

ArmoredSoftware Architecture Version 0.0

Perry Alexander Andy Gill Prasad Kuklarni
Leon Searl

Information and Telecommunication Technology Center
The University of Kansas
`{palexand, andygill, prasatk, lsearl}@ku.edu`

January 8, 2014

Contents

1	Introduction	1
2	System Architecture	3
2.1	Measurement	3
2.2	Attestation	4
2.3	Appraisal	6
2.4	TPM and vTPM	6
2.5	Access Control	6
3	Component Interaction	6
3.1	Remote Attestation	6
3.2	Access Control	6
3.3	vTPM Management and Access	6
3.4	Migration	6
A	Glossary	6

List of Figures

1	ARMORED SOFTWARE component architecture showing major components of the remote attestation process.	2
2	Typical interaction among armored components with an appraiser interacting with an attestation agent to gather information. . . .	3
3	Architecture component interaction	4
4	Measurement request processing and response	5
5	Attestation request processing and response	5
6	Interaction between the Attestation and Measurement subsystems when processing an attestation request and resulting evidence. .	7

Abstract

This document describes the evolving ARMORED SOFTWARE 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 ARMORED SOFTWARE architecture. This is a living document and will be updated frequently.

1 Introduction

The objective of ARMORED SOFTWARE 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 ?.

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

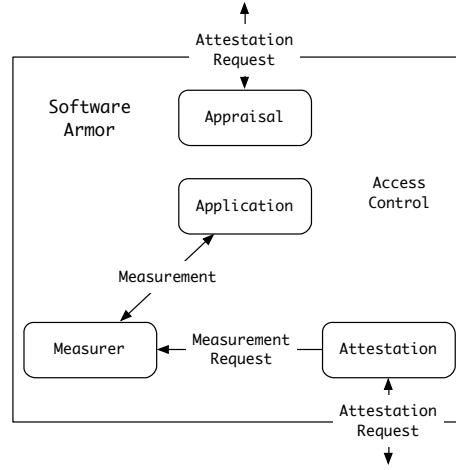


Figure 1: ARMORED SOFTWARE component architecture showing major components of the remote attestation process.

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 ARMORED SOFTWARE instances. The attestation module assembles measurement results into an 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

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

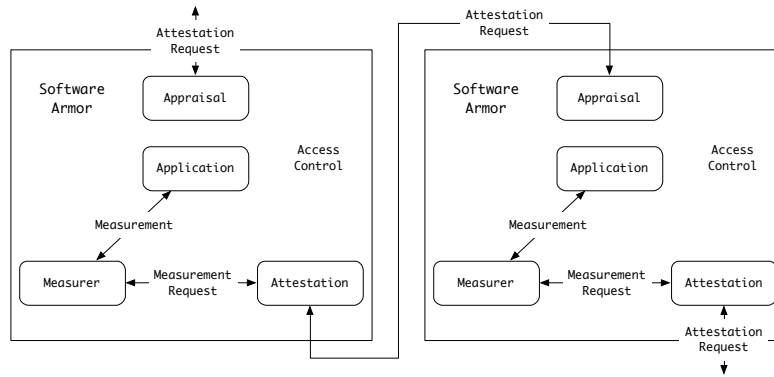


Figure 2: Typical interaction among armored components with an appraiser interacting with an attestation agent to gather information.

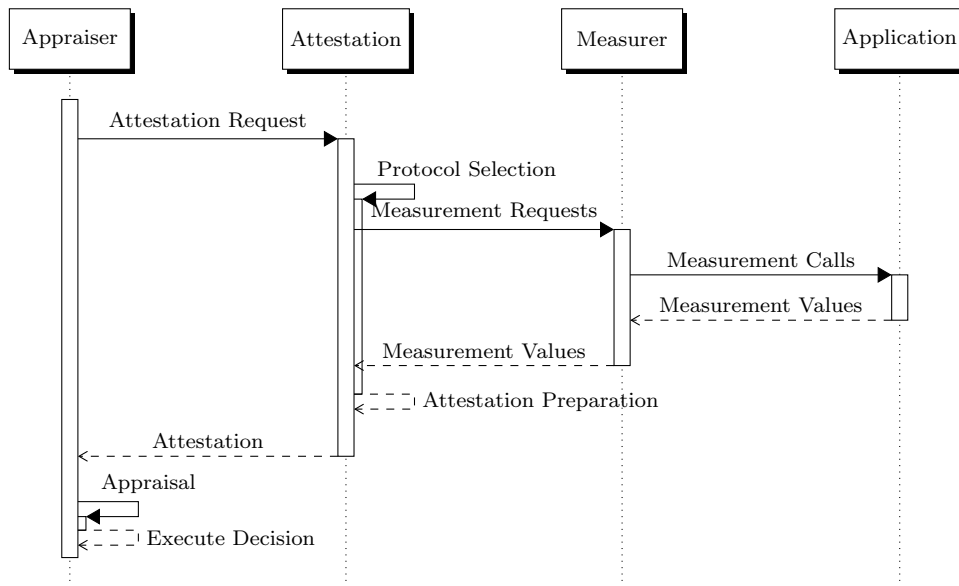


Figure 3: Architecture component interaction

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 sophisticated run-time monitoring capabilities. It will be possible to add and remove measurement capabilities from the measurer at run-time.

Measurement selection 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 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 measure-

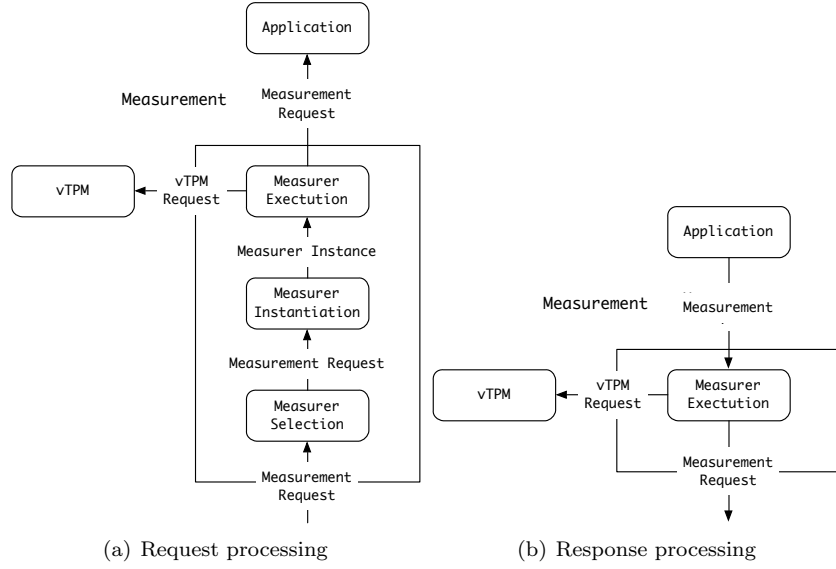


Figure 4: Measurement request processing and response

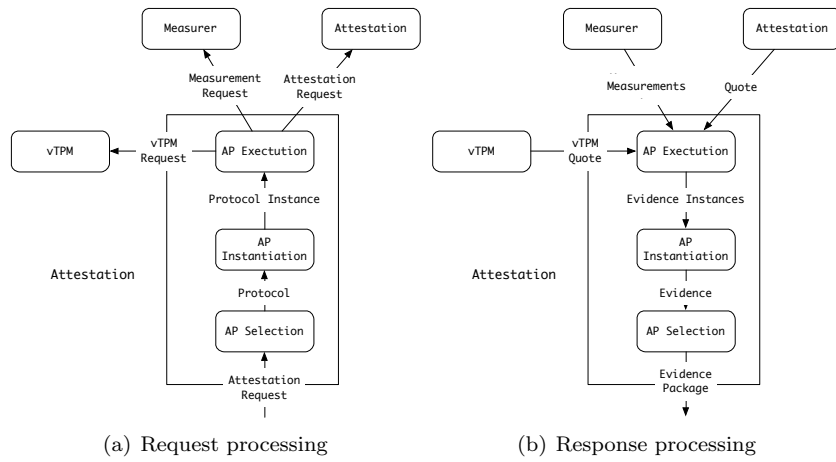


Figure 5: Attestation request processing and response

ments 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.

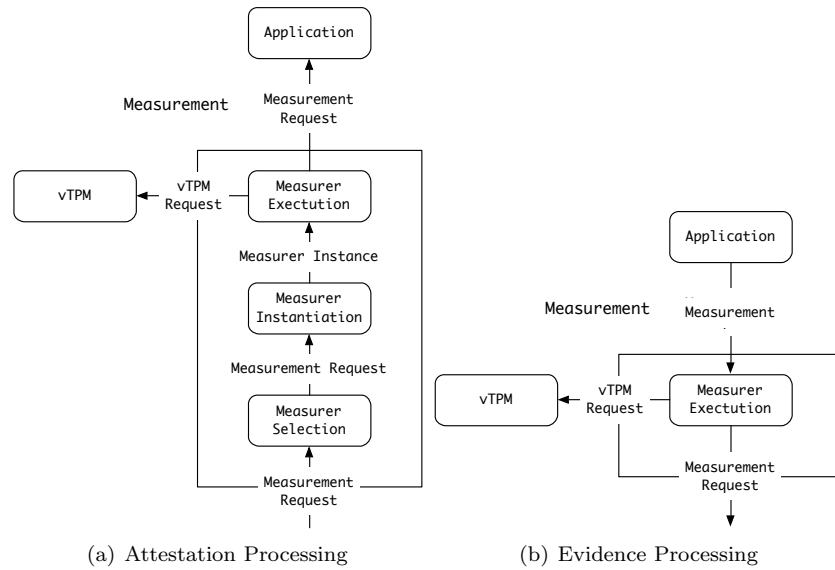


Figure 6: Interaction between the Attestation and Measurement subsystems when processing an attestation request and resulting evidence.

3 Component Interaction

3.1 Remote Attestation

3.2 Access Control

3.3 vTPM Management and Access

3.4 Migration

A Glossary

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.