AnonymousCloud: A Data Ownership Privacy Provider Framework in Cloud Computing

Safwan Mahmud Khan and Kevin W. Hamlen
Department of Computer Science
University of Texas at Dallas
Richardson, Texas, USA
Email: {safwan,hamlen}@utdallas.edu

***Abstract*—A means of reliably concealing ownership of cloud data without impeding computation over the data is presented and evaluated. This facilitates information privacy enforcement in cloud environments by withholding data ownership information from cloud nodes that compute using the data. As a result, nodes that have access to private data in unencrypted form do not know who owns it, what role their computations play in the larger computational task, or to whom their computation results are ultimately delivered. To provide this data ownership privacy, the cloud's distributed computing resources are leveraged to implement an anonymizing circuit based on Tor, through which users submit private data and jobs. A tunable parameter k controls a trade-off between the degree of anonymity and the computational overhead imposed by the system. Anonymous authentication based on public-key cryptography safely links jobs and data to customers for billing purposes without revealing these associations to untrusted computation nodes. Simulation results demonstrate the potency of the system in presence of attackers.**

***Keywords*-anonymity; authentication; cloud computing; information security; privacy; Tor**

I. Introduction

Revolutionary advances in hardware, middleware, and virtual machines over the past few years have elevated cloud computing to a thriving industry that affords customers the ability to increase information processing capacity and add capabilities on the fly without significant investments in new infrastructure, training, or software licensing [1]. Cloud computing platforms (e.g., [2], [3], [4]) extend traditional distributed computing to encompass information infrastructure and resources as IT pay-per-use services in real time over the Internet [5], [6].

A significant barrier to the adoption of cloud services is customer fear of privacy loss in the cloud. In a survey by Fujitsu Research Institute on potential cloud customers, it was found that 88% of them are worried about who has access to their data and demanded more privacy [7]. Privacy activists have additionally argued a need for greater awareness of cloud computing privacy issues in government and law enforcement agencies [8].

This paper concerns the problem of privacy-preserving computation in the cloud. The complementary problem of secure storage of private cloud data has been studied extensively in the literature (cf., [9], [10]), but cannot usually be applied while the data is in decrypted form for the duration of a computation. Secure multiparty computation [11] and differential privacy [12] are both powerful approaches to privacy-preserving cloud computation on decrypted data, but are inapplicable to many real-world cloud computations. In particular, jobs submitted to the cloud as arbitrary binary code are difficult to automatically reformulate as secure multiparty computations, and high differential privacy sometimes comes at the expense of highly imprecise, noisy results.

In these cases, the level of privacy can sometimes be improved by concealing data ownership, provenance, and/or semantics from the participants in a computation in addition to (or instead of) anonymizing the data itself. For example, a computation that mines medical data might be deemed insecure if cloud nodes receive sequences of numbers labeled "patient temperatures" with owner id "Mercy Hospital"; however, the same computation might be deemed suitably private if each node receives only unlabeled sequences of numbers amidst a context of millions of other similar anonymous jobs for thousands of diverse, anonymous users.

Our AnonymousCloud framework therefore conceals data provenance from cloud nodes that compute over the data, and conceals recipient identities in the form of IP addresses and ownership labels. Anonymization is achieved through the instantiation of a Tor anonymizing circuit [13] inside the cloud, through which private data and jobs are anonymously supplied by and returned to users. Circuit length is a tunable parameter k, affording a flexible trade-off between the degree of anonymity and the computational overhead of the circuit.

To maintain a pay-per-use business model, clouds must inevitably track ownership information at some level for billing and auditing purposes. AnonymousCloud therefore implements a public-key cryptography-based anonymous authentication that disassociates data ownership metadata from the private data it labels. Thus, a separate manager node that does not have access to the private data can bill customers appropriately using the ownership metadata, while

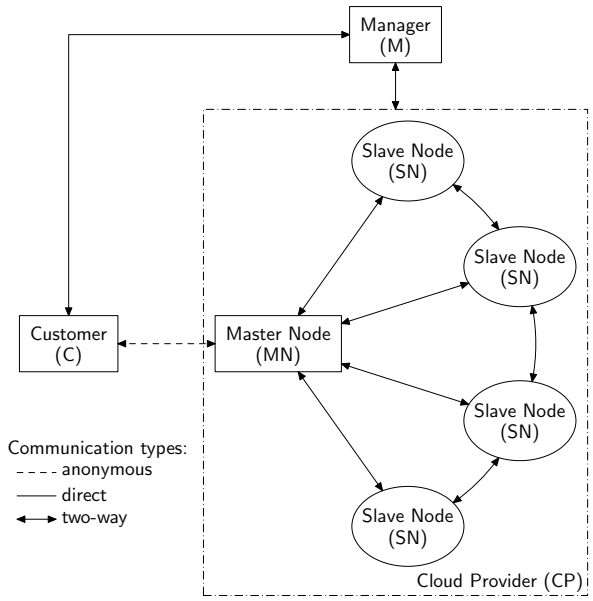

Figure 1. System architecture of AnonymousCloud

computation nodes that have access to the private data but not the metadata can securely carry out the anonymous job. Managers are trusted not to collude with computation nodes to violate privacy, but all other nodes including the master node are potentially malicious.

Our experimental results show via simulation that AnonymousCloud provides data ownership privacy with a high success rate against the collective efforts of a large percentage of attackers in the system, and does so with reasonable computational overhead.

The remainder of the paper is structured as follows: Section II describes the system architecture in detail. Section III reports experimental results and an analysis of the system. Section IV discusses some work related to privacy issues in cloud computing and AnonymousCloud. Section V concludes with a summary of results and suggestions for future work.

II. System Architecture

The system architecture of AnonymousCloud is given in Fig. 1. It consists of a *cloud provider* (CP) and a separate *manager* (M). Each are discussed respectively below, concluding with a discussion of the communication protocol between the two.

A. Cloud Providers

CPs provide computation services to customers (C), who submit computations as *jobs*. Customers can access these services in a pay-as-you-go fashion, with payment managed by the separate manager. Different CPs may vary in the details of their internal architectures (cf., [2], [3], [4]). We assume only that jobs are submitted to the CP via a centralized *master node* (MN), which partitions and schedules sub-computations across a large collection (e.g., hundreds of thousands) of *slave nodes* (SNs). All SNs are therefore directly connected to the MN, and there is arbitrary connectivity between the SNs.

AnonymousCloud amends Tor functionality [13] to the MN and SNs without modifying the job allocation and scheduling details of the cloud in any way. All principals (M, C, MN, and SNs) are additionally equipped with public-private key pairs from a well established *certificate authority* (CA). The public keys work as the symmetric or mutual keys during Tor circuit construction.

B. Managers

Managers are separate from the CP's computing infrastructure, and facilitate only customer authentication and billing. They have four primary responsibilities related to our work:

- M provides central storage of public keys for MN and SNs and serves them to C on request.
- M maintains a graph of SN connectivity. This facilitates Tor circuit construction by encoding the universe of available circuit links for circuit initialization.
- M provides each C a unique access token t and credentials c (e.g., a password) via which Cs can authenticate themselves to M to obtain cloud services.
- M additionally generates a unique nonce n for each of C's transactions to protect the authentication system against replay attacks. The authentication protocol is described in greater detail in Section II-C below.

In deployed implementations, M likely has additional responsibilities related to authentication, such as key revocation, certificate update, auditing, customer billing, etc. These responsibilities are deployment-specific, and are therefore beyond the scope of this paper.

C. Authentication Protocol

The authentication and circuit construction protocol of AnonymousCloud is depicted in Fig. 2 and detailed in Algorithm 1.

C begins each service transaction by communicating its access token and credentials to M, and requesting an anonymizing circuit of length k. If at least k connected nodes are available, M returns such a list; otherwise it may offer a list shorter than k. The returned list includes the public keys K_{SN} of all the selected slave nodes, as well as the public key K_{MN} of the master node. M also generates a fresh nonce n for C and stores a local copy. To prevent replay attacks [14], the next service request from C will only be authenticated by M if it is labeled with n.

In step 8 of Algorithm 1, C verifies the certificates with the certificate authority and stores them locally. To lessen the load, C may cache these results to avoid re-authenticating certificates that have not changed.

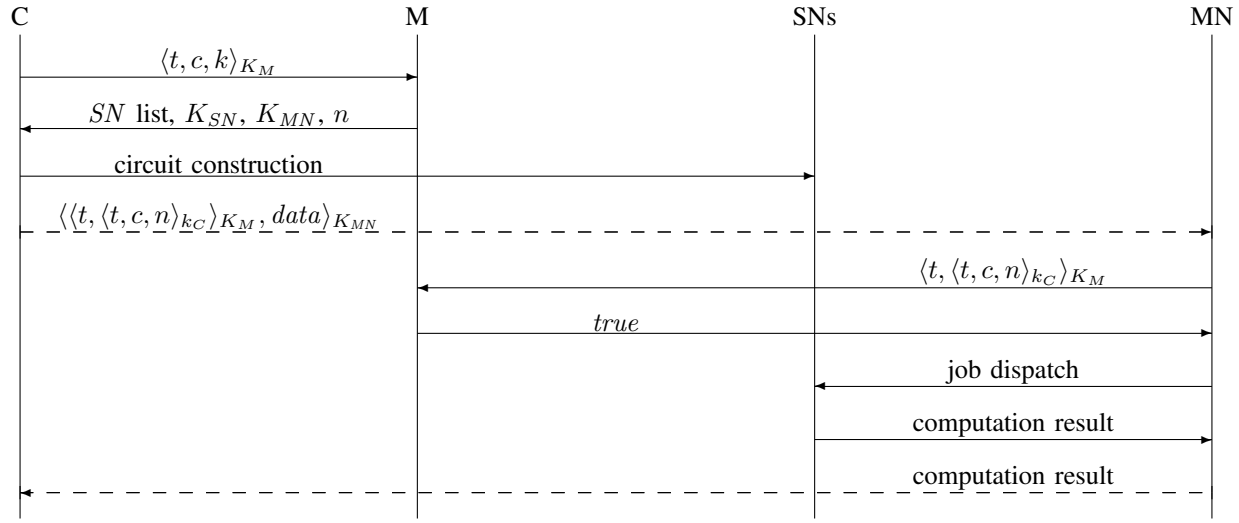

Figure 2. Authentication and circuit construction message sequence. Solid lines denote direct communications, whereas dashed lines denote anonymous communication through the Tor circuit.

Algorithm 1 Authentication and circuit construction protocol

C asks M to choose k available SNs based on SN connectivity
if C has invalid token t or invalid credentials c **then**
  M rejects the request from C
else
  repeat
    M selects k SNs (or the most available)
    M provides C with public keys K_{SN} and K_{MN} and fresh nonce n
    C validates keys K_{SN} and K_{MN} with the CA
    if any key fails validation by the CA **then**
      M revokes the invalid keys
    end if
  until all keys are valid
  C performs Tor circuit construction [13] over the SNs using the Ks as symmetric keys
  C signs t, c, and n with private key k_C and encrypts it with public key K_M, yielding $m = \langle t, \langle t, c, n \rangle_{k_C} \rangle_{K_M}$
  C sends $\langle m, data \rangle_{K_{MN}}$ in layered encryption format over the circuit to MN
  MN anonymously receives and decrypts the message with private key k_{MN}
  MN forwards m to M for authentication
  M decrypts m using k_M and verifies signature k_C using K_C, yielding t, c, and n
  M verifies t, c, and n; and it verifies K_C with the CA
  if authentication fails **then**
    M returns $false$ to MN
    MN discards the service request
  else
    M returns $true$ to MN
    MN dispatches the $data$ computation
    MN anonymously returns the result to C over the circuit
  end if
end if

C then transmits the requested computation and its data anonymously via the Tor circuit to MN in step 15. MN can read the data but not the encrypted ownership metadata $m = \langle t, \langle t, c, n \rangle_{k_C} \rangle_{K_M}$. It therefore forwards m to M for validation. M can read metadata m by decrypting it using its private key k_M, however it has no access to the associated job's data. M verifies C's digital signature using public key K_C, and validates K_C's certificate with the certificate authority (possibly caching the results to more efficiently service future requests). The access tokens t inside and outside the digital signature are additionally compared for equality, the credentials c are validated against t, and the nonce n is checked against the local copy. If all these steps succeed, M invalidates the nonce and returns $true$ to MN; otherwise it returns $false$ and the request is denied.

Upon successful authentication, MN dispatches the requested computation in accordance with the CP's internal architecture and protocols. If customer billing is based on computational resource consumption or other information that only becomes available as the computation progresses, MN can report such information to M without knowing the job's owner by tagging it with encrypted authentication information m. M can then attribute the incurred expenses to the correct customer.

Once the computation is complete, its results are anonymously delivered to C via the Tor circuit. The Tor circuit is then dismantled and its resources reclaimed by the CP.

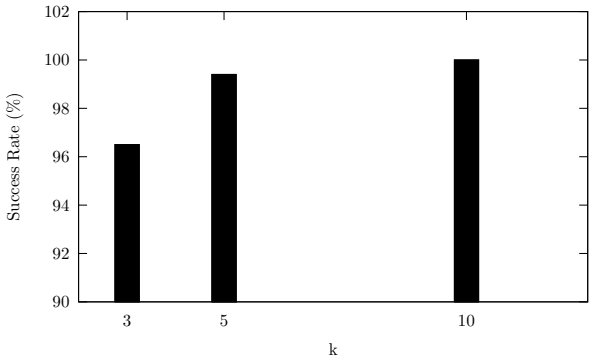

Figure 3. Privacy enforcement success as a function of Tor circuit length k in a cloud of $p = 30\%$ malicious slave nodes

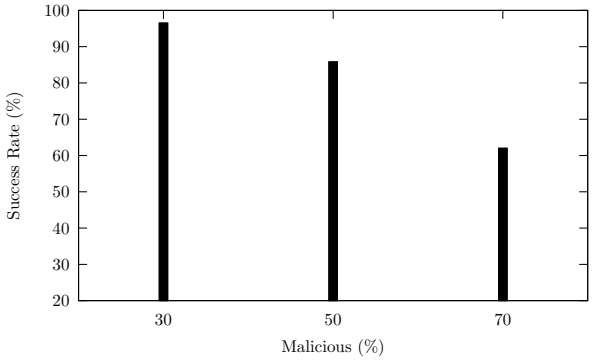

Figure 4. Privacy enforcement success for Tor circuits of length $k = 3$ as a function of percentage p of malicious slave nodes

III. Results and Analysis

We implemented AnonymousCloud in a simulation setup using Java, with experiments designed to measure the resilience of the system against privacy attacks and the computational overhead introduced by privacy protections. Each experimental data point is the result of simulating 1000 customer service requests to a cloud consisting of 1 master node and $N = 1000$ slave nodes. The simulation model includes the high-level protocol outlined in Section II-C but not low-level details of the underlying network and encryption operations, which are expected to be specific to each deployment.

A successful attack against our system is defined as the *linkability* [15] of private data to its corresponding ownership metadata by one or more malicious principals. Principals include the manager, the master node, and all slave nodes. Ownership metadata includes customer pseudonyms (*viz.*, access tokens and IP addresses) and authentication credentials. We assume that private data does not include pseudonyms or other information from which customer identities can be inferred; anonymizing the private data is the subject of related work.

In order for an attack against AnonymousCloud to succeed, the manager or master node (or both) must be malicious. Managers are the only principals that receive decryptable access tokens or credentials, and all other communications involving pseudonyms and data are conducted via Tor circuits having the master node as the only untrusted endpoint. Managers are separate from CPs and have a much smaller attack surface because they do not process customer-submitted computations. Our experiments therefore assume that managers are trusted, but that master nodes are always malicious. In addition, we assume that a percentage p of slave nodes are also malicious and collude with the malicious master node in an effort to violate privacy.

Fig. 3 plots the average privacy enforcement success rate for different Tor circuit lengths k in a cloud with a malicious master node and 30% malicious slave nodes. If $k = 0$, AnonymousCloud does not provide any anonymity; furthermore, any length less than 3 significantly increases the ease of successful end-to-end timing attacks [16]. We therefore restrict our attention to circuit lengths of at least 3. At $k = 3$ we obtain an already high success rate of 96.5%. Increasing k to 5 further elevates this 99.4%, and at $k = 10$ there were no privacy failures at all.

Fig. 4 plots the success rate of a fixed circuit length $k = 3$ in clouds with varying percentages p of malicious slave nodes. The results show how resilient our system is against malicious collectives. Even when clouds are 50% malicious,

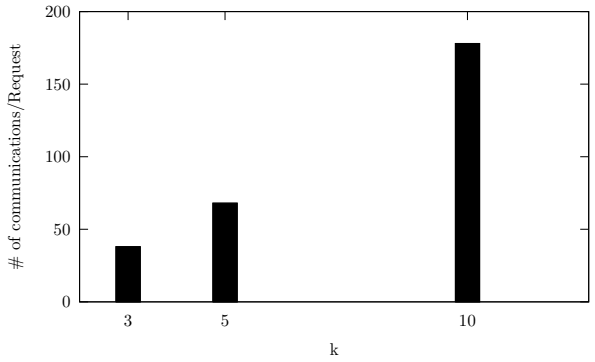

Figure 5. Communication overhead for different circuit lengths k

AnonymousCloud attains an 85.8% privacy preservation rate with just $k = 3$. When 70% of the cloud is malicious, the success rate drops to 62%, indicating that longer circuits are required to resist such pervasive attacks.

The results reported in Figures 3 and 4 can be generalized by observing that with high probability all k slave nodes in a Tor circuit must collude in order to compromise security. Thus, the curves in Figures 3 and 4 approximate the formula for random sampling without replacement:

$$success \approx 1 - \binom{pN}{k} \Big/ \binom{N}{k} \tag{1}$$

Privacy inevitably comes at some computational expense. It is therefore important to consider the computational cost associated with the introduction of anonymizing circuits. Fig. 5 plots the total number of messages per customer service request required to carry out AnonymousCloud's authentication protocol for varying circuit lengths k. Messages are tallied based on the implementation in the real time. This does not include computational overhead for cryptographic operations, which might noticeably increase the overhead in a real deployment. Although we did not consider these in our simulation, Fig. 5 nevertheless provides a general picture of the overhead that can be expected. We observe that as circuit length k increases, the total message count per request rises steeply. For $k = 3$ it is 38, and for $k = 5$, it almost doubles to 68. We conclude that there is a significant tradeoff between escalation of k and the overhead cost of communications.

The sharp increase in communications overhead potentially invites denial-of-service attacks by customers who request unreasonably long circuits. We therefore recommend incentivizing reasonable values of k by charging customers proportionally to the communications overhead incurred by their demanded level of privacy. Recall that master nodes can report computational expense information associated with anonymous jobs to managers by labeling it with the encrypted ownership data they received during authentication. This allows the master node to report the expense without knowing the identity of the customer. Managers may also want to impose a mandatory upper limit on k during authentication to further control congestion.

IV. Related Work

Data privacy concerns are widely recognized as a significant impediment to consumer confidence in cloud computing [7], [9], [10]. Associated challenges span at least three categories of related work: secure remote platform attestation (i.e., trusted computing), secure data storage, and information-centric security [17].

Trusted computing provides users high assurance that they are communicating with a remote server consisting of known, trusted hardware and software [18]. Secure storage regards the problem of safely storing private data in the cloud (usually in encrypted form) between computations that use it (e.g., [19]). In contrast, information-centric approaches imbue data with self-protecting properties, such as by representing it in a form amenable to direct computation on cyphertexts without decryption (e.g., [20]). AnonymousCloud's approach of decoupling private data from its provenance information can be viewed as an instance of the last of these approaches.

General data anonymization is a vast research area spanning many decades; however, the most widely used strategies for anonymization of data content are currently differential privacy [21] and k-anonymity for privacy-preserving microdata release [22]. Such research benefits our work by providing a means for customers to anonymize private data content before submitting it to the cloud. We therefore assume that customers interested in privacy submit data that divulges fewer secrets once it has been decoupled from provenance and semantic metadata, and that therefore benefits from our anonymization protocol.

Prior work has also explored decoupling document content from format and structure for more secure cloud storage and processing [23]. For example, HTML documents can be encoded in a format that separates their tree structures from the textual content of elements and attributes. Since a majority of private data resides in the content, this allows separate processing of structural-based queries in the cloud without divulging the private data.

To decouple and conceal provenance metadata, AnonymousCloud employs onion routing based on Tor [13]. Tor has become the most successful public anonymity communication service in the Internet, with tens of millions of users worldwide [24]. In Tor, initiators choose a path through network and build a circuit in which each node or onion router in the path knows only its successor and predecessor, but no other nodes in the circuit. Based on the chosen path or route, the initiator first encrypts the data with one layer of encryption for each node in the path, from the last node to the first. This is likened to the layers of an onion, with each hop peeling one layer as the data is forwarded to its destination. The data can only be read in plaintext once it

reaches the endpoint of the path and all layers have been peeled.

The Tor Cloud project [25] has implemented a full-scale Tor system within a production-level cloud that runs on the Amazon EC2 cloud computing platform [2]. It provides a user-friendly way of deploying bridges to help users access an uncensored Internet. Tor Cloud conceals user pseudonyms (e.g., IP numbers) from untrusted third-party services, but does not suffice to anonymously access data from a third-party cloud [26], since clouds require a means of authenticating users in order to control access to each user's private data and bill them appropriately.

Our work therefore extends cloud-based onion routing with an anonymous credential system for authentication [27]. Anonymous authentication provides zero-knowledge proof of identity, allowing data to be securely decoupled from provenance for enhanced privacy. More elaborate anonymous credential systems (e.g., [28], [29], [30], [31], [32], [33]) support additional security properties, such as non-transferability, lazy revocation, and access hierarchies. These are not necessary for our system, but could be substituted if such properties are desirable for other reasons.

Our attack analysis and experiments do not consider the threat of end-to-end timing attacks (except that we mandate circuit lengths of at least 3 to preclude the simplest such attacks). Past works have shown that these attacks are potentially effective against Tor and other onion routing systems even when the attacker controls only a few nodes [34], [16]. The Tarzan system protects against timing attacks through generation of artificial cover traffic that masks timing patterns in a sea of mimicry and noise [35]. Future work should consider the feasibility of supplementing AnonymousCloud with similar protections.

Aside from implementing protections and protocols that directly facilitate greater privacy, mechanisms that provide greater transparency for internal cloud operations—particularly distribution and management of security-sensitive data—is critical for instilling greater confidence in end users [36], [37], [38]. Future work should therefore consider augmenting AnonymousCloud with features that afford customers greater control over data distribution and scheduling details after Tor circuit construction, and without sacrificing anonymity.

V. Conclusion

In this paper we proposed an approach to improving data privacy in the cloud by decoupling private data content from metadata concerning its provenance and semantics. Our system, AnonymousCloud, employs Tor onion routing inside cloud providers for customers to anonymously communicate computations and data to the system. An anonymous authentication system based on public-key cryptography facilitates billing of anonymous customers without linking their private data to their identities. Simulation results demonstrate that AnonymousCloud provides superior data ownership privacy even when a large percentage of the cloud is malicious.

For our future research we consider adding incentive-based congestion control to reduce the computational overhead of long Tor circuits, and cover traffic for defense against end-to-end timing attacks. In addition, greater transparency of internal cloud resources is recommended as a means of generating greater consumer confidence in cloud systems.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. NSF-0959096. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] 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," *Communications of the ACM (CACM)*, vol. 53, no. 4, pp. 50–58, 2010.

[2] Amazon, "Amazon elastic compute cloud (Amazon EC2)," http://aws.amazon.com/ec2, 2012.

[3] Microsoft, "Windows Azure: Cloud computing," http://www.windowsazure.com/, 2012.

[4] Apache, "Apache Hadoop," http://hadoop.apache.org/, 2012.

[5] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[6] A. Weiss, "Computing in the clouds," *netWorker – Cloud Computing: PC Functions Move Onto the Web*, vol. 11, no. 4, pp. 16–25, 2007.

[7] Fujitsu Research Institute, "Personal data in the cloud: A global survey of consumer attitudes," http://www.fujitsu.com/global/news/publications/dataprivacy.html, October 2010.

[8] G. Gross, "Cloud computing may draw government action," *IDG News Service*, September 2008.

[9] M. D. Ryan, "Cloud computing privacy concerns on our doorstep," *Communications of the ACM (CACM)*, vol. 54, no. 1, pp. 36–38, 2011.

[10] D. Chen and H. Zhao, "Data security and privacy protection issues in cloud computing," in *Proceedings of the International Conference on Computer Science and Electronics Engineering (ICCSEE)*, 2012, pp. 647–651.

[11] Y. Lindell and B. Pinkas, "Secure multiparty computation for privacy-preserving data mining," *Journal of Privacy and Confidentiality*, vol. 1, no. 1, pp. 59–98, 2009.

[12] I. Roy, S. T. Setty, A. Kilzer, V. Shmatikov, and E. Witchel, "Airavat: Security and privacy for MapReduce," in *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI)*, 2010, pp. 297–312.

[13] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th USENIX Security Symposium*, 2004, pp. 303–320.

[14] P. Syverson, "A taxonomy of replay attacks," in *Proceedings of the 7th IEEE Computer Security Foundations Workshop (CSFW)*, 1994, pp. 187–191.

[15] A. Pfitzmann and M. Hansen, "A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management," http://dud.inf.tu-dresden.de/Anon_Terminology.shtml, August 2010, v0.34.

[16] N. Hopper, E. Y. Vasserman, and E. Chan-Tin, "How much anonymity does network latency leak?" *ACM Transactions on Information and System Security (TISSEC)*, vol. 13, no. 2, 2010.

[17] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka, and J. Molina, "Controlling data in the cloud: Outsourcing computation without outsourcing control," in *Proceedings of the ACM Workshop on Cloud Computing Security (CCSW)*, 2009, pp. 85–90.

[18] C. Mitchell, Ed., *Trusted Computing*. London, UK: The Institution of Engineering and Technology, 2005.

[19] R. Huang, X. Gui, S. Yu, and W. Zhuang, "Research on privacy-preserving cloud storage framework supporting ciphertext retrieval," in *Proceedings of the International Conference on Network Computing and Information Security (NCIS)*, 2011, pp. 93–97.

[20] Q. Liu, G. Wang, and J. Wu, "Secure and privacy preserving keyword searching for cloud storage services," *Journal of Network and Computer Applications (JNCA)*, vol. 35, no. 3, pp. 927–933, 2012.

[21] C. Dwork, "Differential privacy: A survey of results," in *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation (TAMC)*, 2008, pp. 1–19.

[22] P. Samarati, "Protecting respondents' identities in microdata release," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 13, no. 6, pp. 1010–1027, 2001.

[23] J.-S. Xu, R.-C. Huang, W.-M. Huang, and G. Yang, "Secure document service for cloud computing," in *Proceedings of the 1st International Conference on Cloud Computing (CloudCom)*, 2009, pp. 541–546.

[24] A. Greenberg, "The Tor project's new tool aims to map out internet censorship," *Forbes*, April 2012.

[25] ExpressionTech and The Tor Project, "Tor Cloud project," https://cloud.torproject.org/, 2012.

[26] R. Laurikainen, "Secure and anonymous communication in the cloud," Aalto University School of Science and Technology, Department of Computer Science and Engineering, Tech. Rep. TKK-CSE-B10, 2010.

[27] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Communications of the ACM (CACM)*, vol. 28, no. 10, pp. 1030–1044, 1985.

[28] J. Camenisch and E. V. Herreweghen, "Design and implementation of the *idemix* anonymous credential system," in *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS)*, 2002, pp. 21–30.

[29] S. Zarandioon, D. Yao, and V. Ganapathy, "K2C: Cryptographic cloud storage with lazy revocation and anonymous access," in *Proceedings of the 7th International ICST Conference on Security and Privacy in Communication Networks (SecureComm)*, 2011, pp. 491–510.

[30] D. Slamanig, "More privacy for cloud users: Privacy-preserving resource usage in the cloud," in *Selected Papers from the 4th Hot Topics in Privacy Enhancing Technologies (HotPETs)*, 2011, pp. 15–27.

[31] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Proceedings of the 24th Annual International Cryptology Conference (CRYPTO)*, 2004, pp. 56–72.

[32] M. Jensen, S. Schäge, and J. Schwenk, "Towards an anonymous access control and accountability scheme for cloud computing," in *Proceedings of the IEEE 3rd International Conference on Cloud Computing (CLOUD)*, 2010, pp. 540–541.

[33] M. Backes, J. Camenisch, and D. Sommer, "Anonymous yet accountable access control," in *Proceedings of the ACM Workshop on Privacy in the Electronic Society (WPES)*, 2005, pp. 40–46.

[34] T. Abbott, K. Lai, M. Lieberman, and E. Price, "Browser-based attacks on Tor," in *Proceedings of the 7th International Conference on Privacy Enhancing Technologies (PET)*, 2007, pp. 184–199.

[35] M. J. Freedman and R. Morris, "Tarzan: A peer-to-peer anonymizing network layer," in *Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS)*, 2002, pp. 193–206.

[36] J. Abawajy, "Determining service trustworthiness in intercloud computing environments," in *Proceedings of the 10th International Symposium on Pervasive Systems, Algorithms, and Networks*, 2009, pp. 784–788.

[37] R. K. L. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B.-S. Lee, "TrustCloud: A framework for accountability and trust in cloud computing," in *Proceedings of the IEEE World Congress on Services (SERVICES)*, 2011, pp. 584–588.

[38] T. Nguyen and W. Shi, "Improving resource efficiency in data centers using reputation-based resource selection," in *Proceedings of the International Conference on Green Computing (GREENCOMP)*, 2010, pp. 389–396.