Probabilistic Semantic Noninterference

February 25, 2018

1 Parties and Schedulers

Assume an expression language with values in Val. A *state* St is a map $\mathsf{Var} \to \mathsf{Val}$, where Var is a set of variable names. A *buffer* is a value of type $\mathsf{list}(\mathsf{PID} \times \mathsf{Val})$, where PID is a set of party names. Then, a *handler* P is a semantic function of type $\mathsf{St} \to \mathsf{InBuf} \to \mathcal{D}(\mathsf{St} \times \mathsf{OutBuf})$, where $\mathcal{D}$ is the finitely supported distribution monad. It is the intention that InBuf only contains the set of unprocessed input messages, and OutBuf only contains the set of output messages produced during the current invocation of P. It is also the intention that variables in St should not be overwritten, so that the state grows monotonically.

(Later, P can be given monadically, in which we may statically track what variables are being accessed and so on.)

A *scheduler* is defined by the following syntax:

$$c := \mathsf{Run}\ k\ @\ i;\ c \mid \mathsf{stop},$$

where $k \in \mathsf{Handler}$ is a handler name, and i is a PID.

2 Semantics

A *trace* τ is a value of type $(\mathsf{PID} \to \mathsf{St}) \times \mathsf{NewEv} \times \mathsf{OldEv}$, where NewEv and OldEv are logs of the form $\mathsf{list}(\mathsf{PID} \times \mathsf{PID} \times \mathsf{Val})$. The first component is the global state, the second component is ordered buffer of unprocessed messages, and the third component is the ordered buffer of processed messages. Given a buffer B, define $B_{|i}$ to be the pairs in B such that the second component is equal to i (preserving order).

Then, define our scheduler semantics $[\![c]\!] : \tau \to (\mathsf{Handler} \to P) \to \mathcal{D}(\tau)$ by

$$[\![\mathsf{Run}\ k\ @\ i; c]\!](G, B_u, B_p)\mathcal{I} := \mathsf{bind}\ ((\mathcal{I}k)(Gi)B_{u_{|i}})\ (\lambda s' o.[\![c]\!]\ (G[i := s'], o \mathbin{||} (B_u \backslash B_{u_{|i}}), B_{u_{|i}} \mathbin{||} B_p)\ \mathcal{I})$$

and

$$[\![\mathsf{stop}]\!]\ \tau\ \mathcal{I} := \mathbf{1}_\tau,$$

where bind is the monadic bind operation for distributions and $||$ is list concatenation.

Above, $\mathcal{I}$ is an interpretation function from handler names to handlers.

(Corruption and message interference may be modeled by differing semantics.)

3 Noninterference

A *leakage* (or *declassification*) property φ is a function $\mathsf{PID} \to (\mathsf{PID} \to \mathsf{St}) \to \mathsf{Val}$. Given an initial global state G, $\varphi i G$ denotes the information i should be able to learn from G after the execution of the protocol.

Given two distributions D on traces, write $D \equiv_i D'$ if the marginals $\mathcal{D}(\lambda\, G\, B_u\, B_p.\, (G\, i, B_{p_{|i}}))$ are identical for both D and D'. That is, $D \equiv_i D'$ if from party i's position, D contains exactly the same information as D' on both states and processed messages.

Then, say that the pair $(c, \mathcal{I})$ is φ*-noninterferent* if for all G, G', i,

$$(G\ i) = (G'\ i) \wedge \varphi\ i\ G = \varphi\ i\ G' \implies [\![c]\!]\ (G, \emptyset, \emptyset)\ \mathcal{I} \equiv_i [\![c]\!]\ (G', \emptyset, \emptyset)\ \mathcal{I}.$$

That is, $(c, \mathcal{I})$ is φ-noninterferent exactly when, for every party i, if two initial global states look identical to i and agree on values of ϕ, then their induced traces will appear identical to i.

Note that in the above definition, equivalence of traces is sensitive to order of message delivery – but is only sensitive to message ordering from the perspective of individual parties. That is, two send commands in a handler may be safely reordered if they are sent to different recipients.

4 Examples

In *multiparty computation*, each party is given an input x_i, and a protocol is devised so that each party receives the value $f(\vec{x})$, but no further information is shared. This is modeled by the leakage function $\phi\ i\ G := f((G\,1).in, \ldots, (G\,n).in)$.

Functions may also be asymmetric, in which the leakage function is $\phi\ i\ G := f_i((G\ 1).in, \ldots, (G\ n).in)$. A canonical example is *oblivious transfer*, where the sender has two messages m_0 and m_1, and the receiver has a bit b. The sender should learn nothing (i.e., $f_S(m_0, m_1, b) = ()$), while the receiver should learn the bth message (i.e., $f_R(m_0, m_1, b) :=$ if b then m_0 else m_1.)

5 Approximate reasoning and corruption

If we require that $D \approx_i D'$ instead of $D \equiv_i D'$, where we may bound the computational distance between distributions, then we may obtain a computational semantics for protocols. Here, $\approx_i$ means that the distributions in quesion are bounded by a negligible function of η in distance. Crucially, $\approx_i$ is from *the* *point of view of party i*: i.e., we may transform a DDH triple (g^a, g^b, g^{ab}) into (g^a, g^b, g^c) only relative to a specific party.

Given semantics for corrupt adversaries, a *corruption model* $\mathcal{A}$ may be defined to be a set of rewrite rules on schedulers c. This may induce a corruption semantics $\mathcal{A}, c \vdash c'$, where c is an initial, uncorrupted schedule and c' is a rewriting of c according to $\mathcal{A}$. Then, we may change the above main definition to universally quantify over all c' such that $\mathcal{A}, c \vdash c'$.