Personal Authentication Domains for Automatic Pervasive Device Authorization

James Giles and Reiner Sailer
IBM T. J. Watson Research Center
Hawthorne, NY
Outline

- Motivation
- Goals
- Our Solution
- Security Evaluation
- Questions & Comments
Motivation

Many pervasive devices

Want multiple devices to access protected services, without much user intervention
Motivation

- **Could use known authentication and authorization methods:**
  - Enter credentials on each use on desired devices
  - Store credentials long-term in desired device

- **But these have problems for pervasive devices…**
  - Devices may not protect stored long-term credentials
    - loss, theft, weak security
  - Devices may not have convenient user interface
  - Not convenient for multi-device interaction
    - E.g. Use PDA to tell medical server to initiate pace maker monitoring and diagnostics. Pace maker data sent to medical server, with warning messages sent to a watch-based display/alarm.
Goals

- Access to services should be provided only to devices that are authorized by the user.
- It should be easy for the user to authorize many devices automatically, without necessarily interacting directly with the devices.
- Authentication security for future access in case of compromise
  - User IDs & passwords remain confidential
  - Access control remains effective for compromise of pervasive devices.
- A solution should use existing technology whenever possible.
Solution

Personal Authentication Gateway
- Security hub
- Provides security configuration and credentials to devices in the domain

Pervasive Authentication Domain
Solution

Services that require authorization
Domain membership for pervasive devices

- **Configuration**
  - Device is registered with the Authentication Gateway
  - Secret shared with the Authentication Gateway

- **Environment**
  - Device within a certain range from the Authentication Gateway
    - signal strength
    - distance as determined by GPS

- **Integrity**
  - Device has acceptable integrity measures
    - TPM-based hardware and attestation of the device’s integrity status
Solution architecture

Token Request typically includes challenge to pervasive device from the server.
Message Flow Example

Personal Authentication Gateway

Pervasive Device

Server

Service Request

ChallengeReq(NonceA)

TokenReq(NonceA)

Use Secret to create Token

TokenRes(T_{NonceA})

ChallengeRes(T_{NonceA})

ServiceResponse(Cookie)

Check Response

Store cookie for future authentication

ServiceRequest(Cookie)

Check cookie

ServiceResponse
Device registration

- **Token Client sends registration request to Personal Authentication Gateway**
- **Gateway assigns a SlaveID for the device in the domain**
- **Gateway constructs shared secret for the client**
  - SHA1 Secure Hash of a) master key known only to gateway, b) SlaveID, and c) DomainID
- **Shared secret and IDs transferred manually, or automatically to pervasive client**
Pervasive client procedures

- Requests access-controlled service
  - Receives challenge-response nonce from service
- Determines its DomainID, SlaveID, and shared secrets for all Pervasive Authentication Domains to which it belongs (obtained at registration)
- Broadcasts or directly sends Token Requests containing nonce to the gateways
- Resulting token is used to access service
Token Request structure

Networking Header (Broadcast or Addressed)
- Slave ID
- Domain ID
- Nonce – 128 Bit
- Slave ID
- Message Type=Request
- Token Request Parameters

DES Encryption protects against tampering
Personal Authentication Gateway procedures

- **Receive Token Request from client**
  - check Token Request integrity

- **Check policy database**
  - Is client (DomainID / SlaveID) allowed to receive credentials?
  - Is it still registered? Is it still secure (within distance, etc)?

- **Use SlaveSecret to decrypt**

- **Token computed for challenge-response nonce**
  - Returned to client in a Token Response
Token Response

- Networking Header (Addressed)
- Slave ID
- Nonce – 128 Bit
- Slave ID
- Message Type=Response
- Tokens and checksums

- Protected by DES encryption
- Replay protection through Request Nonce
Related Work

- **Credential sharing via many different types of communications (TCP/IP, PDA hotsync, etc.)**
  - Doesn’t securely authenticate devices

- **Security Assertion Markup Language (SAML)**
  - Can be used to exchange authenticate/authorization info between gateway and pervasive devices

- **Kerberos allows automated, repeated authentication**
  - Kerberos ticket can be used by devices while keeping long-term secrets away from the pervasive device

- **Dynamic Host Configuration Protocol (DHCP)**
  - Provides dynamic configuration info for networks
  - Doesn’t do user configuration or provide disclosure protection
Security Evaluation

- Protects against loss of long-term secrets for services if a pervasive device is lost, stolen, or attacked
  - Pervasive devices never receive long-term tokens
  - Devices away from the gateway are excluded

- Protects against token disclosure to devices outside the domain
  - Access is only through DES encrypted tokens using private SlaveSecret
Summary

- Allows many pervasive devices to participate in the authorization of a user
- Personal Authentication Gateway manages authorization for devices in the domain
  - Devices automatically configured and authorized
- Long-term credentials are protected against disclosure if a device is lost/stolen
- Much more convenient for user
  - Logging into one device allows interaction with all devices in the domain
Future Work

- **TPM-based attestation**
  - Personal Authentication Gateway can verify the device’s integrity
  - Send credentials if device passes integrity check

- **Integrate the security configuration and token requests directly into the DHCP protocol**
  - Allows Personal Authentication Gateway to handle both networking and security configurations
Questions
and Comments
The END