Report on DIMACS* Working Group on Challenges for Cryptographers in Health Data Privacy

Date of working group meeting: June 30, 2004

Working Group Organizers:

Benny Pinkas, HP Labs
Kobbi Nissim, Microsoft Research

Report Author:
Krishnaram Kenthapadi
Department of Computer Science
Stanford University

Date of report: July 14, 2004

*DIMACS was founded as a National Science Foundation Science and Technology Center. It is a joint project of Rutgers University, Princeton University, AT&T Labs-Research, Bell Labs, NEC Laboratories America, and Telcordia Technologies, with affiliated partners Avaya Labs, IBM Research, Microsoft Research, and HP Labs.

1 Working Group Focus

The purpose of this meeting was to bring together cryptographers working on information privacy and security and statisticians working on data privacy, especially in the health area. The meeting was the outgrowth of the discussions at the DIMACS Working Group on Privacy/Confidentiality of Health Data. The meeting focused on the problem of computing with sensitive data while hiding sensitive information embedded in it, specifically in the context of healthcare. One possibly relevant technique is secure computation, where different parties compute some function of their private inputs while hiding any additional information about their inputs. Another technique is publishing "sanitized" versions of sensitive data, in which some elements are perturbed or suppressed in order to hide sensitive properties. The security of transferring data between locations/parties was not discussed, since this problem has rather straightforward solutions.

2 Summary of Presentations

2.1 Topics Related to Secure Computation

Speaker: Benny Pinkas, HP Labs.

Dr. Pinkas provided a short introduction to secure multiparty computation and led a discussion on identifying functions of interest for healthcare applications and modeling the adversary.

In the secure function evaluation (SFE) problem, a set of two (or more) parties with private inputs wish to compute some joint function of their inputs, while preserving some security properties such as *privacy* and *correctness*. Security must be preserved in the face of adversarial behavior by some of the participants. For example, in what is known as Yao's millionaires problem, two parties want to find out whose input is the larger of the two without revealing to the other the value of their input. A secure protocol must reveal no more information than the output of the function itself, i.e., the information revealed should be the same as that in the ideal scenario where the parties give their inputs to a trusted party who computes and outputs the function value. There are cryptographic protocols that implement the ideal scenario without actually using a trusted party. The use of these protocols makes sense if the parties are motivated to *submit their true inputs* and can *tolerate the disclosure of the function value*.

For the case where one of the parties is an adversary, the adversary is

modeled either as (1) *semi-honest*: follows the protocol but tries to learn more or as (2) *malicious*: can do anything, such as violating the protocol (for example, the malicious party might provide a biased bit, when the protocol requires flipping a random coin). Clearly it is easier to provide security against semi-honest adversaries. The semi-honest model is appropriate when the parties are semi-trusted (such as cooperating government agencies) or when we use secure hardware/software, where it is easier for the adversary to eavesdrop than to change the program. Note that even in the malicious adversary model, we can make sure that the input is not changed during the course of the protocol, using *input commitment*.

An interesting question is whether there are more appropriate models in between the semi-honest and the malicious. One possibility is the scenario where the parties may behave maliciously but do not want to be detected. Another issue is collusion between the participating parties. The current notion of collusion is very strong – a set of t colluding parties is equivalent to a single party controlling all the t parties. There could be weaker notions, for example, the colluding parties may share the signatures but not their private keys. In general, it would be interesting to obtain trade-offs between efficiency and security guarantees.

Dr. Pinkas described a new setting for multiparty protocols, in which the clients provide inputs and the computation is performed by a different set of computation servers, and security is guaranteed as long as there are no large collusions of computation servers. We achieve separation between input providers and computation.

He described Yao's construction for secure two-party computation of general functions and an implementation of the same, *FairPlay*. In FairPlay, the programs written in a high-level language are first compiled into a low-level language describing Boolean circuits (Secure Hardware Definition Language, SHDL) and then transformed into programs implementing Yao's protocol. This system would investigate if two-party SFE is practical and provide actual measurements of overall computation. Efficient protocols have already been obtained for computing mean, max/min, set intersection, median and quintiles and constructing decision trees. It would be interesting to find candidate practical applications where SFE will be useful. For example, we may want to do regression analysis across horizontally or vertically partitioned datasets securely. We may also want to adaptively change the pre-agreement about the function to be calculated – it is not always possible to decide beforehand if the disclosure of the function value can be tolerated.

For census data or healthcare data, there are two broad approaches that are currently followed:

- Publish sanitized data: Data is perturbed with random noise, in such a way that the macrolevel properties are preserved. SFE techniques are not useful for this approach. Another possibility is to provide an "encrypted" version of the database so that some specific set of functions (statistics) can be computed, but no other information is revealed. But we may not know all the relevant functions a priori; moreover it is impossible to compute every possible statistic.

- Allow certain "secure" queries on the database, without releasing the database: Cumulative disclosure across queries is a major problem in this approach.

Another scenario is when a function needs to be evaluated on data present across multiple agencies. For example, (1) the Federal Aviation Administration might want to detect unusual behavior by integrating data from different airlines. (2) Pharmaceutical companies might want to perform statistical analysis across chemical databases. (3) States might want to perform some function on the student databases they have. An interesting question is to find out the relevant functions that need to be efficiently evaluated in this scenario.

2.2 Online Query Auditing

Speaker: Kobbi Nissim, Microsoft Research.

Dr. Nissim described the problem of online query auditing. Consider a setting where statistical or aggregate queries are posed against a database containing sensitive information about individuals. We would like to ensure that answering such queries does not leak information about individuals. The above problem, known as the statistical database privacy problem, can be handled using either perturbation or query restriction methods. In the former, either noise is added to the input data or noise is added to the output query responses. In the query restriction family of methods, the trail of queries is monitored to ensure that it is not possible to combine answers to queries so as to deduce information about any individual. We consider a special subclass of the latter, known as the *query auditing problem*: Given a sequence of t queries and the corresponding answers and given a new query, provide an answer to the query if and only if revealing the answer would not cause any "privacy loss". Privacy loss is defined to occur only when a database entry may be uniquely deduced. Moreover we assume that, whenever an answer is provided to a query, the answer is exact. [An

interesting direction is to consider the problem of query auditing where approximate answers are provided (as in perturbation methods).] Examples of queries include min, max, median, sum, average and count. This problem has been considered from the mid-70's and has evoked recent interest.

Dr. Nissim emphasized the fact that *denials themselves could leak information*. In the previous work, only the answers to the queries so far were taken into account and denials were ignored. For example, consider a scenario where the (single-attribute) database has real entries, d_i and sum/max queries are considered. Suppose that the first query is $\text{sum}(d_1, d_2, d_3)$ and is answered as 15. Let $\max(d_1, d_2, d_3)$ be the second query. Clearly this query is denied if and only if $d_1 = d_2 = d_3 = 5$. Thus denial reveals the database entries in this example. He gave further examples involving interval-based / max / Boolean auditing where denials could leak information. When it was pointed out that, in the real world, the queries such as "sum of first three records" are never asked but instead SQL[1]-type queries are invoked, Dr. Nissim observed that information can still be extracted from denials (using queries such as "sum of largest 1% of the entries").

The problem is that it is not clear how to incorporate denials in the auditing decision. In the current definitions, the auditor uses information which is not available to the user. Dr. Nissim proposed the idea of *simulatable auditing*, in which the users can decide the denials by themselves. In order for the auditing to be simulatable by the user, the decision should be made based only on the t queries/answers so far and the current query (no other information such as the database contents should be used). Simulatable auditors provably do not leak any information in deciding whether a query can be answered. This approach is suggested as a starting point for further research.

2.3 Offline Query Auditing for Privacy

Speaker: Tomas Sander, HP Labs.

Dr. Sander discussed the issues in offline query auditing, in which a trail of what happens to privacy sensitive data is maintained. Apart from the main purpose of detection of privacy violations, this is also useful for documentation, forensics and demonstrating compliance with privacy policy. The main challenges are:

[1]SQL (Structured Query Language) is a standard computer language for accessing and manipulating database systems.

- How can we audit for the sake of privacy?
- How can auditing itself be performed in a privacy-friendly and secure way?

A variety of data is collected and later analyzed to check if it conforms to simple privacy policy rules and to collect statistics about suspicious behaviors. It might seem somewhat paradoxical that a lot of private information has to be collected in order to protect privacy! In fact, audit control mechanisms are required as part of the HIPAA regulations. We expect that offline auditing can be more sophisticated due to the lack of real-time requirements.

Suspicious behaviors include access to PHI (Protected Health Information) of VIPs, employees and minors, access by anyone not directly related to the patients treatment or payment of healthcare operation, access of records that have not been accessed in a long time, access to sensitive records such as psychiatric records, etc. Pseudonymization and anonymization of audit file data and encrypted storage are some of the techniques used in offline auditing. In the pseudonymization scheme, (predefined) identifying features are substituted by shares, generated via Shamir's secret sharing scheme. When the log data is encrypted and stored, we would like to search the encrypted data (for example, using Identity Based Encryption techniques).

2.4 Topics Related to Data Sanitization

Speaker: Lawrence H. Cox, Office of Research and Methodology, National Center for Health Statistics, Centers for Disease Control and Prevention.

Dr. Cox described the techniques used to achieve statistical disclosure limitation and then led a discussion on the right definitions of privacy breach and modeling the adversary.

Confidential information from individual subjects is collected by government statistical agencies (such as the census bureau) and similar organizations for various purposes. Quite often, the utility lies in dissemination of some version of the collected data. The data collector/disseminator must identify how vulnerable the confidential data is to disclosure to unauthorized third parties, such as neighbors or business competitors, and limit the risk of disclosure of information about individual subjects to an acceptable level. This is done because it is required by law, it is part of ethical statistical practice and there is the practical necessity of maintaining public trust to ensure that the subjects continue to participate and provide accurate information.

Statistical disclosure is said to occur when the release of a statistic enables an unauthorized third party (intruder) to learn more than was possible prior to the release. The aim is to *limit* the disclosure to an acceptable level. The released data could be either in the form of *tabular data* or *microdata*. Each dimension of tabular data corresponds to a domain variable and the cells contain aggregate information. On the other hand, microdata contains unit record data for each subject and is released after sanitization for public use. Another approach is not to release the microdata, but allow aggregate queries on the data. Here we face the issues discussed in Subsection 2.2.

Quantitatively, (outsider) statistical disclosure is said to occur:

- for count data, when some cell exhibits a small count such as $n = 2$ or $n = 3$ (n-threshold rule; value of n is application dependent).
- for magnitude data, when a subject's contribution can be estimated to within p percent of its value (p-percent rule).

On the other hand, insider disclosure occurs when the intruder is another subject in the cell.

The above rules can be considered as special cases of *linear sensitivity measures*. It was observed that even though the original definition of statistical disclosure resembles standard cryptography definitions, the above notions are attack-specific (for example, in the insider model, the attacker is the second largest competitor trying to learn the largest value) unlike those in cryptography. Moreover, auxiliary information could cause disclosure (for example, companies release stock holder information, but do not take this into account for other disclosure limitations). For tabular data, disclosure limitation techniques include complementary cell suppression, input perturbation or data swapping and controlled tabular adjustment. For microdata, disclosure occurs when the intruder can associate a microrecord with a subject. The disclosure limitation techniques include access control, sampling and record deletion, item deletion, recoding, (input) perturbation, data swapping, microaggregation and synthetic microdata. Some of the issues involved are given below.

- Sampling is ineffective if the intruder knows the subject in the sample.
- Deletion and recoding are data specific, affect analysis and tend to focus on salient subjects.
- Perturbation methods provide weak protection.
- Swapping methods affect statistical properties such as correlations.

- In synthetic methods, we obtain a good statistical model and generate (and release) according to this model. However these methods do not capture all relationships of interest, particularly for subdomains, and require statistical and domain expertise and care.

Similarly in the online statistical database query system model, there are other issues related to disclosure limitation techniques. Can we let a set of safe basis queries be answered, from which answers to more interesting queries can be deduced?

The important open problem is: what is the right method of releasing microdata? The techniques such as input perturbation, data swapping and synthetic data all have various problems as outlined above. The statistical agencies handle huge volumes of data (for instance, census table is very large and very sparse) and aim to release the highest quality information to the public while protecting privacy to the extent possible. How can we explicitly quantify both disclosure risk and utility? For example, with risk $<$ α, what is the maximum utility achievable? A challenge for cryptographers is to develop cryptography-type definitions for vulnerabilities and intruder behavior. Any scheme for disclosure limitation should be highly *scalable*.

Another interesting problem is to determine the right notion of privacy for new forms of multimedia data (eg: high resolution satellite imagery, photos of cars taken). What is the level of privacy we can expect when photos/videos are collected without permission and published on the web?

2.5 Discussion: Should We Explore the Relation of Technologies such as Digital Rights Management, Association Rule Hiding and Private Inference Control to Healthcare Data Privacy?

Digital Rights Management (DRM) technology enforces rules that control the usage of content. The enforcement of the rules can be based on the use of secure software or secure hardware such as the hardware designed by the Trusted Computing Group (TCG). In this model, hardware support is used to enforce policies so that only approved programs can be run. One possible approach is to provide interested parties with census data in a laptop containing trusted computing hardware and DRM software, so that only safe queries can be run through a certified software interface. Auditing tools can be used to examine the operations that were applied to the data. (There are many remaining issues even given a successful tool of this type. For example, two users can collude and learn more than is allowed.) Note that

the adversary model for healthcare privacy is likely to be different than the one which is usually considered for DRM / Trusted Systems. In healthcare privacy, we are concerned about a powerful company/organization which trying to learn more information but might be legally accounted for its actions. This is different than the typical scenario DRM of media, where the adversary is a large number of users who are trying to tamper with the hardware to run uncertified software or obtain unauthorized access to the protected content.

Association rule hiding tries to publish (modified) data while hiding sensitive association rules that exist in the original data.

In private inference control, we want to achieve inference control while preserving user privacy. This does not seem very relevant to the current set of problems and state of the art. However at the intermediate aggregation level, this approach is related to the query restriction model.

3 Conclusions and Future Directions of Research / Collaboration

The meeting identified a number of directions for future research / collaboration and led to some general conclusions:

1. The basic question of understanding privacy still is the main research problem this group should address, in particular, which functions may be computed over sensitive data without breaching privacy. Both Benny Pinkas' and Larry Cox's talks touched heavily on this point, noting that:

   - although cryptography gives one the perfect machinery for securely computing a function once we choose the function, it offers absolutely no tools for deciding which functions are safe and which are not (in terms of what they reveal on the underlying data).
   - most (if not all) known sanitization methods are heuristics, with very little underlying PRIVACY theory.

   Kobbi Nissim's talk also touched upon the theme that, sometimes the computation of "harmless looking" predicates of the data (even if privacy aware) may be devastating to privacy.

2. Regarding the DRM / Trusted Computing discussion: One has to take into account that even idealized DRM / Trusted Computing would

only provide a machinery for enforcing the privacy policies for sensitive data. We noted that, although plausible, this itself is a highly non trivial technological challenge. However, even the idealized DRM / Trusted Computing would not solve our main problem: We would need still to decide which functions of the sensitive data should be computed, and further, make sure that we can software-encode this information. In other words, this direction does not free us from tackling the big problem of understanding privacy, including (but not limited to):

- formally defining what data privacy means.
- formally defining our attack modes.
- checking the limits of confidentiality control methods in the literature.
- checking the interaction between notions of privacy.

3. One other conclusion is the importance of communication between the different communities that deal with data privacy: statisticians, cryptographers, data-base and data mining researchers (eventually, we should also consider relevant research in learning and algorithms). This time we had mainly cryptographers and statisticians, and we should pay attention to include researchers from the other communities in following meetings.

4. Cryptographers left the meeting with an interest in looking more closely at functions that are of interest to statisticians in the sense that they would like to have a way of computing them while preserving privacy. Two cryptographers in the group have started to look at the linear regression function and will communicate with statisticians in the group to find out exactly what they need.

5. More generally, there is need for development of encrypted microdata files accompanied by software implementing standard statistical functions in a secure function evaluation mode. Confidential microdata X, encrypted using E, would lead to a release file $X' = E(X)$ together with software for computing statistical functions f so that $f(X)$ can be computed as or from $f(X')$.

6. Currently, the CDC (Centers for Disease Control and Prevention) lets researches come to their offices for a limited time and conduct research with CDC raw data. This is costly for both the researchers and CDC.

An alternative approach is to encode the raw data on a protected laptop and provide it to the researchers. The goal is that the laptop will enforce rules that ensure the privacy of the data and prevent any illegitimate use of the data. This is an instance of "secure computing" initiatives that are pushed by major computer vendors such as Microsoft and HP. This relatively simple problem (compared to the grand goals of secure computing) seems appropriate for a trial/prototype, and members of the group are planning to investigate whether the technology will be feasible in the near future and whether DIMACS' partner companies will help to develop it as a prototype and community service project.

7. Relative to items 5 and 6: The secure PC in item 6 would contain confidential microdata X and statistical software to implement statistical functions f so that the user cannot crack the box or access memory to obtain X directly but can only compute a set of permitted functions $f(X)$. The user would receive the PC under a license promising not to attempt disclosure and the PC would have to be inspected/audited for intrusion and re-authorized electronically periodically.

4 Acknowledgements

The author and the DIMACS Center acknowledge the support of the National Science Foundation under grants number CCR 03-14161 and EIA 02-05116 to Rutgers University.