A Novel and Efficient Secure Scheme for Cloud Storage Data with Key Vulnerability

Pratyush Ranjan Mohapatra[1], YallamatiSobhan Babu[2], Archana Panda[1]

[1]Assistant Professor,[2]Associate Professor,[1,2]Dept. of CSE
[1]Gandhi Institute for Technology, Bhubaneshwar, India
[2]Gandhi Engineering College, Bhubaneshwar, India

Abstract

Cloud storage in public systems has become a major issue to control data access. A pliable and most secure way to secure the data in the cloud severs for cloud storage is by using Ciphertext-Policy Attribute-Based Encryption (CP-ABE). But, in existing scheme the secure key distribution and verification is done by a single attribute authority which take a lot of time. When it is done by the single attribute authority, it is like a single-point play acquired in a big cloud storage system. There will be users who will be waiting for secret keys for a long-time which result in the less efficiency of the system. Many solutions have been proposed like multi authority scheme to overcome this problem but was unable to overcome this problem. Therefore, this articleproposed a data confidentiality against an adversary which knows the encryption key and has access to a large fraction of the ciphertext blocks. To this end, we propose Bastion, a novel and efficient scheme that guarantees data confidentiality even if the encryption key is leaked and the adversary has access to almost all ciphertext blocks. We analyze the security of Bastion, and we evaluate its performance by means of a prototype implementation. We also discuss practical insights with respect to the integration of Bastion in commercial dispersed storage systems. Our evaluation results suggest that Bastion is well-suited for integration in existing systems since it incurs less than 5% overhead compared to existing semantically secure encryption modes.

Keywords: *Key vulnerability, Cloud data, Dispersed storage.*

1. Introduction

The world recently witnessed a massive surveillance program aimed at breaking users' privacy. Perpetrators were not hindered by the various security measures deployed within the targeted services [1]. For instance, although these services relied on encryption mechanisms to guarantee data confidentiality, the necessary keying material was acquired by means of backdoors, bribe, or coercion. If the encryption key is exposed, the only viable means to guarantee confidentiality is to limit the adversary's access to the ciphertext, 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.

In this paper, we study data confidentiality against an adversary which knows the encryption key and has access to a large fraction of the ciphertext blocks. The adversary can acquire the key either by exploiting flaws or backdoors in the key-generation software [1], or by compromising the devices that store the keys (e.g., at the user-side or in the cloud). 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 (since these keys can be leaked as soon as they are generated).

To counter such an adversary, we propose Bastion, a novel and efficient scheme which ensures that plaintext data cannot be recovered if the adversary has access to at most all but two ciphertext 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, if the adversary has access to at most all but one ciphertext blocks. 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. Our main contributions include:

- We evaluate the performance of Bastion in comparison with several 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. Bastion an efficient scheme that ensures data confidentiality against an adversary that knows the encryption key and has access to a large fraction of the ciphertext blocks.
- This article analyzes the security of Bastion and shown that it prevents leakage of any plaintext block if the adversary has access to the encryption key and to all but two ciphertext blocks.Further, evaluated the performance of Bastion analytically and empirically in comparison to several existing encryption techniques.
- Experimental results show that Bastion considerably improves (by more than 50%) the performance of existing AON encryption schemes, and only incurs a negligible overhead when compared to existing semantically secure encryption modes (e.g., the CTR encryption mode).
- In addition, practical insights are discussed with respect to the deployment of Bastion within existing storage systems, such as the HYDRAstor grid storage system.

2. Related Work

Our work addresses the problem of securing data stored in multicloud 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.

2.1. Deniable Encryption

Our work shares similarities with the notion of “sharedkey deniable encryption” [2-4]. An encryption scheme is “deniable” if—when coerced to reveal the encryption key—the legitimate owner reveals “fake keys” thus forcing the ciphertext 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.

2.2. Information Dispersal

Information dispersal based on erasure codes [5] has been proven as an effective tool to provide reliability in a number of cloud-based storage systems [6-9]. Erasure codes enable users to distribute their data on a number of servers and recover it despite some servers' failures. Ramp schemes [10] constitute a trade-off between the security guarantees of secret sharing and the efficiency of information dispersal algorithms. A ramp scheme achieves higher "code rates" than secret sharing andfeatures two thresholds t1,t2. At least t2 shares are required to reconstruct the secret and less than t1 shares provide no information about the secret; a number of shares between t1 and t2 leak "some" information.

2.3. All or Nothing Transformations

All-or-nothing transformations (AONTs) were first introduced in [11] and later studied in [12, 13]. 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 decryptor to have any key material. Resch et al. [14] combine AONT and information dispersal to provide both fault-tolerance and data secrecy, in the context of distributed storage systems. In [14], however, an adversary which knows the encryption key can decrypt data stored on single servers.

2.4. Secret Sharing

Secret sharing schemes [15] allow a dealer to distribute a secret among several shareholders, such that only authorized subsets of shareholders can reconstruct the secret. In threshold secret sharing schemes [16, 17], 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 [18] proposed an information dispersal algorithm with smaller overhead than the one of [17], however the proposal in [18] does not provide any security guarantees when a small number of shares (less than the reconstruction threshold) are available. Krawczyk [19] proposed to combine both Shamir's [17] and Rabin's [18] approaches; in [19] a file is first encrypted using AES and then dispersed using the scheme in [18], while the encryption key is shared using the scheme in [17]. In Krawczyk's scheme, individual ciphertext blocks encrypted with AES can be decrypted once the key is exposed.

2.5. 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 [20]. Different models allow to reason about the "leaks" of real implementations of cryptographic primitives [20]. All 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.

3. Proposed System

This section describes the proposed data confidentiality against an adversary which knows the encryption key and has access to a large fraction of the ciphertext 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, we propose Bastion, a novel and efficient scheme which ensures that plaintext data cannot be recovered if the adversary has access to at most all but *two* ciphertext 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 paradigmcalled 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, if the adversary has access to at most all but one ciphertext blocks.

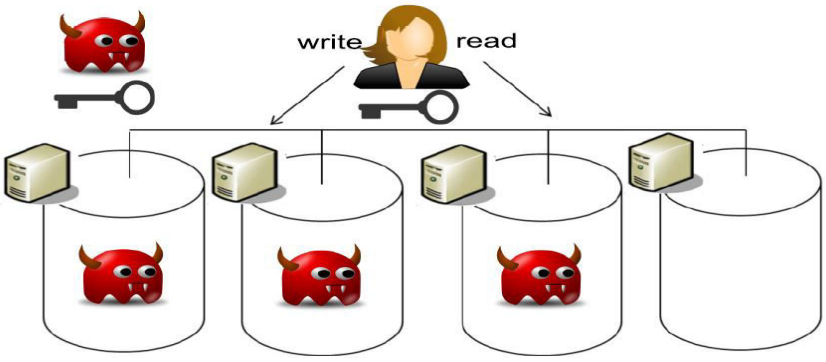


Figure 1. Proposed architecture

3.1. Implementation

1. Data Owner
2. Data User
3. 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

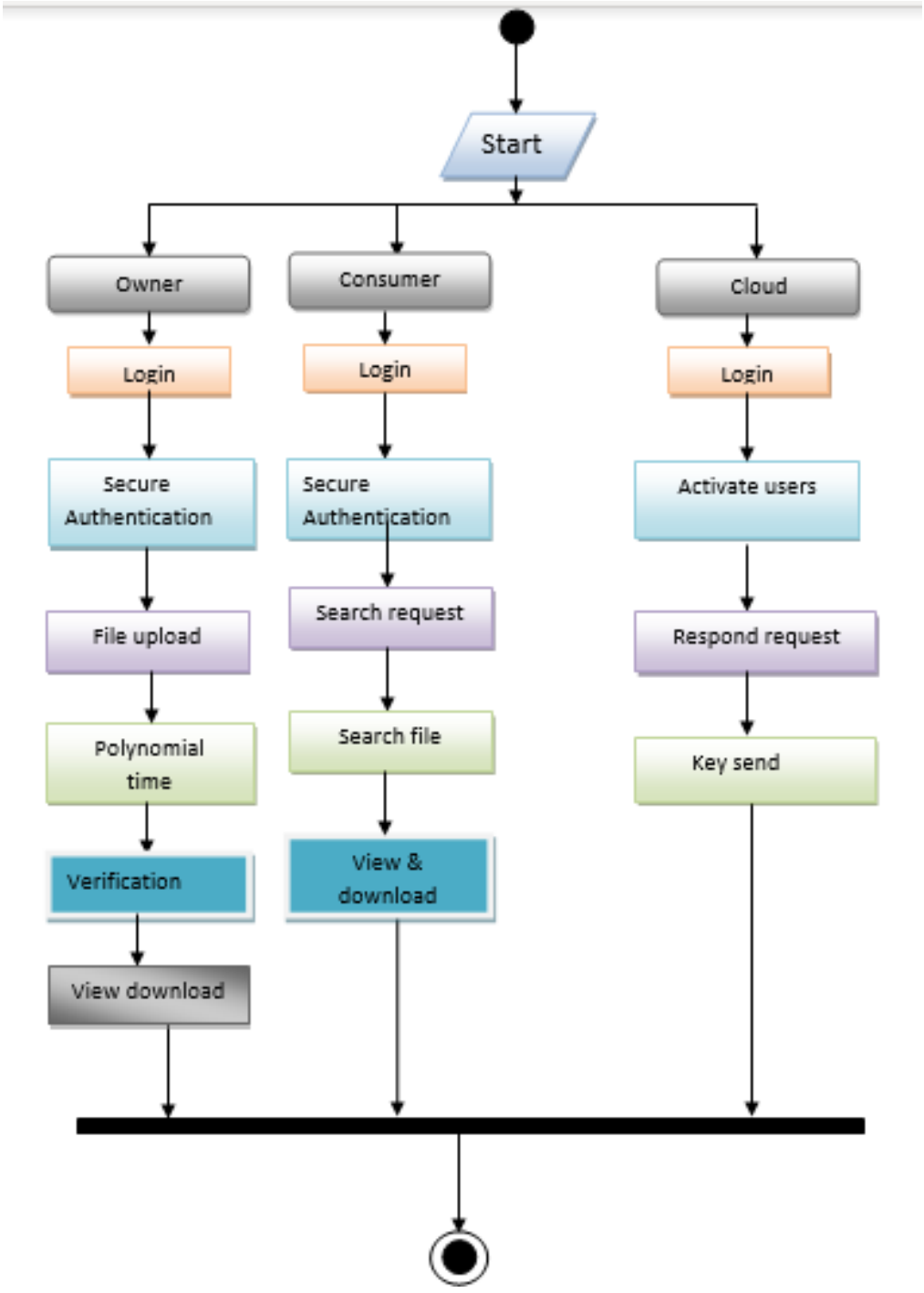
In Data User module, Initially Data Users must have to register their detail and admin will approve the registration by sending signature key and private key through email. After successful login he/she have to verify their login by entering signature and private key. Data Users can search all the files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

3.2. Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



4. Conclusions

This article addressed the problem of securing data outsourced to the cloud against an adversary which has access to the encryption key. For that purpose, a novel security definition is innovated which captures data confidentiality against the new adversary. Then Bastion is proposed, which ensures the confidentiality of encrypted data even when the adversary has the encryption key, and all but *two* ciphertext blocks. Bastion is most suitable for settings where the ciphertext blocks are stored in multi-cloud storage systems.

In these settings, the adversary would need to acquire the encryption key, and to compromise *all* servers, in order to recover any single block of plaintext. Further, security of Bastion is analyzed and evaluated its performance in realistic settings. Bastion considerably improves (by more than 50%) the performance of existing primitives which offer comparable security under key exposure, and only incurs a negligible overhead (less than 5%) when compared to existing semantically secure encryption modes (e.g., the CTR encryption mode). Finally, we showed how Bastion can be practically integrated within existing dispersed storage systems.

Reference

[1] Wikipedia, "Edward Snowden," http://en.wikipedia.org/ wiki/Edward_Snowden#Disclosure.

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

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

[4] M. Klonowski, P. Kubiak, and M. Kutylowski, "Practical Deniable Encryption," in Theory and Practice of Computer Science (SOFSEM), 2008, pp. 599–609.

[5] J. H. van Lint, Introduction to Coding Theory. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 1982.

[6] 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.

[7] M. K. Aguilera, R. Janakiraman, 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.

[8] J. Kubiatowicz, D. Bindel, Y. Chen, S. E. Czerwinski, P. R. Eaton, D. Geels, R. Gummadi, S. C. Rhea, H. Weatherspoon, W. Weimer, C. Wells, and B. Y. Zhao, "OceanStore: An Architecture for Global-Scale Persistent Storage," in International Conference on Architectural Support for Programming Languages and Operating Systems, 2000, pp. 190-201.

[9] H. Xia and A. A. Chien, "RobuSTore: a Distributed Storage Architecture with Robust and High Performance," in ACM/IEEE Conference on High Performance Networking and Computing (SC), 2007, p. 44.

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

[11] R. L. Rivest, "All-or-Nothing Encryption and the Package Transform," in International Workshop on Fast Software Encryption (FSE), 1997, pp. 210–218.

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

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

[14] J. K. Resch and J. S. Plank, "AONT-RS: Blending Security and Performance in Dispersed Storage Systems," in USENIX Conference on File and Storage Technologies (FAST), 2011, pp. 191–202.

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

[16] 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.

[17] A. Shamir, "How to Share a Secret?" in Communications of the ACM, 1979, pp. 612–613.

[18] M. O. Rabin, "Efficient dispersal of information for security, load balancing, and fault tolerance," J. ACM, vol. 36, no. 2, pp. 335–348, 1989.

[19] H. Krawczyk, "Secret Sharing Made Short," in Advances in Cryptology (CRYPTO), 1993, pp. 136–146.

[20] S. Micali and L. Reyzin, "Physically observable cryptography (extended abstract)," in Theory of Cryptography Conference (TCC), 2004, pp. 278–296.