Multi-data-types Interval Decision Diagrams for Attribute-based Access Control Policies Evaluation

Canh Ngo, Yuri Demchenko, Cees de la Laat
System and Network Engineering Group, University of Amsterdam

Problem Statement
Authorization for grid and virtualized cloud service provisioning
• Distributed and open environments.
• Request contexts are flexible, described by attributes.
• Large scale system of resources, end-users, throughputs.

Policy language requirements
• Attributed-based authorization model (ABAC)
• Dynamic assigning roles to users on request contexts.
• Policy management: redundancy detection, integration, conflict resolution, reverse query
• High performance evaluation.
• Efficient mechanisms for ABAC policy language implementations.
• Widely used policy language: XACML version 3.0.

Features
• Support the complete logic expressions evaluation.
• Support complex comparison functions for continuous data types.
• Preserve correctness of original combining algorithms semantic in handling Indeterminate and NotApplicable decisions.
• Support critical attribute priority.
• High performance XACML request evaluation.

Methods
Construct Multi-data-type Interval Decision Diagram (MIDD)
• Extract and aggregate reduced interval partitions from AROF expressions.
• Create decision diagram G(V, E) as a MIDD symbolizing for the AROF expression using Boolean-Shannon expansion.
• Compose MIDDs representing matching rule logic condition: Conjunctive, Disjunctive join MIDDs
• Reduced internal partition aggregation techniques: Union, Intersect, Complement

From MIDDs to X-MIDDs
• X-MIDD: add Condition, Effect, Obligations, Advices elements to leaf-node of MIDD representing a matching rule.
• Internal nodes containing default returned decision: NotApplicable if they are not the "critical" attributes, otherwise set either Indeterminate(P) or "Indeterminate(D)".

Matching rule logic condition
\[ T_i(X) \rightarrow true \]
Target expression:
\[ T(X) = \bigwedge \left( \bigvee \left( \bigwedge m_k \right) \right) \]
Match expression:
\[ m_p(x, f, s) \]

Evaluation Complexity
Policies with n attributes \( a_i, p_i \), each \( p_i \) has k_i distinct values:
• Space complexity: \( O(\sum_{k=1}^{n-1}(2k_i + 1)) \)
• Evaluation time complexity: \( O(\sum_{k=1}^{n}(10g_2(2k_i + 1) + 1)) \)

Implementation and Evaluation
XACML Engine on Java SDK 1.7: http://staff.science.uva.nl/~t.c.ngo/midd-xacml.html

Contact: Canh Ngo <t.c.ngo@uva.nl>, Yuri Demchenko <y.demchenko@uva.nl>