On XACML’s Adequacy to Specify and to Enforce HIPAA

Omar Chowdhury\textsuperscript{1} Haining Chen\textsuperscript{2} Jianwei Niu\textsuperscript{1} 
Ninghui Li\textsuperscript{2} Elisa Bertino\textsuperscript{2}

University of Texas at San Antonio\textsuperscript{1} 
Purdue University\textsuperscript{2}

3rd USENIX Workshop on Health Security and Privacy (HealthSec ’12) 
August 7, 2012
Motivation

- Organizations collect private information from their customer for performing their business operations
  - **Example**: Healthcare providers collect private health information from their patient.

- Federal regulations mandate how the collected information can be used or disclosed
  - **Example**: Health Insurance Portability and Accountability Act (HIPAA), Gramm-Leach-Bliley Act (GLBA), etc.

- Violations of these regulations can bring down heavy financial penalties and sanctions for the organizations

- Violations might also be harmful to the organizations’ reputation
The Problem

- Researchers have proposed formalism to completely specify privacy regulations like HIPAA

- Organizations intended to enforce privacy regulations will have their own access control policies and business privacy policies

- Using different formalisms to capture each of these policies is cumbersome

- An action can be regulated by all of the policies of the organization

- Have to combine the decisions of the different policies manually
The Current Work

OASIS’s eXtensible Access Control Markup Language (XACML) is a widely used access control formalism in both industry and academic research.

The current work evaluates the adequacy of XACML’s specification language and enforcement engine to specify and enforce HIPAA.

XACML has some rich enough features.

**Example**: attributes, policy/policy rule combination, etc.

XACML naturally lacks some features to support HIPAA.

**Example**: event history, obligations, subjective beliefs, etc.

We present high level designs to extend XACML with the missing features.
HIPAA privacy regulations ensure that the consumers can access their health information and also make sure their information is protected from unauthorized disclosure.

It mandates the usage or disclosure of patient’s protected health information by the covered entities.

- **Example:** health plans, health care providers, healthcare clearing houses, *etc.*

*Protected health information* (*phi*) refers to the individually identifiable health information.

The purposes of a usage or disclosure *phi* is also regulated by HIPAA.

The role of the entity to whom the disclosure is made is also regulated by HIPAA.
eXtensible Access Control Markup Language (XACML) Enforcement Architecture

- Policy Enforcement Point (PEP)
- Policy Decision Point (PDP)
- Policy Information Point (PIP)
- Policy Administration Point (PAP)
- Obligation Service

Access Request → Policy Enforcement Point (PEP) → Access Response

Request → Policy Decision Point (PDP) → Response

Attribute query → Policy Information Point (PIP)

Access appropriate policy → Policy Database

Administration → Policy Administration Point (PAP)
Outline

1. Motivation
2. Background
3. Features Necessary for HIPAA
4. Evaluating XACML for HIPAA
5. High Level Design for Extending XACML
6. Related Work
7. Concluding Remarks

On XACML’s Adequacy to Specify and to Enforce HIPAA
Necessary Features for HIPAA

- **Attributes**: sender, receiver, subject, message
  - §164.502(a)(1)(i): a covered entity is permitted to use or disclose \( phi \) to the individual

- **Attribute Inference Policy**: it regulates whether a principal has a particular attribute based on his current attributes
  - §164.502(g)(2): under what conditions a principal is considered another individual’s personal representative

- **Past Events**: past events can influence the permissibility of an action
  - §164.502(e)(1)(i): a covered entity can disclose \( phi \) to its business associate provided that it has received satisfactory assurance about safeguarding the information
Necessary Features for HIPAA (contd.)

- **Obligations**: the regulations can also impose obligatory requirements
  - §164.524(b)(2)(i): a covered entity must act on a request for access no later than 30 days after receiving the request

- **Purpose**: purpose of an action can also influence its permissibility
  - §164.506(c)(1): a covered entity may use or disclose phi for its own treatment, payment, or health care operations

- **Subjective Beliefs**: a subject’s judgement can influence permissibility of an action
  - §164.512(f)(5): a covered entity can disclose phi to a law enforcement official if he thinks it can be used as evidence

- **Reference to Other Laws/Rules**: the regulations can refer to other laws or rules
  - §164.512(a)(1): a covered entity may use or disclose phi when it is required by other law
Assumptions

- The actions we consider are: disclose, request, use, and access
- We only regulate communication messages containing phi of an individual
- The sending principal provides the purpose of the transmission
- It is the responsibility of the sending principal to tag the message with its appropriate attributes
- Any incurred obligations are consistent with the policies
- Patient policies are consistent with HIPAA
- We assume there exists an oracle that makes some decision about some request
  - Example: whether certain action is prohibited by any applicable law
Outline

1. Motivation
2. Background
3. Features Necessary for HIPAA
4. Evaluating XACML for HIPAA
5. High Level Design for Extending XACML
6. Related Work
7. Concluding Remarks

On XACML’s Adequacy to Specify and to Enforce HIPAA
Stateful Policies vs. Stateless Mechanism

- **XACML** policies are largely stateless.
- The enforcement mechanism of **XACML** is also stateless.
- Any stateful information is kept outside the policy engine.
- The **HIPAA** privacy rules are stateful.
- The enforcement mechanism for the **HIPAA** privacy rules needs to be stateful too.

The reason for **HIPAA** requiring stateful mechanisms are:

- Obligations
- Event history
- Policy-directed attribute retrieval
- Policy-directed policy retrieval
Interactive vs. Non-interactive Policy Evaluation

- **XACML**’s policy evaluation is non-interactive

- However, it seems for **HIPAA** an interactive policy evaluation is needed

- The necessity for the interactive policy evaluation:
  - Subjective beliefs
  - Reference to other policy rules and laws

- Determining them from the static context of a request is not always feasible
Other Considerations

- Attribute inference policy vs. privacy rules
  - A disclosure or usage is allowed when the receiver is patient’s personal representative
  - **Example**: is principal $p$ a personal representative of the principal $q$?

- Quantification over the infinite domains
  - Quantification is needed for concise policy specification
  - Domains are: principals, message attributes, messages, etc.
  - **Example**: a disclosure is allowed if the sender received a message containing the authorization before
  - **XACML**’s specification language does not support quantification explicitly
Outline

1 Motivation
2 Background
3 Features Necessary for HIPAA
4 Evaluating XACML for HIPAA
5 High Level Design for Extending XACML
6 Related Work
7 Concluding Remarks
HIPAA Privacy Rules

- Required privacy rules and permitting privacy rules
- Permitting privacy rules are divided into two more types
  - Allowing and prohibitive privacy rules
- Each privacy rule can regulate the following:
  - Sender’s, recipient’s, and subject’s attributes (e.g., role, etc.)
  - Purpose of the disclosure (e.g., treatment, payment, etc.)
  - The message attributes (e.g., phi, ssn, psychotherapy-notes etc.)
  - Obligations
  - Event history
  - Other conditions
Extensions

Obligations

- We use the obligation model of Li et al. 2010
- An obligation is modeled as a state machine that changes state with respect to events
- PEP keeps track of the obligations’ state

Event history

- We propose a “history manager”
- Relation database that keeps track of the important events
- Manually inspect the policy to decide which events to store in the history manager
- Example: A covered entity can disclose the phi if it has received a court-order

Interaction with users and the oracle

- Get information about subjective beliefs
- Obtain information that is not present in the state (e.g., reference to other laws, etc.)
Details of Extensions

\[
<\text{PolicySet}> := <\text{Target}><\text{Policy}>^+[\text{Obligations}]
\]
Attributes: PolicySetId, PolicyCombiningAlgId

\[
<\text{Policy}> := <\text{Target}><\text{Rule}>^+[\text{Obligations}]
\]
Attributes: PolicyId, RuleCombiningAlgId

\[
<\text{Rule}> := [\text{Target}][\text{Condition}]
\]
Attributes: RuleId, Effect

\[
<\text{Policy}> := [\text{RequiredAttributeList}]<\text{Target}><\text{Rule}>^+[\text{Obligations}]
\]
Attributes: PolicyId, RuleCombiningAlgId

\[
<\text{RequiredAttributeList}> := <\text{RequiredAttributeSelector}>^+
\]

\[
<\text{RequiredAttributeSelector}> := [\text{Keys}]
\]
Attributes: AttributeId, DataType, Source, DatabaseId (optional), TableId (optional)

\[
<\text{Source}> := "\text{User}" | "\text{Database}" | "\text{Oracle}"
\]

\[
<\text{Keys}> := <\text{Key}>^+
\]
\[
<\text{Key}> := <\text{KeyValue}>
\]
Attributes: KeyId

On XACML’s Adequacy to Specify and to Enforce HIPAA
Details of Extensions (contd.)

\[
\text{\textit{Condition}} := \text{\textit{Expression}}
\]

The \textit{Expression} element substitution group includes:
\textit{AttributeSelector}, \textit{AttributeValue}, \textit{VariableReference},
\textit{ActionAttributeDesignator}, \textit{ResourceAttributeDesignator}, \textit{Function},
\textit{SubjectAttributeDesignator}, \textit{Apply}, \textit{EnvironmentAttributeDesignator},
\textit{EventSelector}

\[
\text{\textit{EventSelector}} :=
\text{Attributes: EventType, EventField, DataType}
\]
Details of Extensions (contd.)

\[
\text{\texttt{<Condition> := <Expression>}}
\]

The \texttt{<Expression>} element substitution group includes:
\texttt{<AttributeSelector>, <AttributeValue>, <VariableReference>,}
\texttt{<ActionAttributeDesignator>, <ResourceAttributeDesignator>, <Function>,}
\texttt{<SubjectAttributeDesignator>, <Apply>, <EnvironmentAttributeDesignator>,}
\texttt{<EventSelector>, <AttributeInferencePolicyReference>}

\[
\text{\texttt{<AttributeInferencePolicyReference> := <Input>^+}}
\]

Attributes: AttributeInferencePolicyId

Extension for handling attribute inference policy
Additional Policies

Organizational access control policies

- Only the assigned doctors and nurses can access the phi of the patient
- It must be consistent with the HIPAA privacy rules

Patient policies

- According to HIPAA §164.522, a covered entity can agree or disagree to comply with a patient’s policy
- When it agrees to comply with the patient policy, it has to satisfy the patient policy
- The patient policies must be consistent with the HIPAA privacy rules
Policy Combination

- Required policy rules are combined using **Permit-override**
- Allowing policy rules are combined using **Permit-override**
- Prohibitive policy rules are combined using **Deny-override**
- Permitting policies are combined using **Deny-override**
- Required policies and Permitting policy is combined using **Permit-override**

Combining additional policies:
- **Ordered-deny-overrides** policy combination algorithm is used
- Policies are ordered in the following order: access control policy, patient policy, HIPAA policies
Extended XACML Enforcement Architecture

On XACML's Adequacy to Specify and to Enforce HIPAA
Related Work

- Tschantz et al. 2012: Enforcing the purpose restrictions in the privacy policies
- Garg et al. 2011: Formalized HIPAA and present an incremental auditing algorithm
- DeYoung et al. 2010: Formalized HIPAA and GLBA in the logical specification language PrivacyLFP
- Lam et al. 2009: Formalized HIPAA in a datalog based specification language pLogic
- May et al. 2006: Formalized HIPAA in HRU based specification language Privacy API and performed analysis
- Barth et al. 2006, 2007: Formalized HIPAA and GLBA in the first order linear temporal logic (FOTL)
- Breaux et al. 2005, 2006, 2008: Tool support for formalizing legal regulations as requirements
Conclusion

- We evaluate XACML’s adequacy to specify and enforce HIPAA
- XACML has some rich enough features
- XACML lacks some features for HIPAA
- We present high level designs for extending XACML to support HIPAA

Future work:
- Develop a prototype with the proposed extensions
- Relax some of the restrictions
Questions?

Thank you for your attention