Willem Jonker
Milan Petković (Eds.)

LNCS 3674

Secure Data Management

Second VLDB Workshop, SDM 2005
Trondheim, Norway, September 2005
Proceedings

Springer

Willem Jonker Milan Petković (Eds.)

Secure Data Management

Second VLDB Workshop, SDM 2005
Trondheim, Norway, September 2-3, 2005
Proceedings

Springer

Volume Editors

Willem Jonker
Milan Petković
Philips Research Eindhoven
Information and System Security
Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands
E-mail: {Willem.Jonker,Milan.Petkovic}@philips.com

Library of Congress Control Number: 200593521

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

ISSN 0302-9743
ISBN-10 3-540-28798-1 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-28798-8 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, re-use of illustrations, recitation, broadcasting, reproduction on microfilms or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media

springeronline.com

© Springer-Verlag Berlin Heidelberg 2005
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11552338 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 3674

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

Editorial Board

David Hutchison
Lancaster University, UK

Takeo Kanade
Carnegie Mellon University, Pittsburgh, PA, USA

Josef Kittler
University of Surrey, Guildford, UK

Jon M. Kleinberg
Cornell University, Ithaca, NY, USA

Friedemann Mattern
ETH Zurich, Switzerland

John C. Mitchell
Stanford University, CA, USA

Moni Naor
Weizmann Institute of Science, Rehovot, Israel

Oscar Nierstrasz
University of Bern, Switzerland

C. Pandu Rangan
Indian Institute of Technology, Madras, India

Bernhard Steffen
University of Dortmund, Germany

Madhu Sudan
Massachusetts Institute of Technology, MA, USA

Demetri Terzopoulos
New York University, NY, USA

Doug Tygar
University of California, Berkeley, CA, USA

Moshe Y. Vardi
Rice University, Houston, TX, USA

Gerhard Weikum
Max-Planck Institute of Computer Science, Saarbruecken, Germany

Preface

Although cryptography and security techniques have been around for quite some time, emerging technologies such as ubiquitous computing and ambient intelligence that exploit increasingly interconnected networks, mobility and personalization put new requirements on security with respect to data management. As data is accessible anytime anywhere, according to these new concepts, it becomes much easier to get unauthorized data access. Furthermore, 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 dynamic, context-aware, role-based), database security (e.g., efficient database encryption schemes, search over encrypted data, etc.), and privacy-preserving data mining to controlled sharing of data.

In addition to the aforementioned subject, this year we also called for papers devoted to secure data management in healthcare as a domain where data security and privacy issues are traditionally important. The call for papers attracted 38 papers both from universities and industry. The Program Committee selected 16 research papers for presentation at the workshop. These papers are also collected in this volume which we hope will serve you as a useful research and reference material.

The volume is divided roughly into four major sections. The first section focuses on encrypted databases addressing the topics of key and metadata management, as well as searching in the encrypted domain. The second section changes slightly the focal point to access control, which remains an important area of interest. The papers in this section deal with this topic from a different point of view and in a different context: two papers in the medical domain, one in the area of the Semantic Web and one in XML databases. The third section focuses on disclosure detection, control and prevention, again in a database environment. The last paper in this section addresses in particular the topics of inference control and anonymization in medical databases. Finally, the fourth section addresses privacy and security technologies which are required in a modern world to support concepts like ubiquitous computing or location-based services.

July 2005 Willem Jonker and Milan Petković

此为试读,需要完整PDF请访问: www.ertongbook.com

Organization

Workshop Organizers

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

Program Committee

Peter Apers, Twente University, The Netherlands
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
Eduardo Fernández-Medina, University of Castilla-La Mancha, Spain
Simone Fischer-Hübner, Karlstad University, Sweden
Tyrone Grandison, IBM Almaden Research Center, USA
Ehud Gudes, Ben-Gurion University, Israel
Marit Hansen, Independent Centre for Privacy Protection, Germany
Pieter Hartel, Twente University, The Netherlands
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
Stig Frode Mjlsnes, 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, KULeuven, Belgium
Jean-Jacques Quisquater, Universit Catholique de Louvain, Belgium
Kai Rannenberg, Goethe University, Frankfurt, Germany
Morton Swimmer, IBM Zurich Research Lab, Switzerland
Sheng Zhong, Stevens Institute of Technology, USA
Josip Zorić, Norwegian Telecom, Norway

Additional Referees

Maarten Fokkinga, University of Twente, The Netherlands
Ling Feng, University of Twente, The Netherlands

Carlos Gutiérrez, STL, Spain
Jan Muntermann, Frankfurt University, Germany
Christiane Schweitzer, Karlstad University, Sweden
Maurice van Keulen, University of Twente, The Netherlands
Anna Zych, University of Twente, The Netherlands
Amit Jain, University of South Carolina, USA
Marcin Czenko, University of Twente, The Netherlands
Ha Tran, University of Twente, The Netherlands
Ari Saptawijaya, University of Twente, The Netherlands
Huiping Guo, George Mason University, USA
Erez Shmueli, Ben-Gurion University, Israel
Ronen Waisenberg, Ben-Gurion University, Israel
Claudine Conrado, Philips Research, The Netherlands
Djoerd Hiemstra, University of Twente, The Netherlands

Lecture Notes in Computer Science

For information about Vols. 1–3562

please contact your bookseller or Springer

Vol. 3687: S. Singh, M. Singh, C. Apte, P. Perner (Eds.), Pattern Recognition and Image Analysis, Part II. XXV, 809 pages. 2005.

Vol. 3686: S. Singh, M. Singh, C. Apte, P. Perner (Eds.), Pattern Recognition and Data Mining, Part I. XXVI, 689 pages. 2005.

Vol. 3674: W. Jonker, M. Petković (Eds.), Secure Data Management. X, 241 pages. 2005.

Vol. 3672: C. Hankin, I. Siveroni (Eds.), Static Analysis. X, 369 pages. 2005.

Vol. 3671: S. Bressan, S. Ceri, E. Hunt, Z.G. Ives, Z. Bellahsène, M. Rys, R. Unland (Eds.), Database and XML Technologies. X, 239 pages. 2005.

Vol. 3664: C. Türker, M. Agosti, H.-J. Schek (Eds.), Peer-to-Peer, Grid, and Service-Orientation in Digital Library Architectures. X, 261 pages. 2005.

Vol. 3663: W. Kropatsch, R. Sablatnig, A. Hanbury (Eds.), Pattern Recognition. XIV, 512 pages. 2005.

Vol. 3662: C. Baral, G. Greco, N. Leone, G. Terracina (Eds.), Logic Programming and Nonmonotonic Reasoning. XIII, 454 pages. 2005. (Subseries LNAI).

Vol. 3660: M. Beigl, S. Intille, J. Rekimoto, H. Tokuda (Eds.), UbiComp 2005: Ubiquitous Computing. XVII, 394 pages. 2005.

Vol. 3659: J.R. Rao, B. Sunar (Eds.), Cryptographic Hardware and Embedded Systems – CHES 2005. XIV, 458 pages. 2005.

Vol. 3654: S. Jajodia, D. Wijesekera (Eds.), Data and Applications Security XIX. X, 353 pages. 2005.

Vol. 3653: M. Abadi, L.d. Alfaro (Eds.), CONCUR 2005 – Concurrency Theory. XIV, 578 pages. 2005.

Vol. 3649: W.M.P. van der Aalst, B. Benatallah, F. Casati, F. Curbera (Eds.), Business Process Management. XII, 472 pages. 2005.

Vol. 3648: J.C. Cunha, P.D. Medeiros (Eds.), Euro-Par 2005 Parallel Processing. XXXVI, 1299 pages. 2005.

Vol. 3645: D.-S. Huang, X.-P. Zhang, G.-B. Huang (Eds.), Advances in Intelligent Computing, Part II. XIII, 1010 pages. 2005.

Vol. 3644: D.-S. Huang, X.-P. Zhang, G.-B. Huang (Eds.), Advances in Intelligent Computing, Part I. XXVII, 1101 pages. 2005.

Vol. 3642: D. Ślezak, J. Yao, J.F. Peters, W. Ziarko, X. Hu (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Part II. XXIV, 738 pages. 2005. (Subseries LNAI).

Vol. 3641: D. Ślezak, G. Wang, M.S. Szczuka, I. Düntsch, Y. Yao (Eds.), Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing, Part I. XXIV, 742 pages. 2005. (Subseries LNAI).

Vol. 3639: P. Godefroid (Ed.), Model Checking Software. XI, 289 pages. 2005.

Vol. 3638: A. Butz, B. Fisher, A. Krüger, P. Olivier (Eds.), Smart Graphics. XI, 269 pages. 2005.

Vol. 3637: J. M. Moreno, J. Madrenas, J. Cosp (Eds.), Evolvable Systems: From Biology to Hardware. XI, 227 pages. 2005.

Vol. 3636: M.J. Blesa, C. Blum, A. Roli, M. Sampels (Eds.), Hybrid Metaheuristics. XII, 155 pages. 2005.

Vol. 3634: L. Ong (Ed.), Computer Science Logic. XI, 567 pages. 2005.

Vol. 3633: C. Bauzer Medeiros, M. Egenhofer, E. Bertino (Eds.), Advances in Spatial and Temporal Databases. XIII, 433 pages. 2005.

Vol. 3632: R. Nieuwenhuis (Ed.), Automated Deduction – CADE-20. XIII, 459 pages. 2005. (Subseries LNAI).

Vol. 3629: J.L. Fiadeiro, N. Harman, M. Roggenbach, J. Rutten (Eds.), Algebra and Coalgebra in Computer Science. XI, 457 pages. 2005.

Vol. 3628: T. Gschwind, U. Aßmann, O. Nierstrasz (Eds.), Software Composition. X, 199 pages. 2005.

Vol. 3627: C. Jacob, M.L. Pilat, P.J. Bentley, J. Timmis (Eds.), Artificial Immune Systems. XII, 500 pages. 2005.

Vol. 3626: B. Ganter, G. Stumme, R. Wille (Eds.), Formal Concept Analysis. X, 349 pages. 2005. (Subseries LNAI).

Vol. 3625: S. Kramer, B. Pfahringer (Eds.), Inductive Logic Programming. XIII, 427 pages. 2005. (Subseries LNAI).

Vol. 3624: C. Chekuri, K. Jansen, J.D.P. Rolim, L. Trevisan (Eds.), Approximation, Randomization and Combinatorial Optimization. XI, 495 pages. 2005.

Vol. 3623: M. Liśkiewicz, R. Reischuk (Eds.), Fundamentals of Computation Theory. XV, 576 pages. 2005.

Vol. 3621: V. Shoup (Ed.), Advances in Cryptology – CRYPTO 2005. XI, 568 pages. 2005.

Vol. 3620: H. Muñoz-Avila, F. Ricci (Eds.), Case-Based Reasoning Research and Development. XV, 654 pages. 2005. (Subseries LNAI).

Vol. 3619: X. Lu, W. Zhao (Eds.), Networking and Mobile Computing. XXIV, 1299 pages. 2005.

Vol. 3618: J. Jedrzejowicz, A. Szepietowski (Eds.), Mathematical Foundations of Computer Science 2005. XVI, 814 pages. 2005.

Vol. 3617: F. Roli, S. Vitulano (Eds.), Image Analysis and Processing – ICIAP 2005. XXIV, 1219 pages. 2005.

Vol. 3615: B. Ludäscher, L. Raschid (Eds.), Data Integration in the Life Sciences. XII, 344 pages. 2005. (Subseries LNBI).

Vol. 3614: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part II. XLI, 1314 pages. 2005. (Subseries LNAI).

Vol. 3613: L. Wang, Y. Jin (Eds.), Fuzzy Systems and Knowledge Discovery, Part I. XLI, 1334 pages. 2005. (Subseries LNAI).

Vol. 3612: L. Wang, K. Chen, Y. S. Ong (Eds.), Advances in Natural Computation, Part III. LXI, 1326 pages. 2005.

Vol. 3611: L. Wang, K. Chen, Y. S. Ong (Eds.), Advances in Natural Computation, Part II. LXI, 1292 pages. 2005.

Vol. 3610: L. Wang, K. Chen, Y. S. Ong (Eds.), Advances in Natural Computation, Part I. LXI, 1302 pages. 2005.

Vol. 3608: F. Dehne, A. López-Ortiz, J.-R. Sack (Eds.), Algorithms and Data Structures. XIV, 446 pages. 2005.

Vol. 3607: J.-D. Zucker, L. Saitta (Eds.), Abstraction, Reformulation and Approximation. XII, 376 pages. 2005. (Subseries LNAI).

Vol. 3606: V. Malyshkin (Ed.), Parallel Computing Technologies. XII, 470 pages. 2005.

Vol. 3604: R. Martin, H. Bez, M. Sabin (Eds.), Mathematics of Surfaces XI. IX, 473 pages. 2005.

Vol. 3603: J. Hurd, T. Melham (Eds.), Theorem Proving in Higher Order Logics. IX, 409 pages. 2005.

Vol. 3602: R. Eigenmann, Z. Li, S.P. Midkiff (Eds.), Languages and Compilers for High Performance Computing. IX, 486 pages. 2005.

Vol. 3599: U. Aßmann, M. Aksit, A. Rensink (Eds.), Model Driven Architecture. X, 235 pages. 2005.

Vol. 3598: H. Murakami, H. Nakashima, H. Tokuda, M. Yasumura, Ubiquitous Computing Systems. XIII, 275 pages. 2005.

Vol. 3597: S. Shimojo, S. Ichii, T.W. Ling, K.-H. Song (Eds.), Web and Communication Technologies and Internet-Related Social Issues - HSI 2005. XIX, 368 pages. 2005.

Vol. 3596: F. Dau, M.-L. Mugnier, G. Stumme (Eds.), Conceptual Structures: Common Semantics for Sharing Knowledge. XI, 467 pages. 2005. (Subseries LNAI).

Vol. 3595: L. Wang (Ed.), Computing and Combinatorics. XVI, 995 pages. 2005.

Vol. 3594: J.C. Setubal, S. Verjovski-Almeida (Eds.), Advances in Bioinformatics and Computational Biology. XIV, 258 pages. 2005. (Subseries LNBI).

Vol. 3593: V. Mařík, R. W. Brennan, M. Pěchouček (Eds.), Holonic and Multi-Agent Systems for Manufacturing. XI, 269 pages. 2005. (Subseries LNAI).

Vol. 3592: S. Katsikas, J. Lopez, G. Pernul (Eds.), Trust, Privacy and Security in Digital Business. XII, 332 pages. 2005.

Vol. 3591: M.A. Wimmer, R. Traunmüller, Å. Grönlund, K.V. Andersen (Eds.), Electronic Government. XIII, 317 pages. 2005.

Vol. 3590: K. Bauknecht, B. Pröll, H. Werthner (Eds.), E-Commerce and Web Technologies. XIV, 380 pages. 2005.

Vol. 3589: A M. Tjoa, J. Trujillo (Eds.), Data Warehousing and Knowledge Discovery. XVI, 538 pages. 2005.

Vol. 3588: K.V. Andersen, J. Debenham, R. Wagner (Eds.), Database and Expert Systems Applications. XX, 955 pages. 2005.

Vol. 3587: P. Perner, A. Imiya (Eds.), Machine Learning and Data Mining in Pattern Recognition. XVII, 695 pages. 2005. (Subseries LNAI).

Vol. 3586: A.P. Black (Ed.), ECOOP 2005 - Object-Oriented Programming. XVII, 631 pages. 2005.

Vol. 3584: X. Li, S. Wang, Z.Y. Dong (Eds.), Advanced Data Mining and Applications. XIX, 835 pages. 2005. (Subseries LNAI).

Vol. 3583: R.W. H. Lau, Q. Li, R. Cheung, W. Liu (Eds.), Advances in Web-Based Learning – ICWL 2005. XIV, 420 pages. 2005.

Vol. 3582: J. Fitzgerald, I.J. Hayes, A. Tarlecki (Eds.), FM 2005: Formal Methods. XIV, 558 pages. 2005.

Vol. 3581: S. Miksch, J. Hunter, E. Keravnou (Eds.), Artificial Intelligence in Medicine. XVII, 547 pages. 2005. (Subseries LNAI).

Vol. 3580: L. Caires, G.F. Italiano, L. Monteiro, C. Palamidessi, M. Yung (Eds.), Automata, Languages and Programming. XXV, 1477 pages. 2005.

Vol. 3579: D. Lowe, M. Gaedke (Eds.), Web Engineering. XXII, 633 pages. 2005.

Vol. 3578: M. Gallagher, J. Hogan, F. Maire (Eds.), Intelligent Data Engineering and Automated Learning - IDEAL 2005. XVI, 599 pages. 2005.

Vol. 3577: R. Falcone, S. Barber, J. Sabater-Mir, M.P. Singh (Eds.), Trusting Agents for Trusting Electronic Societies. VIII, 235 pages. 2005. (Subseries LNAI).

Vol. 3576: K. Etessami, S.K. Rajamani (Eds.), Computer Aided Verification. XV, 564 pages. 2005.

Vol. 3575: S. Wermter, G. Palm, M. Elshaw (Eds.), Biomimetic Neural Learning for Intelligent Robots. IX, 383 pages. 2005. (Subseries LNAI).

Vol. 3574: C. Boyd, J.M. González Nieto (Eds.), Information Security and Privacy. XIII, 586 pages. 2005.

Vol. 3573: S. Etalle (Ed.), Logic Based Program Synthesis and Transformation. VIII, 279 pages. 2005.

Vol. 3572: C. De Felice, A. Restivo (Eds.), Developments in Language Theory. XI, 409 pages. 2005.

Vol. 3571: L. Godo (Ed.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty. XVI, 1028 pages. 2005. (Subseries LNAI).

Vol. 3570: A. S. Patrick, M. Yung (Eds.), Financial Cryptography and Data Security. XII, 376 pages. 2005.

Vol. 3569: F. Bacchus, T. Walsh (Eds.), Theory and Applications of Satisfiability Testing. XII, 492 pages. 2005.

Vol. 3568: W.-K. Leow, M.S. Lew, T.-S. Chua, W.-Y. Ma, L. Chaisorn, E.M. Bakker (Eds.), Image and Video Retrieval. XVII, 672 pages. 2005.

Vol. 3567: M. Jackson, D. Nelson, S. Stirk (Eds.), Database: Enterprise, Skills and Innovation. XII, 185 pages. 2005.

Vol. 3566: J.-P. Banâtre, P. Fradet, J.-L. Giavitto, O. Michel (Eds.), Unconventional Programming Paradigms. XI, 367 pages. 2005.

Vol. 3565: G.E. Christensen, M. Sonka (Eds.), Information Processing in Medical Imaging. XXI, 777 pages. 2005.

Vol. 3564: N. Eisinger, J. Małuszyński (Eds.), Reasoning Web. IX, 319 pages. 2005.

Table of Contents

Encrypted Data Access

Access Control

Information Disclosure Control in Databases

Privacy and Security Support for Distributed Applications

此为试读,需要完整PDF请访问: www.ertongbook.com

Efficient Key Updates in Encrypted Database Systems

Hakan Hacıgümüş[1] and Sharad Mehrotra[2]

[1] IBM Almaden Research Center, USA
hakanh@acm.org
[2] University of California, Irvine, USA
sharad@ics.uci.edu

Abstract. In this paper, we investigate efficient key updates in encrypted database environments. We study the issues in the context of database-as-a-service (DAS) model that allows organizations to outsource their data management infrastructures to a database service provider. In the DAS model, a service provider employs data encryption techniques to ensure the privacy of hosted data. The security of encryption techniques relies on the confidentiality of the encryption keys. The dynamic nature of the encrypted database in the DAS model adds complexity and raises specific requirements on the key management techniques. Key updates are particularly critical because of their potential impact on overall system performance and resources usage. In this paper, we propose specialized techniques and data structures to efficiently implement the key updates along with the other key management functions to improve the systems' concurrency performance in the DAS model.

1 Introduction

The commodity pricing of processors, storage, network bandwidth, and basic software is continuously reducing the relative contribution of these elements to the total lifecycle cost of computing solutions. Operating and integration costs are increasing, in comparison. The research community has responded by working on approaches to automated system administration as in [2]. Increasingly, large companies are consolidating data operations into extremely efficiently administered data centers, sometimes even outsourcing them [4].

The *Database-as-a-Service* (DAS) model [8] is one manifestation of this trend. In the DAS model, the client's database is stored at the service provider. The provider is responsible for provisioning adequate CPU, storage, and networking resources required to run database operations, in addition to the system administration tasks such as backup, recovery, reorganization etc.

A fundamental challenge posed by the DAS model is that of database privacy and security [8]. In the DAS model, the user data resides on the premises of the database service provider. Most companies and individuals view their data as an asset. The theft of intellectual property already costs organizations great amount of money every year [3]. The increasing importance of security in databases is discussed in [6][13][12][1][8][7][5][9][10]. Therefore, first, the owner of the data

W. Jonker and M. Petković (Eds.): SDM 2005, LNCS 3674, pp. 1–15, 2005.
© Springer-Verlag Berlin Heidelberg 2005

needs to be assured that the data is protected against malicious attacks from the outside of the service provider. In addition to this, recent studies indicate that 40% of those attacks are perpetrated by the insiders [3]. Hence, the second and more challenging problem is the privacy of the data when even the service provider itself is not trusted by the owner of the data. Data encryption is proposed as a solution to ensure the privacy of the users' data. The first problem is examined in [8] and the second one is studied in [7], which explores how SQL queries can be executed over encrypted data.

The security of any encryption technique relies on the confidentiality of the encryption keys. Hence, key management plays an essential role in a system, which employs encryption techniques. In this paper, we mainly focus on the key management issues in the context of the database-as-a-service model, where the clients' databases are stored at the service provider site in the encrypted form. We argue that the key management in the hosted databases requires special consideration especially due to the dynamic nature of the database systems.

The update transactions are an essential part of the database systems and applications. Each update transaction requires at least one invocation of the encryption function to encrypt the data in the system.[1] It is known that encryption is a CPU intensive process [8]. Therefore the update transactions may hold locks on the certain set of database records for an extended period of time causing a decline in the system performance. Besides the database update transactions, re-keying is another process, which requires the invocation of the encryption function in the system. As we discuss in Section 3, re-keying is recommended and sometimes required for the systems that employ encryption. Re-keying large amounts of data entails significant encryption costs and interferes with the other transactions thereby causing performance degradation in the system. In this study, we address these issues by proposing a specialized key management architecture in Section 3. Our main focus is the key updates. We propose new lock modes, key update locks, which are leveraged by the database lock manager to efficiently handle the key updates along with the other database update transactions. We present the necessary lock management protocols based on the key management architecture we explain in the paper. We also introduce a system architecture taxonomy in Section 2.3, which is coupled with the key management architecture to enable the performance-conscious encryption key management in dynamic database environments.

2 System Architectures

2.1 Overall DAS Architecture

The system we use in this study is based on the architecture proposed and described in [7]. The basic architecture and the control flow of the system are

[1] The actual number of invocations depends on various factors such as the data unit subject to the encryption, i.e., the granularity of the encryption, specifics of the transaction, e.g., an insert only transaction, a transaction on a number of data objects, etc.

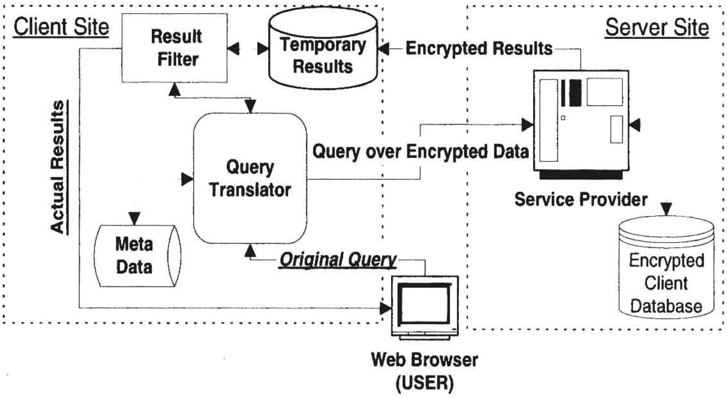

Fig. 1. Database-as-a-Service architecture

shown in Figure 1. It is comprised of three fundamental entities. A *user* poses the query to the client. A *server* is hosted by the service provider that stores the encrypted database. The encrypted database is augmented with additional information (which we call the index) that allows the certain amount of query processing to occur at the server without jeopardizing the data privacy. A *client* stores the data at the server. Client[2] also maintains the *metadata* for translating the user queries to the appropriate representation on the server, and performs post-processing on server query results. From the privacy perspective, the most important feature is, the client's data is always stored in the encrypted form at the server site. The server never sees the unencrypted form of the data, and executes the queries directly over the encrypted data without decrypting it.

2.2 Storing Encrypted Data in the Database

We briefly summarize how the client's data stored at the server in an encrypted fashion in the DAS model.[3]

For each relation $R(A_1, A_2, \ldots, A_n)$, we store, on the server, an encrypted relation: $R^S(RID, KID, \textit{etuple}, P_1^{id}, P_2^{id}, \ldots, P_i^{id})$, where $1 \leq i \leq n$. Here, an *etuple* stores an encrypted string that corresponds to a tuple in a relation R. Each attribute P_i^{id} stores the partition index for the corresponding attribute A_i that will be used for query processing at the server.

For example, consider the relation *emp* given in Table 1 that stores information about employees. The *emp* table is mapped to a corresponding table, shown in Table 2, at the server: $emp^S(RID, KID, etuple,\ eid^{id}, ename^{id},\ salary^{id})$.

The RID represents the *record identifier*, which is a unique number created by the client for each tuple. Here, the RIDs are not the same as unique identi-

[2] Often the client and the user might be the same entity.

[3] We will not repeat all of the details of the storage model here, since it is thoroughly discussed in [7]. Rather, we only provide the necessary notations to explain the constructs we develop in this work.

Table 1. Relation *emp*

eid	ename	salary	addr	did
23	Tom	70K	Maple	40
860	Mary	60K	Main	80
320	John	23K	River	35
200	Sarah	55K	River	10

Table 2. Encrypted representation emp^S of *emp*

RID	KID	*etuple*	eid^{id}	$ename^{id}$	$salary^{id}$
1	45	=*?Ew@R*((¡¡=+,-...	2	19	81
2	78	b*((¡¡(*?Ew@=l,r...	4	31	59
3	65	w@=W*((¡¡(*?E:,j...	7	59	22
4	52	ffTi* @=U(¡?G+,a...	8	49	59

fiers, which are used as references to the records and assigned by the database manager, as it is done in most of the commercial database products. Instead, these RIDs also uniquely identify the records, however, they are created and assigned by the client to facilitate the schemes we present in the study.

The KID represents the *key identifier*, which is also created and assigned by the client. The KID indicates which key is used to encrypt the *etuple* of the corresponding tuple. We elaborate on the use of KIDs in Section 3.5.

The column *etuple* contains the string corresponding to the encrypted tuples in *emp*. For instance, the first tuple is encrypted to "=*?Ew@R*((¡¡=+,-..." that is equal to $\mathcal{E}_k(1, 23, Tom, 70K, Maple, 40)$, where $\mathcal{E}$ is a deterministic encryption algorithm with key k. Any deterministic encryption technique such as AES, DES etc., can be used to encrypt the tuples. The column eid^{id} corresponds to the index on the employee ids.[4]

2.3 Classification of the System Architectures

In this section, we propose different instantiations for the overall system architecture presented above. Our classification of the system architecture alternatives is *client-oriented*. In other words, we identify the architecture model based on how the clients interact with the service provider. We classify the system architecture models under three categories; *standalone clients*, *group of clients*, and *client networks*. Each model has implications on the characteristics of the system including the control flow, index management, key management, and query processing. We first present the details of each architecture below.

Standalone clients: In the standalone clients model, shown in Figure 2(a), each client is a single node connecting to the service provider individually. The client does not directly share the data with the other clients. Possible example

[4] The details of creation of those index values can be found in [7].

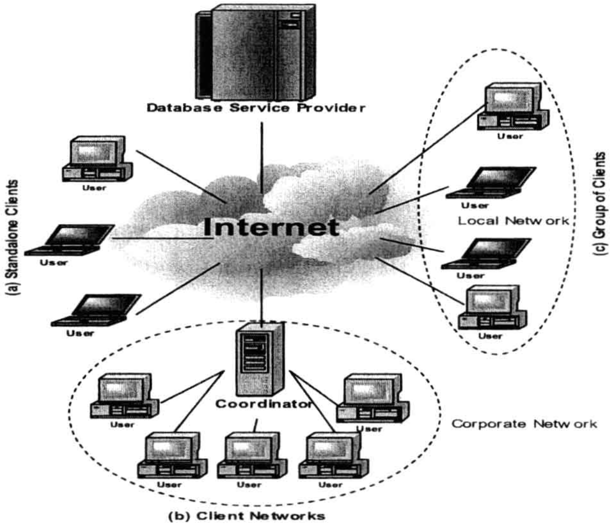

Fig. 2. Architectural model alternatives for database service

for the clients of this architecture is personal users accessing to the services, such as e-mail, rent-a-spreadsheet etc., via a web browser or a lightweight application interfaces.

Client networks: In this architecture, shown in Figure 2(b), the client of the service is a network rather than the individual nodes. A characteristic example for this architecture is larger corporations, which maintain their own network infrastructure as corporate networks and outsource some or all of their IT operations. In this model, the nodes inside the network utilize a connection point (or multiple points) to communicate with the service provider. We call this distinguished node as *coordinator node.* The coordinator node is responsible for set of operational tasks, such as maintaining metadata information required to execute queries directly over encrypted data (as described in Section 2.1), executing transactional semantics in the multi-tier client/server architecture, and the key management tasks as we describe in Section 3.

Group of clients: In this case, as shown in Figure 2(c), multiple clients access to the same service individually. Those clients are somehow related to each other. The relationship can be organizational, i.e., the group of clients belonging to an organization, or data sharing or both. A typical example for this model is small companies, which have multiple but limited number of users. They do not want to (or need to) maintain an integrated network infrastructure containing the coordinator nodes as in client networks case. Nonetheless, they need to enable collaboration among the user nodes in the organization as the users (or employees) of them would be sharing the data in terms of querying and updating and are related by business means. Therefore the user nodes are connected to each other to share local information, such as the metadata. Inherently this information is managed in a distributed fashion. We will not further discuss the distributed data management techniques in this context since it would cause us to diverge from the main content of the paper.

3 Key Management

Key management is a group of policies and procedures that regulate the maintenance of the encryption keys within the system. The key management techniques have been extensively studied in the applied cryptography literature [14]. We discussed the relevant aspects of the key management techniques to database-as-a-service model by considering their implications on the system implementation issues elsewhere [11]. Therefore, here, we only provide necessary background on the specific key management functions. We consider the following components of the key management architecture: *key generation*, *key installation*, *key distribution*, and *key update*. We will discuss each of these functionalities in the context of the DAS model and indicate where the each of the tasks are identified in the respective subsections. Key updates are discussed separately as they are the main focus of the paper. We also define the key assignment granularity, which affects the discussion of the techniques presented in the paper. In addition, we introduce a data structure, called *key registry*. The key registry is used to store the encryption key in the system.

3.1 Key Assignment Granularity

A key can be used to encrypt different database objects in the database, such as a table or a row. We call this as the assignment granularity of the key. The selection of granularity would have its own pros and cons, depending on the system setup, limitations on computing and storage of the client etc., and the security requirements. Discussion on these alternatives can be found in [11]. In this paper, we assume that the key assignment granularity is vertical-partitions-level.

In *vertical-partitions-level* key assignment granularity case, a group of database rows are encrypted with the same key. In the most extreme case, a different key is used for each row. Alternatively, the rows can be grouped. In our system we define the groups as the non-overlapping intervals on the RIDs. All rows in a value interval are encrypted with the same key. For example, the key k_1 can be used to encrypt the rows of *emp* table, whose *mgr.RID* values fall in $[1, 10]$ and the key k_2 can be used for the rows, whose *mgr.RID* values fall in $[11, 25]$.

3.2 Key Generation

Key generation involves the creation of the encryption keys that meet the specifications of the underlying encryption techniques. These specifications define the properties, such as size, randomness, that the keys should have. The medium in which keys are generated is of particular interest for the DAS model since the decision has both security and performance implications [11].

此为试读，需要完整PDF请访问：www.ertongbook.com