Some random thoughts and some potentially relevant ideas from AI

Stuart Russell
Computer Science Division
UC Berkeley

Random thoughts

- Encourage use of formal methods:
  - Guarantees -> liability -> insurance -> proof
  - Develop software ecosystem with few, composable, secure elements wrapping application-specific code and limiting uncontrolled interaction to minimum necessary to achieve functionality: *start simple* (cf salesforce.com)
  - Improve education (problem partly cultural)
- Support clean-slate redesign of the internet
  - (Why wouldn't companies and individuals sign up to use a more secure/accountable version??)
- Can useful secure computation occur when *everything* is measurable by adversary?

Cyberhuman systems

- Cf. "cyberphysical systems" - systems composed on computational and human elements
- Can we design cyberhuman systems with provable desired properties?
  - Cf. economics, political science (humans as rational or empirically designed agents)
  - Cf. HCI (humans as procedural or statistically estimated models)

Cyberhuman systems contd.

- Obvious problem for security: adversarial (worst-case) behavior
- Example: automated driving in control theory: game-theoretic approach with worst-case analysis of other vehicles

Cyberhuman systems contd.

- Obvious problem for security: adversarial (worst-case) behavior
- Example: automated driving in control theory: game-theoretic approach with worst-case analysis of other vehicles
- Solution: stay in garage
- Another solution: assume small probability of adversarial behavior, detect probabilistically*, accept tradeoff

Cyberhuman systems contd.

- (Probabilistic) Modal logics to model what humans know and want
  - Will (probably) know a password if they created it or were given it
  - Won't know it otherwise
  - Can't type it unless they know it or guess it
  - Will (probably) act in organization's interest
  - Will (probably) not reveal bad intent to others unless known co-conspirator
  - Etc .

Cyberhuman systems contd.

- Assumption-based theorem provers
  - What are the weakest assumptions about behavior of humans under which the cyberhuman system works (w.h.p.)?
  - E.g., air traffic control systems print out a slip for each flight, one controller takes slip; *assume* they don't copy it out by hand and give to another controller
  - Enables proofs that one system is provably more secure than another (given a common model); perhaps automated synthesis
- Distinction between *inadvertent* and *deliberate* action is probably useful

Reasoning within systems

- Probabilistic reasoning seems obviously useful due to uncertainty -- e.g., about who is trustworthy, which host is compromised, etc.
- Bayesian network methods (Pearl, 1988) provide concise models, effective algorithms
  - Intrusion detection (Gowadia *et al.*, 2005)
  - Cybersecurity situational awareness (Li and Liu, 2007)
  - Reputation systems (Kamvar *et al.*, 2004; Walsh and Sirer, 2006)
- Relational probability models (Koller, Pfeffer, Poole, etc.) provide object-oriented expressive power for reasoning about many, possibly related objects (cf. Shmatikov and Talcott, 2006)

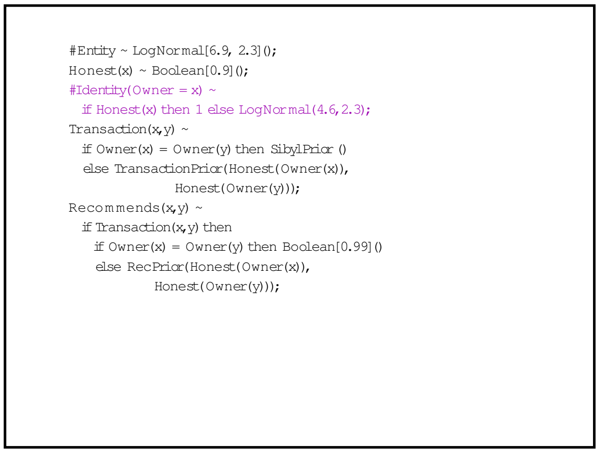
Reasoning within systems contd.

- *Open-universe* languages (Milch and Russell, 2005, 2006) handle worlds where set of objects is not known in advance, object identity is uncertain
- E.g., sibyl attacks on reputation systems (Douceur, 2002), where dishonest participants may generate many false identities

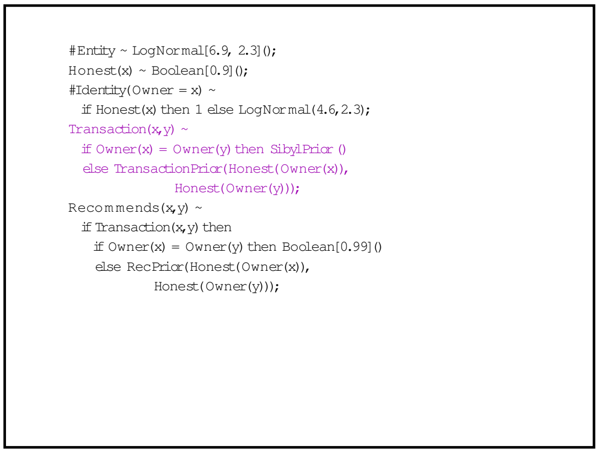
- Typically between 100 and 10,000 real entities
- About 90% are honest, have one identity
- Dishonest entities own between 10 and 1000 identities.
- Transactions may occur between identities
  - If two identities are owned by the same entity (sibyls), then a transaction is highly likely;
  - Otherwise, transaction is less likely (depending on honesty of each identity's owner).
- An identity may recommend another after a transaction:
  - Sibyls with the same owner almost always recommend each other;
  - Otherwise, probability of recommendation depends on the honesty of the two entities.

```
#Entity ~ LogNormal[6.9, 2.3]();
Honest(x) ~ Boolean[0.9]();
#Identity(Owner = x) ~
  if Honest(x) then 1 else LogNormal(4.6,2.3);
Transaction(x,y) ~
  if Owner(x) = Owner(y) then SibylPrior ()
  else TransactionPrior(Honest(Owner(x)),
                Honest(Owner(y)));
Recommends(x,y) ~
  if Transaction(x,y) then
    if Owner(x) = Owner(y) then Boolean[0.99]()
    else RecPrior(Honest(Owner(x)),
             Honest(Owner(y)));
```

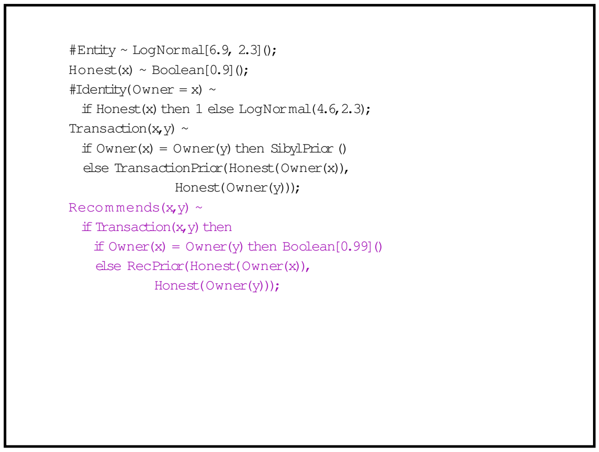
```
#Entity ~ LogNormal[6.9, 2.3]();
Honest(x) ~ Boolean[0.9]();
#Identity(Owner = x) ~
  if Honest(x) then 1 else LogNormal(4.6,2.3);
Transaction(x,y) ~
  if Owner(x) = Owner(y) then SibylPrior ()
  else TransactionPrior(Honest(Owner(x)),
                Honest(Owner(y)));
Recommends(x,y) ~
  if Transaction(x,y) then
    if Owner(x) = Owner(y) then Boolean[0.99]()
    else RecPrior(Honest(Owner(x)),
             Honest(Owner(y)));
```



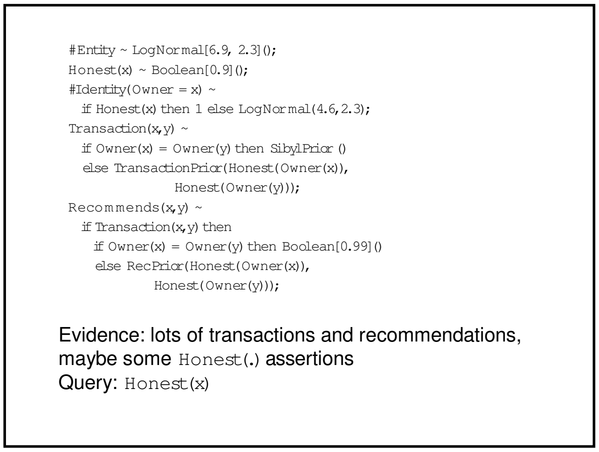
```
#Entity ~ LogNormal[6.9, 2.3]();
Honest(x) ~ Boolean[0.9]();
#Identity(Owner = x) ~
  if Honest(x) then 1 else LogNormal(4.6,2.3);
Transaction(x,y) ~
  if Owner(x) = Owner(y) then SibylPrior ()
  else TransactionPrior(Honest(Owner(x)),
                Honest(Owner(y)));
Recommends(x,y) ~
  if Transaction(x,y) then
    if Owner(x) = Owner(y) then Boolean[0.99]()
    else RecPrior(Honest(Owner(x)),
             Honest(Owner(y)));
```



```
#Entity ~ LogNormal[6.9, 2.3]();
Honest(x) ~ Boolean[0.9]();
#Identity(Owner = x) ~
  if Honest(x) then 1 else LogNormal(4.6,2.3);
Transaction(x,y) ~
  if Owner(x) = Owner(y) then SibylPrior ()
  else TransactionPrior(Honest(Owner(x)),
                Honest(Owner(y)));
Recommends(x,y) ~
  if Transaction(x,y) then
    if Owner(x) = Owner(y) then Boolean[0.99]()
    else RecPrior(Honest(Owner(x)),
             Honest(Owner(y)));
```

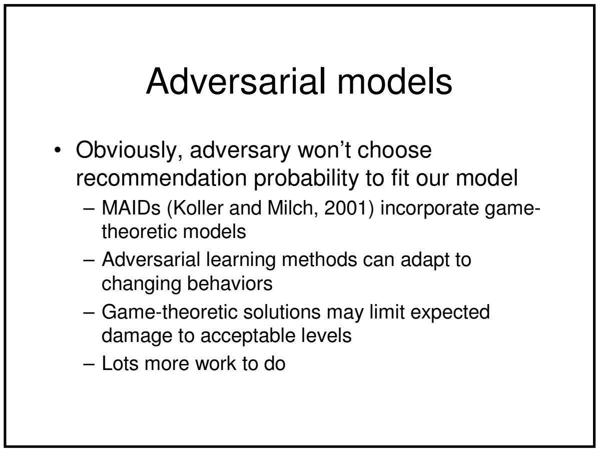


```
#Entity ~ LogNormal[6.9, 2.3]();
Honest(x) ~ Boolean[0.9]();
#Identity(Owner = x) ~
  if Honest(x) then 1 else LogNormal(4.6,2.3);
Transaction(x,y) ~
  if Owner(x) = Owner(y) then SibylPrior ()
  else TransactionPrior(Honest(Owner(x)),
                Honest(Owner(y)));
Recommends(x,y) ~
  if Transaction(x,y) then
    if Owner(x) = Owner(y) then Boolean[0.99]()
    else RecPrior(Honest(Owner(x)),
             Honest(Owner(y)));
```



```
#Entity ~ LogNormal[6.9, 2.3]();
Honest(x) ~ Boolean[0.9]();
#Identity(Owner = x) ~
  if Honest(x) then 1 else LogNormal(4.6,2.3);
Transaction(x,y) ~
  if Owner(x) = Owner(y) then SibylPrior ()
  else TransactionPrior(Honest(Owner(x)),
                Honest(Owner(y)));
Recommends(x,y) ~
  if Transaction(x,y) then
    if Owner(x) = Owner(y) then Boolean[0.99]()
    else RecPrior(Honest(Owner(x)),
             Honest(Owner(y)));
```

Evidence: lots of transactions and recommendations, maybe some `Honest(.)` assertions
Query: `Honest(x)`



Adversarial models

- Obviously, adversary won't choose recommendation probability to fit our model
  - MAIDs (Koller and Milch, 2001) incorporate game-theoretic models
  - Adversarial learning methods can adapt to changing behaviors
  - Game-theoretic solutions may limit expected damage to acceptable levels
  - Lots more work to do