Accountability in the Decentralised-Adversary Setting

Robert Künnemann, Deepak Garg, Michael Backes
Decentralised Adversary

Centralised Adversary
What we talk about when we talk about accountability
Who Keeps Whom Accountable?

Protocol defines "normal" behaviour

Agents may break $\varphi$

informs

hold accountable (for $\varphi$)

verdict = sets of parties blamed
Who Keeps Whom Accountable?

- everybody who steps out of line?
- Requires complete communication. It's the Internet, duh!
- Benign mistakes happen. Moral problem, but also: bad implementations
- all causes (causing parties) of $\neg \varphi$

verdict = sets of parties blamed

CORRECT!
From causation to accountability

A colleague asked for help

Bike broke down

C could not take the bus
From causation to accountability

A ran deviating program A'

B ran deviating program B'

C ran deviating program C'

Loss of authenticity
It works (in the centralised setting)
We can analyse that stuff (in the centralised model)

Protocol spec + accountability lemmas

(tammar-prover)

attack / verification / timeout

- Certificate Transparency
- OCSP Stapling
- MixNets
- Alethea/MixVote
- Accountable Algorithms
Limits of the centralised-adversary setting
The centralised adversary
Provocation - scenario 1

A

\texttt{out}()
There is one cause, \{A, C\}.

Anyone can derive "go!"

Indistinguishable from A if

- private communication possible
- or code of A not known

Not a modeling artefact
- similar problems with causation in general (The Gardener & the Queen of England)
- causation considers different "worlds" and some are more plausible
- ordering of worlds
- "under constrained" (e.g. radical Gardener could despise all inedible flora)
Optimality

- pick smallest possible verdict:
  - logical entailment when verdict interpreted as DNF
    \[ \{\{A, B\}, \{C\}\} = A \land B \lor C \]
  - \\{\{A\}\} < \\{\{A, C\}\} \text{ because } A \land C \implies A

- pick knowledge-optimal explanations, i.e., code for deviating parties
  - if A has knowledge to produce \(\mathbb{A}\), scenario 1 is knowledge-optimal

- pick simple explanations
  - includes knowledge-optimal
  - code cannot have conditionals (because we cannot see their effect)
Accountability

Full

Simple

**fairness:** all blamed parties cause violation

**completeness:** all parties causing violation are blamed

"the real deal"

full accountability

all communication must be visible
Accountability

Full

Simple

weak fairness: all blamed parties deviated

weak completeness: one party of each joint cause is blamed

"try to be specific!"

verdict-optimal accountability

verdicts with intersections (e.g., \{A,B\}, \{B,C\}) impossible

"try to be specific!"
Accountability

- Weak fairness: all blamed parties deviated
- Weak completeness: one party of each joint cause is blamed
- "assume minimal information sharing"
- Knowledge-optimal accountability

"either verdicts always non-intersecting or no indirect communication"
Accountability

Full

- weak fairness: all blamed parties deviated
- weak completeness: one party of each joint cause is blamed

Simple

"minimal information sharing + no conditionals"

simple accountability

= accountability in the centralised-adversary setting
Conclusion
Conclusion

- Accountability is identifying misbehaving parties
- "misbehaving party" = "party whose deviation caused ¬φ"
- the centralised setting is not w.l.o.g.:
  - silent assumptions: optimal information sharing and linear programs
  - guaranteed: weak fairness (party that is blamed deviated)
  - not guaranteed: weak completeness (catch member of each cause)
- verdict-optimality:
  - provides weak completeness
  - applicable for tasks like access control, randomness generation or holding a third party accountable
- all separating examples rely on signalling behavior unrelated to protocol
  - maybe optimality principle is adequate (Occam's razor, optimality & defaults in causation)
  - at least we know what we are doing now
Thank you!