Towards HIPAA-compliant Healthcare Systems

Ruoyu Wu, Gail-Joon Ahn, Hongxin Hu
Arizona State University

SEFCOM
Security Engineering for Future Computing
Outline

- Introduction and Motivation
  - Healthcare IT and its security and privacy
  - Challenges for regulation compliance management
- Our Proposed Approach
  - Extracting policy pattern
  - Generic policy specification
  - Transformation
  - Compliance-oriented analysis
- Case Study
- Implementation and Evaluation
- Related Work
- Conclusion
- Future Work
Healthcare IT

- Currently, paper-based medical records are transforming to Electronic Medical Records (EMRs)
  - Healthcare providers can share information across their care ecosystem.
  - Access to this digital lifeline is critical to saving lives, preventing medical errors and improving efficiency of healthcare delivery.
Security and Privacy

- Threats on EMRs
  - Information leakage
  - Data forgery
  - Data loss

- HIPAA
  - A legal regulation enacted by US congress to address the security and privacy of health data
  - Hence, it is critical to ensure EMR systems to be compliant with HIPAA regulations
Challenges

- Challenges of HIPAA compliance management
  - It is a manual and labor-intensive process
  - It creates additional overheads to health information transactions.
  - HIPAA are complex and in part vague, requiring interpretation.
  - The complexity in implementing compliance objective can rapidly increase as the updates of HIPAA regulation and the upscale of health IT system.

- Hence, to accommodate these requirements and challenges, a novel systematic and automated approach is needed
Our Proposed Approach

- Overview of our compliance analysis framework

Why do we need the abstract layer?

- Facilitates the process of compliance analysis
- Improves interoperability, consistency and irresolubility of the policies from different organizations
- Independent upon underlying reasoning techniques
Extracting Policy Pattern

- **Goal**
  - Extract general policy patterns to define a uniform policy scheme for HIPAA and system policy transformation

- **Methodology**
  - Identify key words from HIPAA rules or system policies;
  - Categorize identified key words into different class and give a label;
  - Given a new rule or a policy, map each word to a class;
  - The composition of different labels construct a structured pattern.

```plaintext
class1: xxx, xxx, xxx, xxx, xxx --label1
class2: xxx, xx, xxx, xxx --label2
class3: xxx, xxx, xxx, xxx, xxx, xxx, xxxx --label3
class4: xxx, xxx, xxxxx, xx, xxxx, xxxxx --label4

given a new rule:
xxx xxx xx xxx xxx xxx xxx

label1 label3 label2 label4 label3 structured pattern
```
Extracting Policy Pattern-Cont’d

- **Structured Examples**
  - 164.506(b)(1) A covered entity may obtain consent of the individual to use or disclose protected health information to carry out treatment, payment, or health care operations.
    
    Extracted Pattern: <actor> <modality> <action> <object> to <action> <object> for <purpose>

  - 164.506(c)(1) A covered entity may use or disclose protected health information for its own treatment, payment, or health care operations.
    
    Extracted Pattern: <actor> <modality> <action> <object> for <purpose>

  - 164.506(c)(2) A covered entity may disclose protected health information for treatment activities of a health care provider.
    
    Extracted Pattern: <actor> <modality> <action> <object> for <purpose>
Generic Policy Specification

- Our policy specification scheme is built upon the identified policy patterns as a 8-tuple \( p = \langle \text{actor}, \text{modality}, \text{action}, \text{object}, \text{purpose}, \text{condition}, \text{id}, \text{effect} \rangle \), where
  - \text{actor} = \langle D, R, O \rangle is a 3-tuple, where \( D, R \) and \( O \) represent disseminator, receiver, and owner, respectively;
  - \text{modality} depends on the implication that a policy expresses;
  - \text{action} is a particular action defined by a policy;
  - \text{object} is a protected healthcare resource;
  - \text{purpose} is the reason for an actor to perform an action on an object;
  - \text{condition} = \langle C_D, C_R, C_O, C_{CON} \rangle is a 4-tuple, where \( C_D, C_R, C_O \) and \( C_{CON} \) indicate conditions on disseminator, receiver, owner and context, respectively;
  - \text{id} is the citation to the portion of HIPAA regulations to which a policy refers to;
  - \text{effect} is the authorization effect of a policy including \textit{permit} and\textit{ deny}. 
**Transformation**

- **Step One**: Transform both HIPAA regulations and healthcare system policies into the uniform abstract representation:

  - **Abstraction**
    - Establishing Word Dictionary
  - **Functionality**
    - Goal: Categorize identified key words
      - Steps:
        1. Identify key words from given text;
        2. Categorize identified key words into different classes;
        3. Assign a label to each class
    - Goal: Divide each given rule in syntactically correlated parts of words
    - Goal: Identify each element of the policy
      - 1. actor (Disseminator, receiver, owner)
      - 2. modality
      - 3. action
      - 4. object
      - 5. purpose
      - 6. condition
      - 7. rule id
      - 8. effect
    - Goal: Remove disjunction
      - Steps:
        1. Record the number of cases
        2. Repeat constructing rules according to the above number
  - **Technical Details**
    - Store each class of words into an arraylist
    - Leverage NLP API
      - sentence detection
      - tokenization
      - post-tagging
      - chunking
    - Based on the chunking results, compare each word with dictionary words, return matched label
    - Based on the label, decide the placement of the word in a policy
    - The elements of receiver, action and purpose in policy can have multiple instances, based on which rules can be split
      - Each of them are stored in an arraylist
      - Other elements can only have on instance.
Transformation-Cont'd

- **Step Two:** Transform the abstract representation into a logical representation.
  - Adopt Answer Set Programming (ASP) as the underlying logic programming;
  - ASP is a form of declarative programming
    - It represents the search problem as a logic program whose stable models, called answer sets correspond to the solutions of the problem
    - It finds these models using an answer set solver - a system for computing stable models.
    - The syntax of a ASP rule is as: `<result> :- <precondition1>, <precondition1>, ..., <preconditionN>`. 
Transformation-Cont'd

- Based on each element of the generic policy definition, we define following ASP predicates:
  - decision(ID, EFFECT) where ID is a policy id variable and EFFECT is a policy authorization decision variable;
  - actor(D, R, O) where D, R and O are variables respectively for disseminator, receiver, and owner;
  - modality(M);
  - action(A);
  - object(OBJ);
  - purpose(P) and condition(C).

- ASP representation of generic policy is expressed as follows:
  - decision(ID, EFFECT) :- actor(D, R, O), modality(M), action(A), object(OBJ), purpose(P), condition(C).
Transformation Example

- **Step 1**
  - **Input** (Original HIPAA Policy)
    - 164.506(c)(1) A covered entity may use or disclose protected health information for its own treatment, payment, or health care operations.
  - **Output** (Generic Policy Representation)
    - (<CE, CE, CE>, may, use, phi, treatment, N/A, 164.506(c)(1), allow)
    - (<CE, CE, CE>, may, use, phi, payment, N/A, 164.506(c)(1), allow)
    - (<CE, CE, CE>, may, use, phi, healthcare operation, N/A, 164.506(c)(1), allow)
    - (<CE, CE, CE>, may, disclose, phi, treatment, N/A, 164.506(c)(1), allow)
    - (<CE, CE, CE>, may, disclose, phi, payment, N/A, 164.506(c)(1), allow)
    - (<CE, CE, CE>, may, disclose, phi, healthcare operation, N/A, 164.506(c)(1), allow)

- **Step 2**
  - **Input** (Generic Policy Representation)
    - (<CE, CE, CE>, may, use, PHI, treatment, N/A, 164.506(c)(1)(1), permit)
  - **Output** (ASP Representation)
    - decision(164506c11, permit):- actor(ce, ce, ce), modality(may), action(use), object(phi), purpose(treatment), condition(na).
Compliance-oriented Analysis

- **Methodology**
  - After two-step transformation, we have both ASP representations of HIPAA regulations and healthcare system policies
  - Consider the ASP representation of HIPAA regulations as the privacy/security property program $F$
  - Consider the ASP representation of system policies as the program $G$
  - Cast the compliance checking problem into the problem of checking whether the program $G \cup F \cup H$ has no answer sets, where
    - $H$ is the program expressing program $G$ and program $F$ has conflict decision results
    - If no answer set is found, that implies the privacy/security property $F$ is verified
    - Otherwise, an answer set returned by an ASP solver servers as a counterexample that indicates why the program $G$ does not entail $F$
Case Study

- Step 1. Choosing Healthcare System Policies
- Step 2. Policy Transformation
- Step 3. Terminology Mapping
- Step 4. Compliance Checking
Case Study - Step 1

- Example healthcare provider - OSF Healthcare
  - It is owned and operated by the Sisters of the Third Order of St. Francis, Peoria, Illinois.
  - http://www.osfhealthcare.org/

- Chosen healthcare local policy examples
  - Example 1. OSF may share your information with your doctors, hospitals or other health care providers to help them provide medical care to you.
  - Example 2. OSF may use or share your information for certain types of public health or disaster relief efforts.
Case Study – Step 2

- **Transformation**
  - *Generic policy specification based representation of policy example 1*
    - (<OSF, doctor, patient>, may, share, information, treatment, N/A, l11, permit)
    - (<OSF, hospitals, patient>, may, share, information, treatment, N/A, l12, permit)
    - (<OSF, health care providers, patient>, may, share, information, treatment, N/A, l13, permit)
  - *Corresponding ASP-based representation*
    - decision(l11, permit) :- actor(osf, doctor, patient), modality(may), action(share), object(information), purpose(treatment), condition(na).
    - decision(l12, permit) :- actor(osf, hospitals, patient), modality(may), action(share), object(information), purpose(treatment), condition(na).
    - decision(l13, permit) :- actor(osf, hcp, patient), modality(may), action(share), object(information), purpose(treatment), condition(na).
Case Study – Step 3

- Terminology Mapping
  - It entails mapping the natural language phrases in healthcare systems' policies onto the terminology used in HIPAA regulations.

<table>
<thead>
<tr>
<th>OSF Terminology</th>
<th>HIPAA Terminology</th>
</tr>
</thead>
<tbody>
<tr>
<td>OSF</td>
<td>covered entity</td>
</tr>
<tr>
<td>doctor</td>
<td>covered entity</td>
</tr>
<tr>
<td>hospital</td>
<td>covered entity</td>
</tr>
<tr>
<td>health care provider</td>
<td>covered entity</td>
</tr>
<tr>
<td>information</td>
<td>PHI</td>
</tr>
<tr>
<td>share</td>
<td>disclose</td>
</tr>
<tr>
<td>provide medical care to you</td>
<td>treatment</td>
</tr>
</tbody>
</table>
Case Study – Step 4

- **Experiment Results**
  - 1\textsuperscript{st} run: no answer set is found, which means the local healthcare policy complies with the HIPAA regulations.
  - 2\textsuperscript{nd} run: add following local healthcare system's policy with a policy ID of I12:
    - decision local(I12, deny) :- actor(osf, hospitals, patient), modality(may), action(share), object(information), purpose(treatment), condition(na).
    - ASP solver find one answer set:
      - modality(may) action(share) action(use) object(information) object(phi) purpose(treatment) condition(na) actor(osf, hospitals, patient) action(ce, ce, ce) decision local(I12, deny) decisionhipa(c11, permit)
    - It indicates a counterexample explaining the violation of HIPAA regulations
Implementation

- A transformation tool to support our two-step transformation is implemented based on OpenNLP open source project using C#
Evaluation

- Analyze **efficiency** and **scalability** of our transformation tool

- Analyze **time** consumption by **ASP**

<table>
<thead>
<tr>
<th># of Policies</th>
<th>10</th>
<th>20</th>
<th>30</th>
<th>40</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (sec.)</td>
<td>0.0128</td>
<td>0.0421</td>
<td>0.1045</td>
<td>0.3056</td>
<td>0.9171</td>
</tr>
</tbody>
</table>
Related Work

- Formalization Efforts for Regulations
  - Proposed PrivacyLFP which is an extension of Logic of Privacy and Utility (LPU)[Young’10]
  - Formalized HIPAA in a fragment of stratified Datalog with one alternation of negation, and built a prototype tool to check the lawfulness of a transmission[Lam’09].
  - Presented privacy APIs, which extends the traditional matrix model of access control, and used them to formalize two versions of HIPAA § 164.506[Map’06].

- Logics for Specifying Policies and Regulations
  - Shown how to specify future obligations from data protection policies in Distributed Temporal Logic(DTL)[Hilty’05].
  - Used an extension of LTL, Metric First-Order Temporal Logic (MFOTL) for specifying security properties[Basin’10].
  - Developed a logic for reasoning about conditions and exceptions in privacy laws[Dinesh’08].

- Requirement Analysis
  - Rigorous extraction of requirements from policies and regulations[Breaux’06][Lee’06].
  - Presented a methodology to extract access rights and obligations directly from regulation texts[Breaux’07].
  - Presented a production rule framework that software engineers can use to specify compliance requirements for software[Maxwell’10].
Conclusion

- We have articulated the necessity and importance of HIPAA-compliant healthcare systems and presented a compliance analysis framework
  - We first extracted policy patterns and defined a generic policy specification scheme.
  - Then we presented our transformation process and compliance analysis methods by leveraging logic-based reasoning techniques.

- Case study and evaluation results demonstrated the efficiency and effectiveness of our approach.
Future Work

- We would further investigate how cross-referenced policies can be analyzed in our framework.
- We would attempt to refine and enhance our framework to deal with most sections of HIPAA regulations.
- We are planning to conduct extensive evaluation of our approach.
- We would study a comprehensive policy enforcement architecture for distributed healthcare systems in clouds, with the consideration of HIPAA compliance.
Questions

The work of Ruoyu Wu, Gail-J. Ahn and Hongxin Hu was partially supported by the grants from National Science Foundation (NSF-IIS-0900970 and NSF-CNS-0831360) and Department of Energy (DE-SC0004308 and DE-FG0203ER25565).