Adaptive, Secure, and Scalable Distributed Data Outsourcing: A Vision Paper

Li Xiong
Emory University
lxiong@emory.edu

Slawomir Goryczka
Emory University
sgorycz@emory.edu

Vaidy Sunderam
Emory University
vss@emory.edu

ABSTRACT

The growing trend towards grid computing and cloud computing provides enormous potential for enabling dynamic, distributed and data-intensive applications such as sharing and processing of large-scale scientific data. It also creates an increasing challenge for automatically and dynamically placing the data in the globally distributed computers or data centers in order to optimally utilize resources while minimizing user-perceived latency. This challenge is further complicated by the security and privacy constraints on the data that are potential sensitive. In this paper, we present our vision of an adaptive, secure, and scalable data outsourcing framework for storing and processing massive, dynamic, and potentially sensitive data using distributed resources. We identify the main technical challenges and present some preliminary solutions. The key idea of the framework is that it combines data partitioning, encryption, and data reduction to ensure data confidentiality and privacy while minimizing the cost for data shipping and computation. We believe the framework will provide a holistic conceptual foundation for secure data outsourcing that enables dynamic, distributed, and data-intensive applications and will open up many exciting research challenges.

Categories and Subject Descriptors

H.2.7 [**Database Administration**]: Security, integrity, and protection; H.2.8 [**Database Applications**]: Statistical databases; H.3.4 [**Systems and Software**]: Distributed systems

General Terms

Design, Management, Security

Keywords

privacy preserving data publishing, secure data outsourcing, differential privacy, cloud computing, data-as-a-service

3DAPAS'11, June 8, 2011, San Jose, California, USA.
Copyright 2011 ACM 978-1-4503-0705-5/11/06 ...$10.00.

1. INTRODUCTION

The growing trend towards grid computing and cloud computing is offering both new opportunities and challenges for enabling dynamic, distributed and data-intensive (D^3) applications such as sharing and processing of large-scale scientific data. Researchers or data collectors can outsource their massive data as well as data processing tasks to grid resources or the enormous data centers offered by cloud providers, exemplified by Google AppEngine, Amazon's EC2 and Microsoft's Azure. Such a computing paradigm also creates an increasing challenge for automatically and dynamically placing the data in the globally distributed computers or data centers in order to optimally utilize the resources while minimizing user-perceived latency. This challenge is further complicated by the security and privacy constraints on the data. Indeed, data security is widely recognized as a major barrier for widespread adoption of deploying data into open distributed systems, e.g. clouds [32, 5, 36]. Users are reluctant to place their sensitive data in the cloud with concerns about data disclosure to potentially untrusted cloud providers and other malicious parties. Untrusted cloud providers may scan cloud users' data and even sell them. New threats also arise from the multi-tenant cloud infrastructure based on virtual machines (VMs) as the resources are shared with other potentially malicious tenants. In fact, a recent study [39] showed a cross-VM attack that can effectively target and extract confidential information from specific cloud instances. Biocompute is an example grid-based system that leverages grid-computing resources composed of a large network of personal computers for solving large bioinformatics problems [11]. The amount of data delivered by the Biocompute triggers new challenges for its developers. In addition, the possibility of opening Biocompute for external users' use will raise many privacy concerns.

Not surprisingly, part of the data confidentiality issues in the distributed outsourcing reflects the well-established security challenges in the traditional data outsourcing or Database-as-a-Service (DAS) setting in which a client outsources its data to a remote untrusted database server which is then responsible for their storage and management [42]. A common approach is to encrypt the data before sending them to a remote server. Many techniques (such as [44, 28, 27, 12, 30, 4, 8, 31]) focus on supporting specific queries on encrypted database. The recent break-through of the fully homomorphic encryption scheme[24, 25, 46] bears the potential to allow a user to store fully encrypted data while enabling arbitrary computations on the encrypted data, however, its computation cost is prohibitive in practice. It remains an open challenge to support versatile and efficient computation on the outsourced data with assurances of data confidentiality and data privacy.

For secure data outsourcing to distributed resources, a potential solution is the secret sharing scheme [43] that distributes the original data into shares and each can be placed on a resource. However, the limitation that each share of the secret must be at least as large as the secret itself makes it not feasible for large scale data. Several works studied data partitioning that allow partial or no encryption [3, 15, 16, 10, 14]. While they mitigate the performance impact of cryptographic operations, the security guarantees on the unencrypted data warrants further research.

In addition, data outsourcing to distributed resources such as cloud brings about new opportunities and challenges. According to the NIST definition [35], key characteristics of cloud computing include on-demand service, broad network access, resource pooling, rapid elasticity, and metered service similar to a utility. In other words, what differentiates cloud-hosting providers from traditional hosting providers is their ability to offer *elastic* resources [36] which can respond to *dynamic* data volumes. This also challenges us to reexamine the existing solutions and design next generation secure data outsourcing techniques that allow users to rapidly and dynamically provision their resource needs for outsourcing and realize the full potential of the elasticity and pay-per-use of the cloud.

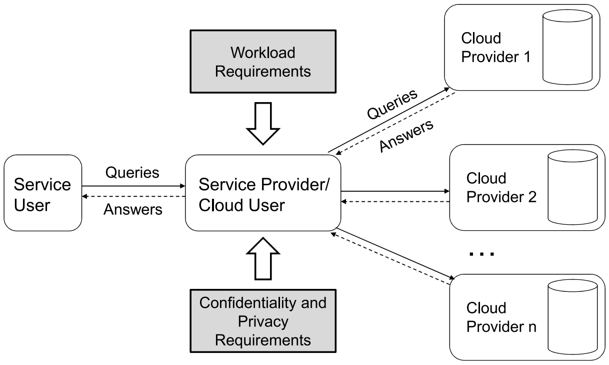

Figure 1: Distributed Data Outsourcing in the Cloud

Contributions. We present our vision of an adaptive, secure, and scalable data outsourcing framework for sharing and processing of massive, dynamic, and potentially sensitive data using distributed resources. Figure 1 depicts an example cloud outsourcing setting in which a user outsources its data to distributed data centers or resource providers for data storage and management. We identify the main technical challenges and present some preliminary solutions.

1. Secure and scalable outsourcing design that combine data partitioning, encryption, and data reduction such as compressed or statistical data outsourcing to ensure data confidentiality and privacy while minimizing the cost for data shipping and computation. Each resource provider may store parts of the data in original, encrypted, or reduced form. Algorithms can be developed to allow users to preprocess their data for secure outsourcing on distributed resource providers that systematically balance the requirements on confidentiality and privacy, scalability, and analytical utility of the data for a given workload.
2. Adaptive outsourcing design that allow users to dynamically provision their outsourcing needs with data updates and changing query workload. Control-theory based mechanisms can be developed to effectively model and estimate the changing query workload and changing data for dynamically adjusting the outsourcing design.

We believe the framework will provide a holistic conceptual foundation for secure data outsourcing that enables large scale sharing and processing of massive and potentially sensitive data on distributed data centers. We expect that this paper will open up exciting perspectives for future systems and database research on D^3 applications.

2. RELATED WORK

We briefly review the research that are closely relevant and discuss how the presented work leverages and advances the state-of-the-art.

Secure Data Outsourcing. Secure data outsourcing or secure Data-As-a-Service (DAS) studies the problem where a client stores data on untrusted and dynamically changing group of database servers. Ensuring security in DAS has received increasing attention in recent years [42]. A common approach is to store entirely encrypted data on the server and many works have focused on supporting specific queries on encrypted database with weaker encryption (such as [28, 12, 30, 4]). Techniques developed in the cryptographic community (such as [44, 7, 27, 13, 8, 1, 31]) provide strong security guarantees but are costly for medium-size to very large databases. The recent break-through of the fully homomorphic encryption scheme [24, 25, 46] bears the potential to allow a user to store fully encrypted data while enabling arbitrary computations on the encrypted data, however, its computation cost is prohibitive in practice.

Several recent works studied the data outsourcing problem in the context of highly distributed environments. In [39] authors showed that the fundamental risk arises from the multi-tenant cloud infrastructure based on virtual machines (VMs) and demonstrated a cross-VM attack that can effectively target and extract confidential information from specific cloud instances. Puttaswamy et al. [38] proposed a solution to encrypt functionally encryptable data, which is sensitive data that can be encrypted without limiting the functionality of the cloud service. Another line of research aims to efficiently check if the client's data stored at untrusted servers has been tampered with or deleted. [6] introduced a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. [21] extends the PDP model to support provable updates to stored data. On the other hand, [9] introduced HAIL (High-Availability and Integrity Layer), a distributed cryptographic system that allows a set of servers to prove to a client that a stored file is intact and retrievable.

Privacy Preserving Data Analysis and Publishing. Privacy preserving data analysis or publishing [18, 22] has received considerable attention in recent years as a promising approach for sharing useful information while preserving data privacy. In a typical model, a data provider collects data from record owners and provides an access mechanism with *sanitized* answers or a *sanitized* version of the data for data users for querying or analysis purposes that ensures privacy protection. Most literature [22] following the seminal work on k-anonymity [41, 45] and l-diversity [33] adopts relaxed privacy notions for the non-interactive model by considering specific attacks and assuming the attacker has limited background knowledge. Differential privacy [18, 19] is emerging as a strong notion for guaranteeing privacy even with arbitrary background knowledge. A notable work in the distributed computation context is [40], which presents a MapReduce-based system for distributed computations on sensitive data and uses differential privacy to guarantee the computation output does not violate individual privacy. While the setting of data publishing is different from that of data outsourcing, this body of research provides a foundation for the inference control on outsourced (published) data that are not encrypted.

Data Placement and Load Balancing for Distributed Data Intensive Systems. While (computation) load balancing has been studied extensively in the distributed environment, there has been relatively less work on automatic data placement and load balancing for data-intensive distributed applications. For example, a load balancing algorithm, which minimizes the total time spent on processing data deployed in a distributed environment, has been introduced by Glimcher, et. al. [26]. They considered data partitioning by dividing the data across multiple data repositories and developed a resource allocation and scheduling algorithm for minimizes the total data processing time. Yu, et. al. evaluated the interfaces and implementations of aggregation functions for several state of the art distributed computing systems [48]. Their computational model is based on the MapReduce approach to distributed computations. [2] presented Volley, a system for automatic data placement across geo-distributed datacenters. It presents a promising approach that takes into account the bandwidth costs between data centers and data center capacities as well as access patterns and user locations. While these techniques are promising, they do not consider privacy and security which are important constraints for outsourcing potentially sensitive data for distributed computations.

3. FRAMEWORK

In this section, we present our problem setting followed by our proposed framework. We identify the main technical challenges and present some preliminary solutions.

3.1 Problem Setting and Types of Data

We consider the data outsourcing setting where a user outsources its data to distributed resource providers, e.g. cloud providers, for data storage and processing as well as sharing with other data users. We assume data users and resource providers are semi-honest or honest-but-curious. They may mine data, if allowed, but do not tamper with it. However, data records may be compromised and the data may be disclosed to other malicious parties. Note that the research on data outsourcing assurance as we discussed earlier provides techniques for checking whether the outsourced data has been tampered with or deleted.

As an example scenario, hospitals may wish to outsource their data and application servers to cloud providers, leveraging the scalable and elastic platform in the cloud. However, personal health information is protected under HIPAA. An outsourcing solution needs to be HIPAA-compliant while maintaining efficient operations on the data.

Our vision is to use an integrative approach of data partitioning, encryption, and data reduction to ensure data confidentiality and privacy while minimizing the cost for data shipping and computation. Each resource provider may store part of the data in original, encrypted, or reduced form. The arrangement should be designed in such a way that it systematically balances the confidentiality and privacy requirement as well as the efficiency and scalability requirement. Based on different types of data and different nature of sensitive information they carry on, different data operations and strategies may be needed. We first briefly discuss the different types of data, then describe potential solutions or alternatives for different kinds of data operations, and finally formalize the optimal design problem.

Structured Records. Given structured data, a common approach to balance the load is to partition the data, for example, horizontally partitioned (based on subsets of records), or vertically partitioned (based on subsets of attributes), such that each partition can be stored and locally processed at a data resource. In addition, we also consider data reduction techniques to precompute data statistics or transformed data to be deployed to distributed resources in order to minimizing the cost for data shipping while maintaining the utility of the data for analysis. The main challenge is how to guarantee the partitioned or reduced data satisfies given privacy constraints while optimizing the data utility as well as computation and communication cost.

Multimedia Data. Text and image data collected from scientific experiments or medical records can be massive and will greatly benefit from outsourcing. Once deployed into open access distributed environment like clouds, may be used for research, further medical consultations and educational purposes. While traditional text and image compression techniques can be potentially used to reduce the data that needs to be moved, a challenge is to guarantee the reduced data also preserve the privacy for individuals while maintaining the utility for data analysis tasks. The inference control on such kinds of data has been significantly less studied compared to structured records. For example, identifying information can be hidden in the text which makes sanitizing text data a challenge. For image data, not only the meta-data needs to be sanitized or encrypted, the images themselves (e.g. a medical image showing a patient's face or Google's street view with people) also need to be sanitized. In addition, the data may be updated frequently. Identifying sets of frequently updated data efficiently and optimizing the data update process is a big challenge for this scenario.

3.2 Encryption and Partitioning

An important building block of our framework is encryption and partitioning (or fragmentation) techniques [16, 42]. Encryption consists in encrypting all the values of an attribute, thus making them unintelligible to unauthorized users. Fragmentation consists in partitioning data records (horizontal partitioning) or attributes (vertical partitioning) in subsets such that only records or attributes in the same fragment are visible together.

Given structured data represented as relations with each relation over a relational schema, we can consider the following confidentiality and privacy requirements.

Definition 3.1 (Confidentiality constraint [16]). *Given a set A of attributes, a confidentiality constraint c over A is:*
1) a singleton attribute $a \in A$, stating that the values of the attribute are sensitive (attribute visibility); or
2) a subset of attributes in A, stating that the association between values of the given attributes is sensitive (association visibility).

While singleton constraints can only be satisfied via encryption, the association constraints can be enforced either by encrypting at least one of the attributes involved in the constraint or by splitting the attributes in such a way that their association cannot be reconstructed.

We note that, unlike previous approaches that only considers disjoint fragmentation, we consider overlapping fragmentation. This essentially allows replication of the data which may allow more efficient data processing and services. Of course, it introduces further complexity to the problem as it increases the search space exponentially. In addition, the security implications or inference of the partitioned data have not been carefully studied in the existing work. In particular, partitioning may not guarantee the constraint on association visibility when an attacker has sufficient background knowledge. For example, when an attacker knows all the attribute values in two partitions of all records but the victim, s/he will be able to associate the attribute values of the victim from the two partitions. The existence of adversaries with any background knowledge is a reasonable assumption when data are deployed in a distributed environment. In our example scenario, possible malicious data users may include any external person with limited background knowledge as well as a physician who moved to another hospital and maintained detailed records of his previous patients.

3.3 Data Synopsis Outsourcing

A novel aspect of our approach is to outsource synopsis of data (or reduced data) and use it together with encryption and partitioning to ensure high usability of data while preserving its privacy. Using all three ways of storing data allow us to adapt to the query workload.

For example, for structured data, many queries or data mining tasks rely on aggregate information such as sum or count that only need statistical information of the original data. Such statistical information can be prepared to preserve privacy and used to reply those queries efficiently. In our example scenario, a hospital may wish to share a statistical synopsis of their data with a research institution for data analytics. In order to protect against inference on the original data using the statistical information, we can use differential privacy [18] to provide a provable privacy guarantee. Recent proposals for differentially private data release such as [29, 47] provide potential solutions. For example, [47] uses an adaptive multi-dimensional partitioning strategy for releasing accurate and differentially private histograms.

Definition 3.2 (Differential privacy [17]). *An access mechanism $\mathcal{A}$ satisfies α-differential privacy if for any neighboring database D_1 and D_2, for any query function Q, $r \subseteq Range(Q)$, $\mathcal{A}_Q(D)$ is the mechanism to return an answer to query $Q(D)$,*

$$Pr[\mathcal{A}_Q(D_1) = r] \leq e^{\alpha} Pr[\mathcal{A}_Q(D_2) = r]$$

Differential privacy makes no assumption on the background knowledge of malicious data users. It requires the statistics to be formally indistinguishable when computed with and without any particular record in the dataset, as if it makes little difference whether an individual is being opted in or out of the database. Thus, in the worst case, an adversary who knows all but one data record will only be able to infer information about this record with limited and pre-defined probability. For example, let $D_2 \subset D_1$ and $D_1 \setminus D_2 = \{d\}$. Assume an adversary knows all records in D_2. Then, the probability of breaching privacy of d is limited by e^{α}, where α is defined by a data provider.

For textual or image data, well known data compression techniques such as string encoding, wavelet transforms, can be explored. However, the inference on the data is less understood. For the raw text (without being compressed), de-identification techniques (such as [23]) can be used to anonymize the text with weak privacy guarantees before outsourcing it. It will be an important and interesting challenge to study the inference implications of different data compression or data transformation techniques and their tradeoff on data utility and data reduction ratio which will have a direct impact on data shipping cost.

3.4 Optimal Outsourcing Design

Given a set of dynamic data with potential confidentiality and privacy constraints, a pool of resources, and an estimated workload, our problem is to design an optimal outsourcing arrangement. The arrangement consists of proper encryption, fragmentation, and synopsis outsourcing that minimizes the cost associated with data shipping and processing for the given workload.

A query workload can be modeled as a set of queries, where each query $Q_i, i = 1, \ldots, m$, is characterized by an execution frequency $freq(Q_i)$. The queries can be regular select queries or aggregate queries which can return results for further data analysis. For each type of queries, we can compute the cost associated with a given outsourcing strategy by aggregating the cost components from different operations. For example, an aggregate query may be directly answered by the outsourced statistics that is stored in the cloud provider even if the attributes involved are encrypted. Another query may involve selection of an unencrypted attribute, decryption of an encrypted attribute, and a join of the two attributes.

A special case of the problem using only fragmentation is shown to be NP-hard [16, 42]. Heuristic strategies will be needed for efficiently finding a good design and query answer strategy. In particular, a multi-dimensional greedy strategy may be promising. At each iteration, we can select a specific partitioning, encryption of certain attribute, or data compression or statistical summarization of a certain partition based on a scoring metric. The process continues until the data satisfies all the constraints. However, with time both query workload and data may change. Updating all data duplications and representations, i.e. synopses and partitions, is a challenge we will face in our research.

3.5 Adapting to Evolving Query Workload

So far, we have assumed the workload is static. In this subsection, we sketch some potential techniques to dynamically estimate the changing workload using controller mechanisms.

Given our model of the query workload, let $freq(Q_i)(t)$ denote the current execution frequency of Q_i at time t. It can be simply computed as the average number of Q_i received by the service provider over the recent period of time. We estimate $freq(Q_i)$ using Equation 1. Note that $freq'(Q_i)(t)$ denotes the derivative of $freq(Q_i)(x)$ at $x = t$.

$$\begin{aligned} freq(Q_i) &= \alpha * freq(Q_i)(t) \\ &+ \beta * \frac{1}{t} * \int_0^t freq(Q_i)(x)dx \\ &+ \gamma * freq'(Q_i)(t) \end{aligned} \tag{1}$$

Equation 1 resembles a Proportional-Integral-Derivative controller used in control systems [37]. The first component (*proportional*) refers to the contribution of the current workload received at time t. The second component (*integral*) represents the past workload (history information). The third component (*derivative*) reflects the sudden changes in the workload in the very recent past. Choosing a larger value for α biases the estimated workload to the current workload. A larger value of β gives heavier weight to the historical workload in the past. The averaging nature of the proportional and integral components enables our model to tolerate errors in the measurement of current frequency and reflect consistent workload. A larger value of γ amplifies sudden changes in the workload in the recent past (as indicated by the derivative of the trust value) and handles sudden fluctuations.

Based on the above abstract model, we can design discretized implementations that efficiently and effectively estimate the query frequency. Suppose the data provider (e.g. cloud resources provider) stores the query frequency over the last $maxH$ (maximum history) intervals. The integral could be derived as a weighted sum over the last $maxH$ values. The weights w_k could be chosen either aggressively or conservatively. An example of an aggressive summarization is the exponentially weighted sum, that is, $w_k = \rho^{k-1}$ (typically, $\rho < 1$). Note that choosing $\rho = 1$ is equivalent to H being the average of the past $maxH$ frequency values. Also, with $\rho < 1$, H gives more importance to the more recent values.

A particular challenge to address is the potential large history of queries that need to be stored in order to accurately estimate the frequency for each type of queries. The computation time also increases with the amount of data to be processed. We propose to aggregate data over intervals of exponentially increasing length in the past $\{k^0, k^1, \ldots, k^{m-1}\}$ into m values (for some integer $k > 0$). Observe that the aggregates in the recent past are taken over a smaller number of intervals and are hence more precise. This permits the system to maintain more detailed information about the recent workload and retain fading memories (less detailed) about the workload in the past. Given a fixed value to the system-defined parameter m, one can trade-off the precision and the history size by adjusting the value of k.

3.6 Adapting to Dynamic Data

Having discussed the techniques to address dynamic workload, we now discuss the new challenges presented by the dynamic data. For different scenarios we may expect different dynamic of data changes. For our example with the hospital, we would expect to have short periods of time where a patient is admitted to the hospital and its records are updated. However, if a patient has a chronic disease then his records may be updated constantly for a long time. In addition, different times of the year may impact those dynamics as well. During a pre-epidemic time the rate of patients will increase dramatically. Exercising cloud flexibilities during that time will ensure fast access to the data.

How to adaptively update the data in the cloud while balancing the computational overhead and accuracy of the synopsis is a challenge. Considering the statistical data outsourcing primitive we have discussed, in order to prevent disclosure of data dynamics, one approach is to estimate the data change rate through the perturbed statistical data that are deployed at the data resource. However, updating the deployed data too often increases the amount of noise that need to be added to the synopsis. Careful privacy budget management needs to be performed [20, 34].

The different versions of outsourced data will create additional disclosure risks from inference. The multiple statistical synopses, if using differential privacy, will still satisfy differential privacy (but with an accumulated privacy parameter or downgraded privacy) because of the composition properties of differential privacy [34]. Adding a dynamic factor to the whole picture shows that such subtle security implications need to be thoroughly considered.

4. CONCLUSION

We presented an adaptive, secure, and scalable data outsourcing framework for sharing and processing of massive, dynamic, and potentially sensitive data using distributed resources. It creates exciting research opportunities for designing new outsourcing design techniques as well as systems techniques that make different types of cloud and local platforms compatible, host practical manifestations of remote databases, and perform at optimal levels in order to make the technology eminently usable. We believe the research direction is exciting in its potential to be transformative in terms of dramatically increasing usability of distributed computational platforms and making "information processing as a utility" a reality. For the foreseeable future, user confidence in data security and confidentiality will be a critical factor in determining adoption; at the same time, acquiring storage and compute resources from professional providers is technologically and economically optimal. Reconciling these two aspects has significant implications.

Acknowledgement

The research is partially supported by a Cisco Research Award. The authors would like to thank the valuable comments from the anonymous reviewers for improving the final version of the paper.

5. REFERENCES

[1] M. Abdalla, M. Bellare, D. Catalano, E. Kiltz, T. Kohno, T. Lange, J. Malone-Lee, G. Neven, P. Paillier, and H. Shi. Searchable encryption revisited: Consistency properties, relation to anonymous ibe, and extensions. *Journal of Cryptology*, 21:350–391, 2008.

[2] S. Agarwal, J. Dunagan, N. Jain, S. Saroiu, A. Wolman, and H. Bhogan. Volley: automated data placement for geo-distributed cloud services. In *Proc. of the 7th USENIX conference on Networked systems design and implementation*, NSDI, page 2, 2010.

[3] G. Aggarwal, M. Bawa, P. Ganesan, H. Garcia-Molina, K. Kenthapadi, R. Motwani, U. Srivastava, D. Thomas, and Y. Xu. Two can keep a secret: A distributed architecture for secure database services. In *CIDR*, pages 186–199, 2005.

[4] G. Amanatidis, A. Boldyreva, and A. O'Neill. Provably-secure schemes for basic query support in outsourced databases. In *DBSec*, 2007.

[5] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. A view of cloud computing. *Commun. ACM*, 53:50–58, April 2010.

[6] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song. Provable data possession at untrusted stores. In *Proc. of the 14th ACM Conference on Computer and Communications Security*, CCS, pages 598–609, 2007.

[7] D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano. Public key encryption with keyword search. In *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 506–522. Springer Berlin / Heidelberg, 2004.

[8] D. Boneh and B. Waters. Conjunctive, subset, and range queries on encrypted data. In *Proc. of the 4th Conference on Theory of Cryptography*, 2007.

[9] K. D. Bowers, A. Juels, and A. Oprea. Hail: a high-availability and integrity layer for cloud storage. In *Proc. of the 16th ACM Conference on Computer and Communications Security*, CCS, pages 187–198, 2009.

[10] M. Canim, M. Kantarcioğlu, and A. Inan. Query optimization in encrypted relational databases by vertical schema partitioning. In *Proc. of the 6th VLDB Workshop on Secure Data Management*, SDM '09, pages 1–16, 2009.

[11] R. Carmichael, P. Braga-Henebry, D. Thain, and S. Emrich. Biocompute: towards a collaborative workspace for data intensive bio-science. In *Proc. of the 19th ACM International Symposium on High Performance Distributed Computing*, HPDC '10, pages 489–498, 2010.

[12] A. Ceselli, E. Damiani, S. D. C. D. Vimercati, S. Jajodia, S. Paraboschi, and P. Samarati. Modeling and assessing inference exposure in encrypted databases. *ACM Trans. Inf. Syst. Secur.*, 8, February 2005.

[13] Y.-C. Chang and M. Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. In *Applied Cryptography and Network Security*, volume 3531 of *Lecture Notes in Computer Science*, pages 442–455. Springer Berlin / Heidelberg, 2005.

[14] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Keep a few: Outsourcing data while maintaining confidentiality. In *Computer Security – ESORICS 2009*, volume 5789 of *Lecture Notes in Computer Science*, pages 440–455. Springer Berlin / Heidelberg, 2009.

[15] V. Ciriani, S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Fragmentation and encryption to enforce privacy in data storage. In *Computer Security – ESORICS 2007*, volume 4734 of *Lecture Notes in Computer Science*, pages 171–186. Springer Berlin / Heidelberg, 2007.

[16] V. Ciriani, S. D. C. d. Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati. Fragmentation design for efficient query execution over sensitive distributed databases. In *ICDCS*, 2009.

[17] C. Dwork. Differential privacy. In *Automata, Languages and Programming*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12, 2006.

[18] C. Dwork. Differential privacy: a survey of results. *Lecture Notes in Computer Science*, 4978:1–19, 2008.

[19] C. Dwork. A firm foundation for private data analysis. *Commun. ACM*, 54:86–95, January 2011.

[20] C. Dwork, F. D. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proc. of the 3rd Theory of Cryptography Conference*, pages 265–284, 2006.

[21] C. C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia. Dynamic provable data possession. In *CCS*, 2009.

[22] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys*, 42(4), June 2010.

[23] J. Gardner and L. Xiong. Hide: An integrated system for health information de-identification. In *EDBT*, 2009.

[24] C. Gentry. Fully homomorphic encryption using ideal lattices. In *Proc. of the 41st annual ACM Symposium on Theory of Computing*, STOC '09, pages 169–178, 2009.

[25] C. Gentry. Computing arbitrary functions of encrypted data. *Commun. ACM*, 53:97–105, 3 2010.

[26] L. Glimcher, V. T. Ravi, and G. Agrawal. Supporting load balancing for distributed data-intensive applications. In *HiPC*, pages 235–244, 2009.

[27] P. Golle, J. Staddon, and B. Waters. Secure conjunctive keyword search over encrypted data. In *Applied Cryptography and Network Security*, volume 3089 of *Lecture Notes in Computer Science*, pages 31–45. Springer Berlin / Heidelberg, 2004.

[28] H. Hacigümüs, B. R. Iyer, C. Li, and S. Mehrotra. Executing sql over encrypted data in the database-service-provider model. In *Proc. of the 2002 ACM SIGMOD International Conference on Management of Data*, SIGMOD, pages 216–227, 2002.

[29] M. Hay, V. Rastogi, G. Miklau, and D. Suciu. Boosting the accuracy of differentially private histograms through consistency. *Proc. VLDB Endow.*, 3:1021–1032, September 2010.

[30] M. Kantarcioğlu and C. Clifton. Security issues in querying encrypted data. In *Data and Applications Security XIX*, volume 3654 of *Lecture Notes in Computer Science*, pages 924–924. Springer Berlin / Heidelberg, 2005.

[31] J. Katz, A. Sahai, and B. Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In *Proc. of the Theory and Applications of Cryptographic Techniques 27th annual international conference on Advances in Cryptology*, EUROCRYPT, pages 146–162, 2008.

[32] L. M. Kaufman. Data security in the world of cloud computing. *IEEE Security and Privacy*, 7:61–64, July 2009.

[33] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. In *ICDE*, 2006.

[34] F. D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *SIGMOD*, 2009.

[35] P. Mell and T. Grance. Nist definition of cloud computing v15, 2009. http://csrc.nist.gov/groups/SNS/cloud-computing/cloud-def-v15.doc.

[36] D. Molnar and S. Schechter. Self hosting vs. cloud hosting: Accounting for the security impact of hosting in the cloud. In *The Ninth Workshop on the Economics of Information Security (WEIS 2010)*, 2010.

[37] H. Ozbay. *Introduction to Feedback Control Theory.* CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1999.

[38] K. P. N. Puttaswamy, C. Kruegel, and B. Y. Zhao. Silverline: Toward data confidentiality in third-party clouds. Technical Report 2010-08, Dept. of Computer Science, University of California Santa Barbara, 8 2010.

[39] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In *Proc. of the 16th ACM Conference on Computer and Communications Security*, CCS, pages 199–212, 2009.

[40] I. Roy, S. T. V. Setty, A. Kilzer, V. Shmatikov, and E. Witchel. Airavat: security and privacy for mapreduce. In *NSDI*, 2010.

[41] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. on Knowl. and Data Eng.*, 13, November 2001.

[42] P. Samarati and S. D. C. di Vimercati. Data protection in outsourcing scenarios: issues and directions. In *ASIACCS*, 2010.

[43] A. Shamir. How to share a secret. *Commun. ACM*, 22:612–613, November 1979.

[44] D. X. Song, D. Wagner, and A. Perrig. Practical techniques for searches on encrypted data. In *Proc. of the 2000 IEEE Symposium on Security and Privacy*, pages 44–55, 2000.

[45] L. Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10, October 2002.

[46] M. van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan. Fully homomorphic encryption over the integers. In *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 24–43. Springer Berlin / Heidelberg, 2010.

[47] Y. Xiao, L. Xiong, and C. Yuan. Differentially private data release through multidimensional partitioning. In *Proc. of the 7th VLDB conference on Secure data management*, SDM, pages 150–168, 2010.

[48] Y. Yu, P. K. Gunda, and M. Isard. Distributed aggregation for data-parallel computing: interfaces and implementations. In *Proc. of the ACM SIGOPS 22nd symposium on Operating systems principles*, SOSP '09, pages 247–260, 2009.