Privacy Policy Modelling And Analysis for Android Applications
IEEE Policy’11

Guillaume Benats$^{2}$  Arosha Bandara$^{1}$  Yijun Yu$^{1}$
Jean-Noel Colin$^{2}$  Bashar Nuseibeh$^{1}$

$^{1}$The Open University, MCS Department,
Milton Keynes, UK

$^{2}$FUNDP Computer Sciences Faculty,
Namur, Belgium

June 7th, 2011
Access Control Policies

→ Mobile applications access user information via phone’s resources
  ▶ Access control policies managed by system: permissions
  ▶ Permissions to access resources of the phone
  ▶ Geolocation, SMS, Contact List, Phone Calls,…

  ▶ Android: Either permission granted, either installation refused
  ▶ No possibilities of constraints over permissions
  ▶ Lack of granularity for managing access to resources
  ▶ Access to resources = mobile privacy management
Access Control Policies

→ Mobile applications access user information via phone’s resources
  ▶ Access control policies managed by system: permissions
  ▶ Permissions to access resources of the phone
  ▶ Geolocation, SMS, Contact List, Phone Calls, …

  ▶ Android: Either permission granted, either installation refused
  ▶ No possibilities of constraints over permissions
  ▶ Lack of granularity for managing access to resources
  ▶ Access to resources = mobile privacy management

→ Lack of granularity for privacy management
CONSTRAINTS OVER POLICIES

→ Several tools exist for adding such constraints
  ▶ On Android: *Apex* (User defined constraints over permissions) \(^1\),
  ▶ *ConUCON* (context-aware usage control mechanisms) \(^2\)
  ▶ ...
→ Those tools allow declaration of constraints over policies

Example

▶ *Facebook* application can only access Internet between 06:00PM and 08AM
▶ *AngryBirds* cannot access localisation
▶ *Twitter* application can only access localisation when inside UK

\(^1\)Khan et al., “Apex : Extending Android Permission Model and Enforcement with User-defined Runtime Constraints”

\(^2\)Bai et al., “Context-Aware Usage Control for Android”
MODELISATION OF POLICIES

- Constraints can either be defined by users, or by system
- User-centric or context-aware privacy management
- Conflicts between constraints can arise
- Tools have to ensure the satisfiability of constraints

- Toward a model based on P-RBAC
- Allows reasoning and modelling of policies
- Manage conflicts between constraints
- Based on a well-known proved access control policy (RBAC)
USING P-RBAC TO EXPRESS POLICIES

Restated mainly from \(^3\)

\(^3\) Q. Ni et al., “Privacy-aware role based access control”, SACMAT ’07
USING P-RBAC TO EXPRESS POLICIES
P-RBAC EXAMPLES

- $\text{permission}(\text{Facebook}, ((\text{access, Internet}), (time > 20 \land time < 8), \text{Log\_Access}())$

- $\text{permission}(\text{Twitter}, ((\text{access, ACCESS\_FINE\_LOCATION}), (time > 20 \land time < 8), \text{Log\_Access}())$

Conflicts could arise if user add a constraint conflicting with one of the above

- $\text{permission}(\text{Twitter}, ((\text{access, ACCESS\_FINE\_LOCATION}), (time > 8 \land time < 12), \text{Log\_Access}())$

Necessity of conflicts checking.
Problem With Dependencies

→ Need to take dependencies into account
Representing Dependencies

- Graphs are created when permissions granted to applications (∼installation)
- Constraints over graphs are added depending on tool used (installation, anytime,...)
- Need for a dynamic monitoring of those graph along evolution of concerned applications
- Indirect links have to be managed to avoid privacy and security breaks
- Idea of groups of applications, policies (constraints+permissions) applied to groups, and not standalone applications.
**Representing Dependencies**

- Graphs are created when permissions granted to applications (\sim installation)
- Constraints over graphs are added depending on tool used (installation, anytime, ...)
- Need for a dynamic monitoring of those graph along evolution of concerned applications
- Indirect links have to be managed to avoid privacy and security breaks
- Idea of **groups of applications**, policies (constraints+permissions) applied to groups, and not standalone applications.

→ Dependency groups
DEPENDENCIES IN P-RBAC
P-RBAC EXAMPLES

Permission$_2$: (Group$_1$, ((use, ACCESS\_FINE\_LOCATION), Location = UK, Log\_Access()))

→ restraining all applications belonging to this group by a location constraint on geolocalisation.

Permission$_2$: (Group$_1$, ((use, INTERNET), time > 16 and time < 8, Log\_Access()))

→ restraining all applications belonging to this group to access Internet only during authorized time interval.

→ Multigroup belonging and groups hierarchy also have to be discussed.
CONCLUSION & FUTURE WORK

Conclusion

- P-RBAC allows to model access control policies of mobile applications
- Dependencies between applications can raise privacy issues
- Managing policies over groups of applications allow to resolve some of them

Future Work

- Extend the model to other mobile platforms ×
- Allow the expression of policies using a REL (ODRL) √
- Development of a prototype for Android applications √
- Study privacy changes along dependencies evolution ×