A Game-Theoretic Framework for Analyzing Trust-Inference Protocols

Bobby Bhattacharjee
Jonathan Katz
Ruggero Morselli

University of Maryland

(Work supported by NSF grant CNS-0310499)

Overview of this talk

- Trust inference
- A new framework for analyzing trust inference protocols
  - Why a new framework is needed
  - Some "guiding principles"
  - Details…
- Feasibility results (preliminary)

Note…

- This is work in progress…
- Comments, questions, and discussion appreciated!

Basic setting

- We consider resource-sharing networks, where our goal is to enforce cooperation
  - I.e., to prevent/limit "free-riding"
  - More specifically, to provide incentives for users to share resources freely

Basic assumptions

- We focus on completely decentralized networks of true peers
  - No external trusted third parties
  - No entry cost, pseudonymity
  - No pre-provisioned trusted parties
  - No global system history
- This is of practical and theoretical interest

Basic assumptions

- *All* users in the network are assumed to be rational
  - No "altruistic" users
  - No "purely malicious" users
- (Note: the first assumption may be overly pessimistic)

Why trust inference?

- Users can always base their actions on their own personal history
  - E.g., if *A* previously cooperated with me, I will now cooperate with *A*
- Numerous drawbacks to this approach
  - "Inefficient"
  - Repeated interactions with same peer may be infrequent
  - System boot-up --- who goes first?
  - Integration of new users

Trust inference

- Idea: information about parties' past behavior can be "propagated" through the network
  - Decisions about future actions no longer based on personal history alone!
- Many trust inference protocols have been developed and analyzed…

But…

- Which trust inference protocol to use?
  - Is any one "better" than the others?
- How do we know that any of the known protocols are "good"?
  - What do we even mean by "good"?
- Can we rigorously prove anything about these protocols (in realistic settings)?

For comparison

- Can design cryptographic protocols (signature schemes, etc.) with ad-hoc security analysis
  - These typically wind up being broken
- Better to design protocols which have rigorous proofs of security
  - Much better assurance in this case!
  - Even developing the "right" definition is useful

Limitations of prior work

- Current protocols rarely have proofs of security (or, "goodness")
  - Even a definition of "goodness" is not usually given
  - Simulations are no substitute for proofs
- Some work makes "centralized"-type assumptions
  - Global knowledge about history (e.g., [FR, BAS])
  - Pre-provisioned trusted nodes [eigentrust]
  - "E-bay" model

Limitations of prior work

- Some work restricts malicious behavior to sharing/not sharing only
  - Assumes that "trust propagation" phase is honestly executed, and/or that users honestly report the actions of others
  - Some work focuses on "keeping users honest", but not clear if it succeeds…

Why a new framework?

- Need a way to compare existing protocols
  - Different protocols may be appropriate for different adversarial/network environments
- A rigorous framework forces us to define the desired properties of a protocol
  - Can consider various adversarial models
- A formal framework potentially enables proofs of "goodness"/security

We hope our work is a first step in this direction --- it is certainly not the last

Some design principles

- Use game theory to analyze protocols
  - The actions prescribed by a "good" protocol should form an equilibrium
- Corollary: it is not enough for a trust inference protocol to compute trust values --- it must also prescribe actions based on these values

Some design principles

- Equilibrium should hold at all times
  - Including (especially) at "boot up"
  - Also for new users
- The trust propagation phase itself should form an equilibrium
  - Dishonest users can "cheat" at any point during the protocol, not just during sharing phase
  - No assumption of shared history; must take into account false accusations and coalitions
  - Similar "flavor" to Byzantine agreement

Basic framework

- All users have pseudonyms which are:
  - Distinct
  - Easily-generated
  - Impossible to impersonate
- We identify these with public keys for a secure digital signature scheme
  - No PKI or central registration authority!
- Actions associated with pseudonyms

Adversarial model

- We give the adversary *A* complete control of the network, via oracles:
  - **NewUser** – creates new (honest) user; *A* learns its pseudonym
  - **HonestPlay**(*i*, *j*) – honest users *i* and *j* play an instance of a 2-player game
  - **Play**(*i*, *id*, action) – *A* plays "action" against honest user *i*, using pseudonym *id* (*id* cannot be held by any honest party)
  - **Send**(*i*, *id*, msg) – sends message "msg" to honest user *i* from pseudonym *id*

Other details

- The trust inference protocol is run among honest users "in the background"
  - Messages sent from one honest party to another are not under *A*'s control
- The 2-player games played can be different, or even selected by *A*
  - For simplicity, we model them as the same instance of a prisoners' dilemma game

Defining utility I

- We incorporate a notion of time, and also a discount factor
  - Oracle calls associated with a particular time (chosen by *A*)
  - Trust inference protocol run (in the background) when *A* increments the time
  - (May limit # of calls --- e.g., **NewUser** calls --- *A* makes in one time unit)

Defining utility II

- *A*'s utility increases after each **Play** oracle call
  - Depending on payoff matrix and the actions chosen by A and its partner
  - Discounted based on time of oracle call

Defining robustness

- A trust-inference protocol is robust if the adversary maximizes its utility by following the protocol
  - I.e., the actions of the protocol form an equilibrium for all users
- Note: the model already incorporates both coalitions and Sybil attacks

Other desiderata

- Robustness alone is not enough! Also need to examine:
  - Expected utility of the protocol
  - Resilience to trembles
  - Incentive for new users to join
  - Efficiency considerations

Advantages of the framework

- Enables proofs of robustness, and objective comparisons of existing trust inference protocols
- Assumes no centralized components
  - But can be augmented, if desired
- Very flexible
  - Handles wide range of adversarial behavior

Remarks…

- The framework assumes a very powerful adversary
  - A robust protocol in this model will certainly be robust in the real world
  - Unclear how else to model real systems
- Impossibility results would be great!
- Can also consider relaxing the model

Variations

- Do not let *A* control network membership
  - Disallow **NewUser** queries; have users join over time instead
- Do not allow *A* to control trading patterns of honest parties
  - Disallow **HonestPlay** queries; have users trade randomly, synchronously, etc.
- No coalitions/Sybil attacks
  - Allow only one **Play** query per time period

Feasibility results I

- We show that robust solutions exist…
  - …but we do not yet know any practical (and provably-robust) protocols
- "Grim trigger" strategy
  - Robust; optimal expected utility in strongest adversarial model
  - Not resilient to trembles
  - Not a subgame-perfect equilibrium

Feasibility results II

- A variant of "pay-your-dues" [FR] is provably robust when synchronous, random trading is assumed
  - No trusted party (as in [FR])
  - Users "broadcast" the result of their interactions
  - Note: users may broadcast false or conflicting information

Concluding remarks

- Better formal models for trust inference are sorely needed
  - Our work provides a starting point
- Open questions:
  - Extend PYD to stronger settings
  - Show that our model is too strong (impossibility results)
  - Show that efficient and robust trust inference is possible within our model