Preserving confidentiality when sharing medical database the Cellsecu system[*]

Yu-Cheng Chiang[†]and Tsan-sheng Hsu[‡]and Sun Kuo[§]and Da-Wei Wang[‡]

October 16, 2001

Abstract

We propose a computer system named Cellsecu that maintains not only the anonymity but also the confidentiality of each cell that contains sensitive information in medical database by automatically removing, generalizing, and expanding information. The system is designed to enhance the data privacy protection for the data warehouse to automatically handle queries. In most of the cases health organizations collect medical data with all explicit identifiers, such as name, address, and phone numbers. Simply removing all the explicit identifiers priori to the release of the data is not enough to preserve the data confidentiality, for the remaining data can be used to re-identify individuals by linking or matching the data to other database or by looking at unique characteristics found in the database.

A formal model based on Modal logic is the theoretical foundation of Cellsecu, a new confidentiality criteria called "non uniqueness" is defined and implemented. We believe modeling this problem formally can clarify the issue as well as clearly identify the boundary of current technology. Base on our preliminary performance evaluation, the confidentiality check module and the confidentiality enhancing module only slightly degrade the system performance.

[*] An abstract of this paper appears in [4].

[†] Department of Information Management, National Taiwan University

[‡] Institute of Information Science, Academia Sinica Taiwan

[§] Department of Medical Informatics, National Yang Ming University, Taiwan

1 Introduction

In Taiwan almost all the citizens are covered by the national health insurance plan. The National Health Insurance Bureau (NHIB) therefore collects and maintains a huge database containing high quality health related data. It is a gold mine for many researchers working on the health care related areas as sharing the data has the potential to benefit the public significantly. For example, the existence of such database would enable researchers to track certain diseases as well as to patients' responses to certain drugs and allow for better organization and more legibility of medical files. However, with the rapid advances in the computerization of medical data, the question of protecting medical records privacy has begun to arise. Storing a large amount of sensitive information in a central location (databases) could open the door to "invasion of privacy". In order to better utilize these valuable information, NHIB has authorized the National Health Research Institute (NHRI) to handle the releasing of the database. Currently, NHRI accepts applications from researchers for requesting data contained in the databases. A review committee grants requests based on the purpose and the relevance of the research to the requested data. The process takes some time and may fail to distinguish requests for highly sensitive data from requests only for general statistical data.

How to publish a database while preserving confidentiality is an old problem, the Social Security Administration (SSA) in USA employs the "bin size" as the measurement of the "anonymity" [2]. Two recently systems Datafly [17] and μ-argus [14] also use bin size as the anonymity measurement. By anonymity we mean that no one can identify certain record belongs to any specific individual, and data confidentiality refers to that a person cannot identify the value of certain field belongs to any specific individual. If the meaning of "identify" is the same for anonymity and confidentiality, then it can be easily deduced that data confidentiality implies anonymity and we believe in some case we do want finer grain privacy protection - the data confidentiality. The data confidentiality corresponds naturally to the Modal logic [5, 7], a mathematical framework to reasoning about the meaning of "knowing". We develop a formal framework for the data confidentiality in [13] and the Cellsecu system is developed based on that framework. Once the confidentiality criteria are defined, the next question is how to enhance the confidentiality if releasing certain database violates the confidentiality crite-

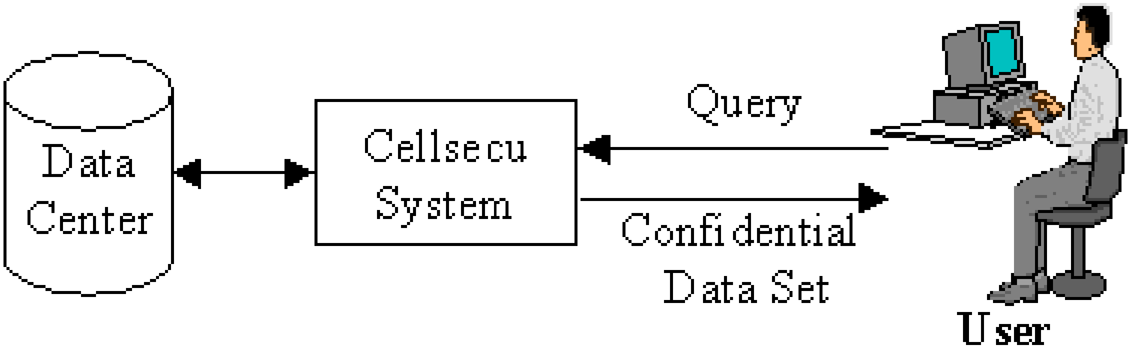

Figure 1: Cellsecu system as a gate-keeper.

ria. We follow the idea of using generalization to enhance the confidentiality proposed in Datafly system. A lattice framework is developed to facilitate searching for the least generalized yet confidential data set. Cellsecu is a web based prototype system developed based on the above-mentioned formalism. We envision as depicted in Figure 1 that Cellsecu can serve as a gate-keeper such that users can freely query the data center and all the answers approved by Cellsecu preserve the data confidentiality where the idea data confidentiality is clearly stated and understood.

The rest of the paper is organized as follows. In Section 2 we give a brief review of previous works on database confidentiality. In Section 3 we present the system architecture of Cellsecu. Section 4 contains the performance evaluation and we conclude with some future research directions in Section 5.

2 Related work

Statistical database inference has been a subject for intensive research for three decades starting with a study by Hoffman and Miller [12]. A statistical database is a database system that enables its users to retrieve only aggregate statistics (e.g., sample mean and count) for a subset of the entities represented in the database. Many data collecting agent facing the dilemma that on one hand, such database systems are expected to satisfy user requests of aggregate statistics related to non-confidential and confidential attributes. On the other hand, the system should be secure enough to guard against user's ability to infer any confidential information related to a specific individual represented in the database. Readers are referred to [1] for a survey on the statistical database system security problem before 1989.

The *inference* problem, the problem that user can deduce classified information from unclassified information, is defined in [18]. Garvey defined three inference channels, deductive inference channel, adductive inference channel and probabilistic inference channel [8, 9]. The *aggregation* problem, the problem that users can aggregate lower level security information to form data having higher security level than any of the forming elements. Denning identified two kinds of aggregation, *attribute association* and *size-based record associations* [6, 19]. Attribute association is also called "data association" and size-based association is called "cardinal association" [11, 15]. We are facing the same problems in spirit, but different in the sense that our goal is to publish the data set itself, instead of the statistical data, e.g., sum or average of certain fields. However, techniques such as query restriction approaches, data perturbation methods, and output-perturbation methods can be helpful in our study.

Datafly [17] and μ-argus [14] are the two systems tackle exactly the same problem as we are. In 1996, the European Union funded an effort to develop specialized software for disclosing data such that the identity of any individual contained in the released data cannot be recognized. Statistics Netherlands has produced a program named μ-argus. The Datafly system developed in MIT by Dr. Sweeney in 1997. Datafly was written in Symantec C version 7.1 and Oracle's pro *C precompiler version 1.4. Both systems make decisions based on the bin sizes, generalize values within fields as needed, and remove extreme out-liner information from the released data. The μ-argus system blanks out the out-liner values at the cell-level with the cell-suppression process. The Datafly system uses generalization as the primary mechanism to enhance anonymity.

3 System architecture and methods

We focus on relational databases. For each data table, the fields are partitioned into the following three sets — Identifying (ID) Fields, Easily-Known (EK) fields and Unknown (U) fields. ID fields, e.g., social security number, are those that can be used to uniquely identify an individual, which cannot be released for any queries. EK fields, e.g., the heigh and eye color of an individual, are those that can be easily found by observing or other sources. Using a combination of several EK fields, it may be plausible to uniquely identify an individual. U fields, e.g., the test result of certain disease, are those we want to protect.

The link mode and the query mode are two ways for users to query the database. In *link* mode, the user already has some data from other sources and wants to link with the data center to get more fields. The *query* mode allows the user to query the database with SQL queries. The system flowchart is shown in Figure 2. When a query is submitted to the data center, the data center first produces the original query results by issuing queries to the corresponding databases. The filter process then removes all the ID fields to form the filtered query-set. The confidentiality test module then tests the confidential condition. If the filtered query-set dose not meet the confidentiality requirements, then it will be processed by the "generalize" module to reduce the specificity of data in the EK fields to produce a confidential query-set. The audit center records the user identity as well as the result of the confidentiality test. The admin configuration allows the data center privacy officer to set the sensitivity of each fields, to three-way partition the fields, to decide the confidentiality conditions, and to set the generalize parameters.

3.1 Confidentiality test

The idea of using bin size as the measurement of anonymity has been incorporated into several previous developed systems, e.g., Datafly and u-argus. Although the bin size more or less reflects the intuition of being anonymous, i.e., the members in the same bin are indistinguishable therefore anonymous, it is not enough to capture the data confidentiality. By data confidentiality, we mean that no one can infer the value of any sensitive field of any individual. Suppose Bob is very interested in the personal data of Alice. After querying the database, Alice's record is contained in a large bin. Bob cannot tell which record belongs to Alice. However Bob notices that the values of for example the values of a certain U field are all the same for all the records in that bin. Then Bob can safely deduce something about Alice, i.e., he knows the value of this field in Alice's record.

The bin size makes if less likely for Bob to know which record belongs to Alice, since the larger the bin the less likely the values of a field in the

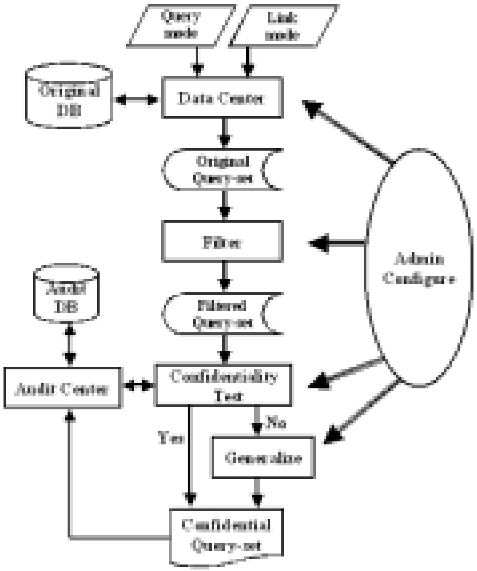

Figure 2: System architecture.

bin are the same. However, it does not prevent Bob to know something about Alice. We therefore propose a new condition for data confidentiality called ”non uniqueness”, which means that the value of any fields in U fields cannot be all the same in a bin. This condition is naturally derived from the formal model we constructed, here we omit all the mathematical details for the formal model. Interested readers are referred to [13] for details.

3.2 Generalization

The generalization approach to handle data confidentiality problem is proposed in Datafly. When a field is generalized, the values in that field are made to be more vague. For example, the birthday field can contain the specific month/day/year information. By generalizing this field to the format month/year, we drop the day information for every field. For numerical data such as height or weight we can group several consecutive numerical values to form intervals. For example, we can group people by heights within 5cm. The effect of generalizing some EK field equals to merging bins. For example, if we generalize the birthday field from month/day/year to month/year, then all the bins having the same birthday year and month are merged together to form a larger bin.

For most of the fields, the generalization relation can be readily modeled as a partial order. A partial order is a structure $(S, >)$, where S is a set of elements and $>$ is a transitive relation for S. Here S denotes the set of all the possible ways to generalize a field and $>$ relation denotes the vagueness of the data after such generalization. For example we can use $(month/year) > (month/day/year)$ to denote the fact that month/year gives vaguer values. The number of ways that a field can be generalized is called its number of *generalization levels.*

As a matter of fact, most of the field can be describe by a lattice where a lattice is a partial order with a top element and a bottom element. An element t is the top element in a partial order $(S, >)$ if and only if for every element e in S, $t > e$ or $t = e$, an element b is the bottom element if and only if for every element e in S, $e > b$ or $e = b$. Given two lattices $L_1(S_1, >_1)$ and $L_2(S_2, >_2)$, we can define the Cartesian product $L_1 * L_2 = L(S, >)$ where $S = S_1 * S_2$, and $(a_1, b_1) > (a_2, b_2)$ if and only if $(a_1 >_1 a_2)$ and $(b_1 >_2 b_2)$. The lattice formed by the Cartesian product of the lattices of fields in the EK fields provides a natural road map for the system to search for an adequate appropriate generalization.

Our current system produces all the minimal generalizations and we expect to develop a user-guided generalization in the future. We give an example below, consider Table 1, bins 2, 3, and 4 contain single record and bin 5 although contain more than one record, but the test results are the same in all of the records. Therefore, we have to generalize some of the EK fields before the table can be released. Table 2 is the result after changing the height and weights to intervals and generalizing the blood type to ABO, where ABO denotes that A, B and O are the possible blood types in that entry.

3.3 Lattice and generalized results

Assume that there are three EK fields in the data table, and for each field, there are 2, 3, and 2 generalization levels, respectively. For example, the first field can be height and it can be either in an interval of length 1cm or 5cm. The second field can be birthday, it can be in the form of month/day/year, month/year or year. The third field is the weight and it can be in an interval

	Easily-Known fields			Unknown
Bin	Height	Weight	Blood type	Test result
1	160	50	A	1
	160	50	A	1
	160	50	A	0
2	165	55	B	1
3	170	60	B	0
4	170	60	O	0
5	170	65	O	1
	170	65	O	1

Table 1: The original table

	Easily-Known fields			Unknown
Bin	Height	Weight	Blood type	Test result
1	160 165	50 55	ABO	1
	160 165	50 55	ABO	1
	160 165	50 55	ABO	0
	160 165	50 55	ABO	1
2	170 175	60 65	ABO	0
	170 175	60 65	ABO	0
	170 175	60 65	ABO	1
	170 175	60 65	ABO	1

Table 2: The generalized table

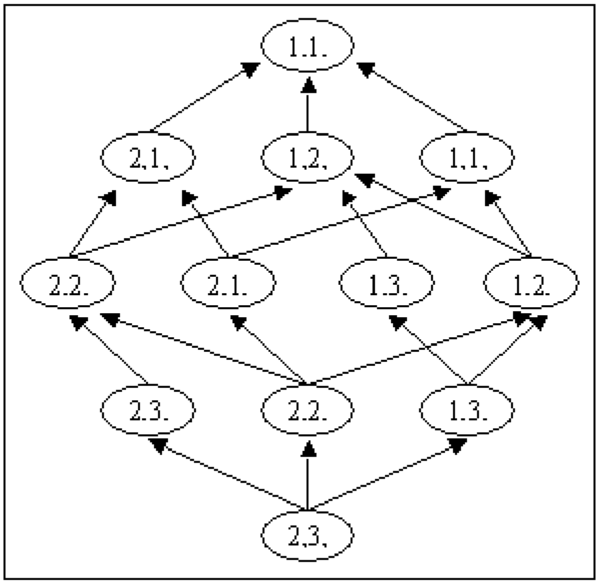

Figure 3: Lattice.

of length 1 kg or 5 kg. Each field can be represented by a lattice. Here the lattice structure is rather simple. It contains either two elements or three elements. We choose to use 1 to represent a more general, i.e., vaguer, value. Figure 3 is the Cartesian product of the three lattices for the three fields. Node (2,3,2) represents the original data set that contains the most specific information, and node (1,1,1) represents the most generalized data set that contains the least specific information.

If, for example, the data set represented by (2,3,2) violates the given confidentiality criteria, Cellsecu searches through the lattice to find a node satisfying the confidentiality criteria. In general, there might be several nodes satisfying the confidentiality criteria. If a node is least specific, alone the arrow in the picture, than another node, then this node is consider a less favored output. Therefore, we can use the *most specific confidentiality preserving anti-chain* (MSCPA) as the set of candidate outputs, where a set of non-comparable nodes in a lattice is called an anti-chain. Cellsecu finds every element in the MSCPA. In real application, we expect to develop an user interface so that the user can provide guidelines regarding the search direction in the lattice.

4 Preliminary system performance evaluation

Cellsecu runs on Celeron 400 with 256 MB SDRAM and Windows NT server 4.0. We use Apache 1.3.9(Win32) and Apache Jserv 1.1 as the web server system. Ten rounds of tests are conducted with various file and database sizes. For each round of test, we measure the following performance indicators:

- Query time: the execution time for querying the databases.
- Test time: the execution time for checking the confidentiality criteria.
- Generalize time: the execution time for the generalize module.
- Write time: the output time to write to a file.
- Upload time: the time needed to upload the user query to the system.

The tests are performed under the link mode:

- Dbsize: represents the size of the database queried, it varies from 20,000, 200,000 to 2,000,000 records.
- Upload-size: represent the size of the uploaded file, it can be either 10 or 100 records.

Based on Table 3, we observe the followings.

1. The size of the database has little impact on the overall execution time.
2. The number of records in the upload file has significant impart on the overall execution time.
3. The time for the confidentiality test and the time to execute the generalize module totally takes less than 4% of the total execution time.
4. Database access , upload and query time are the most time consuming operations.

Dbsize/upload-size		Total	Query	Test(*100)	Generalize	Write	Upload
20,000/10	Average time	0.178	0.036	0.002	0.002	0.009	0.122
	time/total		20.2%	0.9%	0.8%	5.3%	68.4%
20,000/100	Average time	1.539	0.195	0.013	0.008	0.525	0.789
	time/total		12.7%	0.8%	0.5%	34.1%	51.3%
200,000/10	Average time	0.178	0.039	0.003	0.002	0.003	0.113
	time/total		21.9%	1.7%	0.9%	1.7%	63.3%
200,000/100	Average time	1.559	0.272	0.028	0.033	0.525	0.683
	time/total		17.4%	1.8%	2.1%	33.7%	43.8%
2,000,000/10	Average time	0.169	0.039	0.002	0.002	0.005	0.117
	time/total		23.0%	0.9%	0.9%	2.8%	69.5%
2,000,000/100	Average time	3.391	1.375	0.022	0.019	1.170	0.774
	time/total		40.6%	0.7%	0.5%	34.5%	22.8%

Table 3: System performance measurements in seconds.

5 Conclusion

We have developed a prototype system called Cellsecu to protect data confidentiality while sharing health database. Our system is based on a formal model, therefore we can mathematically define the meaning of ”data confidentiality”. Whether our formal definition captures the intuitive idea of ”data confidentiality” is an obvious question. Roughly, the formal notion of ”data confidentiality” demands that the user cannot be sure that the value of any sensitive field of an individual. This deterministic viewpoint seems not enough for our intuition about data confidentiality. For example, if after acquiring the database the user can raise his confidence from 0.5 to 0.999 that the value of someone’s HIV test is positive, then most people would conclude that the releasing of the database ”revealed” some private information. We are working on clarifying this issue by applying the probabilistic knowledge model [10, 16]. Some preliminary result is in [3].

The quality of the generalize data set is also a very important and interesting question. We that believe both theoretical study and experimental works are needed to evaluate the impact of Cellsecu on the quality of research outcomes based on generalized data sets. Last we want to point out that technology alone cannot resolve the complicated issue regarding publishing data for the benefit of the population and protecting individual confidentiality. However, our system can provide some clarification on the boundary between data sets, which are unlikely to cause violation of personal confidentiality, and those are. We believe that privacy policy, privacy protection legislation together with strong technological protection can make sharing data yet preserving confidentiality plausible.

References

[1] Adam, N.R. and Wortmann, J.C., *Security-Control Methods for Statistical Databases: A comparative Study.* ACM Computing Surveys, Vol. 21, No. 4, December 1989.

[2] Alexander, L. and Jabine, T., *Access to social security microdata files for research and statistical purposes.* Social Security Bulletin. (41) No. 8, 1978.

[3] Chiang, Y.-C., *Protecting privacy in public database* (in Chinese). Master's thesis, Graduate Institute of Information Management, National Taiwan University, 2000.

[4] Chiang, Y.-C. and Hsu, T.-s. and Kuo, S. and Wang, D.-W., *Preserving Confidentially When Sharing Medical Data,* In Proceedings Asia Pacific Medical Informatics Conference (APAMI-MIC), 2000.

[5] Chellas, B.F., *Modal Logic.* Cambridge, U.K.: Cambridge University Press, 1980.

[6] Denning, D.E. and Akl, S.G. and Heckman, M. and Lunt, T.F. and Morgenstern, M. and Neumann, P.G. and Schell, R. R., *Views for Multilevel Database Security.* In IEEE Transactions on Software Engineering, Vol.13, No.2, pp.129–140, February 1987.

[7] Fagin, R. and Halpern, J.Y. and Moses, Y. and Vardi, M.Y., *Reasoning about knowledge* MIT Press 1995.

[8] Ford, W.R. and O'Keefe, J. and Thuraisingham, M.B., *Database Inference Controller: An Overview.* Technical Report MTR 10963 Vol. 1. The MITRE Corporation. August 1990.

[9] Garvey, T.D. and Lunt, T.F. and Quin, X. and Stickel, M.E. *Inference Channel Detection and Elimination in Knowledge-Based Systems.* Final Report ECU 2528, SRI International, October 1994.

[10] Halpern, J.K. and Tuttle, M.R., *Knowledge, probability, and adversaries* Journal of the ACM 40(4), 917-962, 1993.

[11] Hinke, T.H., *Inference Aggregation Detection in Database Management Systems.* In Proceedings of the IEEE Symposium on Research in Security and Privacy, pp.96–106, April 1988.

[12] Hoffman, L.J. and Miller, W.F., *Getting a personal dossier from a Statistical data bank.* Datamation, vol. 16, No.5, pp.74–74, May 1970.

[13] Hsu, T.-s. and Liau, C.-J and Wang, D.-W., ”A Logical Model for Privacy Protection”, In Proceedings of Information Security Conference (ISC), Springer-Verlag LNCS# 2200, pp.110–124, 2001.

[14] Hundepool, A. and Willenborg, L., *μ and τ-argus: software for statistical disclosure control.* Third International Seminar on Statistical Confidentiality. Bled, 1996.

[15] Jajodia, S., *Aggregation and Inference Problems in Multilevel Secure Systems.* In Proceedings of the 5th Rome Laboratory Database Security Workshop, 1992.

[16] Krasucki, P. and Parikh, R. and Ndjatou, G. *Probabilistic knowledge and probabilistic common knowledge* (preliminary report). In Ras, Z. W. and Zemankova, M. and Emrich, M.L. editors, Methodologies for Intelligent Systems, volume 5, pp.1–8. Elsevier Science Publishing Co., Inc., The Hague, 1990.

[17] Sweeney, L., *Guaranteeing Anonymity When Sharing Medical Data, the Datafly System.* MIT A.I. Working Paper No. AIWP-WP334. May 1997.

[18] Morgenstern, M., *Controlling Logical Inference in Multilevel Database Systems.* In Proceedings of the IEEE Symposium on Security and Privacy, pp.245-255, April 1988.

[19] Palley, M.A., *Security of statistical databases compromise through attribute correlational modeling.* In Proceedings of IEEE Conference on Data Engineering, pp.67–74, 1986.