**IconIntent**: Automatic Identification of Sensitive UI Widgets based on Icon Classification for Android Apps

Xusheng Xiao¹, Xiaoyin Wang², Zhihao Cao¹, Hanlin Wang¹, and Peng Gao³

¹Case Western Reserve University  
²The University of Texas at San Antonio  
³Princeton University
Privacy Issues of Mobile App

- Mobile apps have become an integral part of our life
  - E.g., business, transportation, education
- Many apps access sensitive data, raising privacy concerns
  - E.g., location, contacts, microphone
Sensitive UI Widgets

- It is crucial to understand the apps’ intentions in using the sensitive information
  - E.g., Inspecting permissions and informing users about sensitive data
- Apps express their intentions to use or collect users’ sensitive data via sensitive UI widgets, i.e., justifying the uses of the data
- App market needs an automatic approach to understand these intentions
Challenges: Understanding Intentions of UI Widgets

• UI widgets’ intentions are expressed via texts and images
  – Prior works focus on analyzing framework APIs (e.g., device identifiers, and contacts) or descriptive texts (e.g., text labels), not images

• Object icons: icons with specific shapes, no co-located texts
  – Different styles, scales, angles

• Text icons: icons embedded with texts
  – Diversified colors and opacities

Automatic Identification of Sensitive UI Widgets with Icons
Sensitive UI Widget Identification

• Given an app, which UI widgets are associated with icons?

• Based on the icons, which sensitive data the UI widgets will use?
IconIntent

• Synergistically combine *program analysis* and *icon classification*

• Associate icons with UI widgets via static analysis

• Classify the intentions of icons (both object icons and text icons) into eight pre-defined sensitive user input categories
  – Including Camera, Contacts, Email, Location, Phone, Photo, SMS, and Microphone
Overview of IconIntent

- Icon-Widget Association: static analysis on UI layout files and code
- Icon Mutation: image mutations on extracted icons
- Icon Classification: classification of icons into sensitive categories
Icon-Widget Association: UI Layout

- Static Analysis: XML parsing and resource resolution

- UI layouts: widgets and icons

```xml
<LinearLayout android:orientation="horizontal">
    <ImageView android:id="@+id/img" android:src="@drawable/loc" />
    <EditText android:id="@+id/TxtCity" />
    <Button android:text="@string/search" />
</LinearLayout>
```

- Drawable objects

```xml
<selector>
    <item android:state_checked="true"
          android:drawable="@drawable/btn_radio_to_on_mtrl_015" />
    <item android:drawable="@drawable/btn_radio_to_on_mtrl_000" />
</selector>
```
Icon-Widget Association: API Calls

- Life cycle methods: load layout files, bind variables to UI widgets, and associate icons to UI widgets

```java
void onCreate(Bundle savedInstanceState) {
    View g = this.findViewById(R.id.button_esc); // FindView ImageView
    h = (ImageView) g; // cast to ImageView
    h.setImageResource(R.drawable.icon2); // associate icon
    ...
}
```

- Static analysis: dataflow analysis with over-approximations to associate UI widgets and icons

Widget ID set: $\Gamma(h) = \{\text{R.id.button_esc}\}$  
Icon ID set: $\Sigma(h) = \{\text{R.drawable.icon2}\}$,
App Icon Varieties

• Icons have different combinations of colors and transparencies in texts, backgrounds, and object shapes

• Challenges for computer vision techniques
  – Small: (a) and (b)
  – Low contrast: (c) and (d)
  – Bright color text and dark background: (e)
  – Opacity: (f) and (g)
Icon Mutation

- RGBA model \(<R,G,B,A>\) to represent an icon
  - \(R,G,B\) for red, green, blue, \(A\) for opacity
- Image mutations:
  - Image Scaling: enlarge pixel values using nearby pixels
  - Grayscale Conversion: convert an image to represent only the amount of light
  - Color Inversion: invert the colors of each pixel
  - Contrast Adjustment: adjust the contrast of colors in the image
  - Opacity Conversion: convert the opacity differences to the color differences
Icon Classification – Object Icon

• Object recognition to classify object icons based on a training icon set labeled with sensitive user-input categories

• Scale-Invariant-Feature-Transform (SIFT)
  – Identifying key locations that are invariant with respect to image translation, scaling, and rotation and matching key locations
  – Challenges:
    • Too few key locations
      – Enlarging icons and FAST
    • Lower tolerance for changes
      – Relative One-to-One Mapping
Icon Classification – Text Icon

• Optical Character Recognition (OCR)
  – Working well for dark text and bright background
  – Still not perfect even with image mutations
    • E.g., location -> Ication or llocation, email -> emai

• Classification based on Keyword Similarity
  – 95+% of 300 text icons extracted from top Google Play apps containing 1 to 3 words
  – Edit distance-based similarity (considering keyword length)
    • $Sim_{w,k} = 1 - \frac{Edit\ Distance}{length(k)}$
Evaluation Setup

• Implementation:
  – Static analysis: Gator and Soot
  – Icon classification: OpenCV and Asprise OCR

• Subject:
  – Training dataset: 1,576 icons
    • Google image search: 800
    • Top 10,000 apps: 776
  – Test dataset: 150 apps with 5,791 icons
    • 539 sensitive object icons
    • 49 sensitive text icons
    • Total: 588 sensitive icons
Identifying Sensitive UI Widgets

- Detecting most sensitive icons (90.1%, 530 / 588) from most apps (135 out of 138 apps that contain sensitive icons)
- Prevalent sensitive UI widgets: 248 UI widgets from 97 apps (prec: 82.4%)
- Sensitive icons not always used in UI widgets
  - 125 SMS icons -> 24 UI widgets, 20 Phone icons -> 38 UI widgets

<table>
<thead>
<tr>
<th>Category</th>
<th>#Detected SIs</th>
<th>#Apps with SIs</th>
<th>#Detected SWs</th>
<th>#Apps with SWs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Object</td>
<td>Text</td>
<td>All</td>
<td></td>
</tr>
<tr>
<td>Camera</td>
<td>148</td>
<td>1</td>
<td>149</td>
<td>47</td>
</tr>
<tr>
<td>Contacts</td>
<td>14</td>
<td>1</td>
<td>15</td>
<td>6</td>
</tr>
<tr>
<td>Email</td>
<td>44</td>
<td>5</td>
<td>49</td>
<td>16</td>
</tr>
<tr>
<td>Location</td>
<td>19</td>
<td>11</td>
<td>30</td>
<td>9</td>
</tr>
<tr>
<td>Microphone</td>
<td>75</td>
<td>3</td>
<td>78</td>
<td>26</td>
</tr>
<tr>
<td>Phone</td>
<td>20</td>
<td>1</td>
<td>21</td>
<td>6</td>
</tr>
<tr>
<td>Photo</td>
<td>41</td>
<td>12</td>
<td>53</td>
<td>13</td>
</tr>
<tr>
<td>SMS</td>
<td>125</td>
<td>11</td>
<td>136</td>
<td>23</td>
</tr>
<tr>
<td>All</td>
<td>486</td>
<td>44</td>
<td>530</td>
<td>135</td>
</tr>
</tbody>
</table>
Combining with SUPOR

- **SUPOR**: text-based sensitive UI widget identification
  - Expand to include buttons, radio buttons, check boxes,
  - Leverage dex2jar to support custom widgets
- **SUPOR**: 242 SUI <-> **SUPOR+IconIntent**: 487 SUI
- Only 3 UI widgets are identified by both SUPOR and IconIntent
Icon Classification

• IconIntent achieves an average F-score of 87.7% (with distance threshold as 0.3)
• IconIntent greatly improves F-score with image mutation (from 36.6% to 89.8%)

<table>
<thead>
<tr>
<th>Setting</th>
<th>P (%)</th>
<th>R (%)</th>
<th>F (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT</td>
<td>43.0</td>
<td>54.5</td>
<td>48.1</td>
</tr>
<tr>
<td>Without Mutation</td>
<td>91.2</td>
<td>64.9</td>
<td>75.8</td>
</tr>
<tr>
<td>IconIntent</td>
<td>88.2</td>
<td>87.3</td>
<td>87.7</td>
</tr>
</tbody>
</table>

Object-Icon Classification

<table>
<thead>
<tr>
<th>Setting</th>
<th>P (%)</th>
<th>R (%)</th>
<th>F (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without Mutation</td>
<td>91.7</td>
<td>22.9</td>
<td>36.6</td>
</tr>
<tr>
<td>IconIntent</td>
<td>89.8</td>
<td>89.8</td>
<td>89.8</td>
</tr>
</tbody>
</table>

Text-Icon Classification
Conclusion

• **IconIntent**
  – Program analysis techniques to associate icons and UI widgets
  – Computer vision techniques to classify the associated icons into eight sensitive categories

• **Evaluation on 150 apps from Google Play**
  – Detect 248 sensitive UI widgets in 97 apps, achieving a precision of 82.4%
  – SUPOR +IconIntent can detect 487 sensitive UI widgets (101.2% improvement over SUPOR only)
  – Image mutations improves icon classification
Thank You!

Questions?