers valid for use on its attached workstation,
a list of keys, one per unit number,
an idle/captured bit,
a "capturing unit" number, and
several saved clock values.

An idle Detector periodically broadcasts a challenge in the form of a system identifier.

An idle WIA checks the system identifier and if it matches, encrypts its clock, system, and unit number under its secret key and broadcasts the resulting ciphertext along with its cleartext system and unit numbers.

The Detector encrypts the received cleartext system identifier and unit number and its own clock value using the key of the indicated unit (and possibly preceding and following clock values as well, to allow for synchronization) and compares the results with the received encrypted value. If there is a match, the WIA is identified and it captures the Detector; the Detector sets its "captured" bit and records the unit number as "capturing unit." If the match fails, the Detector remains in the idle state.

A captured Detector broadcasts an acknowledgment in the form of system identifier, unit number, and (system identifier, unit number, and deciphered clock value plus one) encrypted under the unit number's key. The WIA, having responded to the open challenge, saves the clock value used and listens for the Detector's response (system number, unit number, encrypted string). On receiving the response, it decrypts the encrypted string and checks the clock value to see if it corresponds with the stored value plus one. If it does, the WIA sets its captured bit, beeps, and now will respond only to probes that include this system and unit number. On each such probe, it encrypts its (system, unit, clock value) and transmits it.

Release of captured WIAs and Detectors could be arranged either through timeouts or through a separate user act (e.g. by pressing a WIA button) to signal the end of a session. The system would automatically protect the workstation as the user came and went during the course of the day. At the end of the workday, the user might signal the end of a session.

Note: If the WIA can be plugged in for re-initialization daily (e.g. when user arrives at work) then new key and unit numbers can be assigned to make devices stolen or lost the previous day ineffective. This could eliminate the need for the user to login to the WIA. However, Detector key lists would also have to be updated

Asymmetric crypto version

Each Agent stores

a Secret Key,
an identifier, and
Detector Public Key (same for all Detectors).

Each Detector stores the public keys for Agents it will accept.

An idle Detector periodically broadcasts a challenge.

An idle WIA uses Detector Public Key to decrypt the challenge; it then re-encrypts the challenge under its Secret Key and broadcasts it together with its identity.

Detector checks identifier, selects Public Key to use, deciphers the returned challenge.

If challenge OK, the Detector is captured by this WIA, and operation proceeds similarly to the symmetric crypto example.

5.2 Example Identification Agent Functions and Interfaces

External Interfaces:

1 button for PIN entry
1 on/off switch for inactivation
1 audible beep tone to alert user as appropriate
1 signal light, (optional) to indicate successful / unsuccessful PIN validation and (perhaps) data transmission activity
Small lithium battery for primary power
Solar cells for backup power
Radio transmitter and receiver with internal antennae

Functions:

```
While switch turned on,
If PIN required, flash light or beep, read button presses,
    compare with stored PIN. Signal success or failure
    through beep or signal light.
    If failure, re-initiate (up to M consecutive failures;
    then disable)
End If
```

```
While (non-specific challenge received),
    compute response, wait random interval, transmit
      response
    listen for proper acknowledgment
    if proper ack received, set captured
        While captured, listen for challenge, generate
          reply
            if no challenge within $T_{re-init}$, set idle
        End While
End While
```

5.3 Example Detector Functions and Interfaces (e.g., for Macintosh)

Interfaces

Male/female 8 pin mini-DIN plug (to terminate keyboard and pass through)

Male/female 15 pin connector (to terminate monitor and pass through)

Radio transmitter and receiver

Functions

```
While Detector powered up, set Detector state to idle
    While idle
      Wait to end of polling interval ($T_p$)
      Set re-init timer to zero;
      Generate non-specific challenge
      Listen for response
      If response received prior to $T_p$, check for
              validity and identity of respondent
          If valid, enable keyboard and display;
              save respondent identity, and
              set Detector state to captured,
          While captured,
              generate challenge specific to
                respondent,
              wait for $T_p$ ; listen for responses
              If no valid responses after $T_{re-init}$,
                disable keyboard and display;
                set idle
              Else if no valid response after $T_d$ ,
                disable keyboard and display,
                increment re-init timer
              Else if response from new respondent
                and not from saved respondent,
                ignore (possibly logout saved
                        respondent, clear screen)
              Else if end-session-signal from saved
                respondent,
                clear captured and set idle
          End While
    End While
End While
```

6. Applications and Family Members

Laboratory. Users typically don't share workstations, which may be UNIX boxes, Macintoshes, or PCs. They login in the morning and remain logged in all day; sometimes overnight. But they do leave their workstations and walk away, sometimes for long periods, without logging out. Users are required to wear badges, but there is minimal physical security.

Identification Agent Family Member: All Agent requirements listed in Section 3.1 apply, except it need not be embeddable. The need to resist replay and theft will depend on the value of the information at risk and the general physical access control on the office environment.

Detector Family Member: The requirements in Section 3.2 apply, generally. In the best case, there should be no modifications to host hardware or software. The degree of tamper resistance or tamper evidence will depend on the local physical environment and the value of the information at risk.

Hospital: In this environment, users frequently share workstations (most likely these will be PC's, possibly tailored for use with an in-house system). users are highly mobile and probably wear badges. There will be minimal physical security on entry/exit to building.

Identification Agent Family Member: The requirements are similar to the laboratory example; the device need not be embeddable but otherwise the requirements listed in Section 3.1 apply.

Detector Family Member: In this environment, minor hardware modifications, if similar for all workstations, might be feasible. Tamper resistance would be improved by placing the Detector hardware on a card that could be inserted within the PC cabinet. Minor software modifications to hospital information system might be feasible as well.

Command Center: Physical access to the workstations should be well controlled. Partly for this reason, and for user convenience, the system may not require user login.

Identification Agent Family Member: Again, all requirements from Section 3.1, with the exception of embedability, are relevant.

Detector Family Member: Detector requirements are similar to those in the hospital example. Simple hardware modifications and minor software modifications to the command information system may be acceptable.

7. Security Model and Assurance Argument

An assurance argument is intended to make explicit the basis for believing in the security of the system. It starts with a security model expressed as sets of assumptions and assertions for the system as a whole and for individual system components. A component is responsible for enforcing the assertions assigned to it and for listing as assumptions those properties it requires of other components or the outside world in order to assure secure operation. A predicate should appear in the list of assertions only if its falsification would imply a security violation. The set of assumptions that cannot be matched against assertions is a (partial) indication of the remaining vulnerabilities of the system. The basis for believing each assertion will in fact hold during system operation is the substance of the assurance argument. This security modeling approach is demonstrated in [11] and assurance arguments are proposed as a basis for system certification in [12]. The approach is not formal, but it can be rigorous. Figure 4 outlines the assurance argument for the prototype.

7.1 Definitions:

Workstation: a computing device with attached screen (output only) and keyboard/mouse (input only), and possibly connected to a network.

Wireless Identification Agent: (WIA) a small portable device with a receiver, transmitter, crypto engine, power source, PIN, key, clock, sound output, PIN input, on/off switch. The device has a specified re-identification period $T_{re\text{-}ident}$.

Detector: A device attached to a workstation with a receiver, a transmitter, a clock, a crypto engine, a list of keys of the WIAs of users permitted to employ it, and a record of the current user, if any. It is interposed between the keyboard, the screen, and the computer in such a way that it can enable or disable communication between the computer and those components, and it can blank the screen. The Detector has a specified range *D*, a polling interval T_p, and a detection interval T_d. A Detector is *idle* if it has no identified user at present; otherwise it is *captured.*

Identified user: an individual who possesses a Wireless Identification Agent for which the PIN has been successfully entered since the last prompt and within the last day.

7.2 Model of Operation

A System Administrator attaches a *Detector* to each *workstation* and issues a *Wireless Identification Agent* (WIA) containing a key unique to that WIA to each user, provides the user with the PIN for that WIA, and provides the key to the Detectors attached to workstations this user is permitted to employ. Once per $T_{re\text{-}ident}$ or when prompted by the WIA, the user enters the PIN, becoming an *identified user.* When an identified user approaches within *D* of an *idle Detector* that he or she is permitted to employ and signals "login", the screen and keyboard are unlocked, and the Detector is made to be *captured* for this user. If the Detector fails to get a valid response from the current user's WIA within period T_d, the Detector blanks the screen and disables the keyboard, but the workstation is still *captured.* If the user signals completion (logout) or the Detector fails to receive a valid response from the user's WIA within a longer period $T_{re\text{-}init}$, the Detector returns to *idle*. If the user returns within $T_{re\text{-}init}$, the Detector restores the screen and keyboard connections. To remove a user, the System Administrator removes the user's key from the key lists of all Detectors.

7.3 System Assertions

There is no overarching system component that can enforce system properties, only individual components connected together, so any assertions at this level must be implied by the assertions to be enforced by individual components and the properties of their interconnections.

1. An identified user will not be (with probability less than p_I) denied service by an idle Detector.
2. An unidentified user will not be (with probability less than p_{II}) provided service by a Detector.
3. Replay of valid challenge-response sequence will have no effect.
4. Stolen WIA will provide system access for at most $T_{re\text{-}ident}$

7.4 System Assumptions

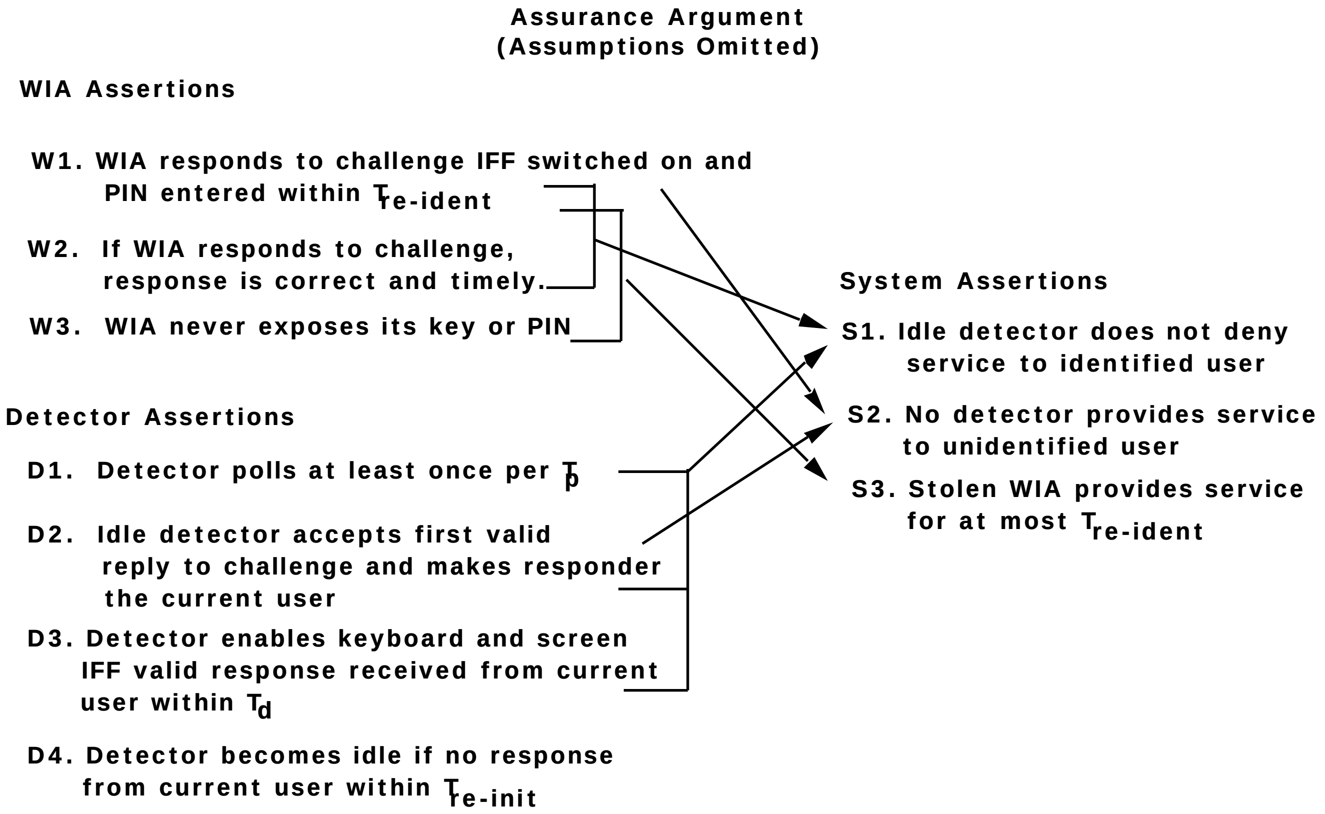
1. System administrator properly dispenses and configures WIAs, Detectors, keys, and PINs.
2. Crypto algorithms have not been broken.
3. WIAs, Detectors, and their connections have not been tampered with.

7.5 Wireless Identification Agent Assertions

1. WIA responds to challenges if and only if PIN has been correctly entered within $T_{re\text{-}ident}$ and WIA is switched on.
2. If WIA responds to challenge, response is correct and timely.
3. WIA never exposes its key or PIN

7.6 Wireless Identification Agent Assumptions

1. Power source remains adequate.
2. Device physical integrity is maintained.
3. Electromagnetic environment remains within design parameters.
4. Detector polls and maintains idle/captured state as designed.



Assurance Argument
(Assumptions Omitted)
WIA Assertions
W1. WIA responds to challenge IFF switched on and PIN entered within $T_{re-ident}$
W2. If WIA responds to challenge, response is correct and timely.
W3. WIA never exposes its key or PIN
Detector Assertions
D1. Detector polls at least once per T_p
D2. Idle detector accepts first valid reply to challenge and makes responder the current user
D3. Detector enables keyboard and screen IFF valid response received from current user within T_d
D4. Detector becomes idle if no response from current user within $T_{re-init}$
System Assertions
S1. Idle detector does not deny service to identified user
S2. No detector provides service to unidentified user
S3. Stolen WIA provides service for at most $T_{re-ident}$

Figure 4. Assurance argument based on prototype specification

7.7 Detector Assertions

1. Detector polls at least once every T_p

2. Detector in idle state accepts first valid reply to challenge and makes responder current user.

3. Detector enables keyboard and screen if and only if valid response to challenge has been received from current user within T_d.

4. Detector becomes idle if no response from current user within $T_{re-init}$

7.8 Detector Assumptions

1. Electromagnetic environment remains within design parameters.

2. WIAs respond to polls as designed.

3. Power source remains adequate.

4. Device physical integrity is maintained.

8. Initial Prototype Design and Results

Based on the requirements and design discussions above, we investigated the commercial marketplace further and developed an operational prototype of such a system. The prototype combines a modified version of a remote key entry system provided by TRW Automotive and a custom designed and built Detector box. The initial prototype does not implement the requirements for resistance to replay or theft of token, and it is based on a Detector that passively receives signals generated by active Agents. Nevertheless, it successfully demonstrates the basic functions envisioned. A second prototype is under development that will incorporate lessons learned from the initial build. Twenty-four copies of the second prototype are to be installed in an office environment for test purposes.

9. Summary and Conclusions

An unattended workstation at which a user has already logged in represents a security risk. Those with physical access to such a workstation may ordinarily observe the display and manipulate the keyboard and mouse to display other information, send messages, or cause other changes to the system. We have suggested a low cost, easy-to-use approach that can reduce the risk of such attacks without inconveniencing the authorized user. The approach requires only replugging the workstation's monitor, keyboard, and power cables, so it could be applied to PC, Macintosh, and UNIX systems equally. We have suggested a family of systems embodying this approach and noted how different family members might be developed to satisfy the requirements of office, hospital, and command center environments. A potential design for the system and the characteristics of an initial prototype implementation have been described. A patent application covering various aspects of this system has been filed, and the technology is available for licensing.

Although the particulars of this problem and solution are of interest, we think there is a more general conclusion to be drawn from this work: the computer security applications community needs to focus more closely on how users will interact with security technology and how the technology can mesh both with old and new computer systems. We need to develop technology that we ourselves are willing to use and that reduces the risks we see in our own backyards. This attitude will help us make a real difference to our customers.

Acknowlegments

It is a pleasure to acknowledge the contributions of Dan Latham of Kaman Sciences Corporation, who collaborated on the design of the prototype system and developed and built the prototype. Thanks also to the anonymous referees for their suggestions, which improved the presentation.

References

All URLs checked for validity September, 1997.

[1]*Redefining Security*. Report to the Secretary of Defense and the Director of Central Intelligence. Joint Security Commission, J.H. Smith, Chairman. Feb 28, 1994. p. 109.

[2] See, for example, URLs for fingerprint identification products:
Security Print:
http://www.hht.com/bus/secure/bio~1.html
National Registry, Inc.: http://www.nrid.com/

[3] Daugman, J.G. High confidence visual recognition of persons by a test of statistical independence. *IEEE Trans on Pattern Analysis and Machine Intelligence, Vol. 15* , 11 (Nov. 1993).pp. 1148-1161.
Iris pattern identification products:
IriScan http://www.iriscan.com/ and
Sensar: http://www.sensar.com

[4] Speaker verification information:
Speaker Verification Application Programming Interface Committee: http://www.srapi.com/svapi/
Products:
T-Netix:
http://www.t-netix.com/CompanyOverview.html
Veritel: http://www.weverify.com/

[5] Hand geometry:
Rognition Systems, Inc.:
http://www.recogsys.com/hkey.htm

[6] U.S.Biometric consortium,
http://www.vitro.bloomington.in.us:8080/~BC

[7] Fastoll web site: http://www.Fastoll.com/

[8] Speedpass web site:
http://www.mobil.com/speedpass

[9] Olivetti Active Badge web site:
http://www.cam-orl.co.uk/ab.html

[10] SecurID token web site:
http://www.securitydynamics.com/solutions/products/tokens.html

[11] Landwehr, C.E., C. L. Heitmeyer, and J. McLean. A Security model for Military Message Systems. *ACM Trans. on Computer Systems Vol. 9*, No. 3 (Aug. 1984). PostScript available at
http://www.itd.nrl.navy.mil/ITD/5540/publications/indexbefore1990.html

[12] Payne, C., J. N. Froscher, and C. E. Landwehr. Toward a comprehensive INFOSEC certification methodology. *Proc. Sixteenth National Computer Security Conference*, Baltimore, MD. Sept. 1993. pp.165-172.[PostScript available at
http://www.itd.nrl.navy.mil/ITD/5540/publications/index1993.html/