Evidence for a *posteriori* security

Alexander Hicks, Steven J. Murdoch

University College London
Evidence?

Liability  Accountability  Law

Due process  Auditability

Verifiability  Integrity  Proof

Non-repudiation  \( P(A|E) > P(A) \)
Some background

- Ideal security goal: build secure systems that won’t fail in any way, control information flow in systems.

- Realistically, systems have many points of failure:
  - Protocol & human failures
  - Flawed rules

- Behaviour of the system cannot always be restricted to fit security goals.
Murphy’s law/First axiom of infosec: anything that can go wrong will go wrong

Impact of a failure varies from party to party
- “Fail safe”: party not responsible for a failure should not bear the costs
- “Fail deadly”: party responsible for a failure should bear the costs

Accountability adds an incentive to avoid failures, but it requires an enforcing mechanism: evidence
Some background

- Assume you have to deal with parties that can do anything e.g., law enforcement, kids

- Assume a trusted third party or a consensus mechanism that can make a decision and enforce it, it requires something to make the decision

- For our purpose, evidence is anything that can be used to make that decision i.e., assign liability

- Rather than help avoid failures, it helps deal with them after the fact
  - Security: identify the issue that caused a failure
  - Privacy: identify a privacy violation (typically policy)
Some examples

- Payment systems

- Access to data
  - VAMS: Verifiable Auditing of Access to Confidential Data (2018)
  - Ian Levy (NCSC Technical Director), on the privacy debate of data interception and surveillance: “My call is for more transparency, more openness and more evidence in this debate” ([https://youtu.be/LRiAcbvSA3A?t=1h11m46s](https://youtu.be/LRiAcbvSA3A?t=1h11m46s))

- Cryptocurrencies

- Many more, pick one related to your interests
Questions

- What’s the correct model for the use of evidence?
- How should evidence be produced?
- What form should evidence take?
Threat model

- Threat modelling: want to account for parties that have “freedom”
- Compare to the “honest” ideal of what the party is doing, as specified by the system rules
- Deviation is realistic, but evidence allows evaluating the deviation and deciding if something was done wrong
- Goal: If someone deviates, there should be evidence so that they can be held accountable
Models

- Games are a standard way of modelling problems
  - Ideally: equilibrium that restricts deviations by ensuring detection

- Our setting has a few special attributes that aren’t usually discussed:
  - Strategies have costs
    - What’s the cost of deviating against the cost of auditing?
  - If the system is open, all players aren’t necessarily known
    - Distribution over player types and computational capabilities?
  - Evidence adds new information to the system as it evolves

- Possible approach: Bayesian machine games (Halpern and Pass)
  - Bayesian game (incomplete information)
  - Takes into account machine types and a complexity function
Evidence principles

- **System independence**: a system failure should not lead to failure of evidence production

- **Reliability**: Evidence should contain all information required to make a decision, but no more
  - Requires a clear definition of decision mechanism
  - Not possible to make liable an innocent party, no deniability for a guilty party

- **Robustness**: Evidence should be tamper-resistant

- **Retention**: Evidence collection and retention should not depend on a single party
Evidence production

- Logging actions
  - Merkle trees (Blockchains, Trillian)
  - Tamper-evident way of keeping records of actions
  - Is there a way of enforcing automatic logging of actions?

- Verifying computations
  - Existing cryptographic tools for verifying execution of programs
  - Generalising to things that aren’t programs?

- Proofs-of-X
  - Task specific proofs only

- Systems aspects: embedding the evidence production on top of a system
Presenting evidence

- Three settings for presentability
- Production level
  - Presented internally, no need for public presentability
- Expert witness:
  - Presented publicly to an expert witness, limited need for presentability but need for explainability
- Non-expert:
  - Presented to the public, convincing levels of presentability and explainability needed
Forms of evidence

- Many trade-offs that have to be considered in context
  - Presentability
  - Verifiability
  - Privacy

- Evidence design should happen alongside system design
Questions

● What’s the correct model for the use of evidence?
  ○ Unknown, varying participants
  ○ Computational costs
  ○ Evolving system

● How should evidence be produced?
  ○ Cryptographic evidence
  ○ Non-cryptographic evidence

● What form should evidence take?
  ○ Presentability
  ○ Privacy
  ○ Verifiability
Questions?

alexander.hicks@ucl.ac.uk

@alexanderlhicks