ISSN : 2394-2975 (Online)
ISSN : 2394-6814 (Print)

Decentralized Admittance Power with Flexible Distributed Storage Integrity Auditing Mechanism

[I]G.Thenmozhi, [II]Dr.S.Dhanalakshmi
[I]M.Phil Full Time Research Scholar, Dept. of Computer Science
[II]Head of the Department, Dept. of Computer Science
[I,II]Vivekananda College of Arts and Sciences for Women (Autonomous), Namakkal, Tamilnadu, India

Abstract
Cloud storage enables users to remotely store their data and enjoy the on-demand high quality cloud applications without the burden of local hardware and software management. Though the benefits are clear, such a service is also relinquishing users' physical possession of their outsourced data, which inevitably poses new security risks toward the correctness of the data in cloud. In order to address this new problem and further achieve a secure and dependable cloud storage service, we propose in this paper a flexible distributed storage integrity auditing mechanism. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost. The auditing result not only ensures strong cloud storage correctness guarantee, but also simultaneously achieves fast data error localization, i.e., the identification of misbehaving server. Considering the cloud data are dynamic in nature, the proposed design further supports secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append. Analysis shows the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks. We consider the problem of building a secure cloud storage service on top of a public cloud infrastructure where the service provider is not completely trusted by the customer. We describe, at a high level, several architectures that combine recent and non-standard cryptographic primitives in order to achieve our goal. We survey the benefits such an architecture would provide to both customers and service providers and give an overview of recent advances in cryptography motivated specifically by cloud storage.

Key Words
Data integrity, dependable distributed storage, error localization, data dynamics, Cloud Computing, Cryptographic Cloud Storage.

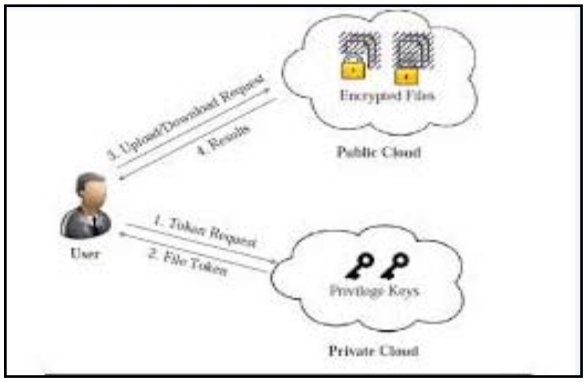
I. Introduction

The new economic and computing model is commonly referred to as cloud computing and includes various types of services such as: infrastructure as a service (IaaS), where a customer makes use of a service provider's computing, storage or networking infrastructure; platform as a service (PaaS), where a customer leverages the provider's resources to run custom applications; and _nally software as a service (SaaS), where customers use software that is run on the providers infrastructure. Cloud infrastructures can be roughly categorized as either private or public. In a private cloud, the infrastructure is managed and owned by the customer and located on-premise. In particular, this means that access to customer data is under its control and is only granted to parties it trusts. In a public cloud the infrastructure is owned and managed by a cloud service provider. This means that customer data is outside its control and could potentially be granted to untrusted parties. An important aspect of a cryptographic storage service is that the security properties described above are achieved based on strong cryptographic guarantees as opposed to legal, physical and access control mechanisms. Several trends are opening up the era of Cloud Computing, which is an Internet-based development and use of computer technology. The ever cheaper and more powerful processors, together with the software as a service (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale. The increasing network bandwidth and reliable yet flexible network connections make it even possible that users can now subscribe high quality services from data and software that reside solely on remote data centers. Moving data into the cloud offers great convenience to users since they don't have to care about the complexities of direct hardware management. The users may not retain a local copy of outsourced data, there exist various incentives for cloud service providers (CSP) to behave unfaithfully towards the cloud users regarding the status of their outsourced data. Therefore, although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, its lacking of offering strong assurance of data integrity and availability may impede its wide adoption by both enterprise and individual cloud users. The data in the cloud prohibits the direct adoption of traditional cryptographic primitives for the purpose of data integrity protection. Hence, the verification of cloud storage correctness must be conducted without explicit knowledge of the whole data files. Meanwhile, cloud storage is not just a third party data warehouse. The data stored in the cloud may not only be accessed but also be frequently updated by the users, including insertion, deletion, modification, appending, etc. It is more advantages for individual users to store their data redundantly across multiple physical servers so as to reduce the data integrity and availability threats. Thus, distributed protocols for storage correctness assurance will be of most importance in achieving robust and secure cloud storage systems. However, such important area remains to be fully explored in the literature. Recently, the importance of ensuring the remote data integrity has been highlighted by the following research works under different system and security models . In this paper, we propose an effective and flexible distributed storage verification scheme with explicit dynamic data support to ensure the correctness and availability of users' data in the cloud. We rely on erasurecorrecting code in the file distribution preparation to provide redundancies and guarantee the data dependability against Byzantine server, where a storage server may fail in arbitrary ways. In order to save the time, computation resources, and even the related online burden of users, we also provide the extension of the proposed main scheme to support third-party auditing, where users can safely delegate the

integrity checking tasks to third-party auditors and be worry-free to use the cloud storage services. Our work is among the first few ones in this field to consider distributed data storage security in Cloud Computing. Our contribution can be summarized as the following three aspects: 1) Compared to many of its predecessors, which only provide binary results about the storage status across the distributed servers, the proposed scheme achieves the integration of storage correctness insurance and data error localization, i.e., the identification of misbehaving server(s). 2) Unlike most prior works for ensuring remote data integrity, the new scheme further supports secure and efficient dynamic operations on data blocks, including: update, delete and append. 3) The experiment results demonstrate the proposed scheme is highly efficient.

II. Related Work

Cloud computing are so similar that grid security technique can be applied to cloud computing. The great contribution to Grid security. Public Key Infrastructure (PKI) is presently deployed in most grid implementations as it is perceived as a stable and mature technology which is widely supported and can be easily integrated with different applications on various platforms. The motivations for the proxy certificates which carry short-term public keys are twofold: (i) to limit exposure of long-term credentials, and (ii) to enable single sign-on (or unattended authentication) and delegation services. It is not clear, however, if the extensive use of certificates in the hierarchical PKI setting within a dynamic grid environment offers the best possible solution for public key management. Identity-Based Cryptography (IBC) is in a very quick development . Identity-Based Encryption (IBE) provides a public key encryption mechanism where a public key is an arbitrary string such as an email address or a telephone number. The corresponding private key can only be generated by a Private Key Generator (PKG) who has knowledge of a master secret. Using this construct, anyone can encrypt messages or verify signatures without prior key distribution beyond the dissemination of public parameters and the public key "strings." This is useful where the deployment of a traditional certificate authority-based PKI is inconvenient or infeasible, as IBE-based systems do not require certificate management, eliminating the need for certificate lookups and complex certificate revocation schemes. A central operational consideration of Identity-Based Cryptography is that private keys must be obtained from the PKG. Another consideration is cost: key generation can be computationally expensive. Recently, presented a dual system encryption, which opened up a new way to prove security of IBE and related encryption systems. A general framework for constructing identity-based and broadcast encryption systems, which solves the application problem of identitybased encrypted e-mail. However, the supposedly dynamic use of identity-based keys has been hindered by some traditional limitations of IBC such as key escrow and the need to distribute private keys through secure channels. The framework improves the user side performance for the current GSI authentication scheme in a considerable degree. The performance improvement is in both computation and communication. The improvement in communication due to being able to batch authentication sessions via a resource broker is significant. However, the authentication framework did not study hierarchy so that the unique Private Key Generator (PKG) becomes the bottleneck of the framework. In addition, they also proposed an interesting application of aggregate signature to save computational costs in verifying chained signatures. As with, however, each user is required to get hold of the intended communicating party's authentic certificate before a dynamic public key can be computed and used. To the best of our knowledge, there are only a few attempts to apply IBC to cloud computing. It provided federated identity management in the cloud such that each user and each server will have its own unique identity, and the identity is allocated by the system hierarchically. With this unique identity and Hierarchical Identity-Based Cryptography (HIBC), the key distribution and mutual authentication can be greatly simplified. It proposed a novel identity-based cryptographic system to avoid the complexity and management problems of certificate-based security infrastructures. However, those works did not study identity-based encryption and signature, and did not make performance analysis and simulation. In this paper, we first present the Hierarchical Architecture for Cloud Computing (HACC). Then, IdentityBased Encryption (IBE) and Identity-Based Signature (IBS) for HACC are proposed. Finally, an Authentication Protocol for Cloud Computing (APCC) is constructed based on HACC, IBE and IBS. APCC aligns well with the demands of cloud computing. Through simulation experiments, it is shown that APCC is more lightweight and efficient than SAP. The lightweight achieved on the user side is especially significant. The merit of our model in great scalability matches well with the needs of massivescale cloud.



Public cloud and Private cloud

Disadvantages of Existing System

Although the existing schemes aim at providing integrity verification for different data storage systems, the problem of supporting both public auditability and data dynamics has not been fully addressed. How to achieve a secure and efficient design to seamlessly integrate these two important components for data storage service remains an open challenging task in Cloud Computing. Especially to support block insertion, which is missing in most existing schemes.

III. Proposed System

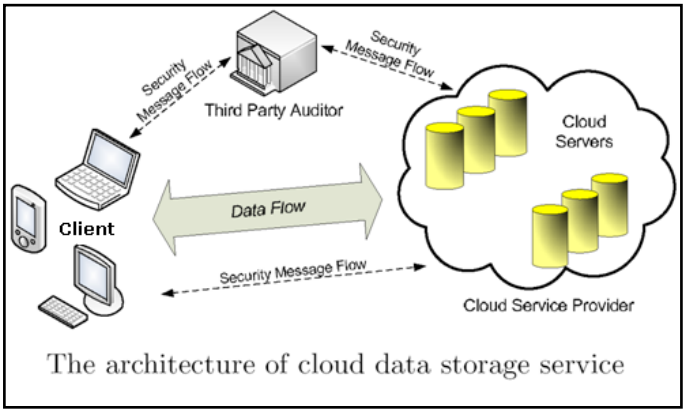
Considering the cloud data are dynamic in nature, the proposed design further supports secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append. Analysis shows the proposed scheme is highly efficient and resilient against Byzantine failure, malicious data modification attack, and even server colluding attacks. We consider the problem of building a secure cloud storage service on top of a public cloud infrastructure where the service provider is not completely trusted by the customer.

• **Client**: an entity, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations.

• **Cloud Storage Server (CSS)**: an entity, which is managed by

Cloud Service Provider (CSP), has significant storage space and computation resource to maintain the clients' data.

• **Third Party Auditor (TPA):** an entity, which has expertise and capabilities that clients do not have, is *trusted* to assess and expose risk of cloud storage services on behalf of the clients upon request.



The architecture of cloud data storage service

Advantages Of Proposed System

1) We motivate the public auditing system of data storage security in Cloud Computing, and propose a protocol supporting for fully dynamic data operations, especially to support block insertion, which is missing in most existing schemes;
2) We extend our scheme to support scalable and efficient public auditing in Cloud Computing. In particular, our scheme achieves auditing tasks from different users can be performed simultaneously by the TPA.
3) We prove the security of our proposed construction and justify the performance of our scheme through concrete implementation and comparisons

IV. Implementing The Core Components

The core components of a cryptographic storage service can be implemented using a variety of techniques, some of which were developed specifically for cloud storage. When preparing data for storage in the cloud, the data processor begins by indexing it and encrypting it with a symmetric encryption scheme (e.g., AES) under a unique key. It then encrypts the index using a searchable encryption scheme and encrypts the unique key with an attribute-based encryption scheme under an appropriate policy. Finally, it encodes the encrypted data and index in such a way that the data veri_er can later verify their integrity using a proof of storage. To enable searching over the data, the customer has to either store an index locally, or download all the (encrypted) data, decrypt it and search locally. The approach obviously negates the benefits of cloud storage while the second has high communication complexity.

A. Searchable Encryption

At a high level, a searchable encryption scheme provides a way to \encrypt" a search index so that its contents are hidden except to a party that is given appropriate tokens. More precisely, consider a search index generated over a collection. Using a searchable encryption scheme, the index is encrypted in such a way that (1) given a token for a keyword one can retrieve pointers to the encrypted _les that contain the keyword; and (2) without a token the contents of the index are hidden. This last point is worth discussing further as it is crucial to understanding the security guarantee provided by searchable encryption to the cloud provider is not leaked by the cryptographic primitives, but by the manner in which the service is being used in terms of communication and computational complexity. There are many types of searchable encryption schemes, each one appropriate to particular application scenarios.

B. Attribute-based Encryption

Another set of cryptographic techniques that has emerged recently allows the specification of a decryption policy to be associated with a cipher text. More precisely, in a (ciphertext-policy) attribute-based encryption scheme each user in the system is provided with a decryption key that has a set of attributes associated with it. A user can then encrypt a message under a public key and a policy. Decryption will only work if the attributes associated with the decryption key match the policy used to encrypt the message. Attributes are qualities of a party that can be established through relevant credentials such as being a Partner Corp employee or living in Washington State.

C. Proofs of Storage

A proof of storage is a protocol executed between a client and a server with which the server can prove to the client that it did not tamper with its data. The client begins by encoding the data before storing it in the cloud. From that point on, whenever it wants to verify the integrity of the data it runs a proof of storage protocol with the server. The main benefits of a proof of storage are that (1) they can be executed an arbitrary number of times; and (2) the amount of information exchanged between the client and the server is extremely small and independent of the size of the data. Proofs of storage can be either privately or publicly verifiable. Privately verifiable proofs of storage only allow the client to verify the integrity of the data. With a publicly verifiable proof of storage, on the other hand, anyone that possesses the client's public key can verify the data's integrity.

V. Cloud Services And Providing Dynamic Data Operation Support

A. Secure Extranet

In addition to simple storage, many enterprise customers will have a need for some associated services. These services can include any number of business processes including sharing of data among trusted partners, litigation support, monitoring and compliance, back-up, archive and audit logs. We refer to a cryptographic storage service together with an appropriate set of enterprise services as a secure extranet and believe this could provide a valuable service to enterprise customers.

B. Electronic Health Records

In February 2009, 19 billion dollars were provisioned by the U.S. government to digitize health records. This move towards electronic health records promises to reduce medical errors, save lives and decrease the cost of healthcare. Given the importance and sensitivity of health-related data, it is clear that any storage platform for health records will need to provide strong confidentiality and integrity guarantees to patients and care givers.

C. Interactive Scientific Publishing

As scientists continue to produce large data sets which have broad value for the scientific community, demand will increase for a storage infrastructure to make such data accessible and sharable.

To incent scientists to share their data, scientific societies such as the Optical Society of America are considering establishing a publication forum for data sets in partnership with industry. Such an interactive publication forum will need to provide strong guarantees to authors on how their data sets may be accessed and used by others, and could be built on a cryptographic cloud storage system like the one proposed here.

D. Providing Dynamic Data Operation Support

The cloud data storage, there are many potential scenarios where data stored in the cloud is dynamic, like electronic documents, photos, or log files etc. Therefore, it is crucial to consider the dynamic case, where a user may wish to perform various block-level operations of update, delete and append to modify the data file while maintaining the storage correctness assurance. Since data do not reside at users' local site but at cloud service provider's address domain, supporting dynamic data operation can be quite challenging. On the one hand, CSP needs to process the data dynamics request without knowing the secret keying material. On the other hand, users need to ensure that all the dynamic data operation request has been faithfully processed by CSP. To address this problem, we briefly explain our approach methodology here and provide the details later. For any data dynamic operation, the user must first generate the corresponding resulted file blocks and parities. This part of operation has to be carried out by the user, since only he knows the secret matrix P. Besides, to ensure the changes of data blocks correctly reflected in the cloud address domain, the user also needs to modify the corresponding storage verification tokens to accommodate the changes on data blocks. Only with the accordingly changed storage verification tokens, the previously discussed challenge-response protocol can be carried on successfully even after data dynamics. In other words, these verification tokens help ensure that CSP would correctly execute the processing of any dynamic data operation request. Otherwise, CSP would be caught cheating with high probability in the protocol execution later on. Given this design methodology, the straightforward and trivial way to support these operations is for user to download all the data from the cloud servers and re-compute the whole parity blocks as well as verification tokens. This would clearly be highly inefficient.

VI. Conclusion And Future Work

In this paper, we investigate the problem of data security in cloud data storage, which is essentially a distributed storage system. To achieve the assurances of cloud data integrity and availability and enforce the quality of dependable cloud storage service for users, we propose an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete, and append. We rely on erasure-correcting code in the file distribution preparation to provide redundancy parity vectors and guarantee the data dependability. By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., whenever data corruption has been detected during the storage correctness verification across the distributed servers, we can almost guarantee the simultaneous identification of the misbehaving server(s).

Considering the time, computation resources, and even the related online burden of users, we also provide the extension of the proposed main scheme to support third-party auditing, where users can safely delegate the integrity checking tasks to third-party auditors and be worry-free to use the cloud storage services. Through detailed security and extensive experiment results, we show that our scheme is highly efficient and resilient to Byzantine failure, malicious data modification attack, and even server colluding attacks.

In future, we would like to hide the attributes and access policy of a user. Our plans include an investigation into alternate CP-ABE constructs to be used with EASiER in order to achieve stronger security guarantees. Designing a more expressive scheme, which can be proved to have full security under the standard model, with better performance.

References

[1] *S. Ruj, M. Stojmenovic, and A. Nayak, "Privacy Preserving Access Control with Authentication for Securing Data in Clouds," Proc. IEEE/ACM Int'l Symp. Cluster, Cloud and Grid Computing, pp. 556-563, 2012.*

[2] *C. Wang, Q. Wang, K. Ren, N. Cao, and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing," IEEE Trans. Services Computing, vol. 5, no. 2, pp. 220-232, Apr.- June 2012.*

[3] *J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy Keyword Search Over Encrypted Data in Cloud Computing," Proc. IEEE INFOCOM, pp. 441-445, 2010.*

[4] *S. Kamara and K. Lauter, "Cryptographic Cloud Storage," Proc. 14th Int'l Conf. Financial Cryptography and Data Security, pp. 136-149, 2010.*

[5] *H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-Based Authentication for Cloud Computing," Proc. First Int'l Conf. Cloud Computing (CloudCom), pp. 157-166, 2009.*

[6] *C. Gentry, "A Fully Homomorphic Encryption Scheme," PhD dissertation, Stanford Univ., http://www.crypto.stanford.edu/craig, 2009.*

[7] *A.-R. Sadeghi, T. Schneider, and M. Winandy, "Token-Based Cloud Computing," Proc. Third Int'l Conf. Trust and Trustworthy Computing (TRUST), pp. 417-429, 2010.*

[8] *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," HP Technical Report HPL-2011-38, http://www.hpl.hp.com/techreports/ 2011/HPL-2011-38.html, 2013.*

[9] *R. Lu, X. Lin, X. Liang, and X. Shen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing," Proc. Fifth ACM Symp. Information, Computer and Comm. Security (ASIACCS), pp. 282-292, 2010.*

[10] *D.F. Ferraiolo and D.R. Kuhn, "Role-Based Access Controls," Proc. 15th Nat'l Computer Security Conf., 1992.*

[11] *D.R. Kuhn, E.J. Coyne, and T.R. Weil, "Adding Attributes to Role-Based Access Control," IEEE Computer, vol. 43, no. 6, pp. 79-81, June 2010.*

[12] *M. Li, S. Yu, K. Ren, and W. Lou, "Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-Owner Settings," Proc. Sixth Int'l ICST Conf. Security and Privacy in Comm. Networks (SecureComm), pp. 89-106, 2010.*

[13] *S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS), pp. 261-270, 2010.*

[14] G. Wang, Q. Liu, and J. Wu, "Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services," Proc. 17th ACM Conf. Computer and Comm. Security (CCS), pp. 735-737, 2010.
[15] F. Zhao, T. Nishide, and K. Sakurai, "Realizing Fine-Grained and Flexible Access Control to Outsourced Data with Attribute-Based Cryptosystems," Proc. Seventh Int'l Conf. Information Security Practice and Experience (ISPEC), pp. 83-97, 2011.
[16] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed Access Control in Clouds," Proc. IEEE 10th Int'l Conf. Trust, Security and Privacy in Computing and Communications (TrustCom), 2011.
[17] http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs- 01-en.pdf, 2013.
[18] http://securesoftwaredev.com/2012/08/20/xacml-in-the-cloud, 2013.
[19] S. Jahid, P. Mittal, and N. Borisov, "EASiER: Encryption-Based Access Control in Social Networks with Efficient Revocation," Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS), 2011.
[20] R.L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," Proc. Seventh Int'l Conf. Theory and Application of Cryptology and Information Security (ASIACRYPT), pp. 552-565, 2001.
[21] X. Boyen, "Mesh Signatures," Proc. 26th Ann. Int'l Conf. Advances in Cryptology (EUROCRYPT), pp. 210-227, 2007.
[22] D. Chaum and E.V. Heyst, "Group Signatures," Proc. Ann. Int'l Conf. Advances in Cryptology (EUROCRYPT), pp. 257-265, 1991.
[23] H.K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-Based Signatures: Achieving Attribute-Privacy and Collusion-Resistance," IACR Cryptology ePrint Archive, 2008.
[24] H.K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-Based Signatures," Topics in Cryptology - CT-RSA, vol. 6558, pp. 376-392, 2011.
[25] A. Beimel, "Secure Schemes for Secret Sharing and Key Distribution," PhD thesis, Technion, Haifa, 1996.
[26] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," Proc. Ann. Int'l Conf. Advances in Cryptology (EUROCRYPT), pp. 457-473, 2005.
[27] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. ACM Conf. Computer and Comm. Security, pp. 89-98, 2006.
[28] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," Proc. IEEE Symp. Security and Privacy, pp. 321-334, 2007.
[29] X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably Secure and Efficient Bounded Ciphertext Policy Attribute Based Encryption," Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS), pp 343-352, 2009.
[30] M. Chase, "Multi-Authority Attribute Based Encryption," Proc. Fourth Conf. Theory of Cryptography (TCC), pp. 515-534, 2007.
[31] R. C. Merkle, "Protocols for public key cryptosystems," in Proc. Of IEEE Symposium on Security and Privacy, Los Alamitos, CA, USA, 1980.
[32] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and secure sensor data storage with dynamic integrity assurance," in Proc. Of IEEE INFOCOM'09, Rio de Janeiro, Brazil, Appril 2009.
[33] J. S. Plank, S. Simmerman, and C. D. Schuman, "Jerasure: A library in C/C++ facilitating erasure coding for storage applications - Version 1.2," University of Tennessee, Tech. Rep. CS-08-627, August 2008.
[32] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in Proc. of Crypto'96, volume 1109 of LNCS. Springer-Verlag, 1996, pp. 1–15.