Alfred Menezes (Ed.)

Topics in Cryptology – CT-RSA 2005

The Cryptographers' Track at the RSA Conference 2005
San Francisco, CA, USA, February 14-18, 2005
Proceedings

Springer

Lecture Notes in Computer Science

The LNCS series reports state-of-the-art results in computer science research, development, and education, at a high level and in both printed and electronic form. Enjoying tight cooperation with the R&D community, with numerous individuals, as well as with prestigious organizations and societies, LNCS has grown into the most comprehensive computer science research forum available.

The scope of LNCS, including its subseries LNAI and LNBI, spans the whole range of computer science and information technology including interdisciplinary topics in a variety of application fields. The type of material published traditionally includes

- proceedings (published in time for the respective conference)
- post-proceedings (consisting of thoroughly revised final full papers)
- research monographs (which may be based on outstanding PhD work, research projects, technical reports, etc.)

More recently, several color-cover sublines have been added featuring, beyond a collection of papers, various added-value components; these sublines include

- tutorials (textbook-like monographs or collections of lectures given at advanced courses)
- state-of-the-art surveys (offering complete and mediated coverage of a topic)
- hot topics (introducing emergent topics to the broader community)

In parallel to the printed book, each new volume is published electronically in LNCS Online.

Detailed information on LNCS can be found at
http://www.springeronline.com

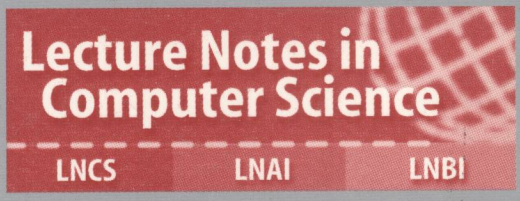
Proposals for publication should be sent to

LNCS Editorial, Tiergartenstr. 17, 69121 Heidelberg, Germany

E-mail: lncs@springer.de

ISSN 0302-9743

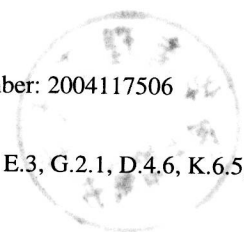
ISBN 3-540-24399-2



springeronline.com

Volume Editor

Alfred Menezes
University of Waterloo
Department of Combinatorics and Optimization
Waterloo, Ontario, N2L 3G1, Canada
E-mail: ajmeneze@uwaterloo.ca



Library of Congress Control Number: 2004117506

CR Subject Classification (1998): E.3, G.2.1, D.4.6, K.6.5, K.4.4, F.2.1-2, C.2, J.1

ISSN 0302-9743
ISBN 3-540-24399-2 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Olgun Computergrafik
Printed on acid-free paper SPIN: 11377726 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3376

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Editorial Board

Lecture Notes in Computer Science

For information about Vols. 1–3262

please contact your bookseller or Springer

Vol. 3385: R. Cousot (Ed.), Verification, Model Checking, and Abstract Interpretation. XII, 483 pages. 2004.

Vol. 3382: J. Odell, P. Giorgini, J.P. Müller (Eds.), Agent-Oriented Software Engineering V. X, 239 pages. 2004.

Vol. 3381: M. Bieliková, B. Charon-Bost, O. Sýkora, P. Vojtáš (Eds.), SOFSEM 2005: Theory and Practice of Computer Science. XV, 428 pages. 2004.

Vol. 3376: A. Menezes (Ed.), Topics in Cryptology – CT-RSA 2005. X, 385 pages. 2004.

Vol. 3363: T. Eiter, L. Libkin (Eds.), Database Theory - ICDT 2005. XI, 413 pages. 2004.

Vol. 3362: G. Barthe, L. Burdy, M. Huisman, J.-L. Lanet, T. Muntean (Eds.), Construction and Analysis of Safe, Secure, and Interoperable Smart Devices. IX, 257 pages. 2004.

Vol. 3360: S. Spaccapietra, E. Bertino, S. Jajodia, R. King, D. McLeod, M.E. Orlowska, L. Strous (Eds.), Journal on Data Semantics II. XI, 233 pages. 2004.

Vol. 3358: J. Cao, L.T. Yang, M. Guo, F. Lau (Eds.), Parallel and Distributed Processing and Applications. XXIV, 1058 pages. 2004.

Vol. 3357: H. Handschuh, M.A. Hasan (Eds.), Selected Areas in Cryptography. XI, 355 pages. 2004.

Vol. 3356: G. Das, V.P. Gulati (Eds.), Intelligent Information Technology. XII, 428 pages. 2004.

Vol. 3353: J. Hromkovič, M. Nagl, B. Westfechtel (Eds.), Graph-Theoretic Concepts in Computer Science. XI, 404 pages. 2004.

Vol. 3350: M. Hermenegildo, D. Cabeza (Eds.), Practical Aspects of Declarative Languages. VIII, 269 pages. 2004.

Vol. 3348: A. Canteaut, K. Viswanathan (Eds.), Progress in Cryptology - INDOCRYPT 2004. XIV, 431 pages. 2004.

Vol. 3347: R.K. Ghosh, H. Mohanty (Eds.), Distributed Computing and Internet Technology. XX, 472 pages. 2004.

Vol. 3344: J. Malenfant, B.M. Østvold (Eds.), Object-Oriented Technology. ECOOP 2004 Workshop Reader. VIII, 215 pages. 2004.

Vol. 3342: E. Şahin, W.M. Spears (Eds.), Swarm Robotics. X, 175 pages. 2004.

Vol. 3341: R. Fleischer, G. Trippen (Eds.), Algorithms and Computation. XVII, 935 pages. 2004.

Vol. 3340: C.S. Calude, E. Calude, M.J. Dinneen (Eds.), Developments in Language Theory. XI, 431 pages. 2004.

Vol. 3339: G.I. Webb, X. Yu (Eds.), AI 2004: Advances in Artificial Intelligence. XXII, 1272 pages. 2004. (Subseries LNAI).

Vol. 3338: S.Z. Li, J. Lai, T. Tan, G. Feng, Y. Wang (Eds.), Advances in Biometric Person Authentication. XVIII, 699 pages. 2004.

Vol. 3337: J.M. Barreiro, F. Martin-Sanchez, V. Maojo, F. Sanz (Eds.), Biological and Medical Data Analysis. XI, 508 pages. 2004.

Vol. 3336: D. Karagiannis, U. Reimer (Eds.), Practical Aspects of Knowledge Management. X, 523 pages. 2004. (Subseries LNAI).

Vol. 3334: Z. Chen, H. Chen, Q. Miao, Y. Fu, E. Fox, E.-p. Lim (Eds.), Digital Libraries: International Collaboration and Cross-Fertilization. XX, 690 pages. 2004.

Vol. 3333: K. Aizawa, Y. Nakamura, S. Satoh (Eds.), Advances in Multimedia Information Processing - PCM 2004, Part III. XXXV, 785 pages. 2004.

Vol. 3332: K. Aizawa, Y. Nakamura, S. Satoh (Eds.), Advances in Multimedia Information Processing - PCM 2004, Part II. XXXVI, 1051 pages. 2004.

Vol. 3331: K. Aizawa, Y. Nakamura, S. Satoh (Eds.), Advances in Multimedia Information Processing - PCM 2004, Part I. XXXVI, 667 pages. 2004.

Vol. 3329: P.J. Lee (Ed.), Advances in Cryptology - ASIACRYPT 2004. XVI, 546 pages. 2004.

Vol. 3328: K. Lodaya, M. Mahajan (Eds.), FSTTCS 2004: Foundations of Software Technology and Theoretical Computer Science. XVI, 532 pages. 2004.

Vol. 3326: A. Sen, N. Das, S.K. Das, B.P. Sinha (Eds.), Distributed Computing - IWDC 2004. XIX, 546 pages. 2004.

Vol. 3323: G. Antoniou, H. Boley (Eds.), Rules and Rule Markup Languages for the Semantic Web. X, 215 pages. 2004.

Vol. 3322: R. Klette, J. Žunić (Eds.), Combinatorial Image Analysis. XII, 760 pages. 2004.

Vol. 3321: M.J. Maher (Ed.), Advances in Computer Science - ASIAN 2004. XII, 510 pages. 2004.

Vol. 3320: K.-M. Liew, H. Shen, S. See, W. Cai (Eds.), Parallel and Distributed Computing: Applications and Technologies. XXIV, 891 pages. 2004.

Vol. 3317: M. Domaratzki, A. Okhotin, K. Salomaa, S. Yu (Eds.), Implementation and Application of Automata. XII, 336 pages. 2004.

Vol. 3316: N.R. Pal, N.K. Kasabov, R.K. Mudi, S. Pal, S.K. Parui (Eds.), Neural Information Processing. XXX, 1368 pages. 2004.

Vol. 3315: C. Lemaître, C.A. Reyes, J.A. González (Eds.), Advances in Artificial Intelligence – IBERAMIA 2004. XX, 987 pages. 2004. (Subseries LNAI).

Vol. 3314: J. Zhang, J.-H. He, Y. Fu (Eds.), Computational and Information Science. XXIV, 1259 pages. 2004.

Vol. 3312: A.J. Hu, A.K. Martin (Eds.), Formal Methods in Computer-Aided Design. XI, 445 pages. 2004.

Vol. 3311: V. Roca, F. Rousseau (Eds.), Interactive Multimedia and Next Generation Networks. XIII, 287 pages. 2004.

Vol. 3309: C.-H. Chi, K.-Y. Lam (Eds.), Content Computing. XII, 510 pages. 2004.

Vol. 3308: J. Davies, W. Schulte, M. Barnett (Eds.), Formal Methods and Software Engineering. XIII, 500 pages. 2004.

Vol. 3307: C. Bussler, S.-k. Hong, W. Jun, R. Kaschek, D.. Kinshuk, S. Krishnaswamy, S.W. Loke, D. Oberle, D. Richards, A. Sharma, Y. Sure, B. Thalheim (Eds.), Web Information Systems – WISE 2004 Workshops. XV, 277 pages. 2004.

Vol. 3306: X. Zhou, S. Su, M.P. Papazoglou, M.E. Orlowska, K.G. Jeffery (Eds.), Web Information Systems – WISE 2004. XVII, 745 pages. 2004.

Vol. 3305: P.M.A. Sloot, B. Chopard, A.G. Hoekstra (Eds.), Cellular Automata. XV, 883 pages. 2004.

Vol. 3303: J.A. López, E. Benfenati, W. Dubitzky (Eds.), Knowledge Exploration in Life Science Informatics. X, 249 pages. 2004. (Subseries LNAI).

Vol. 3302: W.-N. Chin (Ed.), Programming Languages and Systems. XIII, 453 pages. 2004.

Vol. 3300: L. Bertossi, A. Hunter, T. Schaub (Eds.), Inconsistency Tolerance. VII, 295 pages. 2004.

Vol. 3299: F. Wang (Ed.), Automated Technology for Verification and Analysis. XII, 506 pages. 2004.

Vol. 3298: S.A. McIlraith, D. Plexousakis, F. van Harmelen (Eds.), The Semantic Web – ISWC 2004. XXI, 841 pages. 2004.

Vol. 3296: L. Bougé, V.K. Prasanna (Eds.), High Performance Computing - HiPC 2004. XXV, 530 pages. 2004.

Vol. 3295: P. Markopoulos, B. Eggen, E. Aarts, J.L. Crowley (Eds.), Ambient Intelligence. XIII, 388 pages. 2004.

Vol. 3294: C.N. Dean, R.T. Boute (Eds.), Teaching Formal Methods. X, 249 pages. 2004.

Vol. 3293: C.-H. Chi, M. van Steen, C. Wills (Eds.), Web Content Caching and Distribution. IX, 283 pages. 2004.

Vol. 3292: R. Meersman, Z. Tari, A. Corsaro (Eds.), On the Move to Meaningful Internet Systems 2004: OTM 2004 Workshops. XXIII, 885 pages. 2004.

Vol. 3291: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, Part II. XXV, 824 pages. 2004.

Vol. 3290: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, Part I. XXV, 823 pages. 2004.

Vol. 3289: S. Wang, K. Tanaka, S. Zhou, T.W. Ling, J. Guan, D. Yang, F. Grandi, E. Mangina, I.-Y. Song, H.C. Mayr (Eds.), Conceptual Modeling for Advanced Application Domains. XXII, 692 pages. 2004.

Vol. 3288: P. Atzeni, W. Chu, H. Lu, S. Zhou, T.W. Ling (Eds.), Conceptual Modeling – ER 2004. XXI, 869 pages. 2004.

Vol. 3287: A. Sanfeliu, J.F. Martínez Trinidad, J.A. Carrasco Ochoa (Eds.), Progress in Pattern Recognition, Image Analysis and Applications. XVII, 703 pages. 2004.

Vol. 3286: G. Karsai, E. Visser (Eds.), Generative Programming and Component Engineering. XIII, 491 pages. 2004.

Vol. 3285: S. Manandhar, J. Austin, U.B. Desai, Y. Oyanagi, A. Talukder (Eds.), Applied Computing. XII, 334 pages. 2004.

Vol. 3284: A. Karmouch, L. Korba, E.R.M. Madeira (Eds.), Mobility Aware Technologies and Applications. XII, 382 pages. 2004.

Vol. 3283: F.A. Aagesen, C. Anutariya, V. Wuwongse (Eds.), Intelligence in Communication Systems. XIII, 327 pages. 2004.

Vol. 3282: V. Guruswami, List Decoding of Error-Correcting Codes. XIX, 350 pages. 2004.

Vol. 3281: T. Dingsøyr (Ed.), Software Process Improvement. X, 207 pages. 2004.

Vol. 3280: C. Aykanat, T. Dayar, İ. Körpeoğlu (Eds.), Computer and Information Sciences - ISCIS 2004. XVIII, 1009 pages. 2004.

Vol. 3279: G.M. Voelker, S. Shenker (Eds.), Peer-to-Peer Systems III. XI, 300 pages. 2004.

Vol. 3278: A. Sahai, F. Wu (Eds.), Utility Computing. XI, 272 pages. 2004.

Vol. 3275: P. Perner (Ed.), Advances in Data Mining. VIII, 173 pages. 2004. (Subseries LNAI).

Vol. 3274: R. Guerraoui (Ed.), Distributed Computing. XIII, 465 pages. 2004.

Vol. 3273: T. Baar, A. Strohmeier, A. Moreira, S.J. Mellor (Eds.), <<UML>> 2004 - The Unified Modelling Language. XIII, 454 pages. 2004.

Vol. 3272: L. Baresi, S. Dustdar, H. Gall, M. Matera (Eds.), Ubiquitous Mobile Information and Collaboration Systems. VIII, 197 pages. 2004.

Vol. 3271: J. Vicente, D. Hutchison (Eds.), Management of Multimedia Networks and Services. XIII, 335 pages. 2004.

Vol. 3270: M. Jeckle, R. Kowalczyk, P. Braun (Eds.), Grid Services Engineering and Management. X, 165 pages. 2004.

Vol. 3269: J. Lopez, S. Qing, E. Okamoto (Eds.), Information and Communications Security. XI, 564 pages. 2004.

Vol. 3268: W. Lindner, M. Mesiti, C. Türker, Y. Tzitzikas, A. Vakali (Eds.), Current Trends in Database Technology - EDBT 2004 Workshops. XVIII, 608 pages. 2004.

Vol. 3267: C. Priami, P. Quaglia (Eds.), Global Computing. VIII, 377 pages. 2004.

Vol. 3266: J. Solé-Pareta, M. Smirnov, P.V. Mieghem, J. Domingo-Pascual, E. Monteiro, P. Reichl, B. Stiller, R.J. Gibbens (Eds.), Quality of Service in the Emerging Networking Panorama. XVI, 390 pages. 2004.

Vol. 3265: R.E. Frederking, K.B. Taylor (Eds.), Machine Translation: From Real Users to Research. XI, 392 pages. 2004. (Subseries LNAI).

Vol. 3264: G. Paliouras, Y. Sakakibara (Eds.), Grammatical Inference: Algorithms and Applications. XI, 291 pages. 2004. (Subseries LNAI).

Vol. 3263: M. Weske, P. Liggesmeyer (Eds.), Object-Oriented and Internet-Based Technologies. XII, 239 pages. 2004.

Preface

The RSA Conference is attended by over 10,000 security professionals each year. The Cryptographers' Track (CT-RSA), one of several parallel tracks at the conference, provides an excellent opportunity for cryptographers to showcase their research to a wide audience. CT-RSA 2005 was the fifth year of the Cryptographers' Track.

The selection process for the CT-RSA program is the same as for other cryptography research conferences. This year, the program committee selected 23 papers from 74 submissions (two of which were later withdrawn) that covered all aspects of cryptography. The program also included two invited talks by Cynthia Dwork and Moti Yung. These proceedings contain the revised versions of the selected papers. The revisions were not checked, and so the authors (and not the committee) bear full responsibility for the contents of their papers.

I am very grateful to the program committee for their very conscientious efforts to review each paper fairly and thoroughly. The initial review stage was followed by a tremendous amount of discussion which contributed to our high confidence in our judgements. Thanks also to the many external reviewers whose names are listed in the following pages. My apologies to those whose names were inadvertently omitted from this list.

Thanks to Eddie Ng for maintaining the submission server and the Web review system. The submission software was written by Chanathip Namprempre, and the Web review software by Wim Moreau and Joris Claessens. Thanks to Alfred Hofmann and his colleagues at Springer for the timely production of these proceedings. Finally, it is my pleasure to acknowledge Ari Juels and Mike Szydlo of RSA Laboratories for their assistance and cooperation during the past seven months.

October 2004 Alfred Menezes

RSA Cryptographers' Track 2005
February 14–18, 2005, San Francisco, CA, USA

The RSA Conference 2005 was organized by RSA Security Inc. and its partner organizations around the world. The Cryptographers' Track was organized by RSA Laboratories.

Program Chair

Alfred Menezes, University of Waterloo, Canada

Program Committee

Masayuki Abe ..... NTT Laboratories, Japan
Paulo Barreto ..... Scopus Tecnologia, Brazil
Alex Biryukov ..... K.U.Leuven, Belgium
John-Sebastien Coron ..... Gemplus, France
Steven Galbraith ..... Royal Holloway, University of London, UK
Amir Herzberg ..... Bar-Ilan University, Israel
Yuval Ishai ..... Technion, Israel
Stanislaw Jarecki ..... UC Irvine, USA
Lars Knudsen ..... Technical University of Denmark
Kaoru Kurosawa ..... Ibaraki University, Japan
Tanja Lange ..... Ruhr-Universität, Bochum, Germany
Helger Lipmaa ..... Helsinki University of Technology, Finland
Philip MacKenzie ..... DoCoMo, USA
Tal Malkin ..... Columbia University, USA
Wenbo Mao ..... HP Laboratories, UK
Ilya Mironov ..... Microsoft Research, USA
Josef Pieprzyk ..... Macquarie University, Australia
Palash Sarkar ..... Indian Statistical Institute, India
Jessica Staddon ..... Palo Alto Research Center, USA
Rene Struik ..... Certicom, Canada
Michael Szydlo ..... RSA Laboratories, USA
Tsuyoshi Takagi ..... TU Darmstadt, Germany

Steering Committee

Marc Joye ..... Gemplus, France
Tatsuaki Okamoto ..... NTT, Japan
Bart Preneel ..... K.U.Leuven, Belgium
Ron Rivest ..... MIT, USA
Moti Yung ..... Columbia University, USA

External Reviewers

Toru Akishita
Alexandr Andoni
Roberto Avanzi
Sara Bitan
Alexandra Boldyreva
Reinier Bröker
Daniel Brown
Bertrand Byramjee
Christophe De Canniere
Dario Catalano
Liqun Chen
Joe Cho
Carlos Cid
Mathieu Ciet
Scott Contini
Claus Diem
Yevgeniy Dodis
Eiichiro Fujisaki
Juan Garay
Craig Gentry
Philippe Golle
Shai Halevi
Darrel Hankerson
Heng Swee Huay
David Hwang
Kouichi Itoh
Tetsu Iwata
Antoine Joux
Masanobu Katagi
Jonathan Katz
Jeff King
Lea Kissner
Yuichi Komano
Hugo Krawczyk
Caroline Kudla
Joseph Lano
Kerstin Lemke
John Linn
Anna Lysyanskaya
Dahlia Malkhi
Daniele Micciancio
Anton Mityagin
Atsuko Miyaji
David Molnar
Michael Mueller
Jorge Nakahara
Wakaha Ogata
Kazuo Ohata
Yasuhiro Ohtaki
Akira Otsuka
Pascal Paillier
Zulfikar Ramzan
Leo Reyzin
Matt Robshaw
Markku-Juhani Saarinen
Taiichi Saito
Akashi Satoh
Kai Schramm
Daniel Schepers
Igor Shparlinski
Nigel Smart
Angelos Stavrou
Ron Steinfeld
Makoto Sugita
Matti Tommiska
Eran Tromer
Huaxiong Wang
Michael Wiener
Kai Wirt
Christopher Wolf
Shoko Yonezawa
Yunlei Zhao

Table of Contents

Design Principles

Password-Based Protocols

Pairings

Efficient and Secure Implementation

Sub-linear Queries Statistical Databases: Privacy with Power

Cynthia Dwork

Microsoft Research
dwork@microsoft.com

Abstract. We consider a statistical database in which a trusted administrator introduces noise to the query responses with the goal of maintaining privacy of individual database entries. In such a database, a query consists of a pair (S, f) where S is a set of rows in the database and f is a function mapping database rows to $\{0,1\}$. The true response is $\sum_{r \in S} f(DB_r)$, a noisy version of which is released. Results in [3, 4] show that a strong form of privacy can be maintained using a surprisingly small amount of noise, provided the total number of queries is sublinear in the number n of database rows. We call this a sub-linear queries (SuLQ) database. The assumption of sublinearity becomes reasonable as databases grow increasingly large.
The SuLQ primitive – query and noisy reply – gives rise to a calculus of noisy computation. After reviewing some results of [4] on multi-attribute SuLQ, we illustrate the power of the SuLQ primitive with three examples [2]: principal component analysis, k means clustering, and learning in the statistical queries learning model.

1 Introduction

Consider a statistical database in which a trusted administrator introduces noise to the query responses with the goal of maintaining privacy of individual database entries. For concreteness, let the database consist of some number n of rows $DB_1, \ldots, DB_n$, where each row is a d-tuple of Boolean values. A query consists of a pair (S, f) where $S \subseteq [n]$ is a set of rows in the database and $f : \{0,1\}^d \to \{0,1\}$ is a function mapping database rows to $\{0,1\}$. The true response to the query is $\sum_{r \in S} f(DB_r)$, a noisy version of which is released. That is, the administrator algorithm chooses a random quantity in some range and releases the sum of the true response and the random quantity.

Such databases were studied extensively in the early 1980's (see [1] for an excellent survey of results on these and other techniques for statistical disclosure control), with mixed results. However, results in [3, 4] show that a strong form of privacy can be maintained using a surprisingly small amount of noise – a random quantity whose standard deviation is of order $o(\sqrt{n})$ – *provided the total number of queries is sublinear in the number n of database rows.*

This is significant for the following reason. If we think of each row as a sample from some underlying probability distribution and we wish to gather statistics

A.J. Menezes (Ed.): CT-RSA 2005, LNCS 3376, pp. 1–6, 2005.

on a properties P that occur with possibly small but still constant probability in the population, then the sampling error in our population of size n will be of order $\Omega(\sqrt{n})$. Thus, the noise that is added for the sake of protecting privacy is significantly smaller than the sampling error. In other words, providing privacy need not interfere with accuracy, so long as the number of statistical queries is not too large. The assumption of sublinearity is reasonable as databases grow increasingly large.

The basic SuLQ primitive – noisy sums of arbitrary Boolean functions applied to each row in a set $S \subseteq [n]$ of rows – is powerful: statistics for any d-ary predicate can be very accurately obtained simply by querying the database. It is natural to ask, "Which more complex computations can be expressed using few (in n) SuLQ queries?" We have found this class to be quite rich.

Here, we review the results of [4] on multi-attribute SuLQ databases (Section 3) and then give three examples of the power of the SuLQ primitive (Section 4): principal component analysis, k means clustering, and learning in the statistical queries learning model. The treatment here is informal and without proofs. Rigorous treatment of these and other, related, results, is given in [4, 2].

2 Definitions

We model a database as an $n \times d$ binary matrix $DB = \{DB_{i,j}\}$. Intuitively, the columns in DB correspond to Boolean attributes $\alpha_1, \ldots, \alpha_d$, and the rows in DB correspond to individuals, where $DB_{i,j} = 1$ iff attribute α_j holds for individual i.

Let $\mathcal{D}$ be a distribution on $\{0,1\}^d$. We say that a database $DB = \{DB_{i,j}\}$ is chosen according to distribution $\mathcal{D}$ if every row in DB is chosen according to $\mathcal{D}$, independently of the other rows (in other words, DB is chosen according to $\mathcal{D}^n$). To capture partial information that the adversary may have obtained about individuals prior to interacting with the database, this requirement is relaxed in the privacy analysis, allowing each row i to be chosen from a (possibly) different distribution $\mathcal{D}_i$. In that case we say that the database is chosen according to $\mathcal{D}_1 \times \cdots \times \mathcal{D}_n$.

For a Boolean function $f : \{0,1\}^d \to \{0,1\}$ we let $p_0^{i,f}$ be the *a priori* probability that $f(DB_{i,1}, \ldots, DB_{i,d}) = 1$ and $p_T^{i,f}$ be the *a posteriori* probability that $f(DB_{i,1}, \ldots, DB_{i,d}) = 1$, given the answers to T queries, *as well as* all the values in all the rows of DB other than i: $DB_{i'}$ for all $i' \neq i$.

We define the monotonically-increasing 1-1 mapping $\mathrm{conf} : (0,1) \to \mathbb{R}$ as follows:

$$\mathrm{conf}(p) = \log \frac{p}{1-p}.$$

Note that a small additive change in conf implies a small additive change in p [1].

[1] The converse does not hold: conf grows logarithmically in p for $p \approx 0$ and logarithmically in $1/(1-p)$ for $p \approx 1$.

Let $\text{conf}_0^{i,f} = \log \frac{p_0^{i,f}}{1-p_0^{i,f}}$ and $\text{conf}_T^{i,f} = \log \frac{p_T^{i,f}}{1-p_T^{i,f}}$. We write our privacy requirements in terms of the random variables $\Delta\text{conf}^{i,f}$ defined as[2]:

$$\Delta\text{conf}^{i,f} = |\text{conf}_T^{i,f} - \text{conf}_0^{i,f}|.$$

Definition 1 ((δ, T)-Privacy). *A database access mechanism is (δ, T)-private if for every distribution $\mathcal{D}$ on $\{0,1\}^d$, for every row index i, for every function $f : \{0,1\}^d \to \{0,1\}$, and for every adversary $\mathcal{A}$ making at most T queries*

$$\Pr[\Delta\text{conf}^{i,f} > \delta] \leq \text{neg}(n),$$

where $\text{neg}(n)$ grows more slowly than the inverse of any polynomial in n. The probability is taken over the choice of each row in DB according to $\mathcal{D}$, and the randomness of the adversary as well as the database access mechanism.

The definition of (δ, T)-privacy speaks of the probability that any single function experiences a change in confidence. The next definitions speak about sets of functions that together experience little change in confidence.

A *target set* F is a set of d-ary Boolean functions (one can think of the functions in F as being selected by an adversary; they represent information the adversary may wish to learn about someone). A target set F is *δ-safe* if $\Delta\text{conf}^{i,f} \leq \delta$ for all $i \in [n]$ and $f \in F$. Let F be a target set of size polynomial in n. Definition 1 implies that under a (δ, T)-private database mechanism, F is δ-safe with probability $1 - \text{neg}(n)$.

Claim. [4] Consider a (δ, T)-private database with $d = O(\log n)$ attributes. Let F be the target set containing all the 2^{2^d} Boolean functions over the d attributes. Then, $\Pr[F \text{ is } 2\delta\text{-safe}] = 1 - \text{neg}(n)$.

3 Multi-attribute SuLQ Databases

The multi-attribute SuLQ database of [4] is easy to describe. Let $T = T(n) = O(n^c)$, $c < 1$, and define $R = \left(T(n)/\delta^2\right) \cdot \log^\mu n$ for some $\mu > 0$ (taking $\mu = 6$ will work).

SuLQ Database Algorithm $\mathcal{A}$
Input: a query (S, g).

1. Let $a_{S,g} = \sum_{i \in S} g(DB_i)$.
2. Generate a perturbation value: Let $(e_1, \ldots, e_R) \in_R \{0,1\}^R$ and $\mathcal{E} \leftarrow \sum_{i=1}^{R} e_i - R/2$.
3. Return $\tilde{a}_{S,g} = a_{S,g} + \mathcal{E}$.

[2] Our choice of defining privacy in terms of $\Delta\text{conf}^{i,f}$ is somewhat arbitrary, one could rewrite our definitions (and analysis) in terms of the a priori and a posteriori probabilities. Note however that limiting $\Delta\text{conf}^{i,f}$ in Definition 1 is a stronger requirement than just limiting $|p_T^{i,f} - p_0^{i,f}|$.

Note that $\mathcal{E}$ is a binomial random variable with $\mathbf{E}[\mathcal{E}] = 0$ and standard deviation $\sqrt{R}$. The analysis ignores the case where $\mathcal{E}$ largely deviates from zero, as the probability of such an event is extremely small: $\Pr[|\mathcal{E}| > \sqrt{R}\log^2 n] = \mathsf{neg}(n)$. In particular, this implies that the SuLQ database algorithm $\mathcal{A}$ is within $\tilde{O}(\sqrt{T(n)})$ perturbation, meaning that for every query (S, f)

$$\Pr[|\mathcal{A}(S, f) - a_{S,f}| \leq \mathcal{E}] = 1 - \mathsf{neg}(n).$$

The probability is taken over the randomness of the database algorithm $\mathcal{A}$.

Theorem 1. *[4] Let $T(n) = O(n^c)$ and $\delta = 1/O(n^{c'})$ for $0 < c < 1$ and $0 \leq c' < c/2$. Then the SuLQ algorithm $\mathcal{A}$ is $(\delta, T(n))$-private within $\tilde{O}(\sqrt{T(n)}/\delta)$ perturbation.*

Note that whenever $\sqrt{T(n)}/\delta < \sqrt{n}$, restricting the adversary to $T(n)$ queries allows privacy with perturbation magnitude less than $\sqrt{n}$.

Let $i \in [n]$ and $f : \{0,1\}^d \to \{0,1\}$. The proof analyzes the *a posteriori* probability p_ℓ that $f(DB_i) = 1$ given the answers to the first ℓ queries $(\tilde{a}_1, \ldots, \tilde{a}_\ell)$ *and* $DB^{\{-i\}}$ (where $DB^{\{-i\}}$ denotes the entire database except for the ith row). Let $\mathrm{conf}_\ell = \log_2 p_\ell/(1-p_\ell)$. Note that $\mathrm{conf}_T = \mathrm{conf}_T^{i,f}$, and (due to the independence of rows in DB) $\mathrm{conf}_0 = \mathrm{conf}_0^{i,f}$. Following [3], a random walk on the real line is defined, with $\mathrm{step}_\ell = \mathrm{conf}_\ell - \mathrm{conf}_{\ell-1}$. The proof argues that (with high probability) $T(n)$ steps of the random walk do not suffice to reach distance δ.

4 Computation with the SuLQ Primitive

The basic SuLQ operation – query and noisy reply – can be viewed as a noisy computational primitive which may be used to compute other functions of the database than statistical queries. In this section we describe three examples of the power of the primitive. In this setting, the inputs are reals drawn from the unit d-dimensional cube, and the noise is distributed according to a normal variable $N(0, R)$, where $R = R(n)$ is roughly of order $T(n)\log n \log T(n)$. The privacy analysis in the proof of Theorem 1 must be extended accordingly. A rigorous treatment of this work appears in [2].

4.1 Principal Component Analysis

Principal component analysis [6] is an extremely valuable tool in the (frequent) case in which high-dimensional data lies primarily in a low-dimensional subspace.

The input consists of n points in $\mathcal{U}^d$ (the d-dimensional cube of side length 1) and an integer $k \leq d$. The output will be the k largest eigenvalues of the $d \times d$ covariance matrix (defined below), and their corresponding eigenvectors.

For $1 \leq i \leq d$, we let $\mu_i = E_{r\in[n]}[p_r(i)]$,where $p_r(i)$ denotes the ith coordinate of the input point described by row r. We let the $d \times d$ covariance matrix C be defined by $C = \{c_{ij}\}$, where

$$c_{ij} = E_{r\in[n]}[p_r(i)p_r(j)] - \mu_i\mu_j.$$

PCA is known to be remarkably stable under random noise – so much so, that it is often used with the express intention of *removing* noise.

SuLQ Computation of PCA

1. (d queries) For $0 \leq i \leq d$, let $m_i = SuLQ(F(x) := x(i))/n$. By this we mean that $F(x)$ selects the ith coordinate of each row, so the query sums all the ith coordinates (getting a noisy version of this sum), and the algorithm divides this noisy sum by n. This gives an approximation to μ_i in the pure PCA algorithm described above.
2. (Roughly $d^2/2$ queries) Let $c_{ij} = SuLQ(F(x) = x(i)x(j))/n - m_i m_j$. That is, we first obtain a noisy average of the product of the ith and jth coordinates, and then subtract the product of the estimates of μ_i and μ_j.

Given (an approximation to) the covariance matrix C, the k largest eigenvalues and corresponding eigenvectors can be computed directly, without further queries.

We remark that, using the techniques of [4] for vertically partitioned databases, this computation can be carried out even if each column of the database is stored in a separate, independent, SuLQ database.

4.2 k Means

An instance of the k means computation is a set of n points in $\mathcal{U}^d$, together with some number k of initial candidate "means" in $\mathcal{U}^d$. The output will consist of k points in $\mathcal{U}^d$ (the "means"), together with the fraction of points in the database associated with each mean. We next describe the basic step of the k means algorithm.

Basic Step of k Means Algorithm

1. (k queries): For each mean m_i, $1 \leq i \leq k$, count the number of points closer to this mean than to every other mean. This yields cluster sizes. This is approximated via the queries, for $1 \leq i \leq k$,

   $\text{Size}_i = SuLQ(F(x) := 1$ if m_i is the closest mean to x, and 0 otherwise).

2. (kd queries): for each mean m_i, $1 \leq i \leq k$, and coordinate j, $1 \leq j \leq d$, compute the sum, over all points in the cluster associated with m_i, of the value of the jth coordinate. Divide by the size of the cluster.
   (a) $\text{Sum}_{ij} = SuLQ(F(x) := x(j)$ if m_i is the closest center to x, and 0 otherwise).
   (b) $m_{ij} = \text{Sum}_{ij}/\text{Size}_i$

The basic step is iterated until some maximum number of queries have been issued. (In practice, this usually converges after a small number of basic steps.) If any cluster size is below a threshhold (say, $\sqrt{T(n)}$), then output an exception.

For clusters that are of size $\Omega(n)$, one step of the (pure) k means computation differs from one step of the SuLQ-based k means computation by a quantity that is roughly Gaussian with mean zero and variance $\tilde{O}(\sqrt{R}/n)$.

4.3 Capturing the Statistical Queries Learning Model

The Statistical Queries Learning model was proposed by Kearns [5]. In this model the goal is to learn a *concept* $c : \{0,1\}^d \to \{0,1\}$. There is a distribution D on strings in $\{0,1\}^d$, and the learning algorithm has access to an oracle, $\text{stat}_{c,D}$, described next.

On query (f, τ), where $f = f(x, \ell)$ is any boolean function over inputs $x \in D$ and label $\ell \in \{0,1\}$, and $\tau = 1/poly(d)$ is an error tolerance, the oracle replies with a noisy estimate of the probability that $f(x, c(x)) = 1$ for a randomly selected element from D; the answer is guaranteed to be correct within additive tolerance τ. Many (but not all, see [5]) concept classes that are PAC learnable can also be learned in the statistical queries learning model.

To fit the statistical queries learning model into our setting, we require that one of the attributes be the value of c applied to the other data in the row, so that a typical row looks like $DB_r = (x, c(x))$. By definition, on input (f, S) the SuLQ database responds with a noisy version of $\sum_{r \in S} f(DB_r)$. Taking $S = [n]$, we have that so long as the noise added by the SuLQ database is within the tolerance τ, the response (divided by n) is a "valid" response of the $\text{stat}_{c,D}$ oracle. In other words, to simulate the query $\text{stat}_{c,D}(f, \tau)$ we compute $\text{SuLQ}(F(x) := f(x))/n$ a total of $\tilde{O}(R/\tau^2 n^2)$ times and return the average of these values.

With high probability the answer obtained will be within tolerance τ. Also, recall that $\tau = 1/poly(d)$; if $d = n^{o(1)}$ then repetition is not necessary.

References

1. N.R. Adam and J.C. Wortmann, Security-Control Methods for Statistical Databases: A Comparative Study, *ACM Computing Surveys 21*(4), pp. 515–556, 1989.
2. A. Blum, C. Dwork, F. McSherry, and K. Nissim, On the Power of SuLQ Databases, *manuscript in preparation*, 2004.
3. I. Dinur and K. Nissim, Revealing information while preserving privacy, *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pp. 202-210, 2003.
4. C. Dwork and N. Nissim, Privacy-Preserving Datamining on Vertically Partitioned Databases, *Proceedings of CRYPTO 2004*
5. M. Kearns, Efficient Noise-Tolerant Learning from Statistical Queries, *JACM 45*(6), pp. 983–1006, 1998. See also *Proc. 25th ACM STOC*, pp. 392–401, 1993
6. M. J. O'Connel, Search Program for Significant Variables, *Comp. Phys. Comm. 8*, 1974.