

I8734: Foundations of Privacy

Policy Auditing over Incomplete Logs:
The reduce algorithm

Anupam Datta
Carnegie Mellon University

Fall 2015

Example from HIPAA Privacy Rule

A covered entity may disclose an individual's protected health information (phi) to law-enforcement officials for the purpose of identifying an individual if the individual made a statement admitting participating in a violent crime that the covered entity believes may have caused serious physical harm to the victim

▶ Concepts in privacy policies

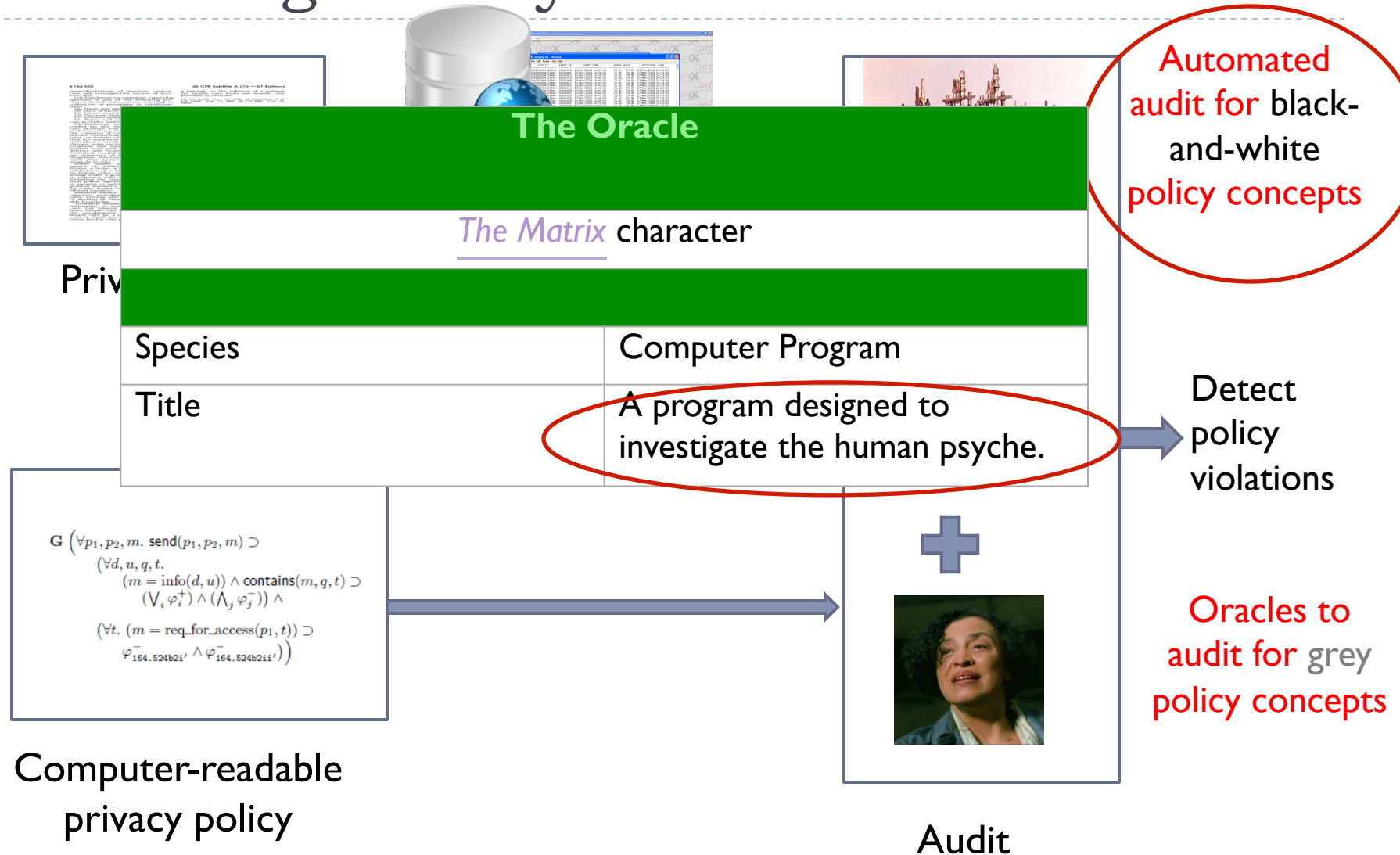
- ▶ **Actions:** send(p1, p2, m)
- ▶ **Roles:** inrole(p2, law-enforcement)
- ▶ **Data attributes:** attr_in(prescription, phi)
- ▶ **Temporal constraints:** in-the-past(state(q, m))

- ▶ **Purposes:** purp_in(u, id-criminal))
- ▶ **Beliefs:** believes-crime-caused-serious-harm(p, q, m)

Black-and-white concepts

Grey concepts

Detecting Privacy Violations



Auditing Black-and-White Policy Concepts

With D. Garg (CMU → MPI-SWS) and L. Jia (CMU)

2011 ACM Conference on Computer and
Communications Security

Key Challenge for Auditing

Audit Logs are Incomplete

Future: store only past and current events

Example: Timely data breach notification refers to future event

Subjective: no “grey” information

Example: May not record evidence for purposes and beliefs

Spatial: remote logs may be inaccessible

Example: Logs distributed across different departments of a hospital

Abstract Model of Incomplete Logs

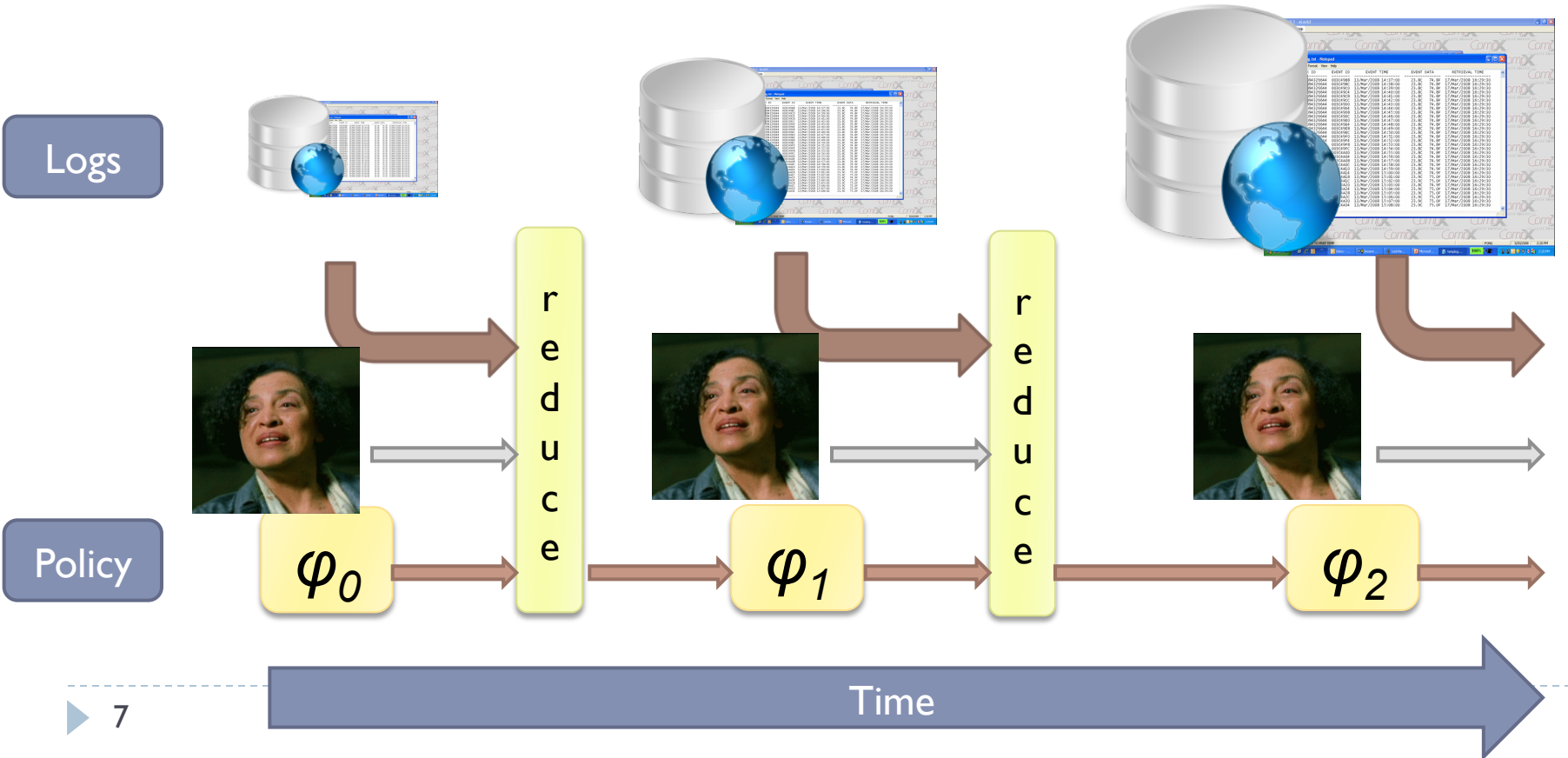
Model **all** incomplete logs uniformly as **3-valued structures**

$$\mathcal{L}(P) \in \{tt, ff, uu\}$$

Define **semantics** (meanings of formulas) over 3-valued structures

reduce: The Iterative Algorithm

$$\text{reduce}(\mathcal{L}, \varphi) = \varphi'$$



Syntax of Policy Logic

Atoms	P	$::=$	$p(t_1, \dots, t_n)$
Formulas	φ	$::=$	$P \mid \top \mid \perp \mid$ $\varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid$ $\forall \vec{x}.(c \supset \varphi) \mid \exists \vec{x}.(c \wedge \varphi)$
Restrictions	c	$::=$	$P \mid \top \mid \perp \mid c_1 \wedge c_2 \mid$ $c_1 \vee c_2 \mid \exists x.c$

- ▶ First-order logic with restricted quantification over *infinite domains* (challenge for reduce)
- ▶ Can express timed temporal properties, “grey” predicates

Example from HIPAA Privacy Rule

A covered entity may disclose an individual's protected health information (phi) to law-enforcement officials for the purpose of identifying an individual if the individual made a statement admitting participating in a violent crime that the covered entity believes may have caused serious physical harm to the victim

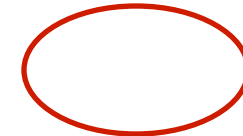
$$\begin{aligned} & \forall p1, p2, m, u, q, t. \\ & (\text{send}(p1, p2, m) \wedge \\ & \text{tagged}(m, q, t, u) \wedge \\ & \text{attr_in}(t, \text{phi})) \\ & \supset \text{inrole}(p1, \text{covered-entity}) \wedge \text{inrole}(p2, \text{law-enforcement}) \\ & \quad (\text{purp_in}(u, \text{id-criminal})) \wedge \\ & \quad \wedge \exists m' \diamond \text{state}(q, m') \wedge \text{is-admission-of-crime}(m') \\ & \quad \wedge \text{believes-crime-caused-serious-harm}(p1, q, m') \end{aligned}$$

reduce: Formal Definition

$(\top \text{ if } \rho(P) = \top\top$

General Theorem: If initial policy passes a syntactic **mode check**, then finite substitutions can be computed

$\text{reduce}(L, \forall x.\varphi)$



c is a finite substitution

Applications: The entire HIPAA and GLBA Privacy Rules pass this check



Example

$\varphi =$

$\forall p1, p2, m, u, q, t.$

$(\text{send}(p1, p2, m) \wedge$
 $\text{tagged}(m, q, t, u) \wedge$
 $\text{attr_in}(t, \text{phi}))$

$\wedge \text{inrole}(p1, \text{covered-entity}) \wedge \text{inrole}(p2, \text{law-enforcement})$

$\wedge \text{purp_in}(u, \text{id-criminal})$

$\wedge \exists m'. \text{state}(q, m')$

$\wedge \text{is-admission-of-crime}(m')$

$\wedge \text{believes-crime-caused-serious-harm}(p1, m')$

$\{ p1 \rightarrow \text{UPMC},$
 $p2 \rightarrow \text{allegeny-police},$
 $m \rightarrow M2,$
 $q \rightarrow \text{Bob},$
 $u \rightarrow \text{id-bank-robber},$
 $t \rightarrow \text{date-of-treatment}$
 $\}$
 $\{ m' \rightarrow M1 \}$

Log

Aug 15, 2014

$\text{state}(\text{Bob}, M1)$

Sept 17, 2014

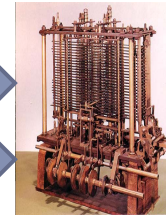
$\text{send}(\text{UPMC}, \text{allegeny-police}, M2)$
 $\text{tagged}(M2, \text{Bob}, \text{date-of-treatment},$
 $\text{id-bank-robber})$

$\varphi' = \top$

$\wedge \text{purp_in}(\text{id-bank-robber}, \text{id-criminal})$

$\wedge \text{is-admission-of-crime}(M1)$

$\wedge \text{believes-crime-caused-serious-harm}(\text{UPMC}, M1)$



Correctness of Reduce

Theorem 3.2 (Partial correctness of reduce). *If $\text{reduce}(\mathcal{L}, \varphi) = \psi$ and $\mathcal{L} \leq \mathcal{L}'$, then (1) $\mathcal{L}' \models \varphi$ iff $\mathcal{L}' \models \psi$ and (2) $\mathcal{L}' \models \bar{\varphi}$ iff $\mathcal{L}' \models \bar{\psi}$.*

Implementation and Case Study

- ▶ Implementation and evaluation over simulated audit logs for compliance with *all* 84 disclosure-related clauses of HIPAA Privacy Rule
- ▶ Performance:
 - ▶ Average time for checking compliance of each disclosure of protected health information is 0.12s for a 15MB log
- ▶ Mechanical enforcement:
 - ▶ reduce can automatically check 80% of all the atomic predicates

Ongoing Transition Efforts

- ▶ Integration of reduce algorithm into Illinois Health Information Exchange prototype
 - ▶ Joint work with UIUC and Illinois HLN
- ▶ Auditing logs for policy compliance
 - ▶ Ongoing conversations with Symantec Research

Applications of Reduce

- ▶ Audit to detect violations of policy or demonstrate compliance
- ▶ Provide explanations for violations (e.g., which clause of HIPAA was violated)
- ▶ Help train employees about privacy laws (e.g., check whether a certain type of disclosure is permitted by HIPAA)

Learning Outcomes for You

- ▶ Translate privacy laws into first-order logic for use by reduce
- ▶ Use reduce tool to check logs for compliance with laws
- ▶ Use reduce to check whether certain types of disclosures are permitted by a privacy law

Homework I will make you work through these problems
Possible project around other privacy laws such as FERPA,
COPPA

Privacy Specification Languages

- P3P[Cranor et al.], XACML[OASIS], EPAL[Backes et al.]:
Less expressive (no temporal ops,..)
- *Logic of Privacy and Utility* [Barth et al]:
Related specification logic;
enforcement only for propositional
fragment

Related Work

Logical Specification of Privacy Laws

Smaller fragments of laws

- *Logic of Privacy and Utility* [Barth et al.]: Example clauses from HIPAA and GLBA
- PrivacyAPIs [Gunter et al.]: HIPAA 164.506
- Datalog HIPAA [Lam et al.]: HIPAA 164.502, 164.506, 164.510

Related Work

Runtime monitoring in MFOTL

[Basin et al '10]

- Pre-emptive enforcement
- Efficient implementation
- Assumes past-completeness of logs
- Less expressive mode checking (“safe-range check”)
- Cannot express HIPAA or GLBA

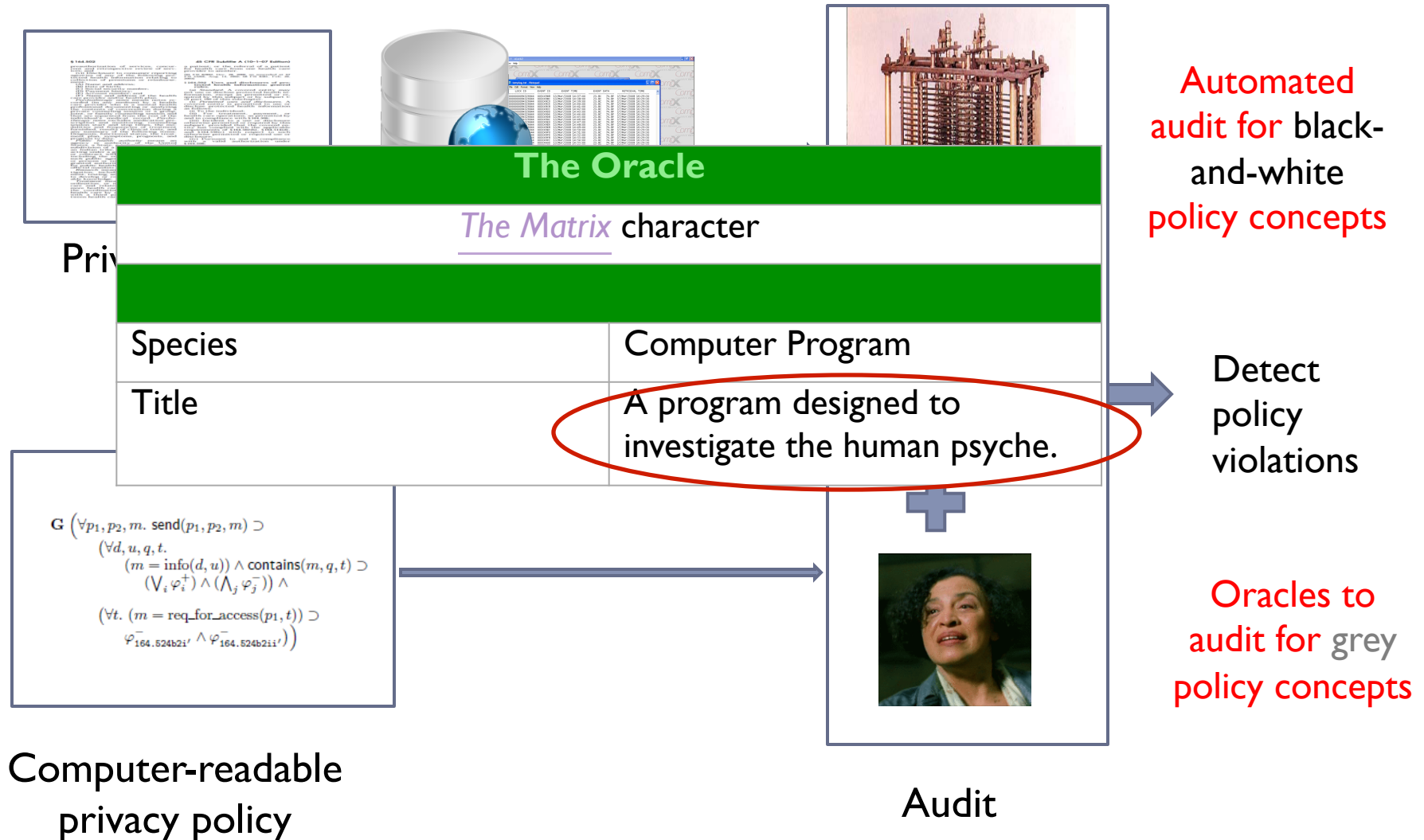
Related Work

Industry practice

Fairwarning Audit Tool

- Customized SQL queries over access logs
- Queries not tied to policy clauses

Detecting Policy Violations



Thanks!

More Technical Details

Definition of $\widehat{\text{sat}}$

Assume: The function $\text{sat}(L, P)$ computes all substitutions σ for variables in P such that $L \models P\sigma$, if certain argument positions in P are ground.

$$\begin{aligned}\widehat{\text{sat}}(L, p_O(t_1, \dots, t_n)) &= \text{sat}(L, p_O(t_1, \dots, t_n)) \\ \widehat{\text{sat}}(L, \top) &= \{\bullet\} \\ \widehat{\text{sat}}(L, \perp) &= \{\} \\ \widehat{\text{sat}}(L, c_1 \wedge c_2) &= \bigcup_{\sigma \in \widehat{\text{sat}}(L, c_1)} \sigma + \widehat{\text{sat}}(L, c_2\sigma) \\ \widehat{\text{sat}}(L, c_1 \vee c_2) &= \widehat{\text{sat}}(L, c_1) \cup \widehat{\text{sat}}(L, c_2) \\ \widehat{\text{sat}}(L, \exists x.c) &= \widehat{\text{sat}}(L, c) \setminus \{x\} \quad (x \text{ fresh})\end{aligned}$$

Mode Analysis: Idea

- ▶ Example 1: $\text{address}(x, y, a) = x + y < a$
- ▶ Key idea: If input positions are grounded, then only finite number of satisfying substitutions for output positions.
- ▶ Example 1 moding: $\text{address}(+, -, +)$
- ▶ Example 2: $\theta = \text{send}(p1, p2, m) \wedge \text{tagged}(m, q, t, u)$
- ▶ $\text{send}(-, -, -)$: all positions are output mode
- ▶ $\text{tagged}(+, -, -, -)$: message position is input mode

Mode Analysis: Predicates

$$\boxed{\chi_I \vdash e : \chi_O}$$

1. $\{\} \vdash \text{send}(p_1, p_2, m) : \{p_1, p_2, m\}$
2. $\{p_1, p_2, m\} \vdash \text{tagged}(m, q, t, u) : \{p_1, p_2, m, q, t, u\}$

$$\frac{\forall k \in I(p_O). \text{fv}(t_k) \subseteq \chi_I \quad \chi_O = \chi_I \cup \left(\bigcup_{j \in O(p_O)} \text{fv}(t_j) \right)}{\chi_I \vdash p_O(t_1, \dots, t_n) : \chi_O}$$

Mode Analysis: Conjunction

1. $\{\} \vdash \text{send}(p1, p2, m): \{p1, p2, m\}$
2. $\{p1, p2, m\} \vdash \text{tagged}(m, q, t, u): \{p1, p2, m, q, t, u\}$
3. $\{\} \vdash \text{send}(p1, p2, m) \wedge \text{tagged}(m, q, t, u): \{p1, p2, m, q, t, u\}$

$$\frac{\chi_I \vdash e_1 : \chi \quad \chi \vdash e_2 : \chi_O}{\chi_I \vdash e_1 \wedge e_2 : \chi_O}$$

Mode Analysis and $\widehat{\text{sat}}$

Example: $\theta = \text{send}(p1, p2, m) \wedge \text{tagged}(m, q, t, u)$

- ▶ $\text{send}(-,-,-)$: all positions are output mode
- ▶ $\text{tagged}(+,-,-,-)$: message position is input mode
- ▶ $\widehat{\text{sat}}(\theta) = \text{sat}(\text{send}(p1, p2, m)) + \text{sat}(\text{tagged}(m, q, t, u) \sigma)$



{ p1 → UPMC,
p2 → allegeny-police,
m → M2,
q → Bob,
u → id-bank-robber,
t → date-of-treatment
}

Log	
Jan 1, 2011	state(Bob, M1)
Jan 5, 2011	send(UPMC, allegeny-police, M2) tagged(M2, Bob, date-of-treatment, id-bank-robber)

Mode Analysis: Termination of $\widehat{\text{sat}}$

General Theorem: If initial policy passes a syntactic **mode check**, then finite substitutions can be computed

Applications: The entire HIPAA and GLBA Privacy Rules pass this check