# An Analytical Solution for Consent Management in Patient Privacy Preservation

Qihua Wang
IBM Almaden Research Center
650 Harry Road
San Jose, California, USA
qwang@us.ibm.com

Hongxia Jin
IBM Almaden Research Center
650 Harry Road
San Jose, California, USA
jin@us.ibm.com

## ABSTRACT

With the growing awareness and enforcement of patient rights, patients are empowered with increasing control on their medical information. In many situations, laws and regulation rules require the acquisition of patients' consent before one can access the patients' health data. However, in practice, patients oftentimes have difficulties determining whether they should permit or deny a certain access request. In this article, we propose an analytical approach to assist patients in the consent management of their medical information. Our consent management system employs a statistical learning method that evaluates the benefits and risks associated with access requests, so as to make personalized recommendation on consent decisions. Multiple factors are considered in the assessment process, including the importance of the request, the sensitivity of the requested information, and correlation information. We have implemented a prototype of our solution and performed evaluation with large-scale medical records.

## Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection—*Access controls*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

## General Terms

Security, Algorithms, Experimentation

## Keywords

Access control, Privacy Protection, Machine Learning, Data Analytics

## 1. INTRODUCTION

With the continuous migration to electronic healthcare systems all over the world, more and more hospitals are storing and transmitting medical information in electronic form. Regional and nationwide health information exchange (HIE) systems further promote the availability of medical data by allowing a patient's health records originated from different places to be accessed via computer networks in a centralized manner. HIE systems not only make it easier for healthcare providers to exchange medical information with each other, but also enable patients to put their medical data to work by downloading and sharing the data with online service providers. For example, patients may gain personalized services from vendors by sharing their medical data via online health management platforms, such as Google Health [7].

The increased availability and sharing of patient data brought by electronic healthcare systems calls for enhanced access control on such data. Recent laws and regulation rules such as HIPAA/HITECH [17] empower patients with more privileges to monitor the access to their medical information. In many situations, a health service provider is required to gain a patient's consent before accessing or sharing her certain medical records. While patients have been granted more control to preserve their privacy, in reality, it is oftentimes difficult for a patient to decide whether she should approve or deny a request on her medical records. On the one hand, refusing to provide the needed medical information may prevent the patient from getting the desired services. On the other hand, releasing sensitive medical information to others could lead to privacy breaches.

Consent management systems aim to assist patients in making access control decisions on their medical data. In this paper, we propose to enhance the consent management of medical information with analytical techniques. Our design rationale is to compare the *benefit* and the *risk* associated with an access request when determining whether to permit or deny it.

- The benefit of a request may be determined by the degree of *importance* of the targeted task. The more important a task is, the more benefit the patient will gain if the task is accomplished, and thus the more likely the patient is willing to provide the requested information for the task. For instance, requests from primary care physicians are highly important and should almost always be granted. In contrast, requests from a drug store with the purpose on targeted advertisement are much less critical.
- The risk of a request may be determined by two factors: *sensitivity* and *relevance*. First, the more sensitive the requested information, the more risky the request. In reality, people are more reluctant to release their sensitive medical records than giving away the less sensitive ones. Records in certain categories such as sexually transmitted diseases (STD) are generally considered to be highly sensitive. Furthermore, the sensitivity of medical records is highly personalized, as individuals may have very different opinions on how sensitive a certain record is. Second, according to the security principle of *need-to-know*, one should disclose only the information

*IHI'12,* January 28–30, 2012, Miami, Florida, USA.

that is relevant to the requestor's tasks. A request that asks for irrelevant information is thus associated with higher risk than one that queries highly relevant information.

When determining the consent decision on a request, our consent management solution compares the associated benefit and risk through a statistical learning method that combines the above three factors, that is, importance, sensitivity, and relevance. Intuitively, if a requestor requests information that is necessary to an important task, the request should be granted, because doing so will gain the patient a lot of benefit. On the contrary, if someone offering a secondary service asks for certain apparently irrelevant information, the patient will be suggested to decline the request, because giving away the requested information increases privacy risk while doing the patient little good.

The rest of this paper is organized as follows. We first present an overview of our solution in Section 2. We then provide technical details on information gathering and risk assessment in Sections 3 and 4, respectively. After that, we evaluate our solution in Section 5. Finally, we study related work in Section 6 and conclude in Section 7.

## 2. OVERVIEW

The architecture of our system is given in Figure 1. In our system, all the requests from information consumers that require patients' approval will be forwarded to the consent manager. The consent manager will evaluate every request and provide personalized suggestion on consent decision to the corresponding patient. The consent manager will then take the patient's decision, either "permit" or "deny", and act accordingly.

DEFINITION 1 (ACCESS REQUEST). An *access request* is represented as a tuple $\langle r, u, t, p\rangle$, where $r$ is the identity of the subject who issues the request, $u$ is the patient whose data is being asked for, $t$ is the type of the requested records, and $p$ is the purpose of the request.

The identity $r$ uniquely identifies a requestor in an electronic healthcare system. Each requestor may be in a certain role or group. Example roles include "primary care physicians", "insurance representatives", "CDC (Centers for Disease Control) agents", and so on. We denote $G(r)$ as the set of requestors in the same role or group with the subject $r$ in the system.

A patient's medical records may be classified into different types based on schemes such as ICD9 (the International Classification of Diseases, 9th Revision). We assume that a requestor may only ask for one type of medical information in a single request. Should multiple types of information are needed, the requestor may issue multiple requests, one for each type.

The purpose of a request may either be specified by the requestor or be automatically determined by factors such as the relationship between the requestor and the patient, the requestor's role and affiliation, and so on. For example, a request from a dentist may be labeled with the purpose "dental" by default.

Our consent management solution consists of two phases: *information gathering* and *decision recommendation*. First, the consent manager creates and maintains its knowledge bases in the information gathering phase. The details of information gathering will be given in Section 3. Second, the consent manager assesses the benefits and risks associated with access requests using the information in its knowledge bases. The approach to perform decision recommendation will be presented in Section 4.

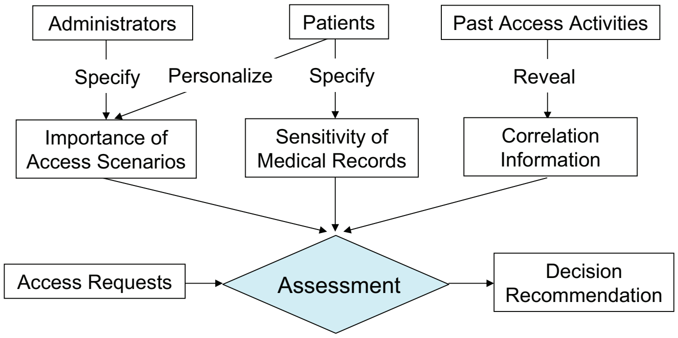

**Figure 2: Data flow in the consent manager**

## 3. INFORMATION GATHERING

Our consent management system creates and maintains three knowledge bases in the information gathering phase. The three knowledge bases, *Importance*, *Sensitivity*, and *Correlation*, correspond to the three factors that affect patients' decisions on consent management as stated in Section 1. The data flow among the components of the consent management system is shown in Figure 2.

DEFINITION 2 (IMPORTANCE MEASURE). The *importance knowledge base* is represented as a function $f_{imp} : R \times P \times U \to \mathbb{R}$, where $R$ is the set of requestor identities, $P$ is the set of purposes, $U$ is the set of patients, and $\mathbb{R}$ is the set of real numbers. In other words, the importance function $f_{imp}$ takes a requestor identity, a purpose, and a patient identity as input, and outputs a real-number importance value. The larger the returned value, the more important the corresponding scenario is.

The importance knowledge base contains information on the importance levels of various scenarios. The importance levels may be pre-defined by administrators. For example, an administrator may specify that the importance level of emergent care is high, daily medical care is medium, and secondary usage of medical records (such as research and targeted advertisement) is low. It is also possible to design a computer program that takes the patient's current medical conditions as input and automatically evaluate the severity of the situation. Important levels of requests may be affected by the roles of requestors as well. For instance, requests from doctors may be more important than those from receptionists in a hospital. We expect the importance levels to be relatively universal and may apply to most users. However, we also allow an individual user to personalize the importance levels for different scenarios if desired.

DEFINITION 3 (SENSITIVITY MEASURE). The *sensitivity knowledge base* is represented as a function $f_{sen} : U \times T \to \mathbb{R}$, where $U$ is the set of patients, $T$ is the set of medical record types, and $\mathbb{R}$ is the set of real numbers. In other words, the sensitivity function $f_{sen}$ takes a patient identity and a medical record type as input, and outputs a real-number sensitivity value. The larger the returned value, the more sensitive the medical record type is with regards to the patient.

The sensitivity knowledge base contains information on the personalized sensitivity levels of patients' various medical records. Unlike the importance levels, which are largely the same among

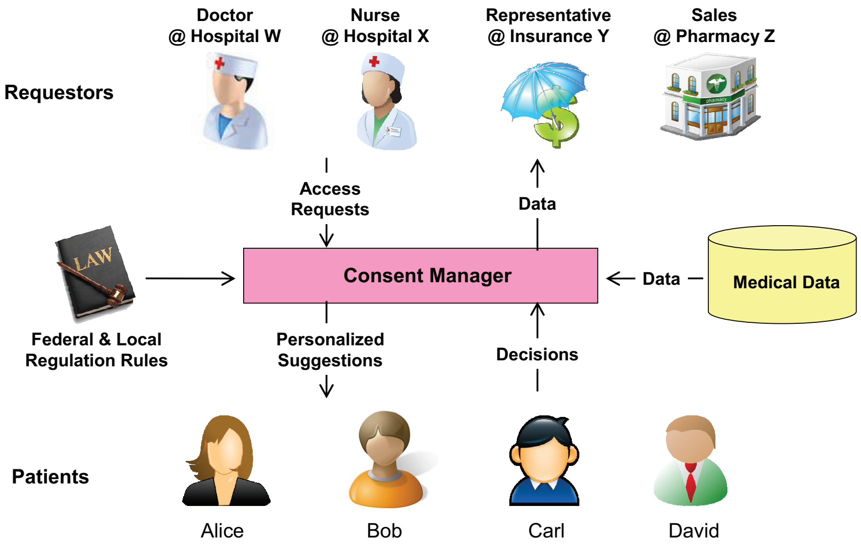

**Figure 1: System architecture of the consent management system**

all users, sensitivity is highly customized and personalized. First, different local laws and regulation rules may have different definitions on sensitive medical information. Second, people may have various opinions on how sensitive a certain medical record is. To create the knowledge base, we first assign high sensitive values to all the disease categories that are classified as sensitive by any applicable regulation rule. Furthermore, we allow every patient to specify his/her perceived sensitivity degree of each type of medical information. A patient may also update the sensitivity degree of any records at any time.

Finally, the system maintains information on how different types of medical records are related to various purposes. Because the number of combinations between medical record types and purposes is very large, it is infeasible to specify and maintain such information manually. Our consent manager logs all the access requests and consent decisions. For example, the function $f_n(r, p, t)$ returns the total number of past requests issued by the requestor $r$ on medical record type $t$ to serve the purpose $p$. For another example, the function $f_n(G(r), p, t)$ returns the total number of past requests issued by any requestor in $G(r)$ on medical record type $t$ to serve the purpose $p$, where $G(r)$ is the set of requestors in the same role/group as $r$. Such information will be used to automatically infer the correlation among the requestors, request purposes, and medical record types. Intuitively, if the record type $t_i$ is frequently requested to serve purpose $p_j$, $t_i$ and $p_j$ has strong correlation. Similarly, if many requests from $G(r)$ are on purpose $p_j$, it is likely that $p_j$ is relevant to the tasks performed by requestors in the corresponding role/group.

# 4. BENEFIT-AND-RISK-BASED CONSENT RECOMMENDATION

In this section, we study the problem of determining whether an access request should be approved or denied by a patient.

Assume that we are given an access request $X = \langle r, u, t, p \rangle$. The consent manager attempts to determine whether $X$ should be classified into *Category 0*, which represents denial, or into *Category 1*, which represents approval. In the decision process, the consent manager quantifies and compares the benefit and the risk associated with denying and approving $X$. We describe the quantified benefit or risk by a *loss function* $\lambda(i|j, X)$, where $i, j \in \{0, 1\}$. The value of $\lambda(i|j, X)$ is a real number that represents the loss incurred when the consent manager decides $X$ is in the category $i$ while $X$ should really be in the category $j$. For instance, $\lambda(0|1, X)$ is the loss when the consent manager suggests to deny $X$, while $X$ should actually be approved. We assume here that $\lambda(0|0, X) = \lambda(1|1, X) = 0$, that is, there is no loss when the consent manager classifies $X$ correctly.

With the loss function, we may compute the *expected loss value* of classifying $X$ into category $i$ ($i \in \{0, 1\}$) as follows:

$$L(i, X) = \lambda(i|1, X)P(1|X) + \lambda(i|0, X)P(0|X)$$

where $P(1|X)$ and $P(0|X)$ are the estimated probabilities that $X$ belongs to categories 1 and 0, respectively. Given a request $X$, the consent manager computes $L(0, X)$ and $L(1, X)$. If $L(0, X) \leq L(1, X)$, the consent manager suggests to deny $X$, as denial leads to less expected loss than approval; otherwise, if $L(0, X) > L(1, X)$, it recommends to approve $X$.

Next, we study the computation of $L(i, X)$. Recall that $\lambda(0|0, X) = \lambda(1|1, X) = 0$. We have

$$L(1, X) = \lambda(1|0, X)P(0|X)$$
$$L(0, X) = \lambda(0|1, X)P(1|X)$$

We need to compare $\lambda(1|0, X)P(0|X)$ with $\lambda(0|1, X)P(1|X)$. As stated earlier, a patient's consent decision is determined by the three factors, importance, sensitivity, and relevance. Here, we take importance and sensitivity into account when computing $\lambda$. Intuitively, denying a request by mistake leads to loss of benefit, while approving a request by error results in privacy risk. The more important the request $X$ is, the larger loss incurs when denying it by mistake (i.e. a larger $\lambda(0|1, X)$); the more sensitive the requested information in $X$ is, the larger loss incurs when approving it by error (i.e. a larger $\lambda(1|0, X)$). The relevance factor is then modeled by the probability $P(i|X)$. The multiplication of $\lambda$ and $P$ effectively combines the three factors that affects consent decisions. In the rest of this section, we first discuss how to estimate the probability $P(i|X)$ and then study how to compute $\lambda(j|i, X)$, where $i \in \{0, 1\}$ and $j = 1 - i$.

## 4.1 Probability Estimation

As stated in Section 1, we incline to grant access to records that are relevant to one's tasks and decline those that are irrelevant, so as to comply with the security principle of need-to-know.

Given a request $X = \langle r, p, t, u \rangle$, the conditional probability $P(1|X)$ measures the likelihood that $X$ is a relevant request, while $P(0|X)$ measures the likelihood that $X$ is an irrelevant request. We assume that such relevance information is independent of individual users, that is, $P(i|r, p, t, u) = P(i|r, p, t)$. [1] The rationale behind this assumption is two-folded:

- In reality, whether the requested information is relevant to a certain purpose is oftentimes independent of individual patients. For example, if a certain type of medical records are required to determine whether treatment $w$ is suitable for Alice, the same type of medical records are probably useful in deciding whether $w$ is suitable for Bob as well. Therefore, the independence assumption is consistent with real-world scenarios.

- From computation point of view, in practice, we may not be able to gather enough data for each individual user to correctly estimate those probabilities that are dependant on individuals. We can thus perform better estimation by relaxing the dependance on individuals. This independent assumption also reduces the complexity of parameter estimation.

According to the independence assumption and Bayes' rule, we have

$$P(i|X) = P(i|r, p, t) = \frac{P(i)P(r|i)P(p|r, i)P(t|p, r, i)}{P(r, p, t)}$$

Since $P(r, p, t)$ is common in $P(1|X)$ and $P(0|X)$, for the purpose of comparison, we just need to estimate the values of $P(i)$, $P(r|i)$, $P(p|r, i)$, and $P(t|p, r, i)$. As stated in Section 3, the consent manager logs all the requests issued by subjects in the electronic healthcare system. The logged activities are now used as observed instances in the estimation of probability values.

To begin with, we have $P(i) = f_n(i)/(f_n(0) + f_n(1))$, where $f_n(i)$ is the number of past requests in the category $i$. In other words, the value of $P(i)$ is the percentage of past requests that are in the category $i$.

As to $P(r|i)$, a straightforward solution is to have $P(r|i) = f_n(r, i)/f_n(i)$, where $f_n(r, i)$ is the number of requests issued by $r$ that are in the category $i$. However, if $r$ is a new user, we will have $f_n(r, i) = 0$, which leads to $P(r|i) = 0$ and then $P(i|X) = 0$. To address this issue, we employ a smoothing method that takes into account the past requests from requestors in the same role/group as $r$. More specifically, we have

$$P(r|i) = \frac{\alpha f_n(r, i) + (1 - \alpha) f_n(G(r), i)}{f_n(i)}$$

where $\alpha \in [0, 1]$, and $f_n(G(r), i)$ is the total number of past requests issued by the requestors in $G(r)$ that are in the category $i$.

The estimation of $P(p|r, i)$ and $P(t|p, r, i)$ are similar. We have

$$P(p|r, i) = \frac{\beta f_n(p, r, i) + (1 - \beta) f_n(p, G(r), i)}{\beta f_n(r, i) + (1 - \beta) f_n(G(r), i)}$$

where $f_n(p, r, i)$ is the number of past requests that are issued by $r$ with the purpose $p$ and are in the category $i$, and $f_n(p, G(r), i)$ is the total number of such requests that are issued by the requestors in $G(r)$. And we have

$$P(t|p, i, r) = \frac{\gamma f_n(t, p, r, i) + (1 - \gamma) f_n(t, p, G(r), i)}{\gamma f_n(p, r, i) + (1 - \gamma) f_n(p, G(r), i)}$$

where $f_n(t, p, r, i)$ is the number of past requests on record type $t$ that are issued by $r$ with the purpose $p$ and are in the category $i$, and $f_n(t, p, G(r), i)$ is the total number of such requests that are issued by the requestors in $G(r)$.

Intuitively, $P(p|r, i)$ measures how relevant (when $i = 1$) or irrelevant (when $i = 0$) the purpose $p$ is with regards to the requestor $r$'s tasks. In other words, whether $r$ is supposed to claim the purpose $p$. For example, it may be rare for a research institute to claim the purpose "emergent care". Similarly, $P(t|p, r, i)$ measures how relevant or irrelevant the medical record type $t$ is with regards to $r$'s purpose $p$. For instance, those records on sexually transmitted diseases may not be correlated to the purpose of treating knee injury.

In the computation of $P(p|r, i)$ and $P(t|p, r, i)$, we expand the past requests from $r$ with the requests from $G(r)$ through smoothing. This not only addresses the "zero count" issue for new users, but also makes it difficult for an individual requestor to manipulate the correlation information. Assume that most requestors in the system are honest. If a malicious requestor issues a large number of irrelevant queries, he might be able to increase $f_n(p, r, 1)$ and $f_n(t, p, r, 1)$. But the malicious requestor is unable to have significant impact on $f_n(p, G(r), 1)$ and $f_n(t, p, G(r), 1)$, especially when the size of $G(r)$ is large. Hence, his ability to increase the estimated $P(p|r, 1)$ and $P(t|p, r, 1)$ (which would make his requests more likely to be approved) is limited. Furthermore, if a malicious requestor issues too many irrelevant requests in attempt to affect the overall correlation information in the knowledge base, his abnormal behavior may be caught by a monitoring system due to excessive amount of requests.

Finally, as the consent management system gathers more and more data on access requests over time, it may re-estimate the above probability values periodically, so as to stay updated with the latest correlation information.

## 4.2 Loss Function Computation

As stated earlier in this section, importance and sensitivity are the two factors that determine the value of $\lambda(j|i, X)$.

On the one hand, $\lambda(0|1, X)$ measures the loss of rejecting a rightful request by mistake. Intuitively, the more important the rejected request is, the larger the loss. Also, the amount of loss is irrelevant to sensitivity in this case, because the access is denied and the requested data is not returned to the requestor. We have

$$\lambda(0|1, X) = \lambda(0|1, r, p, t, u) = f_{im}(p, r, u)$$

where $f_{im}(p, r, u)$ is the importance function defined in Definition 2. When $X$ is an important request, the cost $\lambda(0|1, X)$ of rejecting it by mistake is large. Given $P(1|X)$, a larger $\lambda(0|1, X)$ leads to a greater expected loss value $L(0, X)$ on rejecting $X$, which makes it more likely for $X$ to be approved.

On the other hand, $\lambda(1|0, X)$ measures the loss of approving an inappropriate request. Intuitively, the more sensitive the disclosed information is, the more damage the mistake causes. We have

$$\lambda(1|0, X) = \lambda(1|0, r, p, t, u) = \mu \cdot f_{sen}(t, u)$$

where $f_{sen}(t, u)$ is the personalized sensitivity function introduced in Definition 3 and $\mu$ is a real number that makes the values of $f_{im}$

[1] We will see later that the computation of $\lambda$ depends on individual users. Hence, the computation of $L(i, X)$ is still personalized.

and $f_{sen}$ comparable. When a medical record of type $t$ is sensitive for the patient $u$, the cost $\lambda(1|0, X)$ of disclosing it by mistake is large. Given $P(0|X)$, a larger $\lambda(1|0, X)$ leads to a greater expected loss value $L(1, X)$ on approving $X$, which makes it more likely for $X$ to be denied. An appropriate value of $\mu$ may be determined through training.

Finally, we may represent the above two cases in the following unified form

$$\lambda(i|j, X) = i \cdot \mu f_{sen}(t, u) + j \cdot f_{im}(p, r, u)$$

where $i + j = 1$. Note that both $f_{sen}$ and $f_{im}$ return personalized values for individual patients.

# 5. EVALUATION

We have implemented the consent management solution. In this section, we discuss the evaluation of our solution. The objective of our experiments is three-folded.

- First, we would like to evaluate the overall effectiveness of the consent manager. More specifically, whether the consent manager is able to approve most valid access requests and deny inappropriate ones.

- Second, we would like to examine how different test parameters, such as the number of requests, the percentage of malicious requestors, and the probability of over-requesting, may affect the performance of the consent manager. Such test parameters affect the amount of data and/or noise in the knowledge base of the consent manager.

- Third, we would like to see how does the consent manager perform on requestors in roles of various importance levels.

## 5.1 Experiment Design

We evaluate the performance of the consent manager through simulation on real-world medical history records. Comparing to user study, simulation has a number of advantages. Most importantly, simulation allows us to analyze the effectiveness of various solutions in different settings with limited user effort. In contrast, user study may better reflect the real-world effectiveness of our system, but the study is much more expensive to carried out than simulation. In the future, we plan to complement our findings from simulation with extensive user study.

The followings are the general steps of the simulation on our data set.

1. Generate requestors in various roles based on test parameters.

2. Separate the available medical history records into two disjoint sets: training cases and test cases.

3. Training: for each training case, randomly create a number of access requests for selected requestors based on the corresponding probability distributions; store the created access requests as well as the desired consent decisions to the correlation knowledge base of the consent manager.

4. Testing: for each test case, randomly create a number of access requests for selected requestors based on corresponding probability distributions; ask the consent manager to make a suggestion on each of the created requests; store the answer and compare it with the correct answer.

Next, we describe the design of our experiments in detail.

**Data.** Our experiments are based on the real-world medical history records provided by our hospital partners. Our dataset contains 2.9 million event entries from over 75000 patients. Each event entry is represented as a tuple consisting of three elements: patient ID, diagnosis code, and date. An entry indicates that a patient visited a hospital for a certain health issue (indicated by the diagnosis code) on a certain date. The diagnosis codes in the event entries are in ICD-9. The earliest date of the entries is in 1962, while the latest is in 2009. Note that due to some policy restrictions, we are unable to acquire the log on how healthcare practitioners accessed a patient's medical information during his/her visit. Hence, we need to simulate the access requests for experimental purposes.

**Settings.** We assume that each of the requestors in the system may be in one of the three roles $\{R_1, R_2, R_3\}$. In our experiments, the expected number of requestors in each role is set to 100, and we have $f_{im}(p, r_1, u) : f_{im}(p, r_2, u) : f_{im}(p, r_3, u) = 4 : 2 : 1$ for any purpose $p$ and patient $u$, where $r_1 \in R_1$, $r_2 \in R_2$, and $r_3 \in R_3$. In other words, requests from subjects in $R_1$ are more important than those from $R_2$, which are in turn more important than those from $R_3$. A requestor may be either *honest* or *malicious*. As we will see later, a malicious requestor has a certain probability to *over-request* patients' data (i.e. intentionally ask for information that is irrelevant to his current task). Whether a requestor is honest or malicious is determined at the very beginning of a simulation and the nature of a requestor will remain the same throughout the simulation. The percentage of malicious requestors among all requestors is a test parameter.

The types of medical records are represented in ICD-9. For each patient, we randomly select 2 ICD-9 main categories as high-sensitive, 3 as medium-sensitive, and all others as low-sensitive to the patient. The ratio of sensitivity value among the three categories is 36:6:1 in our experiments. The purpose of an access request is also an ICD-9 code, which represents the health problem the requestor attempts to treat.

**Request Generation.** To generate the access requests used in an experiment, we process the patients' visiting records one by one in time order. Given a visiting record $\langle u, v, d\rangle$ in the dataset, where $u$ is the patient, $v$ is an ICD-9 code representing the reason of the visit, and $d$ is the date of the visit, we generate access requests from the record as follows:

1. Randomly assign a number of requestors to the visit. In our experiments, we assign two requestors in each role to each visiting record.

2. For each assigned requestor $r_i$, we generate a number of access requests for the visiting record. The expected number of requests from a requestor is given as a test parameter. Assume that the expected number is $\theta$. We sample a value $k$ from a Poisson distribution with expected value $\theta$. The sample value $k$ is the actual number of requests the requestor issues for the current visiting record.

3. For each request $X$, its purpose $p$ is the same as the ICD-9 code $v$ in the current visiting record. In other words, the requestor is supposed to treat the patient on his/her current health problem.

4. For each request $X$, we determine a target medical record type $t$. If the requestor is malicious, we sample a random number to determine if the requestor would over-request in the current request. A normal request will be generated if

over-request is not conducted. Note that honest requestors always issue normal requests.

*Invalid/over request*: We randomly select an ICD-9 code that is not in the main category as $v$. The probability that a code is selected is proportional to its sensitivity value. For instance, if a problem code $w$ is 6 times as sensitive as $y$, then the probability that $w$ is selected in an over-request is 6 times as that of $y$. This models the situations that malicious requestors are more interesting in over-requesting sensitive information.

*Valid/normal request*: We sample a random number to determine the target medical record type.

- With probability 68% (i.e. the probability that a value falls into one standard deviation from the mean in a normal distribution), the target medical record type is the same as the problem code $v$ in the visiting record. In other words, there is high probability that the requested information is highly relevant to the patient's problem.
- With probability 27% (i.e. the probability that a value falls between one and two standard deviations from the mean in a normal distribution), the target medical record type is in the same main ICD-9 category as $v$ (an ICD-9 main category contains the codes that represent a number of related health issues). In other words, the requestors may ask for somewhat relevant information.
- With probability 5%, the target medical record type is in a different main ICD-9 category as $v$. In other words, there is a small probability that the request is not directly related to the patient's problem. This simulates exceptions and errors in practice. We decide to model exceptional access requests in the simulation, because exceptions are common in healthcare. Such exceptional requests introduce noises into the correlation information between purposes and requested medical record types.

5. Assume that the purpose $p$ and the medical record type $t$ have been chosen for the request $X$ in the previous steps. We create $X$ as $\langle r_i, u, t, p \rangle$.

**Training.** In our experiments, the training cases contain those visiting records before the year 2000. The training cases make up about 10% of the entire data set. For each visiting record in the training set, we generate a number of access requests using the method described earlier. For each generated access request, if it is a valid request, i.e. not generated as an over-request issued by a malicious requestor, the desired consent decision is "approve"; otherwise, if it is an over-request (also referred to as an invalid request), the desired consent decision is "deny". The access requestors together with their desired consent decisions are then given to the consent manager to create its knowledge base on correlation information.

**Testing.** The test cases contain those visiting records in and after the year 2000. The test cases make up about 90% of the entire data set. For each visiting record in the test set, we generate a number of access requests using the method described earlier. We then ask the consent manager to suggest an access decision for each of the generated access requests. Note that the consent manager does not know whether an access request is valid or not. We record the answer returned by the consent manager as well as the desired decision (i.e. "approve" for valid requests and "deny" for invalid requests). We also record the role of the requestor for each request so as to compare the consent manager's performance with regards to different roles. The consent manager then adds the test requests and the decisions to its knowledge base.

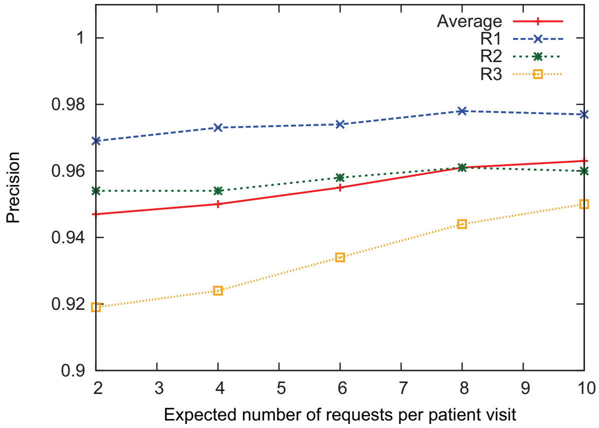

**Figure 3: Overall precisions over the number of requests per patient visit**

**Criteria.** Given a set $S$ of requests, let $A_c$ and $D_c$ be the sets of requests in $S$ that are recommended as "approve" and "deny" by the consent manager, respectively. Similarly, let $A_d$ and $D_d$ be the sets of requests in $S$ whose desired consent decisions are "approve" and "deny", respectively. The *recommendation precision* (or *precision* for short) is computed as $\frac{|A_c \cap A_d| + |D_c \cap D_d|}{|S|}$. In other words, precision is the percentage of requests in $S$ to which the consent manager suggests a decision correctly.

The *overall precision* of a simulation is the precision over all the requests in the test cases. In addition to overall precision, we measure the precision over all the valid requests and the precision over all the invalid ones. We also measure the precisions on requests from the requestors in each role. More specifically, we compute the precisions over valid and invalid requests from the requestors in $R_1$, $R_2$, and $R_3$, respectively. Such fine-grained measurements allow us to have a comprehensive understanding on the performance of the consent manager.

## 5.2 Experimental Results

In this subsection, we present our experimental results with different test parameters. In the experiments, we generally assume that a majority of the requestors, which represent healthcare providers in the real world, are honest. We also assume that the probability for a malicious requestor to over-request patients' data is relatively low, because they want to stay in business (a high percentage of over-requests is easy to be detected).

**Number of Requests.** The first set of experiments are designed to test the effectiveness of our solution over different numbers of requests issued by requestors. In the experiments, 5% of the requestors are malicious; the probability of over-requesting of any malicious requestor is 0.02; the expected numbers of requests issued by a requestor on a patient's visit range from 2 to 10. The experimental results are presented in Table 1 and the results on overall precision are visualized in Figure 3.

We start with overall performance. As we can see from Table 1, the consent manager has overall precision higher than 0.9 in all cases. Also, from Figure 3, we can see that the performance increases with larger numbers of requests. This is because more requests leads to a larger correlation knowledge base, which is beneficial to the statical learning method employed by the consent manager. But it is worth noting that the consent manager has very good

**Table 1: Performance over different numbers of requests per patient visit**

| Request Num. | Overall | | | | Valid | | | | Invalid | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | average | $R_1$ | $R_2$ | $R_3$ | average | $R_1$ | $R_2$ | $R_3$ | average | $R_1$ | $R_2$ | $R_3$ |
| 2 | 0.947 | 0.969 | 0.954 | 0.919 | 0.947 | 0.969 | 0.954 | 0.919 | 0.742 | 0.709 | 0.745 | 0.771 |
| 4 | 0.950 | 0.973 | 0.954 | 0.924 | 0.950 | 0.973 | 0.954 | 0.924 | 0.700 | 0.612 | 0.687 | 0.794 |
| 6 | 0.955 | 0.974 | 0.958 | 0.934 | 0.956 | 0.975 | 0.958 | 0.934 | 0.695 | 0.592 | 0.704 | 0.786 |
| 8 | 0.961 | 0.978 | 0.961 | 0.944 | 0.961 | 0.979 | 0.961 | 0.944 | 0.667 | 0.555 | 0.669 | 0.775 |
| 10 | 0.963 | 0.977 | 0.960 | 0.950 | 0.963 | 0.978 | 0.960 | 0.951 | 0.652 | 0.535 | 0.654 | 0.763 |

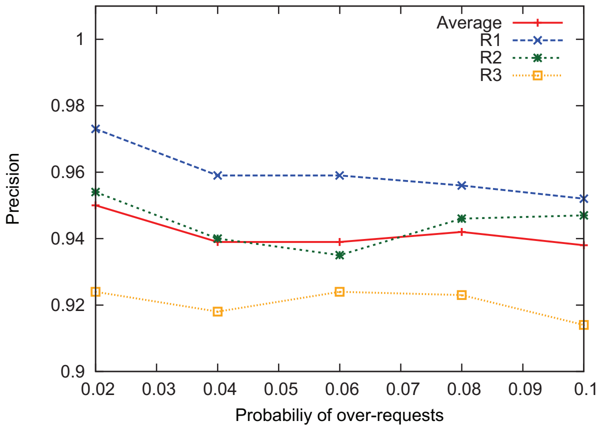

**Figure 4: Overall precisions over the probabilities of over requests**

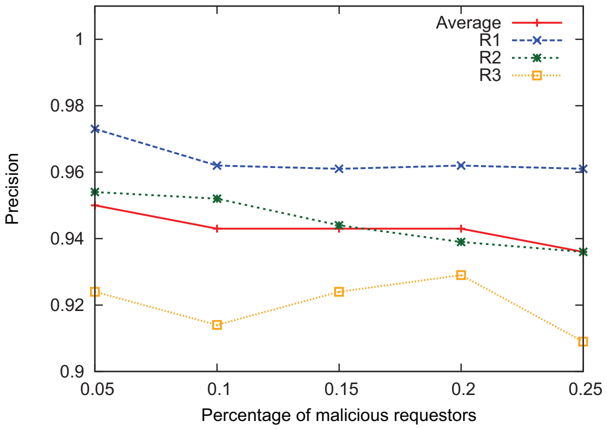

**Figure 5: Overall precisions over the percentage of malicious requestors**

performance even if the expected number of requests per patient visit is as low as 2.

Next, we consider the performance on requestors in different roles. For valid requests, the consent manager has high precision on requests from $R_1$ than those from $R_3$. In contrast, for invalid requests, the consent manager does a better job in denying over-requests from $R_3$ than those from $R_1$. The reason is that requests from $R_1$ are generally more important than those from $R_3$. The consent manager is conservative when denying requests from $R_1$, as a mistake could lead to a lot of damage. It tries to ensure that most valid requests from $R_1$ are approved, even at the cost of approving some over-requests by mistake. On the contrary, the consent manager is more aggressive in denying potential over-requests from $R_3$, as tasks perform by $R_3$ are less critical. Such tradeoffs made by the consent manager should be of best interest for most patients in the real world.

Finally, it is interesting to observe that the precision on invalid requests from $R_1$ and $R_2$ decline with a larger number of requests. An explanation is that the consent manager is increasingly conservative on requests from important requestors, after approving more and more requests from such requestors, even though some of its past approval decisions may be incorrect. This appears to be a weakness of our current statistical learning method. While being conservative on important requests may not be a bad strategy in practice, we plan to refine the learning method to address the issue in our future work.

**Probability of Over-Requesting.** The second set of experiments are designed to test the effectiveness of our solution over different probabilities of over-requesting activities performed by malicious requestors. In the experiments, 5% of the requestors are malicious; the excepted number of requests a requestor issues on a patient's visit is 6; the probabilities of over-request range from 0.02 to 0.10. The experimental results are presented in Table 2 and the results on overall precision are visualized in Figure 4.

As we can see from Table 2, the consent manager has overall precision higher than 0.9 in all cases. From Figure 4, the overall precision slightly declines with the increase of the probability of over-requesting. This is because a larger percentage of invalid requests on irrelevant records introduces noises to the consent manager's correlation knowledge base. Such noise makes the consent manager more likely to make mistakes when processing valid requests.

In contrast to valid requests, the consent manager's performance on invalid requests (especially those from $R_1$ and $R_2$) increases with a larger probability of over-requesting. The reason is that more identified over-requests in the training cases enhances the consent manager's capability in determining which requests are likely to be invalid.

**Percentage of Malicious Requestors.** The third set of experiments are designed to test the effectiveness of our solution over different numbers of malicious requestors in the system. In the experiments, the over-requesting probability of any malicious requestor is 0.02; the excepted number of requests a requestor issues on a patient is 6; the percentages of malicious requestors among all requestors range from 5% to 25%. The experimental results are presented in Table 3 and the results on overall precision are visualized in Figure 5.

As we can see from Table 3, the consent manager has overall precision higher than 0.9 in all cases. From Figure 5, the overall precision slightly declines with the increase of the number of malicious requestors. The reason behind the decline is similar to that of the previous set of experiments, as a larger number of malicious requestors leads to more invalid requests, which add noises to the correlation knowledge base of the consent manager.

We also observe that the consent manager's precision on invalid

**Table 2: Performance over different probabilities of over-requests**

| Snooping Prob. | Overall | | | | Valid | | | | Invalid | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | average | $R_1$ | $R_2$ | $R_3$ | average | $R_1$ | $R_2$ | $R_3$ | average | $R_1$ | $R_2$ | $R_3$ |
| 0.02 | 0.950 | 0.973 | 0.954 | 0.924 | 0.950 | 0.973 | 0.954 | 0.924 | 0.700 | 0.612 | 0.687 | 0.794 |
| 0.04 | 0.939 | 0.959 | 0.940 | 0.918 | 0.939 | 0.960 | 0.940 | 0.918 | 0.733 | 0.632 | 0.745 | 0.820 |
| 0.06 | 0.939 | 0.959 | 0.935 | 0.924 | 0.940 | 0.960 | 0.936 | 0.924 | 0.752 | 0.691 | 0.754 | 0.814 |
| 0.08 | 0.942 | 0.956 | 0.946 | 0.923 | 0.943 | 0.957 | 0.947 | 0.924 | 0.746 | 0.706 | 0.738 | 0.792 |
| 0.10 | 0.938 | 0.952 | 0.947 | 0.914 | 0.939 | 0.954 | 0.948 | 0.914 | 0.763 | 0.725 | 0.751 | 0.813 |

**Table 3: Performance over different percentages of malicious requestors**

| Snooper Percent | Overall | | | | Valid | | | | Invalid | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | average | $R_1$ | $R_2$ | $R_3$ | average | $R_1$ | $R_2$ | $R_3$ | average | $R_1$ | $R_2$ | $R_3$ |
| 0.05 | 0.950 | 0.973 | 0.954 | 0.924 | 0.950 | 0.973 | 0.954 | 0.924 | 0.700 | 0.612 | 0.687 | 0.794 |
| 0.10 | 0.943 | 0.962 | 0.952 | 0.914 | 0.943 | 0.963 | 0.952 | 0.914 | 0.737 | 0.654 | 0.765 | 0.795 |
| 0.15 | 0.943 | 0.961 | 0.944 | 0.924 | 0.944 | 0.962 | 0.945 | 0.924 | 0.741 | 0.674 | 0.763 | 0.786 |
| 0.20 | 0.943 | 0.962 | 0.939 | 0.929 | 0.944 | 0.962 | 0.940 | 0.930 | 0.755 | 0.712 | 0.751 | 0.799 |
| 0.25 | 0.936 | 0.961 | 0.936 | 0.909 | 0.937 | 0.962 | 0.937 | 0.910 | 0.748 | 0.706 | 0.729 | 0.810 |

requests increases when the percentage of malicious requestors increases from 005 to 0.20. This is because more identified over-requests in the training cases enrich the consent manager's knowledge on invalid requests. However, further increasing the percentage from 0.20 to 0.25 reduces the average performance on invalid requests, probably due to the excessive noise introduced by the test cases.

**Summary.** In general, our experimental results show that the consent manager maintains very good performance in various settings. It performs well even when as many as 25% of the requestors are malicious or when up to 10% of the requests from malicious requestors are invalid. We have also observed how different test parameters may affect the performance of the consent manager, and how tradeoffs are effectively made among requestors of different importance levels.

# 6. RELATED WORK

Security and privacy in healthcare has long been a popular research area. One of the most active research topics is to annonymize a set of medical records from a large number of patients, before publishing them. A number of annonymization metrics and solutions have been proposed, such as $k$-annonymization [16], $l$-diversity [12], $M$-invariance [21], and $t$-closeness [11]. Our work differs from the above work in that we focus on assisting an individual patient to manage the access to her own medical data rather than hiding patients' identity in a large amount of mixed medical records.

Access control on health data has been widely studied. Agrawal et al. [1] proposed the concept of Hippocratic Databases, which allows users to store privacy policies in the tables of relational databases and enforce them at database level. In [10], LeFevre et al. further designed a query modification approach to enforce privacy policies in a Hippocratic Database. The limitations and extensions on Hippocratic Database have been studied by Wang et al. [20]. In addition to such data management solutions as Hippocratic Databases, researchers [3, 2] have also studied how to regulate access control exceptions in healthcare using advanced policy technologies. However, none of the above work studies how to employ analytical techniques to assist individual patients on access control decisions.

There also exists work on verifying HIPAA compliance using formal methods [9, 4]. In [9], Lam et al. encoded a subset of HIPAA rules in Prolog. Given a configuration setting, one may issue queries to their logic program to check whether a certain action is HIPAA compliant or not. In their work, the truth values of logic predicates, such as whether a patient has agreed with the action, are explicitly given. They did not study how to decide whether a patient should agree to release certain medical information to a requestor.

Researchers have recently studied the consent management in electronic healthcare [14, 5, 15, 19]. In [14], Russello et al. described a framework for healthcare systems where patients are able to control the disclosure of their medical data. In their framework, context is expressed in terms of workflows. Depending on the context in which the access is being executed, different consent policies can be applied. In [15], Sheppard et al. proposed to use techniques from digital right management to control the dissemination of medical data. Unlike our solution, none of the above work applies analytical methods to perform personalized benefit and risk assessment in their consent management solutions. Furthermore, they did not conduct experiments on real-world medical data to evaluate the performance of their approaches.

In [19], Wang and Jin proposed a decision support system for consent management. They applied a simple approach to combine the three features (importance, sensitivity, and normalcy) to make suggestions on access control decisions to patients. This paper extends and improves the work in [19] by introducing more systematic approaches in quantifying various features and making decisions based on benefit and risk. For example, this work estimates the relevance among requestors, purposes, and record types, which is more comprehensive than the estimation of normalcy in [19], which performs simple counting on record types for a certain purpose. Furthermore, we employ a statistical machine learning method to systematically combine the related features, which is more sophisticated than the ad hoc combination approach in [19]. Finally, the evaluation approach in this paper is different from that in [19]. Unlike our approach in Section 5 which directly measures the accuracy of the consent manager over individual requests, the one in [19] indirectly evaluates its solution by attempting to identify malicious requestors over a large number of requests.

Our work is also related to existing work on quantified risk-based access control models [8, 6, 13, 18]. The JASON report [8] described the concepts of risk quantification and access quotas. Later, Cheng et al. [6] proposed a risk-adaptive access control solution based on the multi-level security model. In [18], Wang and Jin designed a quantified access control framework for healthcare information system. Their solution measures the risk of a batch of access requests rather than individual ones. However, none of such work studied the consent management problem in electronic healthcare systems, nor did they propose to combine multiple factors, such as importance, sensitivity, and correlation, to measure the personalized benefit and risk associated with an access request.

## 7. CONCLUSION

We have proposed an analytical approach to assist patients in the consent management of their medical information. Our system evaluates access requests based on three factors: importance, sensitivity, and relevance. Information on importance and sensitivity is specified by users, while the relevance information is automatically gathered from past access activities. We have designed a statistical learning method for our system to make personalized suggestion on access requestors for every patient by assessing benefits and risks. Furthermore, we have implemented our solution and performed simulations on real-world medical history records. Our experimental results have demonstrated the effectiveness of our solution.

## 8. REFERENCES

[1] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Hippocratic databases. In *Proceedings of the 28th International Conference on Very Large Databases (VLDB)*, 2002.

[2] C. A. Ardagna, S. Capitani Di Vimercati, S. Foresti, T. Grandison, S. Jajodia, and P. Samarati. Access control for smarter healthcare using policy spaces. In *Computers and Security*, 2010.

[3] C. A. Ardagna, S. Capitani Di Vimercati, T. Grandison, S. Jajodia, and P. Samarati. Regulating exceptions in healthcare using policy spaces. In *Proceeedings of the 22nd annual IFIP WG 11.3 working conference on Data and Applications Security*, pages 254–267, Berlin, Heidelberg, 2008. Springer-Verlag.

[4] A. Barth, J. Mitchell, A. Datta, and S. Sundaram. Privacy and utility in business processes. In *CSF '07: Proceedings of the 20th IEEE Computer Security Foundations Symposium*, pages 279–294, Washington, DC, USA, 2007. IEEE Computer Society.

[5] C. J. Bonnici and L. Coles-Kemp. Principled electronic consent management: A preliminary research framework. In *Proceedings of the 2010 International Conference on Emerging Security Technologies*, EST '10, pages 119–123, Washington, DC, USA, 2010. IEEE Computer Society.

[6] P.-C. Cheng, P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner, and A. S. Reninger. Fuzzy multi-level security: An experiment on quantified risk-adaptive access control. In *SP '07: Proceedings of the 2007 IEEE Symposium on Security and Privacy*, pages 222–230, Washington, DC, USA, 2007. IEEE Computer Society.

[7] Google Health. http://www.google.com/health/.

[8] Jason Program Office. Horizontal Integration: Broader Access Models for Realizing Information Dominance. *The MITRE Corporation*, Dec. 2004.

[9] P. E. Lam, J. C. Mitchell, and S. Sundaram. A formalization of hipaa for a medical messaging system. In *TrustBus '09: Proceedings of the 6th International Conference on Trust, Privacy and Security in Digital Business*, pages 73–85, Berlin, Heidelberg, 2009. Springer-Verlag.

[10] K. LeFevre, R. Agrawal, V. Ercegovac, R. Ramakrishnan, Y. Xu, and D. DeWitt. Limiting disclosure in hippocratic databases. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, Aug. 2004.

[11] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *In ICDE'07: Proceedings of the 23rd International Conference on Data Engineering*, 2007.

[12] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. $\ell$-diversity: Privacy beyond $k$-anonymity. In *Proc. 22nd Intnl. Conf. Data Engg. (ICDE)*, page 24, 2006.

[13] Q. Ni, E. Bertino, and J. Lobo. Risk-based access control systems built on fuzzy inferences. In *ASIACCS '10: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*, pages 250–260, New York, NY, USA, 2010. ACM.

[14] G. Russello, C. Dong, and N. Dulay. Consent-based workflows for healthcare management. In *Proceedings of the 2008 IEEE Workshop on Policies for Distributed Systems and Networks*, pages 153–161, Washington, DC, USA, 2008. IEEE Computer Society.

[15] N. P. Sheppard, R. Safavi-Naini, and M. Jafari. A digital rights management model for healthcare. In *Proceedings of the 2009 IEEE International Symposium on Policies for Distributed Systems and Networks*, pages 106–109, Washington, DC, USA, 2009. IEEE Computer Society.

[16] L. Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzz.*, 10(5):557–570, 2002.

[17] The Health Insurance Portability and Accountability Act (HIPAA). http://http://www.hhs.gov/ocr/privacy/.

[18] Q. Wang and H. Jin. Quantified risk-adaptive access control for patient privacy protection in health information systems. In *Proc. ACM Symposium on InformAtion, Computer and Communications Security (ASIACCS)*, Mar. 2011.

[19] Q. Wang and H. Jin. Decision Support for Patient Consent Management. In *Proc. IEEE International Conference on Healthcare Informatics, Imaging and Systems Biology (HISB)*, July. 2011.

[20] Q. Wang, T. Yu, N. Li, J. Lobo, E. Bertino, K. Irwin, and J.-W. Byun. On the correctness criteria of fine-grained access control in relational databases. In *Proceedings of the 28th International Conference on Very Large Databases (VLDB)*, Sept. 2007.

[21] X. Xiao and Y. Tao. M-invariance: towards privacy preserving re-publication of dynamic datasets. In *SIGMOD '07: Proceedings of the 2007 ACM SIGMOD international conference on Management of data*, pages 689–700, New York, NY, USA, 2007. ACM.