Computational Logic and Security

EE219 C Class Presentation

Privacy and Contextual Integrity: Frameworks and Applications –

(Barth, Datta, Mitchell, Nissenbaum)

Preserving Secrecy Under Refinement –

(Pavol Černý, Rajeev Alur, Steve Zdancewic)

Paper 1:

Privacy and Contextual Integrity:

Frameworks and Applications

Privacy

- **Privacy** is the ability of an individual or group to keep their lives and personal affairs out of public view, or to control the flow of information about themselves. Privacy can be seen as an aspect of security—one in which trade-offs between the interests of one group and another can become particularly clear. - Wikipedia

- Privacy is an individual's right of determining, ordinarily, to what extent his thoughts, sentiments, and emotions shall be communicated to others. - Common Law Right to Privacy (Samuel Warren and Louis Brandeis, 1890)

Contextual Integrity (CI)

- Contextual Integrity, is respected when norms of appropriateness and distribution are respected; it is violated when any of the norms are infringed.
- Norms of Appropriateness: types of information are/are not appropriate for a given context
- Norms of Distribution (Flow) determine the principles governing distribution (flow) of information from one party to another.
  - S shares information <u>with</u> R at S's discretion
  - R <u>requires</u> S to share information
  - R may freely share information <u>about</u> S
  - R may <u>not</u> share information about S with anyone
  - R may share information about S <u>under</u> specified constraints

Components of information flow in CI

- Sender
- Recipient
- Subject
- Attributes
- Past
- Future
- Combination

Role Based Access Control

XACML

What this paper presents

- A background on contextual integrity
- Formalization in Linear Temporal Logic
- Policy Relations and Operations
- Example cases of privacy laws: HIPAA, GLBA, COPPA

What this paper presents

- A background on contextual integrity
- Formalization in Linear Temporal Logic
- Policy Relations and Operations
- Example cases of privacy laws: HIPAA, GLBA, COPPA

Contextual Integrity (CI)

A transfer of information is:

A(Alice) gives information to B(Bob) about C(Charlie).

There is always an associated context.

A is doctor, B is insurance agency and C is patient.

A is teacher, B is student and C is hiring firm.

Privacy (security of information) expectation depends on <u>what </u>it is, the agents (A,B,C) involved as well as the context.

Contextual Integrity

- Agents abstracted into *roles* (e.g. doctor, patient)
- Particular information abstracted into *types* (e.g., height, age, medical condition)
- *Norms* state what is allowed and what is disallowed
- *Transmission principles* impose past and future requirements on history of agent interaction

What this paper presents

- A background on contextual integrity
- Formalization in Linear Temporal Logic
- Policy Relations and Operations
- Example cases of privacy laws: HIPAA, GLBA, COPPA

Formalization

Modeling communicating agents

- Set of Agents P (who)
- Set of attributes T (what)
- Knowledge state $K = P \times P \times T$

$(p,q,t) \in K$ is p knows the value of attribute t of q.

Data Model

Modeling attribute inference: If postal address is known, postal code is known.

- Computational rule (T', t) where $T' \subseteq T$ and $t \in T$. We say t is derivable from T
- Learning relation on knowledge states
- $\forall k \, \forall p,q \in P$ if $\{p\} \times \{q\} \times T \subseteq k$ and t is derivable from T, then $k \rightarrow k'$ where

  $k' = k \cup \{(p,q,t)\}$. The transitive closure of $\rightarrow$ defines the new knowledge state from existing state k after adding element (p,q,t).

Communication model

Modeling messages

- A message m ⊆ P x T, which is closed under computation rules.
- A communication action would be (p_1, p_2, m) where p_1 is sender, p_2 is receiver and m is the message.
- A communication action transform knowledge states as follows:

$$\forall \kappa, \hat{\kappa}. \forall p_1, p_2 \in \mathcal{P}. \forall m \in \mathcal{M}.$$

$$\text{if } \kappa \xrightarrow{I} \hat{\kappa} \text{ and } \{p_1\} \times \text{content}(m) \subseteq \hat{\kappa},$$

$$\text{then } \kappa \xrightarrow{(p_1, p_2, m)} \kappa',$$

where $\kappa' = \hat{\kappa} \cup \{p_2\} \times \text{content}(m)$. The contents of the message are first computed by the sender (at $\hat{\kappa}$) and then learned by the recipient (at κ').

CI model

Modeling contextual integrity

- Set of Roles R (partially ordered set - specialization)
- Partition of R ie. Set of contexts C
- A agent can have multiple roles.
- History of agent is an infinite trace: a sequence of triples (k,r,a) where k is knowledge state, r is role state and a is communication action and $\kappa_n \xrightarrow{a_{n+1}} \kappa_{n+1}$, for all $n \in \mathbb{N}$.

Temporal Logic

Syntax of logic

$$\varphi ::= \mathrm{send}(p_1, p_2, m) \mid \mathrm{contains}(m, q, t) \mid \mathrm{inrole}(p, r) \mid \mathrm{incontext}(p, c) \mid t \in t' \mid$$
$$\varphi \wedge \varphi \mid \neg\varphi \mid \varphi\mathcal{U}\varphi \mid \varphi\mathcal{S}\varphi \mid \bigcirc\varphi \mid \exists x : \tau.\varphi$$

- t ∈ t′ means t can be inferred from t′ ⊆ T.
- Rest are familiar to us – LTL with existential quantifier.

Norms

Formula representing contextual norms

$$\sigma \models \Box \forall p_1, p_2, q : P. \forall m : M. \forall t : T. \text{incontext}(p_1, c) \wedge$$
$$\text{send}(p_1, p_2, m) \wedge \text{contains}(m, q, t) \rightarrow \bigvee_{\varphi^+ \in \text{norms}^+(c)} \varphi^+ \wedge \bigwedge_{\varphi^- \in \text{norms}^-(c)} \varphi^-$$

where norm$^+$ and norms$^-$ are as follows

$$\text{positive norm:} \quad \text{inrole}(p_1, \hat{r}_1) \wedge \text{inrole}(p_2, \hat{r}_2) \wedge \text{inrole}(q, \hat{r}) \wedge (t \in \hat{t}) \wedge \theta \wedge \psi$$
$$\text{negative norm:} \quad \text{inrole}(p_1, \hat{r}_1) \wedge \text{inrole}(p_2, \hat{r}_2) \wedge \text{inrole}(q, \hat{r}) \wedge (t \in \hat{t}) \wedge \theta \rightarrow \psi$$

- θ is an agent constraint (free of temporal operators)
- ψ represents principle of transmissions and is temporal phenomenon describing past and future actions of agents.
- Attribute closed downward for positive norm and upward for negative norm

What this paper presents

- A background on contextual integrity
- Formalization in Linear Temporal Logic
- Policy Relations and Operations
- Example cases of privacy laws: HIPAA, GLBA, COPPA

Policy Relations and Operations

- Policy Consistency -> LTL satisfiability
- Policy Refinement -> Implication
- Policy Combination -> Conjunction/Disjunction
- Strong compliance -> Satisfiability
- Weak compliance -> LTL runtime verification (efficient)

Benefits:

- Non-ambiguous representation and enforcement
- Automated standard LTL tools

Policy Relations and Operations

- Policy Consistency -> LTL satisfiability

Def. A privacy policy θ is *consistent* with a purpose α if there exists a trace σ such that $\sigma \models \theta \wedge \alpha$.

- Policy Refinement/Entailment -> implication

Def. A privacy policy θ_1 *entails* a policy θ_2 if the LTL formula $\theta_1 \rightarrow \theta_2$ is valid over traces.

Compliance modeling

- Weak compliance -> LTL runtime verification

Def. Given a finite past history σ, an action a *weakly complies* with privacy policy θ if $\sigma \cdot a$ is a path in the tableau of θ that starts at an initial θ-atom. The *future requirements* of $\sigma \cdot a$ is the LTL formula ψ such that, for all traces σ',

$$\sigma' \models \psi \text{ if, and only if, } \sigma \cdot a \cdot \sigma' \models \theta.$$

- Strong Compliance -> Satisfiability

Def. Given a finite past history σ, an action a *strongly complies* with a privacy policy θ if there exists a trace σ' such that $\sigma \cdot a \cdot \sigma' \models \theta$.

What this paper presents

- A background on contextual integrity
- Formalization in Linear Temporal Logic
- Policy Relations and Operations
- Example cases of privacy laws: HIPAA, GLBA, COPPA

Example (HIPAA Privacy Rule)

- Addressed information flow: Transfer of protected health information (phi) about patients from covered entities (e.g. hospitals) to health care providers.
- Sender role: Covered entity (e.g. hospitals)
- Recipient role: Health care provider
- Subject role: Patient
- Information type: Protected health information

Example (HIPAA Privacy Rule)

Legislative statement expresses
  permissible actions - positive norms
  forbidden actions - negative norms

Let us look at 5 examples of norms in HIPAA and how it is modeled.
(4 positive and 1 negative norm)

Norm 1 Positive

Any person may be given information about himself/herself until it conflicts with some forbidding norm.

Norm 1 Positive

Any person may be given information about himself/herself until it conflicts with some forbidding norm.

$$\mathrm{inrole}(p_1, \textit{covered-entity}) \wedge \mathrm{inrole}(p_2, \textit{individual}) \wedge (q = p_2) \wedge (t \in \textit{phi})$$

Norm 2 Positive

Healthcare provider is entitled to information about its patient from hospital.

Norm 2 Positive

Healthcare provider is entitled to information about its patient from hospital.

$$\text{inrole}(p_1, \textit{covered-entity}) \wedge \text{inrole}(p_2, \textit{provider}) \wedge \text{inrole}(q, \textit{patient}) \wedge (t \in \textit{phi})$$

Norm 3 Negative

A psychotherapy-note can not be shown to the subject until the psychiatrist approves.

$$\mathrm{inrole}(p_1, \mathit{covered\text{-}entity}) \wedge \mathrm{inrole}(p_2, \mathit{individual}) \wedge (q = p_2) \wedge (t \in \mathit{psychotherapy\text{-}notes}) \rightarrow$$
$$\Diamond\!\!\!\!\!-\;\exists p : P.\,\mathrm{inrole}(p, \mathit{psychiatrist}) \wedge \mathrm{send}(p, p_1, \mathit{approve\text{-}disclose\text{-}psychotherapy\text{-}notes})$$

S only if A has happened in past

Norm 1 permits, norm 3 prohibits; no contradiction as norm 1 only permits doesn't mandate.

Norm 4 Positive

Covered entity (hospital, health-centre) can release information about location and condition of any individual to anyone enquiring about him with name.

$$\mathrm{inrole}(p_1, \mathit{covered\text{-}entity}) \wedge \mathrm{inrole}(p_2, \mathit{individual}) \wedge \mathrm{inrole}(q, \mathit{individual}) \wedge (t \in \mathit{condition\text{-}and\text{-}location}) \wedge$$
$$\diamondminus \exists m' : M.\mathrm{send}(p_2, p_1, m') \wedge \mathrm{contains}(m', q, \mathit{name})$$

Some body (p2) sent hospital (p1) message with name of q

The condition and location of q is given to p2 by the hospital (p1)

Norm 5 Positive

Clergy can obtain directory information that contains (directly or transitively) individual's name, general condition, location.

$$\mathrm{inrole}(p_1, \textit{covered-entity}) \wedge \mathrm{inrole}(p_2, \textit{clergy}) \wedge \mathrm{inrole}(q, \textit{individual}) \wedge (t \in \textit{directory-information})$$

COPPA and GLBA

An exercise in specifying informal specifications in LTL.

Lets “run over” a couple of examples from COPPA.

COPPA

- When a child sends information to a website the parents must have received a notice, granted permission and since not revoked permission.

$$\mathrm{inrole}(p_1, \mathit{child}) \wedge \mathrm{inrole}(p_2, \mathit{web\text{-}site}) \wedge (q = p_1) \wedge (t \in \mathit{protected\text{-}info}) \rightarrow$$
$$\exists p : P.\, \mathrm{inrole}(p, \mathit{parent}) \wedge \neg\, \mathrm{send}(p, p_2, \mathit{revoke\text{-}consent})\, \mathcal{S}$$
$$(\mathrm{send}(p, p_2, \mathit{grant\text{-}consent}) \wedge \diamondminus\, \mathrm{send}(p_2, p, \mathit{privacy\text{-}notice}))$$

Website sends privacy notice to parents and they give consent

No revoking of consent

Child sends protected information to website

COPPA

- Website must delete information after a parent revokes permission ?

"Present infrastructure does not support removal of information.''

Can we model revoke as reassignment of existing attributes to "unassigned" – is there a better way to model "forgetting actions" (without actually having a stack!)

What this work does not cover?

- Anonymous information (like in HIPAA)

Name-SID-Year in grad school-number of library visits is private BUT

Year in grad school-number of library visits is NOT

- 'Averaged' information (group attribute)

Name-Age-Telephone Bill is private BUT

Average data Age-Telephone Bill is NOT

- Data value based policy not just type-based

Load distribution of a network – peak hours when it is vulnerable to DOS attack could be kept confidential

Paper 2:

Preserving Secrecy Under Refinement

Motivation

- Privacy = Secrecy
- Implementation = Refinement
- Secrecy preserving refinement needed to implement privacy preserving laws

- Given the HIPAA, after we have written the laws using the previous paper's technique as LTL , how do we ensure that an Hospital Information System is consistent with the privacy laws ?
- Instead of model checking the entire system, can we build an abstraction which would be safe with respect to the privacy (secrecy) rules ?

TO DO THIS WE NEED "SECRECY PRESERVING REFINEMENT"

Summarization

Property under consideration – P

T' – abstract trace Ti – concrete trace

T1(P) T2(P) T3(~P) → T' (P is secret)

T1(P) T2(P) T3(P) → T' (P inferred)

T1(~P) T2(~P) T3(~P) → T' (~P inferred)

Outline

- Defining a framework for secrecy
- Comparison with existing notions of secrecy
- Non-expressibility in mu-calculus
- Secrecy preserving refinement
- Simulation based proof method
- Applications

Outline

- Defining a framework for secrecy
- Comparison with existing notions of secrecy
- Non-expressibility in mu-calculus
- Secrecy preserving refinement
- Simulation based proof method
- Applications

Defining Secrecy

3 parameters :

- Property (predicate over system variales) to be kept secret α
  (like first_letter(password) = s)

- Distinguishing power of the observer (Observation equivalence of runs) $\equiv$
  (like “r1 $\equiv$ r2 if the respective last states are equivalent obs(last(r1)) = obs(last(r2))”)

- Executions of interest β
  (like “all runs terminating without error”)

Defining Secrecy

"α is secret in β with respect to ≡"

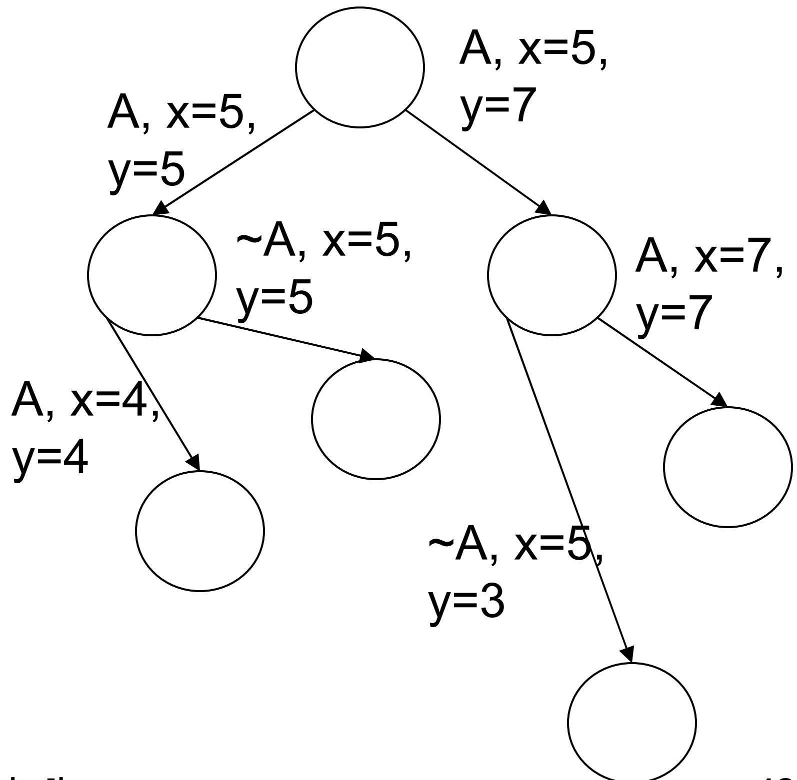
$$IP(r,\alpha,\equiv)=\begin{cases} T \text{ iff } \forall r' \equiv r \Rightarrow r' \in \alpha \\ F \text{ iff } \forall r' \equiv r \Rightarrow r' \notin \alpha \\ M \text{ otherwise} \end{cases}$$

α is secret in β with respect to ≡ iff for all r in β, IP(r, α, ≡)=M.

Example

- ≡ : Capital letters denote secret information
  – not available to observer

α , β	2 A or G A	Only 1 A
G x = 5	False	True
G y = x	True	m



Outline

- Defining a framework for secrecy
- Comparison with existing notions of secrecy
- Non-expressibility in mu-calculus
- Secrecy preserving refinement
- Simulation based proof method
- Applications

Linear-time Secrecy

Special case with

- r≈r’ iff the sequences of labels are the same; strong, timing-sensitive equivalence
- r≈wr’ iff the sequence of labels are the same, modulo ε label

Example:

A: x=?; y=0; z=x; send z;

B: x=?; y=0; z=y; send z;

By looking at sent bit, both yield ttt0. A reveals x was set to 0, B does not.

Noninterference

Special case with

- r≈r' iff their initial states share the values of low variables and the same holds for their final states
- β set of all terminating runs
- noninterference w.r.t. P iff for all α in P, α is secret in β w.r.t. ≈

The above ensures that if two input states share the same values of low variables, then the behaviors of the program executed from these states are indistinguishable by the observer."

Perfect Security Property

Special case with

- r≈r' iff their sub-sequences of low-security labels are equal.
- P contains a property αh for each high-security action h.
- αh holds for a run r if h occurs in r
- β is the set of all runs
- PSP holds iff for all αh in P, αh is secret in β w.r.t. ≈

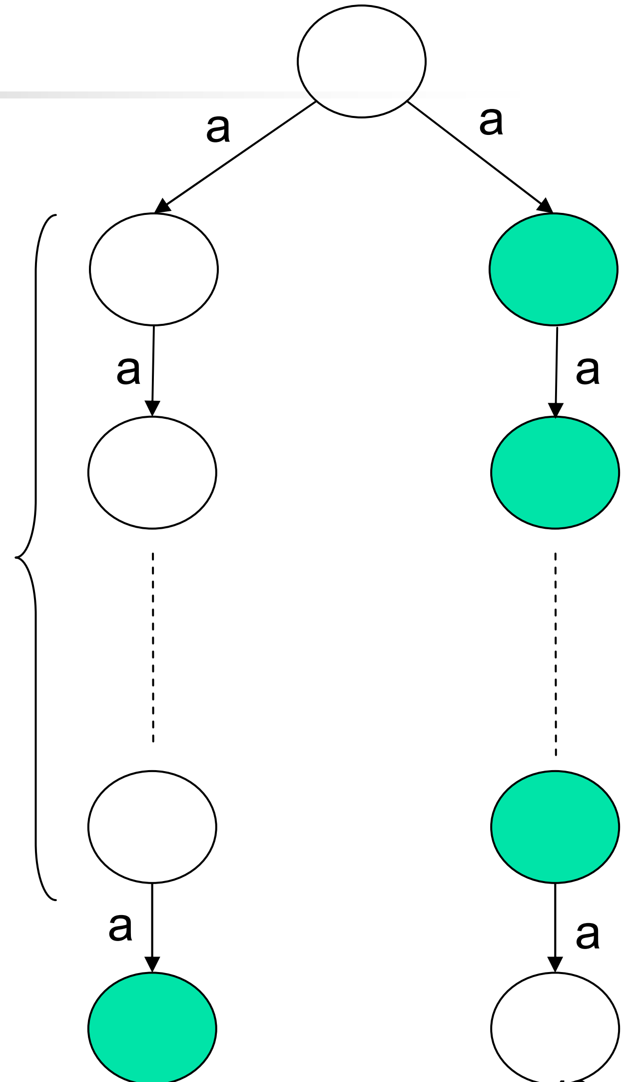
- The above ensures that though the observer knows the specification (the set of all possible traces) and observes the low events, but he or she cannot deduce whether a high-security event occurred or not.

Outline

- Defining a framework for secrecy
- Comparison with existing notions of secrecy
- Non-expressibility in mu-calculus
- Secrecy preserving refinement
- Simulation based proof method
- Applications

Specifying Secrecy in Temporal Logics

Secrecy is not a property of a single run.



Mu-Calculus

- **Thm:** Secrecy is **not** definable in μ-calculus.

  Proof: It is not a regular tree language.

(We can work it on the board if required subject to limitations of time)

Outline

- Defining a framework for secrecy
- Comparison with existing notions of secrecy
- Non-expressibility in mu-calculus
- Secrecy preserving refinement
- Simulation based proof method
- Applications

Secrecy-preserving Refinement

- “If α is secret in S, then α is secret in I”, where implementation I refines specification S ?

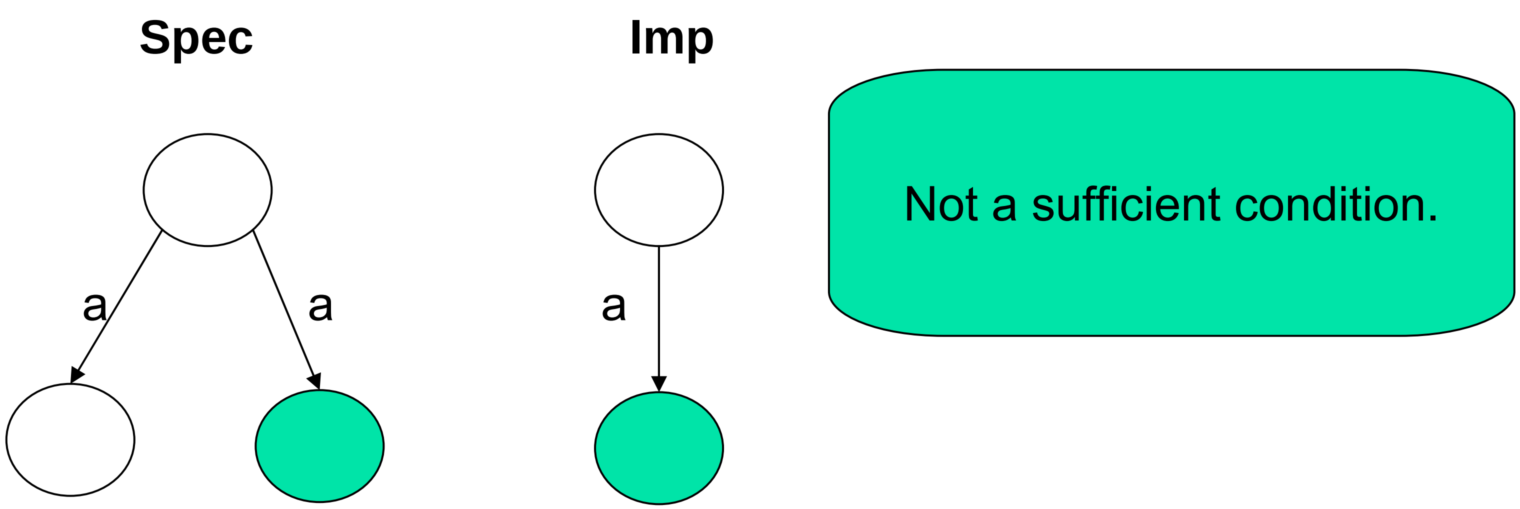
- Notation : Let P be the set of all the secret properties, then we need to define “ I P-refines S” which is consistent with above notion of secrecy preservation.

(Slide from icalp06 presentation)

Standard refinement

- Definition (standard refinement):

  All behaviors of Imp are allowed by Spec (*Runs(Imp)* $\subseteq$ *Runs(Spec)*).

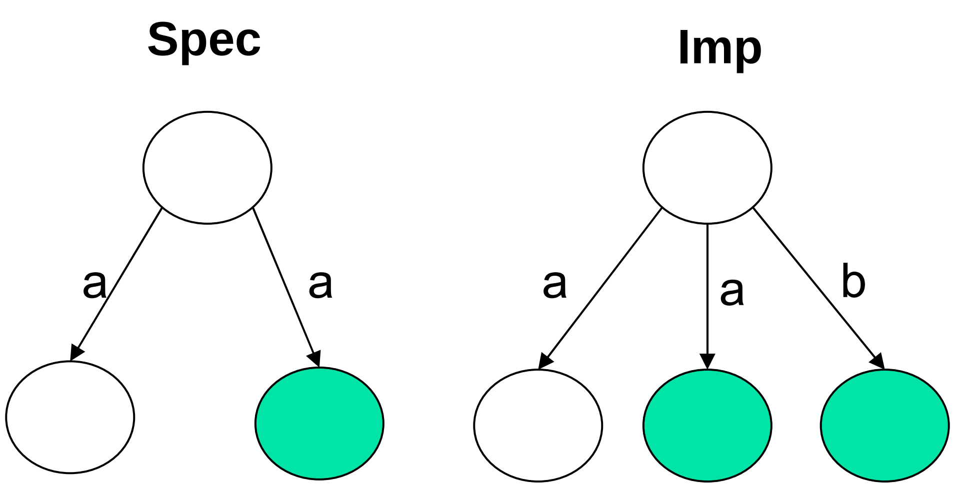


(Slide from icalp06 presentation)

Standard refinement

- Definition (standard refinement):

  All behaviors of Imp are allowed by Spec.
  (*Runs(Imp)* $\subseteq$ *Runs(Spec)*)



It is a necessary condition.

Definition

Intuition: Refinement which preserves secrecy needs extending equivalence relation to the runs of the two systems.

Equiv ≡ is now a subset of

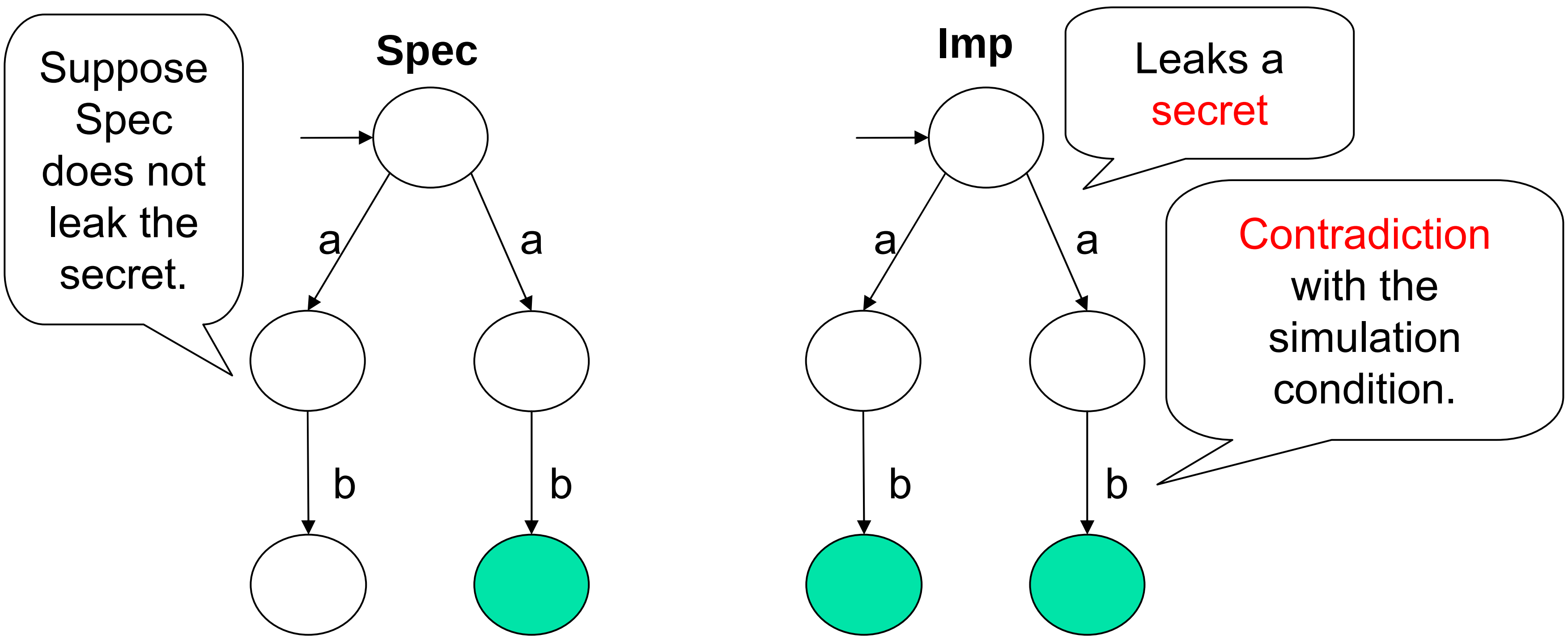
(Runs in Spec U Runs in Imp) x (Runs in Spec U Runs in Imp)

Secrecy-preserving refinement
Let T_{spec}, T_{imp} be two labeled transition system, let $\mathcal{P}$ be a set of properties and let $\equiv$ be an equivalence relation on $R(T_{spec}) \cup R(T_{imp})$. T_{imp} *$\mathcal{P}$-refines* T_{spec} w.r.t. $\equiv$ iff for all runs $r \in R(T_{imp})$, there exists a run $r' \in R(T_{spec})$ such that $r \equiv r'$ and for all properties $\alpha \in \mathcal{P}$, $IP(r, \alpha, \equiv) \sqsubseteq IP(r', \alpha, \equiv)$.

(Slide from icalp06 presentation)

Simulation.

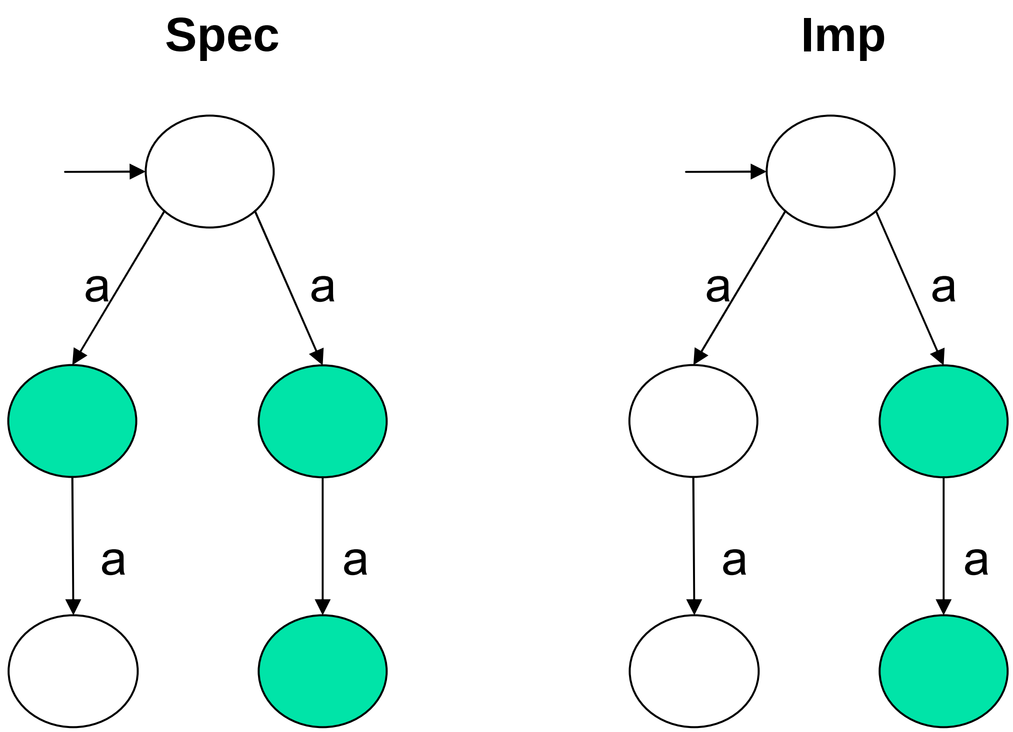
Thm: If *Runs(Imp)* $\subseteq$ *Runs(Spec)* and *Imp* simulates *Spec*, then *Imp P*-refines *Spec*.



(Slide from icalp06 presentation)

Simulation.

Runs(imp) $\subseteq$ *Runs(Spec)* and *Imp* simulates *Spec*



Not a necessary condition.

Outline

- Defining a framework for secrecy
- Comparison with existing notions of secrecy
- Non-expressibility in mu-calculus
- Secrecy preserving refinement
- Simulation based proof method
- Applications

Applications

- Verifying cryptographic algorithms.
- Validating refinement and implementation of protocols.
- Validating refinement and implementation of formal policy formulations as we saw in the first part.
- Malware detection: if M|m P-refines M, then m is not a spyware with respect to properties in P about M. Is it so ?

Thanks ! 

Any Questions ?

Where else to look at –

1. J. Jurjens. Secrecy-preserving refinement. In J. N. Oliveira and P. Zave, editors, FME 2001.
2. H. Nissenbaum, Privacy as Contextual Integrity. Washington Law Review, v79 #1, February 04, 2004. 119-158.
3. L. Introna and H. Nissenbaum. Shaping the Web: Why the Politics of Search Engines Matters. The Information Society, 16(3):1-17, 2000.
4. http://www.nyu.edu/projects/nissenbaum/papers/economist.pdf
5. Jan Jurjens. Composability of secrecy. In International Workshop on Mathematical Methods, Models and Architectures for Computer Networks Security (MMM-ACNS 2001), LNCS, St. Petersburg, 21-23 May 2001. Springer. 16