Musings on Gathering and Bundling Evidence

Perry Alexander
palexand@ku.edu

March 26, 2015

Abstract

This document captures discussions on evidence bundling in semantic remote attestation. Evidence Bundling is described as aggregation of evidence from multiple sources into a package whose overall trustworthiness can be appraised. This necessarily includes primary evidence from the system being appraised and meta-evidence describing the evidence gathering process. This is a living document and will be updated frequently.

Notation

Throughout we use a trivial monadic notation for protocols that serves as our "assembly language" target for protocol compilation. The following conventions hold:

```
do {                % evaluate functions in sequence
     f(x);          % calculate f(x) and discard the result
     y <- f(x);      % calculate f(x) and bind the result to y
     sndXXX(x,a);   % operation on x that sends something to a
     y <- rcvXXX(a) % an operation that receives something from a
                    %  and binds it to y
}
```

This is early work, so we play fast and loose with specific syntax and semantics.

Bundling

A working definition of bundling from ongoing discussions:

> "If an attestation [protocol] is negotiated and an appraiser receives evidence that some set of components is uncompromised, then ensure the appraiser can verify that those uncompromised components executed their part of the plan" – Paul Rowe and Joshua Guttman

Said differently, bundling is the assembly of evidence from multiple sources in a way that enables an appraiser to determine that each element was obtained by the correct measurement component at the correct time.

Following we will outline working examples of when and how bundling occurs. We will start by defining a simple negotiation among an appraiser and attestation manager. Then we will define several potential results of that negotiation and discuss how those examples reflect bundling issues.

Negotiation

Consider the problem of determining the status of virus checking on a remote system. This might occur when a new computer is asking to join a controlled network such as a university wireless system. The appraiser representing the university wireless system wants to determine the status of an attestation manager representing the new computing system.

On the appraiser side a trivial protocol runs sending a request (`r`) to a target (`t`) and in response receiving a protocol (`Some p`) or refusal to participate (`None`):

```
do { sndRequest(r,t);
     q <- rcvProt(t)
     e <- case q of
            Some p : sndRemote(p,t)
            None : None
          end;
     case e of
       Some v : appraise(v)
       None : None
     end
}
```

If the request returns a protocol (`p`), the appraiser sends the protocol to execute on the target. Evidence (`e`) is received from the target and appraised if it is valid evidence.

On the attestation manager, a protocol runs that receives a request from the appraiser (`a`), evaluates its privacy policy (`priv`) with respect to the request and the appraiser's ID. If its privacy policy is respected and the appraiser trusted, it returns a single attestation protocol (`p0`). Otherwise, it refuses to participate (`None`):


```
do { (r,a) <- rcvRequest;
     p <- if (priv r a) then (Some p0) else None;
     sndProt(p,a);
     (q,a) <- rcvRemote;
     e <- if p=q then execute(q) else None;
     sndEvidence(e,a)
}
```

There is no negotiation, just a simple determination if the request satisfies the target's privacy policy. If so, the only protocol that can be returned and executed is `p0`.

The interface between this negotiation protocol and the attestation protocol resulting from it is the `execute` function that causes the attestation protocol to run on the attestation manager.

Attestation

To consider bundling, we look at several examples of an attestation protocols (`p0`) for determining the status of a target's virus checking software.

First is a protocol that assesses several properties and runs locally on the attestation manager's system:

```
do { id <- getVCID;
     sig <- getSigFileEvidence;
     src <- getSigFileSrc;
     e <- createEvidence(id,sig,src);
     returnEvidence(e)
}
```

This protocol gets the checker ID (`getVCID`), checks the signature file (`getSigFileEvidence`), and checks the source of the signature file (`getSigFileSrc`). Then it bundles all evidence into a single evidence package (`createEvidence`) and returns it (`returnEvidence`) to be sent back to the appraiser by the negotiation protocol. The appraiser then consumes the resulting evidence to perform appraisal (`appraise`).

The evidence returned has approximately the following form:

$$\langle (id, sig, src), \{||(id, sig, src)|, PCRComp_0|\}_{AIK_0^-} \rangle$$

where (id, sig, src) is primary evidence and hashes and signatures are meta-evidence. An appraiser can check: (i) primary evidence to assess the measured

virus checking subsystem; and (ii) the signature on the quote to determine its authenticity and *PCRComp* to assess the platform construction.

This is a simple example of evidence bundling—combining primary evidence from three ASPs and meta-evidence from the appraisal environment in the same package. In this case, primary evidence gives us information about the running system and meta-evidence gives us information about how the evidence was gathered. The appraiser first needs to determine if the quote and *PCRComp* are sufficient meta-evidence for it to trust its primary evidence and then determine if (id, sig, src) is sufficient to trust that the remote system is running an appropriately configured virus checker.

Another protocol represents simply providing the appraiser evidence that a virus checker is running:

```
do { id <- getVCID;
     e <- createEvidence(id);
     returnEvidence(e)
}
```

This simpler protocol generates less evidence than the first, but may be more acceptable to the target system. The appraiser could determine this is insufficient or limit access to resources and services based on its appraisal.

Note: This example will expand into a more involved negotiation of a protocol.

A more interesting case for the attestation protocol happens in the presence of complex bundling:

```
do { id <- getVCID;
     sig <- getSigFileEvidence;
     src <- getSigFileSrc;
     srcEvidence <- appraiseSrc(src);
     e <- createEvidence(id,sig,src,srcEvidence)
     returnEvidence(e)
}
```

In this case the `appraiseSrc` function does a full appraisal of the signature file server, `src`. The function communicates its need for evidence from the signature file source to the appraiser associated with the attestation manager. That attestation manager then executes a similar negotiation protocol as the one that started the appraisal process. The resulting evidence, `srcEvidence`, has the form:

$$\langle (e), \{||e|, PCRComp_1|\}_{AIK_1^-} \rangle$$

is signed by the signature file server, not the original appraisal target with its own AIK. Thus, the information from the encapsulated evidence is bundled in the outer evidence:

$$b = \langle (e), \{||e|, PCRComp_1|\}_{AIK_1^-} \rangle$$
$$\langle (id, sig, src, b), \{||(id, sig, src, b)|, PCRComp_0|\}_{AIK_0^-} \rangle$$

The appraiser must somehow interpret the evidence from the signature file server without *a priori* knowledge of its identity.

This more general version of the bundling problem requires:

- Strong identification of the signature file server to the appraiser
- Nested evaluation of the privacy policy for the "other" attestation manager before executing the entire attestation protocol.

An alternative implementation would have the attestation manager return the identity of the signature server and allow the appraiser to negotiate separately with the signature server. This could be called *flattening* the protocol if there is a desire to name it.

```
do { sndRequest(r0,t0);
     q0 <- rcvProt(t0);
     e0 <- case q0 of
             Some p : sndRemote(p,t0)
             None : None
           end;
     r1 <- appraise(e0);

     sndRequest(r1,t1);
     q1 <- rcvProt(t);
     e1 <- case q1 of
             Some p : sndRemote(p,t1)
             None : None
           end;
     appraise(e1)
}
```

The evidence produced from both requests from the appraiser has the following form:

$$\langle (e), \{||e|, PCRComp_1|\}_{AIK_1^-} \rangle$$
$$\langle (id, sig, src), \{||(id, sig, src)|, PCRComp_0|\}_{AIK_0^-} \rangle$$

Note that this is quite similar to the evidence produced by the earlier, unflattened attestation. We have the same PCR composites and evidence signed by the same AIK instances. The distinction is the evidence packages are not nested. Semantically, what is the distinction between these two results? Is one preferable to another from an evidence perspective?

Is this flattening operation—eliminating hierarchy in the attestation protocol—a general operation or something that applies only here? If it is not general, can we live with its limitations? Can we assert any correctness properties for it? Could there be an interesting man-in-the-middle attack where the outer attestation manager could negotiate in bad faith and return instructions to the appraiser that produce a bad result?

Privacy Policies

Each protocol executed in the virus checking example is evaluated by a privacy policy. This policy governs access to information available from the attestation target.

Formally, let T_i be a security type that may be associated with appraisers, attestation managers, and evidence. Denote p as belonging to type T_i by $p : T_i$. Given an appraiser, $a : T_a$, making a request and target, $t : T_t$, with measurable evidence $e : T_e$, a is authorized to access e if the t's privacy policy allows domains of type T_a access to evidence of type T_e. A privacy policy for target type T_t is expressed as a relation, PRIV:

$$\mathsf{PRIV}_{T_t}(T_a, T_e)$$

In the protocols above, `priv` corresponds to PRIV and checking privacy policy `(priv r a)` is determining the truth value of $\mathsf{PRIV}_{T_t}(T_a, T_e)$. This semantics is derived from SELinux policy developed by Hicks et al. [2007].

Assertion 1 (PRIV is not transitive) $\mathsf{PRIV}_{T_t}(T_a, T_e) \wedge \mathsf{PRIV}_{T_a}(T_b, T_e)$ *does not imply* $\mathsf{PRIV}_{T_t}(T_b, T_e)$. *Informally, if t trusts a with evidence of type T_e and a trusts b with evidence of type T_e, it does necessarily follow that t trusts b with evidence of type T_e.*

Assertion 2 (PRIV is not symmetric) $\mathsf{PRIV}_{T_t}(T_a, T_e)$ *does not imply that* $\mathsf{PRIV}_{T_a}(T_t, T_e)$. *In formally, if t trusts a with evidence of type T_e, it does not necessarily follow that a trusts t with evidence of type T_e.*

Assertion 3 (PRIV is reflexive) $\forall t, e \cdot \mathsf{PRIV}_{T_t}(T_t, T_e)$. *This is not a particularly useful property in that it only states T_t trusts itself with its own evidence.*

Glossary

Primary Evidence Evidence describing the measured application.

Meta-Evidence Evidence describing properties of other evidence.

Negotiation Protocol Sequence of events used to determine what protocol(s) to run and when to run it/them.

Attestation Protocol Sequence of events used to gather and prepare evidence by invoking attestation service providers.

Attestation Protocol Block (APB) See Attestation Protocol Instance (API).

Notes

- relative order of measurements (temporal ordering)
- order measures are constructed relative to request (freshness)
- trust status of keys
- relevant FLASK policies that restrict communications (platform MAC)
- vantage point from which evidence was created

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.

B. Hicks, S. Rueda, L. St.Clair, T. Jaeger, and P. McDaniel. A logical specification and analysis for selinux mls policy. In *Proceedings of the 12th ACM symposium on Access control models and technologies*, SACMAT '07, pages 91–100, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-745-2. doi: 10.1145/1266840.1266854. URL `http://doi.acm.org/10.1145/1266840.1266854`.

P. A. Loscocco, S. D. Smalley, P. A. Muckelbauer, R. C. Taylor, S. J. Turner, and J. F. Farrell. The inevitability of failure: The flawed assumption of security in modern computing environments. In *In Proceedings of the 21st National Information Systems Security Conference*, pages 303–314, 1998.