Carnegie Mellon CyLab
4616 HENRY STREET
PITTSBURGH, PA 15213
PH: (412) 268-7195
FX: (412) 268-7196
www.cylab.cmu.edu

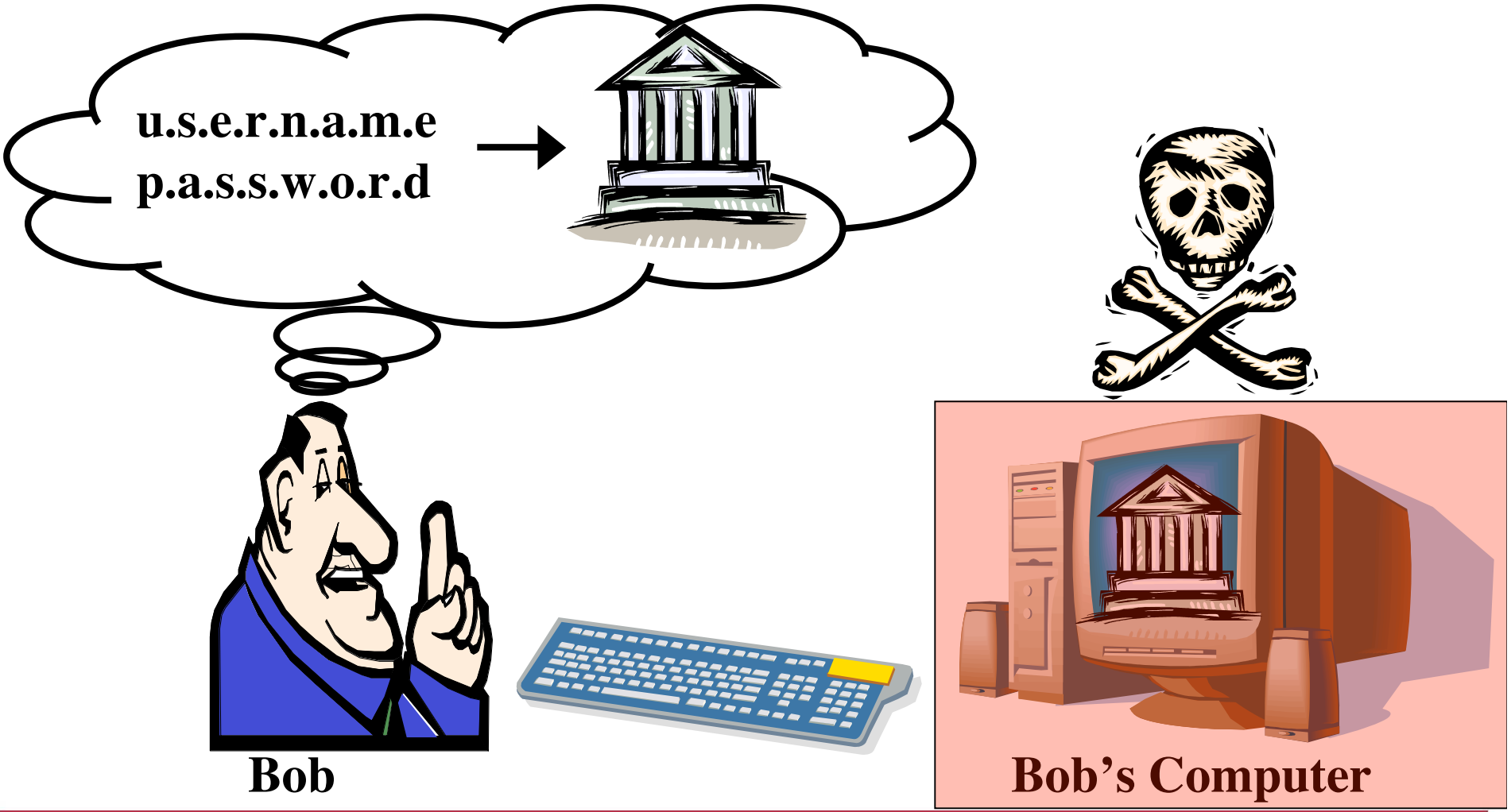
Bump in the Ether: A Framework for Securing Sensitive User Input

June 2, 2006

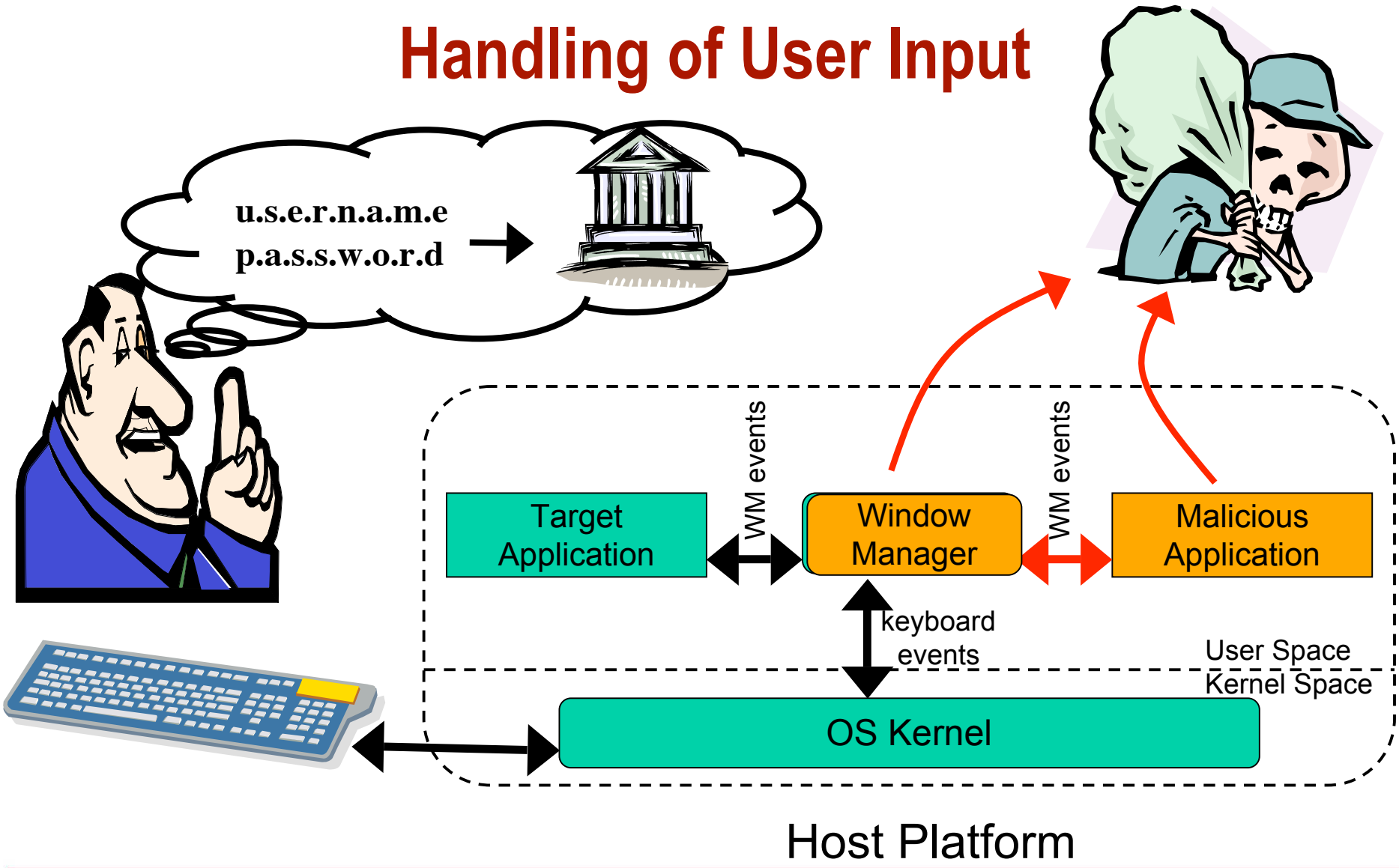
Jonathan McCune
Adrian Perrig, Mike Reiter

Carnegie Mellon University

Scenario



Handling of User Input



Problem Definition

- **Challenge: Preventing malicious parties from capturing user input**

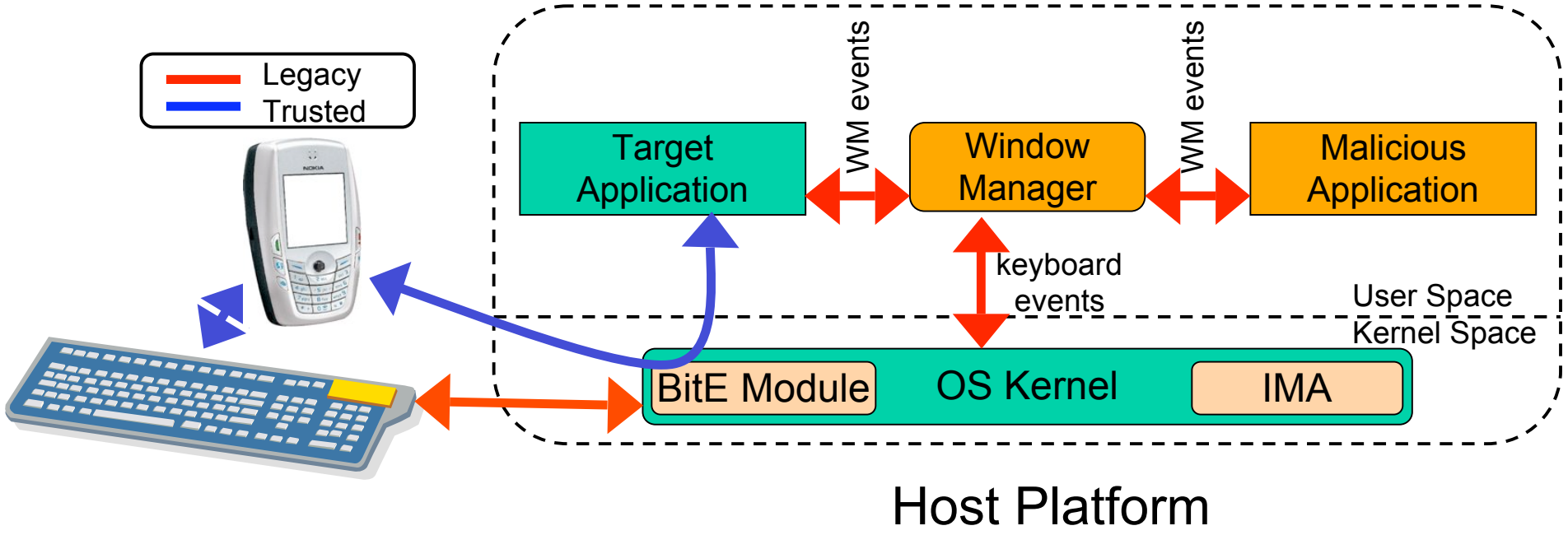
- **Threat model**
  - Malicious user-space applications
  - Compromised window manager (e.g., X, MS Explorer)
  - Passive monitoring and active injection on wired and wireless networks

- **Assumptions**
  - Host platform has a TPM
  - No run-time compromise of OS
  - No run-time compromise of *target application*

BitE System Architecture

- **Trusted mobile device, runs BitE Mobile Client software**
  - Evaluates software state of host using attestation
  - Provides trusted display and input out of reach of malware on host
  - Proxies user input between input device and host
- **Partially trusted host platform, runs BitE Kernel Module**
  - Generates attestations of software state using TPM
  - Maintains secrets in TPM-based sealed storage
- **BitE Kernel Module and Mobile Client participate in key setup**
  - Enables end-to-end encrypted, authenticated tunnel from mobile device to application
- **Bypasses traditional input path**
  - Window manager
  - Accessible to user-space malware

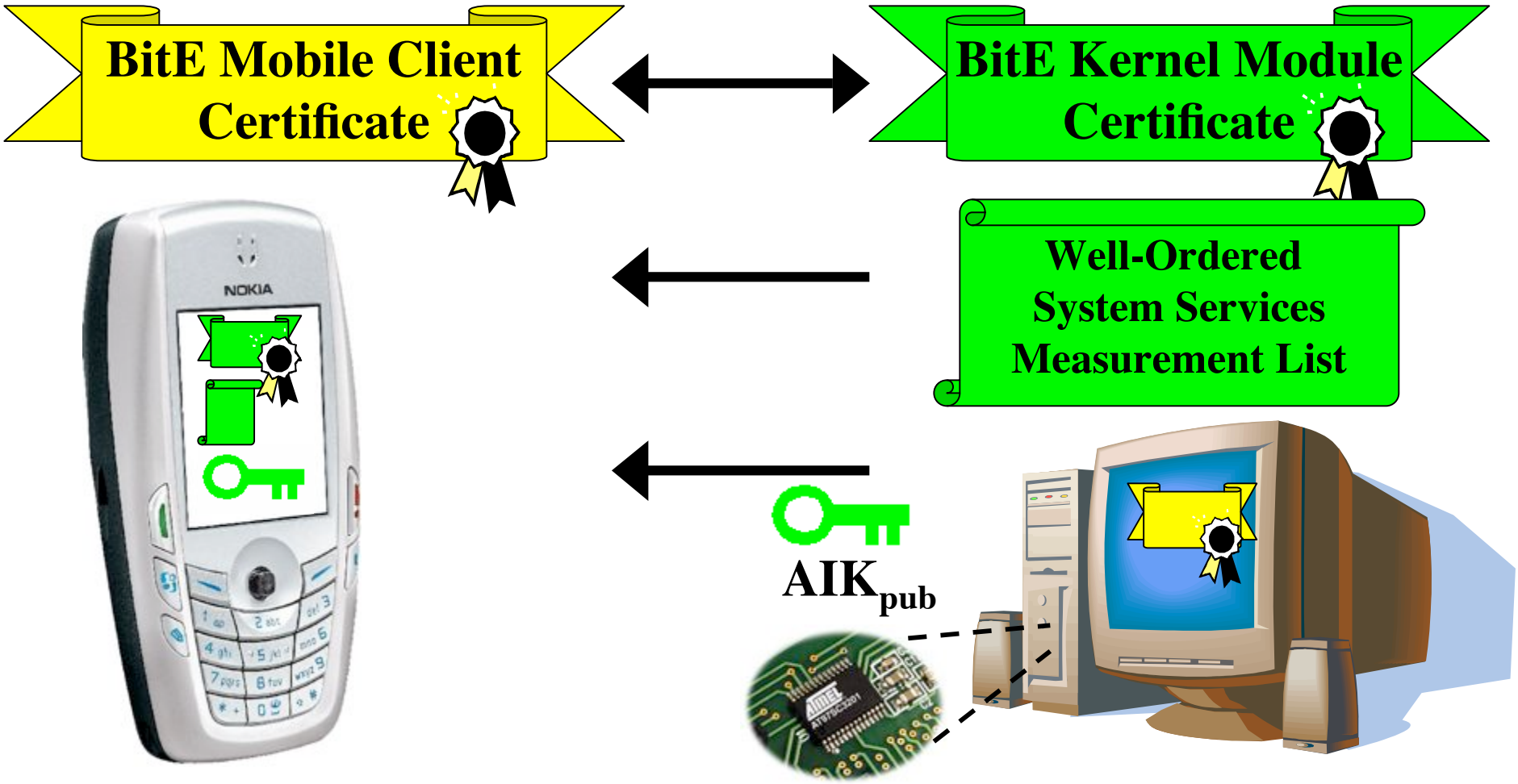
BitE System Architecture



Outline

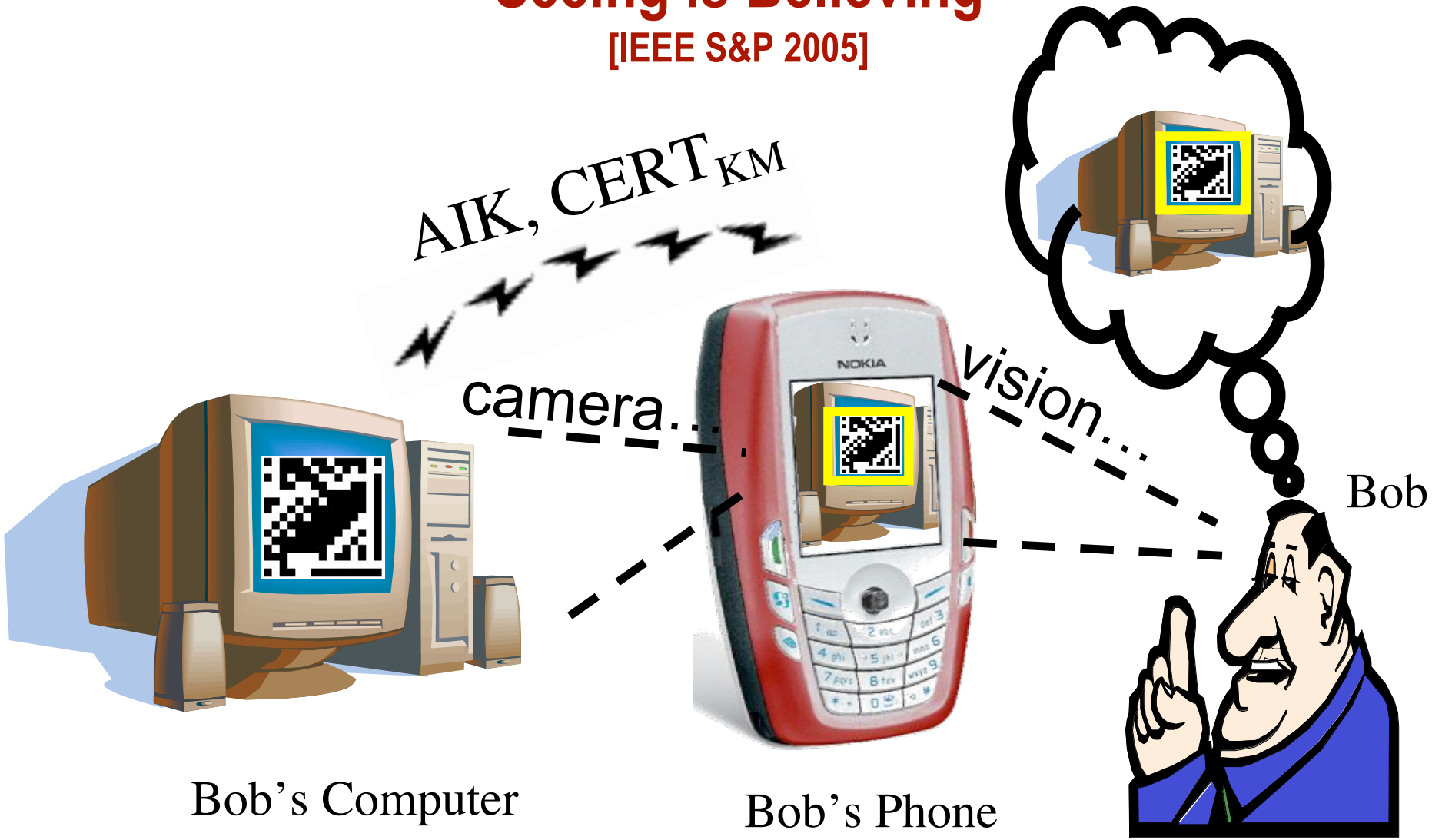
- **BitE setup**
  - Device association
    - Key exchange
    - Attestation mechanism
  - Application registration
- **BitE operation**
- **Security analysis**
- **BitE prototype**

BitE Setup: Device Association

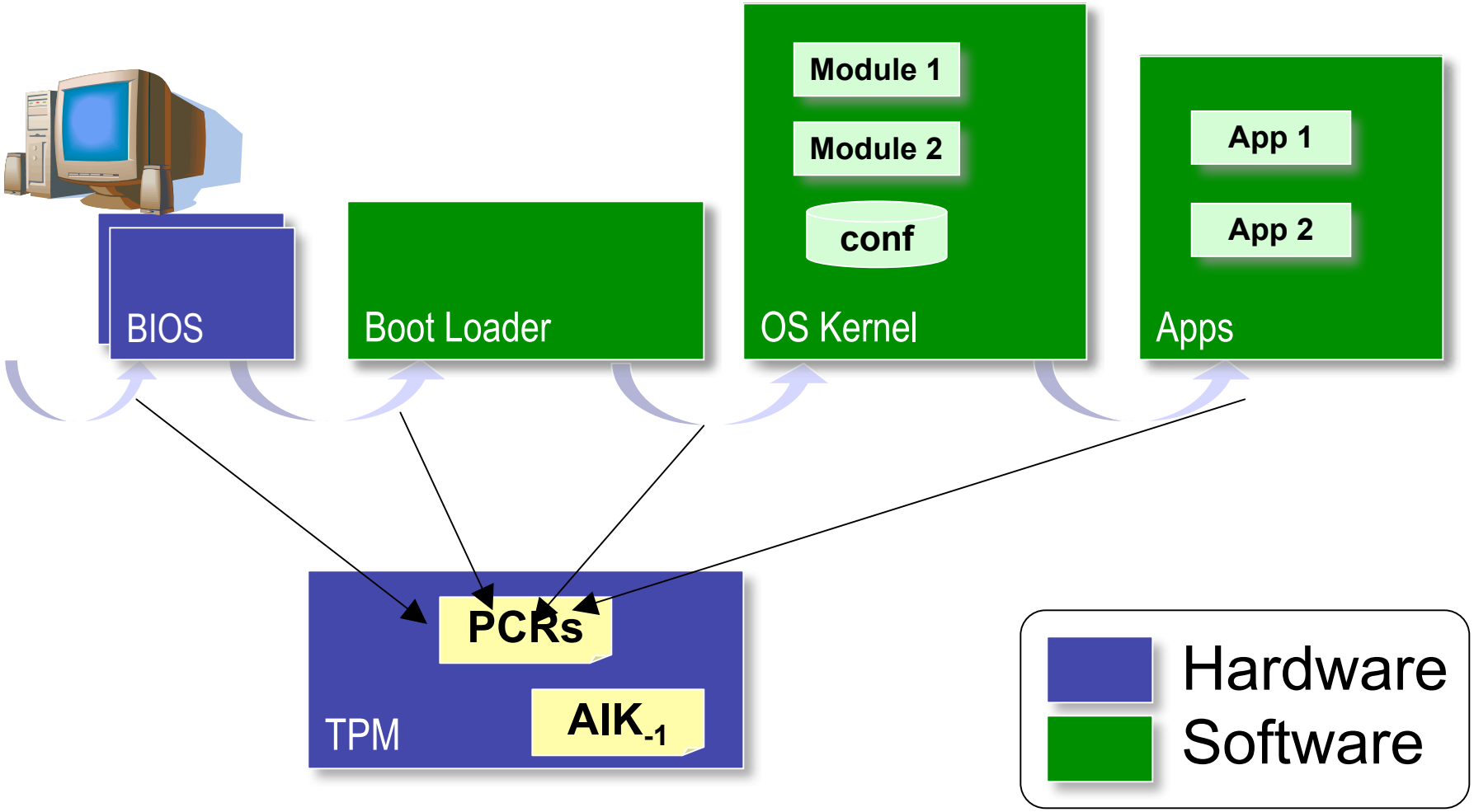


Seeing is Believing

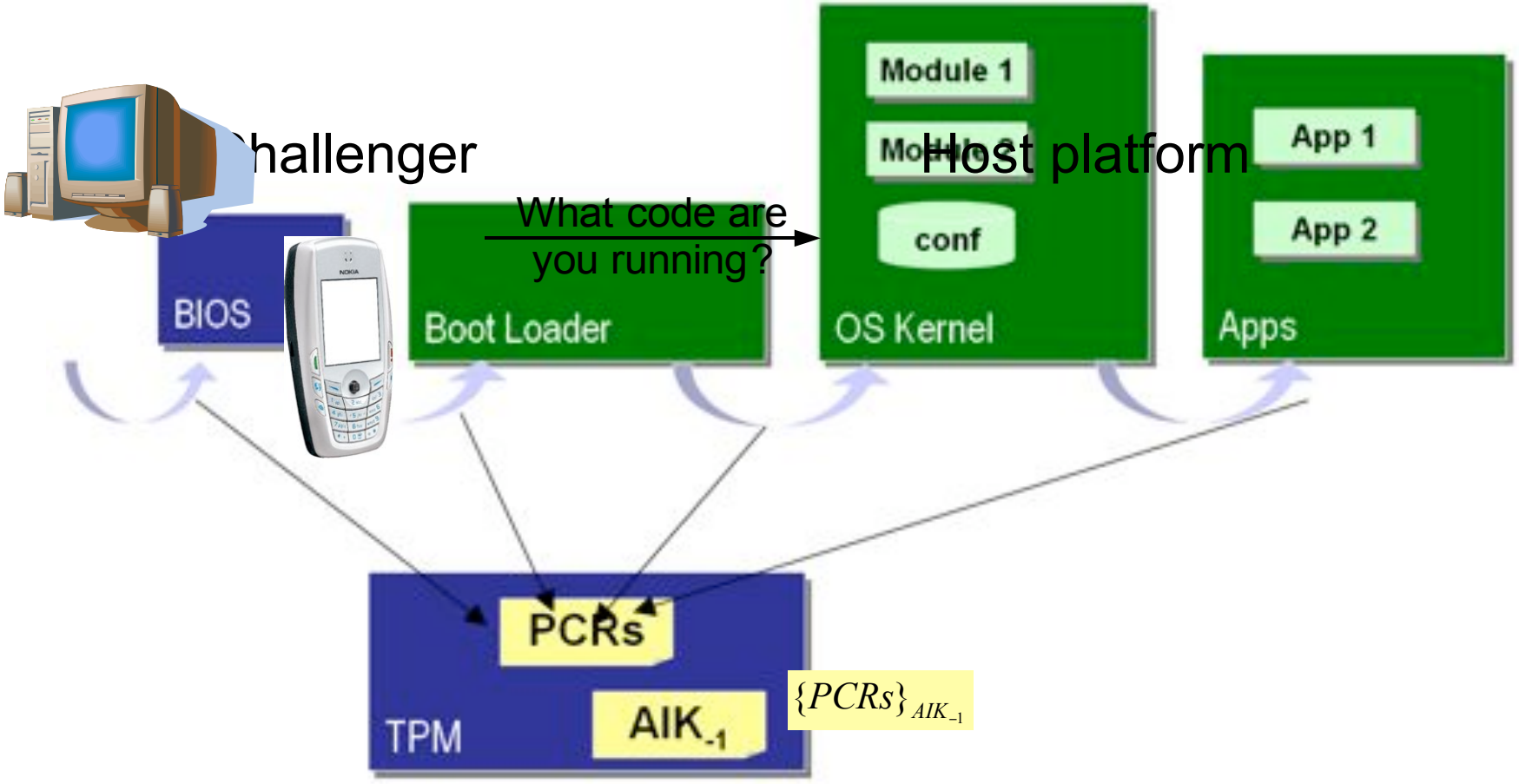
[IEEE S&P 2005]



TCG-Style Attestation



TCG-Style Attestation

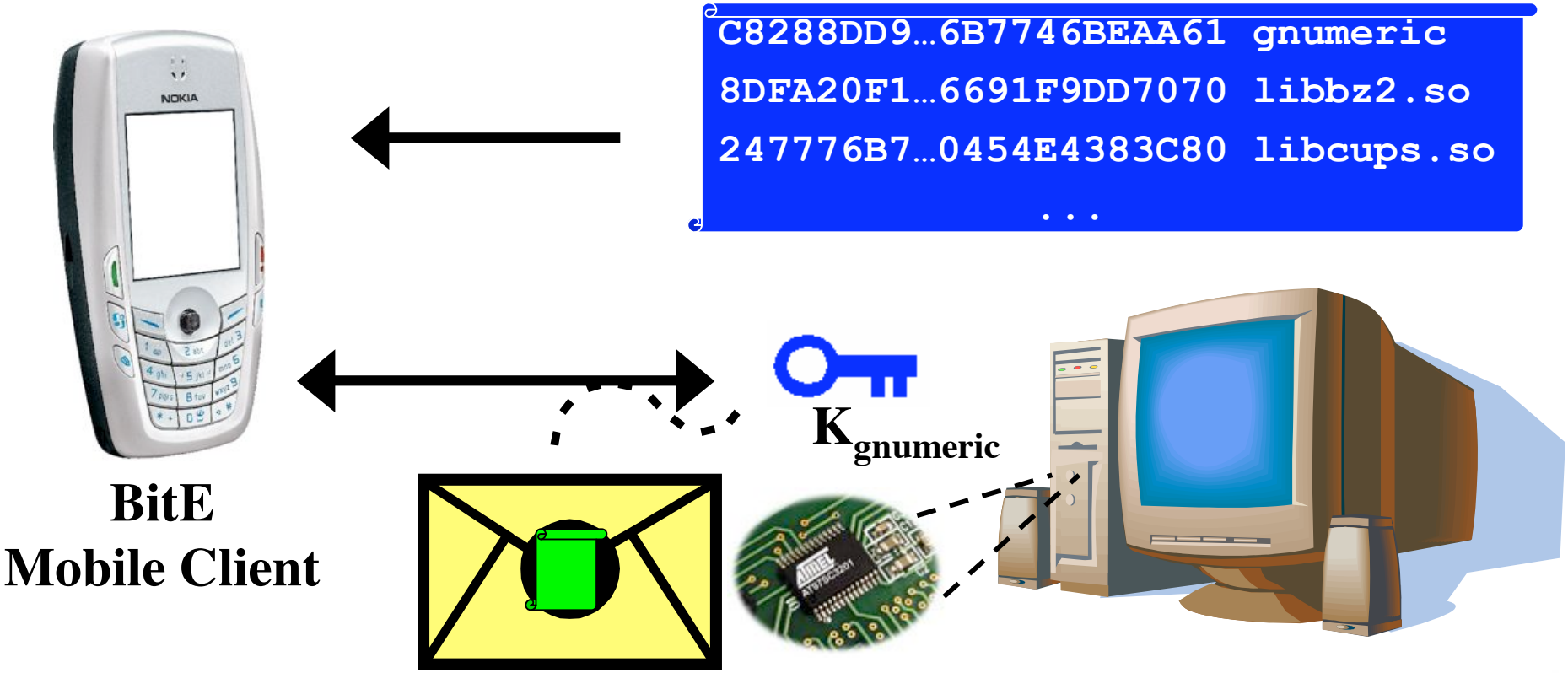


Why Attestation Alone is Insufficient

- **Maintaining a database of all possible measurements too hard**
  - Several attempts exist
    - `knowngoods.org`
    - `www.nsrl.nist.gov`
  - Not always current
- **Too much unknown software**
  - Some application downloads from the Internet
  - Pre-release quality software (alpha, beta, etc.)
  - User-compiled open-source software (e.g., Gentoo Linux)

BitE Setup: Application Registration

- Measurements of Gnumeric and its dependencies sent to BitE Mobile Client
- $K_{gnumeric}$ established using standard protocols
- $K_{gnumeric}$ kept in TPM-protected sealed storage



Outline

- BitE setup
  - Device association
    - Key exchange
    - Attestation mechanism
  - Application registration
- **BitE operation**
  - Application request
  - Verify attestation
  - User interaction
  - Establish session keys
  - Input sensitive data
- **Security analysis**
- **BitE prototype**

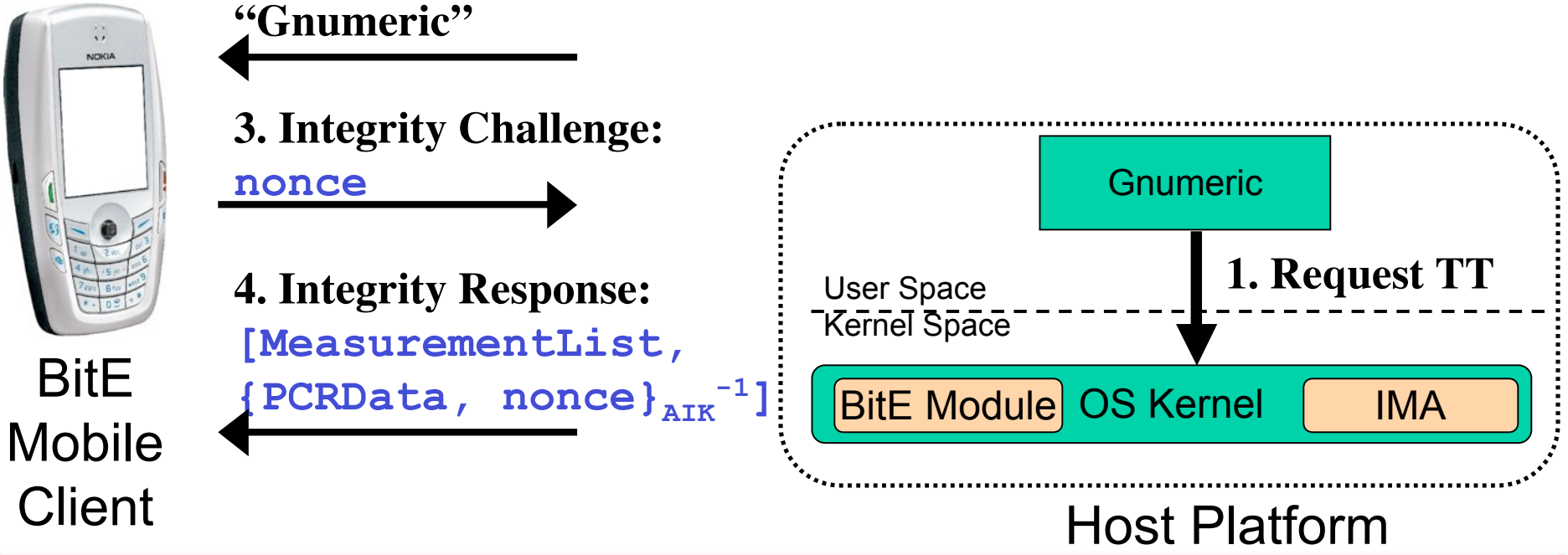
BitE Operation: Application Input Process

- **BitE-aware applications**
  - Request trusted tunnel for sensitive input
  - Release it when finished
- **Legacy applications**
  - All input can be sent through tunnel, or
  - The user can manually enable and disable the tunnel as desired
  - Achieved via a wrapper

- **Keystrokes encrypted with per-application keys by mobile device**
- **Keystrokes do not pass through window manager**
- **Protects secrecy and integrity of input**

BitE Operation: Application Request

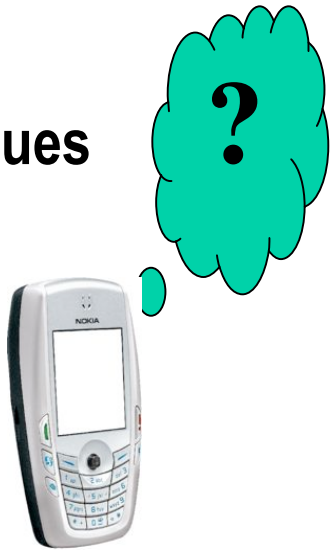
- Target application (e.g., Gnumeric) requests secure input



BitE Operation: Verify Attestation

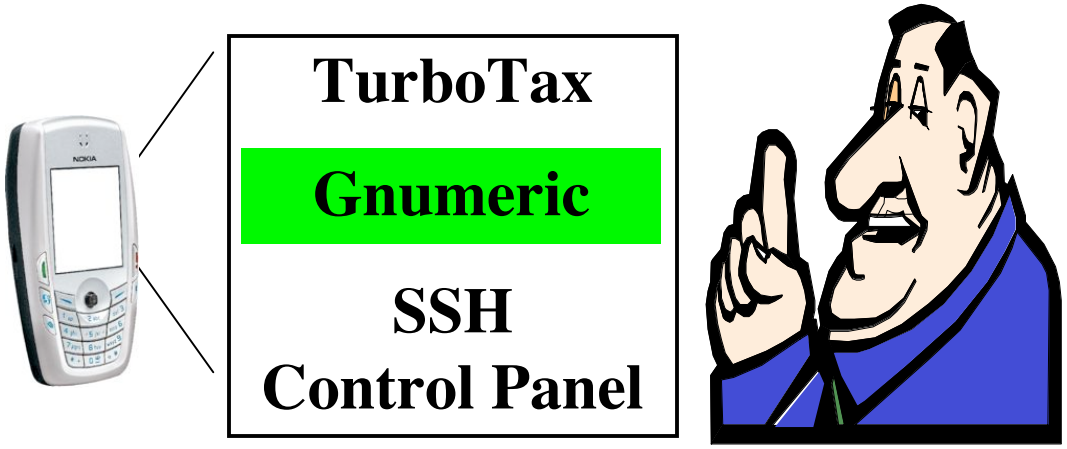
- State of well-ordered system services on host platform should be identical to state during device association
- Measurement for Gnumeric and its dependencies (e.g., `libbz2`) should be identical to measurement during application registration

- BitE Mobile Client checks attestation for expected values
  - Verify Integrity Response
  - Validate Measurement-List
  - For *j* in {well-ordered system services}
    - Find(*j*, Measurement-List)
  - Find(*Gnumeric*, Measurement-list)



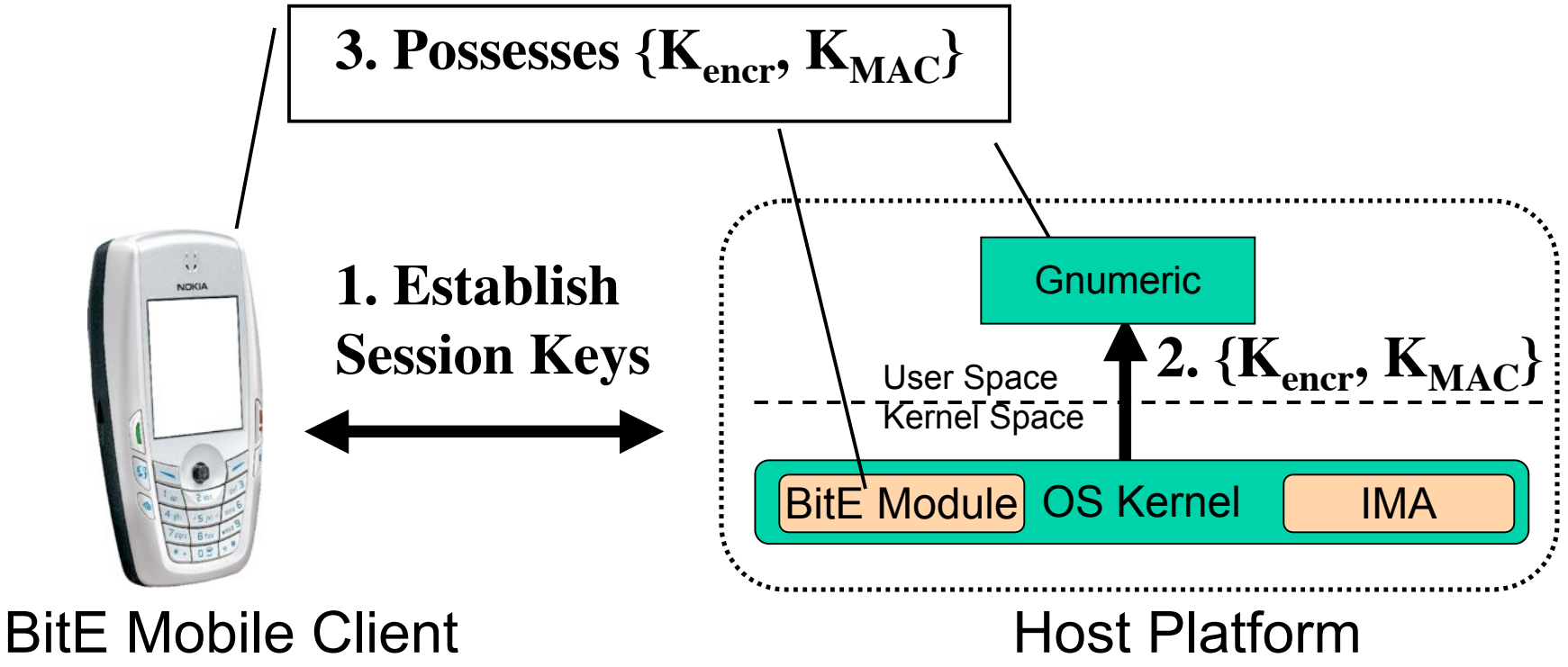
BitE Operation: User Interaction

- **User must select the application which requested the tunnel from a list displayed by the BitE Mobile Client**
  - Order of list is randomized to avoid user's forming bad habits
  - Items on list are other registered applications
  - Malicious application was never registered, so it is not on the list



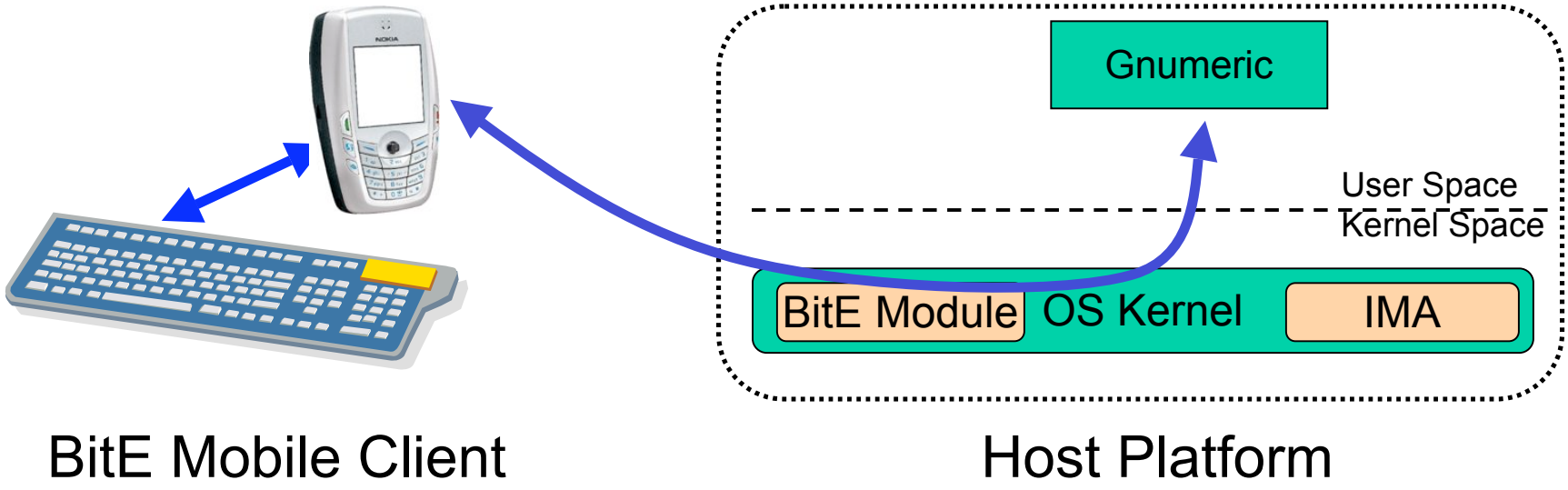
BitE Operation: Establish Session Keys

- Standard protocols used to derive $\{K_{encr}, K_{MAC}\}$ session keys from $K_{Gnumeric}$



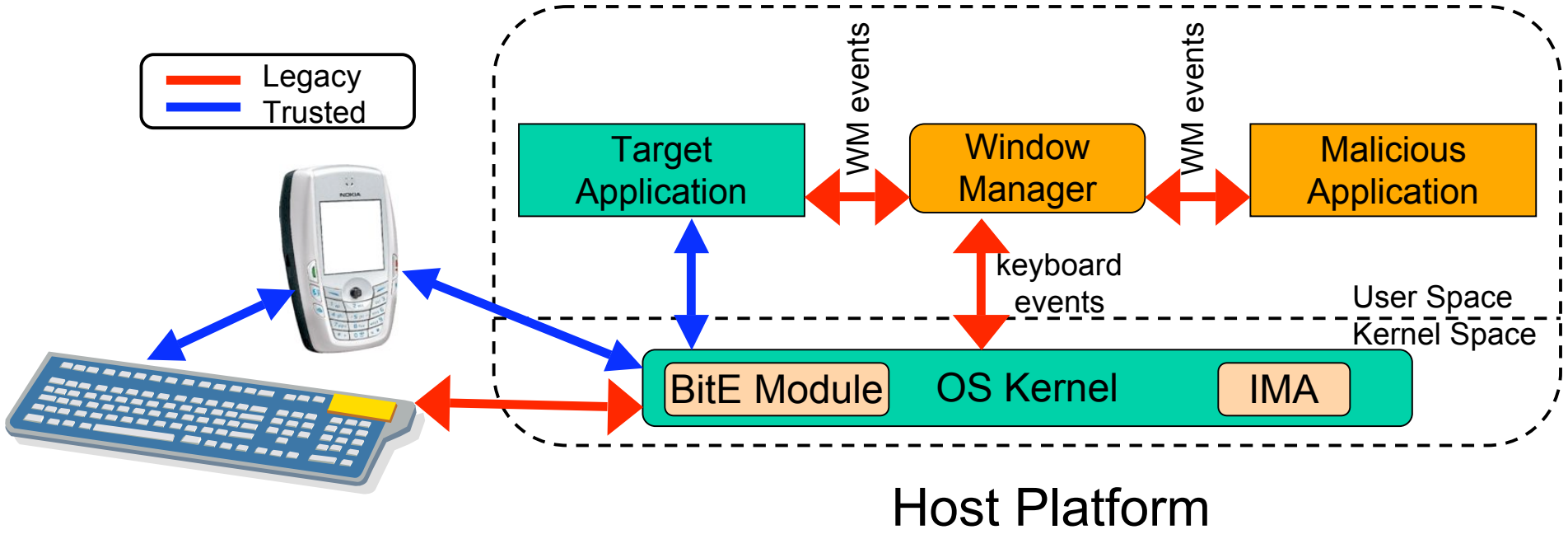
BitE Operation: Input Sensitive Data

- **Keystrokes proxied by BitE Mobile Client**
  - Encrypted and authenticated with $\{K_{encr}, K_{MAC}\}$
  - End-to-end trusted tunnel from mobile device to the application



BitE Input vs. Legacy Input

- Legacy path is no longer used; malicious application receives no data



Security Analysis

- **User-space malware prevented from accessing user input**
  - Input traverses trusted tunnel
- **Modified binaries prevented from accessing decryption keys**
  - Verification of attestation will fail, preventing tunnel setup
- **Wrappers increase security for legacy applications as well**
- **Defended attacks:**
  - Capturing keystrokes with X
  - User-space software keyloggers
  - Bluetooth eavesdropping or injection
  - Modification of registered applications on disk
  - Modification of OS kernel on disk

BitE Prototype Details

- **Mobile Device**
  - Nokia 6620 smartphone
  - J2ME MIDP 2.0 App
  - Bluetooth: phone – host
  - IR: keyboard – phone
- **Host Platform**
  - IBM T42p laptop
  - Linux 2.6
  - Trusted Platform Module (TPM)
  - Integrity Measurement Architecture (IMA) from IBM

Crypto Performance on Mobile Phones

- 1024-bit RSA keys, public exponent of 65537
- 325 and 401 IMA measurements for N70, 6620, respectively

Action	Nokia N70	Nokia 6620
	Mean (ms)	Mean (ms)
RSA PSS (sign)	1332	1757
RSA verify	40	54
SHA-1 aggregate	91	171
Data manipulation	906	2087

Selecting a Trusted Mobile Device

- **Device is trusted**
  - Its compromise gives attacker ability to capture keystrokes
  - Thus, choice of device should be made carefully
- **We used a mobile phone in our prototype**
  - Widely deployed
  - Single-user device, less accessible for attacker than host platform
- **Options exist for higher sensitivity use (e.g., military scenarios)**
  - Atmel AT97SC3203S security module for embedded systems
    - TCG v1.2 TPM, 2048 bit RSA sign in 500 ms
    - True random number generator, Non-volatile storage
  - Higher cost to add display, I/O capabilities
  - Not deployed

Related Work

- **Mobile devices**
  - Hand-helds as smart cards [Balfanz et al.]
  - Splitting trust [Ross et al., Sharp et al.]
- **Secure window managers**
  - Trusted X [Picciotto et al., Epstein et al.]
  - EROS Trusted Window System [Shapiro et al.]
  - Microsoft's NGSCB
- **Trusted computing primitives**
  - IBM's Integrity Measurement Architecture [Sailer et al.]
  - Trusted Computing Group (TCG) specifications

Read the paper for…

- Additional details on legacy applications
- How to handle concurrent requests for trusted input
- Extension to mutual attestation between host platform and mobile device
- Alternative system architectures
- Alternative user interface design

Conclusions

- Malware (spyware, keyloggers, Trojans) running at user level is unable to capture user input sent via BitE
- Operation of BitE is convenient and intuitive for users
- BitE is feasible today on commodity hardware
- BitE still offers some protections for legacy applications

Thank you

- jonmccune@cmu.edu
- Questions?