Understanding Privacy Violations

<table>
<thead>
<tr>
<th></th>
<th>No inference</th>
<th>Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User</strong></td>
<td>(i) OSN showing the user’s media without consent or user wrongly configuring privacy constraints</td>
<td>(iii) Identifying user’s location from a geotag in the pictures</td>
</tr>
<tr>
<td><strong>Others</strong></td>
<td>(ii) Friend tags the user and makes the picture public where the user did not want to be seen</td>
<td>(iv) Friend tags the user revealing friendship status even when the user had hid her friend list</td>
</tr>
</tbody>
</table>

We have conducted an online survey with 330 participants. More than 96% of the participants face privacy violations that occur through inferences.
Why does a Violation Take Place?

- The privacy policy might not have been applied correctly.
- Multiple privacy policies might have been conflicting
- The privacy policy might not have been defined properly (e.g., in detail)
Commitment $C_{id}$

Violation Statement $v_id$

$\text{Domain} \rightarrow \text{State} \rightarrow \text{Norms}$

$v_id$ holds?

$\text{no}$

$C_{id}$ is fulfilled

$\text{yes}$

$C_{id}$ is violated
PriGuard

Commitment $C_{id}$

Violation Statement $v_{id}$

Ontology

State $v_{id}$ holds?

Norms

yes

$C_{id}$ is violated

no

$C_{id}$ is fulfilled
ABS N Ontology

- OSN domain concepts; e.g., Agent, Post
- Relation properties; isFriendOf, isColleagueOf
- Privacy properties; e.g., canSeePost
PriGuard

Commitment $C_id$

Violation Statement $v_id$

Ontology

State

Norms

$v_id$ holds?

yes

$C_id$ is violated

no

$C_id$ is fulfilled
PriGuard

Commitment $C_{id}$

Violation Statement $v_{id}$

Ontology

Assertions

Norms

$v_{id}$ holds?

no

$C_{id}$ is fulfilled

yes

$C_{id}$ is violated
ABS N State

ABS N state is a set of instances from this ontology.

**Table**: Charlie shares a post :pc1

<table>
<thead>
<tr>
<th>ClassAssertion(Agent :alice)</th>
<th>ClassAssertion(Agent :bob)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassAssertion(Agent :charlie)</td>
<td>ClassAssertion(Agent :dennis)</td>
</tr>
<tr>
<td>ClassAssertion(Agent :eve)</td>
<td>ClassAssertion(Audience :audience)</td>
</tr>
<tr>
<td>ClassAssertion(Post :pc1)</td>
<td>ClassAssertion(Picture :pictureConcert)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ObjectPropertyAssertion(isFriendOf :alice :bob)</th>
<th>ObjectPropertyAssertion(isFriendOf :alice :charlie)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ObjectPropertyAssertion(isFriendOf :bob :charlie)</td>
<td>ObjectPropertyAssertion(isFriendOf :charlie :dennis)</td>
</tr>
<tr>
<td>ObjectPropertyAssertion(isFriendOf :dennis :eve)</td>
<td></td>
</tr>
<tr>
<td>ObjectPropertyAssertion(sharesPost :charlie :pc1)</td>
<td>ObjectPropertyAssertion(hasAudience :pc1 :audience)</td>
</tr>
<tr>
<td>ObjectPropertyAssertion(hasMedium :pc1 :pictureConcert)</td>
<td>ObjectPropertyAssertion(taggedPerson :pictureConcert :alice)</td>
</tr>
<tr>
<td>ObjectPropertyAssertion(hasMember :audience :alice)</td>
<td>ObjectPropertyAssertion(hasMember :audience :dennis)</td>
</tr>
<tr>
<td>ObjectPropertyAssertion(hasMember :audience :eve)</td>
<td>ObjectPropertyAssertion(hasMember :audience :bob)</td>
</tr>
</tbody>
</table>
PriGuard

Commitment $C_{id}$

Violation Statement $v_{id}$

Ontology → Assertions → Norms

$v_{id}$ holds?

no → $C_{id}$ is fulfilled

yes → $C_{id}$ is violated
PriGuard

Commitment $C_i$

Violation Statement $V_i$

Ontology

Assertions

DL Rules

$v_i$ holds?

Yes $C_i$ is violated

No $C_i$ is fulfilled
Norms are represented with DL Rules.

\[(n_1)\] \(\text{sharesPost} \sqsubseteq \text{canSeePost}\)

\[(n_2)\] \(\text{hasMember} \circ \text{hasAudience} \circ R_{\text{sharedPost}} \sqsubseteq \text{canSeePost}\)

\[(n_3)\] \(\text{includesPerson} \circ R_{\text{sharedPost}} \sqsubseteq \text{sharesPost}\)

\[(n_4)\] \(\text{hasLocation} \circ \text{withPerson} \sqsubseteq \text{includesPerson}\)

\[(n_5)\] \(\text{hasMedium} \circ \text{taggedPerson} \sqsubseteq \text{includesPerson}\)

\[(n_6)\] \(\text{hasText} \circ \text{mentionedPerson} \sqsubseteq \text{includesPerson}\)

\[(n_7)\] \(\text{Post} \sqcap \exists \text{hasMedium}.\exists \text{hasGeotag}.\text{Location} \sqsubseteq \text{LocationPost}\)

\[(n_8)\] \(\text{LocationActivity} \sqsubseteq \text{FriendshipActivity}\)
User’s Privacy Constraints → Commitment $C_{id}$ → Violation Statement $v_{id}$ →

- Domain
- State
- Norms

$v_{id}$ holds?

- no $C_{id}$ is fulfilled
- yes $C_{id}$ is violated
Specification of a Privacy Agreement

[Diagram of Privacy Check interface]

Who can see?
- Everyone
- Friends
- Not friends
- Friends of friends
- Not friends of friends
- Colleagues
- Not colleagues
- Only me

Specific people
- Protos: 'eve'
- Protos: 'charlie'

Who cannot see?
- Everyone
- Friends
- Not friends
- Friends of friends
- Not friends of friends
- Colleagues
- Not colleagues
Commitments are a powerful representation for modeling multiagent interactions.

Here used to represent the privacy agreement between a user and the OSN.

A commitment is denoted as a four-place relation:

\[ C(\text{debtor}; \text{creditor}; \text{antecedent}; \text{consequent}) \]

\[ C_1(:\text{osn}; :\text{dennis}; \text{isFriendOf}(\text{dennis},X), \text{sharesPost}(\text{dennis},P), \text{MediumPost}(P); \text{canSeePost}(X,P)) \]

Friends of Dennis can see medium posts of Dennis

\[ C_2(:\text{osn}; :\text{dennis}; \text{isFriendOf}(\text{dennis},X), \text{sharesPost}(\text{dennis},P), \text{LocationPost}(P); \text{not(canSeePost}(X,P))) \]

Friends of Dennis cannot see location posts of Dennis
Generation of Commitments

A user specifies:
- neither \textit{canSeeGroup} nor \textit{cannotSeeGroup} (no commitment)
Generation of Commitments

A user specifies:

- neither `canSeeGroup` nor `cannotSeeGroup` (no commitment)
- `canSeeGroup` OR `cannotSeeGroup` (one commitment)
A user specifies:

- neither *canSeeGroup* nor *cannotSeeGroup* (no commitment)
- *canSeeGroup* OR *cannotSeeGroup* (one commitment)
Generation of Commitments

A user specifies:

- neither `canSeeGroup` nor `cannotSeeGroup` (no commitment)
- `canSeeGroup` OR `cannotSeeGroup` (one commitment)
- `canSeeGroup` AND `cannotSeeGroup` (two commitments)
Generation of Commitments

A user specifies:

- neither \textit{canSeeGroup} nor \textit{cannotSeeGroup} (no commitment)
- \textit{canSeeGroup} OR \textit{cannotSeeGroup} (one commitment)
- \textit{canSeeGroup} AND \textit{cannotSeeGroup} (two commitments)
PriGuard

User’s Privacy Constraints

Commitment $C_{id}$

Violation Statement $v_{id}$

Ontology

Assertions

DL Rules

$v_{id}$ holds?

no

C_{id} is fulfilled

yes

C_{id} is violated
PriGuard

User’s Privacy Constraints → Commitment $C_{id}$ → RDF Statement $V_{id}$

Ontology → Assertions → DL Rules

$V_{id} \text{ holds?}$

- yes, $C_{id}$ is violated
- no, $C_{id}$ is fulfilled
Violation Statements

- We identify violation statements according to the commitments.
- A violation statement is modeled as: 
  \[ \text{antecedent, not( consequent)} \]
- We represent the violation as a SPARQL query and check if it holds in the current state of the social network.
A Detection Example

Dennis wants his friends to see his pictures but not his location. He posts a picture but hides the location. However, Dennis is not aware that his picture is geotagged.

\[
\begin{align*}
C_1(\text{:osn}; \text{:dennis}; & \text{isFriendOf}(\text{:dennis}, X), \text{sharesPost}(\text{:dennis}, P), \text{MediumPost}(P); \text{canSeePost}(X, P)) \\
v_1: & \text{isFriendOf}(\text{:dennis}, X), \text{sharesPost}(\text{:dennis}, P), \text{MediumPost}(P), \text{not(canSeePost}(X, P)) \\
C_2(\text{:osn}; \text{:dennis}; & \text{isFriendOf}(\text{:dennis}, X), \text{sharesPost}(\text{:dennis}, P), \text{LocationPost}(P); \text{not(canSeePost}(X, P)) \\
v_2: & \text{isFriendOf}(\text{:dennis}, X), \text{sharesPost}(\text{:dennis}, P), \text{LocationPost}(P), \text{canSeePost}(X, P)
\end{align*}
\]
### PriGuard: Performance Results

<table>
<thead>
<tr>
<th></th>
<th>ABSN</th>
<th>depth=0</th>
<th>depth=1</th>
<th>depth=2</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(♯Ą,♯R)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$G_1$: #Axioms</td>
<td>2175</td>
<td>3ms</td>
<td>(39,412)</td>
<td>(535,5347)</td>
<td>(535,5347)</td>
</tr>
<tr>
<td>Time</td>
<td></td>
<td></td>
<td>4267</td>
<td>29959</td>
<td>29959</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.74ms</td>
<td>30.19ms</td>
<td>29.79ms</td>
</tr>
<tr>
<td></td>
<td>(♯Ą,♯R)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$G_2$: #Axioms</td>
<td>2175</td>
<td>2.96ms</td>
<td>(51,579)</td>
<td>(1035,27783)</td>
<td>(1035,27783)</td>
</tr>
<tr>
<td>Time</td>
<td></td>
<td></td>
<td>5079</td>
<td>125703</td>
<td>125703</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>5.49ms</td>
<td>123.95ms</td>
<td>122.46ms</td>
</tr>
<tr>
<td></td>
<td>(♯Ą,♯R)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$G_3$: #Axioms</td>
<td>2175</td>
<td>3.09ms</td>
<td>(123,4199)</td>
<td>(1046,27795)</td>
<td>(4039,88234)</td>
</tr>
<tr>
<td>Time</td>
<td></td>
<td></td>
<td>20423</td>
<td>125883</td>
<td>403555</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>18.01ms</td>
<td>121.15ms</td>
<td>530.01ms</td>
</tr>
<tr>
<td></td>
<td>(♯Ą,♯R)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$G_4$: #Axioms</td>
<td>2175</td>
<td>3.07ms</td>
<td>(37,235)</td>
<td>(848,8543)</td>
<td>(60001,728596)</td>
</tr>
<tr>
<td>Time</td>
<td></td>
<td></td>
<td>3535</td>
<td>46463</td>
<td>3636547</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>4.13ms</td>
<td>47.09ms</td>
<td>18397.26ms</td>
</tr>
<tr>
<td></td>
<td>(♯Ą,♯R)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$G_5$: #Axioms</td>
<td>2175</td>
<td>3.11ms</td>
<td>(157,2669)</td>
<td>(2787,74217)</td>
<td>(65328,1435168)</td>
</tr>
<tr>
<td>Time</td>
<td></td>
<td></td>
<td>14711</td>
<td>332463</td>
<td>6526759</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>19.03ms</td>
<td>406.91ms</td>
<td>25890.27ms</td>
</tr>
</tbody>
</table>
Model Checking

- Rather than checking a single state check the model
- Automatic verification of properties based on the model of a system
- The system is represented as a state transition graph
- The property is represented as a logic formula
- Model checker verifies whether the property holds for the system
- Example:
  - System: the social network (with entire details of relations, agreements, and so on)
  - Property: location of user $X$ is accessible by user $Y$
  - Model checking will say whether this property holds