Willem Jonker
Milan Petković (Eds.)

LNCS 4165

Secure Data Management

Third VLDB Workshop, SDM 2006
Seoul, Korea, September 2006
Proceedings

Springer

Willem Jonker Milan Petković (Eds.)

Secure Data Management

Third VLDB Workshop, SDM 2006
Seoul, Korea, September 10-11, 2006
Proceedings

Springer

Volume Editors

Willem Jonker
Philips Research Europe
High Tech Campus 34
5656 AE Eindhoven
The Netherlands
E-mail: willem.jonker@philips.com

Milan Petković
Philips Research Laboratories
High Tech Campus 34
5656 AE Eindhoven
The Netherlands
E-mail: Milan.Petkovic@philips.com

Library of Congress Control Number: 2006931629

CR Subject Classification (1998): H.2.0, H.2, C.2.0, H.3, E.3, D.4.6, K.6.5

LNCS Sublibrary: SL 3 – Information Systems and Application, incl. Internet/Web and HCI

ISSN	0302-9743
ISBN-10	3-540-38984-9 Springer Berlin Heidelberg New York
ISBN-13	978-3-540-38984-2 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springer.com

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11844662 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 4165

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

Preface

Recent developments in computer, communication, and information technologies, along with increasingly interconnected networks and mobility have established new emerging technologies, such as ubiquitous computing and ambient intelligence, as a very important and unavoidable part of everyday life. However, this development has greatly influenced people's security concerns. As data is accessible anytime from anywhere, according to these new concepts, it becomes much easier to get unauthorized data access. As another consequence, the use of new technologies has brought some privacy concerns. It becomes simpler to collect, store, and search personal information and endanger people's privacy. Therefore, research in the area of secure data management is of growing importance, attracting the attention of both the data management and security research communities. The interesting problems range from traditional ones such as access control (with all variations, like role-based and/or context-aware), database security, operations on encrypted data, and privacy preserving data mining to cryptographic protocols.

The call for papers attracted 33 papers both from universities and industry. The program committee selected 13 research papers for presentation at the workshop. These papers are also collected in this volume, which we hope will serve you as useful research and reference material.

The volume is divided roughly into four major sections. The first section focuses on privacy protection addressing the topics of indistinguishability, sovereign information sharing, data anonymization, and privacy protection in ubiquitous environments. The second section changes slightly the focal point to privacy preserving data management. The papers in this section deal with search on encrypted data and privacy preserving clustering. The third section focuses on access control which remains an important area of interest. The papers cover role-based access control, XML access control and conflict resolution. The last section addresses database security topics.

Finally, let us acknowledge the work of Richard Brinkman, who helped in the technical preparation of these proceedings.

July 2006 Willem Jonker and Milan Petković

Organization

Workshop Organizers

Willem Jonker (Philips Research/University of Twente, The Netherlands)
Milan Petković (Philips Research, The Netherlands)

Program Committee

Gerrit Bleumer, Francotyp-Postalia, Germany
Ljiljana Branković, University of Newcastle, Australia
Sabrina De Capitani di Vimercati, University of Milan, Italy
Ernesto Damiani, University of Milan, Italy
Eric Diehl, Thomson Research, France
Csilla Farkas, University of South Carolina, USA
Ling Feng, Twente University, Netherlands
Eduardo Fernández-Medina, University of Castilla-La Mancha, Spain
Elena Ferrari, Università degli Studi dell'Insubria, Italy
Simone Fischer-Hübner, Karlstad University, Sweden
Tyrone Grandison, IBM Almaden Research Center, USA
Ehud Gudes, Ben-Gurion University, Israel
Hacan Hacigümüş, IBM Almaden Research Center, USA
Marit Hansen, Independent Centre for Privacy Protection, Germany
Pieter Hartel, Twente University, The Netherlands
Dong Hoon Lee, Korea University, Korea
Mizuho Iwaihara, Kyoto University, Japan
Sushil Jajodia George Mason University, USA
Ton Kalker, HP Research, USA
Marc Langheinrich, Institute for Pervasive Computing ETH Zurich, Switzerland
Nick Mankovich, Philips Medical Systems, USA
Sharad Mehrotra, University of California at Irvine, USA
Stig Frode Mjølsnes, Norwegian University of Science and Technology, Norway
Eiji Okamoto, University of Tsukuba, Japan
Sylvia Osborn, University of Western Ontario, Canada
Günther Pernul, University of Regensburg, Germany
Birgit Pfitzmann, IBM Zurich Research Lab, Switzerland
Bart Preneel, KU Leuven, Belgium
Kai Rannenberg, Goethe University Frankfurt, Germany
Andreas Schaad, SAP Labs, France
Morton Swimmer, IBM Zurich Research Lab, Switzerland
Sheng Zhong, Stevens Institute of Technology, USA

Additional Referees

Srikanth Akkiraju, University of Twente, The Netherlands
Richard Brinkman, University of Twente, The Netherlands
Ileana Buhan, University of Twente, The Netherlands
Lothar Fritsch, Johann Wolfgang Goethe University, Germany
Ludwig Fuchs, University of Regensburg, Germany
Bijit Hore, University of California at Irvine, USA
Ravi Chandra Jammalamadaka, University of California at Irvine, USA
Heiko Rossnagel, Johann Wolfgang Goethe University, Germany
Falk Wagner, Johann Wolfgang Goethe University, Germany
Lingyu Wang, George Mason University, USA
Chao Yao, George Mason University, USA
Xingbo Yu, University of California at Irvine, USA

Lecture Notes in Computer Science

For information about Vols. 1–4062

please contact your bookseller or Springer

Vol. 4185: R. Mizoguchi, Z. Shi, F. Giunchiglia (Eds.), The Semantic Web – ASWC 2006. XX, 778 pages. 2006.

Vol. 4184: M. Bravetti, M. Nuñes, G. Zavattaro (Eds.), Web Services and Formal Methods. X, 289 pages. 2006.

Vol. 4180: M. Kohlhase, OMDoc – An Open Markup Format for Mathematical Documents [version 1.2]. XIX, 428 pages. 2006. (Sublibrary LNAI).

Vol. 4176: S.K. Katsikas, J. Lopez, M. Backes, S. Gritzalis, B. Preneel (Eds.), Information Security. XIV, 548 pages. 2006.

Vol. 4168: Y. Azar, T. Erlebach (Eds.), Algorithms – ESA 2006. XVIII, 843 pages. 2006.

Vol. 4165: W. Jonker, M. Petković (Eds.), Secure, Data Management. X, 183 pages. 2006.

Vol. 4163: H. Bersini, J. Carneiro (Eds.), Artificial Immune Systems. XII, 460 pages. 2006.

Vol. 4162: R. Královič, P. Urzyczyn (Eds.), Mathematical Foundations of Computer Science 2006. XV, 814 pages. 2006.

Vol. 4159: J. Ma, H. Jin, L.T. Yang, J.J.-P. Tsai (Eds.), Ubiquitous Intelligence and Computing. XXII, 1190 pages. 2006.

Vol. 4158: L.T. Yang, H. Jin, J. Ma, T. Ungerer (Eds.), Autonomic and Trusted Computing. XIV, 613 pages. 2006.

Vol. 4156: S. Amer-Yahia, Z. Bellahsène, E. Hunt, R. Unland, J.X. Yu (Eds.), Database and XML Technologies. IX, 123 pages. 2006.

Vol. 4155: O. Stock, M. Schaerf (Eds.), Reasoning, Action and Interaction in AI Theories and Systems. XVIII, 343 pages. 2006. (Sublibrary LNAI).

Vol. 4153: N. Zheng, X. Jiang, X. Lan (Eds.), Advances in Machine Vision, Image Processing, and Pattern Analysis. XIII, 506 pages. 2006.

Vol. 4152: Y. Manolopoulos, J. Pokorný, T. Sellis (Eds.), Advances in Databases and Information Systems. XV, 448 pages. 2006.

Vol. 4151: A. Iglesias, N. Takayama (Eds.), Mathematical Software - ICMS 2006. XVII, 452 pages. 2006.

Vol. 4150: M. Dorigo, L.M. Gambardella, M. Birattari, A. Martinoli, R. Poli, T. Stützle (Eds.), Ant Colony Optimization and Swarm Intelligence. XVI, 526 pages. 2006.

Vol. 4149: M. Klusch, M. Rovatsos, T.R. Payne (Eds.), Cooperative Information Agents X. XII, 477 pages. 2006. (Sublibrary LNAI).

Vol. 4146: J.C. Rajapakse, L. Wong, R. Acharya (Eds.), Pattern Recognition in Bioinformatics. XIV, 186 pages. 2006. (Sublibrary LNBI).

Vol. 4144: T. Ball, R.B. Jones (Eds.), Computer Aided Verification. XV, 564 pages. 2006.

Vol. 4139: T. Salakoski, F. Ginter, S. Pyysalo, T. Pahikkala, Advances in Natural Language Processing. XVI, 771 pages. 2006. (Sublibrary LNAI).

Vol. 4138: X. Cheng, W. Li, T. Znati (Eds.), Wireless Algorithms, Systems, and Applications. XVI, 709 pages. 2006.

Vol. 4137: C. Baier, H. Hermanns (Eds.), CONCUR 2006 – Concurrency Theory. XIII, 525 pages. 2006.

Vol. 4136: R.A. Schmidt (Ed.), Relations and Kleene Algebra in Computer Science. XI, 433 pages. 2006.

Vol. 4135: C.S. Calude, M.J. Dinneen, G. Păun, G. Rozenberg, S. Stepney (Eds.), Unconventional Computation. X, 267 pages. 2006.

Vol. 4134: K. Yi (Ed.), Static Analysis. XIII, 443 pages. 2006.

Vol. 4133: J. Gratch, M. Young, R. Aylett, D. Ballin, P. Olivier (Eds.), Intelligent Virtual Agents. XIV, 472 pages. 2006. (Sublibrary LNAI).

Vol. 4130: U. Furbach, N. Shankar (Eds.), Automated Reasoning. XV, 680 pages. 2006. (Sublibrary LNAI).

Vol. 4129: D. McGookin, S. Brewster (Eds.), Haptic and Audio Interaction Design. XII, 167 pages. 2006.

Vol. 4128: W.E. Nagel, W.V. Walter, W. Lehner (Eds.), Euro-Par 2006 Parallel Processing. XXXIII, 1221 pages. 2006.

Vol. 4127: E. Damiani, P. Liu (Eds.), Data and Applications Security XX. X, 319 pages. 2006.

Vol. 4126: P. Barahona, F. Bry, E. Franconi, N. Henze, U. Sattler, Reasoning Web. X, 269 pages. 2006.

Vol. 4124: H. de Meer, J.P. G. Sterbenz (Eds.), Self-Organizing Systems. XIV, 261 pages. 2006.

Vol. 4121: A. Biere, C.P. Gomes (Eds.), Theory and Applications of Satisfiability Testing - SAT 2006. XII, 438 pages. 2006.

Vol. 4119: C. Dony, J.L. Knudsen, A. Romanovsky, A. Tripathi (Eds.), Advanced Topics in Exception Handling Components. X, 302 pages. 2006.

Vol. 4117: C. Dwork (Ed.), Advances in Cryptology - CRYPTO 2006. XIII, 621 pages. 2006.

Vol. 4116: R. De Prisco, M. Yung (Eds.), Security and Cryptography for Networks. XI, 366 pages. 2006.

Vol. 4115: D.-S. Huang, K. Li, G.W. Irwin (Eds.), Computational Intelligence and Bioinformatics, Part III. XXI, 803 pages. 2006. (Sublibrary LNBI).

Vol. 4114: D.-S. Huang, K. Li, G.W. Irwin (Eds.), Computational Intelligence, Part II. XXVII, 1337 pages. 2006. (Sublibrary LNAI).

Vol. 4113: D.-S. Huang, K. Li, G.W. Irwin (Eds.), Intelligent Computing, Part I. XXVII, 1331 pages. 2006.

Vol. 4112: D.Z. Chen, D. T. Lee (Eds.), Computing and Combinatorics. XIV, 528 pages. 2006.

Vol. 4111: F.S. de Boer, M.M. Bonsangue, S. Graf, W.-P. de Roever (Eds.), Formal Methods for Components and Objects. VIII, 447 pages. 2006.

Vol. 4110: J. Díaz, K. Jansen, J.D.P. Rolim, U. Zwick (Eds.), Approximation, Randomization, and Combinatorial Optimization. XII, 522 pages. 2006.

Vol. 4109: D.-Y. Yeung, J.T. Kwok, A. Fred, F. Roli, D. de Ridder (Eds.), Structural, Syntactic, and Statistical Pattern Recognition. XXI, 939 pages. 2006.

Vol. 4108: J.M. Borwein, W.M. Farmer (Eds.), Mathematical Knowledge Management. VIII, 295 pages. 2006. (Sublibrary LNAI).

Vol. 4106: T.R. Roth-Berghofer, M.H. Göker, H. A. Güvenir (Eds.), Advances in Case-Based Reasoning. XIV, 566 pages. 2006. (Sublibrary LNAI).

Vol. 4104: T. Kunz, S.S. Ravi (Eds.), Ad-Hoc, Mobile, and Wireless Networks. XII, 474 pages. 2006.

Vol. 4099: Q. Yang, G. Webb (Eds.), PRICAI 2006: Trends in Artificial Intelligence. XXVIII, 1263 pages. 2006. (Sublibrary LNAI).

Vol. 4098: F. Pfenning (Ed.), Term Rewriting and Applications. XIII, 415 pages. 2006.

Vol. 4097: X. Zhou, O. Sokolsky, L. Yan, E.-S. Jung, Z. Shao, Y. Mu, D.C. Lee, D. Kim, Y.-S. Jeong, C.-Z. Xu (Eds.), Emerging Directions in Embedded and Ubiquitous Computing. XXVII, 1034 pages. 2006.

Vol. 4096: E. Sha, S.-K. Han, C.-Z. Xu, M.H. Kim, L.T. Yang, B. Xiao (Eds.), Embedded and Ubiquitous Computing. XXIV, 1170 pages. 2006.

Vol. 4095: S. Nolfi, G. Baldassare, R. Calabretta, D. Marocco, D. Parisi, J.C. T. Hallam, O. Miglino, J.-A. Meyer (Eds.), From Animals to Animats 9. XV, 869 pages. 2006. (Sublibrary LNAI).

Vol. 4094: O. H. Ibarra, H.-C. Yen (Eds.), Implementation and Application of Automata. XIII, 291 pages. 2006.

Vol. 4093: X. Li, O.R. Zaïane, Z. Li (Eds.), Advanced Data Mining and Applications. XXI, 1110 pages. 2006. (Sublibrary LNAI).

Vol. 4092: J. Lang, F. Lin, J. Wang (Eds.), Knowledge Science, Engineering and Management. XV, 664 pages. 2006. (Sublibrary LNAI).

Vol. 4091: G.-Z. Yang, T. Jiang, D. Shen, L. Gu, J. Yang (Eds.), Medical Imaging and Augmented Reality. XIII, 399 pages. 2006.

Vol. 4090: S. Spaccapietra, K. Aberer, P. Cudré-Mauroux (Eds.), Journal on Data Semantics VI. XI, 211 pages. 2006.

Vol. 4089: W. Löwe, M. Südholt (Eds.), Software Composition. X, 339 pages. 2006.

Vol. 4088: Z.-Z. Shi, R. Sadananda (Eds.), Agent Computing and Multi-Agent Systems. XVII, 827 pages. 2006. (Sublibrary LNAI).

Vol. 4087: F. Schwenker, S. Marinai (Eds.), Artificial Neural Networks in Pattern Recognition. IX, 299 pages. 2006. (Sublibrary LNAI).

Vol. 4085: J. Misra, T. Nipkow, E. Sekerinski (Eds.), FM 2006: Formal Methods. XV, 620 pages. 2006.

Vol. 4084: M.A. Wimmer, H.J. Scholl, Å. Grönlund, K.V. Andersen (Eds.), Electronic Government. XV, 353 pages. 2006.

Vol. 4083: S. Fischer-Hübner, S. Furnell, C. Lambrinoudakis (Eds.), Trust and Privacy in Digital Business. XIII, 243 pages. 2006.

Vol. 4082: K. Bauknecht, B. Pröll, H. Werthner (Eds.), E-Commerce and Web Technologies. XIII, 243 pages. 2006.

Vol. 4081: A. M. Tjoa, J. Trujillo (Eds.), Data Warehousing and Knowledge Discovery. XVII, 578 pages. 2006.

Vol. 4080: S. Bressan, J. Küng, R. Wagner (Eds.), Database and Expert Systems Applications. XXI, 959 pages. 2006.

Vol. 4079: S. Etalle, M. Truszczyński (Eds.), Logic Programming. XIV, 474 pages. 2006.

Vol. 4077: M.-S. Kim, K. Shimada (Eds.), Geometric Modeling and Processing - GMP 2006. XVI, 696 pages. 2006.

Vol. 4076: F. Hess, S. Pauli, M. Pohst (Eds.), Algorithmic Number Theory. X, 599 pages. 2006.

Vol. 4075: U. Leser, F. Naumann, B. Eckman (Eds.), Data Integration in the Life Sciences. XI, 298 pages. 2006. (Sublibrary LNBI).

Vol. 4074: M. Burmester, A. Yasinsac (Eds.), Secure Mobile Ad-hoc Networks and Sensors. X, 193 pages. 2006.

Vol. 4073: A. Butz, B. Fisher, A. Krüger, P. Olivier (Eds.), Smart Graphics. XI, 263 pages. 2006.

Vol. 4072: M. Harders, G. Székely (Eds.), Biomedical Simulation. XI, 216 pages. 2006.

Vol. 4071: H. Sundaram, M. Naphade, J.R. Smith, Y. Rui (Eds.), Image and Video Retrieval. XII, 547 pages. 2006.

Vol. 4070: C. Priami, X. Hu, Y. Pan, T.Y. Lin (Eds.), Transactions on Computational Systems Biology V. IX, 129 pages. 2006. (Sublibrary LNBI).

Vol. 4069: F.J. Perales, R.B. Fisher (Eds.), Articulated Motion and Deformable Objects. XV, 526 pages. 2006.

Vol. 4068: H. Schärfe, P. Hitzler, P. Øhrstrøm (Eds.), Conceptual Structures: Inspiration and Application. XI, 455 pages. 2006. (Sublibrary LNAI).

Vol. 4067: D. Thomas (Ed.), ECOOP 2006 – Object-Oriented Programming. XIV, 527 pages. 2006.

Vol. 4066: A. Rensink, J. Warmer (Eds.), Model Driven Architecture – Foundations and Applications. XII, 392 pages. 2006.

Vol. 4065: P. Perner (Ed.), Advances in Data Mining. XI, 592 pages. 2006. (Sublibrary LNAI).

Vol. 4064: R. Büschkes, P. Laskov (Eds.), Detection of Intrusions and Malware & Vulnerability Assessment. X, 195 pages. 2006.

Vol. 4063: I. Gorton, G.T. Heineman, I. Crnkovic, H.W. Schmidt, J.A. Stafford, C.A. Szyperski, K. Wallnau (Eds.), Component-Based Software Engineering. XI, 394 pages. 2006.

Table of Contents

Privacy Protection

Privacy Preserving Data Management

Access Control

Database Security

Indistinguishability: The Other Aspect of Privacy*

Chao Yao[1,**], Lingyu Wang[2], Sean X. Wang[3], and Sushil Jajodia[1]

[1] Center for Secure Information Systems
George Mason University
{cyao, jajodia}@gmu.edu
[2] CIISE, Concordia University
wang@encs.concordia.ca
[3] Department of Computer Science
The University of Vermont
xywang@cs.uvm.edu

Abstract. Uncertainty and indistinguishability are two independent aspects of privacy. Uncertainty refers to the property that the attacker cannot tell which private value, among a group of values, an individual actually has, and indistinguishability refers to the property that the attacker cannot see the difference among a group of individuals. While uncertainty has been well studied and applied to many scenarios, to date, the only effort in providing indistinguishability has been the well-known notion of k-anonymity. However, k-anonymity only applies to anonymized tables. This paper defines indistinguishability for general situations based on the symmetry among the possible private values associated with individuals. The paper then discusses computational complexities of and provides practical algorithms for checking whether a set of database views provides enough indistinguishability.

1 Introduction

In many data applications, it's necessary to measure privacy disclosure in released data to protect individual privacy while satisfying application requirements. The measurement metrics used in prior work have mainly been based on uncertainty of private property values, i.e., the uncertainty what private value an individual has. These metrics can be classified into two categories: non-probabilistic and probabilistic. The non-probabilistic metrics are based on whether the private value of an individual can be uniquely inferred from the released data [1,20,7,17,5,16] or whether the cardinality of the set of possible private values inferred for an individual is large enough [26,27]. The probabilistic metrics are based on some characteristics of the probability distribution of the possible private values inferred from the released data [3,2,10,9,15,4] (see Section 4 for more details).

* The work was partially supported by the NSF grants IIS-0430402, IIS-0430165, and IIS-0242237.

** Part of work of this author was done while visiting the University of Vermont.

W. Jonker and M. Petkovic (Eds.): SDM 2006, LNCS 4165, pp. 1–17, 2006.

However, uncertainty is only one aspect of privacy and it alone does not provide adequate protection. For example, we may reveal employee John's salary to be in a large interval (say, $100K$ to $300K$ annually). There may be enough uncertainty. However, if we also reveal that the salaries of all other employees are in ranges that are totally different from John's range (say, all are subranges of $50K$ to $100K$), then John's privacy may still be violated. As another example, suppose from the released data we can infer that all patients in a hospital may only have *Cold* or *SARS* except that *John* may have *Cold* or *AIDS*. Even though the uncertainty of *John*'s sickness has the same "magnitude" as that of the other patients, *John* may still feel his privacy is violated, since he is the only one who possibly has *AIDS*.

To adequately protect privacy, we need to consider the other aspect, namely, *indistinguishability*. Indeed, the privacy breach in the above examples can be viewed as due to the fact that from the released data, an individual is different from all other individuals in terms of their possible private values. In other words, the examples violate a privacy requirement, namely, the "protection from being brought to the attention of others" [11]. What we need is to have each individual belong to a group of individuals who are indistinguishable from each other in terms of their possible private values derived from the released data. In this way, an individual is hidden in a crowd that consists of individuals who have similar/same possible private values. For instance, in the above salary example, to protect John's privacy, we may want to make sure that attackers can only derive from the released data that a large group of employees have the same range as John's for their possible salaries.

Uncertainty and indistinguishability are two independent aspects for providing privacy; one does not imply the other. From the above examples, we can see that uncertainty cannot ensure good indistinguishability. Likewise, good indistinguishability cannot ensure enough uncertainty. For instance, if in the released data many employees have the same single possible salary value, then these employees are indistinguishable from each other in terms of their salaries, but there is not enough uncertainty to protect their privacy (all their salaries are the same and revealed!).

Our idea of indistinguishability is inspired by the notion of k-anonymization [24,25,21,14,18] as it can be viewed as a generalization of anonymization. The idea of k-anonymization is to recode, mostly by generalization, publicly available quasi-IDs in a single released table, so that at least k individuals will have the same recoded quasi-IDs. (Quasi-IDs are values on a combination of attributes that can be used to identify individuals through external sources [24,25].) In our view, this is an effort to provide indistinguishability among k individuals, since the recoding makes the individuals indistinguishable from each other. (As noted above, indistinguishability does not guarantee uncertainty. This is also true for k-anonymization, which is illustrated by the improvement reported in [19]. The authors impose an additional requirement on anonymization, namely, by requiring diverse private values among the tuples with the same recoded quasi-ID, in order to achieve, in our view, both indistinguishability and uncertainty.)

While k-anonymity is an interesting notion, it only applies to anonymized tables. In this paper, we define two kinds of indistinguishability, and the corresponding privacy metrics, that can be applied to general situations, including anonymized tables and relational views. We show that k-anonymization is a special case of one kind of indistinguishability under a certain assumption (see Section 2.5).

Both notions of indistinguishability introduced in this paper are based on certain symmetry between individuals and their private values in the released data. More specifically, the first definition requires symmetry for all possible private values while the second definition bases on symmetry referring only to certain subsets of possible private values. With the two kinds of indistinguishability defined, we turn to study the problem of deciding whether a set of database views provides enough indistinguishability. We study the computational complexity as well as practical algorithms. We focus on checking for indistinguishability since checking for uncertainty has been extensively studied [1,7,17,5,26,16,27].

We summarize the contributions of this paper as follows. (1) We identify indistinguishability as a requirement for privacy in addition to uncertainty, provide formal definitions of different kinds of indistinguishability, and study their properties. (2) We analyze the computational complexity and introduce practical checking methods for deciding whether a set of database views provides enough indistinguishability.

The rest of paper is organized as follows. We give formal definitions of indistinguishability and privacy metrics in Section 2. We then focus on checking database views against these privacy metrics in Section 3. In Section 4 we review the related work. Finally, we conclude with a summary in Section 5.

2 Indistinguishability

2.1 Preliminaries

In this paper, we consider releasing data from a single private table Tbl with schema D. The attributes in D are partitioned into two sets, B and P. The set B consists of the public attributes; P consists of the private attributes. For simplicity and without loss of generality, we assume P only has one attribute.

We assume that the projection on B, $\Pi_B(Tbl)$, is publicly known. In the salary example, this means that the list of employees is publicly known. We believe this assumption is realistic in many situations. In other situations where this is not true, we may view our approach as providing a conservative privacy measure.

Given a relation r_B on B, we will use $\mathcal{I}^B$ to denote the set $\{I|\Pi_B(I) = r_B\}$, i.e., the set of the relations on D whose B-projection coincides with r_B. The domain of P is denoted by $Dom(P)$. A tuple of an instance in $\mathcal{I}^B$ is denoted by t or (b, p), where b is in $\Pi_B(Tbl)$ and p is in $Dom(P)$. The set $\mathcal{I}^B$ corresponds to all possible private table instances by only knowing $\Pi_B(Tbl)$.

Furthermore, we assume B is a key in D, which means that each composite value on B appears at most once in the private table. We also assume B is a quasi-ID, and hence, the tuples in Tbl describe associations of the private attribute values with individuals. (Recall that a quasi-ID is a combination of attribute values that can be used to identify an individual.) Such associations are the private information to be protected.

In Figure 1, our running example is shown. The public attributes in B are *Zip*, *Age*, *Race*, *Gender*, and *Charge*. We use $t_1, ..., t_{12}$ to denote the tuples in the table. By the assumption that B is a quasi-ID, $t_i[B]$ identifies a particular individual for each i. In the sequel, we use $t_i[B]$ and the individual identified by $t_i[B]$ interchangeably. The private attribute is *Problem*. Here, *Problem* is drawn from a finite discrete domain. (In general the private attribute also can be drawn from an infinite or a continuous domain; but it should not be difficult to extend our study to infinite discrete or continuous domains).

We assume that the data in *Tbl* are being released with a publicly-known function M. We also use v to denote the result of $M()$ on the private table, i.e., $v = M(Tbl)$. Examples of function $M()$ include an anonymization procedure, and a set of queries (views) on a single table on D.

	Zip	Age	Race	Gender	Charge	Problem
t_1	22030	39	White	Male	1K	Cold
t_2	22030	50	White	Male	12K	AIDS
t_3	22030	38	White	Male	5K	Obesity
t_4	22030	53	Black	Male	5K	AIDS
t_5	22031	28	Black	Female	8K	Chest Pain
t_6	22031	37	White	Female	10K	Hypertension
t_7	22031	49	Black	Female	1K	Obesity
t_8	22031	52	White	Male	8K	Cold
t_9	22032	30	Asian	Male	10K	Hypertension
t_{10}	22032	40	Asian	Male	9K	Chest Pain
t_{11}	22033	30	White	Male	10K	Hypertension
t_{12}	22033	40	White	Male	9K	Chest Pain

Fig. 1. A patient table (Tbl)

	Zip	Problem
t_9	22032	Hypertension
t_{10}	22032	Chest Pain
t_{11}	22033	Hypertension
t_{12}	22033	Chest Pain

Fig. 2. A released view $\Pi_{Zip,Problem}(Tbl)$ $\sigma_{Zip='22032' or '22033'}(Tbl)$ provides 2-SIND

2.2 Symmetric Indistinguishability

As $v = M(Tbl)$ is released, we denote by $\mathcal{I}^v$ the subset of possible instances in $\mathcal{I}^B$ that yield v. We introduce the definition of indistinguishability based on $\mathcal{I}^v$.

Definition 1. (**Symmetric Indistinguishability**) *Given a released data v and two tuples b_i and b_j in $\Pi_B(Tbl)$, we say b_i and b_j are* symmetrically Indistinguishable *w.r.t. v if the following condition is satisfied: for each instance I in $\mathcal{I}^v$ containing (b_i, p_i) and (b_j, p_j), there exists another instance I' in $\mathcal{I}^v$ such that $I' = (I - \{(b_i, p_i), (b_j, p_j)\}) \cup \{(b_i, p_j), (b_j, p_i)\}$.*

We abbreviate Symmetric Indistinguishability as *SIND*. This definition requires that for each possible instance in $\mathcal{I}^v$, if two symmetrically indistinguishable B tuples swap their private values while keeping other tuples unchanged, the resulting new instance can still yield v. In the sequel, we say *two B tuples $t_1[B]$ and $t_2[B]$ can swap their private values in an instance*, or simply $t_1[B]$ *swaps with* $t_2[B]$, if the resulting instance can still yield v.

Note that such a swap is required for all the instances yielding v, hence this definition is in terms of v, not the current table Tbl (although we used the projection $\Pi_B(Tbl)$ in the definition, this projection is not Tbl itself and is assumed publicly known). In other words, to be SIND is to be able to swap their private values in all the possible instances, including Tbl.

For example, consider the released view v in Figure 2 on the table in Figure 1. The two B tuples $t_9[B]$ and $t_{10}[B]$ are SIND, because they can swap their *Problem* values in any instance that yields v while still yielding the same v. Similarly, the two B tuples $t_{11}[B]$ and $t_{12}[B]$ are also SIND. However, $t_9[B]$ and $t_{11}[B]$ are not SIND, even though they have the same *Problem* value *Hypertension* in the current private table. To show this, consider an instance obtained by swapping the *Problem* values of t_9 and t_{10} in Tbl (while other tuples remain unchanged). So now t_9 has *ChestPain* while t_{10} has *Hypertension*. Denote the new instance Tbl'. Clearly, Tbl' also yields the view v. However, in Tbl', if we swap the *Problem* values of t_9 (i.e., *ChestPain*) with that of t_{11} (i.e., *Hypertension*), then both t_9 and t_{10} will have *Hypertension*. Therefore, the new instance obtained from Tbl' does not yield v, and hence t_9 and t_{11} are not SIND.

The definition of SIND requires a complete symmetry between two B tuples in terms of their private values. The sets of possible private values of the SIND tuples are the same, because in each possible instance two SIND B tuples can swap their private values without changing the views. Furthermore, the definition based on swapping makes SIND between two B tuples independent on other B tuples. That is, even if attackers can guess the private values of all other B tuples, they still cannot distinguish between these two B tuples because the two B tuples still can swap their private values without affecting the views.

We can also use a probability model to illustrate the indistinguishability by SIND. If we assume each B tuple has the same and independent *a priori* distribution over its private values, then we can easily prove that the two B tuples have the same *a posteriori* distribution over their private values after data released, due to complete symmetry in terms of their private values.

The binary relation SIND is *reflexive*, *symmetric* and *transitive*. That is, SIND is an *equivalence* relation. It is easy to see that it is reflexive and symmetric. We prove the transitivity as follows. If a B tuple b_1 can swap with another B tuple b_2 and b_2 can swap with b_3, then b_1 can swap with b_3 by the following steps: b_1 swaps with b_2; b_2 swaps with b_3; b_2 swaps with b_1; by the definition of SIND, the final instance still yields v.

Thus, all the B tuples that are indistinguishable from each other form a partition of the B tuples. Each set in the partition, which we call a *SIND set*, is the "crowd" that provides individual privacy. The sizes of these crowds reflect how much protection they give to the individuals in the crowd. So we have the following metric.

Definition 2. (k-**SIND**) *Given a released data v, if each SIND set has a cardinality of at least k, we then say v provides* k-SIND.

2.3 Relationship with k-Anonymity

In this subsection, we discuss the relationship between k-SIND and k-anonymity. In the k-anonymity literature (e.g., [24,25,21,14,18]), the released data is an anonymized table. Anonymization is a function from quasi-IDs to recoded quasi-IDs, and the anonymization process (the function M in Section 2.1) is to replace quasi-IDs with recoded quasi-IDs. We assume that the anonymization algorithm and the input quasi-IDs are known. In fact, we make a stronger assumption, called "mapping assumption", which says that (1) each quasi-ID maps to one recoded quasi-ID and (2) given a recoded quasi-ID, attackers know which set of quasi-IDs map to it.

As an example, there is a table and an anonymized table as the following, respectively. The tuples on $(Zip, Race)$ are quasi-IDs. Under the mapping assumption, attackers know which quasi-ID maps to which recoded quasi-ID. For instance, (22031, $White$) maps to (2203$*$, $*$) but not (220$**$, $White$). (In contrast, without the mapping assumption, only from the anonymized table, (22031, $White$) may map to either (2203$*$, $*$) or (220$**$, $White$).)

Zip	**Race**	**Problem**
22021	White	Cold
22031	White	Obesity
22032	White	AIDS
22033	Black	Headache

Zip	**Race**	**Problem**
220$**$	White	Cold
220$**$	White	Obesity
2203$*$	$*$	AIDS
2203$*$	$*$	Headache

Under the above assumption, we have the following conclusion about the relationship between k-SIND and k-anonymity. Here the attributes of quasi-IDs are assumed to be exactly the public attributes B.

Proposition 1. *Under the mapping assumption, if an anonymized table v provides k-anonymity, where $k \geq 2$, then v provides k-SIND.*

Intuitively, if v provides k-anonymity, then at least k quasi-IDs map to each recoded quasi-ID in v. In any instance yielding v, suppose two quasi-IDs b_1 and b_2 map to the same recoded quasi-ID. Then swapping the private values of b_1 and b_2 in the original table gives an instance yielding the same v. Therefore, v provides k-SIND.

By definition, k-anonymity is applicable only to a single anonymized table, but not to other kinds of released data such as multiple database views.

2.4 Restricted Symmetric Indistinguishability

Since SIND requires symmetry in terms of all possible private values, it is a rather strict metric. We define another metric based on the symmetry in terms of not all possible private values but only a subset that includes the actual private values in the current private table. If B tuples are symmetric in terms of this subset of private values, even though they are not symmetric in terms of other values, we may still take them as indistinguishable. The intuition here is that we intend to provide more protection on the actual private values.