MAKE SAFE CLOUD DATA UNDER KEY CONTACT

[1] Dr.P.Chitti Babu, [2] M.Maheswara Reddy
[1]Professor &Principal , [2]MCA Student
[1,2]MCA Department,
[1,2]Annamacharya PG college of Computer Studies , Rajampet, Y.S.R kadapa, Andhra Pradesh, India

Abstract : In this work, we study data confidentiality against an adversary which knows the encryption key and has access to a large fraction of the cipher text blocks. The adversary can acquire the key either by exploiting flaws or backdoors in the key-generation software, or by compromising the devices that store the keys. As far as we are aware, this adversary invalidates the security of most cryptographic solutions, including those that protect encryption keys by means of secret-sharing. To counter such an adversary, I propose Bastion, a novel and efficient scheme which ensures that plaintext data cannot be recovered as long as the adversary has access to at most all but two cipher text blocks, even when the encryption key is exposed. Bastion achieves this by combining the use of standard encryption functions with an efficient linear transform. In this sense, Bastion shares similarities with the notion of all-or-nothing transform. An AONT is not an encryption by itself, but can be used as a pre-processing step before encrypting the data with a block cipher. This encryption paradigm called AON encryption was mainly intended to slow down brute-force attacks on the encryption key. However, AON encryption can also preserve data confidentiality in case the encryption key is exposed, as long as the adversary has access to at most all but one cipher text blocks.

Index Terms: Introduction, cloud computing, existing work, proposed work

I. INTRODUCTION

The main purpose of this application is providing more confidentiality data in the cloud through the proposed scheme. Also using security for data sharing in cloud with polynomial updated time keys. Ensures data confidentiality against an adversary that knows the encryption key. It prevents leakage of any plaintext block as long as the adversary has access to the encryption key. Evaluate the performance of Bastion analytically and empirically in comparison to a number of existing encryption techniques. Discuss practical insights with respect to the deployment of Bastion within existing storage systems.

II RELATED WORK

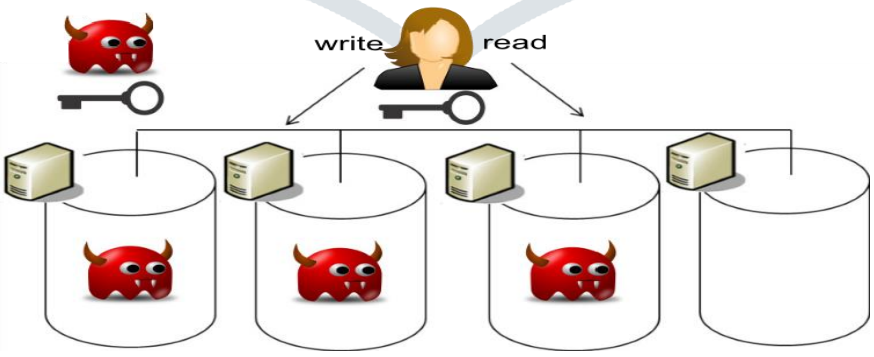


Fig 1: System Architecture

The problem of securing data stored in multicloud storage systems when the cryptographic keyis exposed. In the following, I survey relevant related work in the areas of deniable encryption, information spreading, all-or-nothing transformations, secret-sharing techniques, and leakage-resilient cryptography.

Deniable Encryption

Our work shares similarities with the view of "shared key deniable encryption". An encryption scheme is "deniable" if when pressed to disclose the encryption key the real owner exposes "fake keys" thus forcing the cipher text to "look like" the encryption of a plaintext different from the original one hence keeping the original plaintext private. Deniable encryption therefore aims to deceive an adversary which does not know the "original" encryption key but, e.g., can only acquire "fake" keys. Our security definition models an adversary that has access to the real keying material.

Information Spreading

Information spreading based on removal codes has been verified as an effective tool to provide reliability in a number of cloud-based storage systems. Erasure codes enable users to distribute their data on a number of servers and recover it despite some server's failures.

All or Nothing Transformations

All-or-nothing transformations (AONTs) were first introduced in and later studied in. The majority of AONTs leverage a secret key that is embedded in the output blocks. Once all output blocks are available, the key can be recovered and single blocks can be inverted. AONT, therefore, is not an encryption scheme and does not require the decrypt or to have any key material. combine AONT and information dispersal to provide both fault-tolerance and data secrecy, in the context of distributed storage systems. In however, an adversary which knows the encryption key can decrypt data stored on single servers.

Secret Sharing

Secret sharing schemes allow a dealer to distribute a secret among a number of shareholders, such that only authorized subsets of shareholders can reconstruct the secret. In threshold secret sharing schemes the dealer defines a threshold t and each set of shareholders of cardinality equal to or greater than t is authorized to reconstruct the secret. Secret sharing guarantees security against a non-authorized subset of shareholders; however, they incur a high computation/storage cost, which makes them impractical for sharing large files. Rabin proposed an information dispersal algorithm with smaller overhead than the one of, however the proposal in does not provide any security guarantees when a small number of shares are available. Krawczyk proposed to combine both Shamir's and Rabin's approaches; in a file is first encrypted using AES and then dispersed using the scheme in, while the encryption key is shared using the scheme in. In Krawczyk's scheme, individual cipher text blocks encrypted with AES can be decrypted once the key is exposed.

Leakage-resilient Cryptography

Leakage-resilient cryptography aims at designing cryptographic primitives that can resist an adversary which learns partial information about the secret state of a system, e.g., through side-channels. Different models allow to reason about the "leaks" of real implementations of cryptographic primitives. All of these models, however, limit in some way the knowledge of the secret state of a system by the adversary. In contrast, the adversary is given all the secret material in our model. To the best of our knowledge, this is the first work that addresses the problem of securing data stored in multi cloud storage systems when the cryptographic material is exposed. In the following, we survey relevant related work in the areas of deniable encryption, information dispersal, all-or-nothing transformations, secret-sharing techniques, and leakage-resilient cryptography.

III EXISTING WORK

An opposition prepared with the encryption key, can still compromise a single server and decrypt the cipher text blocks stored there in. If the encryption key is exposed, the only viable means to guarantee confidentiality is to limit the adversary's access to the cipher text, e.g., by spreading it across multiple administrative domains, in the hope that the adversary cannot compromise all of them .However, even if the data is encrypted and dispersed across different administrative domains, an adversary equipped with the appropriate keying material can compromise a server in one domain and decrypt ciphertext blocks stored therein.

Disadvantages

- Existing systems are not having time updated keys.
- It is not secure against attackers.
- It is not efficient.

IV PROPOSED WORK

Propose Bastion, a new and efficient system that securities data confidentiality even if the encryption key is leaked and the opposition has access to almost all cipher text blocks.I study data confidentiality against an adversary which knows the encryption key and has access to a large fraction of the cipher text blocks. The adversary can acquire the key either by exploiting flaws or backdoors in the key-generation software, or by compromising the devices that store the keys (e.g., at the user-side or in the cloud). To counter such an adversary, I propose Bastion, a novel and efficient scheme which ensures that plaintext data cannot be recovered as long as the adversary has access to at most all but two cipher text blocks, even when the encryption key is exposed.

Advantages

- ✓ Polynomial time updated keys.
- ✓ More data confidential.
- ✓ It is more efficient, secure and accurate.

PROPOSED FRAME WORK

- Data Owner
- Data User
- Admin

Data Owner

In Data Owner module, Initially Data Owner must have to register their detail and admin will approve the registration by sending signature key and private key through email. After successful login he/she have to verify their login by entering signature and private key. Then data Owner can upload files into cloud server with Polynomial key generation. He/she can view the files that are uploaded in cloud by entering the secret file key.

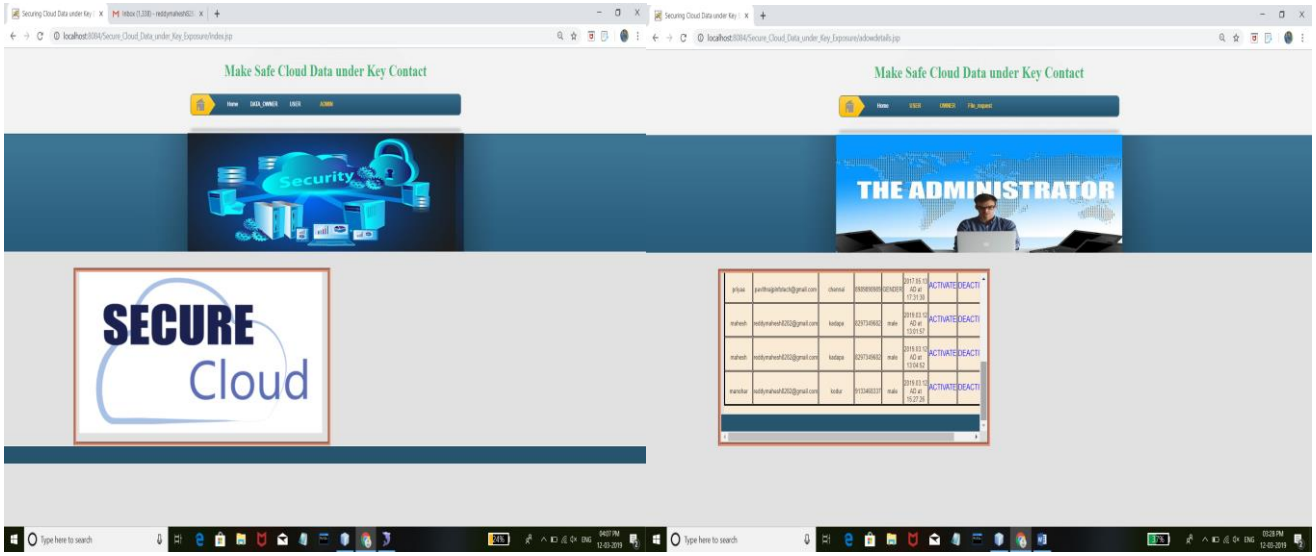
Data User

In Data User module, Initially Data Users must have to register their detail and admin will approve the registration by sending signature key and private key through email. After successful login he/she have to verify their login by entering signature and private key. Data Users can search all the files upload by data owners. He/she can send search request to admin then admin will send the search key. After entering the search key, he/she can view the file

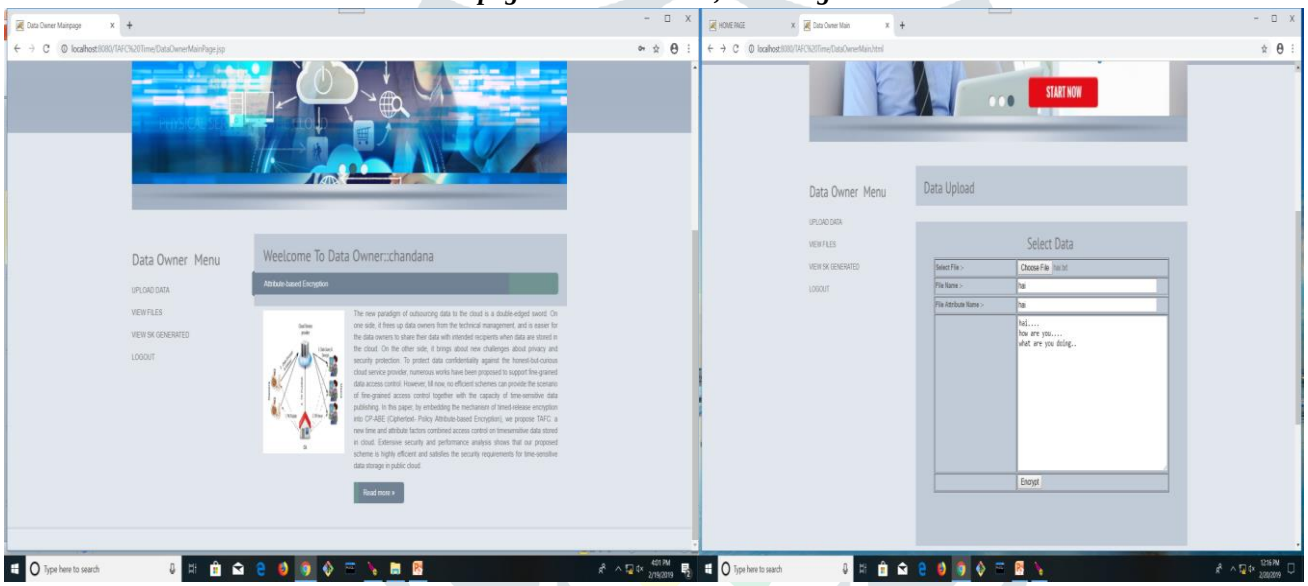
Admin

In Admin module, Admin can view all the Data owners and data user's details. Admin will approve the users and send the signature key and private key to the data owners and data users. Also admin will send the search request key to the users. Admin can able see the files in cloud uploaded by the data owners.

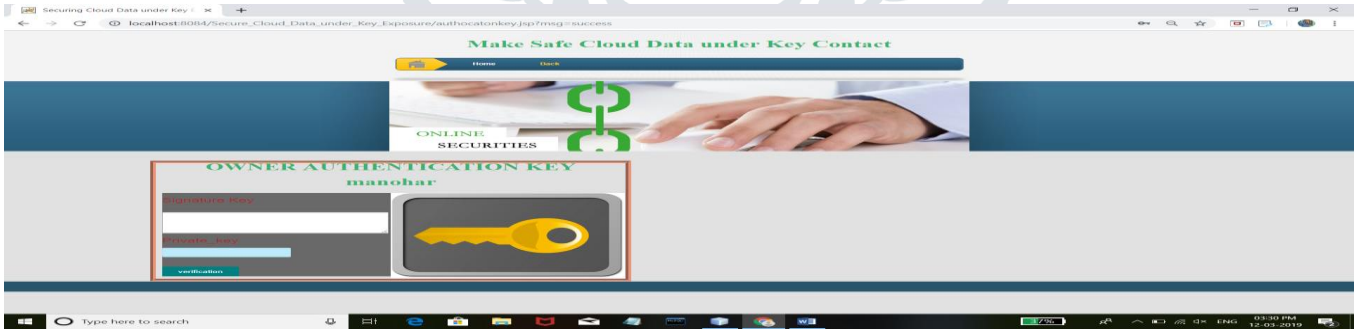
I. RESULTS



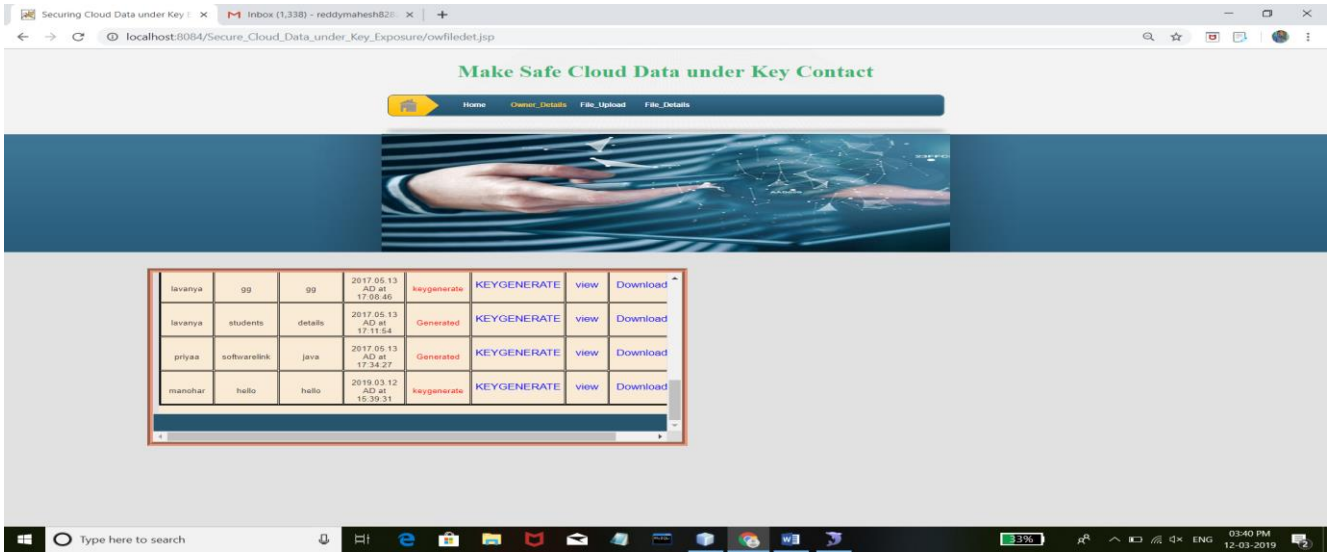
Screen 1 : home page and data owner, user registration details



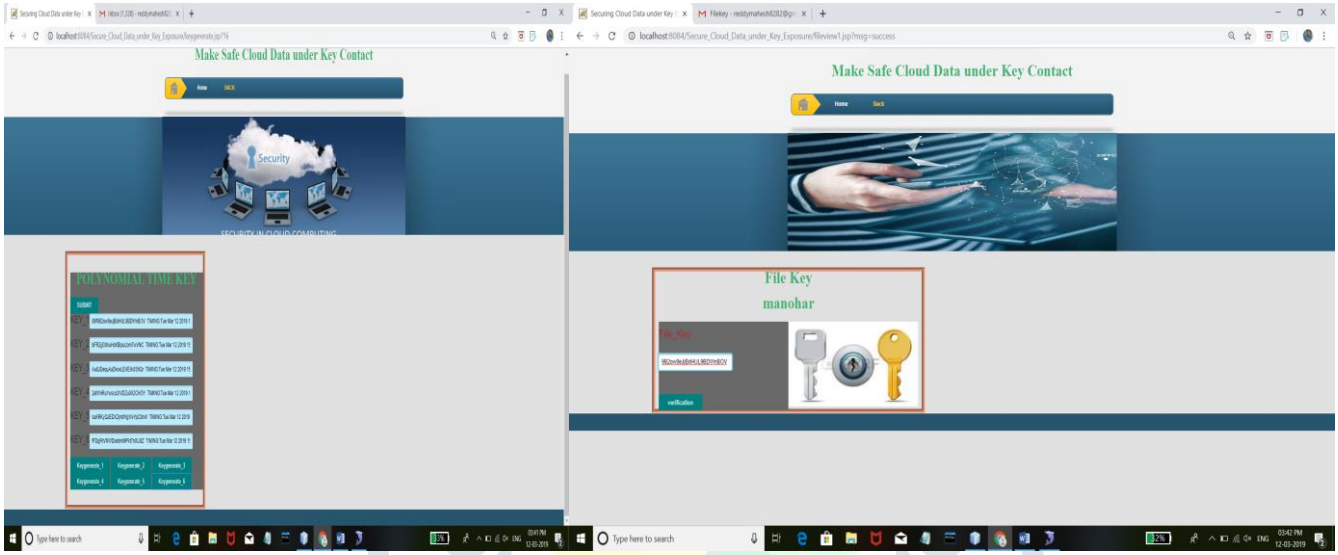
Screen 2 : home page and data uploaded window



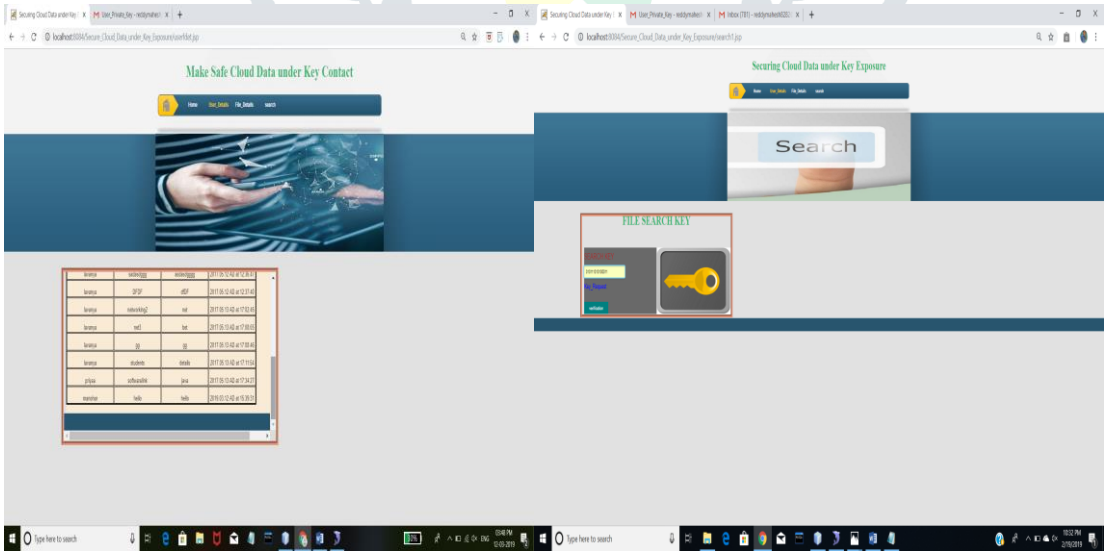
***Screen 3* Data Owner Authentication Key**



Screen 4 **Polynomial Time Keys Generation**



Screen 5 Data Owner View the File Key



Screen 6 User Search Specific File

V CONCLUSION AND FUTURE WORK

Existing AON encryption schemes, however, require at least two rounds of block cipher encryptions on the data: one preprocessing round to create the AONT, followed by another round for the actual encryption. Notice that these rounds are sequential, and cannot be parallelized. This results in considerable often unacceptable overhead to encrypt and decrypt large files. On the other hand, Bastion requires only one round of encryption which makes it well-suited to be integrated in existing dispersed storage systems. we evaluate the performance of Bastion in comparison with a number of existing encryption schemes. Our results show that Bastion only incurs a negligible performance deterioration (less than 5%) when compared to symmetric encryption schemes, and considerably improves the performance of existing AON encryption schemes. We also discuss practical insights with respect to the possible integration of Bastion in commercial dispersed storage systems. In future work Multi storage cloud can be used with high authentication oriented keys are used to encrypt and decrypt files.

REFERENCES

[1] [1] M. Abd-El-Malek, G. R. Ganger, G. R. Goodson, M. K. Reiter, and J. J. Wylie, "Fault-Scalable Byzantine Fault-Tolerant Services," in ACM Symposium on Operating Systems Principles (SOSP), 2005, pp. 59–74.

[2] M. K. Aguilera, R. Janaki Raman, and L. Xu, "Using Erasure Codes Efficiently for Storage in a Distributed System," in International Conference on Dependable Systems and Networks (DSN), 2005, pp. 336–345.

[3] W. Aiello, M. Bellare, G. D. Crescenzo, and R. Venkatesan, "Security amplification by composition: The case of doublyiterated, ideal ciphers," in Advances in Cryptology (CRYPTO), 1998, pp. 390–407.

[4] C. Basescu, C. Cachin, I. Eyal, R. Haas, and M. Vukolic, "Robust Data Sharing with Key-value Stores," in ACM SIGACTSIGOPS Symposium on Principles of Distributed Computing (PODC), 2011, pp. 221–222.

[5] A. Beimel, "Secret-sharing schemes: A survey," in International Workshop on Coding and Cryptology (IWCC), 2011, pp. 11–46.

[6] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, "DepSky: Dependable and Secure Storage in a Cloud-ofclouds," in Sixth Conference on Computer Systems (EuroSys), 2011, pp. 31–46.

[7] G. R. Blakley and C. Meadows, "Security of ramp schemes," in Advances in Cryptology (CRYPTO), 1984, pp. 242–268.

[8] V. Boyko, "On the Security Properties of OAEP as an Allor-nothing Transform," in Advances in Cryptology (CRYPTO), 1999, pp. 503–518.

[9] R. Canetti, C. Dwork, M. Naor, and R. Ostrovsky, "Deniable Encryption," in Proceedings of CRYPTO, 1997.

[10] Cavalry, "Encryption Engine Dongle," http://www. cavalrystorage.com/en2010.aspx/.

[11] C. Charnes, J. Pieprzyk, and R. Safavi-Naini, "Conditionally secure secret sharing schemes with disenrollment capability," in ACM Conference on Computer and Communications Security (CCS), 1994, pp. 89–95.

[12] A. Desai, "The security of all-or-nothing encryption: Protecting against exhaustive key search," in Advances in Cryptology (CRYPTO), 2000, pp. 359–375.

[13] C. Dubnicki, L. Gryz, L. Heldt, M. Kaczmarczyk, W. Kilian, P. Strzelczak, J. Szczepkowski, C. Ungureanu, and M. Welnicki, "HYDRAstor: a Scalable Secondary Storage," in USENIX Conference on File and Storage Technologies (FAST), 2009, pp. 197–210.

[14] M. Dürmuth and D. M. Freeman, "Deniable encryption with negligible detection probability: An interactive construction," in EUROCRYPT, 2011, pp. 610–626.

[15] EMC, "Transform to a Hybrid Cloud," http://www.emc. com/campaign/global/hybridcloud/index.htm.

[16] IBM, "IBM Hybrid Cloud Solution," http://www-01.ibm. com/software/tivoli/products/hybrid-cloud/.