A Game-Theoretic Framework for Analyzing Trust-Inference Protocols

Morselli, Katz, Bhattacharjee

baptiste.alcalde@uni.lu

Introduction

Why is trust necessary in P2P?

- Cooperation is necessary
- Simple punish/incentive scheme using own interactions is problematic (rare direct interaction $=$ low chance of redeem, the first time problem...)
- Use the other agents' interactions (propagated in the system) $\rightarrow$ reputation/recommendation system

Claimed contribution

The provided definition

- Enables proofs
- Enables comparisons
- Is appropriate for decentralized systems
- Enables the use of a wide range of adversarial behavior

Pseudonyms are

- Distinct (unique)
- Easy to create by the users themself (no trusted party)
- Impossible to impersonate by others

Protocol Π prescribes

- how trust should be infered
- how a user's actions should depend upon the inferred value

The Adversarial Framework

Adversary A's oracles ($=$actions)

- **NewUser** creates a new honest user and A learns it
- **HonestPlay(i,j)** 2-players game according to the protocol Π between i and j
- **Play(i, id, action)** 2-players game between A (id) and i (honest player)
- **Send(i, id, msg)** A sends a message msg to i
- $+$ A can see any message between honest users

A 2-players game

Prisoners' Dilemma

	C	D
C	(1, 1)	(-1, 2)
D	(2, -1)	(0, 0)

Rational adversaries are assumed

Adversary's utility increases after each *Play* by $\delta^t\mu$ where μ is the payoff (cf. table) and $\delta < 1$ is a discount factor

Network model

Broadcast Network reliable

Complete P2P Network trusted infrastructure (?)

- Every user learns the arrival of a new user
- Any user can send messages to others using the infrastructure
- **NotifyJoin(i,j)** additional A's oracle

Timing Model - part1/2

In a time period t

- A makes at most N *NewUser* calls
- A makes at most N' *Play* calls

The value t always increases and each time period is divided into *play phase* and *protocol phase*

- *play phase*: A can issue *NewUser*, *Play*, and *HonestPlay*
- *play phase* ends at first *Send* or *Activate* call (stamped with t)
- *protocol phase*: *Send*, *Activate*, *Done*, and messages between honest users are exchanged
- *protocol phase* ends when A makes a call stamped with $t + 1$

But A can label a call with $t+1$ only if

- in *protocol phase* of t
- the last n calls where *Activate* answered with *Done* (n is the number of current honest users)

In addition, A cannot issue a *Play* on a honest user created in the current period

Robustness

Definition 1: "Π is robust if A maximizes its utility by following Π, i.e. if the actions prescribed by Π form a subgame-perfect equilibrium"

Other notions:

- **Expected utility**: utility when everyone is honest
- **Resilience to trembles**: "honest" defects (network fault . . .)
- **Efficiency at admitting newcomers**: not too severe penalty
- **Efficiency**: number of messages . . .

Π_1 - Grim Trigger

- A player that has never received a grim trigger message always cooperate
- If players i and j interact and i defects, then j sends a grim trigger message to everyone (and himself) in the following protocol phase
- A player that has received a grim trigger message will always defect and will send grim trigger messages to everyone at every subsequent time period

Lemma 1: “The grim trigger strategy is robust if the future (?) discount factor δ is at least $\frac{1}{2}$, and it achieves optimal expected utility when the probability of trembles is 0, in the strongest adversarial model considered here”

Π_1 - Proof

- Let adversary G be compliant with the protocol (no defect)
- G creates N honest users at t_0
- $t > 0$, G fair-*Play*s with each honest user
- The utility of G is : $u_G = \sum_{t=1}^{\infty} N\delta^t = \dfrac{N\delta}{1-\delta}$
- If A defects at time t', its utility is:

$$u_A = \sum_{t=1}^{t'-1} N\delta^t + 2N\delta^{t'} = \frac{N\delta(1-\delta^{t'})}{1-\delta} + N\delta^{t'}$$

- Then $u_A > u_G$ iff $\delta < \frac{1}{2}$