Usable Security:
Oxymoron or Challenge?

Diana Smetters
Security Group
Computing Science Laboratory
PARC

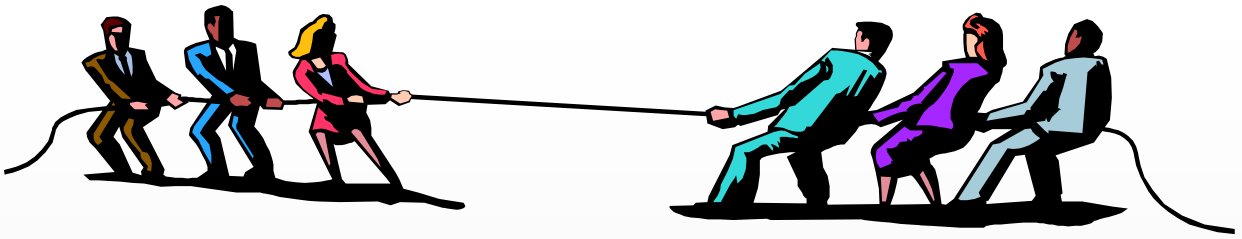
parc
Palo Alto Research Center

Quiz

- Do you have more passwords/PINs than you can remember?
- Have you ever gotten a phishing email?
  - *Have you ever gotten an email message where you weren't sure whether it was phishing or not?*
- Have you ever needed to do something but couldn't because you:
  - didn't know the password/key/PIN
  - couldn't get through the firewall
  - the permissions were set wrong
- Have you ever been asked a question by a piece of security software *that you couldn't answer?*

parc
Palo Alto Research Center

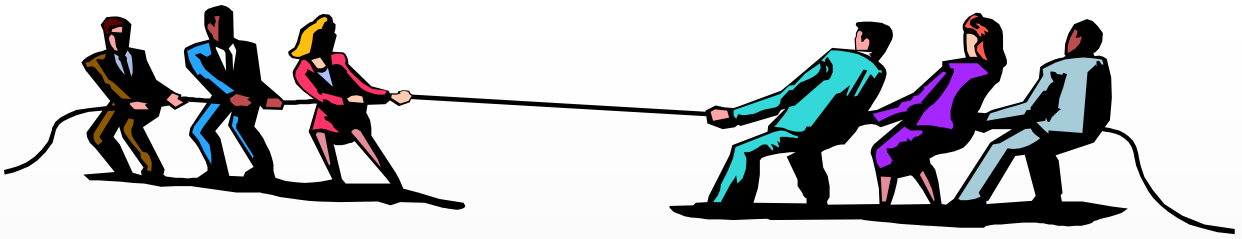
Problem



- Usability and security often seen to be at odds
  - if security gets in the way of getting work done people will turn it off
- Most deployed security technology not designed with usability in mind
  - security seen as far more important
- Can't add usability in "after the fact"
  - any more than you can security
- New realities require that you have both
  - no longer any safe spaces where you can ignore security
  - simply not cost-effective to ignore usability
    - » end user often the only one in a position to make security decisions
    - » systems are designed primarily to get real work done, not just to be secure

parc
Palo Alto Research Center

Approach



Research Question:

If you put usability first, how much security can you get?

- **Design security technologies that make usability easier**
  - reuse existing technology when appropriate
  - change it when it is not
    - » Don't just "make passwords better"
    - » Adding a better GUI to badly designed technology almost never good enough
- **Test systems with real users**
  - security researchers almost never have accurate intuitions about what is "usable"
- **Involve interdisciplinary project teams**
  - both security researchers and HCI researchers/fieldworkers

parc
Palo Alto Research Center

Things not to do: Passwords

parc
Palo Alto Research Center

Passwords

- cheap(er) and easy to implement
  - *so everybody uses them everywhere*
- people have trouble remembering them
  - *choose bad ones*
  - *reuse them*
  - *write them down*
- systems administrators try to compensate
  - *password quality rules*
  - *password change requirements*
  - *password lockouts*
- once you give them away, they're gone
  - *phishing attacks and other forms of social engineering*
  - ***no easy way for user to authenticate server***

parc
Palo Alto Research Center

How to make passwords more usable?

While not making them less secure?

Easy places to start:

- increase password lockout thresholds (e.g. 3->10)
  - *doesn't significantly change attacker's ability to brute force passwords*
  - *does dramatically drop lockouts from, e.g. CAPS LOCK*
- don't require frequent password changes
- **decrease the number of passwords people have**
  - *don't require passwords when you're not protecting anything*
    - » *e.g. accounts whose purpose is customer identification, not data protection*

These help usability, but not security.

parc
Palo Alto Research Center

Can we make passwords easier to remember, but not easier to guess?

- phrase-based mnemonic passwords (Yan, et al '05)
  - *pick the first letter of each word in a phrase*
    - » *& for "and", 4 for "four", etc.*
    - » *e.g. "Four score and seven years ago, our fathers" -> 4s&7ya,of*
  - *easier to remember than random passwords*
  - *as resilient to cracking attempts using standard dictionaries*

- graphical password systems (too many to mention)
  - *e.g. select a set of points in an image and select them again later (PassPoints)*
  - *…and many more*

Can we make passwords easier to remember, but not easier to guess?

No. We are too predictable.

- phrase-based mnemonic passwords
  - *everybody turns out to pick the same phrases (Kuo et al '06)*
    - » *"Oh I wish I were an Oscar Meyer Wiener…"*
  - *just make a phrase-based cracking dictionary*

- graphical password systems
  - *e.g. select a set of points in an image and select them again later (PassPoints)*
  - *everybody turns out to pick the same sets of points…*
    - » ***can build "graphical cracking dictionaries"***

parc
Palo Alto Research Center

What else can we do?

- two-factor authentication
  - *give users something else they need to present*
    - » *physical token, securid value, etc…*
    - » *typically implemented with **One-Time Passwords***
  - **attackers mount online "Man-in-the-Middle" attacks**
    - » *get the one-time password, present it to the server, give an error to the user*
- make it possible for user to authenticate server
  - *user picks a personal image, should only give password to server that knows the image*
    - » *SiteKey, Security Skins (Dhamija et al, '05)*
  - ***turns out users don't notice if image is missing** (Schechter et al '07)*

What else can we do?

- enlist software to help
  - ***browsers that attempt to recognize and discourage use of illegitimate sites***
    - » *IE7, Firefox, you name it*
    - » **depends on how strongly they "discourage"**
  - ***better brands of password managers***
    - » ***beyond just "adding a better GUI to passwords"***
    - » *keep users from reusing passwords*
    - » *automatically generate per-site strong password*
    - » *require users to remember only one master password*
    - » *warn/prevent users if they try to enter passwords into incorrect sites*
    - » *PassPet (Yee, et al '06), WebWallet (Wu et al '06)*
    - » ***requires users to use them***
      - *lots of variants available, with wide range of security properties – how is a user to pick?*
    - » ***still vulnerable to keyloggers, etc***

parc
Palo Alto Research Center

Example: Securing a Wireless Network

Balfanz, Durfee, Grinter, Smetters, and Stewart

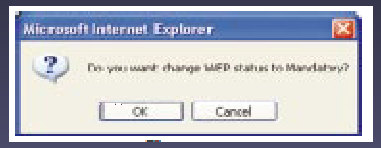
Usenix Security 2004

parc

Palo Alto Research Center

Wireless Usability versus Wireless Security

WEP: "Easy" but not secure



Turn WEP on…

…generate and select keys…

…configure cards and AP

WPA-PSK: More difficult, still has security problems

`0xa47a3471fbc843149038…`

Turn WPA on…

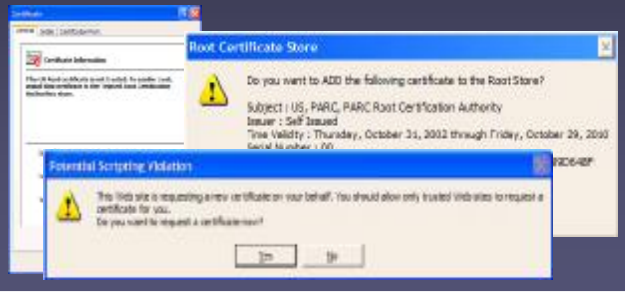
`0xa47a3471fbc843149038…`

`0xa47a3471fbc843149038…`

…configure cards and AP with long hex keys

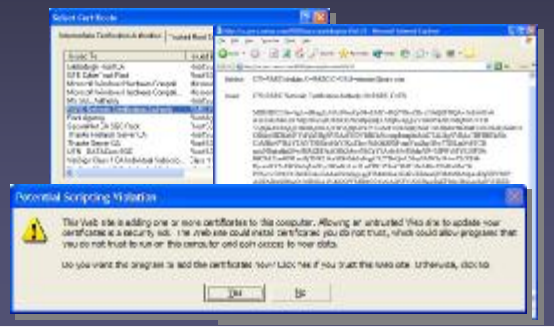
802.1x (EAP-TLS): Today's best available wireless security is too hard

User Generates Keys, Obtains Root Cert, Requests Certificate…



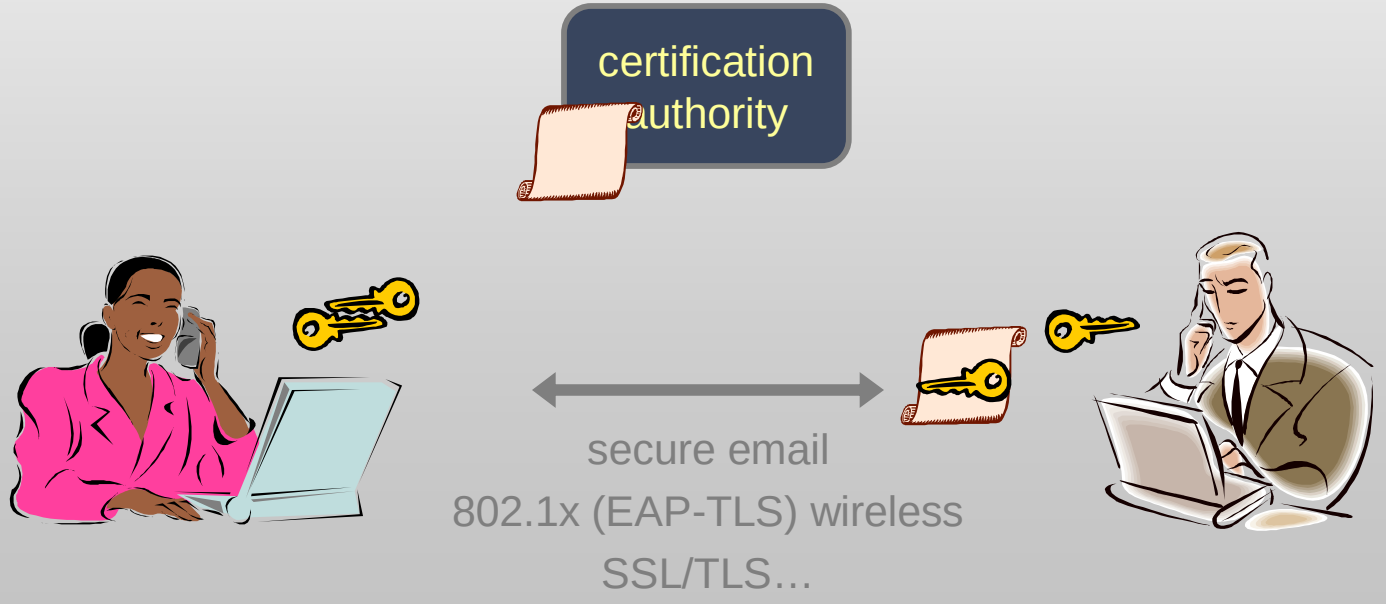
…Registration agent approves request…

…User Receives & Installs Certificate



Public Key Infrastructures

- Public Key Infrastructure technology can improve security and mitigate usability problems *once it has been deployed*.
  - Participants must generate key pairs
  - Public keys are digitally signed by a certification authority
  - Participants get *digital certificates* for authentication, encryption, …



parc
Palo Alto Research Center

38 Simple Steps for Enrolling in a WLAN with PKI-Based Security

parc
Palo Alto Research Center

Usable Wireless Security

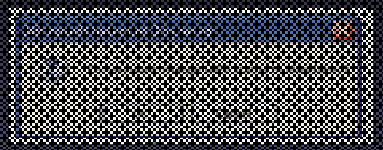
How do we build a highly secure wireless network easy enough for any user to set up?

- What does the user have to do?
  - Get an access point (AP) and turn it on
  - Tell her laptop which network to join
  - Tell the AP it's okay for her laptop to join that network
    - » can skip these last two steps (and just "fall" onto the network), but not with any security
- User should do no more than this.

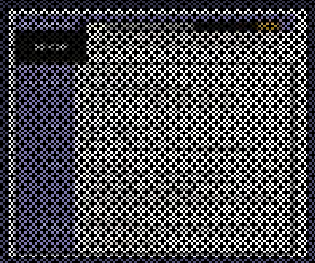
parc
Palo Alto Research Center

Wireless Security

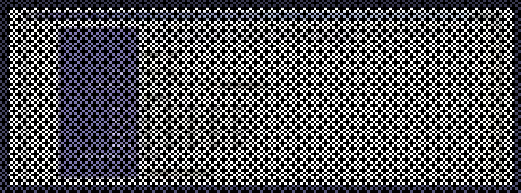
WEP: "Easy" but not secure



Turn WEP on…

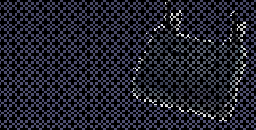


…generate and select keys…

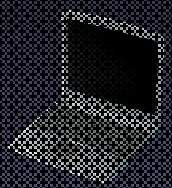


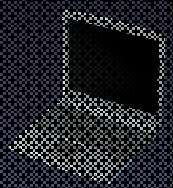
…configure cards and AP

WPA-PSK: More difficult, still has security problems



Turn WPA on…

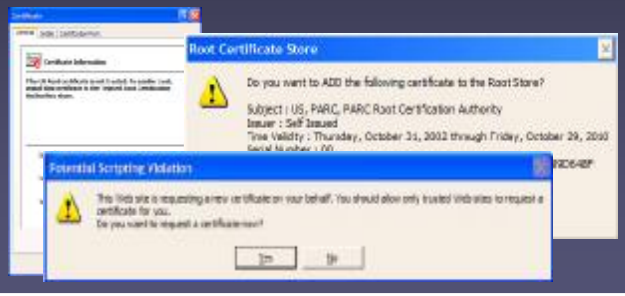




…configure cards and AP with long hex keys

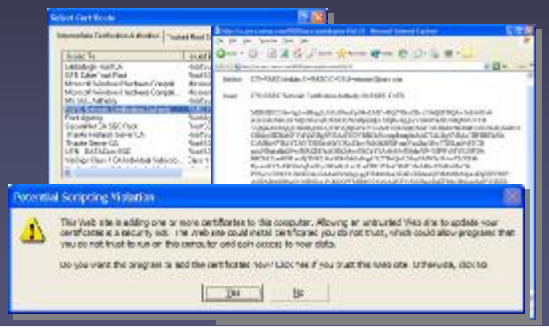
802.1x (EAP-TLS): Today's best available wireless security is too hard

User Generates Keys, Obtains Root Cert, Requests Certificate…



…Registration agent approves request…

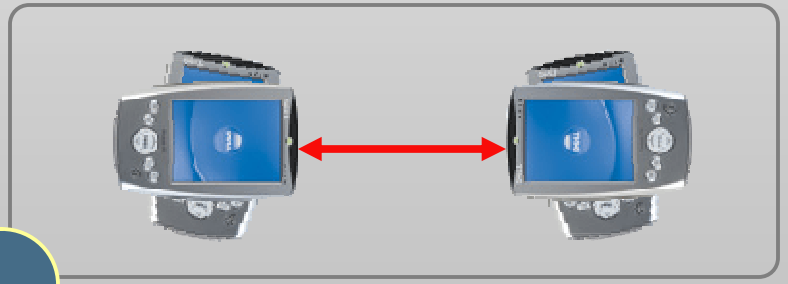
…User Receives & Installs Certificate



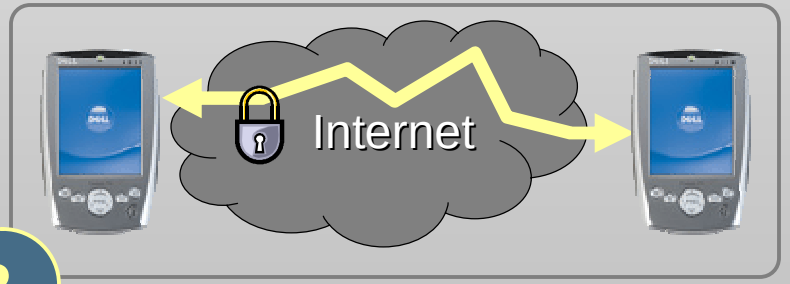
Authentication in Ad-Hoc Networks

Establishing secure communication requires *identifying communication partners* and *sharing trust information*

- Gesture-Directed Automatic Configuration
  - Intuitive gestures, e.g. pointing, identify communication partners
    - » "I want this handheld to speak to that printer"
- Location-Limited Channels (LLC)
  - Bootstrap trust from proximity: exchange cryptographic keys over out-of-band/secondary channel (IR, contact, audio, bodynet, etc…)



1. Use LLC to exchange addressing information and bootstrap trust

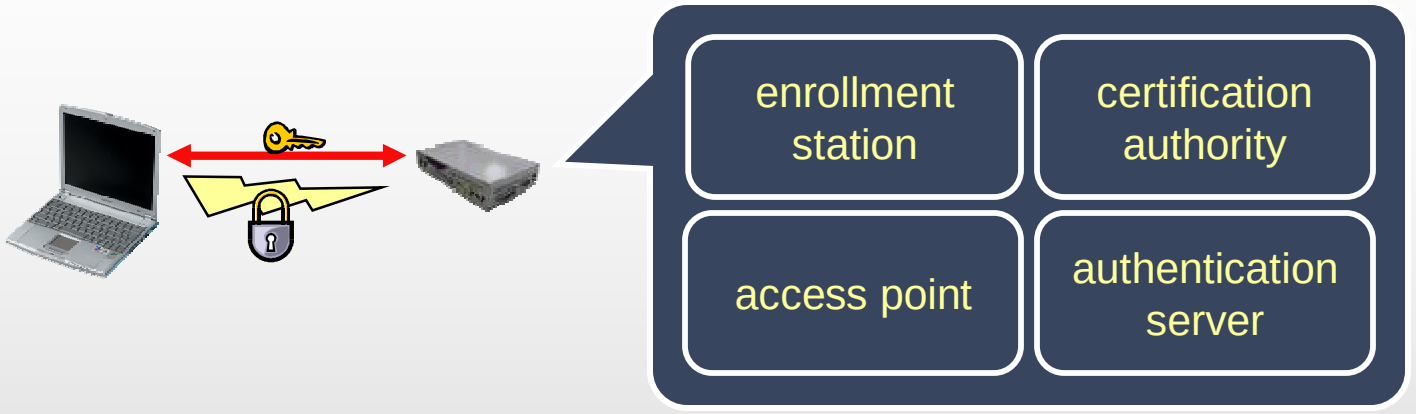


2. Find and communicate securely over any network

parc
Palo Alto Research Center

“Network-in-a-Box” (NiaB)

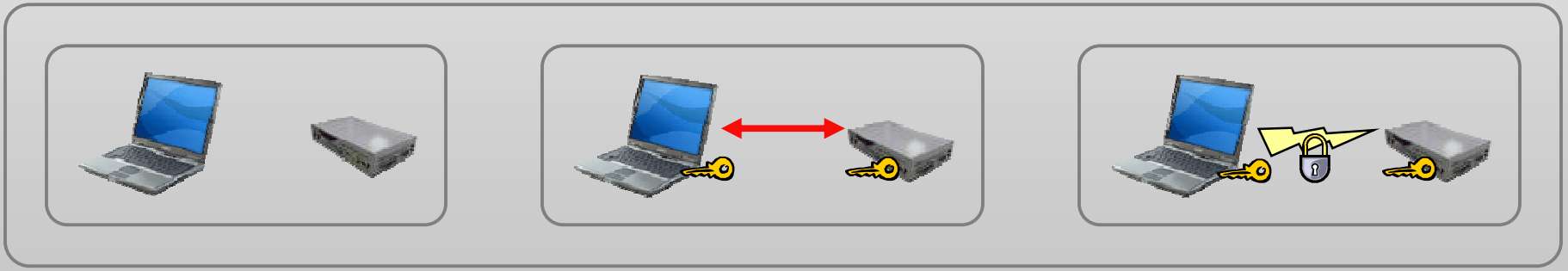
(Ballanz, Durfee, Grinter, Smetters, Stewart 2004)



- NiaB Access Point plays all key roles in 802.1x network:
  - automatically configures itself on first boot
    - » generates SSID for wireless network
    - » generates key pair & root certificate for Certification Authority
    - » configures and starts authentication server, access point, and web-based configuration and monitoring services
  - provides location-limited enrollment mechanism
    - » IR or USB based

Joining a Secure Wireless Network

1. Introduce laptop to access point using infrared
   - *Capture user intention to add this device to that network*
   - *Relate digital security to physical security*
2. Certification authority issues digital certificates
   - Automatic configuration: no input from user required
3. Laptop joins 802.1x-authenticated wireless network
   - Best wireless security available



Sets up a secure wireless network in under a minute

parc
Palo Alto Research Center

Joining a Secure Wireless Network

parc
Palo Alto Research Center

User Experience

- We conducted usability tests to compare a commercial Access Point (AP) against NiaB.
  - Results showed NiaB took less time, and fewer configuration steps

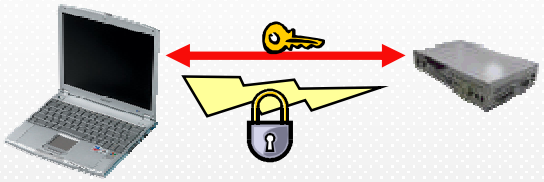
System	Time (min)	Steps
Commercial AP	9:39	14
Network-in-a-Box	0:51	2

  - Users rated NiaB *(1 = most positive, 5 = most negative)*
    - » easier to use
    - » a more satisfying experience (satisfaction)
    - » more confident that they configured security correctly (confidence)

System	Ease of Use	Satisfaction	Confidence
Commercial AP	3	3	2
Network-in-a-Box	1	1	1

Enterprise-Scale NiaB

Network-in-a-Box



enrollment station

certification authority

access point

authentication server

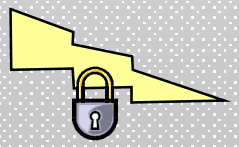
Enterprise NiaB

parc

enrollment station

certification authority

parc



access points

authentication server

parc

Palo Alto Research Center

Enterprise-Scale NiaB

- Simple, intuitive enrollment mechanism
  - User walks down to “enrollment station”, lines up IR, leaves.
- Simple, intuitive trust model
  - Users must request access to locked enrollment station room to request their wireless certificates.
- Certificates bound to capabilities, not identities
  - Certificate used for wireless access only
- No need for direct user interaction with certificates
  - Wireless clients are set up to use new certificates automatically, and to renew them when they expire

User Experience

- We conducted usability tests to compare traditional PKI + 802.1x enrollment against Enterprise NiaB.
  - Results showed ours took less time, and fewer configuration steps

System	Time (min)	Steps
Traditional PKI	140:00	38
Enterprise NiaB	1:39	4

  - Users rated NiaB *(1 = most positive, 5 = most negative)*
    - easier to use
    - a more satisfying experience (satisfaction)
    - more confident that they configured security correctly (confidence)

System	Ease of Use	Satisfaction	Confidence
Traditional PKI	5	4	4
Enterprise NiaB	1	1	1

parc
Palo Alto Research Center

Conclusions

- Can build practical systems that are highly secure and highly usable
  - Don’t assume you know what is usable
    - Security researchers are not good models for “typical” users
  - Use existing security tools when appropriate
    - PKI
    - Standard security protocols (802.1x, WPA, TLS)
  - Think about them in a different way
    - Small-scale “Instant” PKI
    - Authentication via physical proximity
    - Transparently automate handling of both security and wireless settings
      - users don’t think of security and management as separate activities

A Few General Principles

(Everybody's got to have a set…)

- Implicit Security
  - if an explicit user request requires security changes to effect, don't make them ask again
    - » no separate but (un)equal security interface
    - » doesn't mean security must always be invisible, just that it ought to be when it can
- "And Nothing Else" Security Policy
  - easy and intuitive metric for what a system ought to do
- Who are the users?
  - usability also impacts developers, designers, administrators…
- User decisions should be framed in terms of application tasks
  - that is all that they care/know about
  - pushes security back up to the level of the application
- Users don't need to know and love your abstractions
  - "share the pain" not a good design mindset

Thank you!

parc
Palo Alto Research Center

Extra Slides

parc
Palo Alto Research Center

Example: Secure Access to Remote Resources

Smetters, Balfanz, Durfee, Smith and Lee

Ubicomp 2006

parc
Palo Alto Research Center

The Problem

Want to seamlessly integrate and *personal* data and computing environments…

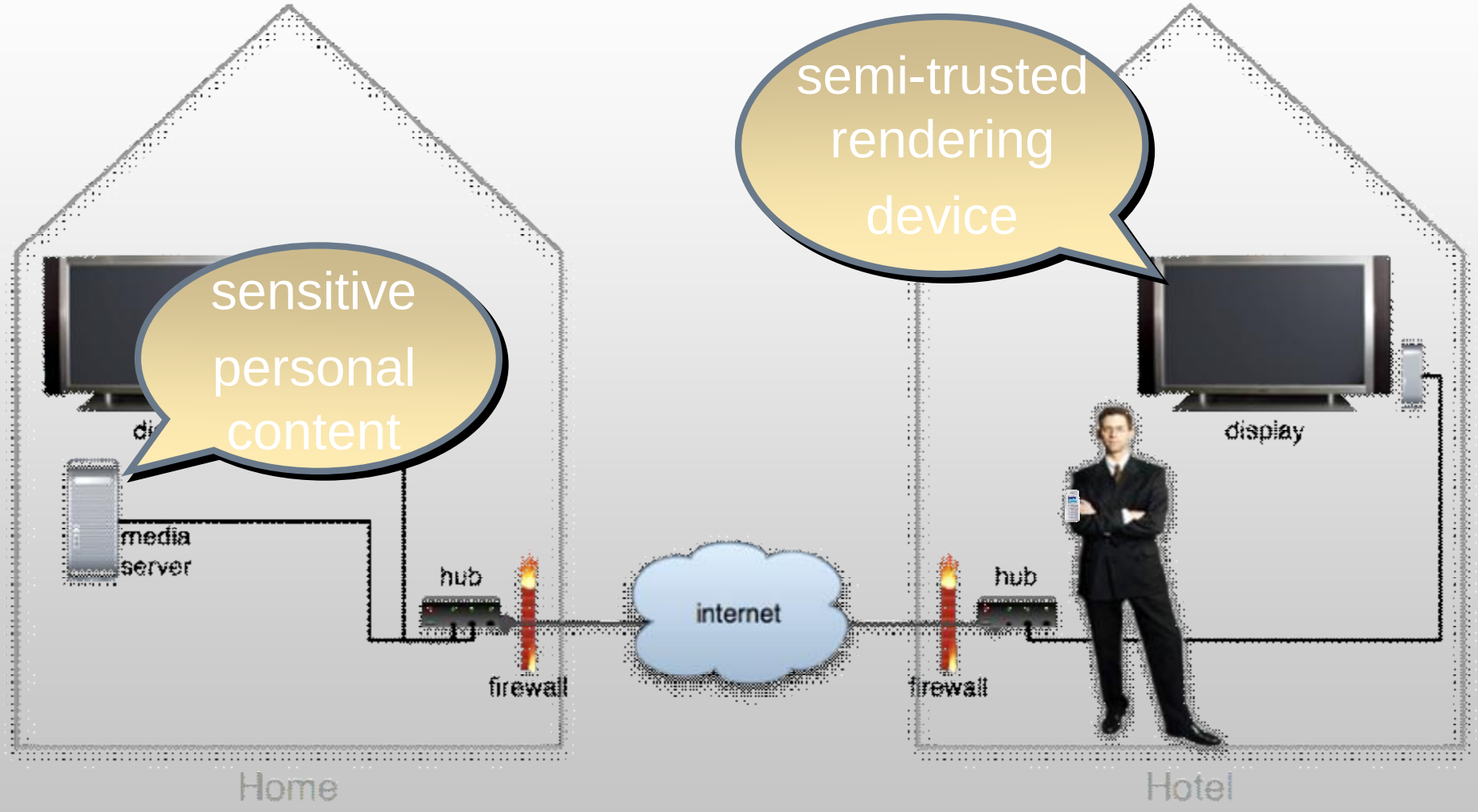
Work

Home

Family/Friends

parc
Palo Alto Research Center

Our scenario



parc
Palo Alto Research Center

Our scenario

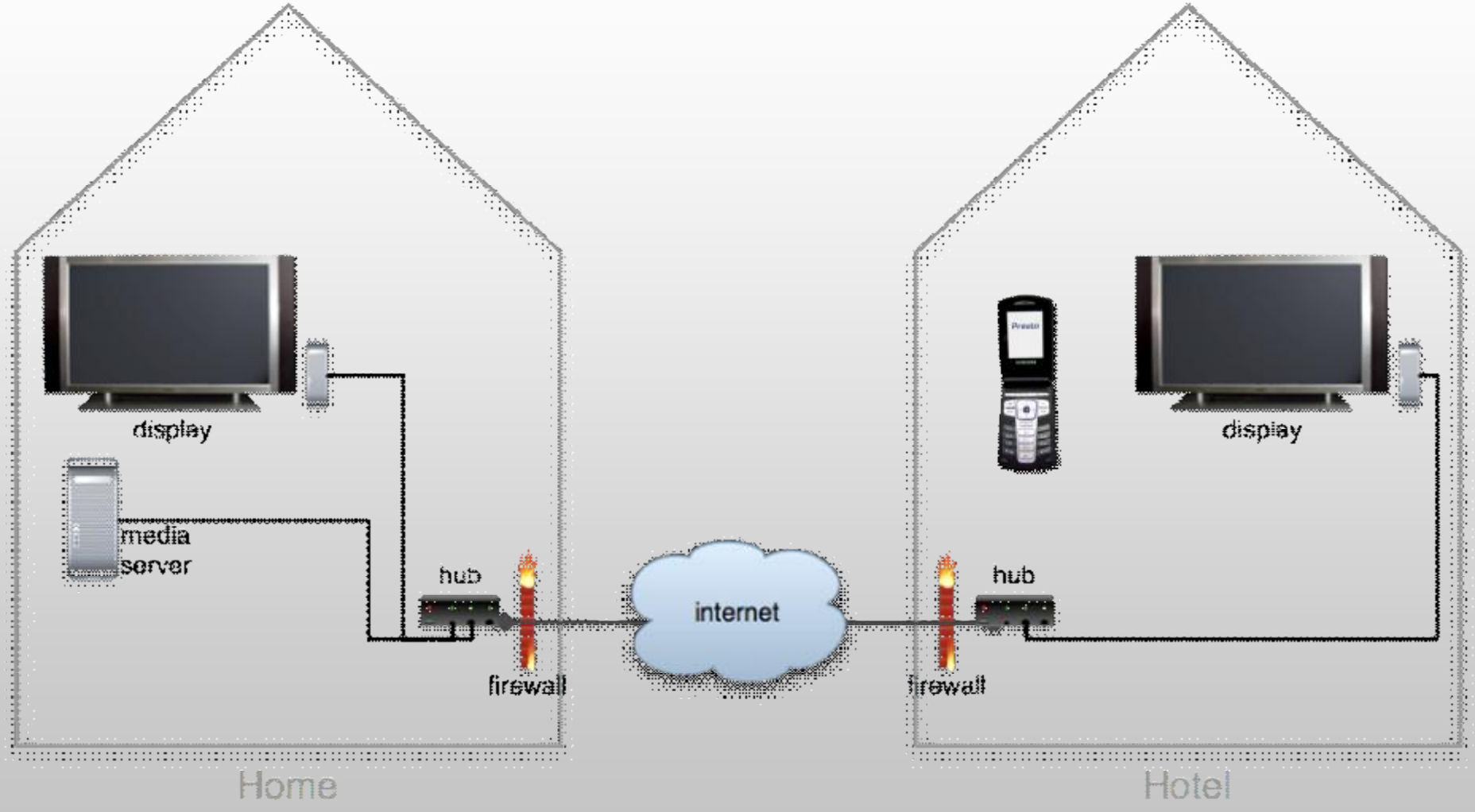
Requirements

- functionality
  - opportunistic access to encountered resources
  - universal access to personal resources
    - » no need to specify access policy *a priori*
  - minimize required user effort
- security
  - least privilege access boundary
    - » only the device the user selects can access
    - » can only access the resources the user indicates
    - » ***"and nothing else" policy***
      - implies requirements for authentication, encryption, etc
  - reliable and secure indicators of user intent
    - » implicit security
    - » ***trusted input path***

The Solution

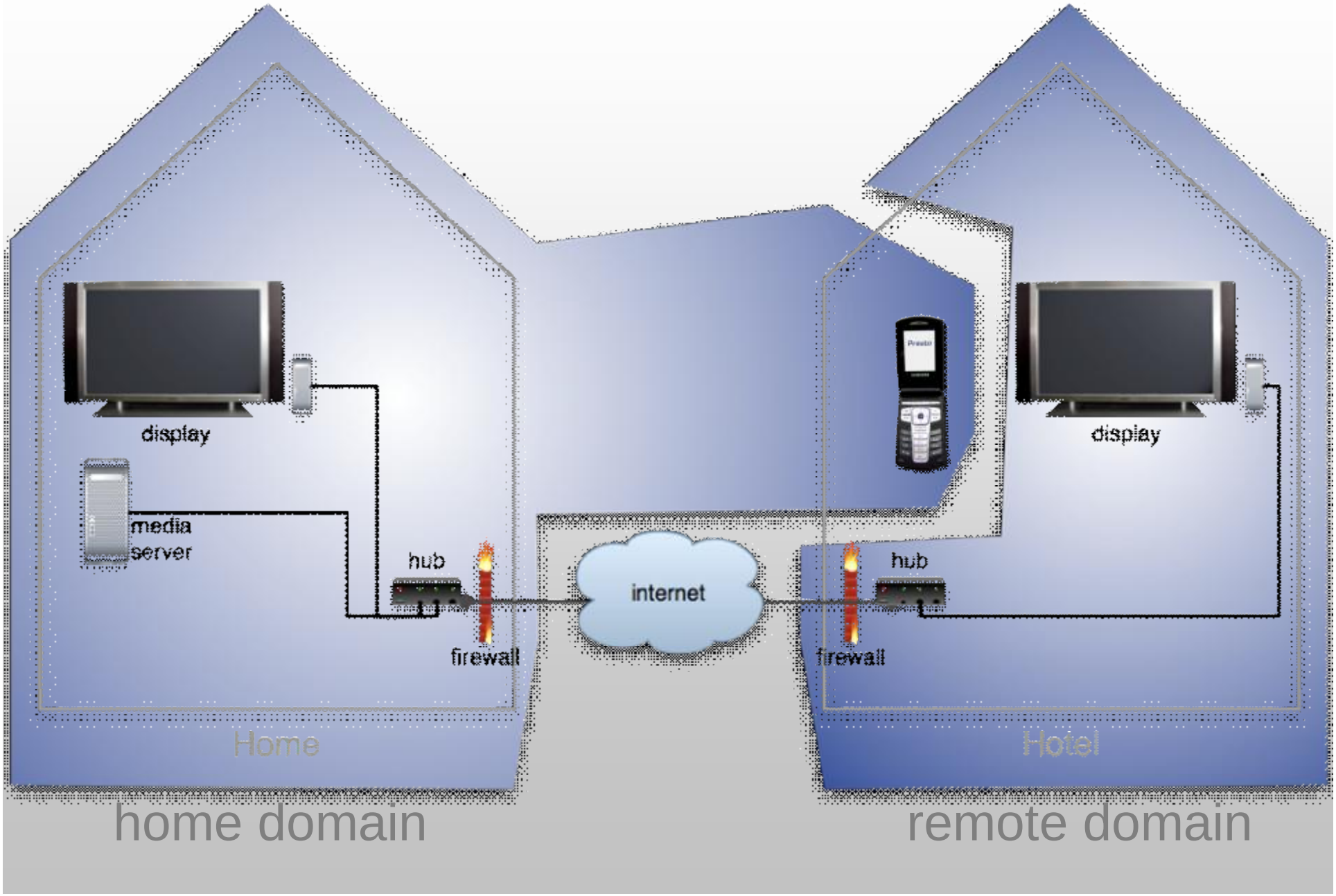
parc
Palo Alto Research Center

Our Solution

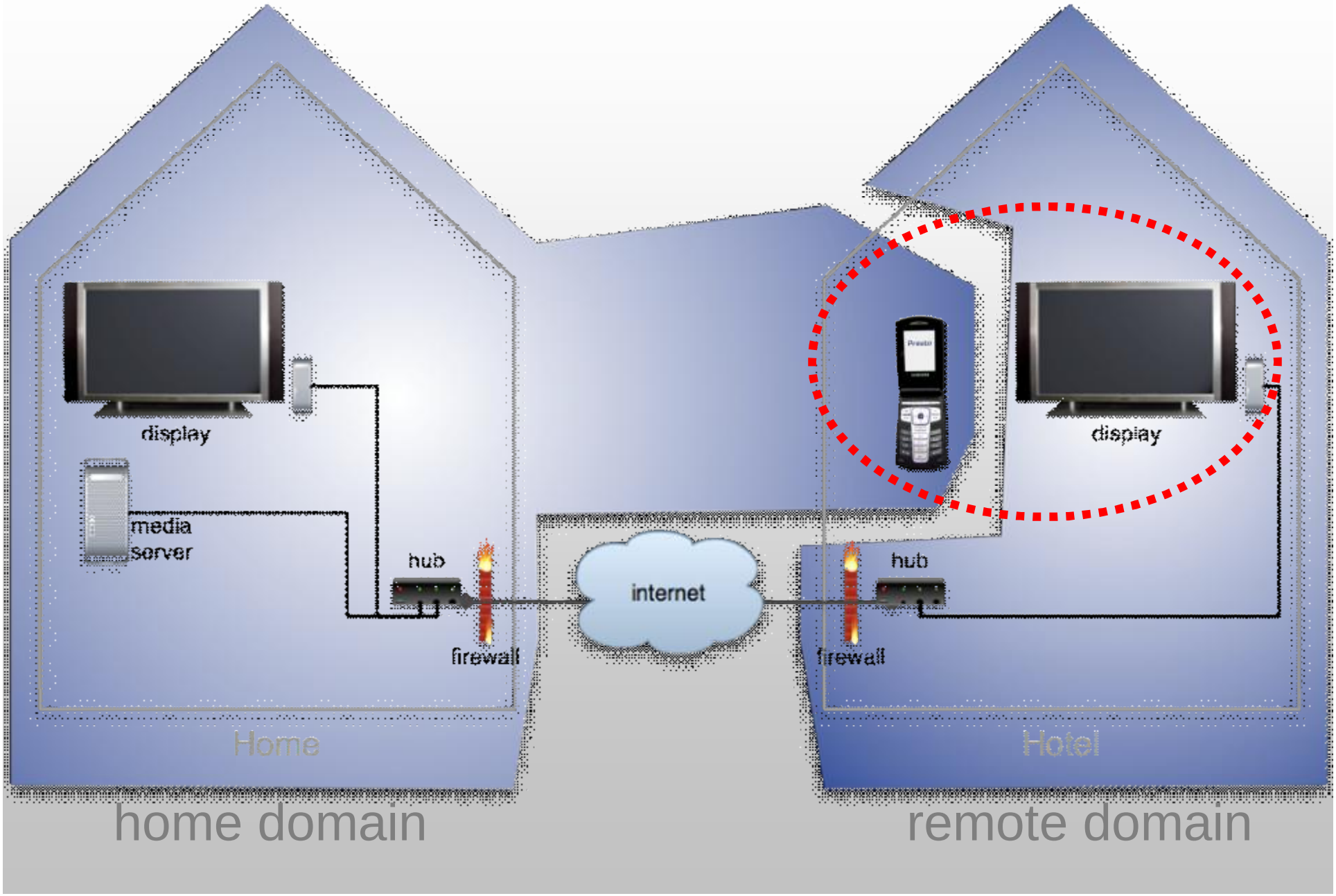


parc
Palo Alto Research Center

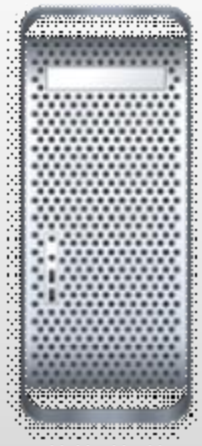
Security by Introduction



Security by Introduction



Instant Matchmaker as Security Mediator



trusted resource

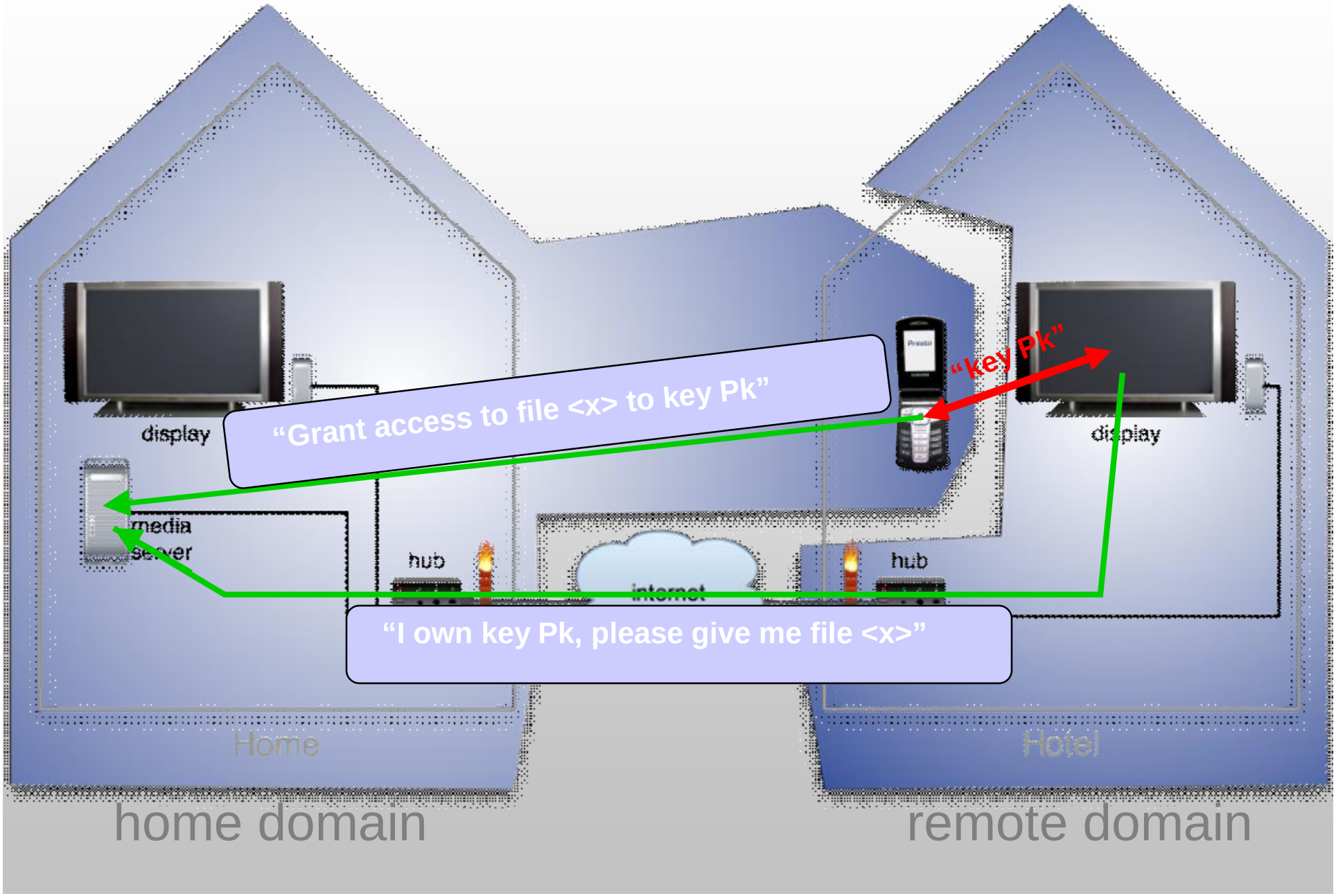
untrusted access device

long-term trust

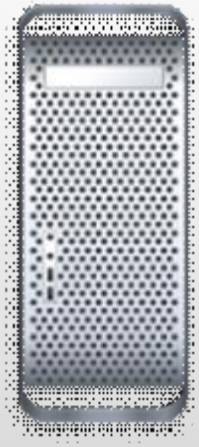
proximity

instant matchmaker

Security by Introduction



Demonstration System



trusted home resource media store

untrusted local TV device

local (“instant”) PKI

key exchange over infrared

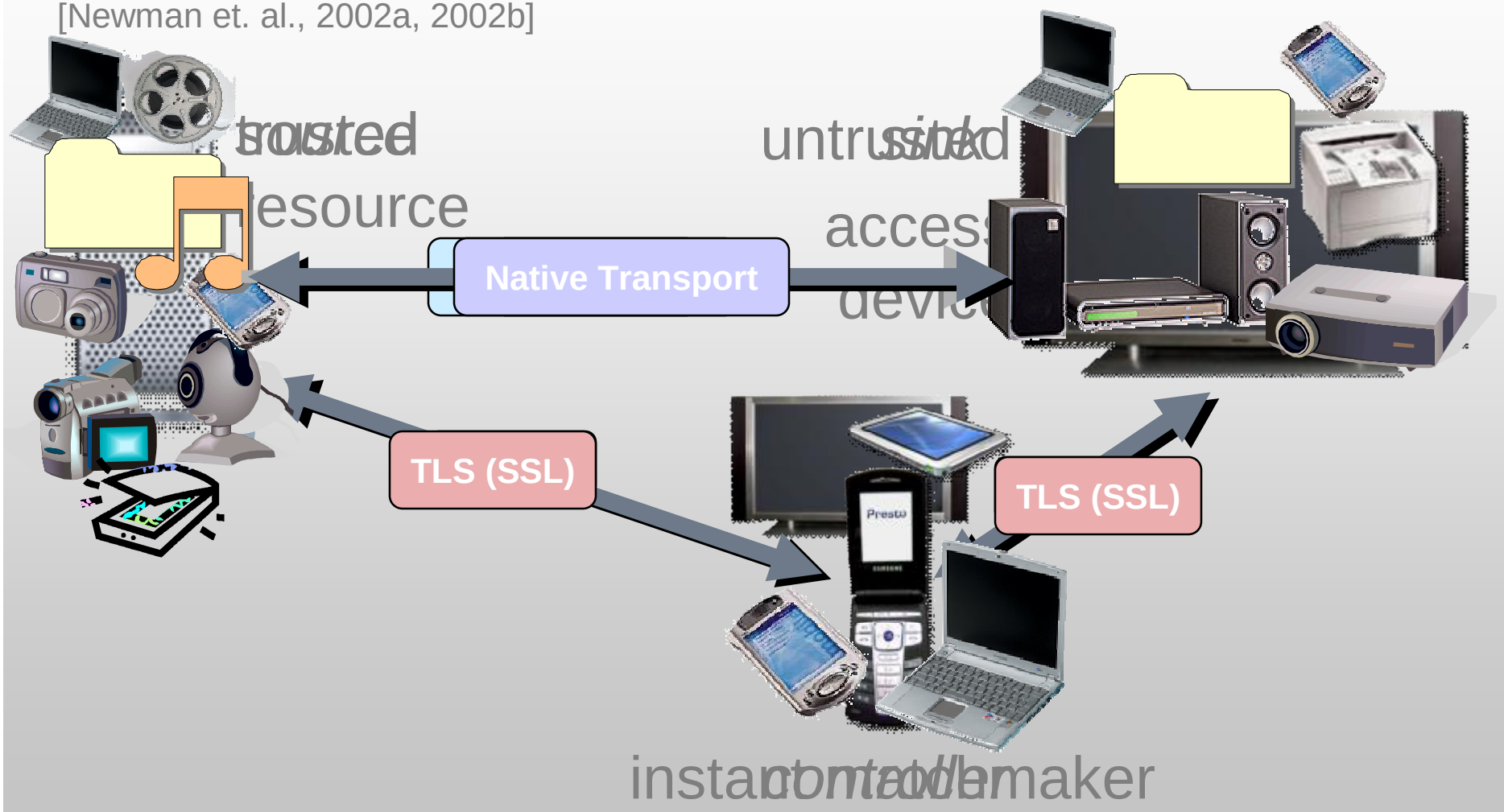
802.11a/b/g handheld phone

parc
Palo Alto Research Center

Obje (Speakeasy): Middleware for Radical Interoperability

[Edwards et. al., 2002]

[Newman et. al., 2002a, 2002b]

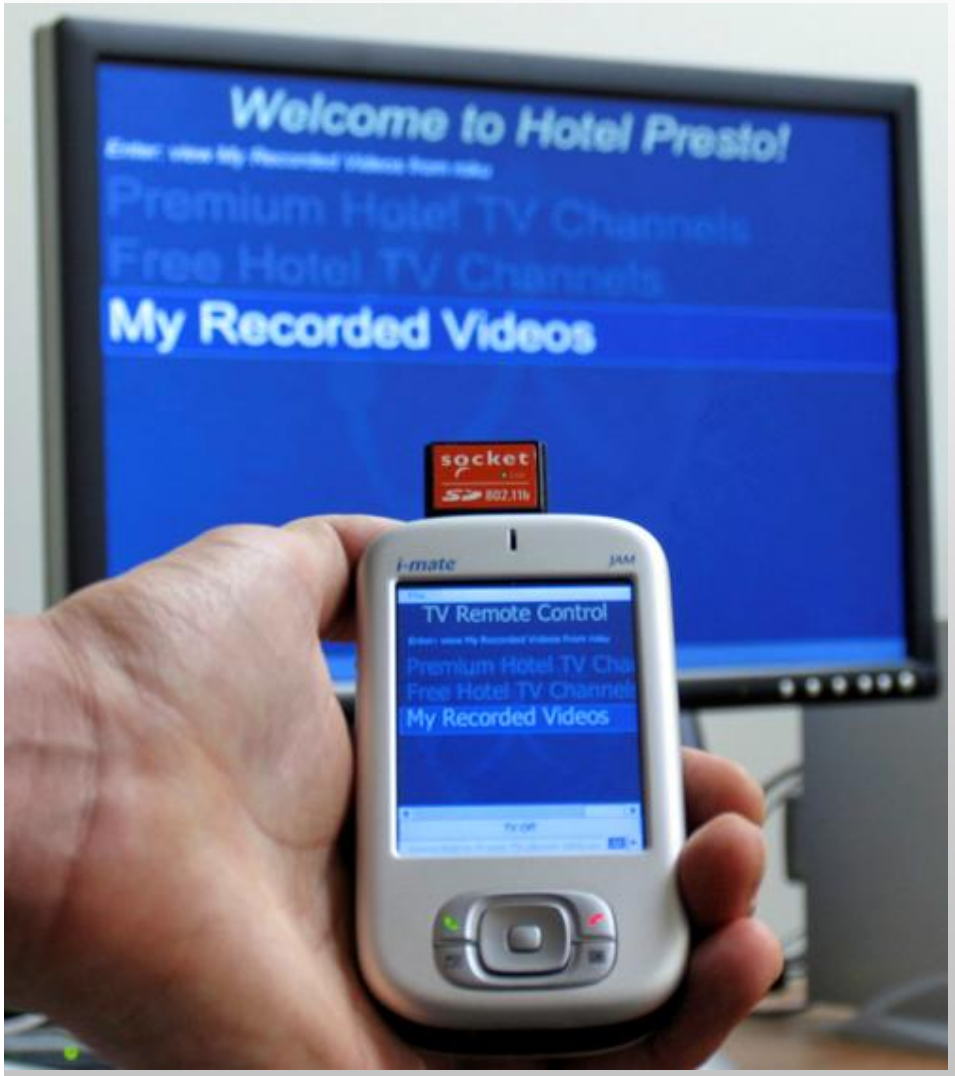


Establishing Trust

- embed public key information in infrared exchange
  - in this case, use phone as remote control to turn on TV
- establish secure connection and limited trust between phone and nearby display

parc
Palo Alto Research Center

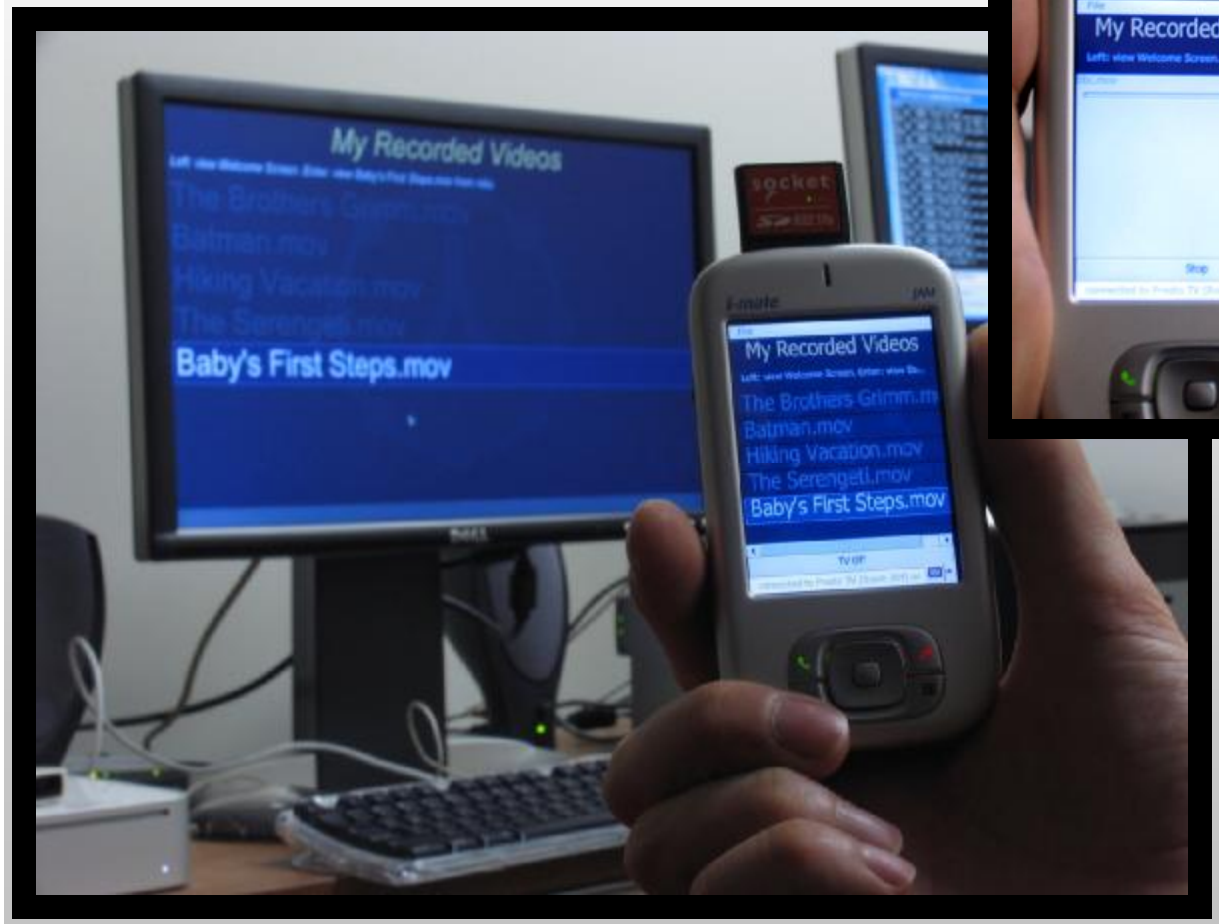
Determining Intent

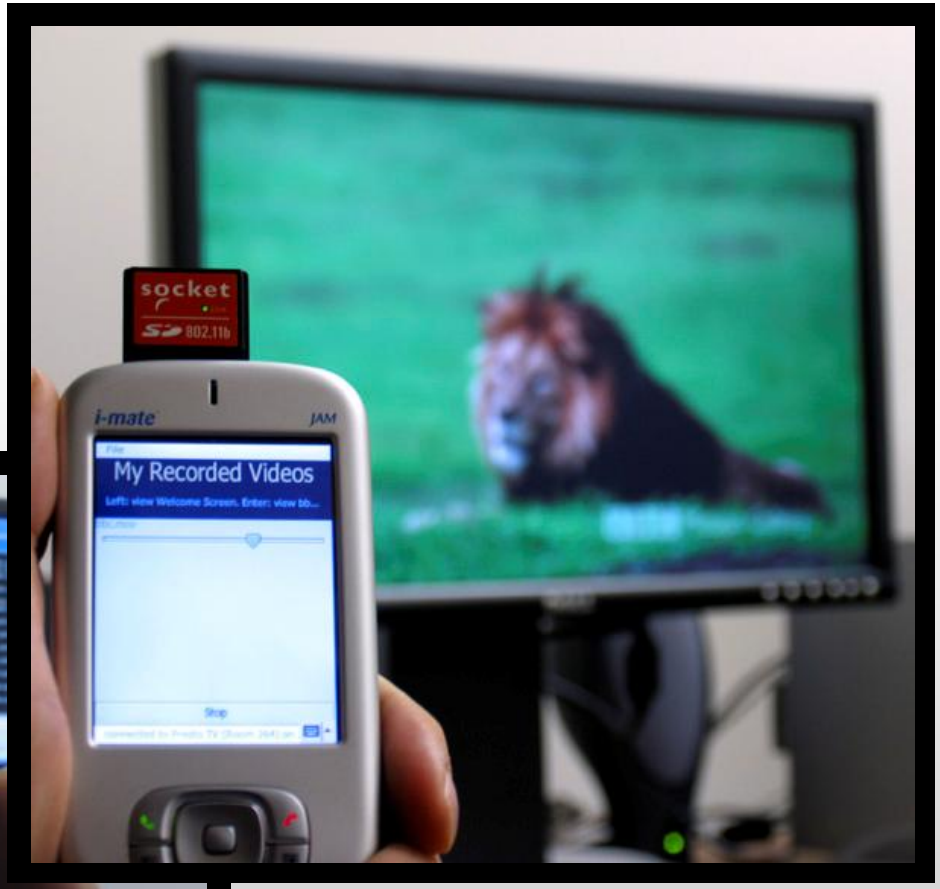


Trusted input path: gathering all input via the trusted matchmaker device gives reliable indicator of user intent.

parc
Palo Alto Research Center

Granting Access

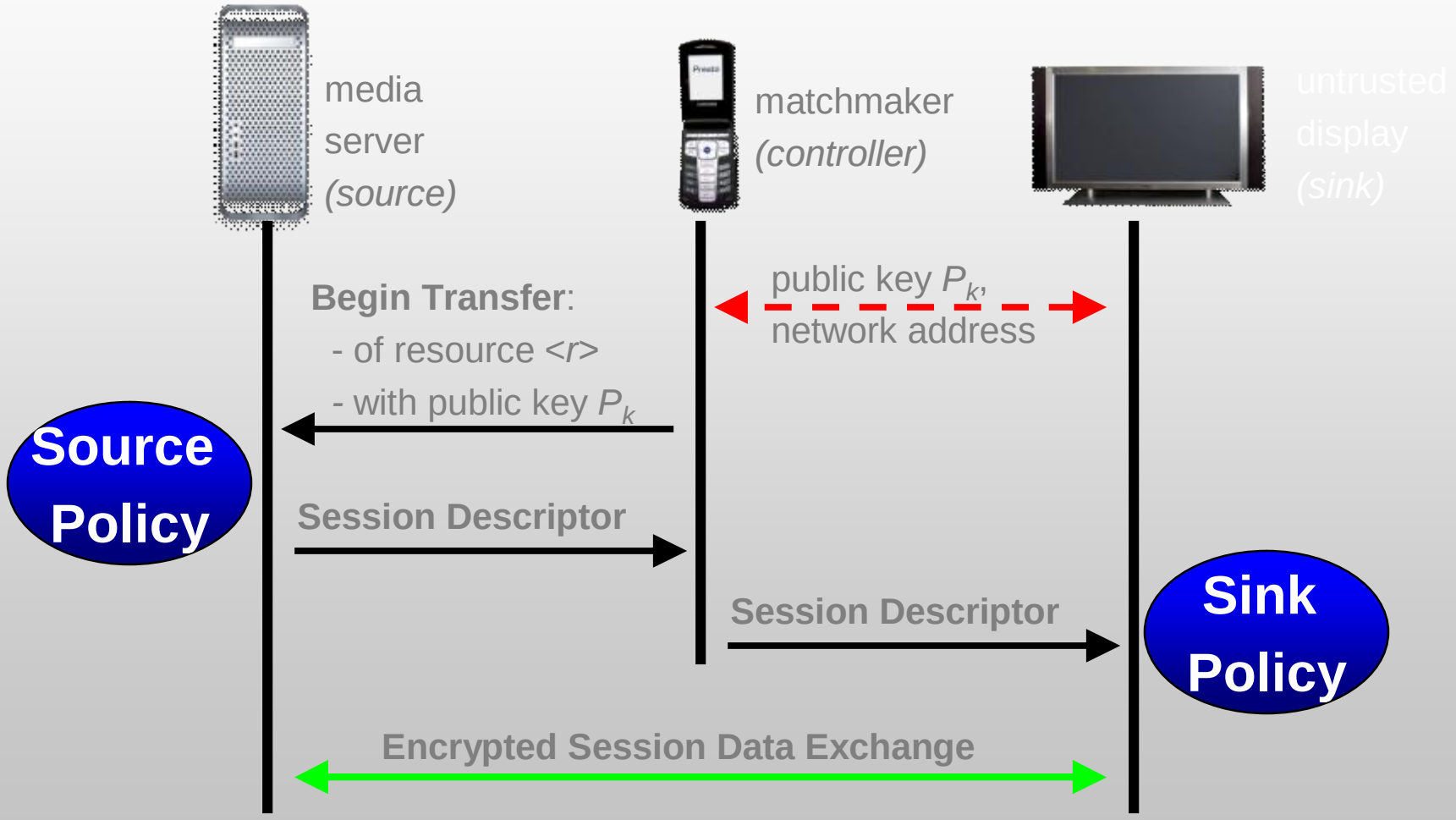




Selecting “Play”: designated device is allowed to play user-selected piece of content **one time**.

parc
Palo Alto Research Center

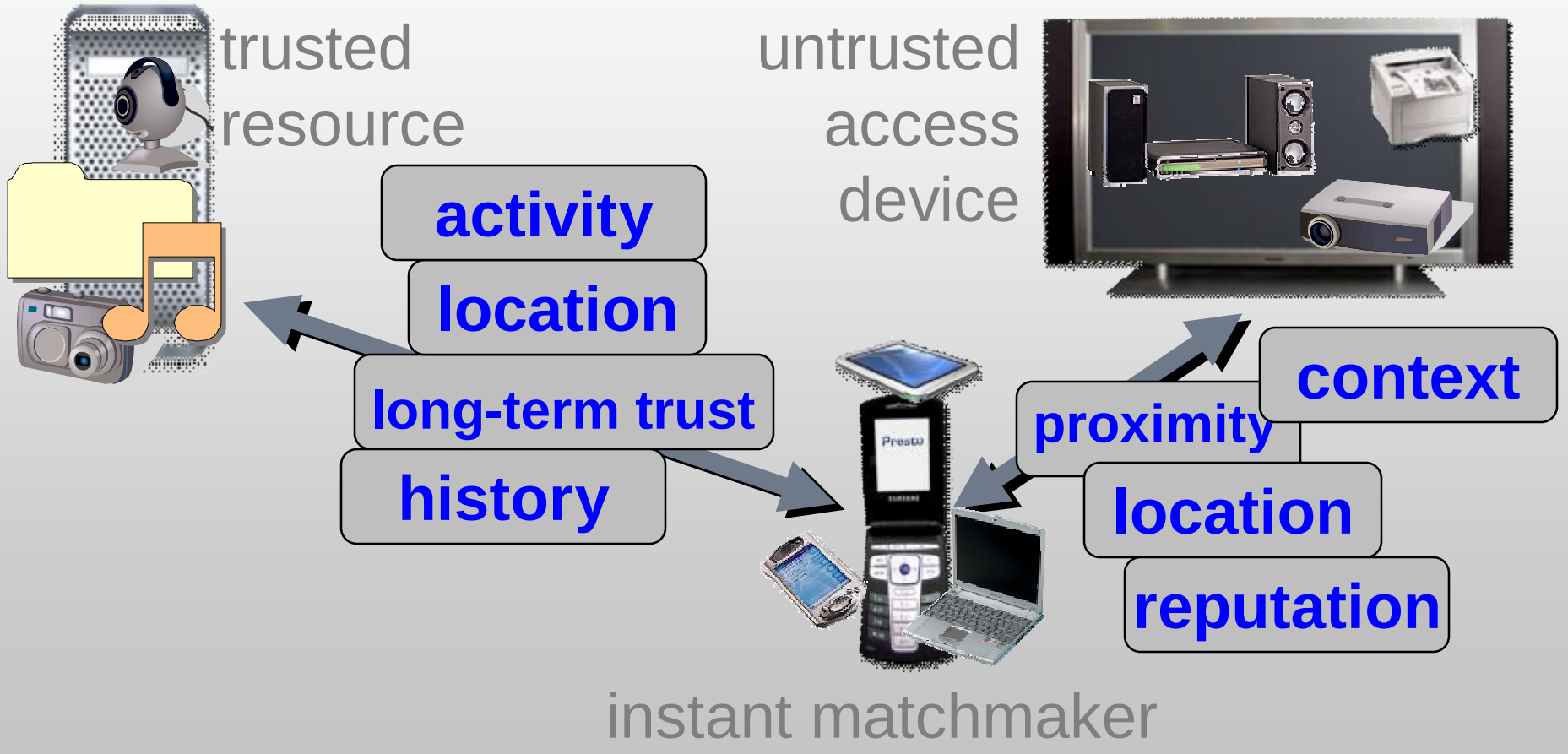
Obje Dataflow



Meta-Policies

Access Control Point	Home Device	Instant Matchmaker (TV Remote)	Hotel TV
Source Policy	Allow session with any sink, *as long as controller is from home domain*. *(instant matchmaker policy)*	Only allow a session where controller is IM itself and sink is IR-authenticated local device *(for mirroring GUI)*	Only allow sessions with TV in source role if TV is also the sink. *(allow playing local content)*
Sink Policy	Allow session with any source, *as long as controller is from home domain*. *(instant matchmaker policy)*	No sessions in sink role allowed.	Allow session with any source, as long as controller is IR-authenticated. *(allow control only by most recent device to turn on TV)*

Instant Matchmaker as Security Mediator



Conclusions

- demonstrated approach for securely integrating personal and local ubiquitous computing resources
- introduced use of personal device as “instant matchmaker” to:
  - securely introduce new untrusted devices to user’s personal resources
  - provide reliable indication of what resources user wishes to access
- flexible approach to specifying fine-grained access control policy on the fly

A Few General Principles

(Everybody's got to have a set…)

- Implicit Security
  - if an explicit user request requires security changes to effect, don't make them ask again
    - » no separate but (un)equal security interface
    - » doesn't mean security must always be invisible, just that it ought to be when it can
- "And Nothing Else" Security Policy
  - easy and intuitive metric for what a system ought to do
- Who are the users?
  - usability also impacts developers, designers, administrators…
- User decisions should be framed in terms of application tasks
  - that is all that they care/know about
  - pushes security back up to the level of the application
- Users don't need to know and love your abstractions
  - "share the pain" not a good design mindset

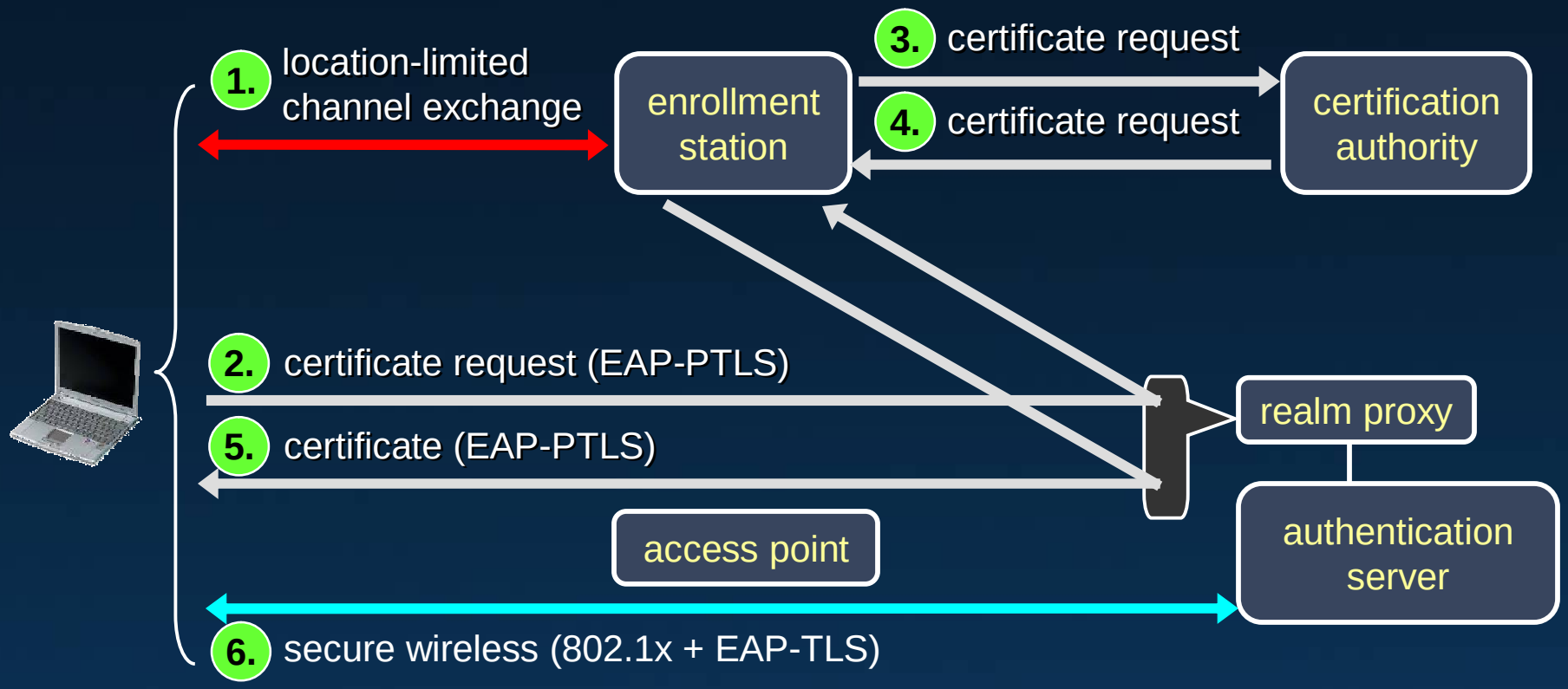
Extra Slides

parc
Palo Alto Research Center

Other NiaB Features

- Network status monitoring and individual device revocation
  - simple web interface
  - see who is currently associated with the network
  - one-button revocation of any enrolled device
- “Phone Home” VPN service
  - enrolled device configured to enable easy VPN access back to their home NiaB
  - IPsec connection authenticated using same certificate as used to access wireless LAN
  - can be separately enabled/disabled from WLAN access

System Architecture



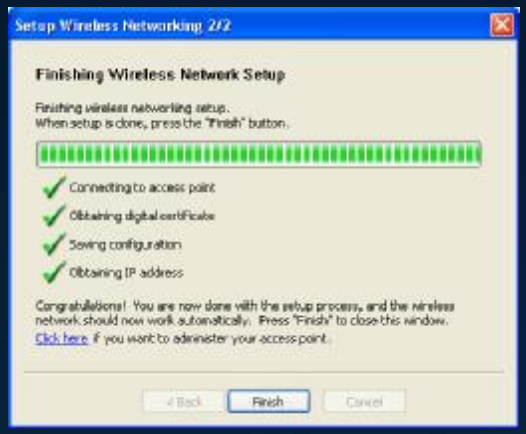
System Protocols

- EAP-PTLS: Client to Enrollment Station (EAP Traffic) (Trivial variant of EAP-TLS)
  - EAP-TLS performs standard TLS handshake then stops
  - We simply send messages inside the TLS tunnel
  - Endpoints use self-signed certificates
    - Hashes are compared against those received in location-limited channel
- CMP: Client to Enrollment Station (Certification Requests)
  - send PKCS#10 certification requests via Certificate Management Protocol (CMP, RFC 2510)
- Misc: Enrollment Station to Certification Authority
  - CA-specific protocols and transports

NiaB Implementation Details

- Lightweight implementation on commodity hardware, using mostly open source tools
- Implemented on standard Linux platform (RedHat 9.0, kernel 2.6)
  - Access Point: HostAP
  - Authentication Server: FreeRADIUS 1.0.0
  - Certification Authority
    - Small standalone CA using OpenSSL 0.9.7
    - Implemented as part of a FreeRADIUS plugin
  - Enrollment Station
    - Based on set of portable protocol libraries
    - Implemented as part of a FreeRADIUS plugin
  - VPN Server
    - Linux kernel 2.6 built-in IPsec
    - ipsec-tools' port of Racoon IKE daemon
  - Configuration Interface: mini_httpd
- Hardware platforms
  - OpenBrick modified to include front-access IR port
  - Linksys access point modified to support hardware IR

NiaB Client Implementation



- Primary Platform: Windows XP
  - Iterative design used to improve interface
  - Once client is configured, all further wireless network access done by native Windows XP 802.1x client – our software no longer used
- Basic implementations on other platforms (Linux, OS X)
  - all core libraries cross-platform

parc
Palo Alto Research Center

User Experience

- We conducted usability tests to compare traditional PKI + 802.1x enrollment against Enterprise NiaB.
  - Results showed ours took less time, and fewer configuration steps

System	Time (min)	Steps
Traditional PKI	140:00	38
Enterprise NiaB	1:39	4

  - Users rated NiaB *(1 = most positive, 5 = most negative)*
    - easier to use
    - a more satisfying experience (satisfaction)
    - more confident that they configured security correctly (confidence)

System	Ease of Use	Satisfaction	Confidence
Traditional PKI	5	4	4
Enterprise NiaB	1	1	1

parc
Palo Alto Research Center