Bootstrapping Privacy Compliance in Big Data Systems

Anupam Datta
CMU

Fall 2016
Privacy Compliance for Bing

**Setting:**

- Auditor has access to source code
The Privacy Compliance Challenge

Legal Team
- Crafts Policy

Privacy Champion
- Interprets Policy

Developer
- Writes Code

Audit Team
- Verifies Compliance

English Privacy Policy

Compliant?

 Millions of Lines of Undocumented Code
A Streamlined Audit Workflow

Legal Team
Crafts Policy
Interprets Policy
Develop Annotations

Code analysis
Annotated Code
Update Grok
Fix code
Potential violations

Grok
Data inventory with policy labels

Legalease
A formal policy specification language

Audit Team
Verifies Compliance

Encode
Refine
Update Grok
Fix code
Legalease Policy
A Streamlined Audit Workflow

Workflow for privacy compliance

Legalease, usable yet formal policy specification language

Grok, bootstrapped data inventory for big data systems

Scalable implementation for Bing

Legal Team
Crafts Policy

Encode

Legalease
A formal policy specification language

Grok
Data inventory with policy datatypes

Code analysis, developer annotations

Developer
Writes Code

Fix code

Audit Team
Verifies Compliance

Privacy champion
Interprets Policy

Update Grok
Potential violations
Privacy as Restrictions on Personal Information Flow

- Direct
- Interference
- Probabilistic Interference
- Differential Privacy

Restrictions

- Purpose & Role based
- Temporal

Direct Interference

- EPAL
- XACML
- Prototype Access Control

Interference

- Purpose → Planning
- [Hayati & Abadi]
- Grok + Legalease

Probabilistic Interference

- Information Flow Experiments

Differential Privacy

- Differential Privacy

Temporal

- FOTLs
  - [Formal Contextual Integrity, Reduce audit algorithm, Basin et al.]
A Streamlined Audit Workflow

**Legal Team**
Crafts Policy

**Privacy Champion**
Interprets Policy

---

**Legalease**
A formal policy specification language

**Grok**
Data inventory with policy datatypes

**Checker**
Annotated Code
Legalease Policy
Potential violations

---

**Developer**
Writes Code

**Audit Team**
Verifies Compliance

Fix code
Update Grok
Annotated Code

---

Encode
Refine
## Specification: Legalease

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Usable by lawyers and privacy champs.</td>
<td>Expressive enough for real-world policies.</td>
<td>Precise semantics for local reasoning.</td>
</tr>
</tbody>
</table>
Legalease : Syntax

Policy Clause $C$ ::= $D \mid A$
Deny Clause $D$ ::= \text{DENY } T_1 \cdots T_n \text{ EXCEPT } A_1 \cdots A_m$
\quad \mid \text{DENY } T_1 \cdots T_n$
Allow Clause $A$ ::= \text{ALLOW } T_1 \cdots T_n \text{ EXCEPT } D_1 \cdots D_m$
\quad \mid \text{ALLOW } T_1 \cdots T_n$
Attribute $T$ ::= $\langle \text{attribute-name} \rangle \; v_1 \cdots v_l$
Value $v$ ::= $\langle \text{attribute-value} \rangle$
DENY  Datatype  IPAddress
   UseForPurpose Advertising

We will not use full IP Address for Advertising.
We will not use full IP Address for Advertising. IP Address may be used for detecting abuse. In such cases, it will not be combined with account information.
Designed for Usability

Exceptions

How legal texts are structured
One-to-one correspondence

Local Reasoning

Each exception refines its immediate parent
Formally proven property

H. DeYoung, D. Garg, L. Jia, D. Kaynar, and A. Datta, “Experiences in the logical specification of the HIPAA and GLBA privacy laws”
We will not use full IP Address for Advertising. IP Address may be used for detecting abuse. In such cases, it will not be combined with account information.
A Lattice of Policy Labels

- If “IPAddress” use is allowed then so is everything below it
- If “IPAddress:Truncated” use is denied then so is everything above it
### Designed for Precision

**Table I**

**Grammar for Legalease**

- **Policy Clause** $C ::= D | A$
- **Deny Clause** $D ::= \text{DENY } T_1 \cdots T_n \ \text{EXCEPT } A_1 \cdots A_m$
- **Allow Clause** $A ::= \text{ALLOW } T_1 \cdots T_n \ \text{EXCEPT } D_1 \cdots D_m$
- **Attribute** $T ::= \langle \text{attribute-name} \rangle \ v_1 \cdots v_l$
- **Value** $v ::= \langle \text{attribute-value} \rangle$

**Table III**

**Inference Rules for Legalease**

\[
\begin{align*}
T^G \not\subseteq T^C & \quad \text{ALLOWS } T^C \ \text{EXCEPT } D_1 \cdots D_m \text{ allows } T^G & (A_1) \\
T^G \subseteq T^C & \quad \exists_i D_i \text{ denies } T^G \\
T^G \subseteq T^C & \quad \forall_i D_i \text{ allows } T^G & (A_3) \\
\bot \in T^G \cap T^C & \quad \text{DENY } T^C \ \text{EXCEPT } A_1 \cdots A_m \text{ allows } T^G & (D_1) \\
\bot \not\in T^G \cap T^C & \quad \exists_i A_i \text{ allows } T^G \cap T^C & (D_2) \\
\bot \not\in T^G \cap T^C & \quad \forall_i A_i \text{ denies } T^G \cap T^C & (D_3)
\end{align*}
\]
ALLOW

EXCEPT

DENY $dataType$ IPAddress:Expired
DENY $dataType$ UniqueIdentifier:Expired
DENY $dataType$ SearchQuery, PII InStore Store
DENY $dataType$ UniqueIdentifier, PII InStore Store

DENY $dataType$ BBEPData UseForPurpose Advertising

DENY $dataType$ BBEPData, PII InStore Store

DENY $dataType$ BBEPData:Expired

DENY $dataType$ UserProfile, PII InStore Store

DENY $dataType$ PII UseForPurpose Advertising
DENY $dataType$ PII InStore AdStore

DENY $dataType$ SearchQuery UseForPurpose Sharing

EXCEPT

ALLOW $dataType$ SearchQuery:Scrubbed

“we remove the entirety of the IP address after 6 months”
“[we remove] cookies and other cross session identifiers, after 18 months”
“We store search terms (and the cookie IDs associated with search terms) separately from any account information that directly identifies the user, such as name, e-mail address, or phone numbers.”
“We do not use any of the information collected through the Bing Bar Experience Improvement Program to identify, contact or target advertising to you”
“We take steps to store [information collected through the Bing Bar Experience Improvement Program] separately from any account information we may have that directly identifies you, such as name, e-mail address, or phone numbers”
“We delete the information collected through the Bing Bar Experience Program at eighteen months.”
“We store page views, clicks and search terms used for ad targeting separately from contact information you may have provided or other data that directly identifies you (such as your name, e-mail address, etc.).”
“Our advertising systems do not contain or use any information that can personally and directly identify you (such as your name, email address and phone number).”
“Before we [share some search query data], we remove all unique identifiers such as IP addresses and cookie IDs from the data.”
Designed for Expressivity (Google, October 2013)

ALLOW
EXCEPT
DENY Data Type PII UseForPurpose Sharing

EXCEPT
ALLOW Data Type PII: OptIn
EXCEPT
ALLOW Access By Role Affiliates
EXCEPT
ALLOW Use For Purpose Legal

DENY Data Type DoubleClick Data, PII
EXCEPT
ALLOW Data Type DoubleClick Data, PII: OptIn

“We do not share personal information with companies, organizations and individuals outside of Google unless one of the following circumstances apply:”
- “We require opt-in consent for the sharing of any sensitive personal information.”
- “We provide personal information to our affiliates or other trusted businesses or persons to process it for us”
- “We will share personal information [if necessary to] meet any applicable law, regulation, legal process or enforceable governmental request.”
- “We will not combine DoubleClick cookie information with personally identifiable information unless we have your opt-in consent”
Legalease Usability

Survey taken by 12 policy authors within Microsoft
Encode Bing data usage policy after a brief tutorial

Time spent
2.4 mins on the tutorial
14.3 mins on encoding policy

High overall correctness
A Streamlined Audit Workflow

Legal Team
Crafts Policy

Privacy Champion
Interprets Policy

Encode

Refine

Legalease
A formal policy specification language

Grok
Data inventory with policy labels

Checker
Annotated Code
Update Grok
Legalease Policy
Potential violations

Developer
Writes Code

Audit Team
Verifies Compliance

Fix code

Developer annotations
Code analysis
A Streamlined Audit Workflow

Legal Team
Crafts Policy

Privacy Champion
Interprets Policy

Encode

Refine

Legalease
A formal policy specification language

Grok
Data inventory with policy labels

Code analysis, developer annotations

Update Grok

Potential violations

Annotated Code

Legalease Policy

Checker

Fix code

Audit Team
Verifies Compliance

Developer
Writes Code
Map-Reduce Programming Systems

Scope, Hive, Dremel
Data in the form of Tables

Code Transforms Columns to Columns
No Shared State
Limited Hidden Flows

users =
    SELECT _name, _age FROM datasetAB
user_tag =
    SELECT GenerateTag(_name, _age)
    FROM users
OUTPUT user_tag TO datasetC
Grok
Grok

Purpose Labels
Annotate programs with purpose labels
**Grok**

**Purpose Labels**
Annotate programs with purpose labels

**Initial Data Labels**
Heuristics and Annotations

```sql
users = SELECT Name, Age FROM datasetAB
user_tag = SELECT GenerateTag(_name, _age) FROM users
OUTPUT user_tag TO datasetC
```

```
users =
  SELECT Name, Age FROM datasetAB
user_tag =
  SELECT GenerateTag(_name, _age)
  FROM users
OUTPUT user_tag TO datasetC
```
Grok

Purpose Labels
Annotate programs with purpose labels

Initial Data Labels
Heuristics and Annotations

Flow Labels
Source labels propagated via data flow graph

D. E. Denning. “A lattice model of secure information flow”
A Lattice of Policy Labels

- If “Profile” use is allowed then so is everything below it
- If “Name” use is denied then so is everything above it
Implicit flows

Beyond direct flows discussed in healthcare audit examples
Map-Reduce

Map
Operate on rows in parallel
eg. filtering

Reduce
Combine groups of rows
eg. aggregation

users =
SELECT Name, Age FROM datasetAB

users_35 =
SELECT _name, _age
FROM users
WHERE (_age > 35)

ages_35 =
SELECT _age, COUNT(_name) AS Profile
FROM users_35
GROUP BY _age

OUTPUT ages_35 TO datasetC
Combine Noisy Sources

Carefully curated regular expressions

Leverages developer conventions

Significant Noise

Variable Name Analysis

Expense

Low Noise

Developer Annotations

Very Expensive

Definitive

Need very few of these

Auditor Verification
Why Bootstrapping Grok Works

Pick the nodes which will label the most of the graph

~200 annotations label 60% of nodes

A small number of annotations is enough to get off the ground.
77,000 jobs run each day
- By 7000 entities
- 300 functional groups
- 1.1 million unique lines of code
- 21% changes on avg, daily
- 46 million table schemas
- 32 million files
- Manual audit infeasible
- Information flow analysis takes ~30 mins daily
### Nightly Compliance Process

**Static code analysis**
- Schemas: 23M+

**Generate report**
- Positive Patterns: (40 Taxonomy values, 400 patterns)
- Negative Patterns: (2500 total entries)
- Granular Overrides: (116 total entries)

**Manual Audit**
- **files**: 32
- **schems**: 23M+
- **teams**: 8

**SQL Query**
```sql
SELECT *
FROM (SELECT *
      FROM Report
      WHERE Taxonomy='ANID'
      AND Confidence>='High') AS ID
INNER JOIN (SELECT *
             FROM Report
             WHERE TaxonomyGroup='PII'
             AND Confidence>='High') AS P
ON ID.VC = P.VC
```
A Streamlined Audit Workflow

- Legal Team: Crafts Policy
  - Encode
  - Legalease: A formal policy specification language
- Privacy Champion: Interprets Policy
  - Refine
- Grok: Data inventory with policy labels
  - Code analysis
  - Developer annotations
  - Annotated Code
- Checker: Legalease Policy
  - Update Grok
- Developer: Writes Code
  - Fix code
- Audit Team: Verifies Compliance
  - Potential violations
A Streamlined Audit Workflow

**Workflow** for privacy compliance

**Legalease**, usable yet formal policy specification language

**Grok**, bootstrapped data inventory for big data systems

**Scalable** implementation for Bing
Reference

Policy Labels: Datatypes

Going down within a lattice:
finer specification

TypeState specifies limited temporal properties
Policy Types: Concept Lattices

InStore Lattice

UseForPurpose Lattice

AccessByRole Lattice
Formal Semantics

\[
\begin{align*}
& T^G \sqsubseteq T^C \\
& \exists_i D_i \text{ denies } T^G \\
& \text{ALLOW } T^C \text{ EXCEPT } D_1 \cdots D_m \text{ denies } T^G
\end{align*}
\] (A_2)

Based on Lattice Orderings on Policy Types
Formal Semantics

\[ T^G \subseteq T^C \quad \exists_i D_i \text{ denies } T^G \]

ALLOW \[ T^C \] EXCEPT \[ D_1 \cdots D_m \] denies \[ T^G \] (A_2)

Recursively check exceptions
ALLOW clauses have DENY clauses as exceptions
Top Level clause determines Blacklist/Whitelist