HICCUPS: Health Information Collaborative Collection Using Privacy and Security

Andres D. Molina, Mastooreh Salajegheh, Kevin Fu
Department of Computer Science
University of Massachusetts Amherst
Computer Science Building
140 Governors Drive
Amherst, Massachusetts
{amolina,negin,kevinfu}@cs.umass.edu

ABSTRACT

A recent national survey suggests that the HIPAA privacy rule has not only failed to preserve patient privacy adequately, but also has had a negative impact on clinical research. Our work suggests that researchers revisit the possibilities of homomorphic encryption and apply the techniques to secure aggregation of medical telemetry. A primary goal is to maintain the privacy of individual patient records while also allowing clinical researchers to have flexible access to aggregated information.

We discuss the preliminary design of *HICCUPS*, a distributed system that uses homomorphic encryption to allow only the caregivers to have unrestricted access to patients' records and at the same time enable researchers to compute statistical values and aggregation functions across different patients and caregivers. In the context of processing medical telemetry, we advocate *expressibility* of aggregation functions more than fast computation as a primary metric of system quality.

Categories and Subject Descriptors

K.4.1 [**COMPUTERS AND SOCIETY**]: Public Policy Issues—*Privacy*; D.4.6 [**Security and Protection**]: Information flow controls; K.4.1 [**COMPUTERS AND SOCIETY**]: Public Policy Issues—*Regulation*

General Terms

Security, Design, Legal Aspects

Keywords

Data privacy, medical telemetry, homomorphic encryption

SPIMACS'09, November 13, 2009, Chicago, Illinois, USA.

1. INTRODUCTION

A recent national survey [24] concluded that the HIPAA privacy rule was perceived by clinical researchers as having a substantial negative influence on research activities. Moreover, only a quarter of the participants perceived that the rule has enhanced patients' confidentiality and privacy despite lawmakers' efforts to protect patient privacy. Gostin and Nass argued in a follow-up paper that HIPAA's failure to protect patient privacy may be due to coverage gaps, inconsistencies, and variable interpretations of the rule [16]. As a consequence, Gostin and Nass made the case for a reform to HIPAA to enhance the way privacy is safeguarded while promoting research. They suggested that a new approach should emphasize data security; transparency and accountability; should facilitate the use of de-identified data, and should clarify the distinctions between *research* and *practice*, avoiding the current inconsistencies and variable interpretations.

We argue that, before resorting to the use of such de-identified data, the strengths and limitations of computing on encrypted data should be re-examined in the context of medical telemetry. To this end, we introduce *HICCUPS*, the preliminary design of a system that uses homomorphic encryption to cryptographically protect patient privacy while enabling more open discovery of aggregated results (Section 3). Our primary contributions in this early-stage study include:

- Motivating research to enable medical device manufacturers and health researchers to increase availability of statistical data while protecting patient privacy and minimizing the potential exposure of large collections of sensitive information.
- Introducing an application of homomorphic encryption schemes for computing statistics on aggregated medical telemetry.

In addition, we suggest a way in which *HICCUPS* could facilitate the proper recording of pacemaker and ICD malfunctions in a way that researchers could access without compromising the privacy of patients (Section 5). We believe that because of the delay-tolerant nature of many studies using medical data, *expressibility* of aggregation functions will serve as a more important evaluation metric than speed of computation so long as computation demands are not excessive.

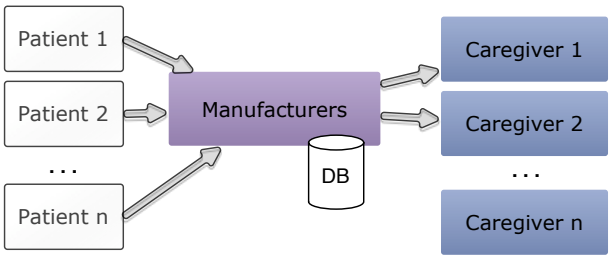

Figure 1: The figure illustrates the data flow that prevails currently. Patients upload their telemetry to the manufacturer's servers, from which it is made available to caregivers for analysis.

1.1 Complexities of Aggregating Medical Data

Today, preserving privacy of aggregated medical data often focuses on manual removal of personal information or rigid processing of data by highly trusted information brokers. While such systems can work well on a small scale, drawbacks include the ad-hoc nature of manual redactions and the trusting approach to putting all power in the hands of a single entity that could be compromised. Our approach to secure aggregation instead maximizes privacy of individual data without reducing the utility of aggregate information that could identify trends and system-wide causes of disease, procedural mistakes, or device malfunction.

Medical telemetry generated by home monitoring is an example of an instance in which patient privacy could be at risk due to current practices that do not limit the amount of data given to device manufacturers. There is also no formal infrastructure to allow clinical researchers to gain access to key pieces of data of a statistical nature. The trend toward remote monitoring allows patients with implanted devices to ensure proper functioning without the inconvenience of having to travel frequently to a clinical office. The current mechanism employed to make this remote monitoring possible involves sending all of the medical telemetry *directly to the manufacturers* which then make it available to a patient's caregiver remotely (Figure 1).

Patient privacy could be greatly improved over present-day methods of data collection. For instance, the data distribution model need not allow manufacturers to have access to all the data in a device in order to provide service to the patient. In fact, making it easier to access de-identified data, as proposed by Gostin and Nass [24], may still pose unanticipated problems. For example, Biel et al. [3] showed that it is possible to identify an individual in a predetermined group by using ECG data. Moreover, centralized locations that even briefly have access to unencrypted data create single points of failure where entire national databases could be compromised by clever hackers or conspiring insiders. As an alternative, *HICCUPS* proposes that both device manufacturers and clinical researchers be able to obtain the information that they need about the data, via computing aggregate functions on encrypted data.

1.2 Background on Homomorphic Encryption

Secure aggregation keeps input values encrypted in databases. Moreover, inputs are never decrypted—enabling a strong notion of privacy. But what use is input if it is never decrypted? Modern cryptography allows for computation on *encrypted* values. This seemingly impossible idea is made possible by a 1970s technique known as homomorphic encryption [26] that became popular in the 1990s for secure tallying of encrypted votes [25, 12]. Algorithms from number theory allow arithmetic on encrypted data.

The beauty of homomorphic encryption is that the information aggregator need not be trusted. That is, the aggregator could not easily violate individual patient privacy in a mathematically provable sense. The scheme would prevent rogue insiders from violating privacy and would prevent accidental leakage of private information. The tradeoff is that homomorphic encryption requires sophisticated computation on a modern computer, which we believe is feasible on commodity hardware for workloads common to medical telemetry.

The potential of computing on encrypted data has promising theoretical results, especially after the recent findings of Gentry [12] that prove fully homomorphic encryption schemes can be implemented using lattices. This discovery suggests that the research community may be close to extending the capabilities of this technique to essentially allow arbitrary computations on encrypted data.

2. MODELING ACCESS TO AGGREGATED MEDICAL TELEMETRY

This paper focuses on the problem of making medical telemetry available to medical device manufacturers and clinical researchers in a way that protects patient privacy. This section lists some desired properties for a solution to this problem.

Our assumption is that direct caregivers need access to all of a patient's medical data in order to perform proper treatment. Under this assumption, there is an unrestricted data flow between patients and their direct caregivers. In such a model, multiparty computation techniques can be applied to allow distinct caregivers to compute collective *answers* to queries posed by clinical researchers and manufacturers (Figure 2).

For simplicity, we will avoid the problem of a patient having multiple caregivers in this model. In principle, multiple associations could lead to false aggregates due to duplication. However, such a situation can be addressed by requesting that each patient has only one caregiver that reports data on his behalf.

The queries required by researchers and manufacturers can be classified into the following types:

1. **Selective individual disclosure: Queries to obtain a list of records.** This type of query would present the problem of returning a set of records that match a set of criteria from data distributed across the set of caregivers $\{\mathcal{C}_1, \mathcal{C}_2, \ldots \mathcal{C}_n\}$. The result of such a query can be seen as a matrix, the rows of which are the union of the rows of the sub-matrices that each of the caregivers returns to the query posted. For example, a query could request the age, gender and number of critical events in a given month for all the patients that have an ICD and that use a home monitor nationwide. Then each caregiver C_i would return a matrix (query table) with three columns and as many rows as patients are being treated by C_i. The final result to the query should be a union of all these records.

Thus the result can be thought of as a matrix with the same three columns and with as many rows as the total number of relevant records found nationwide.

2. **Queries to obtain aggregated disclosures.** This type of query would present the problem of computing an aggregate function on data distributed across the set of caregivers $\{\mathcal{C}_1, \mathcal{C}_2, \ldots \mathcal{C}_k\}$ while preserving the privacy of the data between any two different caregivers.

The main focus of this paper is to design a system using homomorphic encryption to address the queries of the second type (aggregation), which are posed when clinical researchers and device manufacturers need to compute aggregates and other similar statistical information. We recognize that the objectives for obtaining the data may differ between clinical researchers and device manufacturers, and it is important to note that this model requires that both researchers and manufacturers specify clearly and openly the kind of information that they require.

Answering a query of the first type requires a different approach. Further discussion of this point appears in related work (Section 7).

2.1 Threat Model

The goal of *HICCUPS* is to prevent unnecessary disclosure of large collections of medical telemetry. We assume that locations with transient access to plaintext records (e.g., caregivers) are relatively small collections such that a compromise will result in only localized disclosure of information rather than system-wide, catastrophic disclosure. The threat model for *HICCUPS* considers both external and internal adversaries. For instance, an external hacker who gains unauthorized access to a machine should cause at worst a localized disclosure of patient information. An external entity should not be able to cause a catastrophic disclosure of the entire collection. Potential insiders include medical researchers and aggregators. When these players follow the established protocol, we expect proper availability of information. However, if the players act maliciously they should not be able to compromise the entire system but at worst delay the availability of information. In our model, caregivers are fully trusted by their corresponding patients. That is, caregivers are not considered potential insiders but are potential targets of external adversaries.

2.2 HICCUPS Desired Properties

In order to answer queries for aggregated information, it is necessary to solve the problem of computing aggregates from encrypted values using the public key $\mathcal{R}_p$ of a researcher $\mathcal{R}$ by caregivers $\{\mathcal{C}_1, \mathcal{C}_2, \ldots \mathcal{C}_k\}$. Desirable properties for such a solution include:

1. **Anonymity of Data Provider**. Given a set of ciphertexts $\{\mathrm{Enc}_{R_p}(a_1), \mathrm{Enc}_{R_p}(a_2), \ldots, \mathrm{Enc}_{R_p}(a_k)\}$ provided by a set of entities $\{\mathcal{C}_j\}$, the probability of determining that $\mathrm{Enc}_{R_p}(a_i)$, for a given i, was computed by $\mathcal{C}_j$ for some j should differ by a negligible quantity from guessing this association.

2. **Distributivity**. It is important to emphasize that a proposed solution should be implemented using a distributed model as opposed to a centralized model.

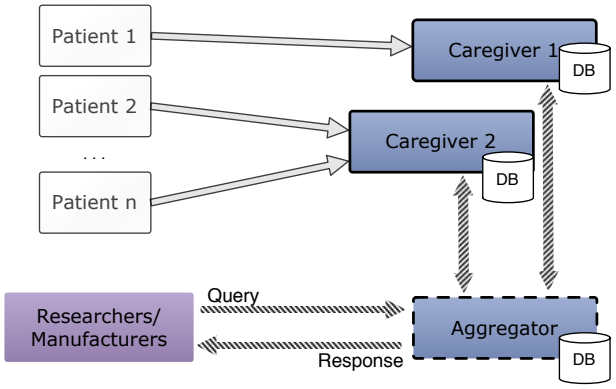

Figure 2: A researcher or manufacturer needs to compute an aggregate function from data across various caregivers. The query can be handled by an aggregator chosen among the caregivers that computes on encrypted data.

A centralized approach would give too much power to the holder and would create a single point of failure. As discussed by Jefferson et al. [20] centralized systems introduce several privacy risks in voting systems [19] for example.

3. **Semantic security**. A final implementation of *HICCUPS* should be done using an encryption system that is semantically secure to avoid chosen plaintext attacks, as will be discussed in more detail later (Section 7.3).

3. DESIGN OF HICCUPS

Homomorphic encryption schemes have for decades been successfully applied to several problems exploiting the fact that at least addition or multiplication commute with the encryption algorithm.

This section shows how having an encryption scheme with the homomorphic property may provide an answer to the problem of computing aggregate functions with data that is distributed among various caregivers. Furthermore, some sample aggregation functions that can be computed using the homomorphic property are discussed (Section 4). Finally, how having a singly homomorphic encryption allows us to compute interesting functions is discussed. The number of functions that can be computed using this technique is dramatically extended if instead a fully homomorphic encryption is used.

Unless stated otherwise, we assume the existence of a public key infrastructure with a semantically secure homomorphic encryption scheme with key generation algorithm, encryption algorithm and decryption algorithm (Gen, Enc, Dec respectively).

Under the assumption that direct caregivers should have access to all of a patient's medical data in order to perform proper treatment, it is possible to propose a model in which patients give all of their medical data to their caregivers. In this model, there could be various patients' data clustering around caregivers. The tasks of computing aggregates across caregivers in order to answer the questions of manufacturers

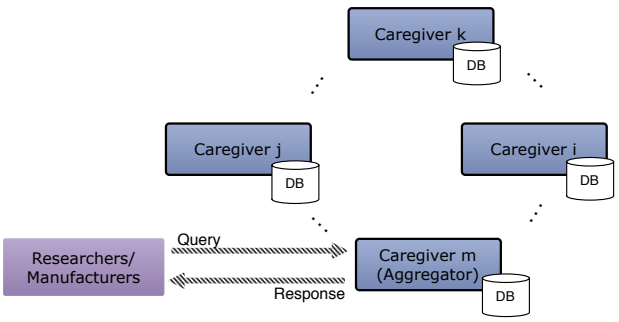

Figure 3: The aggregator is chosen at random to eliminate the probability of a compromised aggregator systematically leaking data. The rest of the caregivers compute sub-aggregates which can be combined by the aggregator to produce a total aggregate for the manufacturers and researchers.

and researchers could then be thought of as a multiparty computation (Figure 3).

Let us consider the problem of computing a publicly known aggregate function f for a variable x with data distributed among a set of caregivers $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$ from sub-aggregates $a_1, a_2, ..., a_n$, pre-computed by the caregivers $\mathcal{C}_i$. That is, $f(a_1, a_2, ..., a_n)$ is the aggregate value needed for the publicly known function f. In Section 4, we show how such aggregates can be combined to compute functions such as sample mean, sample variance, maxima, linear regression, and sample correlation.

A global aggregate could be computed as follows:

1. **Request for an aggregate**. A researcher $\mathcal{R}$ interested in computing a global aggregate submits the request specifying one of the possible aggregate functions f. For example, the function could be a simple aggregate sum $f(x_{ij}) = \sum_{i,j} x_{ij}$, over a defined set of values x_{ij} distributed among all the set of participant caregivers $\{\mathcal{C}_i\}$.

2. **Selection of an aggregator**. All the caregivers run a distributed algorithm to determine a random aggregator within the set of participant caregivers $\{\mathcal{C}_i\}$. At the end of this step, one of the participant caregivers is designated the aggregator for the request. We will denote this aggregator by $\mathcal{A}$. Note that having a fixed aggregator could lead to an attack in which the privacy guarantees could be reduced to essentially those of handing over the sub-aggregates per caregiver directly to the researcher $\mathcal{R}$. This may be undesirable, for example, if a query is asking for the age of the oldest patient in a group distributed nationwide. By exposing sub-aggregates, the researcher would obtain not only the age, but also potentially the name of the patient's caregiver. The selection of an aggregator randomly can be done using distributed techniques similar to those proposed by Kapron, et al. [21].

3. **Computation of sub-aggregates**. Each caregiver $\mathcal{C}_i$ receives the request and computes its corresponding sub-aggregate a_i and encrypts it first using the public key of the researcher $\mathcal{R}_p$, and then using the public key of the aggregator $\mathcal{A}_p$. The caregiver $\mathcal{C}_i$ can then use, for example, a bulletin board style protocol to share the encrypted result $\text{Enc}_{\mathcal{A}_p}(\text{Enc}_{\mathcal{R}_p}(a_i))$. Coming back to our example, the caregivers would create the a_is such that $\sum_i a_i = \sum_{i,j} x_{ij}$, but would not send the a_is unencrypted.

4. **Unwrapping**. The caregiver $\mathcal{A}$ chosen to compute the aggregate obtains the encrypted values $\{\text{Enc}_{\mathcal{R}_p}(a_i)\}$ by decrypting the first layer of $\text{Enc}_{\mathcal{A}_p}(\text{Enc}_{\mathcal{R}_p}(a_i))$ for each i. This is needed to ensure that only the designed aggregator $\mathcal{A}$ is able to compute the aggregate. A complimentary technique can be used to ensure participation; for example, the outer wrap can be signed by the corresponding caregiver.

5. **Aggregation**. The caregiver $\mathcal{A}$ computes $f^*(\text{Enc}_{\mathcal{R}_p}(a_i))$, where f^* can be obtained from f using the homomorphic property. Subsequently $\mathcal{A}$ destroys each individual $\text{Enc}_{\mathcal{R}_p}(a_i)$. Implicitly, this assumes that the majority of the caregivers are honest. If that were not the case, and the majority of the caregivers were willing to forward the individual $\text{Enc}_{\mathcal{R}_p}(a_i)$ then the probability of falling victim to an attack like the one in the case of having a fixed aggregator could not be considered negligible. However, this assumption may not be unrealistic, given that if the majority of the caregivers were dishonest, there would potentially be greater privacy concerns. Also note that the aggregator was not in the position to learn anything about the sub-aggregates from the other caregivers since the sub-aggregates were encrypted using the researcher's public key. In our example f^* corresponds to adding encrypted aggregates with the operation $\oplus$, while f corresponds to addition of plaintexts, thus $f^*(\text{Enc}_{\mathcal{R}_p}(a_i)) = \oplus_i \text{Enc}_{\mathcal{R}_p}(a_i)$.

6. **Handing over**. The aggregator $\mathcal{A}$ returns the encrypted aggregate value $\text{Enc}_{\mathcal{R}_p}(f(a_i))$ to the researcher $\mathcal{R}$, which can be decrypted using its corresponding secret key $\mathcal{R}_s$.

Note that in an alternative approach this aggregation could be done by encrypting each of the values $\{\text{Enc}_{\mathcal{R}_p}(x_{i,j})\}$ distributed among caregivers $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$, where $x_{i,j}$ is a value known to $\mathcal{C}_i$ and j ranges from $1, \ldots, n_i$ the sample size in $\mathcal{C}_i$. However, in the case of computing aggregates for medical data nationwide this could potentially involve transmitting millions of values to the aggregator $\mathcal{A}$, instead of sending only one sub-aggregate a_i per caregiver. Thus, the proposed approach could potentially avoid a large unnecessary overhead.

4. COMPUTING AGGREGATES

This section gives some examples of how multiple aggregates computed as described in Section 3 can be combined to calculate common statistical functions. The ability to compute these aggregates is dependent upon whether a singly homomorphic encryption scheme or a fully homomorphic scheme is used. That is, whether one or two operations with ciphertexts with the homomorphic property are performed.

4.1 Computing Aggregates Using a Singly Homomorphic Encryption Scheme

Consider a homomorphic encryption scheme that is IND-CPA secure with key generation algorithm, encryption algorithm and decryption algorithm (Gen, Enc, Dec respectively) and with semigroups of plaintexts and ciphertexts $\mathbf{M}$ and $\mathbf{C}$ respectively. That is, the homomorphic property is such that for $m_1, m_2 \in \mathbf{M}$ an a pair of public and secrete keys $\mathcal{A}_p, \mathcal{A}_s$

$$\mathrm{Enc}_{\mathcal{A}_p}(m_1 + m_2) =_p \mathrm{Enc}_{\mathcal{A}_p}(m_1) \oplus \mathrm{Enc}_{\mathcal{A}_p}(m_2)$$

where $\oplus$ is the corresponding operation to $+$ on $\mathbf{C}$ and $=_p$ denotes indistinguishability of distributions, that is, an adversary would not be able to tell that $\mathrm{Enc}_{\mathcal{A}_p}(m_1+m_2)$ and $\mathrm{Enc}_{\mathcal{A}_p}(m_1) \oplus \mathrm{Enc}_{\mathcal{A}_p}(m_2)$ correspond to the same plaintext, however

$$\mathrm{Dec}_{\mathcal{A}_s}(\mathrm{Enc}_{\mathcal{A}_p}(m_1+m_2)) = \mathrm{Dec}_{\mathcal{A}_s}(\mathrm{Enc}_{\mathcal{A}_p}(m_1) \oplus \mathrm{Enc}_{\mathcal{A}_p}(m_2))$$

A randomly selected caregiver computes an aggregate such as the sample mean and variance of a sample of values $\{x_{i,j}\}$ distributed among caregivers $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k$, where $x_{i,j}$ is a value known to $\mathcal{C}_i$ and j ranges from $1, \ldots, n_i$, the sample size in $\mathcal{C}_i$. This procedure does not require the actual knowledge of the values $\{x_{i,j}\}$. Instead the knowledge of appropriate encrypted sub-aggregate values suffices, as will be shown next.

In this case, the mean of $x_{i,j}$ over all $\mathcal{C}_i$s is given by

$$\bar{x} = \frac{\sum_{i=1}^{k} \sum_{j=1}^{n_i} x_{i,j}}{\sum_{i=1}^{k} n_i}$$

And the variance of the sample is given by

$$s^2 = \frac{\sum_{i=1}^{k} \sum_{j=1}^{n_i} (x_{i,j})^2 - \frac{(\sum_{i=1}^{k} \sum_{j=1}^{n_i} x_{i,j})^2}{\sum_{i=1}^{k} n_i}}{(\sum_{i=1}^{k} n_i) - 1}$$

Now, if we denote $a_i = \sum_{j=1}^{n_i} x_{i,j}$, and $b_i = \sum_{j=1}^{n_i} (x_{i,j})^2$ then the formulas can be rewritten as:

$$\bar{x} = \frac{\sum_{i=1}^{k} a_i}{\sum_{i=1}^{k} n_i}$$

and

$$s^2 = \frac{\sum_{i=1}^{k} b_i - \frac{(\sum_{i=1}^{k} a_i)^2}{\sum_{i=1}^{k} n_i}}{(\sum_{i=1}^{k} n_i) - 1}$$

Thus, given the additive homomorphic scheme, researcher $\mathcal{R}$ can compute these aggregates preserving privacy as follows: In order to compute aggregates such as the mean and the variance, the aggregator $\mathcal{A}$ first collects the encrypted values of a_i, b_i, n_i (using the public key, $\mathcal{R}_p$ of the researcher $\mathcal{R}$) from the caregivers. More explicitly, if each of the $\mathcal{C}_i$s provides $\mathrm{Enc}_{\mathcal{R}_p}(a_i)$, $\mathrm{Enc}_{\mathcal{R}_p}(b_i)$, and $\mathrm{Enc}_{\mathcal{R}_p}(n_i)$, then the aggregator $\mathcal{A}$ computes $a = \bigoplus_{i=1}^{k} \mathrm{Enc}_{\mathcal{R}_p}(a_i)$, $b = \bigoplus_{i=1}^{k} \mathrm{Enc}_{\mathcal{R}_p}(b_i)$, and $n = \bigoplus_{i=1}^{k} \mathrm{Enc}_{\mathcal{R}_p}(n_i)$ and sent these three values to researcher $\mathcal{R}$. Finally, the researcher $\mathcal{R}$, using the corresponding secret key $\mathcal{R}_s$, would simply compute:

$$\bar{x} = \frac{\mathrm{Dec}_{\mathcal{R}_s}(a)}{\mathrm{Dec}_{\mathcal{R}_s}(n)}$$

since we know that

$$\bigoplus_{i=1}^{k} \mathrm{Enc}_{\mathcal{R}_p}(a_i) =_p \mathrm{Enc}_{\mathcal{R}_p}(\sum_{i=1}^{k} a_i).$$

Similarly,

$$s^2 = \frac{\mathrm{Dec}_{\mathcal{R}_s}(b) - \frac{\mathrm{Dec}_{\mathcal{R}_s}(a)}{\mathrm{Dec}_{\mathcal{R}_s}(n)}}{\mathrm{Dec}_{\mathcal{R}_s}(n) - 1}.$$

These ideas can be extended to compute a linear regression $y = \beta_0 + \beta_1 x$ and a sample correlation r_{xy}. That is, since the problem of estimating β_0, and β_1 can be obtained from the sums: $\sum x, \sum x^2, \sum y$, $\sum y^2$, and $\sum xy$ as follows:

$$\beta_1 = \frac{\sum y \cdot \sum x - n \cdot \sum xy}{(\sum x)^2 - n \cdot (\sum x^2)}$$

and

$$\beta_0 = \frac{\sum x \cdot \sum xy - \sum y \sum x^2}{(\sum x)^2 - n \cdot x^2}$$

and

$$r_{xy} = \frac{n \cdot \sum xy - \sum x \sum y}{\sqrt{n \cdot \sum x^2 - (\sum x)^2} \sqrt{n \cdot \sum y^2 - (\sum y)^2}}$$

It may also be useful to compute maxima or minima. The following shows how to compute maxima; the computation of minima is completely analogous. While this described method is not optimal, it serves to illustrate the feasibility of computing such functions using aggregates.

In order to compute a global maximum for a variable among a set of caregivers $\{C_i\}$, the problem must first be redefined in a convenient way. In particular, it will be necessary to have an idea of the range $(r_{\min}, r_{\max})$ and precision d for these values. For example, if it were required to find the maximum temperature for all the patients meeting certain conditions, one could specify the range to be between 35 and 48 degrees Celsius. Also, precision may need to be obtained up to one decimal place, for instance.

Given these two values, one can define a vector with l entries where l is equal to the number of intervals of size d in the interval $(r_{\max}, r_{\min})$, or simply $l = \frac{r_{\max} - r_{\min}}{d}$ if $r_{\min}, r_{\max} \in \mathbb{Z}$. Then each of the entities in $\{\mathcal{C}_i\}$ can return a vector

$$v_i = (\mathrm{Enc}_{\mathcal{R}_p}(c_0), \ldots, \mathrm{Enc}_{\mathcal{R}_p}(c_j), \ldots, \mathrm{Enc}_{\mathcal{R}_p}(c_{l-1}))$$

where $c_j = 1$, if $r_{\min} + c_j$ is the maximum value within the caregiver $\mathcal{C}_i$'s data, and $c_j = 0$ otherwise.

By adding the vectors $\{v_i\}$ across caregivers, we obtain a vector $v = \sum_i v_i$ such that the last non-zero entry of the decrypted vector $\mathrm{Dec}_{\mathcal{R}_s}(v)$ (decrypted entry by entry) corresponds to the global maximum.

This approach would require the computation of l sum aggregations, and, therefore, the complexity of this algorithm increases linearly with the number of possible values for the maximum. This complexity can be greatly reduced by using a typical tree-like approach to determine if the maximum is on the *left* or on the *right* of a given interval. That is, one can run this aggregation scheme with only two possible values for the maximum, implementing it so that the participant caregivers return either one or the other as being

closer to the maximum. After one side has been chosen, the algorithm would be used recursively on this selected subinterval to again determine if the maximum is on the *left* or on the *right* side of the interval until the desired precision has been achieved. While this approach would require a logarithmic number of sum aggregations on l, there would be an overhead in the communication caused by multiple protocol interactions. Therefore, a detailed complexity analysis of both methods would have to be evaluated through implementation. This analysis could be undertaken in future work.

4.2 Computing Aggregates Using a Fully Homomorphic Encryption Scheme

As shown in section 4.1, some interesting functions can be computed when the homomorphic scheme holds for one operation. Let us now consider a fully homomorphic encryption scheme that is IND-CPA secure with key generation algorithm, encryption algorithm and decryption algorithm (Gen, Enc, Dec respectively) and with sets of plaintexts and ciphertexts $\mathbf{M}$ and $\mathbf{C}$ respectively with ring structures. Then, the homomorphic properties are as follows: For $m_1, m_2 \in \mathbf{M}$ and a pair of public and secret keys $\mathcal{A}_p, \mathcal{A}_s$,

$$\mathrm{Enc}_{\mathcal{A}_p}(m_1 + m_2) =_p \mathrm{Enc}_{\mathcal{A}_p}(m_1) \oplus \mathrm{Enc}_{\mathcal{A}_p}(m_2)$$

and

$$\mathrm{Enc}_{\mathcal{A}_p}(m_1 \cdot m_2) =_p \mathrm{Enc}_{\mathcal{A}_p}(m_1) \odot \mathrm{Enc}_{\mathcal{A}_p}(m_2),$$

where $(+, \cdot)$ are the operations in $\mathbf{M}$, and $(\oplus, \odot)$ are the corresponding operations on $\mathbf{C}$. And, $=_p$ denotes indistinguishability of distributions.

Then, as was briefly mentioned earlier, having this structure expands dramatically the type of functions that can be computed. In particular, one can see that matrix multiplication on matrices with encrypted data is easy to compute. This would allow the computation of multiple regression models, for example. The challenge, however, is to eliminate the need for communicating the large matrices to an aggregator, and instead, think of subdividing the problem into smaller problems that each of the caregivers can compute separately.

5. PRACTICAL APPLICATIONS

This section presents some real-world telemetry applications that would benefit from the implementation of *HICCUPS*. These examples show that even basic statistical aggregations could solve important problems when they are computed on data-sets that are distributed among multiple institutions nationwide and that may contain private information.

5.1 Pacemaker and ICD Reliability Studies

The way in which pacemaker and ICD malfunctions are recorded does not allow for the determination of the frequency of failure of a particular device model. That is, while there have been studies that attempt to measure the reliability of these devices, the studies are limited by the fact that malfunctions are only included in the records if the malfunction results in a device replacement.

For example, in 2006, cardiologist Dr. William Maisel [22] published a meta analysis of pacemaker and ICD registries to assess the rates of pacemaker and ICD malfunctions in order to be able to identify trends in the reliability of these devices. The analysis included data from various international registries. One of the most important limitations of the analysis as described by the author was that the malfunctions of these devices are underreported because most often the malfunctions are reported only when a replacement is made. Therefore, the study could not determine the true clinical implications of device malfunctions that did not require device replacement.

Another similar study also aimed at counting the malfunctions of pacemakers and ICDs examined data from multiple years included in the annual reports of the Federal Drug Administration of the US [23]. The authors of this study pointed out that the database registries that monitor the performance of these devices are limited primarily by their small size or by their voluntary nature. Additionally, while manufacturers do provide performance reports, historically these have not contained comprehensive information about the number of device malfunctions or the rate of or reasons for malfunction. The authors noted yet again that the study only accounted for malfunctions that were significant enough to warrant device replacement. One of the conclusions of this study was that the FDA should require more thorough monitoring of postmarket performance by manufacturers for selected devices, including pacemakers and ICDs.

Both of the studies cited above concluded that ongoing surveillance of pacemaker and ICD performance should be required. We argue that a system like the one described in this paper could allow institutions such as the FDA to conduct reports that provide a more detailed aggregation of device malfunctions including device model numbers and types of malfunction.

Additionally, the data-sets made available using this model could be larger, and the aggregation could potentially be computed more frequently and in an automated way. Furthermore, if manufacturers published their parameters for identifying a malfunctioning device without extracting them from the patient, then not only the caregiver but also the FDA, would be able to react to reliability or safety issues. For example, the FDA would be in a better position to issue recall information.

5.2 Identifying the Impact of Low Income on Preterm Birth Risk

This year the Center for Democracy and Technology in the US published a document to encourage the use of de-identified and anonymized health data and to rethink the protection of this data by regulations such as the HIPAA Privacy Rule [27]. This document notes that one of the common uses for this kind of data is research. In particular, the document mentions as an example that de-identified data has been used to perform research on the prevention of premature births.

One such research study, performed by DeFranco, et al. studied the effect of living in a socioeconomically deprived area on the risk of preterm births [10]. In this study, a number of counties in Missouri were identified as being below the US poverty line based on census information. These counties were then classified according to various levels of poverty. The number of pregnancies that resulted in various periods of preterm birth were counted using de-identified records, and the aggregates were analyzed. The study con-

cluded that women residing in socioeconomically deprived areas are at an increased risk of having a preterm birth, above other underlying risk factors.

We argue that our system could help in the validation or extension of this analysis to regions all over the nation. That is, the homomorphic techniques described in this paper could allow researchers to aggregate the number of preterm births in various counties across the nation that are below the US poverty line. These aggregates could also be computed for different preterm birth periods and then analyzed to determine the impact of low income conditions.

6. DEFINING EVALUATION METRICS

As mentioned early in this paper, this work is a preliminary attempt to learn the problems and limitations that arise when trying to employ homomorphic encryption techniques to aggregate medical telemetry. It is tempting to ask questions regarding the performance of a system implementing these techniques. This section would like to argue that finding good performance *metrics* for evaluating a solution to this problem is in itself an achievement.

It is well known, for instance, that in biomedical image analysis, processing a single fMRI brain image may take a few hours or even more if processing the image requires tasks such as manual skull stripping or manual selection of areas of interest by an expert. At other times, gathering the appropriate medical data to perform research involves contacting multiple institutions and having different IT departments gather the data according to multiple parameters. This task would probably take days at best. Therefore, running time of homomorphic encryption likely should not serve as the primary metric of quality given that instantaneous results are not expected.

6.1 The Case for Expressibility

The last observation suggests that if gathering medical telemetry for research purposes using homomorphic techniques does not require days of processing, then computation time may not be the only performance metric, or even an important one.

In fact, we believe that determining the extent to which a system like *HICCUPS* is useful should involve measuring its *expressibility*. That is, if a system allows the computation of only one type of aggregate, then the capability to express an aggregate problem is limited. On the other hand, if a system allows the computation of *most* types of aggregates, or even further, *most* types of functions on aggregates, this would mean that the capability of expressing an aggregate problem is high.

From this perspective, we recognize that this paper just scratches the surface in an attempt to determine the expressibility of *HICCUPS*. That is, while throughout this paper we have discussed the various advantages that would result from having a system such as *HICCUPS*, we have also pointed out that there are essentially two major limitations.

The first of these is that researchers or institutions that require the computation of aggregates must define the kind of aggregates that they need beforehand. We acknowledge the dynamic nature of science, and recognize that it is possible that the types of questions that researchers need to answer may change substantially. For this reason, we feel that it would be beneficial to characterize the nature of the differences between the questions that health researchers ask. Additionally, it would be compelling to determine the extent to which it is possible to define a framework that would enable the formulation of all of these different questions. The framework would have to allow for the computation of solutions to these questions through a process of finding solutions to a series of more basic "'bulding block" questions.

The second of these limitations is that the existence of doubly homomorphic encryption schemes was just recently demonstrated, and issues surrounding their implementation have not yet been fully explored. For this reason, it would be important to determine what common statistical questions require the existence of a doubly homomorphic encryption scheme as opposed to singly homomorphic encryption schemes like those assumed for the most part in the above section (Section 4). It would also be important to know how often these problems requiring the stronger property arise in research using telemetry. In other words it would be helpful to determine the impact of a double homomorphic property on the *expressibility* of a system like *HICCUPS*.

6.2 The Case against Strict CPU Metrics

In order to emphasize the fact that we believe that future systems designed with purposes similar to *HICUPPS* should not solely be evaluated with computational performance metrics, we note that in fact the overhead added by computing aggregates on encrypted data is minimal in comparison to other aspects such as communication costs. To do this, let us compare the following scenarios for computing aggregates on medical telemetry.

- **Current Practice:** With the current infrastructure, researchers and manufactures acquire statistical data by accessing patients' medical data directly. This scenario requires that *all* relevant data be gathered and aggregated under the management of one institution.

  In the current situation, the manufacturers query the patients frequently and collect all the patients data. On the other hand, the researchers must submit their requests to manufacturers or doctors to obtain the statistical data—potentially taking months to access aggregated data.

- **Distributed aggregation without privacy:** An improvement to the above system would result from the distribution of the workload among caregivers. A manufacturer or researcher submits a request for an aggregate to a designated aggregator. This aggregator broadcasts the manufacturer's request to all of the caregivers. Each caregiver computes the sub-aggregate of his patients' medical data and sends back one single value to the aggregator. Finally, the aggregator needs to combine these sub-aggregates into one aggregate value that will be returned to the manufacturer or researcher. The designated aggregator could be the manufacturer itself, one of the caregivers, or an external entity.

  An estimate of the time that is required to compute an aggregate under this scenario is made up of: the time needed to submit the request to the aggregator; the time needed to broadcast the request to the caregivers; the time caregivers need to compute the sub-aggregate value; the time needed to transmit and process the data from all the caregivers; and finally, the

time needed to compute and return the aggregate value to the manufacturer or researcher. If we assume that the caregivers work in parallel, then the time required to compute the sub-aggregates can be assumed to be the maximum time needed by any one caregiver.

- ***HICCUPS*:**

  In practical terms, the hypothetical performance of *HICCUPS* differs from the distributed aggregation scenario mentioned above in that: each sub-aggregate computed by the caregivers is encrypted using, first, the researcher's or manufacturer's public key and then using the aggregator's public key. Additionally, the aggregator would have to be chosen randomly for each request.

While we believe that speed of computation should not be a primary metric of quality, a practical system should not require excessive computation (e.g., weeks) to accomplish aggregation.

In order to estimate the computation overhead added by *HICCUPS*, we compare the time overhead of our protocol to the second scenario. There are essentially four pieces of overhead: the time that it takes to perform two encryptions of a single value; the time that is needed by the aggregator to decrypt the first layer of encryption; and the time that is needed to perform k operations on encrypted data, where N is the number of caregivers. By operations on encrypted data we mean, for example, additions on a group of elliptic curve points. Therefore, the performance of *HICCUPS* can be estimated by calculating the encryption overhead and adding it to the performance time of the distributed aggregation scenario above.

HICCUPS could be implemented using a variety of encryption schemes. For the purposes of this hypothetical evaluation, we will refer to only two of the more commonly used encryption schemes that have the homomorphic property. These schemes are RSA and ElGamal based on ECC.

We use the data provided by Gupta et al. [17] to estimate the time overhead of *HICCUPS* due to security in comparison to the second scenario. As Table 1 shows, a manufacturer or researcher would need to tolerate only a delay in order of 100 milliseconds in order to protect the security and privacy of patients data. Moreover, the addition operation is about 0.59 μseconds and 0.71 μseconds for ECC-160 and ECC-224 bits respectively [7].

7. RELATED WORK

Related work on secure aggregation includes research on information disclosure for queries on encrypted data, applications of homomorphic encryption to electronic voting, and advances in understanding the theoretical limits of homomorphic encryption.

7.1 Queries on Encrypted Data

Earlier in this paper we discussed that an alternative approach to computing aggregates on encrypted data could be to enable queries on encrypted data that return a list of records. This approach differs from the one suggested in this paper. For example Song et al. [30] and Shi et al. [29] have explored different mechanisms for allowing authorized users to share data and store private information on untrusted servers.

We would like to emphasize the difference between information disclosure and cryptographic aggregation. That is, in the scenario portrayed in this paper, medical telemetry should remain private across caregiver institutions, and therefore approaches like the ones cited above may not be appropriate for this application. However, if researchers determine that information disclosure is necessary we believe that the following aspects should be considered:

1. A solution to a query must be based on a precise definition of a de-identified row. In particular, it should be clear which fields, such as name and national identification numbers, should not be disclosed in any given row.
2. A solution to a query must protect the origin of any given row. That is, the probability of determining which caregiver provided any given row must be negligible.
3. A solution to a query should provide a proper model for performing information leakage analysis. It is conceivable, for example, that multiple crafted queries could be combined to increase the probability of determining the origin of a given row.

7.2 Other Homomorphic Encryption Applications

Homomorphic encryption has been highly studied since its introduction by Rivest et al. [26]. Their paper proposed the idea of being able to compute on encrypted data without the need to decrypt. The goal was to design cryptosystems in such a way that the encryption operation commuted with the operations on plaintexts. In other words, the desired property was to be able to construct an encryption scheme such that you would obtain the same result by *multiplying* two plaintexts and then encrypting the result, or, by first encrypting two plaintexts and then multiplying their corresponding ciphertexts.

Castelluccia et al. have shown that it is possible to compute aggregates such as averages, variances and standard deviations in a scenario similar to the one described, using only a single homomorphic operation [6]. However, due to the resource constraints of sensor networks, their work uses symmetric key cryptography which imposes a different set of requirements than those in this work. For instance, aggregation in sensor networks is hierarchical as opposed to the one-layer aggregator in *HICCUPS*.

In the case that giving access to *de-identified* records is necessary, there should be a precise definition of what this practice entails. It may not be sufficient to eliminate clearly identifiable fields in a record, such as name and national identification numbers. Techniques such as the ones used in mixnets could also be employed. Additionally, models of information leakage, such as the one given by Xia et al. in their recent proposal for a voting system [31], must be given. These models are necessary to provide a formal analysis of the privacy leaked by releasing de-identified records.

The research problem addressed by our *HICCUPS* design shares similar goals with electronic voting designs. Both systems aim to protect user (patient vs. voter) privacy while providing aggregated result of the private data (statistical function vs. count). However, the solutions for voting systems cannot be easily applied to the telemetry systems because of two major differences of the problems: (1) The

Table 1: Estimated overhead added by *HICCUPS* for performing a simple aggregation with 100 caregivers with 1000 records each. The table shows the overheads using four different primitives: securely equivalent RSA-1024 and ECC-160, as well as RSA-2048 and ECC-224.

Protocol	RSA-1024	ECC-160	RSA-2048	ECC-224
Time Overhead	901.309 ms	380.66 ms	5786.611 ms	527.431 ms

constraints of voting systems (voter privacy and result verifiability) are believed to be self-contradictory. This is not the case in telemetry systems. (2) While the voting system requires only one fixed query (counting), it is not feasible to require medical researchers to completely fix their queries.

Homomorphic encryption schemes have been successfully applied to voting schemes [25]. As was briefly mentioned, the problem of aggregating votes has some similarities to the problem of computing aggregates with medical data. However, the desired properties and the conditions pose different problems, as discussed in Section 4. For example, the problem of generating ballots, aggregating votes individually and allowing individual verifications may impose a non-negligible overhead for computing queries on a regular basis. Also, homomorphic encryption has been used in the implementation of universal re-encryption for mixnets [15].

There have been attempts to provide privacy-preserving systems for sharing medical data. In one of the most recent attempts, Au and Croll proposed a privacy-preserving centralized e-health system to provide access to health record data from medical databases distributed across various clinics and hospitals [1]. However, in the related field of voting, Jefferson et al. [20] exposed some of the privacy risks introduced by a centralized system, such as the voting system studied in their work [19]. Furthermore, Sahai suggested that the existence of an efficient and practical semantically secure public key encryption scheme that is also algebraically homomorphic, would enable minimally interactive distributed data-mining and secure computation [28].

7.3 Theoretical Advancements

Goldwasser and Micali introduced the term *semantic security* when they were defining the first probabilistic cryptosystem [13]. This notion of security was needed to formalize the fact that deterministic cryptosystems are not secure against chosen ciphertext attacks. That is, if a deterministic encryption scheme is used, then it is possible that an eavesdropper observing several messages may be able to detect ciphertexts coming from identical messages. This first probabilistic system, proposed by Goldwasser and Micali [14] was a homomorphic encryption system that, while impractical, served as the basis for many other homomorphic encryption systems. This implies that the highest security level cannot be reached by a deterministic homomorphic cryptosystem. Even further, Boneh and Lipton showed that any deterministic algebraically homomorphic cryptosystem can be broken in sub-exponential time [5].

Homomorphic encryption does not provide the *nonmalleability* security requirement. For a cryptosystem to be *nonmalleable* it is necessary that given a ciphertext $c = E(m)$, it should be hard for an adversary to create a ciphertext $c' = E(m')$ such that a relationship between m' and m can be established. It is clear that homomorphic encryption schemes do not satisfy this property since a relationship between m' and m would be given by the homomorphic property. For a formal discussion on this, refer to Dolev et al. [8, 9]. Bellare et al. [2] showed that if a cryptosystem does not provide the *nonmalleability* security requirement, then chosen plaintext indistinguishability IND-CPA is the strongest requirement that may be satisfied by it. In fact, there are homomorphic encryption schemes that satisfy IND-CPA, for example Elgamal [11] and Pallier [25] cryptosystems.

The highly desirable properties of homomorphic encryption made it an important topic of research. Various cryptosystems have been designed to exploit the homomorphic property for a single operation. The question of the existence of a fully homomorphic cryptosystem, i.e. one that commutes with both addition and multiplication efficiently, was an open problem until recently. Craig Gentry proved that it is possible to create a fully homomorphic encryption using lattices [12].

The work of Boneh et al. [4] presents a homomorphic encryption scheme that allows the evaluation of 2-DNF formulas on encrypted boolean variables. The defined encryption function supports addition and one multiplication. The same technique can be used in our paper to enhance the system capabilities for researchers and manufacturers.

The assumption of having honest majority of caregivers made in our paper can be relaxed by applying the techniques from the work of Ishai et al. [18]. Their work proposes several solutions to perform secure arithmetic computation with no honest majority.

8. CONCLUSION

Our research examines the use of homomorphic encryption to make aggregated information contained in medical data sets more available to clinical researchers and device manufacturers while preserving patient privacy. We questioned the need to release completely de-identified records to solve this problem as suggested by Gostin and Nass [24]. Our work focuses on a enabling an aggregation of the medical telemetry across different caregivers. The preliminary design of *HICCUPS* uses homomorphic encryption to allow researchers to obtain statistical information while preserving the data privacy of individuals. The paper gives an overview of the design of the system and several examples of common aggregation functions that can be computed using homomorphic properties. We believe that *expressibility* of aggregation functions will serve as a more important metric of system quality than absolute running time in the context of delay-tolerant processing of medical data. We hope that the discussion of *HICCUPS* will help us prepare for its implementation and measurement.

9. ACKNOWLEDGMENTS

We thank the anonymous reviewers and workshop participants for their feedback. This research was supported by NSF grants CNS-0831244 and CNS-0716386.

10. REFERENCES

[1] R. Au and P. Croll. Consumer-centric and privacy-preserving identity management for distributed e-health systems. In *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, pages 234–234, Jan. 2008.

[2] M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations among notions of security for public-key encryption scheme. *Lecture Notes in Computer Science*, 1462, 1998.

[3] L. Biel, O. Pettersson, L. Philipson, and P.Wide. ECG Analysis:A New Approach in Human Identification. In *IEEE Transaction on Instrumentation and Measurement*, pages 808–812, June 2001.

[4] D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-DNF formulas on ciphertexts. *Lecture Notes in Computer Science*, 3378:325–341, 2005.

[5] D. Boneh and R. Lipton. Algorithms for black box fields and their applications to cryptography. *Advances in Cryptology*, 1109 of Lecture Notes in Computer Science:223–238, 1996.

[6] C. Castelluccia, E. Mykletun, and G. Tsudik. Efficient aggregation of encrypted data in wireless sensor networks. *IEEE Mobiquitous*, 2005.

[7] H. Cohen, A. Miyaji, and T. Ono. Efficient elliptic curve exponentiation using mixed coordinates. In *ASIACRYPT '98: Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security*, pages 51–65, London, UK, 1998. Springer-Verlag.

[8] D.Dolev, C.Dwork, and M.Naor. Non-malleable cryptography. *Proceedings of the 23rd Annual ACM Sympusium on the Theory of Computing*, pages 542–552, 1991.

[9] D.Dolev, C.Dwork, and M.Naor. Non-malleable cryptography. *SIAM Journal of Computing*, 30(2):391–437, 2000.

[10] E. A. DeFranco, M. Lian, L. A. Muglia, and M. Schootman. Area-level poverty and preterm birth risk: A population-based multilevel analysis. *BioMed Central Public Health*, 8(316):doi:10.1186/1471–2458–8–316, 2008.

[11] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *Information Theory, IEEE Transactions on*, 31(4):469–472, Jul 1985.

[12] C. Gentry. Fully homomorphic encryption using ideal lattices. In *STOC '09: Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 169–178, New York, NY, USA, 2009. ACM.

[13] S. Goldwasser and S. Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. *Proceedings of the 14th Annual ACM Symposium on the Theory of Computing*, pages 365–377, 1982.

[14] S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[15] P. Golle, M. Jakobson, A. Juels, and P. Syverson. Universal re-encryption for mixnets. *Lecture Notes in Computer Science*, 2964:1988, 2004.

[16] L. O. Gostin and S. Nass. Reforming the HIPAA privacy rule: Safeguarding privacy and promoting research. *JAMA*, 13(301):1373–1375, 2009.

[17] V. Gupta, D. Stebila, and S. C. Shantz. Integrating elliptic curve cryptography into the web's security infrastructure. pages 402–403, 2004.

[18] Y. Ishai, M. Prabhakaran, and A. Sahai. Secure arithmetic computation with no honest majority. Cryptology ePrint Archive, Report 2008/465, 2008. http://eprint.iacr.org.

[19] D. Jefferson, A. Rubin, B. Simons, and D. A. Wagner. Security analysis of the secure electronic registration and voting experiment (SERVE). Technical report, 2004. www.servesecurityreport.org.

[20] D. Jefferson, A. D. Rubin, B. Simons, and D. Wagner. Analyzing internet voting security. *Commun. ACM*, 47(10):59–64, 2004.

[21] B. Kapron, D. Kempe, V. King, J. Saia, and V. Sanwalani. Fast asynchronous byzantine agreement and leader election with full information. In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1038–1047, Philadelphia, PA, USA, 2008. Society for Industrial and Applied Mathematics.

[22] W. H. Maisel. Pacemaker and ICD Generator Reliability: Meta-analysis of Device Registries. *JAMA*, 295(16):1929–1934, 2006.

[23] W. H. Maisel, M. Moynahan, B. D. Zuckerman, T. P. Gross, O. H. Tovar, D.-B. Tillman, and D. B. Schultz. Pacemaker and ICD Generator Malfunctions: Analysis of Food and Drug Administration Annual Reports. *JAMA*, 295(16):1901–1906, 2006.

[24] R. B. Ness. Influence of the HIPAA privacy rule on health research. *JAMA*, 18(298):2164–2170, 2007.

[25] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. *Lecture Notes in Computer Science, EUROCRYPT'99: Proceedings of Advances in Cryptology*, 1592:223–238, 1999.

[26] M. D. R. Rivest, L. Adleman. On data banks and privacy homomorphisms. *Foundations of Secure Computation*, 1978.

[27] L. Ricciardi and A. Rubel. Encouraging the use of, and rethinking protections for de-identified (and "anonymized") health data.

[28] A. Sahai. Computing on encrypted data. *ICISS*, (5352):148–153, 2008.

[29] E. Shi, J. Bethencourt, T.-H. H. Chan, D. Song, and A. Perrig. Multi-dimensional range query over encrypted data. *IEEE Symposium on Security and Privacy*, pages 350–364, 2007.

[30] D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. *IEEE Symposium on Security and Privacy*, pages 44 – 55, 2000.

[31] Z. Xia, S. A. Schneider, J. Heather, and J. Traoré. Analysis, improvement and simplification of prêt à voter with paillier encryption. In *EVT'08: Proceedings of the conference on Electronic voting technology*, pages 1–15, Berkeley, CA, USA, 2008. USENIX Association.