Specification and Deployment of Integrated Security Policies for Outsourced Data

Anis Bkakria

Télécom Bretagne, Rennes, France

Frag&Tag Project
1 Introduction

- Context
- Motivation Scenario
- Outline of Our Contribution
Contents

1 Introduction
   - Context
   - Motivation Scenario
   - Outline of Our Contribution

2 System Specification
   - System modeling
   - Basic Knowledge Axioms
   - Goal specification
Contents

1. Introduction
   - Context
   - Motivation Scenario
   - Outline of Our Contribution

2. System Specification
   - System modeling
   - Basic Knowledge Axioms
   - Goal specification

3. Policy specification
   - Security Constraint
   - Utility Constraint
Contents

1 Introduction
   - Context
   - Motivation Scenario
   - Outline of Our Contribution

2 System Specification
   - System modeling
   - Basic Knowledge Axioms
   - Goal specification

3 Policy specification
   - Security Constraint
   - Utility Constraint

4 Mechanisms specification
Contents

1 Introduction
   - Context
   - Motivation Scenario
   - Outline of Our Contribution

2 System Specification
   - System modeling
   - Basic Knowledge Axioms
   - Goal specification

3 Policy specification
   - Security Constraint
   - Utility Constraint

4 Mechanisms specification

5 Policy satisfaction
Contents

1 Introduction
   - Context
   - Motivation Scenario
   - Outline of Our Contribution

2 System Specification
   - System modeling
   - Basic Knowledge Axioms
   - Goal specification

3 Policy specification
   - Security Constraint
   - Utility Constraint

4 Mechanisms specification

5 Policy satisfaction

6 Conclusion
Current Section

1. Introduction
   - Context
   - Motivation Scenario
   - Outline of Our Contribution

2. System Specification
   - System modeling
   - Basic Knowledge Axioms
   - Goal specification

3. Policy specification
   - Security Constraint
   - Utility Constraint

4. Mechanisms specification

5. Policy satisfaction

6. Conclusion
Context

- **Security, privacy**: major issues impacting the uptake of cloud computing, particularly in public database outsourcing.
- All of end users are concerned with the same security issues.
- Several mechanisms are defined in order to protect sensitive information in outsourced databases.
Motivation Scenario: Banking Scenario

- Two entities:
  - A bank.
  - A credit company.

- Observe different sets of attributes about the same set of individuals, e.g.:
  - $T_1(\text{SSN}, \text{Age}, \text{Adress}, \text{Gender}, \text{Balance})$.
  - $T_2(\text{SSN}, \text{Job}, \text{ZIP}, \text{Nationality}, \text{Salary})$.

- Each party defines a set of constraints to be satisfied.

- **Goal:** Integrate their data in order to be able to make better decision, such as loan, or card limit approval.

- **Problem:** How to choose the best security mechanisms allowing the satisfy data owner’s requirements?
Outline of Our Contribution

Using an Epistemic Linear Temporal Logic (Epistemic LTL), we define an expressive language allowing to:

- Formally express the used system.
- Formally specify the policy.
Using an Epistemic Linear Temporal Logic (Epistemic LTL), we define an expressive language allowing to:

- Formally express the used system.
- Formally specify the policy.

Study of the security properties of the mechanisms that can be used to satisfy the security requirements.

Express these security properties using our language.
Outline of Our Contribution

- Using an Epistemic Linear Temporal Logic (Epistemic LTL), we define an expressive language allowing to:
  - Formally express the used system.
  - Formally specify the policy.
- Study of the security properties of the mechanisms that can be used to satisfy the security requirements.
- Express theses security properties using our language.
- Formally identify the relevant mechanisms according to the security policy to be enforced.
1 Introduction
   • Context
   • Motivation Scenario
   • Outline of Our Contribution

2 System Specification
   • System modeling
   • Basic Knowledge Axioms
   • Goal specification

3 Policy specification
   • Security Constraint
   • Utility Constraint

4 Mechanisms specification

5 Policy satisfaction

6 Conclusion
System modeling

$\langle O, T, A, R, V, L, E \rangle$

- $O = O_1, \cdots, O_l$: a finite set of owners
- $T = \{T_1, \cdots, T_n\}$: a finite set of relational tables
- $A = \{A_{T_1}, \cdots, A_{T_n}\}$: a finite set attributes
- $R = \{R_{T_1}, \cdots, R_{T_n}\}$: a finite set records
- $V = \{v_1, \cdots, v_p\}$: a finite set values
- $E$: a finite set of security mechanisms
- $L$: a finite set of security levels
- $F$: a finite set of functional requirements that can be required over the data.
System modeling

\[ \langle O, T, A, R, V, L, E \rangle \]

- \( O = O_1, \cdots, O_l \): a finite set of owners
- \( T = \{ T_1, \cdots, T_n \} \): a finite set of relational tables
- \( A = \{ A_{T_1}, \cdots, A_{T_n} \} \): a finite set attributes
- \( R = \{ R_{T_1}, \cdots, R_{T_n} \} \): a finite set records
- \( V = \{ v_1, \cdots, v_p \} \): a finite set values
- \( E \): a finite set of security mechanisms
- \( L \): a finite set of security levels
- \( F \): a finite set of functional requirements that can be required over the data.

List of static predicates

- \( \text{Exp} : \text{value}(v_1) \)

List of fluents

- \( \text{Exp} : \text{knows}(o_1, v_1) \)
Data Model

- Table \( T \) with attributes \( A_1, A_2, A_3, A_4 \) and records \( R_1, R_2, R_3 \)
  - Record \( R_3 \):
    - \( v_9 \), \( v_{10} \), \( v_{11} \), \( v_{12} \)
- Attributes
  - \( attribute(A_4) \)
- Values
  - \( value(V_{12}) \)
- Relations
  - \( relation \)
  - \( symbol \)
Data Model

- Table (T)
  - Attributes: A₁, A₂, A₃, A₄
  - Records:
    - R₁: v₁, v₂, v₃, v₄
    - R₂: v₅, v₆, v₇, v₈
    - R₃: v₉, v₁₀, v₁₁, v₁₂

- Relations and Values
  - owner (O)
  - attributeOf (T, A₄)
  - attribute (A₄)
  - valueOf (R₃, A₄, v₁₂)
  - recordOf (T, R₃)
  - value (V₁₂)
  - record (R₃)
  - table (T)

- Relation symbol

- Attributes:
  - v₁, v₂, v₃, v₄
  - v₅, v₆, v₇, v₈
  - v₉, v₁₀, v₁₁, v₁₂

- Values:
  - v₁, v₂, v₃, v₄
  - v₅, v₆, v₇, v₈
  - v₉, v₁₀, v₁₁, v₁₂
Data Model

- **Table** ($T$)
  - **Attributes**: $A_1, A_2, A_3, A_4$
  - **Values**:
    - $R_1$: $v_1, v_2, v_3, v_4$
    - $R_2$: $v_5, v_6, v_7, v_8$
    - $R_3$: $v_9, v_{10}, v_{11}, v_{12}$

- **Relations**:
  - $\text{attributeOf}(T, A_4)$
  - $\text{valueOf}(R_3, A_4, v_{12})$
  - $\text{owner}(O)$
Data Model

- **Attributes**
  - `attribute(\(A_4\))`
  - `valueOf(\(R_3, A_4, v_{12}\))`
- **Relations**
  - `recordOf(\(T, R_3\))`
  - `belongsTo(\(O, T\))`
  - `attributeOf(\(T, A_4\))`

**Table:**

<table>
<thead>
<tr>
<th></th>
<th>(A_1)</th>
<th>(A_2)</th>
<th>(A_3)</th>
<th>(A_4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(R_1)</td>
<td>(v_1)</td>
<td>(v_2)</td>
<td>(v_3)</td>
<td>(v_4)</td>
</tr>
<tr>
<td>(R_2)</td>
<td>(v_5)</td>
<td>(v_6)</td>
<td>(v_7)</td>
<td>(v_8)</td>
</tr>
<tr>
<td>(R_3)</td>
<td>(v_9)</td>
<td>(v_{10})</td>
<td>(v_{11})</td>
<td>(v_{12})</td>
</tr>
</tbody>
</table>
Basic Knowledge Axioms

1. An owner knows all information stored in tables that belong to him:
   $$\forall T_1, O_1, A_1, R_1, V_1. \left[ \text{belongs}(O_1, T_1) \land \text{attributeOf}(T_1, A_1) \land \text{recordOf}(T_1, R_1) \land \text{valueOf}(R_1, A_1, V_1) \rightarrow \text{knows}(O_1, V_1) \right]$$  (1)

2. An owner has no knowledge about the information stored in tables that not belong to him:
   $$\forall T_1, O_1, A_1, R_1, V_1. \left[ \neg \text{belongs}(O_1, T_1) \land \text{attributeOf}(T_1, A_1) \land \text{recordOf}(T_1, R_1) \land \text{valueOf}(R_1, A_1, V_1) \rightarrow \neg \text{knows}(O_1, V_1) \right]$$  (2)

3. Data owners never forget an information they know:
   $$\forall O_1, V_1. \text{knows}(O_1, V_1) \rightarrow \text{knows}(O_1, V_1)$$  (3)
Basic Knowledge Axioms

- An owner knows all information stored in tables that belong to him

\[ \forall T_1, O_1, A_1, R_1, V_1. \ [ \text{belongs}(O_1, T_1) \land \text{attributeOf}(T_1, A_1) \land \text{recordOf}(T_1, R_1) \land \text{valueOf}(R_1, A_1, V_1) \rightarrow \text{knows}(O_1, V_1) ] \] (1)
Basic Knowledge Axioms

• An owner knows all information stored in tables that belong to him

\[ \forall T_1, O_1, A_1, R_1, V_1. \ [ \text{belongs}(O_1, T_1) \land \text{attributeOf}(T_1, A_1) \land \text{recordOf}(T_1, R_1) \land \text{valueOf}(R_1, A_1, V_1) \rightarrow \text{knows}(O_1, V_1) ] \]  (1)

• An owner has no knowledge about the information stored in tables that not belong to him

\[ \forall T_1, O_1, A_1, R_1, V_1. \ [ \neg \text{belongs}(O_1, T_1) \land \text{attributeOf}(T_1, A_1) \land \text{recordOf}(T_1, R_1) \land \text{valueOf}(R_1, A_1, V_1) \rightarrow \neg \text{knows}(O_1, V_1) ] \]  (2)
Basic Knowledge Axioms

- An owner knows all information stored in tables that belong to him

\[ \forall T_1, O_1, A_1, R_1, V_1. \ [ \text{belongs}(O_1, T_1) \land \text{attributeOf}(T_1, A_1) \land \text{recordOf}(T_1, R_1) \land \text{valueOf}(R_1, A_1, V_1) \rightarrow \text{knows}(O_1, V_1) ] \] (1)

- An owner has no knowledge about the information stored in tables that not belong to him

\[ \forall T_1, O_1, A_1, R_1, V_1. \ [ \neg \text{belongs}(O_1, T_1) \land \text{attributeOf}(T_1, A_1) \land \text{recordOf}(T_1, R_1) \land \text{valueOf}(R_1, A_1, V_1) \rightarrow \neg \text{knows}(O_1, V_1) ] \] (2)

- Data owners never forget an information they know

\[ \forall O_1, V_1. \ \text{knows}(O_1, V_1) \rightarrow \bigcirc \text{knows}(O_1, V_1) \] (3)
If the data owners of two tables $T_1$ and $T_2$ want to integrate their private data then eventually, there will exist a table $T_j$ representing the joint of $T_1$ and $T_2$. $orall T_1, T_2$ join($T_1, T_2$) $\rightarrow$ ♦($\exists T_j$ JoinOf($T_1, T_2, T_j$) $\{4\}$

The set of attributes of the joint table $T_j$ is composed of the union of sets of join-involved attributes of private tables $T_1$ and $T_2$. $orall T_1, T_2, T_j$, $A$. JoinOf($T_1, T_2, T_j$) $\land$ (joinInvolved($T_1, A$) $\lor$ joinInvolved($T_2, A$)) $\rightarrow$ attributeOf($T_j, A$) $\{5\}$

The table $T_j$ representing the joint of $T_1$ and $T_2$ belongs to both owner of $T_1$ and owner of $T_2$. $orall T_1, T_2, T_j, O_1, O_2$. JoinOf($T_1, T_2, T_j$) $\land$ belongs($O_1, T_1$) $\land$ belongs($O_2, T_2$) $\rightarrow$ belongs($O_1, T_j$) $\land$ belongs($O_2, T_j$) $\{6\}$
Goal specification

If the data owners of two tables $T_1$ and $T_2$ want to integrate their private data then eventually, there will exist a table $T_j$ representing the joint of $T_1$ and $T_2$

$$\forall T_1, T_2. \ join(T_1, T_2) \rightarrow \Diamond (\exists T_j \ JoinOf(T_1, T_2, T_j)) \tag{4}$$
Goal specification

- If the data owners of two tables $T_1$ and $T_2$ want to integrate their private data then eventually, there will exists a table $T_j$ representing the joint of $T_1$ and $T_2$

$$\forall T_1, T_2. \ join(T_1, T_2) \rightarrow \exists T_j \ JoinOf(T_1, T_2, T_j)$$ (4)

- The set of attributes of the joint table $T_j$ is composed of the union of sets of join-involved attributes of private tables $T_1$ and $T_2$

$$\forall T_1, T_2, T_j, A. \ JoinOf(T_1, T_2, T_j) \land (joinInvolved(T_1, A) \lor joinInvolved(T_2, A)) \rightarrow attributeOf(T_j, A)$$ (5)
Goal specification

- If the data owners of two tables \( T_1 \) and \( T_2 \) want to integrate their private data then eventually, there will exists a table \( T_j \) representing the joint of \( T_1 \) and \( T_2 \)

\[
\forall T_1, T_2. \ join(T_1, T_2) \rightarrow \Diamond \ (\exists T_j \ JoinOf(T_1, T_2, T_j)) \tag{4}
\]

- The set of attributes of the joint table \( T_j \) is composed of the union of sets of join-involved attributes of private tables \( T_1 \) and \( T_2 \)

\[
\forall T_1, T_2, T_j, A. \ JoinOf(T_1, T_2, T_j) \land (joinInvolved(T_1, A) \lor joinInvolved(T_2, A)) \rightarrow attributeOf(T_j, A) \tag{5}
\]

- The table \( T_j \) representing the join of \( T_1 \) and \( T_2 \) belongs to both owner of \( T_1 \) and owner of \( T_2 \).

\[
\forall T_1, T_2, T_j, O_1, O_2. \ JoinOf(T_1, T_2, T_j) \land belongs(O_1, T_1) \land belongs(O_2, T_2) \\
\rightarrow belongs(O_1, T_j) \land belongs(O_2, T_j) \tag{6}
\]
Current Section

1 Introduction
   - Context
   - Motivation Scenario
   - Outline of Our Contribution

2 System Specification
   - System modeling
   - Basic Knowledge Axioms
   - Goal specification

3 Policy specification
   - Security Constraint
   - Utility Constraint

4 Mechanisms specification

5 Policy satisfaction

6 Conclusion
Security Constraint

- **Confidentiality Constraint**
  
  States that the values assumed by an attribute are considered sensitive
Security Constraint

- **Confidentiality Constraint**
  - States that the values assumed by an attribute are considered sensitive
  - **Abstract level:** $S\text{AttributeOf}(T, A)$
Security Constraint

- Confidentiality Constraint
  - States that the values assumed by an attribute are considered sensitive
  - **Abstract level**: $S\text{AttributeOf} (T, A)$
  - **Concrete level**:

    \[
    \forall A, T. \ S\text{AttributeOf} (T, A) \rightarrow (\forall O, R, V. \text{recordOf}(T, R) \land \\
     \text{valueOf}(R, A, V) \land \neg \text{belongs}(O, T) \rightarrow \neg \text{knows}(O, V))
    \]
Security Constraint

- Anonymization Constraint
  - require the prevention of identity disclosure by protecting personal identifiers
Security Constraint

- **Anonymization Constraint**
  - require the prevention of identity disclosure by protecting personal identifiers
  - **Abstract level**: $\text{withoutIDDisclosure}(T)$
Security Constraint

- **Anonymization Constraint**
  - require the prevention of identity disclosure by protecting personal identifiers
  - **Abstract level**: \( \text{withoutIDDisclosure}(T) \)
  - **Concrete level**:
    \[
    \forall T. \ \text{withoutIDDisclosure}(T) \rightarrow \left( \forall A, O, R, V. \ \text{IDAttributeOf}(T, A) \land recordOf(T, R) \land valueOf(R, A, V) \land \neg \text{belongs}(O, T) \right) \rightarrow \neg \text{knows}(O, V) \]
Utility Constraint

- A data owner can require that particular properties on his data must be respected.

\[ utility\_requirement(U) \land provides(U, A). \] (9)

- Four classes of utility requirements:
  - Equality check requirements.
  - Order check requirements.
  - Computational requirements.
  - Keyword search requirements.
Current Section

1. Introduction
   - Context
   - Motivation Scenario
   - Outline of Our Contribution

2. System Specification
   - System modeling
   - Basic Knowledge Axioms
   - Goal specification

3. Policy specification
   - Security Constraint
   - Utility Constraint

4. Mechanisms specification

5. Policy satisfaction

6. Conclusion
Security mechanisms specification

Security mechanisms is specified using three groups of formulas:
Security mechanisms specification

Security mechanisms is specified using three groups of formulas:

- **Preconditions**: Conditions under which the security mechanism can be applied.

\[
\square (is\_applicable(M, Obj) \rightarrow \Delta_M)
\]  

(10)
Security mechanisms specification

Security mechanisms is specified using three groups of formulas:

- **Preconditions:** Conditions under which the security mechanism can be applied.

  \[ \square (\text{is\_applicable}(M, Obj) \rightarrow \Delta_M) \]  
  \[ (10) \]

- **Effects:** Modifications applied to the system during the transition due to the application of the mechanism.

  \[ \Phi(w_i, \text{apply}(M, Obj)) = w_j \rightarrow (w_j \models \Sigma_M) \]  
  \[ (11) \]
Security mechanisms specification

Security mechanisms is specified using three groups of formulas:

- **Preconditions:** Conditions under which the security mechanism can be applied.
  \[ \square (is\_applicable(M, Obj) \rightarrow \Delta_M) \] (10)

- **Effects:** Modifications applied to the system during the transition due to the application of the mechanism.
  \[ \Phi(w_i, apply(M, Obj)) = w_j \rightarrow (w_j \models \Sigma_M) \] (11)

- **Properties:** The set of properties \( P_1, \ldots, P_n \) that can be derived from the effects of the mechanisms.
  \[ \Sigma_M \rightarrow \bigwedge_{i=1}^{n} P_i \] (12)
Example: Encryption based mechanisms

- Preconditions:
  \[
  \square \left[ \forall M, A. \ enc\_based\_mechanism(M) \land is\_applicable(M, A) \rightarrow \exists T. \ SAttributeOf(T, A) \land join\_involved(T, A) \right]
  \] (13)
Example: Encryption based mechanisms

- **Preconditions:**
  \[ \forall M, A. \ enc\_based\_mechanism(M) \land is\_applicable(M, A) \rightarrow \exists T. \ SAttributeOf(T, A) \land join\_involved(T, A) \]  
  \[ (13) \]

- **Effects:**
  \[ \forall M, A, T, K. \ enc\_based\_mechanism(M) \land attribute\_of(T, A) \land enc\_key(K) \land apply(M, A) \rightarrow encrypted(T, A, K) \]  
  \[ (14) \]
Example: Encryption based mechanisms

- **Preconditions:**
  \[
  \forall M, A. \ enc\_based\_mechanism(M) \land \text{is\_applicable}(M, A) \rightarrow \\
  \exists T. \ S\text{AttributeOf}(T, A) \land \text{join\_involved}(T, A)
  \]

- **Effects:**
  \[
  \forall M, A, T, K. \ enc\_based\_mechanism(M) \land \text{attribute\_of}(T, A) \land \\
  enc\_key(K) \land \text{apply}(M, A) \rightarrow \text{encrypted}(T, A, K)
  \]

- **Properties:**
  \[
  \forall A, T, K, O. \ enc\_key(K) \land \text{encrypted}(T, A, K) \land \neg \text{knows}(O, K) \rightarrow \\
  \text{protected}(T, A, O)
  \]
Example: Encryption based mechanisms

- **Preconditions:**
  \[\forall M, A. \text{enc\_based\_mechanism}(M) \land \text{is\_applicable}(M, A) \rightarrow \exists T. \ S\text{AttributeOf}(T, A) \land \text{join\_involved}(T, A)\]

- **Effects:**
  \[\forall M, A, T, K. \ \text{enc\_based\_mechanism}(M) \land \text{attribute\_of}(T, A) \land \text{enc\_key}(K) \land \text{apply}(M, A) \rightarrow \text{encrypted}(T, A, K)\]

- **Properties:**
  \[\forall A, T, K, O. \ \text{enc\_key}(K) \land \text{encrypted}(T, A, K) \land \neg \text{knows}(O, K) \rightarrow \text{protected}(T, A, O)\]

Partially homomorphic encryption based mechanisms

\[\forall M, A. \ \text{hom\_mechanism}(M) \land \text{apply}(M, A) \rightarrow \text{sec\_level}(A, \text{random}) \land \text{provides}(\text{computation}, A)\]
Example: Encryption based mechanisms

Searchable encryption based mechanisms

\[ \forall A. \ searchable\_enc\_mechanism(M) \land \ apply(M, A) \rightarrow \]
\[ \text{sec\_level}(A, \text{random}) \land \ provides(\text{keyword\_search}, A) \]
Example: Encryption based mechanisms

Searchable encryption based mechanisms

\[ \forall A. \ searchable\_enc\_mechanism(M) \land \ apply(M, A) \rightarrow \]
\[ \text{sec\_level}(A, \text{random}) \land \text{provides(keywork\_search, A)} \]

Deterministic encryption based mechanisms

\[ \forall M, A. \ det\_enc\_mechanism(M) \land \ apply(M, A) \rightarrow \]
\[ \text{provides(equality\_check, A)} \land \text{sec\_level}(A, \text{det}) \]
Example: Encryption based mechanisms

**Searchable encryption based mechanisms**

\[ \forall A. \; \text{searchable}\text{-}\text{enc}\text{-}\text{mechanism}(M) \land \text{apply}(M, A) \rightarrow \text{sec}\text{-}\text{level}(A, \text{random}) \land \text{provides}(\text{keywork}\text{-}\text{search}, A) \]

**Deterministic encryption based mechanisms**

\[ \forall M, A. \; \text{det}\text{-}\text{enc}\text{-}\text{mechanism}(M) \land \text{apply}(M, A) \rightarrow \text{provides}(\text{equality}\text{-}\text{check}, A) \land \text{sec}\text{-}\text{level}(A, \text{det}) \]

**Order-preserving encryption based mechanisms**

\[ \forall M, A. \; \text{ope}\text{-}\text{mechanism}(M) \land \text{apply}(M, A) \rightarrow \text{sec}\text{-}\text{level}(A, \text{ope}) \land \text{provides}(\text{equality}\text{-}\text{check}, A) \land \text{provides}(\text{order}\text{-}\text{check}, A) \]
Current Section

1. Introduction
   - Context
   - Motivation Scenario
   - Outline of Our Contribution

2. System Specification
   - System modeling
   - Basic Knowledge Axioms
   - Goal specification

3. Policy specification
   - Security Constraint
   - Utility Constraint

4. Mechanisms specification

5. Policy satisfaction

6. Conclusion
**Policy satisfaction**

**Goal:** Find the best combination of security mechanisms that can satisfy the set of security and utility constraints
Goal: Find the best combination of security mechanisms that can satisfy the set of security and utility constraints.

Step 1: Satisfy the chosen goal.

Find the set of mechanisms $\mathcal{M}$ that satisfy the chosen goal.

$$\Sigma \cup \Sigma_{M_g} \vdash \Sigma_G,$$

where $M_g \in \mathcal{M}$.
Goal: Find the best combination of security mechanisms that can satisfy the set of security and utility constraints.

Step 1: Satisfy the chosen goal.

Find the set of mechanisms $\mathcal{M}$ that satisfy the chosen goal.

$$\Sigma \cup \Sigma_{M_g} \vdash \Sigma_G,$$

where $M_g \in \mathcal{M}$

Step 2: Violated security constraints.

- For each $M_i \in M_g$, we look for the set of violated security and utility constraints $C_i$.

$$\Sigma \cup \Sigma_{M_i} \cup \Sigma_C \vdash \bot.$$

- Get the best goal satisfier $M_j$

$$\forall M_i \in \mathcal{M}, \quad |C_j| \leq |C_i|$$
Policy satisfaction

Step 3: Satisfying the violated constraints.

We find the set of properties that can satisfy each violated constraint.

\[ \bigwedge_{i=1}^{l} \Sigma_{P_i} \cup \Sigma \cup \Sigma_{M_{bgs}} \vdash \Sigma_{C} \]
Policy satisfaction

Step 3: Satisfying the violated constraints.

We find the set of properties that can satisfy each violated constraint.
\[
\bigwedge_{i=1}^l \Sigma_{P_i} \cup \Sigma \cup \Sigma_{M_{bgs}} \vdash \Sigma_C
\]

Step 4: Choosing the best security mechanisms.

- Find the combination of mechanisms \( MC \) that provides the required security properties while satisfying set of the utility constraints \( U_{ob} \).

\[
\Sigma \cup \left\{ \bigwedge_{M \in MC} \text{apply}(M, Ob) \right\} \models \left( \bigwedge_{P \in \mathcal{P}} P \bigwedge_{U \in U_{ob}} \text{provides}(U, Ob) \right)
\]

- Many combinations of mechanisms can be used. We choose the best combination of mechanisms:
\[
\bigwedge_{i=1}^n \left( \text{more}_{-}\text{secure}_{-}\text{than}(l_i, l_j) \lor (l_i = l_j \land |CM_i| < |CM_j|) \right)
\]
We present a formal model allowing to:

- Formally express the security policy defined by the security administrators.
- Formally express Security and utility properties that characterize each security mechanism in our toolbox.
- Formally identify the relevant combination of mechanisms to efficiently enforce the defined security policy.

Future work

- Extend our approach to be able to deal with other kinds of security constraints, e.g. integrity and traceability.
Questions