ArmoredSoftware: Trust in the cloud

Annual Demonstration

Dr. Perry Alexander, Dr. Andrew Gill, Dr. Prasad Kulkarni,
Adam Petz, Paul Kline, Justin Dawson, Jason Gevargizian,
Leon Searl, Edward Komp

Information and Telecommunication Technology Center
Electrical Engineering and Computer Science
The University of Kansas
palexand@ku.edu,andygill@ku.edu,prasadk@ku.edu

January 15, 2015

Outline

Trust in the Cloud

Provide new capabilities that establish and maintain trustworthy cloud-based application deployment

- Establish trust among cloud components
  - trust among cohorts of processes
  - trust among processes and environment
- Promote informed decision making
  - data confidentiality can be confirmed
  - execution and data integrity can be confirmed
- Autonomous run-time response and reconfiguration
  - responds to attack, failure, reconfiguration, and repair
  - response varies based on measurement

Delivery Platform

Open source, standards compliant

- Lightweight integration with existing cloud infrastructure
  - OpenStack cloud infrastructure
  - Xen+XSM VM infrastructure
  - Fedora, HotSpot JVM, GHC
- Trusted Computing Group standards compliant
  - Trusted Platform Module 1.2
  - TCG vTPM (in principle)
  - Trusted OS infrastructure
- Standard communication mechanisms
  - JSON structures for all exchanged data
  - *vchan* for on-platform communication
  - TCP/IP for off-platform communication

New Technologies

- Trustworthy protocol execution
  - executable protocol representation
  - protocol execution generates evidence of trustworthiness
  - highly focused protocols
  - strand space formal semantics
- Application specific measurement
  - managed and traditional execution environments
  - compile-time assistance for measurer synthesis
  - specialized measurement bundled with applications
- Attestation driven cloud application and data management
  - health monitoring
  - problem mitigation
  - application migration
  - access control

Research & Development Plan

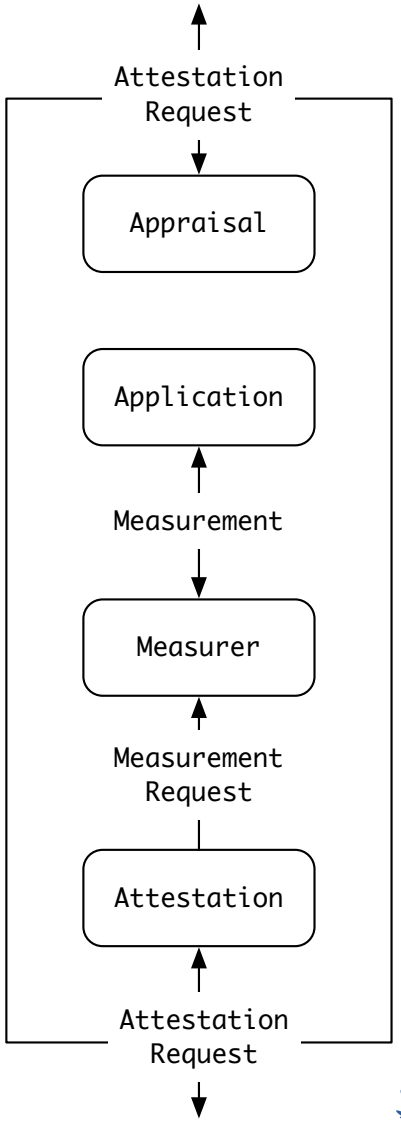
- Development and integrate measurement capabilities
  - hosted languages (Java)
  - traditional compiled languages (C, C++)
  - integrate with environment measurers (Xen,OpenStack,OS)
- Develop attestation capabilities
  - flexible, user configurable protocol representation
  - measured protocol execution
  - protocol execution appraisal
- Develop infrastructure trust argument
  - develop lightweight vTPM infrastructure supporting mobility
  - launch from known roots of trust
  - maintain trust evidence at run time
  - maintain trust over migration

Research & Development Plan

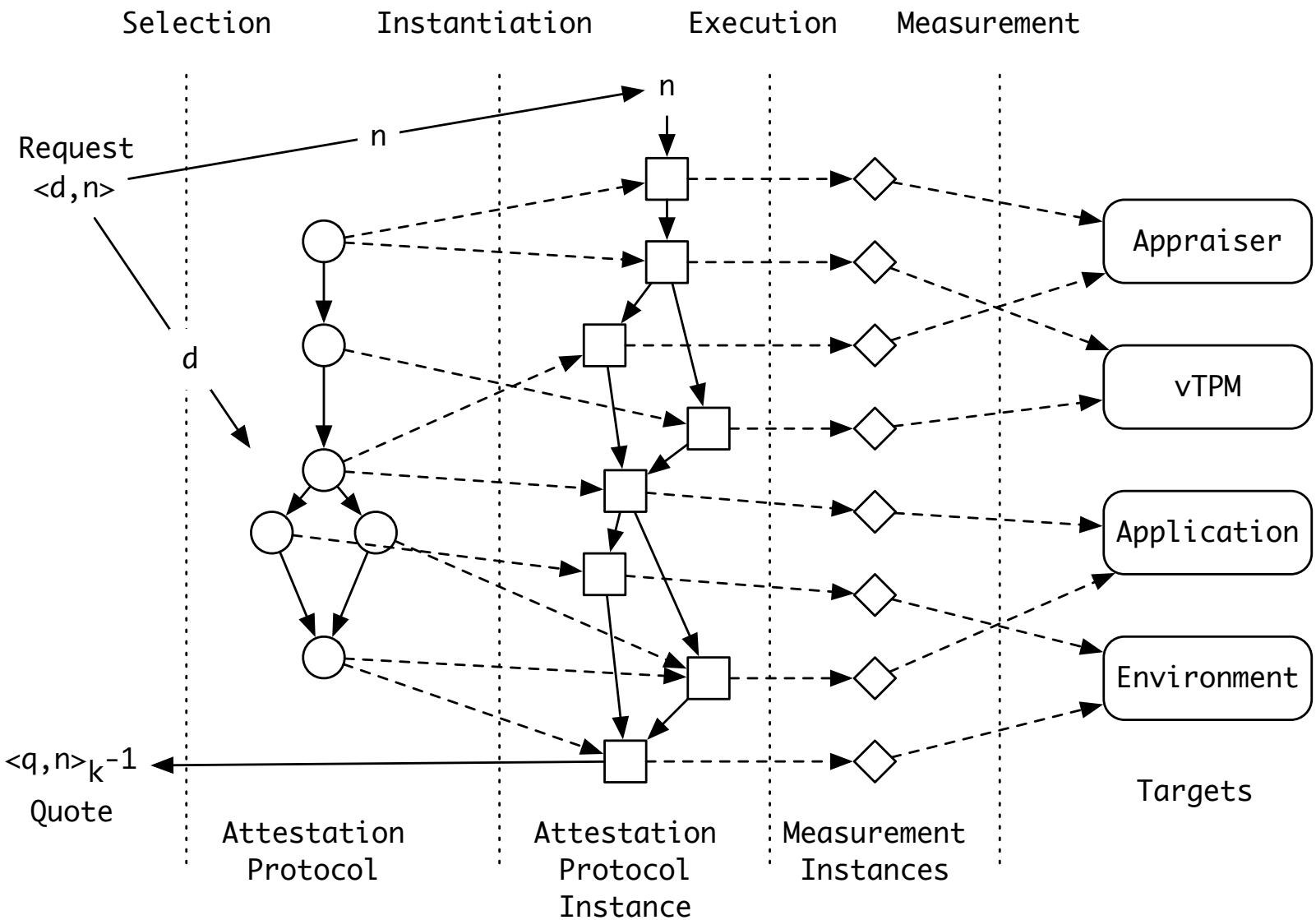
- Automated synthesis and verification
  - measurer synthesis at application compile time
  - automated evidence appraisal from protocols
  - formal trust argument
- Demonstrations
  - initial simple infrastructure demonstrations
  - cloud-based “big data” environment demonstration
  - federated trust demonstration
  - *demonstrations as discovered/directed*
- Scale up and roll out
  - integration with Xen, OpenStack, Linux
  - installation management and packaging
  - effective web presence

Armored Application Architecture

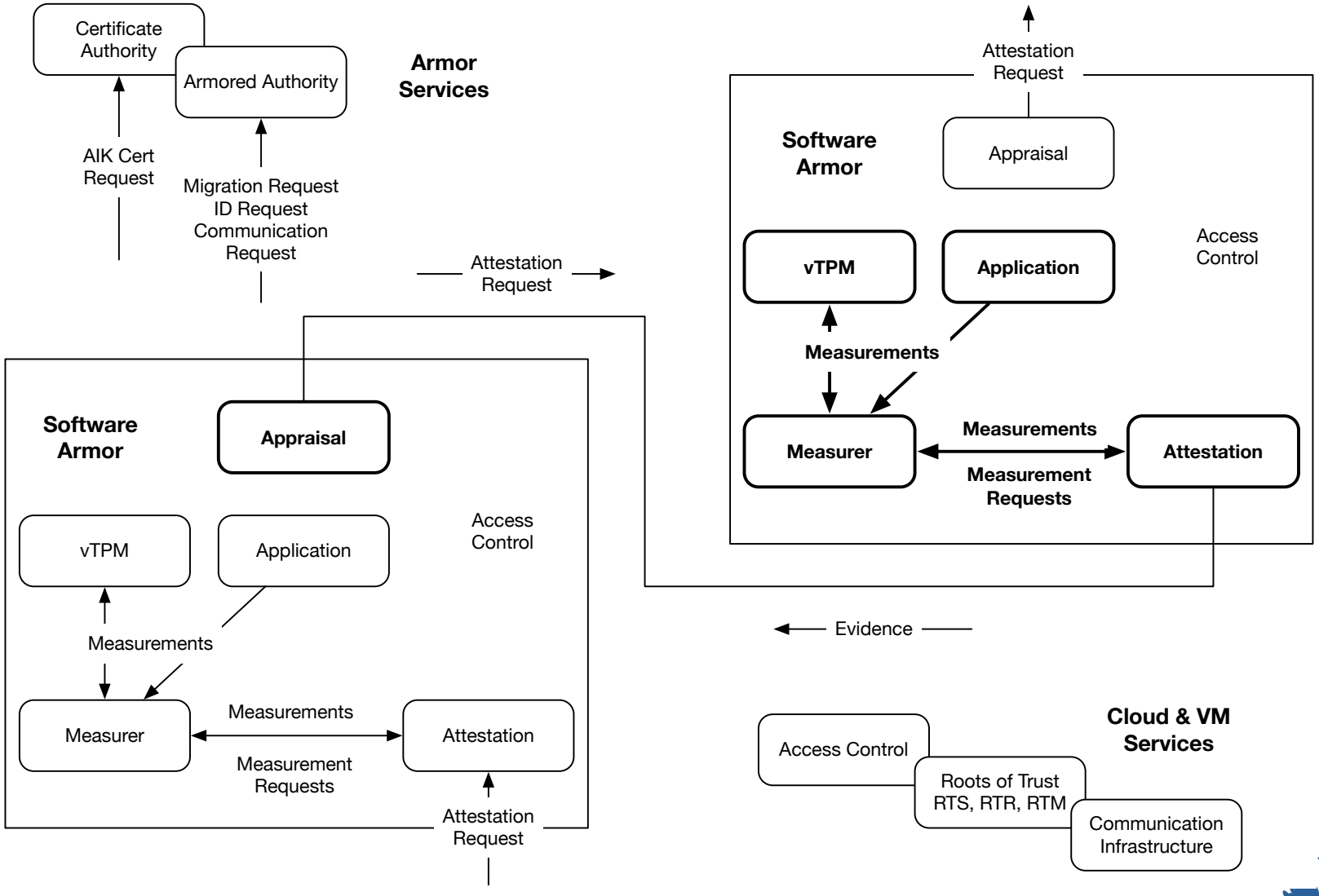
- Focus is user-space applications
- Assesses the cloud infrastructure and environment
- Attests to the state of its application
- High-assurance, lightweight infrastructure
- Influenced by the *Trusted Research Platform* and *Principles of Remote Attestation*



Measurement and Attestation



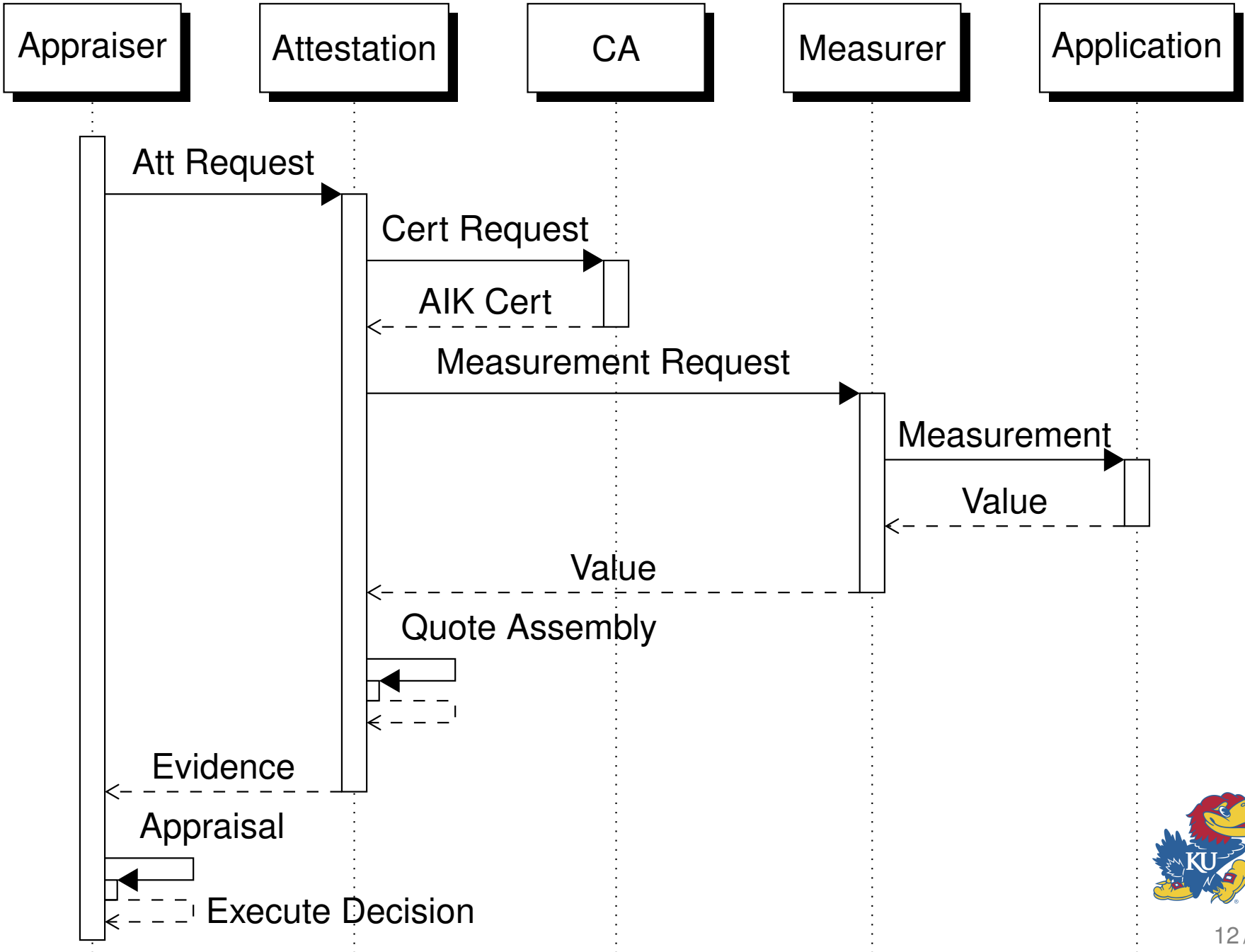
System-Level Architecture



What We Are Demonstrating

- Execution of a CA-based Attestation Protocol
  - Attestation request
  - Protocol execution
  - Evidence appraisal
- Major architectural subsystems
  - Appraiser
  - Attestation Manager
  - Measurer
  - Instrumented JVM
  - vTPM and Certificate Authority
- Anomaly Detection
  - Bad signatures and PCRs
  - Bad CA certificates
  - Bad quotes and AIKs
  - Bad measurements

Abstract CA-Based Attestation Protocol



Message List Representation

$App \rightarrow Att : d, N_{\text{App}}, PCR_{\text{m}}$ on C_{AppAtt}

$Att \rightarrow TPM : make\_and\_load\_identity$ on C_{AttTPM}

$TPM \rightarrow Att : AIK^{+}, AIK_{h}$ on C_{TPMAtt}

$Att \rightarrow CA : Att, AIK^{+}$ on C_{AttCA}

$CA \rightarrow Att : \{K, |AIK|\}_{EK^{+}}, \{[AIK^{+}]_{CA^{-}}\}_{K^{+}}$ on C_{CAAtt}

$Att \rightarrow TPM : activate\_identity(AIK_{h}, |AIK|)$ on C_{AttTPM}

$TPM \rightarrow Att : K$ on C_{TPMAtt}

$Att \rightarrow Meas : d$ on $C_{AttMeas}$

$Meas \rightarrow Att : e$ on $C_{MeasAtt}$

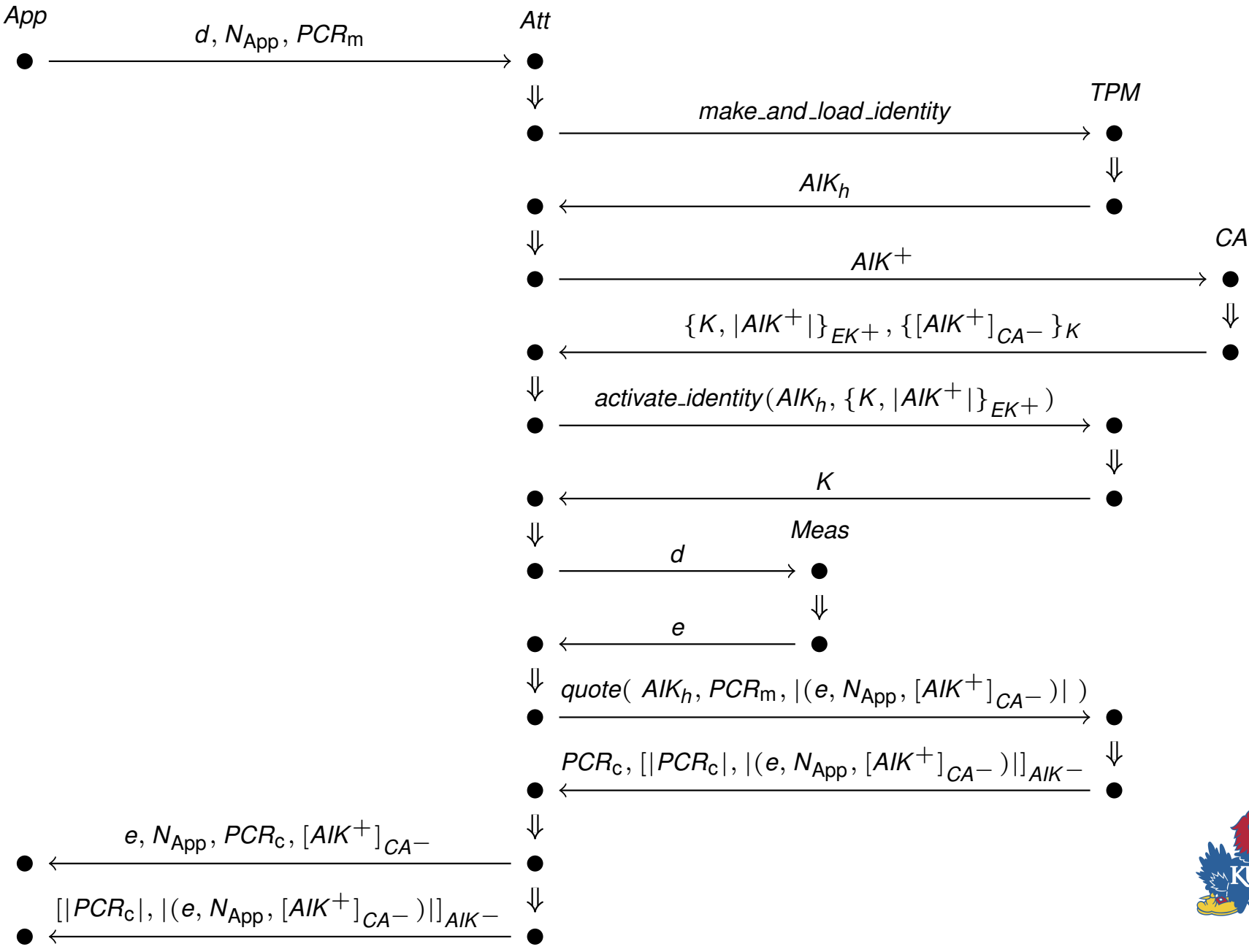
$Att \rightarrow TPM : quote(\ AIK_{\text{h}}, PCR_{\text{m}}, |(e, N_{\text{App}}, [AIK^{+}]_{CA^{-}})|\ )$ on C_{AttTPM}

$TPM \rightarrow Att : PCR_{\text{c}}, [|PCR_{\text{c}}|, |(e, N_{\text{App}}, [AIK^{+}]_{CA^{-}})|]_{AIK^{-}}$ on C_{TPMAtt}

$Att \rightarrow App : e, N_{\text{App}}, PCR_{\text{c}}, [AIK^{+}]_{CA^{-}}$ on C_{AttApp}

$Att \rightarrow App : [|PCR_{\text{c}}|, |(e, N_{\text{App}}, [AIK^{+}]_{CA^{-}})|]_{AIK^{-}}$ on C_{AttApp}

Strand Space Diagram Representation

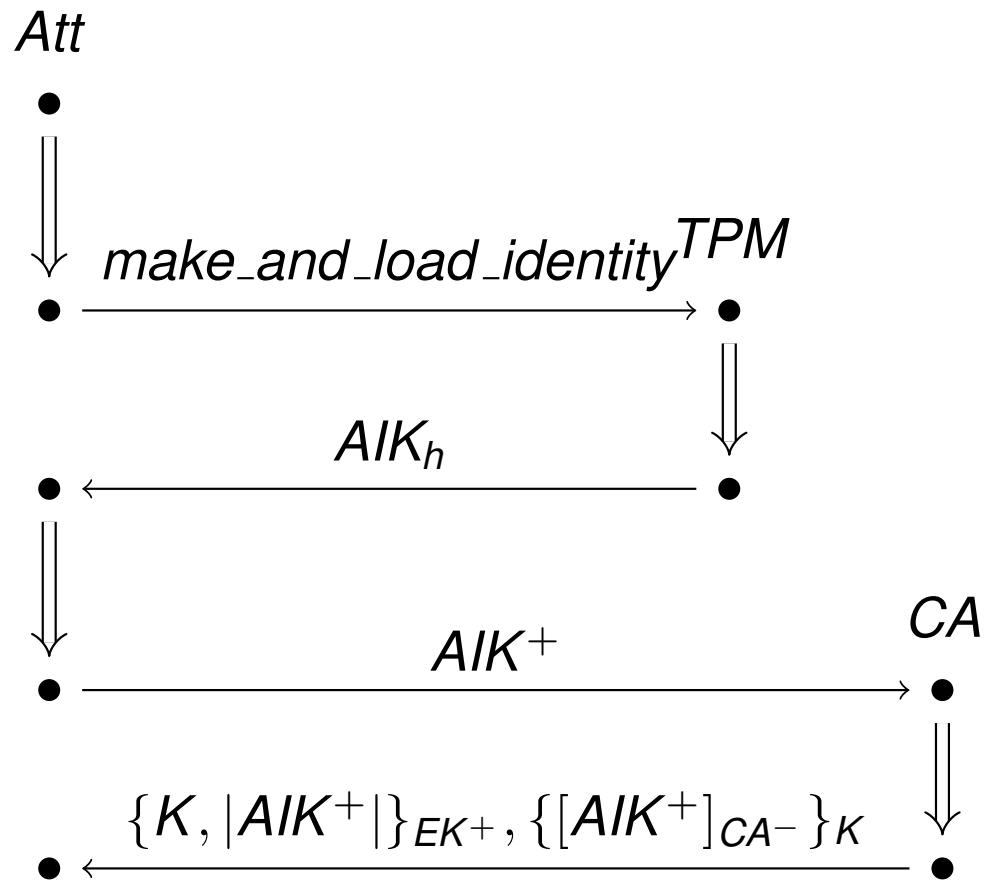


Attestation Request

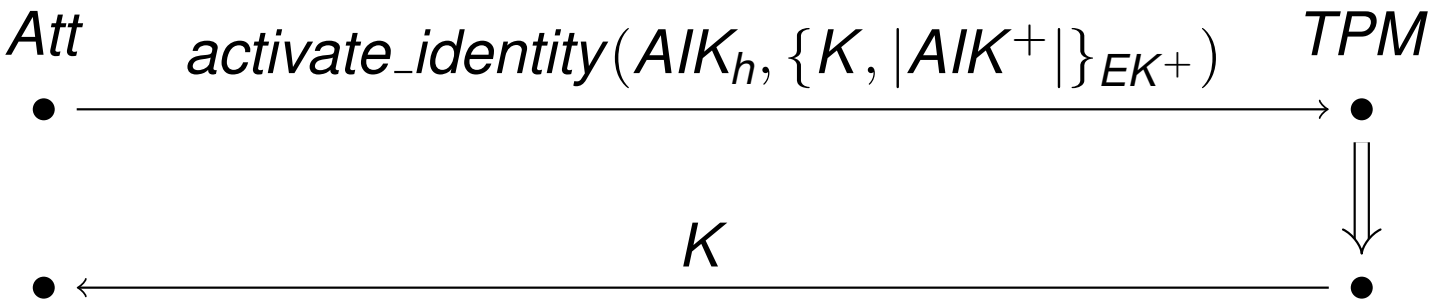
- Initiate with an attestation request
  - d abstractly defines desired evidence
  - N_{App} is the appraiser's nonce
  - PCR_{m} selects PCRs
- Attestation agent selects and executes protocol based on request

Generating and Certifying an AIK

- Request a new *AIK* from TPM (optional)
- Receive *AIK* handle
- Request AIK^+ signed by CA (*AIK* cert)
- Receive *AIK* cert encrypted with session key *K*
- Receive *K* encrypted with public *EK*

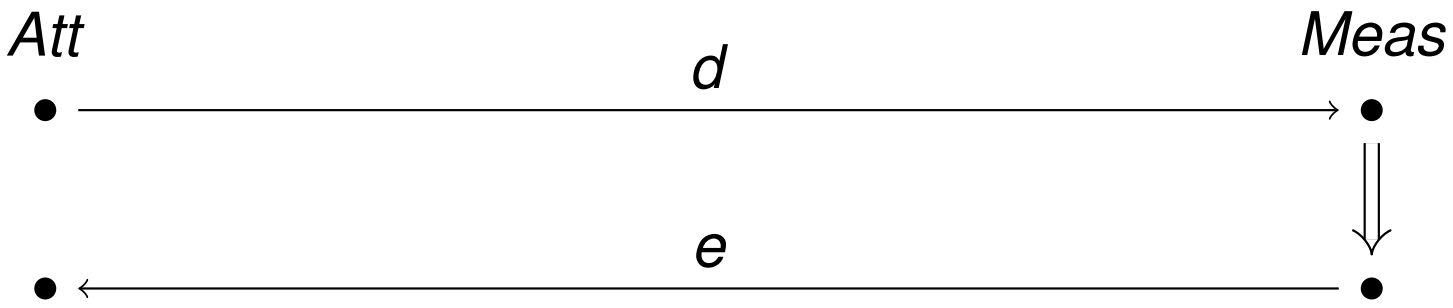


Activating the AIK



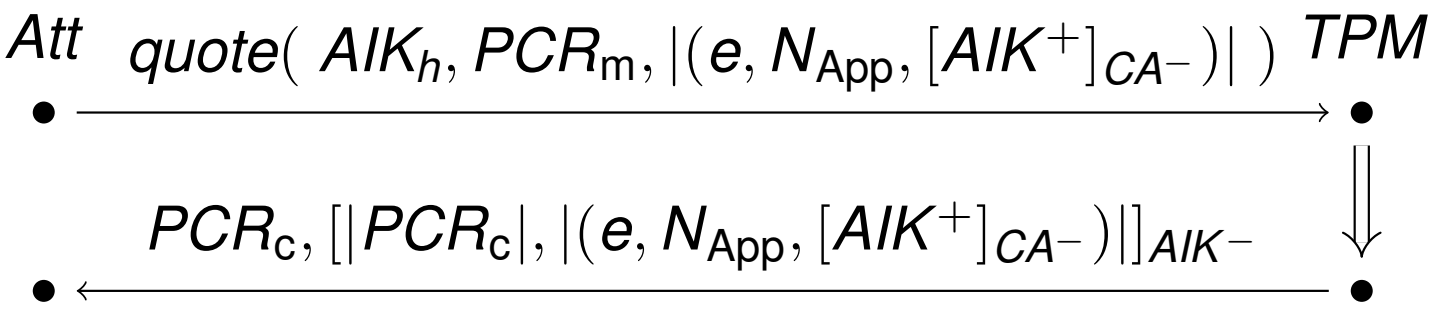
- Request TPM decryption of the *AIK* cert
- Receive K used to decrypt signed public *AIK*
- Only TPM can gain access to K
- Only TPM can obtain signed, public *AIK*
- Oddly, No manipulation of the *AIK* in this “activation” process

Measurement



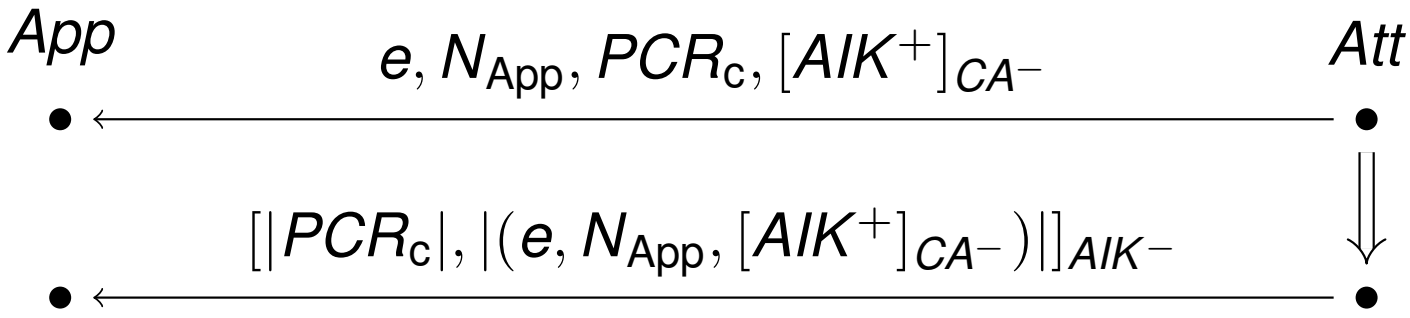
- Request information from measurer
- Receive evidence *e* from measurer
- *d* is abstract allowing protocol reuse
- Most protocols make many requests of the measurer

Generating a Quote



- Request a quote from the TPM
  - *AIK* identifies the signing *AIK*
  - PCR_m identifies desired PCRs
  - $|(e, N_{App}, [AIK^+]_{CA^-})|$ guarantees integrity of returned evidence
- Receive quote from TPM
  - PCR_c is PCR composite built from requested PCRs
  - $[|PCR_c|, |(e, N_{App}, [AIK^+]_{CA^-})|]_{AIK^-}$ is the signed quote

Appraisal



- Receive evidence from the attestation manager
  - evidence
  - original nonce
  - PCR composite
  - signed AIK^{+}
- Receive TPM quote from the attestation manager
  - hash of all evidence
  - PCR composite
  - signed by AIK^{-}
- Evaluate evidence and quote

Evidence Appraisal

Demonstration detects failure of all aspects of attestation

$$e, N_{\text{App}}, PCR_{\text{c}}, [AIK^{+}]_{CA^{-}}$$

- e – evidence gathered from running application
- N_{App} – prevents replay
- PCR_{c} – evidence in the form or PCR data from the vTPM
- $[AIK^{+}]_{CA^{-}}$ – ensures validity of AIK^{+}

$$[|PCR_{\text{c}}|, |(e, N_{\text{App}}, [AIK^{+}]_{CA^{-}})|]_{AIK^{-}}$$

- PCR_{c} – hash ensures integrity of PCR data
- $|(e, N_{\text{App}}, [AIK^{+}]_{CA^{-}})|$ – hash ensures integrity of evidence, nonce, and signed AIK^{+}

Attestation Protocol Execution

3-4 Slides on Measurement

2-3 Slides on Communication Mechanisms

CA communication

Shared notion of AIKCertRequest, AIKCert, and CAResponse JSON structures.

Attester

- creates an AIKCertRequest (containing attester *ID*, AIK^+) and converts to JSON
- JSON sent as POST request to CA running as web server

Certificate Authority

- POST body bytes $\rightarrow$ UTF8 $\rightarrow$ JSON $\rightarrow$ AIKCertRequest
- looks up EK^+ associated with *ID* in sql database
- AIKCert $=$ AIK^+ signed with CA^-
- generates key *K* and encrypts with EK^+
- AIKCert encrypted with *K*
- both wrapped in a CAResponse, converted to JSON and sent as response.

CA communication continued

Properties

- CA only responds to receiving an $AIKCertRequest_{JSON}$
- The CACert can *only* be decrypted by knowing K (and therefore EK^{-})

Appraiser Knowledge after receiving Cert:

- signature on AIK ensures it was CA who generated signature
  +
- only an entity knowing EK^{-} could decrypt and send the CACert
  =
- **Attester is using a registered TPM**

Current Status

Completed four demonstrations culminating in running an attestation protocol in response to an attestation request.

- Attestation and Appraisal development
  - CA-Based attestation protocol execution example
  - integration with Berlios TPM 1.2 emulator
  - simple dynamic appraisal of attestation results
- Measurement development
  - on demand Java program measurement
  - HotSpot-based Java VM run time measurements
  - standard mechanism for extending measurement capabilities
- Communication infrastructure
  - vchan, TCP/IP and socket communication infrastructure
  - language-based interface with TPM 1.2
  - JSON-based data exchange formats
  - initial certificate authority API

Goals and Milestones for 2015

- Push to the cloud
- Establish roots of trust and trust argument
- Executable protocol representation and protocol semantics
- Operational, integrated vTPM prototype
- Name Server / Certificate Authority prototype
- More capable measurement
- Downloadable demonstration

Questions and Guidance

- What problems are interesting?
- What problem would be a nice attention grabber?
- What should we be watching and integrating with?

References

G. Coker, J. Guttman, P. Loscocco, A. Herzog, J. Millen, B. O'Hanlon, J. Ramsdell, A. Segall, J. Sheehy, and B. Sniffen.
Principles of remote attestation.
International Journal of Information Security, 10(2):63–81, June 2011.

F. J. T. Fábrega, J. C. Herzog, and J. D. Guttman.
Strand spaces: Proving security protocols correct.
Journal of computer security, 7(2):191–230, 1999.

V. Haldar, D. Chandra, and M. Franz.
Semantic remote attestation – a virtual machine directed approach to trusted computing.
In *Proceedings of the Third Virtual Machine Research and Technology Symposium*, San Jose, CA, May 2004.