ArmoredSoftware Architecture Version 0.0

Perry Alexander Andy Gill Prasad Kuklarni
Leon Searl
Information and Telecommunication Technology Center
The University of Kansas
{palexand,andygill,prasadk,lsearl}@ku.edu

November 22, 2014

Contents

List of Figures

Abstract

This document describes the evolving ArmoredSoftware architecture. This includes the system architecture and descriptions of appraisal, attestation, measurement, and the vTPM infrastructure. Basic interactions among components are documented to define the overall ArmoredSoftware architecture. This is a living document and will be updated frequently.

1 Introduction

The objective of ArmoredSoftware is to *provide a portable trusted computing capsule for applications executing in the cloud.* This capsule, referred to as *armor*, provides three major functions:

Appraisal – Request and assess measurement information from the operational environment and other armored components.
Measurement – Gather run-time measurement information from its application
Attestation – Assemble and deliver evidence to appraisers in a manner that assures measurement integrity

and is based on concepts from Coker et al. [2011].

Figure 1 graphically depicts the major architectural components of a protected application. The *application* is the application to be protected by the infrastructure. The *measurement* component performs measurement operations on the running application while the *attestation* component gathers measurements and delivers them with cryptographic assurance of integrity and confidentiality. The *appraisal* component requests information from the environment and other

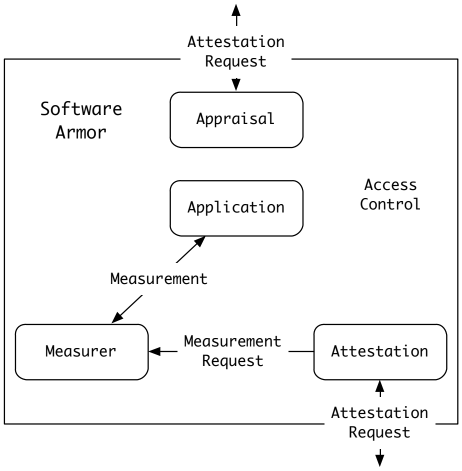


Figure 1: ArmoredSoftware component architecture showing major components of the remote attestation process.

components to assess the overall operational environment. *Access control* governs access to all critical resources in the protected application to assure secrets are preserved and enforce information flow restrictions.

Figure 2 graphically represents the interaction among protected components while Figure 3 shows the sequencing of interactions during appraisal. A component's appraisal module will request information from a second component's attestation module. The attestation model will select an attestation protocol that instructs the measurer what information to gather and in what sequence. The measurer executes that protocol that in turn gathers information from the running process, accesses the module's virtual TPM (vTPM) and makes appraisal requests of other ArmoredSoftware instances. The attestation module assembles measurement results into a evidence package that is returned to the requesting appraiser with cryptographic assurances of integrity and confidentiality as required. Upon receiving the package, the appraiser assesses cryptographic signatures and encryption to determine the trustworthiness of the measurements, then assesses measurements to determine the trustworthiness of the component being appraised.[1]

[1] Note that the same process occurs when appraising the component's operational environment with either the appraiser or target replaced by operational infrastructure.

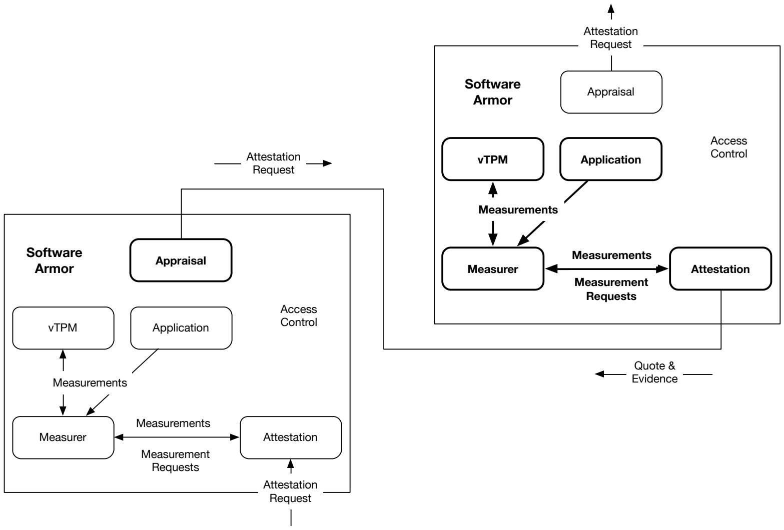


Figure 2: Typical interaction among armored components with an appraiser interacting with an attestation agent to gather information.

2 System Architecture

2.1 Measurement

Figure 4 graphically depicts the processing of requests for measurements by the measurement component. Figure 4(a) shows how a measurement request is processed while Figure 4(b) shows how a response is generated.

The measurement component is a collection of measurement capabilities for achieving various measurement goals. Measurement capabilities will range from simple checks to determine the presence or absence of resources to more sophisticated run-time monitoring. Measurement capabilities are broadly classified into four major groups:

- Application measurers—Measurement capabilities targeting a specific application protected by armor. Such capabilities range from simple hashes to persistent run-time monitoring with periodic measurement.
- Appraisal requests—Capabilities that will request attestation results from other armored components or components in the environment capable of performing attestation. Such capabilities reflect a means of reaching outside the component to other, armored components.
- Environment measurers—Measurement capabilities targeting the operational environment. Such capabilities target aspects of the environment that are not armored or do not otherwise have attestation capabilities. A

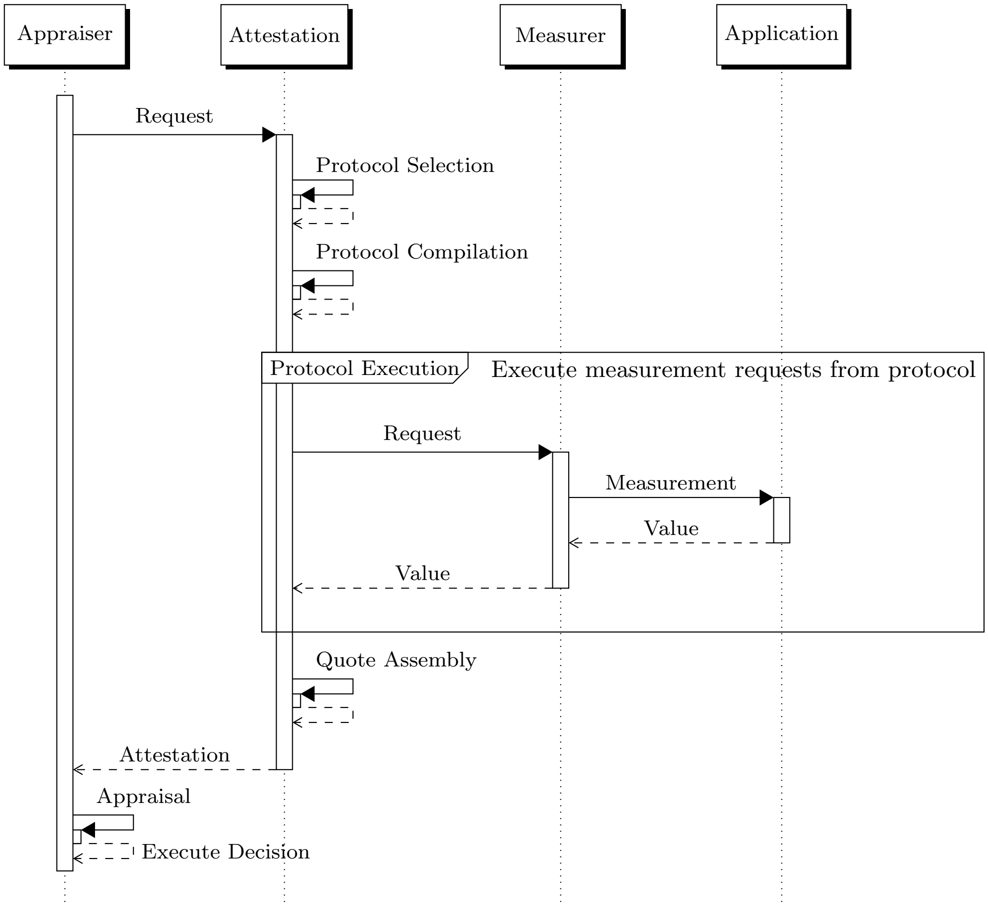
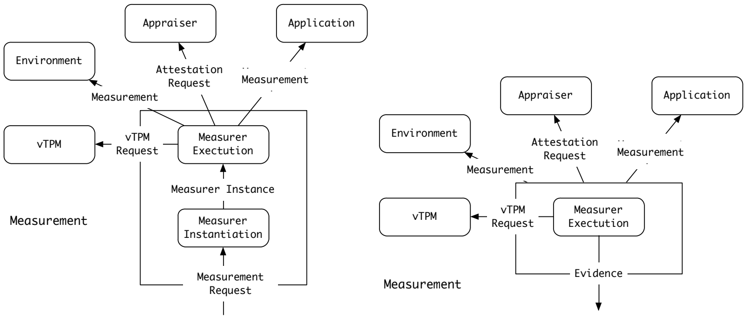


Figure 3: Architecture component interaction



(a) Request processing (b) Response processing

Figure 4: Measurement request processing and response

significant application of environment measurers will be legacy operating systems and applications.

- vTPM interactions—Capabilities that interact with the vTPM to both store and save measurement artifacts and perform other vTPM operations as needed. Such capabilities will enforce access control over PCRs and other operations in the vTPM in addition to performing their primary functions.

Measurer Instantiation chooses and instantiates a measurement capability for a particular measurement task. Measurement tasks from the attestation component are abstract requests for information. Measurement selection is responsible for instantiating those requests with specific, executable measurement instances.

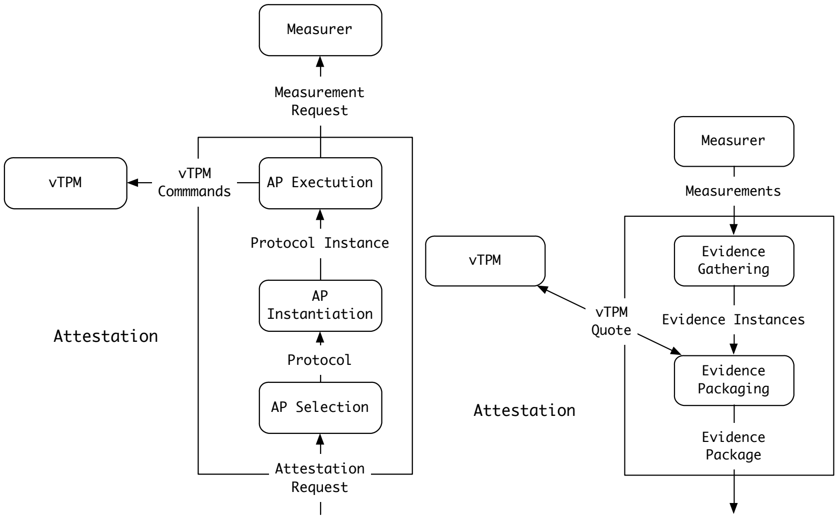
Measurement Execution causes a measurement capability to execute, manages execution and returns measurement results. Execution is more than simply execution of a program and includes monitoring, exception handling and any bookkeeping required to prepare and clean up following execution.

Measurement capabilities may need to access the vTPM for storing measurement results, accessing stored results, or for generating cryptographic evidence in support of assessment. The vTPM will not be a part of the measurer, but will be shared by all components comprising the armor.

The measurement subsystem is protected by access control. It is responsible for protecting the application and ensuring sensitive information is not leaked via the measurement process.

2.2 Attestation

Figure 5 graphically depicts the processing of an attestation request by an ArmoredSoftware attestation component. The attestation request specifies information requested by an appraiser. Upon receipt, the *Attestation Protocol*



(a) Request processing (b) Response processing

Figure 5: Attestation request processing and response

Selector or *AP Selector* identifies one or more *Attestation Protocols* that could satisfy the appraisers request. The protocol is passed to the *AP Instantiation* process that selects specific mechanisms for achieving individual requests in the AP. The resulting protocol instance is passed to *AP Execution* where it is executed by: (i) making requests to the component's vTPM; making requests to another components attestation service; or (iii) invoking the measurer on the component's associated application.

The results of request to the vTPM, another attestation component, and the local measurer are returned as vTPM quotes, evidence packages, and measurements respectively. The AP Execution component monitors execution and collects various results for processing by the AP Instantiation component. AP Instantiation assembles individual measurement, vTPM and appraisal results into a package representing information requested in the attestation request. The AP Selection component uses cryptographic techniques to provide assurances to the appraiser requesting information that evidence can be trusted. Finally, the evidence package is returned to the requesting appraiser where it is evaluated.

2.3 Appraisal

Appraisal assesses the result of an attestation request to determine if: (i) the evidence received is trustworthy; and (ii) if the system described by that evidence is trustworthy. While attestation is responsible for gathering and delivering measurement information, appraisal is responsible for assessing it.

2.4 TPM and vTPM

Elements of the ArmoredSoftware capsule will share access to a vTPM associated with each armor instance. That vTPM will be constructed with root-of-trust in a hardware TPM (hTPM), but must be migratable among elements of the cloud infrastructure.

2.5 Access Control

Access control is omnipresent in the armor and is responsible for protecting both the application and elements of the armor. We anticipate using a Flask-based MAC style, but may change that approach as we become more aware of our needs.

3 Component Interaction

3.1 Remote Attestation

3.2 Access Control

3.3 vTPM Management and Access

3.4 Migration

A Glossary

- $\mathbf{0}$ - null process
- $|M|$ - hash of M
- K^+ - public half of asymmetric key K
- K^- - private half of asymmetric key K
- $\{M\}_K$ - encrypt M with symmetric key K
- $\{M\}_{K^+}$ - encrypt M with the public key from K
- $\{M\}_{K^-}$ - decrypt M with the public key from K
- $\{|M|\}_{K^-}$ - sign M with the private key from K
- $\{|M|\}_{K^+}$ - check signature on M with the public key from K
- $(\nu x)P$ - new variable x defined in scope of P
- $\overline{c}\langle M\rangle$ - send M on channel c
- $c(M)$ - receive M on channel c
- $!P$ - infinite replication of P
- $P\ +\ Q$ - P or Q
- $P\ |\ Q$ - P in parallel with Q
- $\mathsf{case}\ \{M\}_k\ \mathsf{of}\ x\ \mathsf{in}\ P$ - attempt to decrypt $\{M\}_k$ and bind to x in P if successful. Stuck if unsuccessful

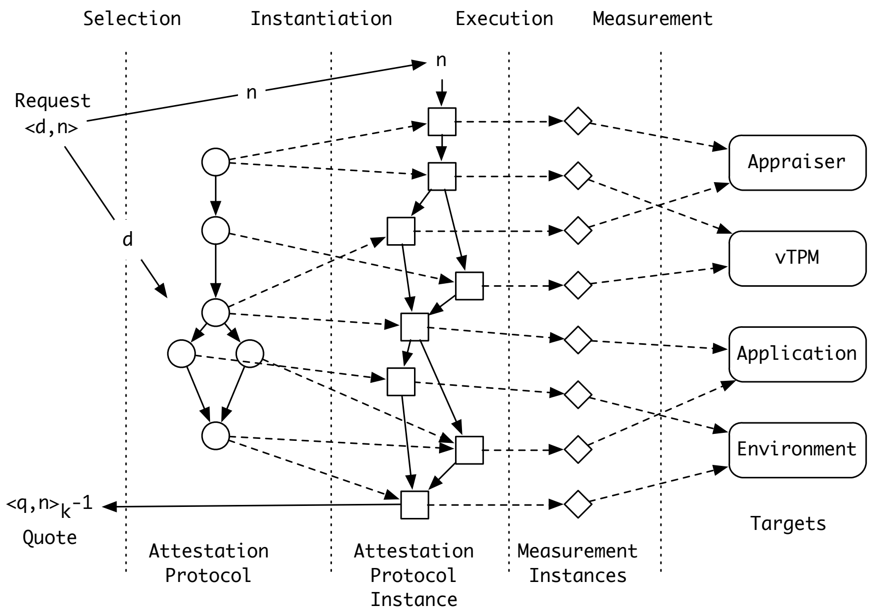


Figure 6: Interaction between the Attestation and Measurement subsystems when processing an attestation request.

- case $\{M\}_{k^-}$ of x in P - attempt to decrypt $\{M\}_{k^+}$ and bind to x in P if successful. Stuck if unsuccessful
- case $\{|M|\}_{k^+}$ of x in P - attempt to check signature $\{|M|\}_{k^-}$ and bind to x in P if successful. Stuck if unsuccessful
- case x of y $0 : Psuc(x) : Q$ - case splitting over integers. x is bound in Q.
- let $(x, y) = M$ in y - match M to (x, y) binding x and y to pair elements in M
- $A \triangleq B$ - define an equivalence
- $A \rightarrow B : M$ on c - A sends B message M on channel c

$$\begin{aligned} A \quad \triangleq \quad & (\nu c)\, \overline{c}\langle M\rangle.\mathbf{0} \;\mid \\ & c(M).A \end{aligned}$$

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.