Context Relation Musings

Anna Fritz

February 8, 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.

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

Definition 3 (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?*

Definition 4 (Better). *One protocol is* better *than another when...*

2 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:

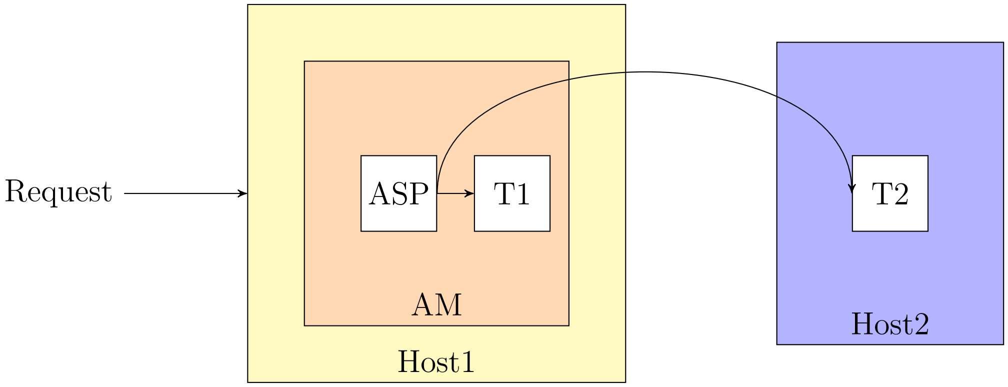

Figure 1: Example Target System.

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

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 → represents sequenced measurement and ~ 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 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

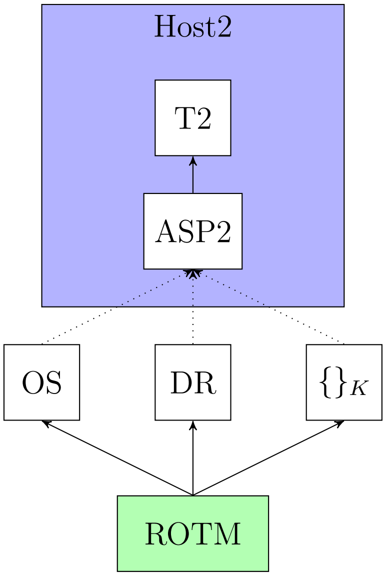

Figure 2: Example Host2 System.

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?

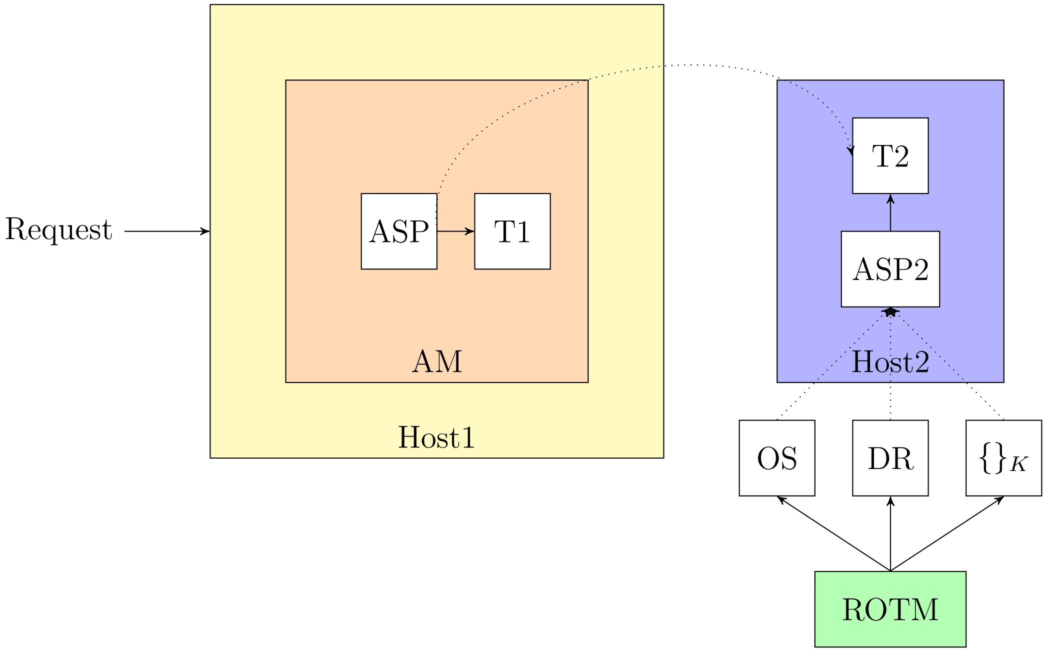

Figure 3: Total system.

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 ≤ 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?