Context Relation Musings

Anna Fritz

February 17, 2023

1 Context 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 and realize the following example. Below are some words 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 [4].

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 [3])*

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 emphtrust 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 Mathematics

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

Definition 7 ((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]*

I believe this definition raises many questions. What is the initial state? What are the set of state? What is the transition relation? A state could be some atomic action and a transition relation could be any control flow action. Another thought is only the @ operator is a transition relation as it transfers the data and control from one plate to another. These questions are all answered below in the Transition System section.

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 8 (Weak Bisimulation)**.** *A weak bisimulation is a relation* ***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'* ***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 μ and some potentially different steps from Q to Q' but again through μ then P' is related to Q'.

this doesn't seem right?

Definition 9 (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 emphlower bound of the poset for possible interesting properties.

define our ordering relation. Maybe its ordered by dependencies?

3 Example System

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}\ $ (ASP Host2 t2)] which says some relying party would like the target system to preform a 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:

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

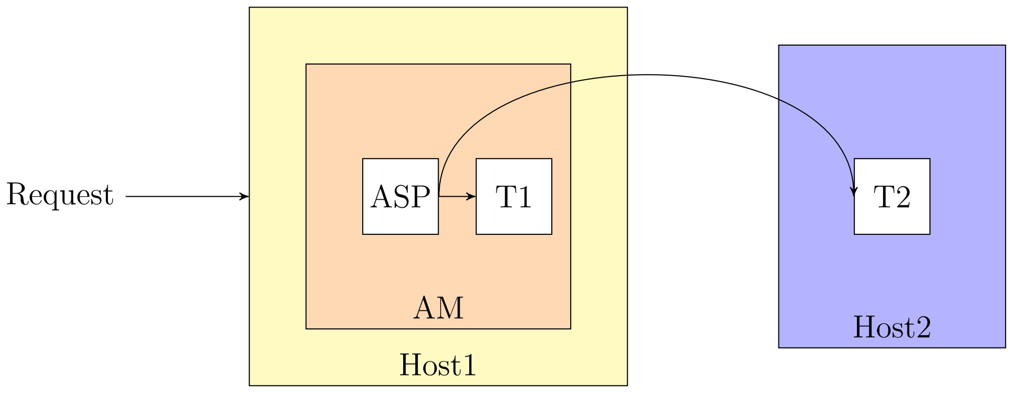

Figure 1: Example Target System.

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 an abstract-ish version of Copland, the measurement chain for HOST2 could look like:

$$@_{Host2}\,(ROTM \rightarrow (OS \sim DR \sim \{\}_K) \rightarrow (ASP2\ \ Host2\ \ T2))$$

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

where $\rightarrow$ represents sequenced measurement and $\sim$ represents parallel measurement. In other words, this phrase says, measure the root of trust,

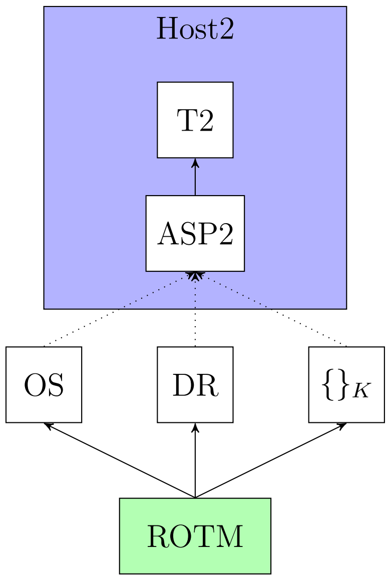

Figure 2: Example Host2 System.

then measure OS, DR, and $\{\}_K$ in parallel, then finally use ASP2 to take a measurement of T2.

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

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
$R =$ [(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.

define similar?

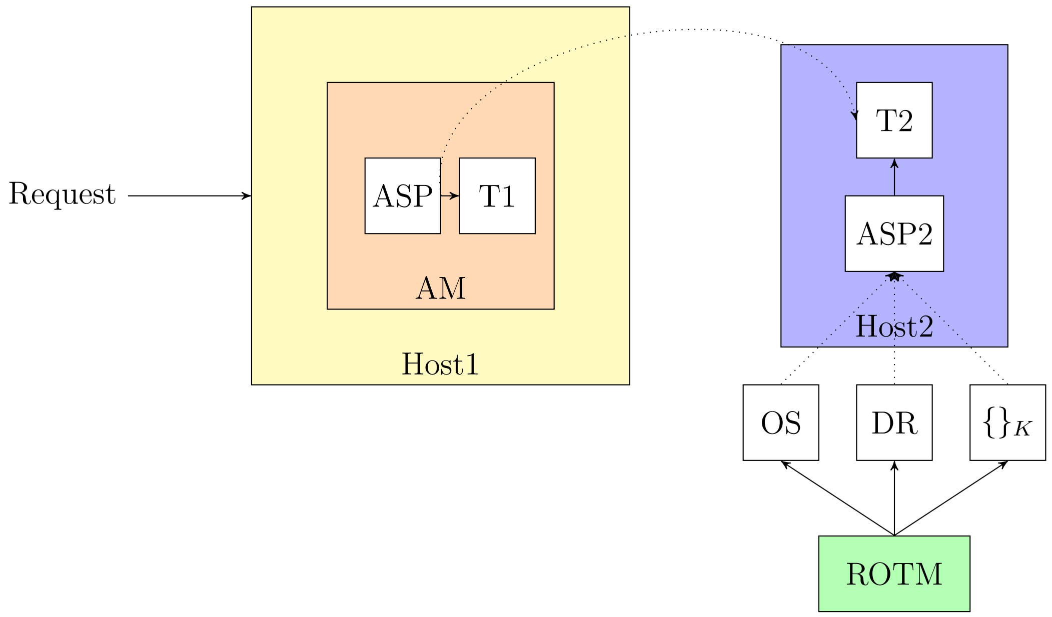

Figure 3: Total system.

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?

4 Protocols as Transition System

Perry's 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 [5] below:

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

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 [4, 3].

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.

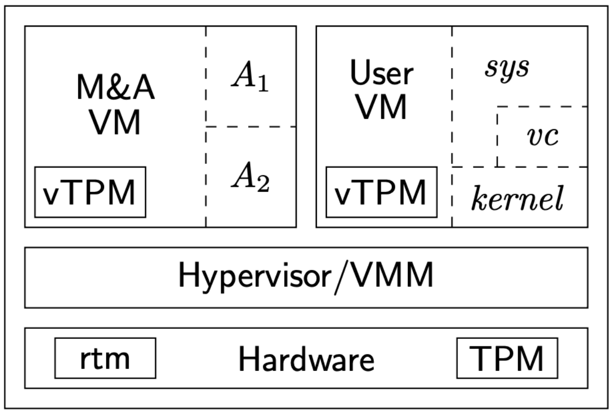

Figure 4: An example attestation system defined by [4].

Realizing this system, we can derive a bottom up measurement scheme which, taken from [4], 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 rtm.

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

$$\begin{tikzcd}
\mathsf{ms}(\mathsf{rtm}, A_1) \arrow[dr] & \mathsf{att\text{-}start}(n) \arrow[d] & \mathsf{ms}(\mathsf{rtm}, A_2) \arrow[dl] \\
& \bullet \arrow[dl] \arrow[dr] & \\
\mathsf{ms}(A_1, vc) \arrow[dr] & & \mathsf{ms}(A_2, ker) \arrow[dl] \\
& \mathsf{ms}_{ker}(vc, sys) &
\end{tikzcd}$$

Figure 5: An example of measurement ordering from the previous system as defined by [4].

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** event. THE TRANSITION RELATION THEN IS THE MEASUREMENT RELATION.

Now, back to the example system presented in Figure 2. We can represent this in a dependency like structure below similar to teh structure used by [4].

We can then easily see how to create a transition system using the measures relation.

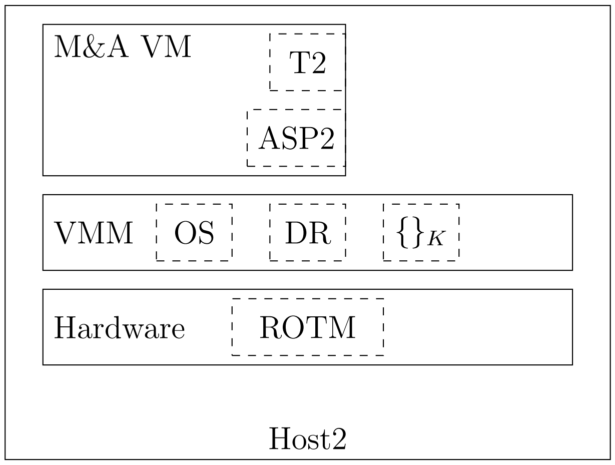

Figure 6: Host2 presented as one system.

References

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

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

[3] 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.

[4] 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.

[5] 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.