

Title	Theorem-proving Privacy and Anonymity
Author(s)	KAWABE, Yoshinobu
Citation	
Issue Date	2006-11-29
Type	Presentation
Text version	publisher
URL	http://hdl.handle.net/10119/8305
Rights	
Description	Theorem Proving and Provers Meeting(2nd TPP)での発表資料, 開催: 2006年11月29日 ~ 30日, 開催場所: JAIST 情報科学研究科棟II・Collaboration Room 7 (5F)

Theorem-proving Privacy and Anonymity

Yoshinobu KAWABE

NTT Communication Science Laboratories

NTT Corporation

References

- Simulation-based proof method of privacy/anonymity
 - Y. Kawabe, K. Mano, H. Sakurada and Y. Tsukada
Theorem-proving anonymity of infinite state systems
Information Processing Letters, vol. 101, No.1, 2007
 - Y. Kawabe, K. Mano, H. Sakurada and Y. Tsukada
Backward simulations for anonymity
WITS '06 (Full version: submitted for journal publication)
 - I. Hasuo and Y. Kawabe
Probabilistic anonymity via coalgebraic simulations
Submitted for publication

Online *privacy*

Online *anonymity*

is attracting *growing*

- Threats
 - ISPs in EU are forced to keep logs of your web access
- Public concerns
 - You don't care?
- Research interest
 - See *Anonymity Bibliography*
<http://freehaven.net/anonbib/>
 - No decisive definition for “privacy”, “anonymity”, etc.

Overview of this talk

A formal definition of anonymity which is based on **traces**

[ESORICS '96, Schneider & Sidiropoulos]

Proving trace inclusion by simulation

[Lynch & Vaandrager]

- **Simulation-based proof method for trace anonymity**
- **Theorem-proving anonymity**

Contents

- A method to prove anonymity (=privacy)

- Formalization of anonymity
& anonymous simulation technique
- Theorem-proving anonymity/privacy
 - Crowds protocol



What is anonymity?

- Nobody can know “who it is”.
- Key notion: Principle of confusion



Who?



What is anonymity?

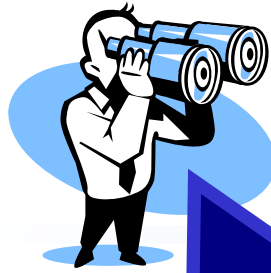
Adversary's viewpoint

This person looks like Kawabe ... but his face is hidden. This person might not be Kawabe.

- Nobody can know “who is ...”
- Key notion: Principle of confusion



Who?



What is an adversary?

Adversary's viewpoint

This person looks like Kawabe ... but his face is hidden. This person might not be Kawabe.

The guys on this photo are **too small**! I cannot recognize Kawabe!

“who is this?”
state of confusion



Releasing sea turtles



Who?

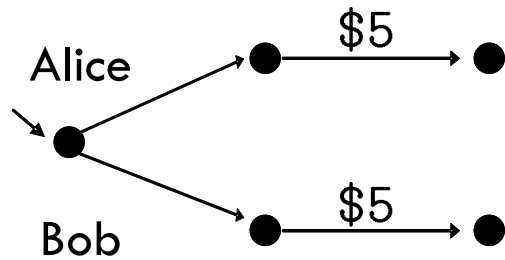
Can you find me?

“Trace” anonymity

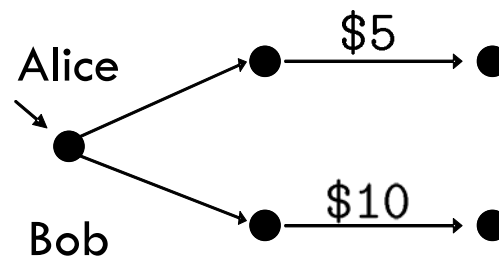
[Schneider&Sidiropoulos, ESORICS’96]

- **Anonymous donation** as an example

X



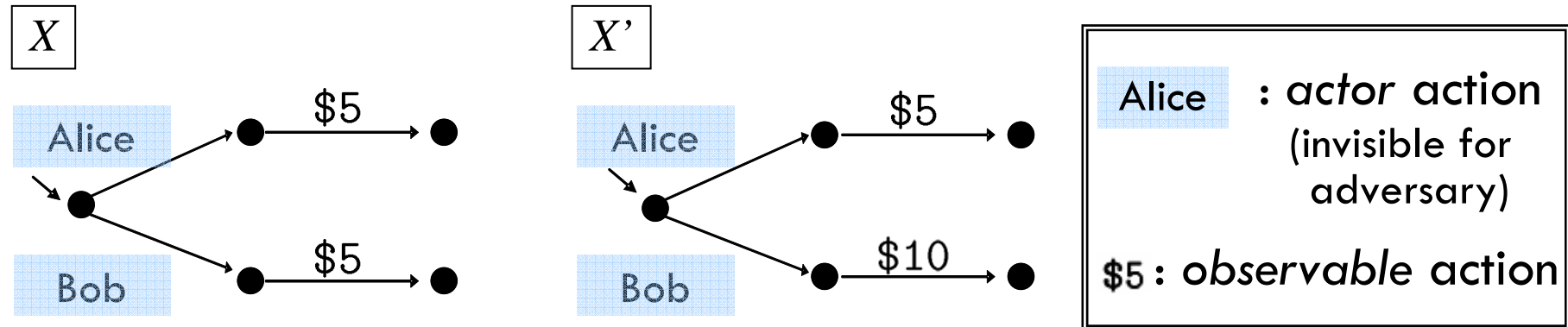
X'



“Trace” anonymity

[Schneider&Sidiropoulos, ESORICS'96]

- **Anonymous donation** as an example

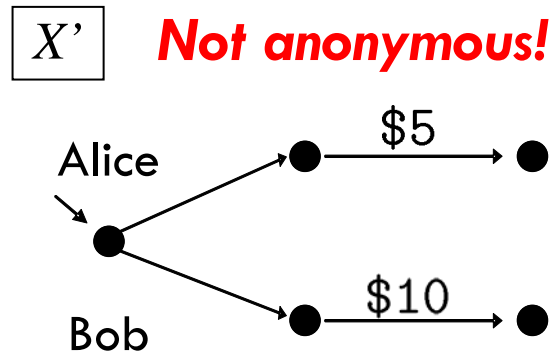
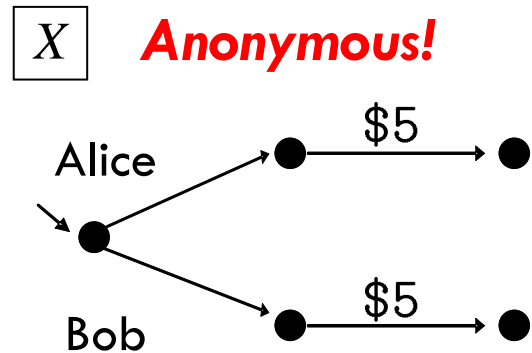


Are these protocols anonymous?

“Trace” anonymity

[Schneider&Sidiropoulos, ESORICS’96]

- **Anonymous donation** as an example

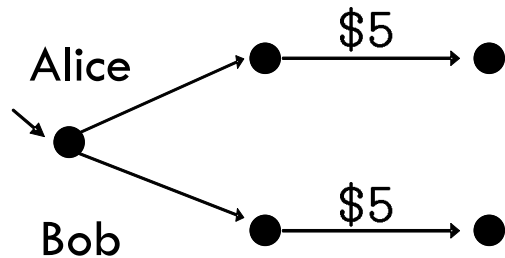


“Trace” anonymity

[Schneider&Sidiropoulos, ESORICS'96]

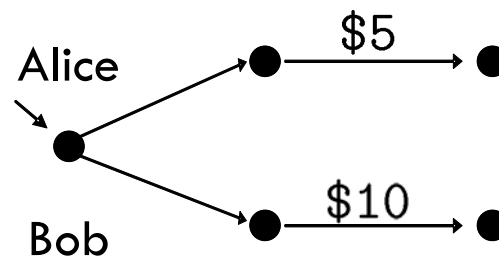
- Anonymous donation as an example

X **Anonymous!**



$$traces(X) = \left\{ \begin{array}{l} Alice.\$5 \\ Bob.\$5 \end{array} \right\}$$

X' **Not anonymous!**



$$traces(X') = \left\{ \begin{array}{l} Alice.\$5 \\ Bob.\$10 \end{array} \right\}$$

Observation \vec{o}, \vec{o}'
can be attributed to
anybody (**confusion!**)

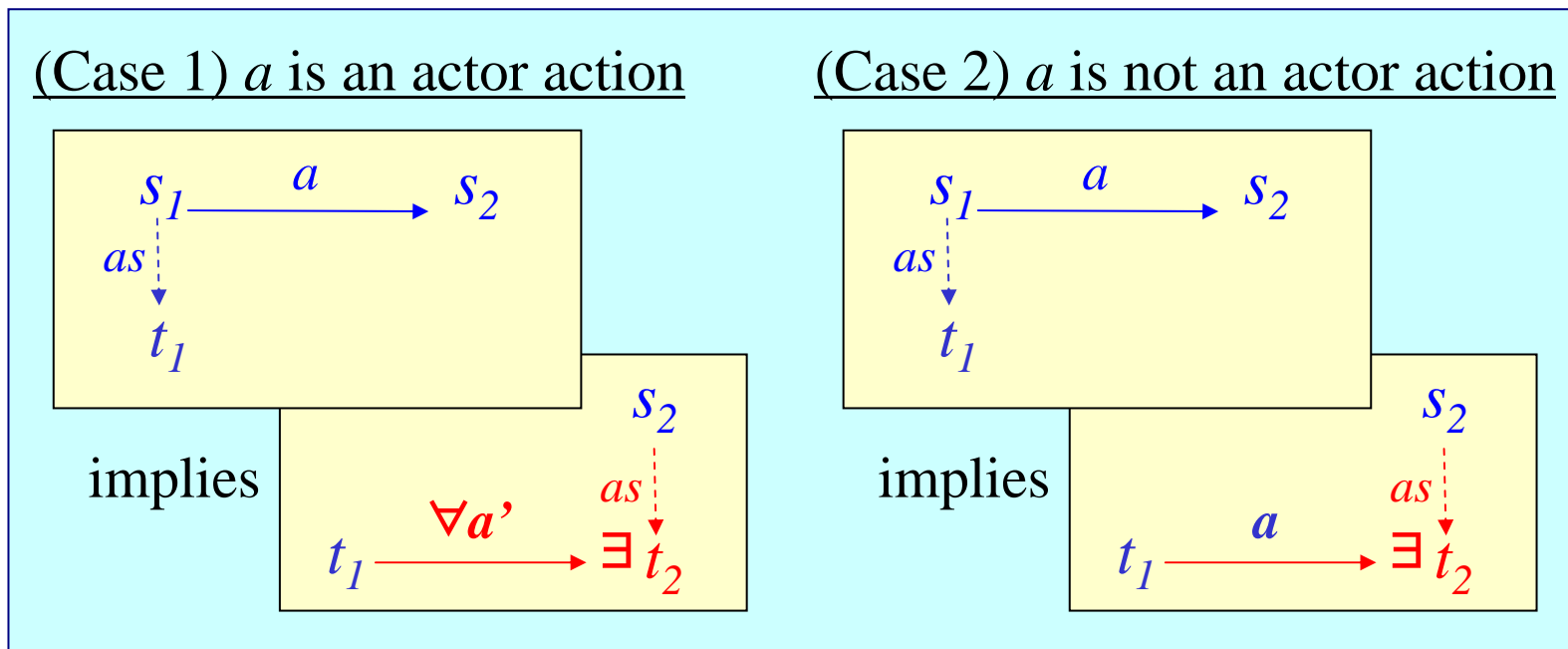
Definition (Trace anonymity)

$$X \text{ is anonymous} \Leftrightarrow \forall \vec{o}, \vec{o}'. \vec{o} \cdot \text{Alice} \cdot \vec{o}' \in traces(X) \\ \implies \vec{o} \cdot \text{Bob} \cdot \vec{o}' \in traces(X), \\ \vec{o} \cdot \text{Chris} \cdot \vec{o}' \in traces(X), \dots$$

How to prove anonymity?

--- Find an **anonymous simulation!**

- Binary relation as over $states(X)$
 1. **Initial state condition:** $as(s, s)$ for any $s \in start(X)$
 2. **Step correspondence condition:**

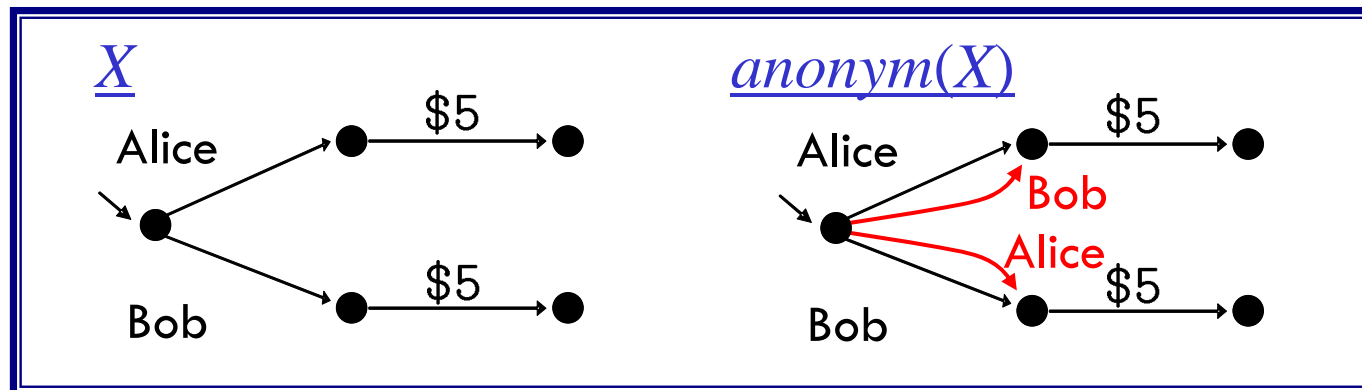


Soundness of the technique

- An anonymous simulation is a **simulation from $\text{anonym}(X)$ to X** .

[Thm] \exists simulation from X to $Y \Rightarrow \text{traces}(X) \subseteq \text{traces}(Y)$.

[Lynch and Vaandrager, Inform.&Comput. 1995]



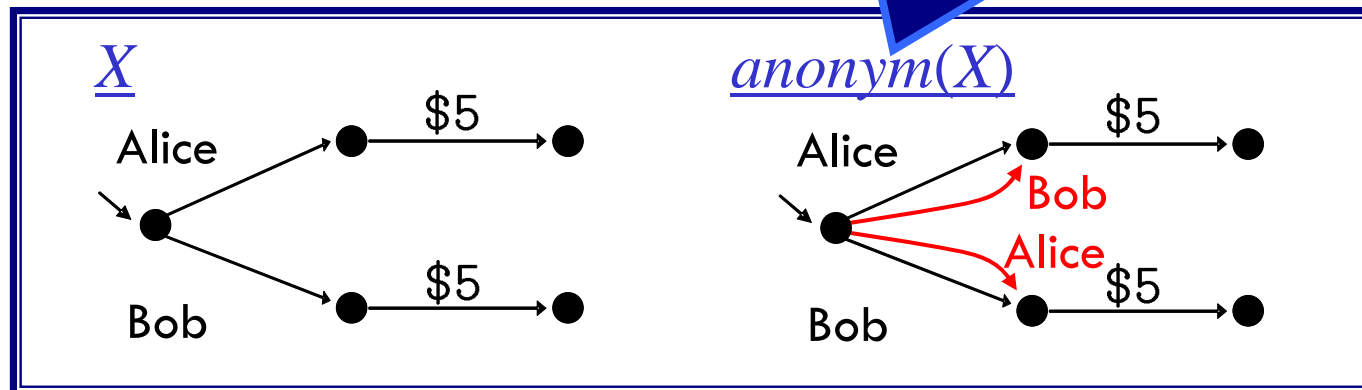
Soundness of the technique

- An anonymous simulation is a **simulation from $\text{anonym}(X)$ to X** .

[Thm] \exists simulation from X to

[Lynch and Vaandrager

“anonymized” version
of X
(trivially anonymous)



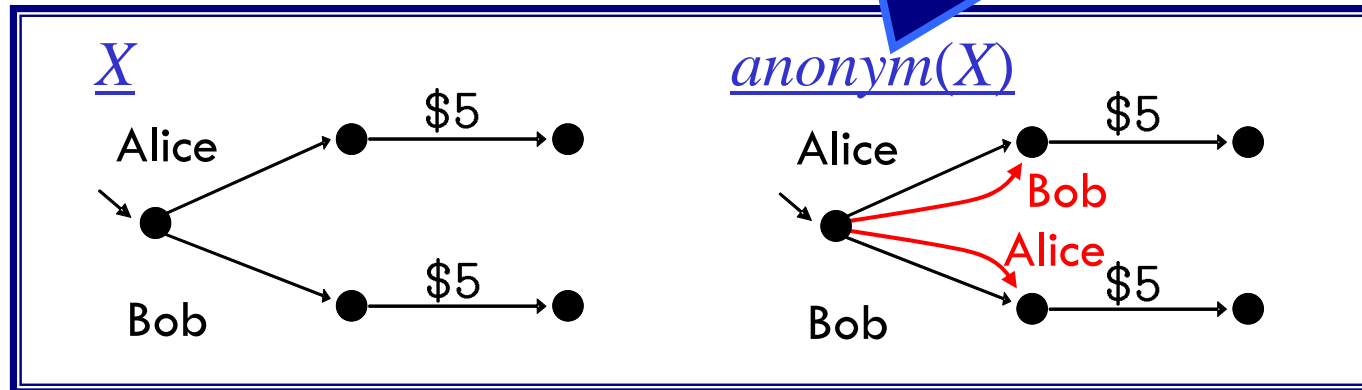
Soundness of the technique

- An anonymous simulation is a **simulation from $\text{anonym}(X)$ to X** .

[Thm] \exists simulation from X to

[Lynch and Vaandrager

“anonymized” version
of X
(trivially anonymous)



$\text{traces}(X) \subseteq \text{traces}(\text{anonym}(X))$ is trivial.

$\Rightarrow \text{traces}(X) = \text{traces}(\text{anonym}(X))$ holds!

Contents

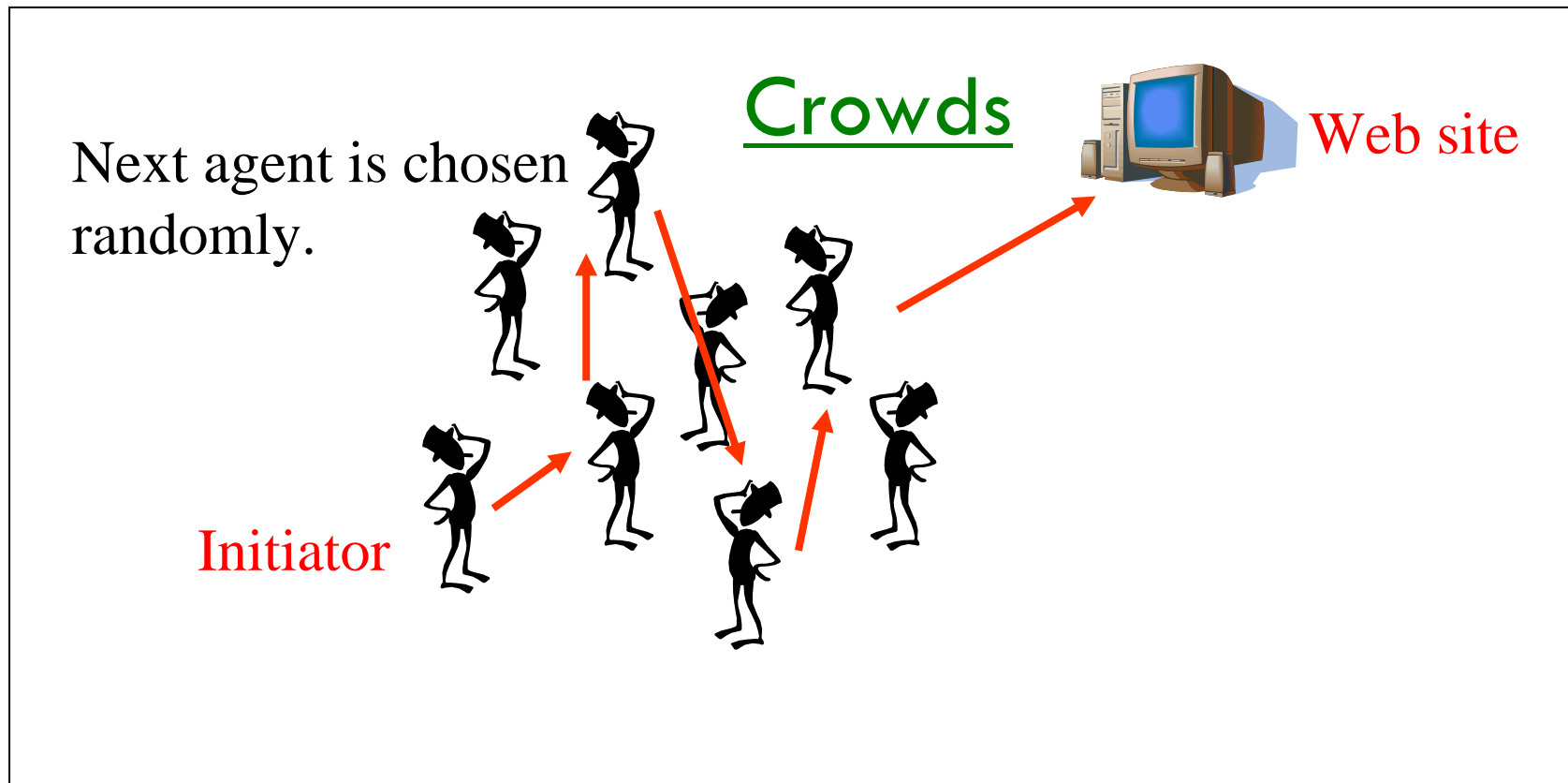
- A method to prove anonymity (=privacy)

- Formalization of anonymity
& anonymous simulation technique
- Theorem-proving anonymity/privacy
 - Crowds protocol

An example: Crowds

[Reiter & Rubin, ACM Trans. 1998]

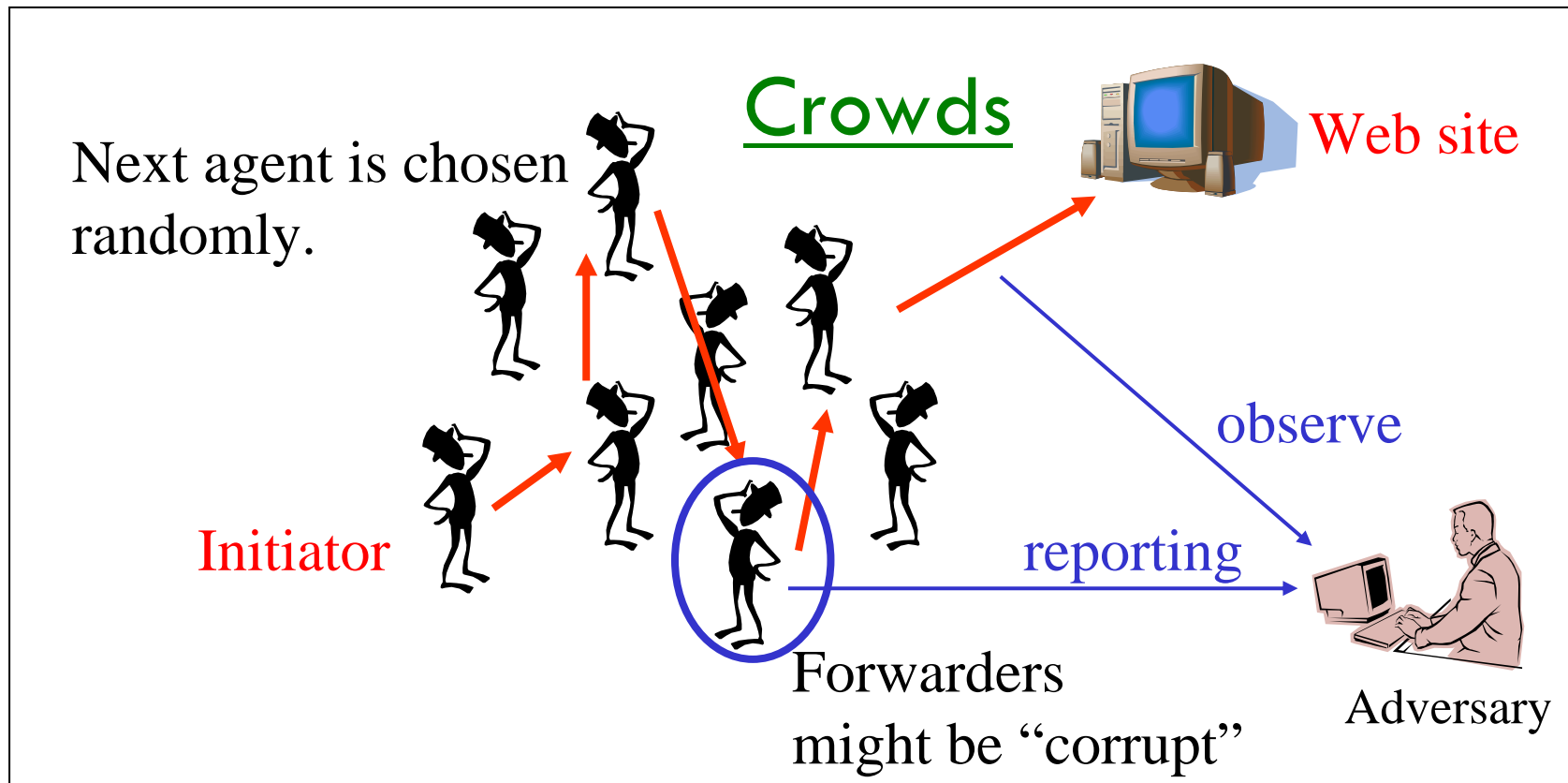
- Comm. system for anonymous web access



An example: Crowds

[Reiter & Rubin, ACM Trans. 1998]

- Comm. system for anonymous web access



Anonymous = the adversary cannot know the initiator.

Theorem-proving anonymity of the Crowds example

- Steps
 - Specify the system in IOA language which is a formal specification language based I/O-automaton
 - Translate the specification into LP's language --- first-order logic formulae --- with IOA-Toolkit
 - Prove anonymity with Larch Prover by proving there is an anonymous simulation

IOA language

- Formal specification language based on I/O-automaton
 - I/O-automaton (N. Lynch): formal system to describe and analyze distributed algorithms
- Formalization of distributed algorithms in IOA
 - Actions: precondition-effect style (i.e. if \sim then \sim)
 - Data: (many-sorted) equational theory
 - LSL (Larch Specification Language)

Specification of Crowds

uses crowdsDatatype

automaton crowds

```
signature
  output start (i:ID)
  internal pass(i:ID, j:ID)
  output source(i:ID, j:ID)
  output out (i:ID)
```

states

```
pc:          PC := init,
mesIsAt:     ID,
mesIsFrom:   ID,
corrp:       Array[ID, Bool]
```

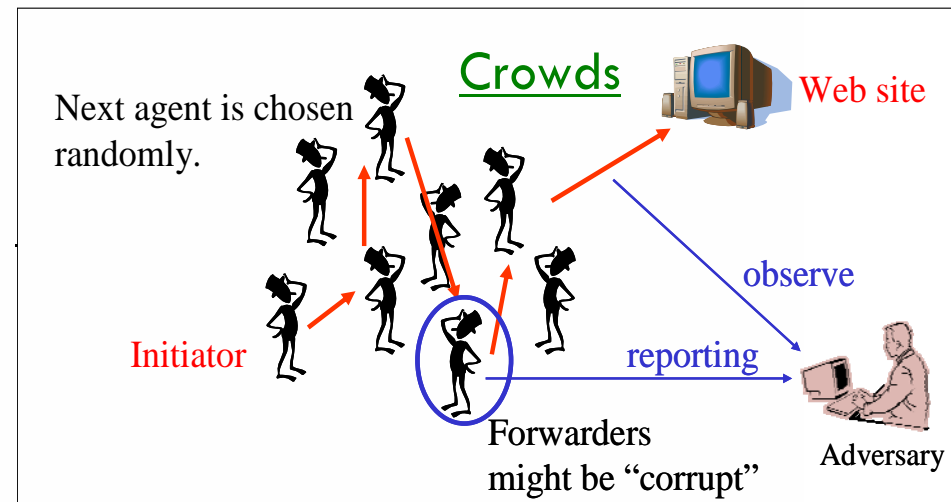
transitions

```
output start(i)          % actor action that represents who is the starter
  pre pc = init
  eff pc := shuffle;
  mesIsAt := i;
  mesIsFrom := i
```

```
internal pass(i, j)
  pre pc = shuffle
  ^ i = mesIsAt
  eff mesIsFrom := i;
  mesIsAt := j
```

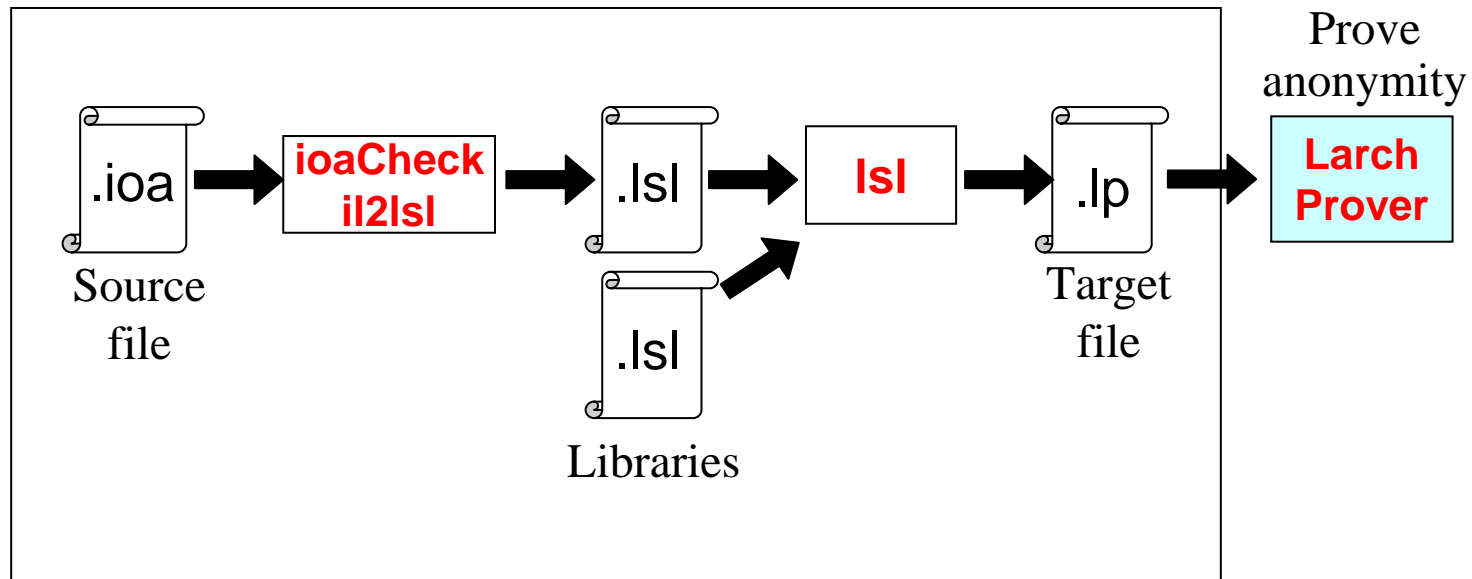
```
output source(i, j)
  pre pc = shuffle
  ^ i = mesIsFrom
  ^ j = mesIsAt
  ^ corrp[j]
  eff do nothing
```

```
output out (i)
  pre pc = shuffle
  ^ i = mesIsAt
  eff pc := terminate
```



IOA-Toolkit

- Collection of formal verification tools for distributed systems



Compiling .ioa into .lp with IOA-Toolkit

Theorem-proving anonymity

- Introducing a candidate relation

```
assert
  as(s, s')
  <=> (s.pc = s'.pc
        ^ (s.corrp[s.mesIsAt] <=> s'.corrp[s'.mesIsAt]))
..
```

- Proving that *as* is an anonymous simulation

```
% --- start state condition
prove start (s:States[crowds]) => as(s, s)
qed
```

Initial state condition

```
% --- step correspondence
prove
  (reachable(s1)
   ^ reachable(s1')
   ^ as(s1, s1')
   ^ enabled(s1, a)
   ^ effect(s1, a) = s2
   ^ a = start(i)
   ^ s1.corrp[i])
=> (\A i':ID (\E s':States[crowds] (\E i'':ID (\E s2':States[crowds]
  ( enabled(s1', start(i'))
    ^ effect(s1', start(i')) = s'
    ^ enabled(s', pass(i', i''))
    ^ effect(s', pass(i', i'')) = s2'
    ^ as(s2, s2'))))))
..
```

Step correspondence
condition
(for actor actions)

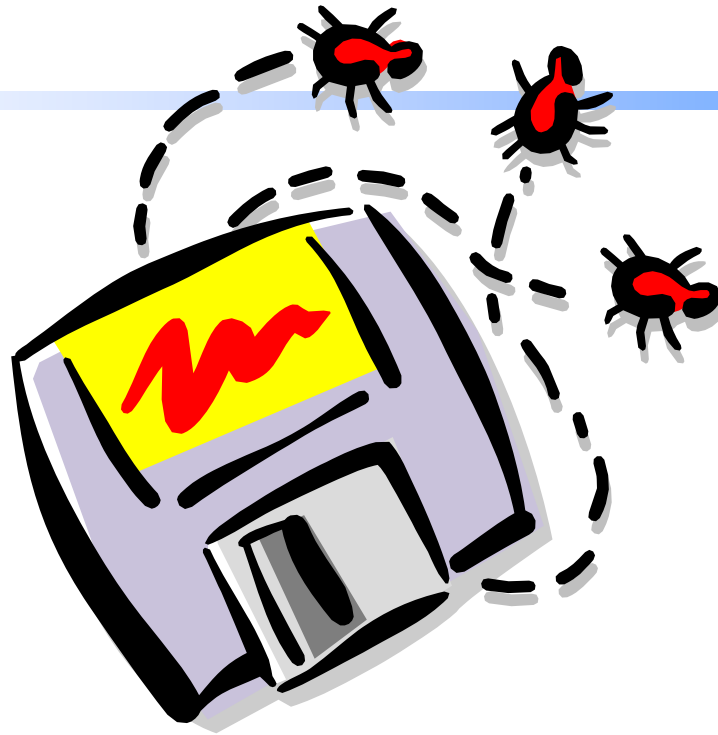
Conclusion

- A technique to theorem-prove anonymity of security protocols
 - Simulation technique for trace-based anonymity
- Example
 - Crowds

Coming soon with
theorem provers

Ongoing work

- Simulation-based proof techniques for **probabilistic anonymity**
 - Conditional anonymity (with Ichiro Hasuo)
 - With coalgebras, our method is extended.
 - Probable innocence (with Hideki Sakurada and Ichiro Hasuo)
- Verifying **anonymity for protocols in the presence of intruders**



Questions?