# Protocol Ordering Musings

Anna Fritz

February 24, 2023

GOAL: The goal of this document is to:

1. Reason about the context relation so that the target is able to better realize the dependencies within their system
2. Define an example system to realize necessary properties
3. Define weak bisimulation of measurement. In order to do this, we must have a (labeled?) transition system so we need to define a transition system and all supporting details.

We aim to solve the following research questions: Can we define a weak bisimulation over measurements such that one measurement is better than another?

## Contents

# 1 Context Relation Defined

First, in order to do any real reasoning about a system's context, we must formally define it.

**Definition 1** (Context)**.** *A system's* context *is a relationship represented within a Manifest that describes the dependencies within the system.*

Some questions and interesting points arise when attempting to grasp this definition. Below are some words, related to context, I think we need to define as we work towards the ordering problem.

Correctly defining a dependency is critical as we need to capture the intricacies of a layered system. We want to take bottom up measurements for many reasons, one being to confine an adversary [5].

**Definition 2** (Dependency)**.** *One component* depends *on another when...*

**Definition 3** (Separation)**.** Separation *between components within the system is realized when... (this definition is motivated by [4] )*

**Definition 4** (Similar)**.** *One protocol is* similar *to another when... They measure the same targets? The measurement chain reaches the same depth? The measurement chain reaches the same range? .. consider defining "similar" as a weak bisimulation*

**Definition 5** (Better)**.** *One protocol is* better *than another when... consider better as the top of the lattice or the "highest" protocol in the preorder.*

**Definition 6** (Trust Chain)**.** *A* trust chain *is a measurement chain that begins at the root of trust and proceeds by taking measurements of the next closest layer until the top layer is measured thereby producing a layered attestation.*

# 2 Measures Relation Defined

**Definition 7** (Measures)**.** *One attestation manager can measure another when the location of the measurement target is known and the resulting evidence produced by the target does not violate a policy.*

While context describes dependencies within the system, measures defines additional attestation component the current system is aware of. Both relations are important for building a trust chain.

# 3 Mathematics

Here are some interesting mathematically defined structures that could be useful.

**Definition 8** ((Labeled) Transition System)**.** *A* transition system *is a triple* $< S, S_0, \rightarrow >$*, with a set of states* $S$*, a set of initial states* $S_0 \subseteq S$*, and a transition relation* $\rightarrow \subseteq S \times S$ *[1]*

Related to a transition system, we could define *reachable*, an *invariant*, and possibly some *correctness* condition. Recall from class, an invariant is a property that is true in all states. A state is reachable if there is some starting state from which we can transition to the reachable state.

We may want to reason about protocol behavior to say that if two protocols produce the same evidence, then they are behaviorally equivalent. To do this, we may need to use a weak bisimulation.

**Definition 9** (Weak Bisimulation)**.** *A weak bisimulation is a relation* $\boldsymbol{R}$ *such that* $P \ \boldsymbol{R} \ Q \implies \forall \mu, P, P' (P \xrightarrow{\mu} P' \implies \forall Q'.Q \overset{\mu}{\Rightarrow} Q'$ *and* $P' \ \boldsymbol{R} \ Q'$ *) [2]*

This says if we have some relation that relates P and Q and we can take some step from P to P' through $\mu$ and some potentially different steps from Q to Q' but again through $\mu$ then P' is related to Q'.

this doesn't seem right?

**Definition 10** (Partially Ordered Set)**.** *A partially ordered set* $(E, \preceq)$ *is defined by it's underlying set* $E$ *and it's ordering relation* $\preceq$

Consider *upper bound* and *lower bound* of the poset for possible interesting properties.

define our ordering relation. Maybe its ordered by dependencies?

# 4 Example System

A key facet of attestation system design is the idea of *means of isolation* which protects attestation components from potentially untrusted measurement targets [3]. In essence, we assume an attestion component (ie measurer) is good because of its formal guarentees rather than stating it is good because of the result of measurement. As we present the following example, we assume all attestation components uphold a means of seperation and are thus individually trusted.

Consider the example presented in Figure 1 where a request is sent to a target $Host1$. We know a request is a list of protocols as Copland phrases. Let's assume that $R = [\ @_{Host1}\ (attest\ \text{Host2 t2})\ ]$ which says some relying party would like the target system to preform an *attest* measurement of the target $T2$ located on $Host2$.

The arrow between ASP and T2 reveals that ASP can measure T2. This results in the following Copland phrase:

$$@_{Host1}\,(attest\ \ Host2\ \ T2)$$

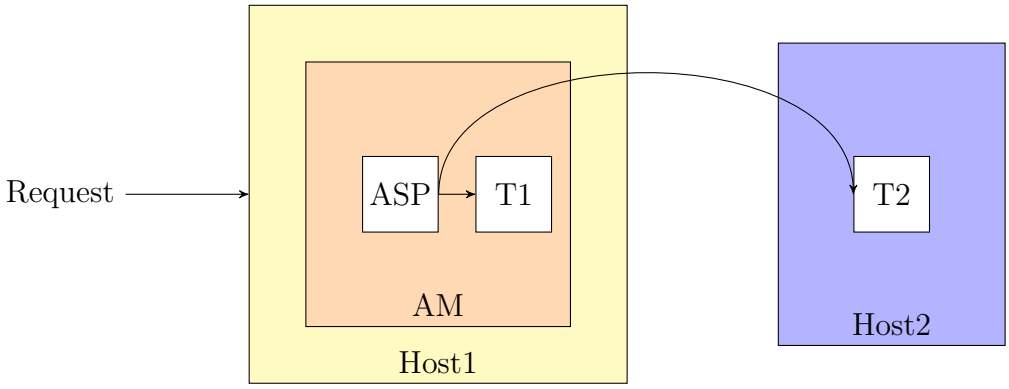


Figure 1: Example Target System.

For examples sake, say that ASP no longer has direct access to the measurement target T2 yet the goal of the attestation was to measure T2. This means we will have to realize a measurement of T2 is a different way. To start, we can expand Host2 to get a view of its attestation capabilities. This is represented with Figure 2. We see Host2 has some ASP, $ASP2$ which can preform a measurement of $T2$. The integrity of the measurer $ASP2$ depends on the integrity of the following components: the operating system (OS), the device driver (DR), and some cryptographic operations $\{\}_K$. Then, these components all depend on the integrity of the root of trust measurement (RoTM). These dependencies are represented within Host2's manifest.

In Copland, Host2 may have the following measurements:

- ROTM – (query_img bootMem RoTM) [3]
- OS – (kim Host2 OS)
- DR – (kim Host2 DR)

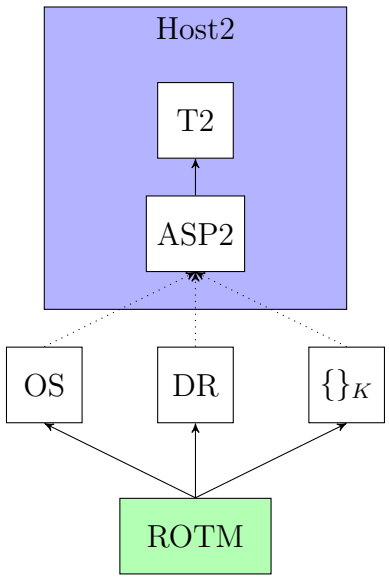


Figure 2: Example Host2 System.

- $\{\}_K$ – (kim Host2 crypto)
- *attest* Host2 T2

Combining these measurements together may produce the following Copland phrase:

```
@ Host2 [
  (query_img bootMem RoTM) ->
  ((kim Host2 OS) ~ (kim Host2 DR) ~ (kim Host2 crypto)) ->
  (ASP2 Host2 T2)]
```

where $\rightarrow$ represents sequenced measurement and $\sim$ represents parallel measurement. In other words, this phrase says, measure the root of trust, then measure OS, DR, and $\{\}_K$ in parallel, then finally use the attest ASP to take a measurement of T2.

I have no idea how to use parathesis and Copland...

For examples sake, say that ASP no longer has access to the measurement of T2 directly yet the goal of the attestation is to measure T2. Combining Host1 and Host2 and representing this new fact, we have the following system in Figure 3. If the request remains

To be technically correct, how do you phrase the relationship between OS, DR, and SIG and the ROTM?

$R =$ [ (ASP Host2 T2) ] we must consider what would be a "similar" measurement? This idea captures the concept of ordering measurements using the *measures* and *context* relations.

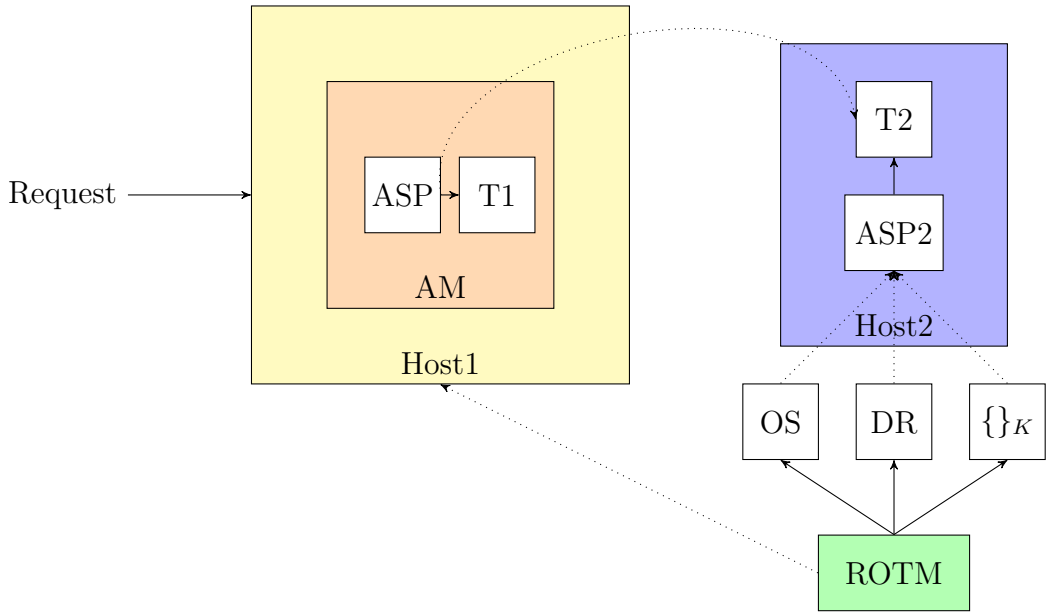


Figure 3: Total system.

define similar?

Say we have the following protocol, as would be represented in Figure 1 and Figure 2.

$$
\begin{aligned}
P1 &= @_{Host1}\,(ASP\ \ Host2\ \ T2) \\
P2 &= @_{Host2}\,(ROTM \rightarrow (OS \sim DR \sim \{\}_K) \rightarrow (ASP2\ \ Host2\ \ T2))
\end{aligned}
$$

Is it correct to say that P1 = P2? Both protocols satisfy the goal of measuring T2. It feels wrong to say they are equivalent. Maybe the relationship between protocols is best represented by an ordering operator. We could say that P2 $\leq$ P1. The certainly feels like a step in the right direction of the lattice. However, we need to explicitly define how P1 is "better". Is it better because it more directly measures T2 and therefore requires fewer resources.

Or maybe I have this ordering wrong and P2 is better because it reveals more about the integrity of the measurer ASP2.

Maybe now time to formally define better?

# 5 Protocols as Transition System

One thought is to define a weak bisimulation over Copland terms but, in order to do this, we must have a transition system for protocols. To do this, we start with the Copland grammar as presented abstractly by [6] below:

$$
\begin{array}{rcll}
C & :: & A(V) & \text{Actions with arguments} \\
 & | & C \rightarrow C & \text{Linear sequence} \\
 & | & C \prec C & \text{Sequential branching} \\
 & | & C \sim C & \text{Parallel branching} \\
 & | & @_P\,[C] & \text{At place} \\
 & | & (C) & \text{Grouping}
\end{array}
$$

The grammar is parameterized over atomic actions. The atomic actions are placed together by nonterminal actions such as: linear sequencing, sequential branching, parallel branching, at operations, and grouping. Because of this structure and other works on layered attestation, I believe the set of states are the set of measurement operations [5, 4]. This thought has been confirmed after looking at LTS.v semantics.

To encourage this idea, consider the architecture presented by Rowe in Figure 4. Here, the UserVM has three measurement components *sys*, *vc*, *kernel*. The UserVM has a sibling VM which hosts two measurement components *A1* and *A2*. Both these VMs are managed by a Hypervisor which runs on top of some Hardware contain a TPM. This hardware includes the root of trust for measurement.

Realizing this system, we can derive a bottom up measurement scheme which, taken from [5], can be visualized below. In this diagram, we propose that time begins at the top with the measurement of the *rtm*. This implies time flows downwards. In this notation, we consider a measurement event $ms_{ker}(vc, sys)$ imples the event of *vc* measuring *sys* in the context of *ker*. Another example is $ms(rtm, A_1)$ where the measurement of $A_1$ is performed in the context of a sound measurement of *rtm*. This clearly captures system dependencies.

With this notation, I think contextual dependencies are represented as pairs

So, now we have the set of **states** which are measurement events. The initial state is either some **rtm** (root of trust measurement) or some **att-start**

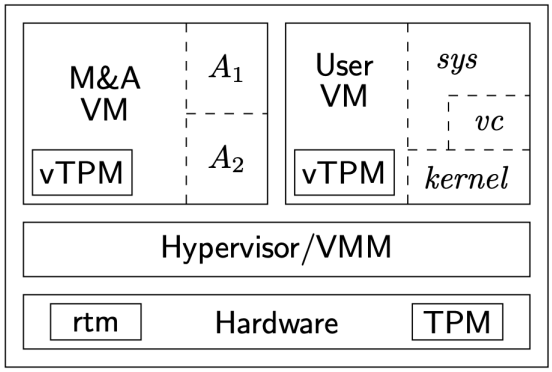


Figure 4: An example attestation system defined by [5].

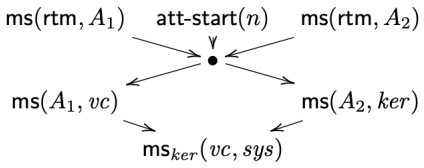


Figure 5: An example of measurement ordering from the previous system as defined by [5].

event. **The transition relation is then the measurement relation.**

What should the labels of the transition be??

Now, back to the example system presented in Figure 2. We can represent this in a dependency like structure below similar to the structure used by [5].

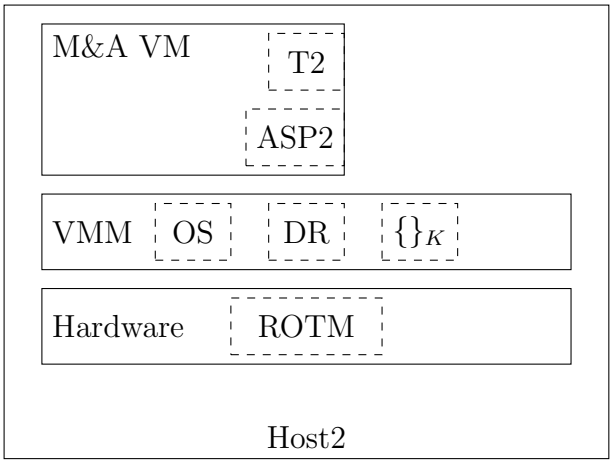


Figure 6: Host2 presented as one system.

We can then maybe(?) see how to create a transition system using the measures and context relation. Assume each component within the architecture runs their own attestation manager: one for the hardware, one for the VMM, and one for the M&A VM. Both the Hardware and VMM attestation managers take static measurements during the boot sequence while the M&A VM takes dynamic measurements during runtime. This distinction is noted but could be insignificant for this discussion.

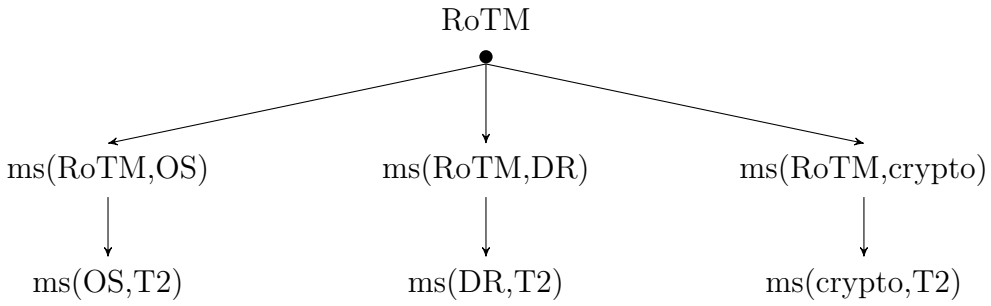


Figure 7: Host2 measuring order.

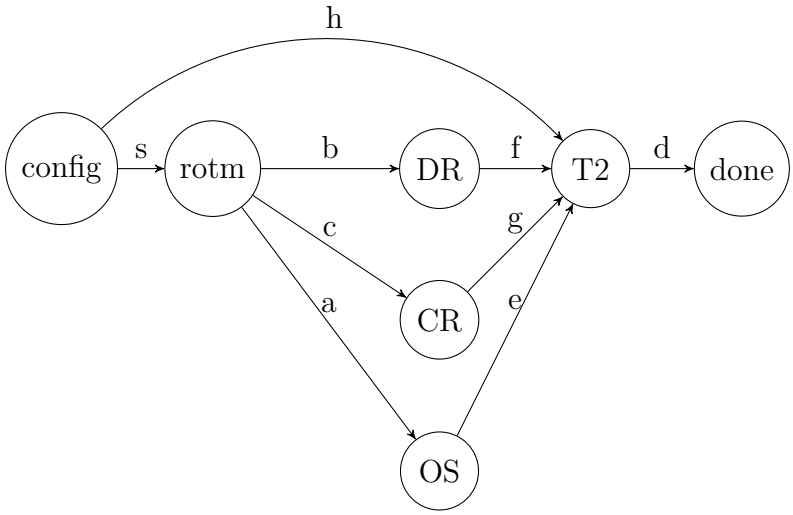


Figure 8: Protocols as transition system.

# 6 Small-Step Semantics

Small step semantics breaks a program's execution down into small sequential steps [1]. This idea differs from the idea of big-step semantics which reasons about a program's start state and final state, disregarding intermediate operations that may occur. Since we care about the intermediate steps of a protocol, we will move forward with defining a small-step semantics of protocol evaluation (?).

Within small-step semantics, one can define single step relations which transform the program from one state to the next by taking one execution step. For example, $c \rightarrow c'$ says start at state $c$ then take a step to the resulting state $c'$. In Orchestrating paper [4], authors define small-step semantics for Copland protocols. In this grammar, we define states $S$ where P is place, E is evidence, and T is annotated terms.

| | | | |
|---|---|---|---|
| $S$ | $\leftarrow$ | $\mathcal{C}(T, P, E)$ | configuration state |
| | \| | $\mathcal{D}(P, E)$ | stop state (place p with evidence t) |
| | \| | $\mathcal{A}(\mathbb{N}, P, S)$ | state resulting from @ term |
| | \| | $\mathcal{LS}(S, T)$ | state used when evaluating linear sequence |
| | \| | $\mathcal{BS}^l(\mathbb{N}, S, T, P, E)$ | evaluating left subterm |
| | \| | $\mathcal{BS}^r(\mathbb{N}, E, S)$ | evaluating right subterm |
| | \| | $\mathcal{BP}(\mathbb{N}, S, S)$ | parallel evaluation |

Authors proved these semantics exhibit correctness, progress, and termination.

- **Correctness** says that the operational semantics are proven equivalent to the denotational semantics or that if you have some configuration state $C(t, p, e)$ and you can take one or many transitions to a done state $D(p, e')$ then that relation maps to the evidence semantics of $\bar{\mathcal{E}}(t, p, e) = e'$.
- **Progress** says that either the phrase is in the done state $\mathcal{D}(p, e)$ or it it in some state such that it can be evaluated further.
- **Termination** says that for some number of transitions n, the configuration state can be transformed to the done state.

In these semantics **measurement order** is preserved. This means that each step of the transition system is labeled with a natural number to introduce order. I thought I could start at the Orchestrating paper and utilize their semantics to prove a weak bisimulation between protocols. But labeling with only the natural number isn't enough.

# References


[1] Chlipala, A. *Formal Reasoning About Programs*. Online, 2021.

[2] Milner, R. *Communication and Concurrency*. Prentice-Hall, Inc., USA, 1989.

[3] Petz, A., Jurgensen, G., and Alexander, P. Design and formal verification of a copland-based attestation protocol. In *Proceedings of the 19th ACM-IEEE International Conference on Formal Methods and Models for System Design* (New York, NY, USA, 2021), MEMOCODE '21, Association for Computing Machinery, p. 111–117.

[4] Ramsdell, J. D., Rowe, P. D., Alexander, P., Helble, S. C., Loscocco, P., Pendergrass, J. A., and Petz, A. Orchestrating layered attestations. In *Principles of Security and Trust* (Cham, 2019), F. Nielson and D. Sands, Eds., Springer International Publishing, pp. 197–221.

[5] Rowe, P. D. Bundling evidence for layered attestation. In *Trust and Trustworthy Computing* (Cham, 2016), M. Franz and P. Papadimitratos, Eds., Springer International Publishing, pp. 119–139.

[6] Rowe, P. D. On orderings in security models. In *Protocols, Strands, and Logic - Essays Dedicated to Joshua Guttman on the Occasion of his 66.66th Birthday* (2021), D. Dougherty, J. Meseguer, S. A. Mödersheim, and P. D. Rowe, Eds., vol. 13066 of *Lecture Notes in Computer Science*, Springer, pp. 370–393.